Dell EMC SC Series and Red Hat 7.3 High Availability Cluster

Best Practices for a High Availability Red Hat 7.3 Cluster using Dell SC Series Storage with iSCSI Boot from SAN Nodes

Dell Storage Engineering
February 2017

A Dell EMC Best Practices

Revisions

Date	Description
December 2011	Initial release
February 2017	Major Update for RHEL 7.3 and SCOS 7.1

Acknowledgements

Author: Steven Lemons

Table of contents

Executive summary

Red Hat® Enterprise Linux® (RHEL) version 7 is a versatile server operating system that was designed for mission-critical enterprise computing to allow for the deployment of highly available information technology services with exceptional performance and reduced risk. Using the best practices presented in this publication, the RHEL operating system provides an optimized boot from SAN HA Cluster for delivering fault resilient services when coupled with Dell EMC™ SC Series storage.

This best practices paper provides guidelines for configuring RHEL 7.3 cluster nodes to have their OS boot from SC Series storage (boot from SAN) and how to configure a multi-node High Availability (HA) cluster providing a Network File System (NFS) front-end service from back-end shared iSCSI storage using SC Series storage.

1 Red Hat Enterprise Linux High Availability Cluster

The High Availability Add-On for Red Hat Enterprise Linux provides resiliency, scalability and robust availability to critical production services running on RHEL servers. A cluster is a grouping of two of more computers (often referred to as nodes or members) that share the same pool of back-end resources while working together to present a customer-facing and fault redundant service. There are typically four major types of clusters leveraged within the Enterprise datacenter:

1. Storage
2. High Availability
3. Load Balancing
4. High Performance

This paper focuses on the best practices around configuring a storage cluster that provides a consistent file system image across servers in a cluster. This allows multiple servers to simultaneously read and write to a single shared file system from the back-end SAN. By leveraging failover services between nodes within a cluster, the High Availability Add-On of RHEL for Storage simplifies storage administration by eliminating the need for redundant copies of data thus simplifying backup and disaster recovery. The High Availability Add-On of RHEL supports high availability for up to 16 nodes. This paper will focus on the use of four nodes within a single Storage HA cluster. For more information on Red Hat Enterprise Linux High Availability Clusters – go to Red Hat High Availability Add-On Configuration and Management Reference Overview

1.1 RHEL High Availability Add-On Features

The High Availability Add-On is an integrated set of software components that can be deployed in a variety of configurations to suit your needs for performance, high availability, load balancing, scalability, file sharing, and economy.

The High Availability Add-On consists of the following major components:

- Cluster Infrastructure – Functions of nodes within a cluster through the following fundamental tasks: membership management, lock management, configuration file management and fencing.
- Cluster Administration Tools – A suite of scripts to assist with the configuring and management tasks of the High Availability Add-On features.
- High Availability Service Management – When a node within a cluster becomes unstable or reliable, this suite of monitoring scripts provides the failover of services from the failing node to an alternate stable node within the cluster.

1.2 iSCSI

iSCSI technology is a standard technology used for block storage. This technology permits organizations to scale their block storage infrastructure while leveraging existing infrastructure. The open-iSCSI implementation is the only implementation discussed here. For other vendor-provided implementations, refer to the vendor-specific documentation.

iSCSI protocol requires a network port to communicate with the Dell SC Series storage arrays. A dedicated network port or a dedicated VLAN for iSCSI traffic is critical. The type of data determines the network

topology. Data that is sensitive or confidential is treated differently than data that requires immediate, high availability and low latency. These needs drive architecture configurations such as the dedication of ports, VLAN usage, multipathing, and redundancy.

Production application and end-user TCP/IP data ideally uses separate paths for iSCSI traffic. Even better are 10 GB switches dedicated to iSCSI traffic, distinct from 1 GB switches for other server traffic. Examples of constrained architectures might include the use of VLAN-tagged traffic, with iSCSI network traffic tagged differently from general traffic. Whenever possible, use multipath for iSCSI data for redundancy. In the absence of VLAN tagging, different traffic can be routed to different destinations with static routing or at the iSCSI level in the configuration. A complete listing of all Ethernet switches and host CNAs validated with SC Series arrays can be found in the *Dell Storage Compatibility Matrix*.

2 Installing RHEL on Cluster Nodes for iSCSI Boot from SAN

This paper describes creating a highly available active/passive NFS service between four nodes within a configured cluster. During this creation process, a command-line based program called **pcs** is used to configure the Pacemaker cluster and Corosync heartbeat daemon for service monitoring and movement of access/configuration files between the active nodes within the cluster. The use of a virtual IP allows NFS clients to access the highly available NFS service regardless of which node the NFS service is actually running on. The active/passive HA NFS service will run on only one of the four nodes within the cluster at any given time. If at any time the node on which the NFS service is running from becomes unstable or inoperative, the NFS service starts up again on one of the other three standby nodes within the cluster with minimal service interruption.

For the purposes of this paper, the Storage HA cluster created consists of four nodes (alpha, bravo, charlie and delta), each a RHEL 7.3 installation with iSCSI boot from SAN capability from an SC Series array. The SC Series array was also used to provide the shared storage for NFS service presentation.

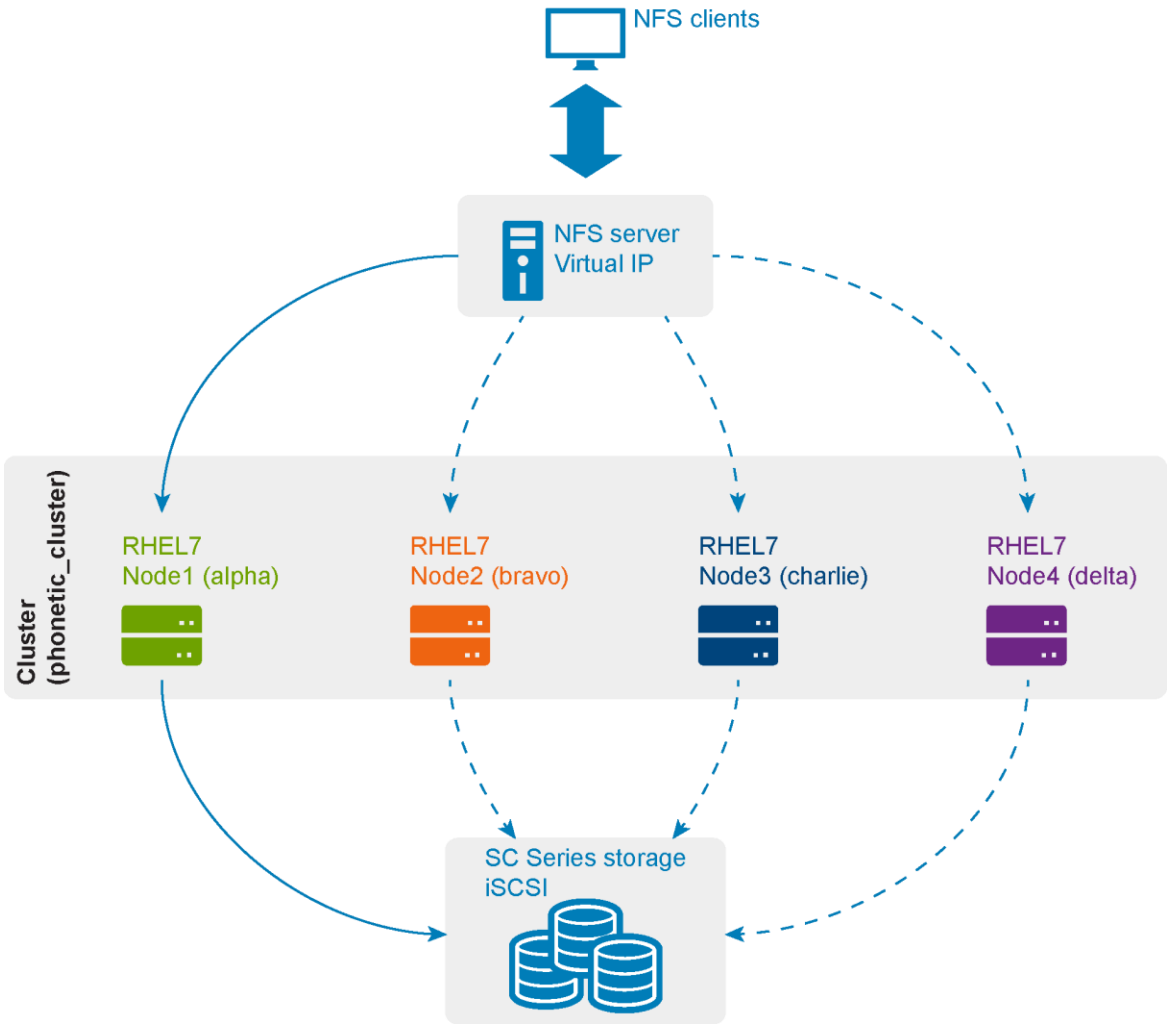

Figure 1 RHEL 7 Highly Available Active/Passive NFS Server

2.1 Create SC Series volumes and server targets

In preparing this environment, the SC Series volumes and server objects must be created prior to each nodes OS installation through the use of the Dell Storage Manager (DSM, reference Figure 2). With this being an iSCSI boot from SAN environment, the actual mapping or defining of each server HBA will be done during the installation process.

Since this is an iSCSI boot from SAN environment, there will be two volumes per server target. The first will be the node_BOOT volume which will hold the contents of /boot. The second will be the node_LVM volume which will hold the contents of /. Since the two are separated, the node_LVM volume can be created as a Logical Volume Manager (LVM) within Linux allowing greater flexibility and growth capability in the future.

