CSC563 Multithreaded Distributed Programming Assignment 2: Building HA Cluster

Jason Taylor
Southern Connecticut State University, Graduate Student
New Haven, Connecticut
taylorj13@southernct.edu

***Abstract*— For this project, I learned how to build a virtual cluster using CentOS 7. Using the Pacemaker software solution, I designed and built a virtual cluster of 5 nodes with one data node.**

Keywords—High-Availability, PCS, Cluster

I. Introduction

In this paper, I will discuss the design and system structure for the Pacemaker virtual cluster. This includes the host machine configuration and the virtual machine configuration settings. This report also discusses the implementation of the cluster with a step-by-step setup of the virtual cluster and data node. I also show the readers the results of adding/removing nodes and node failure tests. At the end of this paper, I analyze the results of the experiments run during this project. This project had the following requirements.

- The cluster should have at least four nodes.
- The cluster should have global shared storage space.
- The cluster should be able to host a website.
- Show how to add/remove a node from the cluster.
- We need to simulate the situation where one of the nodes fails.

II. Design and System Structure

System Architecture:

- Four nodes
- Shared Storage data node

Host machine configuration:

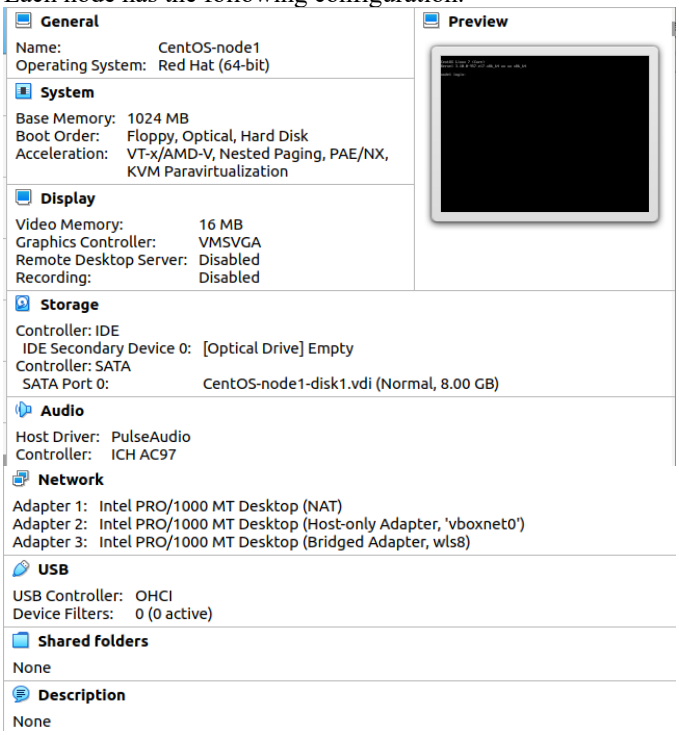
Dell Inspiron 17 7000 Series

- 16Gb ram
- CPU: Intel Core i7-7500U 4 Cores
- GPU: NVIDIA GeForce 940MX (2GB GDDR5)

Virtual Machine configuration:

Hypervisor: Oracle VirtualBox 6.1

Each node has the following configuration.



III. Cluster Implementation

Update Network:

cd /etc/sysconfig/network-scripts
cp ifcfg-enp0s3 ifcfg-enp0s8

Update ifcfg-enp0s3

```
[root@localhost network-scripts]# cat ifcfg-enp0s3
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp0s3
UUID=772cbf25-2c82-418e-9165-7e640ab20d2d
DEVICE=enp0s3
ONBOOT=yes
[root@localhost network-scripts]#
```

Update ifcfg-enp0s8

```
[root@localhost network-scripts]# cat ifcfg-enp0s8
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=enp0s8
IPADDR=10.0.0.118
DEVICE=enp0s8
ONBOOT=yes
[root@localhost network-scripts]#
```

Rename hostname

echo node5.mycluster.localdomain > /etc/hostname

Update /etc/hosts

```
[root@localhost network-scripts]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

10.0.0.110  node1 node1.mycluster.localdomain
10.0.0.111  node2 node2.mycluster.localdomain
10.0.0.112  node3 node3.mycluster.localdomain
10.0.0.113  node4 node4.mycluster.localdomain
10.0.0.118  node5 node5.mycluster.localdomain
10.0.0.114  datanode datanode.mycluster.localdomain
```

Reboot node:

shutdown now -r

Install Required packages and setup:

yum install pcs

```
Installed:
  pcs.x86_64 0:0.9.169-3.el7.centos.1
```

Setup firewall:

firewall-cmd --permanent --add-service=high-availability &&
firewall-cmd --reload

```
[root@node5 ~]# firewall-cmd --permanent --add-service=high-availability &&  firewall-cmd --reload
Warning: ALREADY_ENABLED: high-availability
success
success
```

Enable and start PCS services

systemctl start pcsd.service
systemctl enable pcsd.service

```
[root@node5 ~]# systemctl start pcsd.service;systemctl enable pcsd.service;systemctl status pcsd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to /usr/lib/systemd/system/pcsd.service.
● pcsd.service - PCS GUI and remote configuration interface
   Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-10-31 14:13:45 EDT; 117ms ago
     Docs: man:pcsd(8)
           man:pcs(8)
 Main PID: 14637 (pcsd)
   CGroup: /system.slice/pcsd.service
           ├─14637 /usr/bin/ruby /usr/lib/pcsd/pcsd
           └─14653 /usr/bin/python2 -Es /usr/sbin/pcs status nodes corosync

Oct 31 14:13:45 node5.mycluster.localdomain systemd[1]: Starting PCS GUI and remote configuration interface...
Oct 31 14:13:45 node5.mycluster.localdomain systemd[1]: Started PCS GUI and remote configuration interface.
```

Disable STONITH

```
[root@node1 ~]# pcs property set stonith-enabled=false
[root@node1 ~]# crm_verify -L
[root@node1 ~]#
[root@node1 ~]#
[root@node1 ~]#
```

Virtual IP address:

pcs resource create vClusterIP ocf:heartbeat:IPaddr2 ip=10.0.0.115 cidr_netmask=32 nic=enp0s8 op monitor interval=30s

Install Apache HTTP Server as a cluster Service

yum install -y httpd wget

```
Installed:
  httpd.x86_64 0:2.4.6-97.el7.centos.1                          wget.x86_64 0:1.14-18.el7_6.1
```

Configure firewall for http traffic:

```
[root@node5 ~]# firewall-cmd --permanent --add-service=http
success
[root@node5 ~]# firewall-cmd --reload
success
```

Create Website:

```
[root@node5 ~]# cat <<-END>/var/www/html/index.html
>   <html>
>   <body>My Test Site - $(hostname)</body>
>   </html>
> END
```

Enable the Apache status URL

```
[root@node5 ~]# cat <<-END >/etc/httpd/conf.d/status.conf
>  <Location /server-status>
>  SetHandler server-status
>  Require local
>  </Location>
> END
```

Create Resource Website:

```
pcs resource create WebSite ocf:heartbeat:apache \
 configfile=/etc/httpd/conf/httpd.conf \
 statusurl="http://localhost/server-status" \
 op monitor interval=1min
```

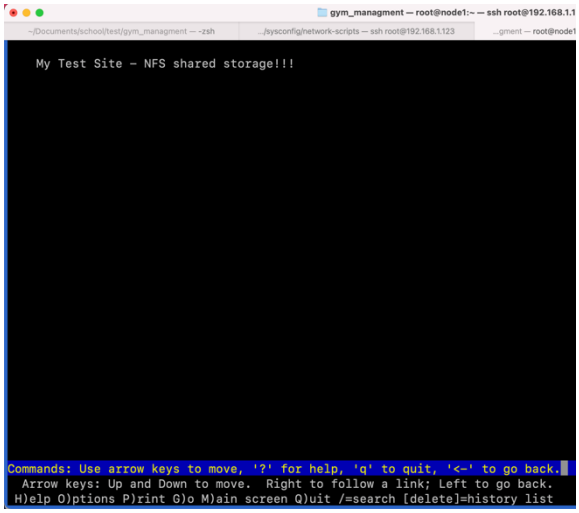
Ensure Resources Run on the same host:
pcs constraint colocation add WebSite with vClusterIP INFINITY