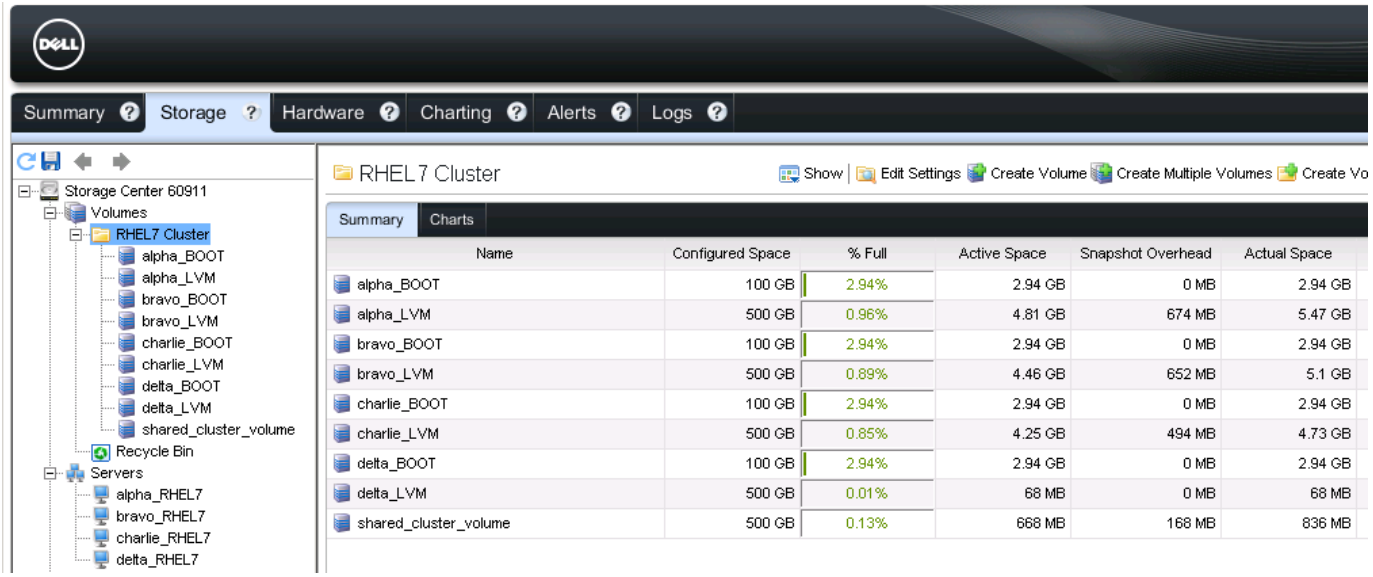

Figure 2 DSM view of SC Series volumes and server objects for each node in cluster

2.2 Configure iSCSI CNA for Boot from SAN

For this environment, each node utilized a dual-port 10G Intel® X520 iSCSI CNA. To benefit from its boot from SAN capabilities, the Intel iSCSI Remote Boot software must be configured. Upon rebooting each node and after POST, enter the Intel iSCSI Remote Boot menu by typing the key combination [Ctrl] + [D] when prompted. These instructions are strictly for the Intel X520 CNA model and may vary with other Intel CNA models.

Figure 3 Intel iSCSI Remote Boot setup

The iSCSI initiator name is inside the iSCSI boot setup and needed for HBA assignment to the respective server node Server object within DSM. This is also where each port of the iSCSI HBA is configured with an IP assignment. The SC Series target IP and name are also defined for each port within this setup menu.

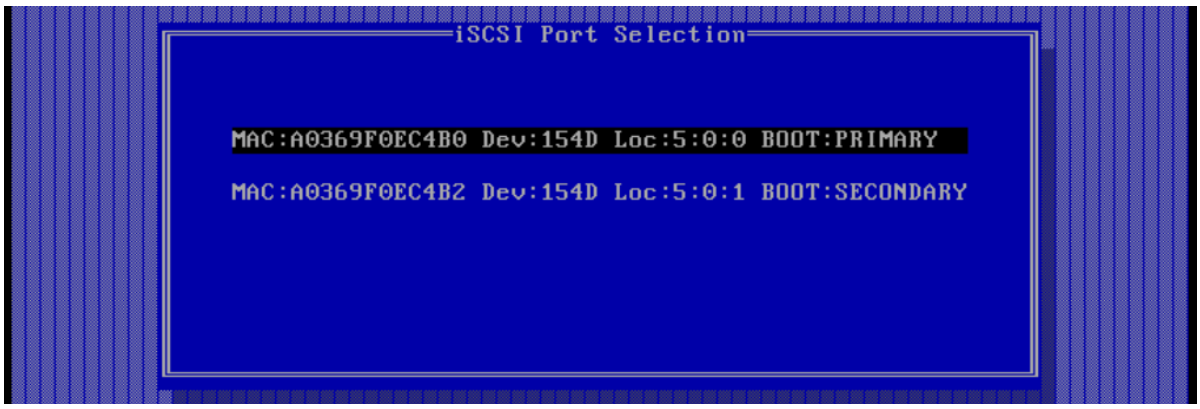

Figure 4 iSCSI Port Selection

Figure 5 iSCSI Port Configuration

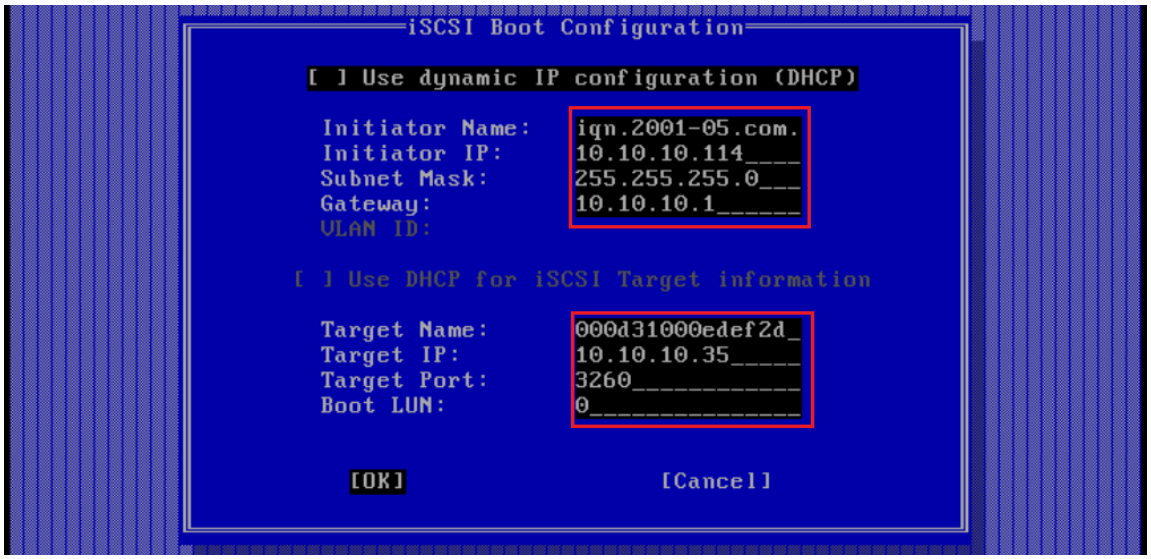

Figure 6 iSCSI Boot Configuration

After selecting the first port (Figure 4), then selecting the iSCSI boot configuration option (Figure 5), the the necessary configuration information is displayed. In Figure 6, the top red box is the defined iSCSI Initiator Name for the first port of the node HBA and is the value used when assigning an HBA to the server object within DSM (Figure 7).

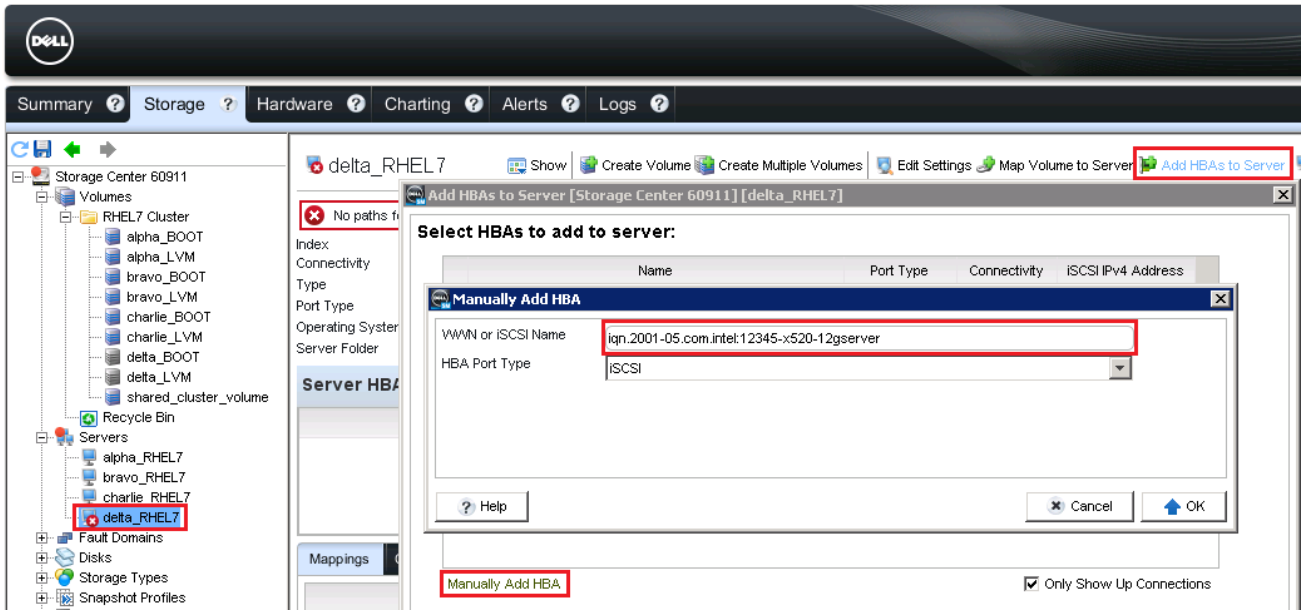

Figure 7 Add HBA to node server object in DSM

Note: The use of the Dell RACADM command can assist in retrieving the iSCSI Initiator Name during this configuration process. More information pertaining to the iDRAC (Integrated Dell Remote Access Controller) RACADM command line can be found here: http://www.dell.com/support/manuals/yu/en/yudhs1/poweredge-r630/iDRAC_RACADM_Pub-v3. For example, to retrieve this information from the first port of the first iSCSI

HBA on this node the following RACADM command was executed from a Windows 2012 Server R2 host residing in the same network segment as the cluster against the OOB (Out Of Band) iDRAC IP address of the alpha node:

```
C:\Windows\system32>racadm -r 192.168.168.218 -i get NIC.IscsiInitiatorParams.5
UserName: root
Password:
[Key=NIC.Slot.3-1-1#IscsiInitiatorParams]
#IscsiInitiatorChapId=
#IscsiInitiatorChapPwd=
IscsiInitiatorGateway=10.10.10.1
IscsiInitiatorIpAddr=10.10.10.114
IscsiInitiatorName=iqn.2001-05.com.intel:12345-x520-12gserver
IscsiInitiatorSubnet=255.255.255.0
```

Once the iSCSI Name has been assigned to the node server object, the node_BOOT volume can be mapped to the server object to provide the Target Name. Enter this target name in the second red box in Figure 6. When mapping this volume to the node server object, since this is a BOOT volume, there are two settings in the **Advanced Options** menu that must be enabled:

1) Map volume using LUN 0 (this is usually reserved for boot volumes)
2) Create maps to down server ports

The **down server ports** option must be enabled since the node has not logged into the SC Series. Without this enabled, the node would not see any volumes mapped and available during the discovery process.

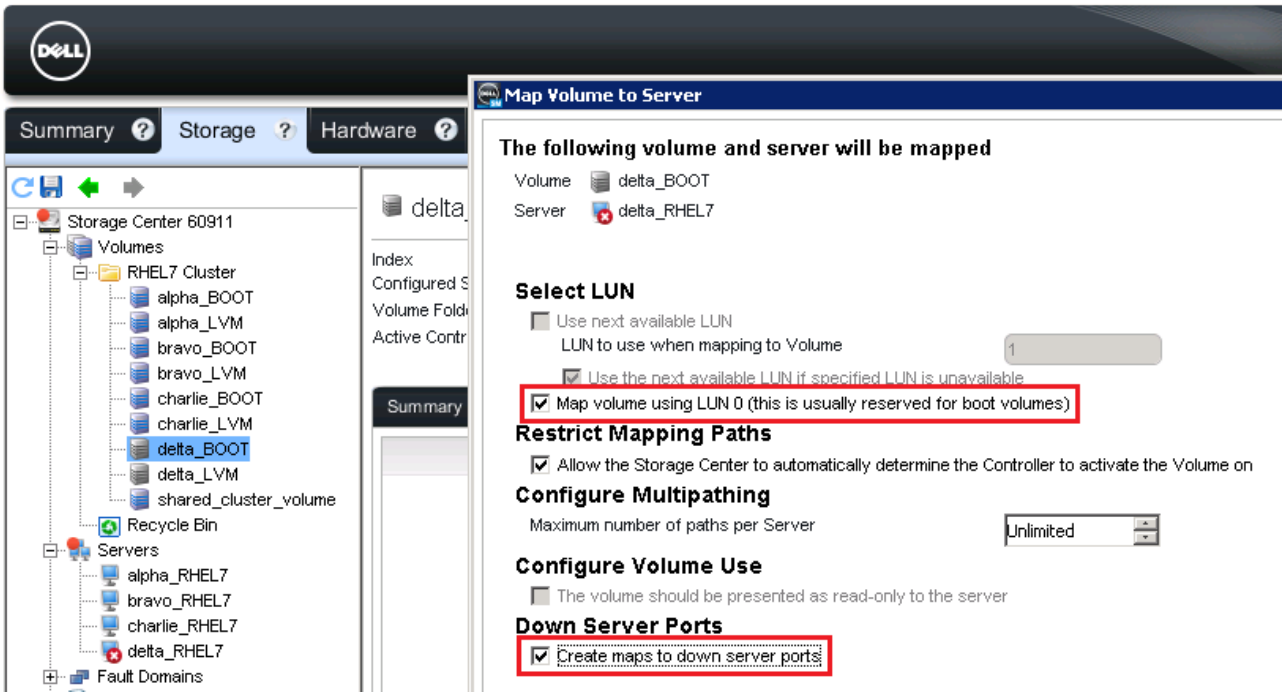

Figure 8 Mapping Volume to Server Object with Advanced Options

Now that the node_BOOT volume has been mapped to the node's Server Object, the iSCSI target IP address and iSCSI Name need to be configured through the iSCSI boot configuration menu. This information is obtained by reviewing the Virtual Port information for each Fault Domain configured on the SC Series array through DSM. Reference Section 2.2.1 for information regarding SC Series host connectivity and IP assignments within Fault Domains.

In the following example, configuring the first port of the iSCSI HBA on this node to the target IP of the first Fault Domain within the SC Series array allows us to match up the following information for configuration within the iSCSI Boot Configuration screen. For this first port, the target IP used is 10.10.10.35 so select the fault domain within DSM that matches this information. Once the match is found, record the first Virtual Ports name listed within the Fault Domains informational window. This value, combined with the iSCSI Name prefix produces the value needed for configuration. For example, from the below screenshots, the desired iSCSI Target Name to use for iSCSI Target IP of 10.10.10.35 is: iqn.2002-03.com.compellent:5000D31000EDEF2F.

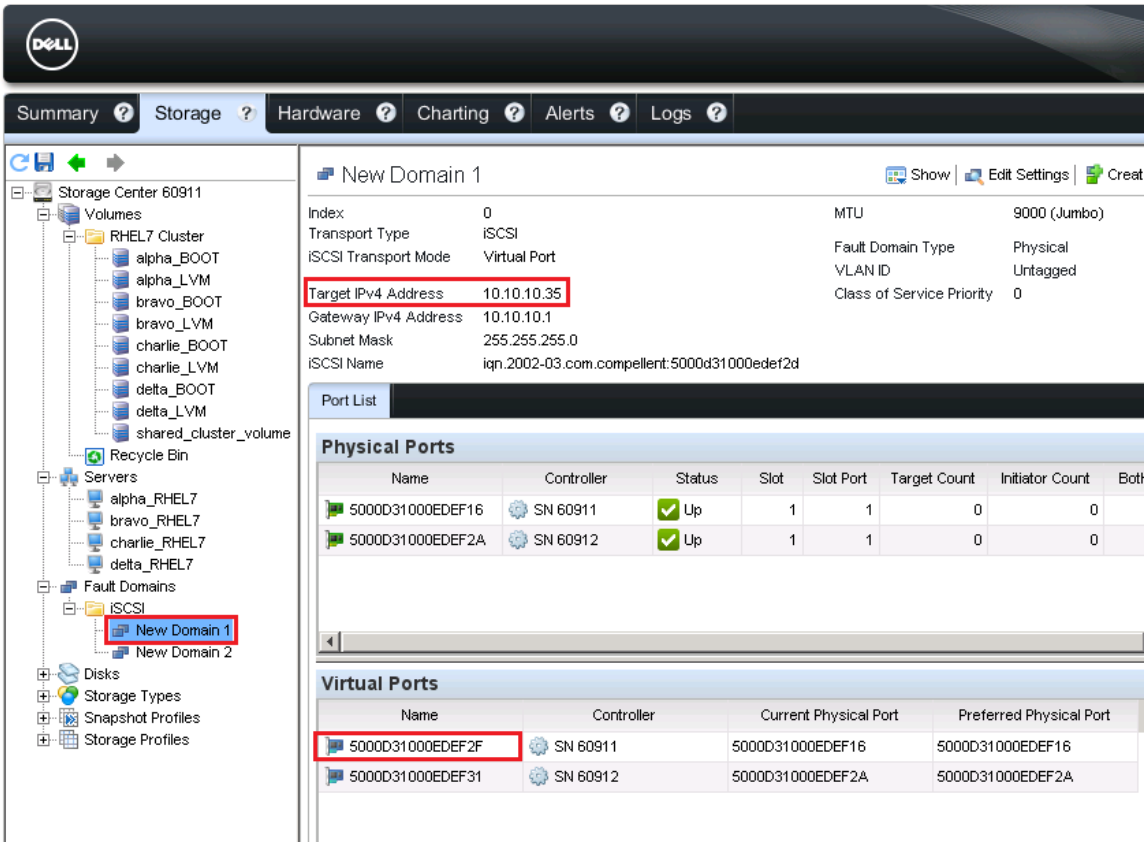

Figure 9 SC Series Fault Domain Information

Once the target iSCSI IP and iSCSI Name have been configured within the iSCSI boot configuration, save these changes and repeat these steps for the secondary port of the iSCSI HBA interface.

Note: During the configuration of the secondary iSCSI HBA interface within the iSCSI boot configuration screen, ensure information is obtained from the alternate (or secondary) Fault Domain as each iSCSI interface will have a unique value for Target iSCSI IP and Target iSCSI Name.

After both iSCSI HBA ports have been configured within the iSCSI boot configuration menu, exiting the system will allow discovery of the node_BOOT volume mapped to the server node object. If no LUNs are reported back during discovery process, investigate the configuration information entered during setup. The below screenshot reflects both ports of the iSCSI CNA from the alpha node being able to see the SC Series target IP and discovering the 100GB node_BOOT volume mapped therein.

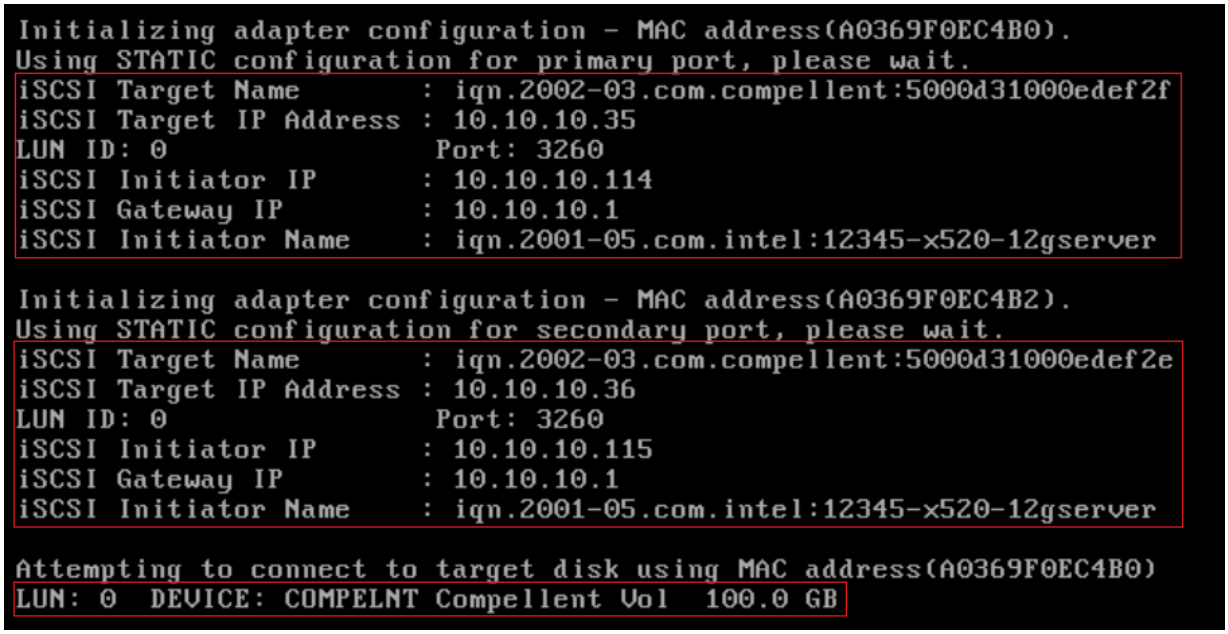

Figure 10 iSCSI HBA boot volume discovery

Now that the iSCSI boot configuration is complete, the final step before booting the RHEL 7.3 installation media is to map the node_LVM volume to the same server object. This allows adding disks to the node, visible by the installation media thus preparing a true boot from SAN RHEL 7.3 node. The screenshot below reflects the Map Volume to Server process from within DSM (Figure 11).