Set resources Start/Stop order:
pcs constraint order vClusterIP then WebSite

PCS Cluster:

```
[root@node1 ~]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node2 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Sun Oct 31 14:15:38 2021
Last change: Sat Oct 30 13:44:20 2021 by hacluster via crmd on node1

4 nodes configured
2 resource instances configured

Online: [ node1 node2 node3 node4 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node2
 WebSite        (ocf::heartbeat:apache):        Started node2

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

Verify website
lynx http://10.0.0.115/index.html



NSF Setup:

Create NFS server by cloning node1 and updating network, hostname, and /etc/hosts file.

On all node install:

yum install nfs-utils

Then update the firewall.

firewall-cmd --permanent --add-service=nfs && firewall-cmd –reload

On the data node:

systemctl enable nfs-server
systemctl start nfs-server

mkdir /var/nfsshare

vi /etc/exports

```
/var/nfsshare 10.0.0.0/24(rw,sync,no_root_squash)
~
```

Reload the exports
exportfs -r

Update the firewall:

```
[root@datanode ~]# firewall-cmd --permanent --add-service=rpc-bind && firewall-cmd --reload
Warning: ALREADY_ENABLED: rpc-bind
success
success
[root@datanode ~]# firewall-cmd --permanent --add-service=mountd && firewall-cmd --reload
Warning: ALREADY_ENABLED: mountd
success
success
[root@datanode ~]#
```

On the nodes

```
[root@node5 ~]# showmount -e datanode
Export list for datanode:
/var/nfsshare 10.0.0.0/24
[root@node5 ~]#
```

```
[root@node5 ~]# showmount -e datanode
Export list for datanode:
/var/nfsshare 10.0.0.0/24
[root@node5 ~]# mount datanode:/var/nfsshare /var/www
[root@node5 ~]# df -hT |grep nfs
datanode:/var/nfsshare  nfs4      6.2G  1.3G  5.0G  21% /var/www
[root@node5 ~]#
```

Enable clients to use NFS for web services:

setsebool httpd_use_nfs on

```
[root@node5 ~]# getsebool -a | grep httpd_use_nfs
httpd_use_nfs --> on
[root@node5 ~]#
```

Edit /etc/fstab to make the NFS mount permanent.

```
#
# /etc/fstab
# Created by anaconda on Tue Feb 26 12:31:34 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root /                       xfs     defaults        0 0
UUID=b11fd697-6232-4a6d-a502-c44dcecb610d /boot                   xfs     defaults        0 0
/dev/mapper/centos-swap swap                    swap    defaults        0 0
~
```

Copy files to NFS share and update website:

cp -r /var/www/* /var/nfsshare/

vi /var/www/html/index.html

```
<html>
<body>My Test Site - NFS shared storage!!!</body>
</html>
~
```

PCS Status:

```
[root@node1 ~]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node2 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Tue Nov  2 19:21:02 2021
Last change: Sat Oct 30 13:44:20 2021 by hacluster via crmd on node1

4 nodes configured
2 resource instances configured

Online: [ node1 node2 node3 node4 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node2
 WebSite        (ocf::heartbeat:apache):        Started node2

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
[root@node1 ~]#
```

IV. Experiment

Add/Remove nodes

On a node in the existing cluster, you need to run the following commands.

First, you need setup authentication for the hacluster user on the new node:

```
[root@node1 ~]# pcs cluster auth node5 -u hacluster -p Gi926ToR@WoRk --force
node5: Authorized
```

Then you can add the node to the cluster. This action adds the new node to the existing cluster. The command syncs the cluster configuration file corosync.cnfg to all the nodes in the cluster [1].