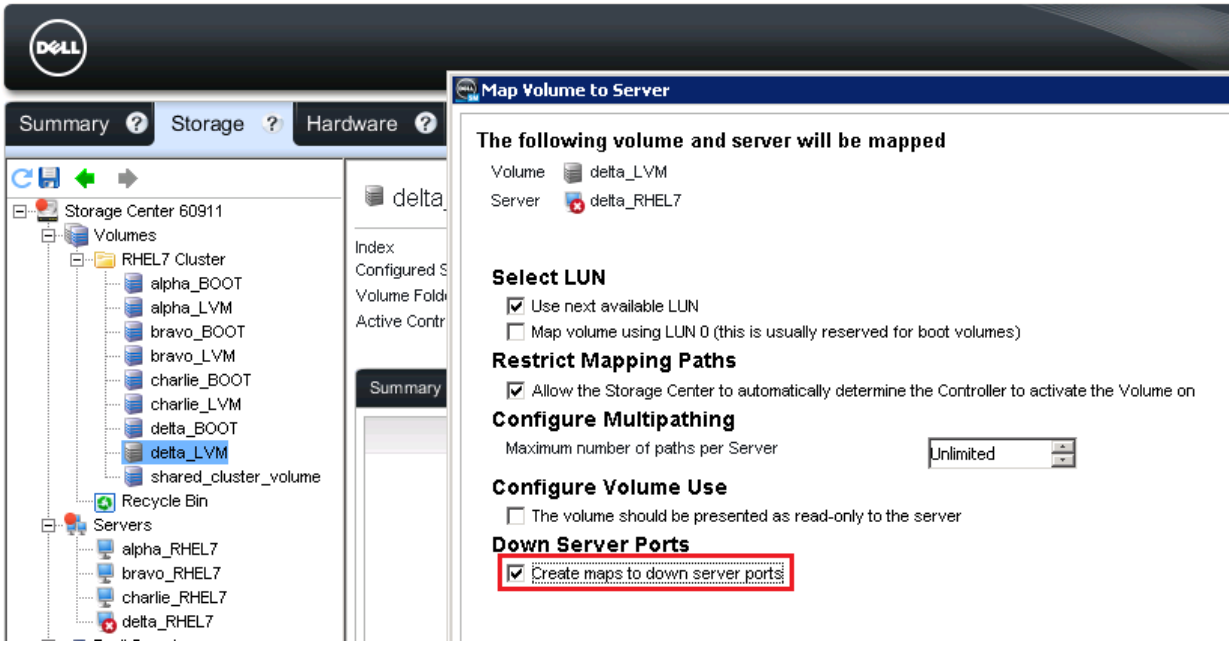

Figure 11 Map node_LVM volume to same node's server object within DSM

Note: During the map volume to server process enable **Create maps to down server ports** because the iSCSI HBA interfaces have not logged into the SC Series array. Without this option enabled, the installation media will not discover any available volumes for disk creation.

2.2.1 SC Series host physical connectivity and IP assignment

Depending on the OS-specific implementation, different methods are used to connect the arrays and assign IP addresses. Since SC Series fault domains are connected by an ISL and are in a single IP subnet, it is important to ensure that iSCSI sessions are properly established within their fault domains. Host ports connected to Fault Domain 1 should connect to switch fabric and storage ports on Fault Domain 1 physically. The same rule applies for Fault Domain 2. This step is important because, with a single subnet, it is possible for the hosts to access SC Series storage ports on both fault domains. The correct connectivity minimizes ISL traffic and ensures that at least some iSCSI sessions will persist in the event of a component failure.

For more information regarding Dell SC proper setup, cabling, zoning and fault domain creation, please see *Storage Center Connectivity Guide*, available on the Dell SC Series Customer Portal (login required).

Figure 12 depicts proper connection from each host port to the SC Series storage ports within the same fault domain without traversing the switch interconnection.

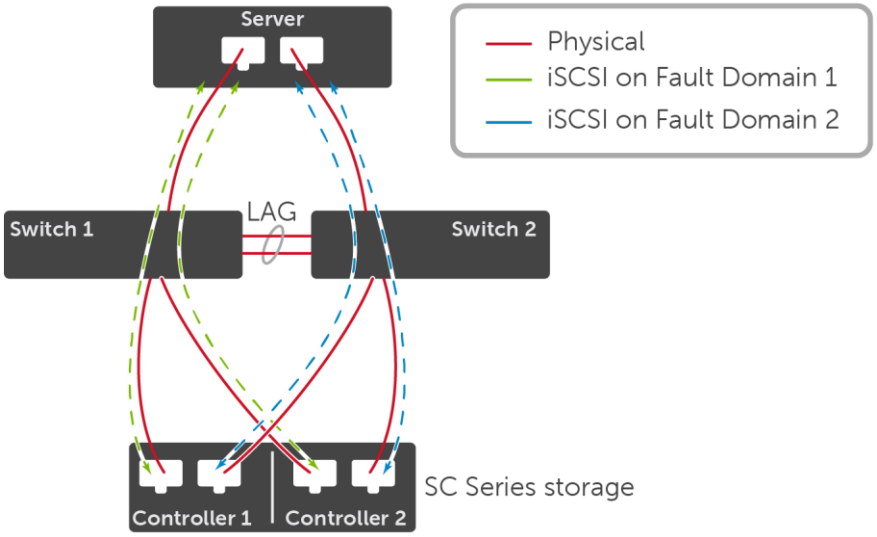

Figure 12 Connecting the host to SC Series ports

Note: With the approach discussed in this paper, misconfiguration of the SC Series connectivity (for example, host ports not connected to the correct fault domain) can lead to loss of volume access in the event of a switch failure.

2.3 Configure networking

Once the installation media has booted, the first configuration step is to configure the 10GB NICs with IPs within the SAN network where the SC Series array resides. At the INSTALLATION SUMMARY screen, click **NETWORK & HOSTNAME** to enter networking configuration.

For more information on how to install RHEL 7, please see the Red Hat Enterprise Linux 7 Installation Guide

Figure 13 Configure network and hostname during RHEL 7 installation

For the purposes of this paper, three NICs per node were configured: 1 data (em3) & 2 10GB iSCSI (p1p1 & p1p2). For each NIC that shows a connection, select it and then click **configure** in the bottom right-hand corner. Once configured, click the toggle button in the upper right-hand corner (from **off** to **on**) to enable the NIC interface.

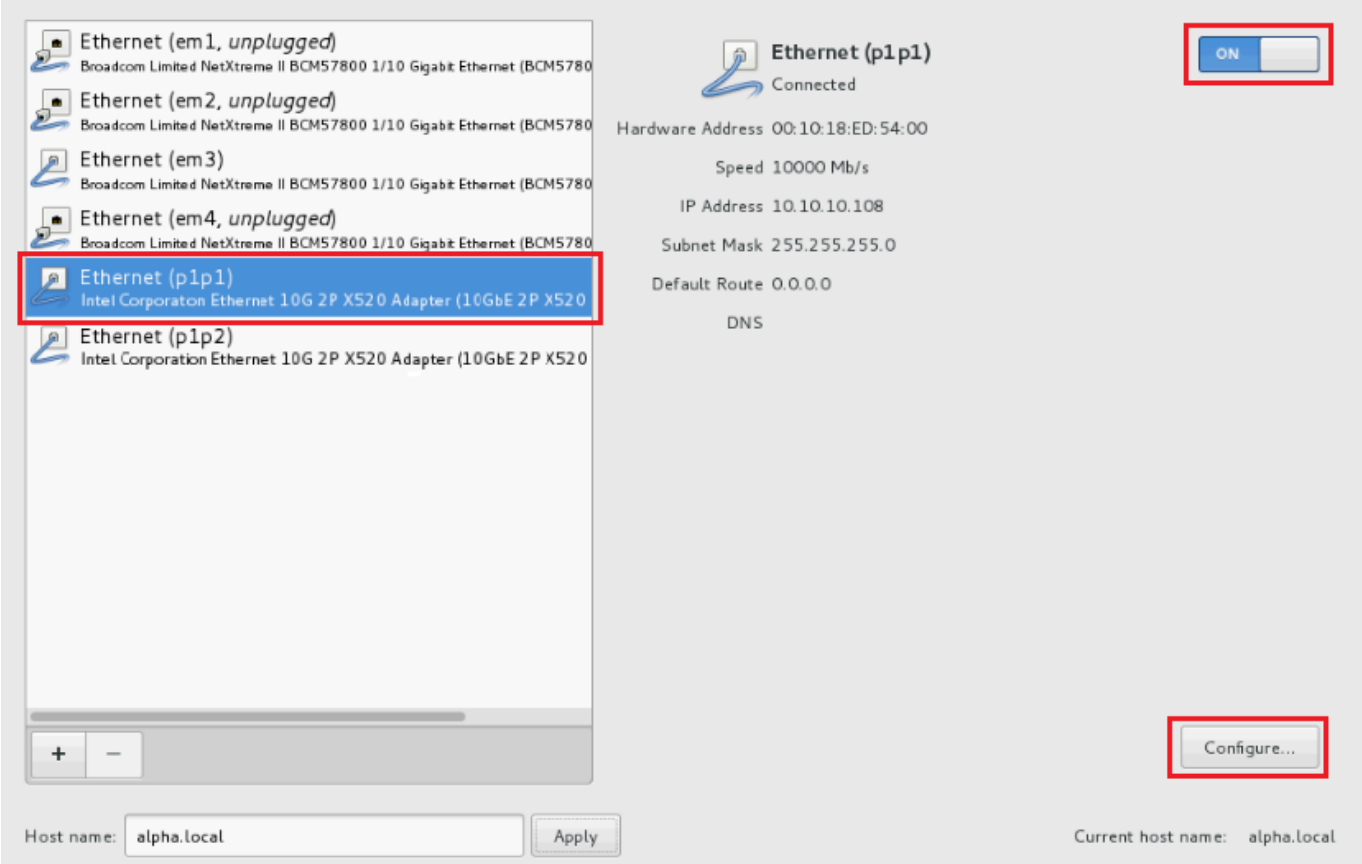

Figure 14 Network & Hostname configuration

For the purposes of this paper, the NICs configured and enabled are: p1p2, p1p1 & em3.

Figure 15 Configured Network Interfaces

2.4 Add iSCSI targets for multipath volumes as installation destination

Once the NIC ports are configured and enabled, adding the iSCSI targets for creation of a multipath enabled disks is the next step. From within the INSTALLATION SUMMARY menu, click on the “INSTALLATION DESTINATION” option.

Figure 16 Installation Destination

From the Installation Destination screen, uncheck any local standard disks listed as none will be used for these boot from SAN nodes. Once all local standard disks and unchecked, next is to select “Add a disk” under the Specialized & Network Disks section.

Figure 17 Add a disk

This next process is to provide the same SC Series iSCSI Target IP information that was used from Section 2.2. After selecting “Add a disk” and then “Add iSCSI Target” from the next provided menu options, enter the first iSCSI storage target IP address and select “Start Discovery”. This should produce two nodes from the same storage target IP address. Ensure to check both nodes and then select “Log In”. Repeat this process for the secondary iSCSI storage target IP address until there are two multipath volumes displayed within the Installation Destination screen (Figure 20). The volumes displayed are the “node_BOOT” and “node_LVM” volumes that were previously mapped to the node’s server object, in Figure 8 & Figure 11. The below screenshots outline this process.

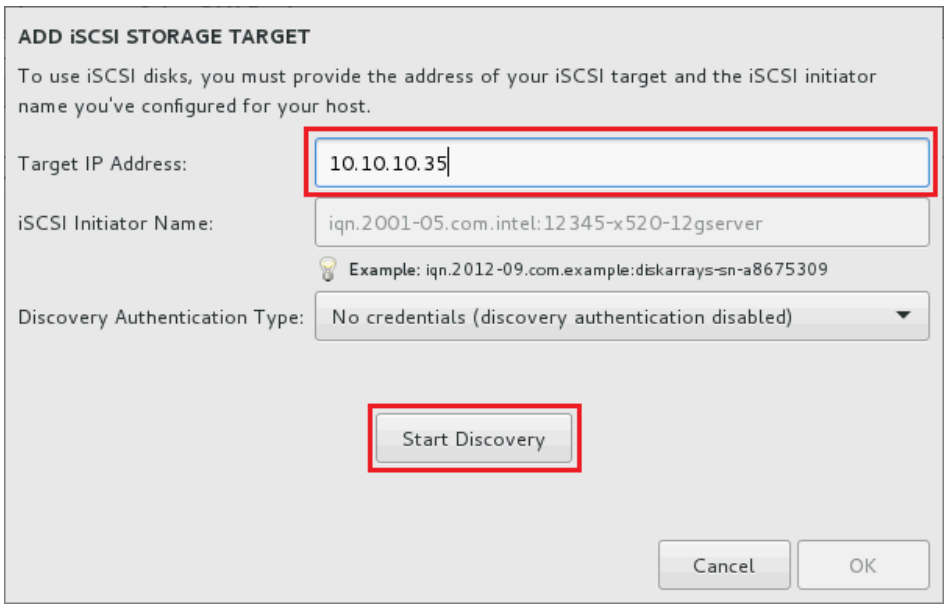

Figure 18 Add first iSCSI Target IP Address

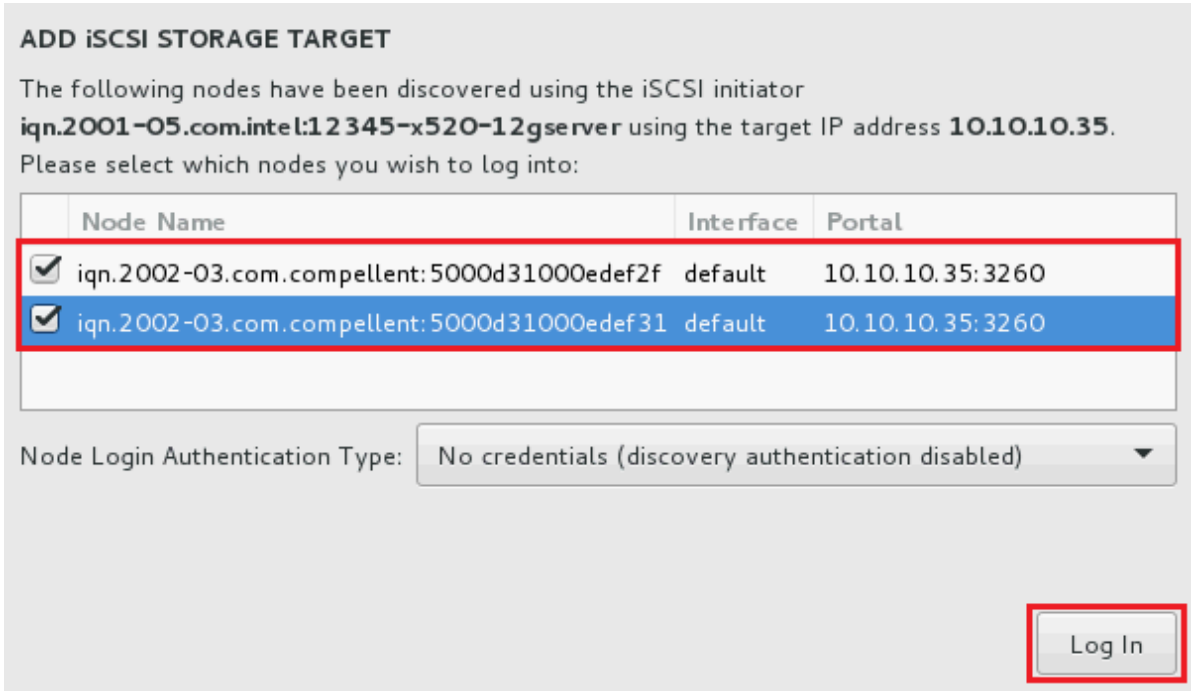

Figure 19 iSCSI Node Selection

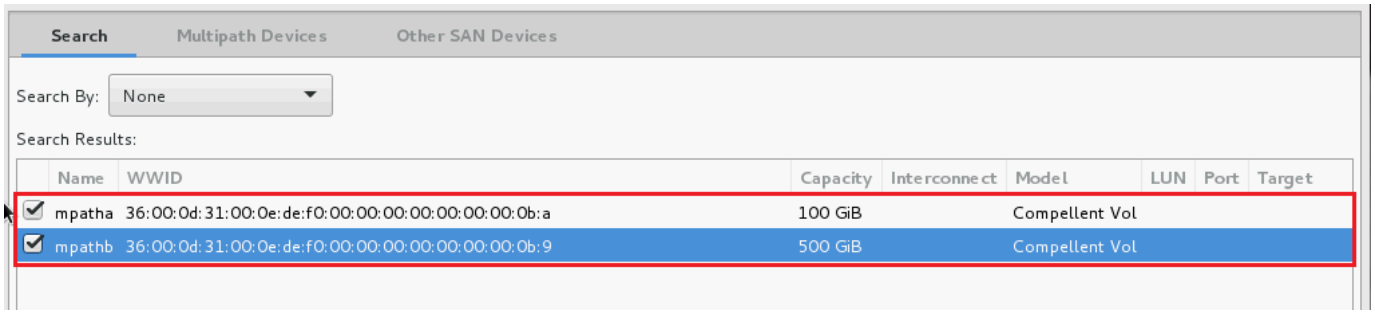

Figure 20 Multipath volume selection

Once the multipath volumes are selected, the final process before installation of the OS begins is to confirm the disk layout for installation and allow the installation media to automatically configure partitioning on the discovered volumes. This is reflected in the below screenshot by **deselecting any local standard disks** and selecting the two multipath volumes presented.

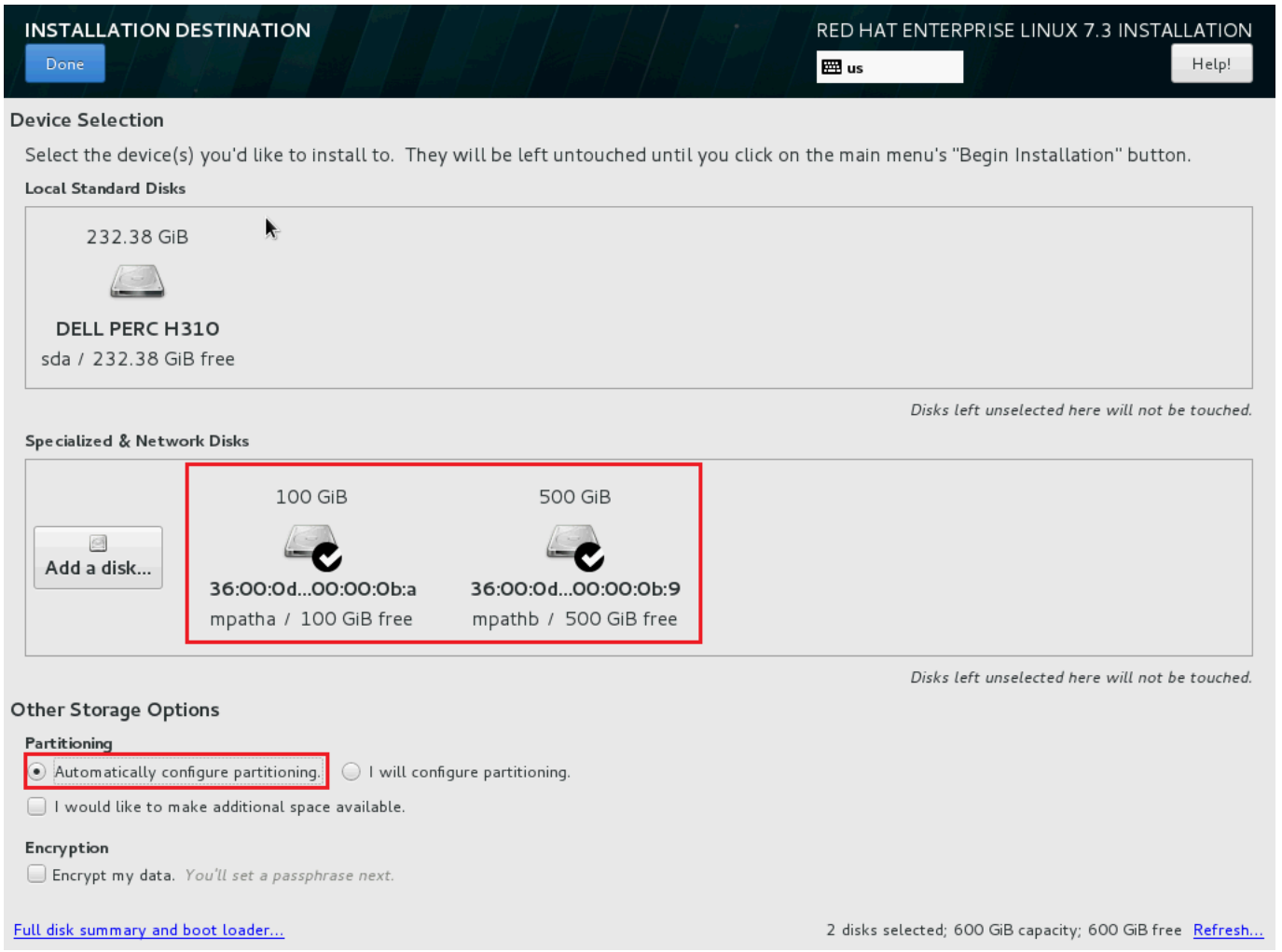

Figure 21 Installation destination confirmation

2.5 Installing RHEL 7

Now that networking and installation destination have been configured, the OS installation process can begin. Select “Begin Installation” for the OS to begin installation.