```
[root@node1 ~]# pcs cluster node add node5 --start --enable
Disabling SBD service...
node5: sbd disabled
Sending remote node configuration files to 'node5'
node5: successful distribution of the file 'pacemaker_remote authkey'
node1: Corosync updated
node2: Corosync updated
node3: Corosync updated
node4: Corosync updated
Setting up corosync...
node5: Succeeded
node5: Cluster Enabled
node5: Starting Cluster (corosync)...
Starting Cluster (pacemaker)...
Synchronizing pcsd certificates on nodes node5...
node5: Success
Restarting pcsd on the nodes in order to reload the certificates...
node5: Success
```

PCS Status after adding the new node.

```
[root@node1 ~]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node2 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Tue Nov  2 19:29:06 2021
Last change: Tue Nov  2 19:26:32 2021 by hacluster via crmd on node2

5 nodes configured
2 resource instances configured

Online: [ node1 node2 node3 node4 node5 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node2
 WebSite        (ocf::heartbeat:apache):        Started node2

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
[root@node1 ~]#
```

To remove the node.

The following command shuts down the specified node and removes it from the cluster configuration file /etc/corosync/corosync.conf. The command also removes the node from all the other nodes in the cluster [1].

```
[root@node1 ~]# pcs cluster node remove node5
node5: Stopping Cluster (pacemaker)...
node5: Successfully destroyed cluster
node1: Corosync updated
node2: Corosync updated
node3: Corosync updated
node4: Corosync updated
[root@node1 ~]#
```

Standby mode:

The following commands will put a specified node in and out of standby mode. The selected node in the standby command will no longer be able to host resources [1].

```
[root@node1 corosync]# pcs cluster standby node2
[root@node1 corosync]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node2 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Tue Nov  2 20:04:05 2021
Last change: Tue Nov  2 20:03:57 2021 by root via cibadmin on node1

4 nodes configured
2 resource instances configured

Node node2: standby
Online: [ node1 node3 node4 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node1
 WebSite        (ocf::heartbeat:apache):        Started node1

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

To put the node back into the cluster, run the following command after putting it into standby mode.

pcs cluster unstandby node_name

```
[root@node1 corosync]# pcs cluster unstandby node2
[root@node1 corosync]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node2 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Tue Nov  2 20:09:41 2021
Last change: Tue Nov  2 20:09:35 2021 by root via cibadmin on node1

4 nodes configured
2 resource instances configured

Online: [ node1 node2 node3 node4 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node1
 WebSite        (ocf::heartbeat:apache):        Started node1

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

Testing Nodes failure:

PCS status before shutdown node1

```
[root@node1 corosync]# pcs cluster unstandby node2
[root@node1 corosync]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node2 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Tue Nov  2 20:09:41 2021
Last change: Tue Nov  2 20:09:35 2021 by root via cibadmin on node1

4 nodes configured
2 resource instances configured

Online: [ node1 node2 node3 node4 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node1
 WebSite        (ocf::heartbeat:apache):        Started node1

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

On node1, I ran shutdown now -r to reboot the node. From the screenshot of the log below, you can see that node1 was lost.

```
  info: pcmk_quorum_notification:     Quorum retained | membership=212 members=3
notice: crm_update_peer_state_iter:   Node node1 state is now lost | nodeid=1 previous=member source=crm_reap_unseen_nodes
  info: pcmk_quorum_notification:     Quorum retained | membership=212 members=3
notice: crm_update_peer_state_iter:   Node node1 state is now lost | nodeid=1 previous=member source=crm_reap_unseen_nodes
  info: peer_update_callback:  Cluster node node1 is now lost (was member)
```

PCS Status after node1 went down.

```
[root@node4 ~]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node2 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Tue Nov  2 20:16:00 2021
Last change: Tue Nov  2 20:09:36 2021 by root via cibadmin on node1

4 nodes configured
2 resource instances configured

Online: [ node2 node3 node4 ]
OFFLINE: [ node1 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node2
 WebSite        (ocf::heartbeat:apache):        Started node2

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
[root@node4 ~]#
```

Now I will shut down node2.