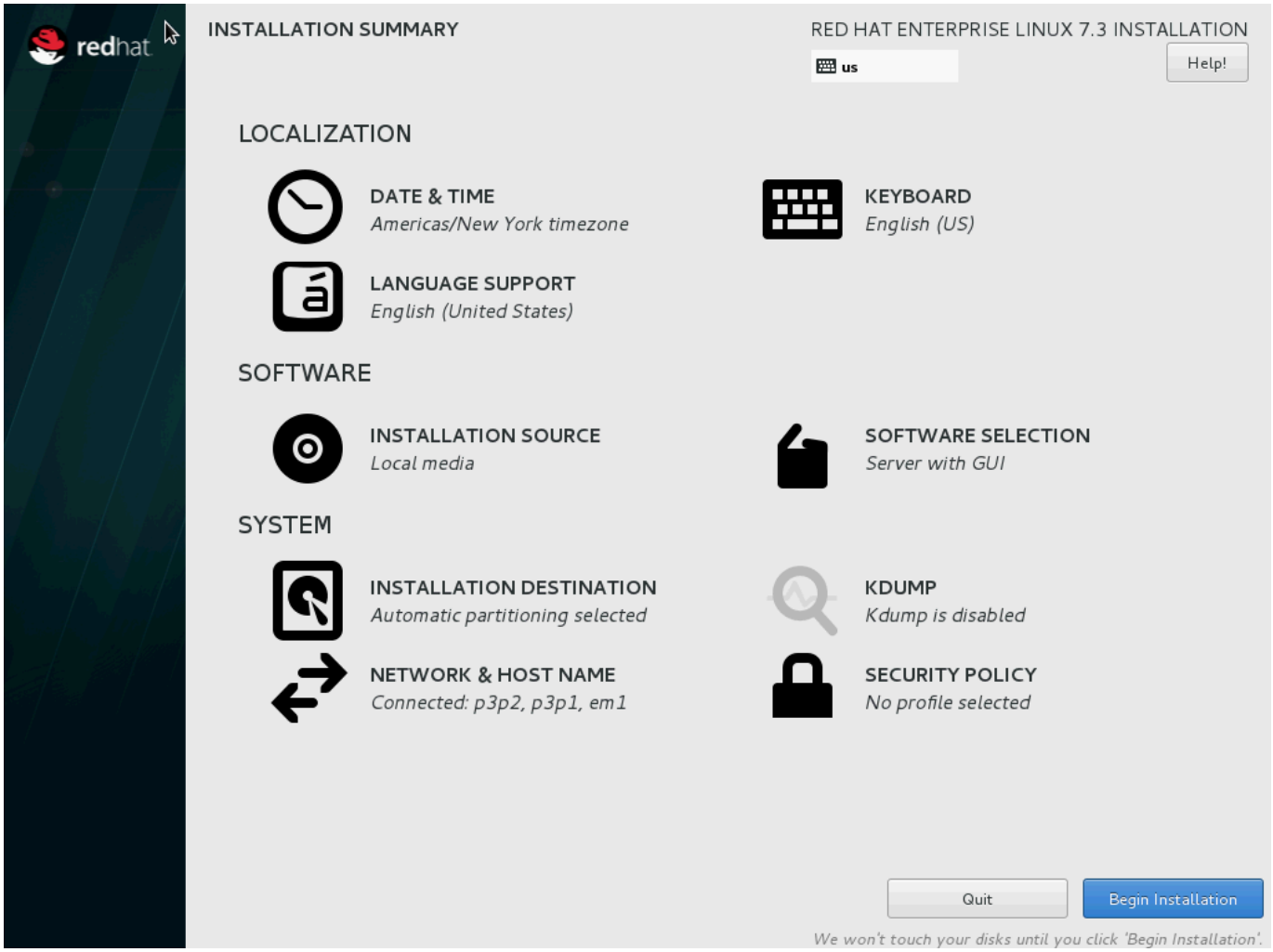

Figure 22 Begin Installation

Repeat this installation process for each host which will be configured as a node with in the cluster. For the purposes of this paper, the installation process was only outlined for node alpha but repeated for host nodes bravo, charlie and delta.

3 Creating a RHEL HA Cluster with Pacemaker

Now that all nodes within the cluster are running RHEL 7.3 with boot from SAN functionality, the next step is to configure each node to subscribe to the High Availability Add-On channel which introduces an integrated suite of software components that can be deployed in a wide variety of configurations to suit your specific environments needs. Some of these needs may be around performance, high availability, scalability, file share, and load balancing.

The High Availability Add-On is made up of the following major components:

- Cluster Infrastructure – Functions of nodes within a cluster through the following fundamental tasks: membership management, lock management, configuration file management and fencing.
- Cluster Administration Tools – A suite of scripts to assist with the configuring and management tasks of the High Availability Add-On features.
- High Availability Service Management – When a node within a cluster becomes unstable or reliable, this suite of monitoring scripts provides the failover of services from the failing node to an alternate stable node within the cluster.

3.1 Subscribe to RHEL High Availability channel

Subscribing each node to the RHEL High Availability Add-On channel provides the basic functions from a group of computers (often referred to as nodes or members) to work together as a cluster through cluster infrastructure resources.

The following subscription action must be completed on each node within the cluster.

```
[root@alpha ~]# rhn-channel –a –c rhel-x86_64-server-ha-7
```

3.2 Host names and definitions

During the cluster configuration process, each node will communicate with the other nodes through DNS resolution or hostname values. For this paper, each node's /etc/hosts file was edited to ensure hostname resolution of all nodes rather than enterprise level active directory or DNS resolution for each node. In a production level environment, DNS is the preferred method of host name resolution and is a fundamental piece for the stable operation of the cluster.

The following node definition matrix was added to each nodes /etc/hosts file:

```
192.168.168.213      alpha alpha.local
192.168.168.215      bravo bravo.local
192.168.168.217      charlie charlie.local
192.168.168.219      delta delta.local
```

3.3 Install cluster software

Once each node has been subscribed to the High Availability Add-On channel, the following cluster software packages must be installed: **pcs** & **pacemaker**

```
[root@alpha ~]# yum install pcs pacemaker –y
```

Note: Installing and configuring the cluster infrastructure package “fence-agents-all” is recommended when utilizing RHEL HA Clustering technology in a production environment. For the purposes of this paper, configuring and enabling fence-agents will not be discussed. More information concerning RHEL HA Clustering and fence-agents can be found at: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/High_Availability_Add-On_Overview/s1-fencing-HAAO.html

3.4 Configure nodes for HA Cluster

In order to use the **pcs** software package to configure the cluster and communicate between the nodes, a password must be set on each node for the use ID account of **hacluster**. This **hacluster** account is the **pcs** administration account and will be used for all cluster administration tasks. It is highly recommended that the password for the **hacluster** account be the same on each node.

This configuration action should be performed on each node within the cluster. The following example is executed from the alpha node and then repeated for the bravo, charlie and delta nodes to ensure each nodes local /etc/passwd database is updated with the correct cluster administration login credentials.

```
[root@alpha ~]# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
Passwd: all authentication tokens updated successfully.
```

On each node, before the cluster can be configured, the **pcsd** service must be started and enabled to boot on startup. This **pcsd** service works with the **pcs** command to manage cluster configuration across all nodes within the cluster.

This configuration action should be performed on each node within the cluster. The following example is executed from the alpha node and then repeated for the bravo, charlie and delta nodes to ensure each nodes service manager is configured to enable and start the **pcsd** service.

```
[root@alpha ~]# systemctl start pcsd.service
[root@alpha ~]# systemctl enable pcsd.service
```

From the alpha node (the first in the cluster), authenticate the **pcs** user **hacluster** for each node in the cluster. This action is only needed on a single node (alpha), not every node within the cluster.

```
[root@alpha ~]# pcs cluster auth alpha bravo charlie delta
Uername: hacluster
Password:
alpha: Authorized
bravo: Authorized
charlie: Authorized
delta: Authorized
```

3.5 Create HA Cluster

Now that all nodes with in the cluster have the necessary cluster infrastructure services running and are configured with the **pcs** administration account, the next step is to create the HA cluster which is done only from a single node (alpha).

```
[root@alpha ~]# pcs cluster setup --start --name phonetic_cluster alpha bravo
charlie delta
Destroying cluster on nodes: alpha, bravo, charlie, delta...
alpha: Stopping Cluster (pacemaker)...
bravo: Stopping Cluster (pacemaker)...
charlie: Stopping Cluster (pacemaker)...
delta: Stopping Cluster (pacemaker)...
alpha: Successfully destroyed cluster
bravo: Successfully destroyed cluster
charlie: Successfully destroyed cluster
delta: Successfully destroyed cluster

Sending cluster config files to the nodes...
alpha: Succeeded
bravo: Succeeded
charlie: Succeeded
delta: Succeeded

Starting cluster on nodes: alpha, bravo, charlie, delta...
alpha: Starting Cluster...
bravo: Starting Cluster...
charlie: Starting Cluster...
delta: Starting Cluster...

Synchronized pcsd certificates on nodes alpha, bravo, charlie, delta...
alpha: Success
bravo: Success
charlie: Success
delta: Success

Restarting pcsd on the nodes in order to reload the certificates...
alpha: Success
bravo: Success
charlie: Success
delta: Success
```

Now enable the cluster infrastructure services to run on each node within the cluster whenever the node is booted or rebooted. Again, this command is only run from a single node (alpha), not all nodes within the cluster.

```
[root@alpha ~]# pcs cluster enable --all
alpha: Cluster Enabled
```

```
bravo: Cluster Enabled
charlie: Cluster Enabled
delta: Cluster Enabled
```