After shutdown node2, the entire cluster has stopped.

```
[root@node4 ~]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node4 (version 1.1.23-1.el7_9.1-9acf116022) - partition WITHOUT quorum
Last updated: Tue Nov  2 20:33:41 2021
Last change: Tue Nov  2 20:09:36 2021 by root via cibadmin on node1

4 nodes configured
2 resource instances configured

Online: [ node3 node4 ]
OFFLINE: [ node1 node2 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Stopped
 WebSite        (ocf::heartbeat:apache):        Stopped

Failed Resource Actions:
* WebSite_start_0 on node3 'unknown error' (1): call=12, status=Timed Out, exitreason='',
    last-rc-change='Tue Nov  2 20:28:41 2021', queued=0ms, exec=40005ms

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

Reviewing the logs, I have found the following that states node2 is down.

```
info: pcmk_cpg_membership:   Group crmd event 6: node2 (node 2 pid 3893) left via cpg_leave
info: crm_update_peer_proc:  pcmk_cpg_membership: Node node2[2] - corosync-cpg is now offline
info: peer_update_callback:  Client node2/peer now has status [offline] (DC=node2, changed=4000000)
tice: peer_update_callback:  Our peer on the DC (node2) is dead
```

Digging deeper in the logs, I have found the reason the cluster stopped.

```
 warning: cluster_status:        Fencing and resource management disabled due to lack of quorum
    info: determine_online_status:       Node node4 is online
    info: determine_online_status:       Node node3 is online
 warning: unpack_rsc_op_failure: Processing failed start of WebSite on node3: unknown error | rc=1
    info: unpack_node_loop:      Node 4 is already processed
    info: unpack_node_loop:      Node 3 is already processed
    info: unpack_node_loop:      Node 4 is already processed
    info: unpack_node_loop:      Node 3 is already processed
    info: common_print:  vClusterIP      (ocf::heartbeat:IPaddr2):       Stopped
    info: common_print:  WebSite (ocf::heartbeat:apache):        Stopped
    info: pe_get_failcount:      WebSite has failed INFINITY times on node3
 warning: check_migration_threshold:     Forcing WebSite away from node3 after 1000000 failures (max=1000000)
  notice: LogAction:       * Start      vClusterIP      ( node4 )   due to no quorum (blocked)
  notice: LogAction:       * Start      WebSite         ( node4 )   due to no quorum (blocked)
```

Since the cluster no longer had quorum pcs, shut down the cluster.

To bring the cluster online again, the administrator could shut down the online nodes and start the cluster again.

```
[root@node4 ~]# pcs cluster stop --all
node2: Stopping Cluster (pacemaker)...
node1: Stopping Cluster (pacemaker)...
node3: Stopping Cluster (pacemaker)...
node4: Stopping Cluster (pacemaker)...
node1: Stopping Cluster (corosync)...
node2: Stopping Cluster (corosync)...
node4: Stopping Cluster (corosync)...
node3: Stopping Cluster (corosync)...
[root@node4 ~]# pcs cluster start --all
node1: Starting Cluster (corosync)...
node3: Starting Cluster (corosync)...
node2: Starting Cluster (corosync)...
node4: Starting Cluster (corosync)...
node4: Starting Cluster (pacemaker)...
node3: Starting Cluster (pacemaker)...
node1: Starting Cluster (pacemaker)...
node2: Starting Cluster (pacemaker)...
```

PCS Status after starting all the nodes again:

```
[root@node4 ~]# pcs status
Cluster name: pacemaker1
Stack: corosync
Current DC: node1 (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Tue Nov  2 21:00:53 2021
Last change: Tue Nov  2 20:09:36 2021 by root via cibadmin on node1

4 nodes configured
2 resource instances configured

Online: [ node1 node2 node3 node4 ]

Full list of resources:

 vClusterIP     (ocf::heartbeat:IPaddr2):       Started node1
 WebSite        (ocf::heartbeat:apache):        Started node1

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

V. Analysis

For this project, I did several experiments to understand better how the Pacemaker software worked. During these experiments, I added and removed nodes from the cluster, put them in standby mode, took them out of standby mode, and performed node failure tests. As a system administrator administering a PCS cluster, adding and removing nodes would be a common occurrence. Suppose there was an issue with one node, the administrator would need to remove that broken node from the cluster and add a new working node back. To add a new node to the cluster, the required software must first be installed on that node [1]. In addition to the software install, the firewall must also be set up to allow the high availability service [1]. Then the password needs to be set for the hacluster user. According to the Redhat documentation, it is recommended that this password be the same for each node in the cluster [1]. Next, the pcsd service should be started and then enabled [1]. The next step in adding a new node to the cluster is authenticating the hacluster user to the new cluster node [1]. Once the user is authenticated, the node can be added to the cluster. The pcs cluster node add command will add the node to the existing cluster. This command will also sync the cluster configuration file to all the nodes in the cluster [1]. The administrator would run the pcs command "pcs cluster node remove nodename" to remove an existing cluster. This command would shut down the specified node and remove it from the cluster [1]. The cluster configuration file would be updated on the remaining nodes in the cluster [1]. Another action an administrator would apply to the cluster is putting nodes in and out of standby mode. One reason this would be used is for maintenance. Putting a node in standby mode will make the specified node unable to host resources [1]. This would allow the admin to update the nodes while the node is no longer active in the cluster. The standby command could also be used to test node failure without shutdown the node [1]. Once the maintenance or testing has been completed, the node can be removed from standby mode by running "pcs cluster unstanby nodename" command [1]. During the node failure test, I discovered that after shutting down two of the nodes in the cluster, the entire cluster was shut down. Reviewing the logs, I found the reason this happened was because of quorum. To maintain cluster integrity and high availability, clustered environments use a concept known as quorum to prevent data corruption [2]. A cluster has quorum when more than half the clusters nodes are available [2]. To avoid the possibility of data corruption due to failure, Pacemaker by default stops all resources in the cluster if the cluster no longer has quorum [2]. Quorum is established using a voting system. When a cluster's node is no longer functioning or loses communication with the rest of the cluster, the other nodes can vote to remove that node from the cluster [2]. We can lose one node in our four-node cluster without stopping the cluster since there is still quorum. But once the second node is removed, the cluster no longer has quorum and stops the cluster. One problem that can exist with a cluster partition is split-brain [3]. A split-brain can occur when the cluster communication between nodes is lost, and the cluster becomes partitioned into subclusters, and each subcluster believes it is the primary partition. The subcluster is not aware of the other subcluster and causes conflicts in shared resources [3]. A split-brain can be avoided by giving each node in the cluster one vote and mandating a majority of votes in the cluster for an operational cluster [3]. A cluster partition with a majority of the votes has quorum and is allowed to operate [3].

References

[1]4. Nodes, "4.4. Managing Cluster Nodes Red Hat Enterprise Linux 7 | Red Hat Customer Portal", *Red Hat Customer Portal*, 2021. [Online]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_reference/s1-clusternodemanage-haar. [Accessed: 08- Nov- 2021].
[2]"Chapter 1. High Availability Add-On Overview Red Hat Enterprise Linux 8 | Red Hat Customer Portal", *Red Hat Customer Portal*, 2021. [Online]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_high_availability_clusters/assembly_overview-of-high-availability-configuring-and-managing-high-availability-clusters#quorum. [Accessed: 08- Nov- 2021].
[3]"Quorum Devices (Sun Cluster Overview for Solaris OS)", *Docs.oracle.com*, 2021. [Online]. Available: https://docs.oracle.com/cd/E19787-01/820-2553/concepts-4/index.html. [Accessed: 08- Nov- 2021].