With all nodes within the cluster now configured and enabled, checking the status of the cluster can be done through the **pcs cluster status** or **pcs status** commands. The following two outputs of these commands display the clusters status.

```
[root@alpha ~]# pcs cluster status
Cluster Status:
 Stack: corosync
 Current DC: delta (version 1.1.15-11.el7_3.2-e174ec8) - partition with quorum
 Last updated: Thu Jan  5 12:45:06 2017         Last change: Thu Jan  5 12:37:20
2017 by hacluster via crmd on delta
 4 nodes and 0 resources configured

PCSD Status:
  charlie: Online
  alpha: Online
  bravo: Online
  delta: Online

[root@alpha ~]# pcs status
Cluster name: phonetic_cluster
WARNING: no stonith devices and stonith-enabled is not false
Stack: corosync
Current DC: delta (version 1.1.15-11.el7_3.2-e174ec8) - partition with quorum
Last updated: Thu Jan  5 12:45:10 2017          Last change: Thu Jan  5 12:37:20
2017 by hacluster via crmd on delta

4 nodes and 0 resources configured

Online: [ alpha bravo charlie delta ]

No resources


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

4 Creating an Active/Passive NFS Server in HA Cluster

With all of the nodes within the cluster now configured and enabled, creating a highly available active/passive NFS server between all four nodes is the focus of this chapter. During this creation process, **pcs** is used to configure the Pacemaker cluster resources for monitoring and movement of access/configuration between the active nodes within the cluster. The NFS server will run on one of the four nodes within the cluster. If at any time the node which NFS resides upon becomes unstable or inoperative, the NFS server starts up again on one of the other three standby nodes within the cluster with minimal service interruption.

4.1 Map single shared volume to all nodes

The first step in creating this highly available active/passive NFS server is to create a shared storage Volume on the SC Series array accessible by each node with in the cluster. This will require mapping the same SC Series volume to each node server object within DSM. After mapping the shared volume to the alpha node, each subsequent mapping request will present a warning message. It is safe to acknowledge this warning message since the data I/O to this single volume from the multiple nodes will be controlled through the high availability cluster infrastructure software. The below screenshot shows this warning message and ultimately the final result of the single volume mapped to all nodes within the cluster.

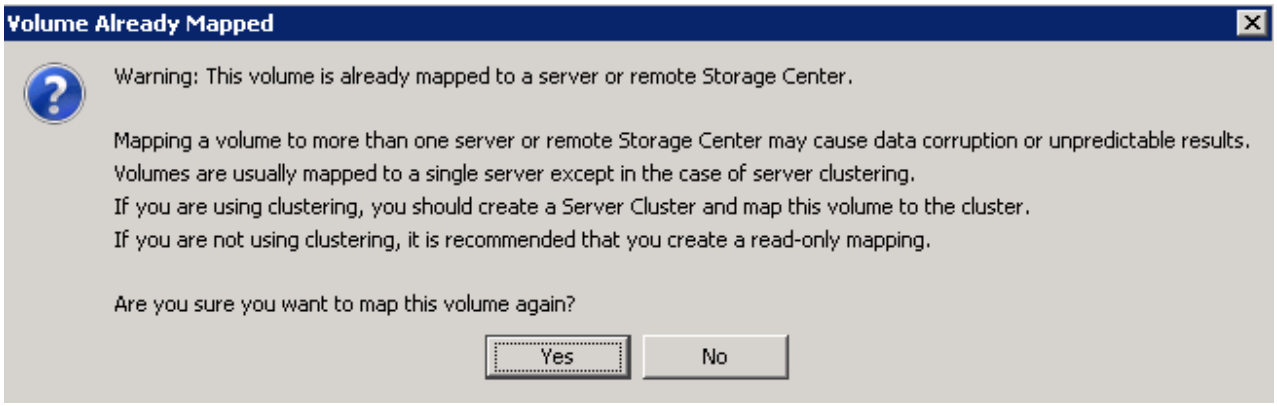

Figure 23 Single volume to multiple server object warning in DSM

For more information regarding Dell SC proper setup, cabling, zoning and fault domain creation, please see *Storage Center Connectivity Guide*, available on the Dell SC Series Customer Portal (login required).

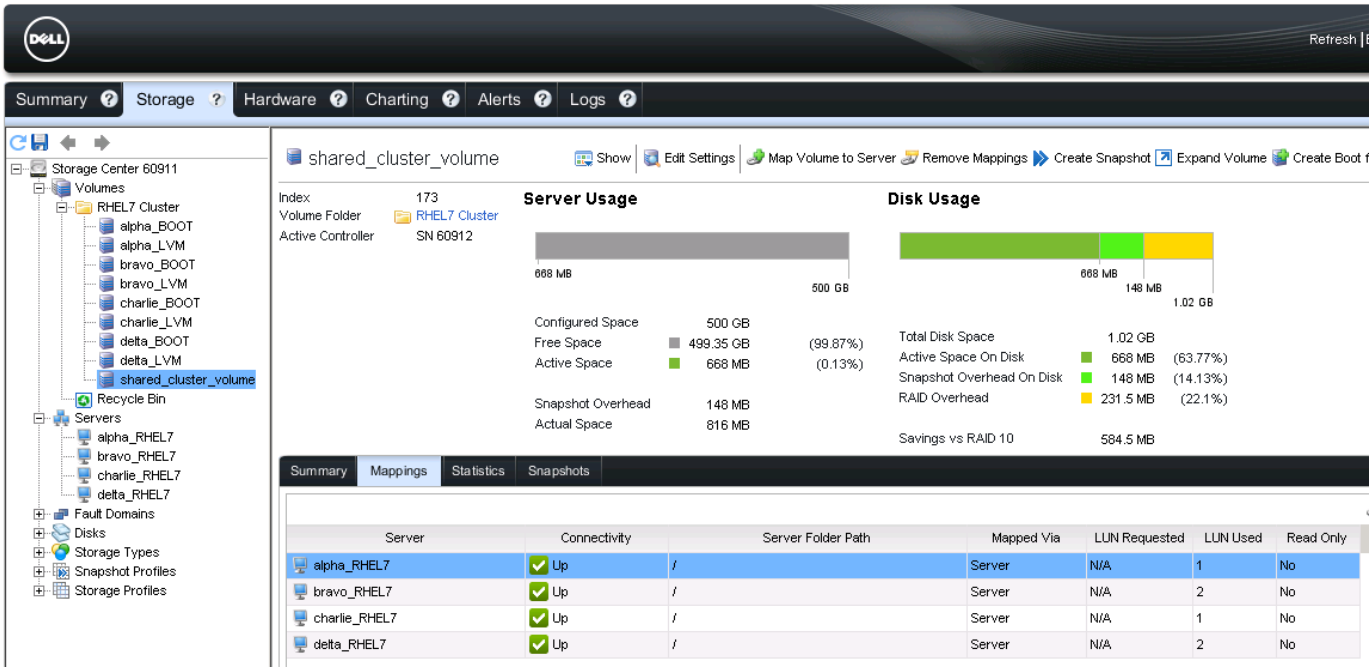

Figure 24 Single volume mapped to all four nodes within the cluster

Note: While there is an option within SCOS 7 & DSM to create a Server Cluster with multiple server objects contained therein so mapping a single volume to multiple servers is easier and without warning, this paper does not utilize that functionality due to each node being configured for iSCSI boot from SAN with only a single iSCSI CNA. If an additional iSCSI CNA were installed in each node, this additional functionality could be used.

Once the single shared volume has been mapped to all nodes within the cluster, each node will need to rescan the SC Series array for the new discovered target and then logout/login to the SC Series array for the new volume to become assessable. This is done with this **iscsiadm** utility, as displayed below. Again, this should be executed on each node within the cluster.

```
[root@alpha ~]# iscsiadm -m discoverydb -p 10.10.10.35 -t st -o new -D
[root@alpha ~]# iscsiadm -m discoverydb -p 10.10.10.36 -t st -o new -D
[root@alpha ~]# iscsiadm -m node -u
[root@alpha ~]# iscsiadm -m node -l
```

4.2 Create LVM Volume with an ext4 file system on shared volume

Now that each node can access the same single shared volume from the SC Series array, the next step is to create an LVM logical volume and then create an ext4 file system on that mapped volume. The below command sequence walks through this configuration process. In this example, the multipath object of mpathb will be used to store the LVM physical volume and will be used to create the LVM logical volume for use within the NFS server.

These configuration actions should only be performed on a single node and in this example, will be performed from the alpha node.

1. Create the LVM physical volume on the multipath /dev/mapper/mpathb device:

```
[root@alpha ~]# pvcreate /dev/mapper/mpathb
  Physical volume "/dev/mapper/mpathb" successfully created.
```

2. Create the phonetic_vg volume group that consists of the multipath volume /dev/mapper/mpathb:

```
[root@alpha ~]# vgcreate phonetic_vg /dev/mapper/mpathb
  Volume group "phonetic_vg" successfully created
```

3. Create the phonetic_lv logical volume using the phonetic_vg volume group:

```
[root@alpha ~]# lvcreate -L450G -n phonetic_lv phonetic_vg
  Logical volume "phonetic_lv" created.
```

4. Display the currently configured logical volumes on the system with lvs:

```
[root@alpha ~]# lvs
  LV          VG          Attr       LSize   Pool Origin Data%  Meta%  Move
Log Cpy%Sync Convert
  phonetic_lv phonetic_vg -wi-a----- 450.00g
  root        rhel_alpha  -wi-ao---- 372.53g
  swap        rhel_alpha  -wi-ao----  46.57g
```

5. On the logical volume phonetic_lv create an ext4 file system:

```
[root@alpha ~]# mkfs.ext4 /dev/phonetic_vg/phonetic_lv
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=512 blocks, Stripe width=512 blocks
29491200 inodes, 117964800 blocks
5898240 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2267021312
3600 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
      4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
      102400000
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

4.3 Configure NFS share

Configuring the NFS share for NFS service failover is the next step. The below command sequence walks through this configuration process. In this example, the NFS share configuration process will use the previously created logical group and local volume on a single node.

These configuration actions should only be performed on a single node and in this example, will be performed from the alpha node.

1. Create the /cluster_share directory:

```
[root@alpha ~]# mkdir /cluster_share
```

2. Mount the ext4 file system created on /dev/phonetic_vg/phonetic_lv on the /cluster_share directory:

```
[root@alpha ~]# mount /dev/phonetic_vg/phonetic_lv /cluster_share
```

3. Create an exports directory tree within the /cluster_share directory:

```
[root@alpha ~]# mkdir –p /cluster_share/exports
[root@alpha ~]# mkdir –p /cluster_share/exports/export1
[root@alpha ~]# mkdir –p /cluster_share/exports/export2
```

4. Create tests files within the exports directory tree for NFS clients to have access to. For this example, two files were created: testfile1 and testfile2:

```
[root@alpha ~]# touch /cluster_share/exports/export1/testfile1
[root@alpha ~]# touch /cluster_share/exports/export2/testfile2
```

5. With necessary exports directory tree and test files created, unmount the logical volume share for further configuration actions:

```
[root@alpha ~]# umount /dev/phonetic_vg/phonetic_lv
[root@alpha ~]# vgchange -an phonetic_vg
```

4.4 Configuring shared volume for exclusive cluster activation

With the LVM volume group now configured for NFS export, it must now be configured to ensure that only the cluster is capable of enabling and interacting with the volume group. This will ensure that the volume group is not activated outside of the cluster on node startup. If the volume group is allowed to be activated outside of the cluster group, the risk of data corruption to the volume group metadata is introduced.

The following configuration actions will define the volume_list variable within the /etc/lvm/lvm.conf configuration file which defines which volume groups are automatically allowed to activate on the local node

outside of the clusters infrastructure management control. Any volume group related to the node's local root and home directories, including /boot, should be defined with in this volume_list variable array. Any volume group that is part of the highly available active/passive NFS server must be excluded from the volume_list variable array entry within the /etc/lvm/lvm.conf configuration file.

These configuration actions should be performed on each node within the cluster. The following examples are executed from the alpha node and then repeated for the bravo, charlie and delta nodes.

1. Configure locking_type is set to 1 and that use_lvmetad is set to 0 in the /etc/lmv/lvm.conf configuration file:

```
[root@alpha ~]# lvmconf --enable-halvm --services –startstopservices
```

2. Using the vgs command, display the currently configured volume groups on the node:

```
[root@alpha ~]# vgs --noheadings -o vg_name
  phonetic_vg
  rhel_alpha
```

3. Any volume group listed in the previous command other than phonetic_vg must be added to the volume_list variable array within the /etc/lvm/lvm.conf configuration file. For example, for the alpha node the following will be added to the /etc/lvm/lvm.conf configuration file:

```
volume_list = [ "rhel_alpha" ]
```

4. The initramfs boot image must be rebuilt to guarantee the boot image will not try and activate the logical volume controlled by the cluster infrastructure scripts. The following command updates the initramfs device and may take upwards of two minutes to complete:

```
[root@alpha ~]# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

5. Once the initramfs boot image has been rebuilt, reboot the node:

```
[root@alpha ~]# reboot
```

6. Once the node has rebooted, the cluster service can be checked to see if cluster services are up and running once more. This action is completed with the pcs cluster status command. If any error messages are received, wait until all nodes in the cluster are rebooted and then perform the following command from a single node:

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

4.5 Configure cluster resources

With the NFS exports and shared volume for exclusive cluster activation configured, to enable the NFS server to become a highly available service between the nodes in the cluster, cluster resources must be configured. The cluster resources created in the below examples will start in the order that their added to the resource group nfsgroup and will stop in the reverse order that they were added to the resource group nfsgroup. This group, nfsgroup, is created to ensure all resources are run on the same node.

Note: For production environments using high availability cluster add-on technology, the use of fencing agents by defining a Fencing Configuration with STONITH (Shoot The Other Node In The Head) is recommended. However, for the purposes of this paper, STONITH has been disabled since fencing is not in use. If STONITH is enabled yet no fencing agents defined then any created resources will not start within the cluster.

These configuration actions should only be performed on a single node within the cluster and in this example, will be performed from the alpha node.

1. Disable STONITH since no fencing is defined within this 4 node cluster:

```
[root@alpha ~]# pcs property set stonith-enabled=false
```

2. Create an LVM resource named phonetic_lvm with the exclusive=true parameter to ensure only the cluster can activate the LVM logical volume. As the nfsgroup has yet to be created, this command creates this resource group as well:

```
[root@alpha ~]# pcs resource create phonetic_lvm LVM volgrpname=phonetic_vg
exclusive=true --group nfsgroup
```

3. Verify cluster status to ensure the resource is running (highlighted in yellow):

```
[root@alpha ~]# pcs status
Cluster name: phonetic_cluster
Stack: corosync
Current DC: charlie (version 1.1.15-11.el7_3.2-e174ec8)-partition with quorum
Last updated: Thu Jan  5 14:49:19 2017 Last change: Thu Jan  5 14:49:08 2017
by root via cibadmin on alpha

4 nodes and 1 resource configured

Online: [ alpha bravo charlie delta ]

Full list of resources:

 Resource Group: nfsgroup
      phonetic_lvm       (ocf::heartbeat:LVM):   Started alpha

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

4. Create a filesystem monitoring resource for the /cluster_share NFS export

```
[root@alpha ~]# pcs resource create cluster_share Filesystem
device=/dev/phonetic_vg/phonetic_lv directory=/cluster_share fstype=ext4 --group
nfsgroup
```

5. Verify cluster status to ensure the phonetic_lvm and cluster_share resources are running (highlighted in yellow):

```
[root@alpha ~]# pcs status
Cluster name: phonetic_cluster
Stack: corosync
Current DC: charlie (version 1.1.15-11.el7_3.2-e174ec8)-partition with quorum
Last updated: Thu Jan  5 14:51:37 2017 Last change: Thu Jan  5 14:51:24 2017
by root via cibadmin on alpha

4 nodes and 2 resources configured

Online: [ alpha bravo charlie delta ]

Full list of resources:

 Resource Group: nfsgroup
      phonetic_lvm       (ocf::heartbeat:LVM):    Started alpha
      cluster_share      (ocf::heartbeat:Filesystem):    Started alpha

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

6. Create an nfsserver resource named nfs-daemon as part of the nfsgroup resource group:

```
[root@alpha ~]# pcs resource create nfs-daemon nfsserver
nfs_shared_infodir=/cluster_share/nfsinfo nfs_no_notify=true --group nfsgroup
```

7. Add the exportfs shares to the /nfsshare/exports directory, adding these resources to the nfsgroup resource group:

```
[root@alpha ~]# pcs resource create nfs-root exportfs
clientspec=192.168.168.0/255.255.255.0 options=rw,sync,no_root_squash
directory=/cluster_share/exports fsid=0 --group nfsgroup
[root@alpha ~]# pcs resource create nfs-export1 exportfs
clientspec=192.168.168.0/255.255.255.0 options=rw,sync,no_root_squash
directory=/cluster_share/exports/export1 fsid=1 --group nfsgroup
[root@alpha ~]# pcs resource create nfs-export2 exportfs
clientspec=192.168.168.0/255.255.255.0 options=rw,sync,no_root_squash
directory=/cluster_share/exports/export2 fsid=2 --group nfsgroup
```

8. Create the virtual IP address resource the NFS clients within your organization will use to access this specific NFS share. For the purposes of this paper, an IPv4 address of 192.168.168.227 was used:

```
[root@alpha ~]# pcs resource create nfs_ip IPaddr2 ip=192.168.168.227
cidr_netmask=24 --group nfsgroup
```

9. Once all resources and their respective constraints are created, verify the status of the cluster and notice that all resources are running on the same node (in this example, node delta) within the cluster (highlighted in yellow):

```
[root@alpha ~]# pcs status
Cluster name: phonetic_cluster
Stack: corosync
Current DC: charlie (version 1.1.15-11.el7_3.2-e174ec8)-partition with quorum
Last updated: Thu Jan  5 15:15:03 2017 Last change: Thu Jan  5 15:15:00 2017
by root via cibadmin on alpha

4 nodes and 8 resources configured

Online: [ alpha bravo charlie delta ]

Full list of resources:

 Resource Group: nfsgroup
      phonetic_lvm       (ocf::heartbeat:LVM):    Started delta
      cluster_share      (ocf::heartbeat:Filesystem):    Started delta
      nfs-daemon (ocf::heartbeat:nfsserver):      Started delta
      nfs-root   (ocf::heartbeat:exportfs):       Started delta
      nfs-export1        (ocf::heartbeat:exportfs):       Started delta
      nfs-export2        (ocf::heartbeat:exportfs):       Started delta
      nfs_ip     (ocf::heartbeat:IPaddr2):        Started delta
      nfs-notify (ocf::heartbeat:nfsnotify):      Started delta

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

5 Testing Active/Passive NFS Server in HA Cluster

The four node cluster should now be operational with a highly available NFS Server accessible on the virtual IP of 192.168.168.227. Testing the available service through normal operation and downed node operation is the focus of this chapter.

To verify the available shares from the virtual IP of the NFS server, perform the following command on a system/node residing in the same network segment as the deployed cluster. For this example, a host named echo with an IP of 192.168.168.221 was used.

```
[root@echo ~]# showmount -e 192.168.168.227
Export list for 192.168.168.227:
/cluster_share/exports/export1 192.168.168.0/255.255.255.0
/cluster_share/exports         192.168.168.0/255.255.255.0
/cluster_share/exports/export2 192.168.168.0/255.255.255.0
```

5.1 Map NFS share

To map the NFS share on the echo node and verify the contents of the export directories are visible, the following commands are executed from the echo host against the virtual IP of the active NFS server. Using the virtual IP when mapping the resources ensures high availability as it's a cluster resource mapped to all configured nodes within the cluster.

```
[root@echo ~]# mkdir /mount_cluster_share
[root@echo ~]# mount –o "vers=4" 192.168.168.227:export2 /mount_cluster_share
[root@echo ~]# ls /mount_cluster_share
testfile2
```

5.2 Test failover of NFS server between nodes

With the NFS export still mapped to the echo node, which is a node outside of the configured cluster yet on the same clusters network segment, testing the failover functionality of this highly available service between nodes is accomplished through the following process:

1. Determine which node is running the resource group nfsgroup by executing pcs status from one of the nodes within the cluster. The below example was taken from the alpha node showing node charlie as running the nfsgroup resource (highlighted in yellow):

   ```
   [root@alpha ~]# pcs status
   Cluster name: phonetic_cluster
   Stack: corosync
   Current DC: charlie (version 1.1.15-11.el7_3.2-e174ec8)-partition with
   quorum
   Last updated: Thu Jan 12 13:14:39 2017 Last change: Thu Jan 12 09:56:41
   2017 by hacluster via crmd on charlie

   4 nodes and 8 resources configured
   ```

```
Online: [ alpha bravo charlie delta ]

Full list of resources:

 Resource Group: nfsgroup
     phonetic_lvm       (ocf::heartbeat:LVM):   Started charlie
     cluster_share      (ocf::heartbeat:Filesystem):    Started charlie
     nfs-daemon (ocf::heartbeat:nfsserver):     Started charlie
     nfs-root   (ocf::heartbeat:exportfs):      Started charlie
     nfs-export1        (ocf::heartbeat:exportfs):      Started charlie
     nfs-export2        (ocf::heartbeat:exportfs):      Started charlie
     nfs_ip     (ocf::heartbeat:IPaddr2):       Started charlie
     nfs-notify (ocf::heartbeat:nfsnotify):     Started charlie

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

2. Place the node that is running the nfsgroup resource into standby mode. This is executed from the alpha node against the target node of charlie:

```
[root@alpha ~]# pcs cluster standby charlie
```

3. Verify cluster status to ensure the nfsgroup resource started on one of the other cluster nodes (highlighted in yellow) while showing the charlie node in standby and the other three nodes online and functional. This is executed from the alpha node:

```
[root@alpha ~]# pcs status
Cluster name: phonetic_cluster
Stack: corosync
Current DC: charlie (version 1.1.15-11.el7_3.2-e174ec8)-partition with
quorum
Last updated: Thu Jan 12 13:19:37 2017 Last change: Thu Jan 12 13:18:24
2017 by root via crm_attribute on alpha

4 nodes and 8 resources configured

Node charlie: standby
Online: [ alpha bravo delta ]

Full list of resources:

 Resource Group: nfsgroup
     phonetic_lvm       (ocf::heartbeat:LVM):   Started bravo
     cluster_share      (ocf::heartbeat:Filesystem):    Started bravo
     nfs-daemon (ocf::heartbeat:nfsserver):     Started bravo
```

```
     nfs-root   (ocf::heartbeat:exportfs):      Started bravo
     nfs-export1        (ocf::heartbeat:exportfs):      Started bravo
     nfs-export2        (ocf::heartbeat:exportfs):      Started bravo
     nfs_ip     (ocf::heartbeat:IPaddr2):       Started bravo
     nfs-notify (ocf::heartbeat:nfsnotify):     Started bravo

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

4. On the echo node, which still has the NFS service export2 share mounted, verify the hosts continued access to the test file with in the NFS share:

```
[root@echo ~]# ls /mount_cluster_share
testfile2
```

5. With the failover of the NFS service confirmed, remove the charlie node from standby thus adding it back into the cluster as an operational node:

```
[root@alpha ~]# pcs cluster unstandby charlie
```

6. Verify the cluster status to confirm the charlie node is back online and the nfsgroup resource is still resident on the bravo node:

```
[root@alpha ~]# pcs status
Cluster name: phonetic_cluster
Stack: corosync
Current DC: charlie (version 1.1.15-11.el7_3.2-e174ec8)-partition with
quorum
Last updated: Thu Jan 12 13:27:22 2017 Last change: Thu Jan 12 13:27:17
2017 by root via crm_attribute on alpha

4 nodes and 8 resources configured

Online: [ alpha bravo charlie delta ]

Full list of resources:

 Resource Group: nfsgroup
     phonetic_lvm       (ocf::heartbeat:LVM):   Started bravo
     cluster_share      (ocf::heartbeat:Filesystem):    Started bravo
     nfs-daemon (ocf::heartbeat:nfsserver):     Started bravo
     nfs-root   (ocf::heartbeat:exportfs):      Started bravo
     nfs-export1        (ocf::heartbeat:exportfs):      Started bravo
     nfs-export2        (ocf::heartbeat:exportfs):      Started bravo
     nfs_ip     (ocf::heartbeat:IPaddr2):       Started bravo
     nfs-notify (ocf::heartbeat:nfsnotify):     Started bravo
```

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

6 RHEL High Availability WEB UI Management

With the installation of the high availability add-on software package of pcs from Section 3.3, upon completion there is a Web UI available for configuration and/or monitoring of your clustered node environment. From any system on the same network segment as one of the pcs configured nodes, open a browser to the following URL, specifying one of the nodes IP address or DNS name that's authorized within the cluster. This brings up the PCSD Web UI login screen. For this example, the alpha node is used.

```
https://alpha:2224
```

Note: The PCSD Web UI URL uses the https protocol and a self-signed SSL certificate to help secure the configuration of your clustered environment.

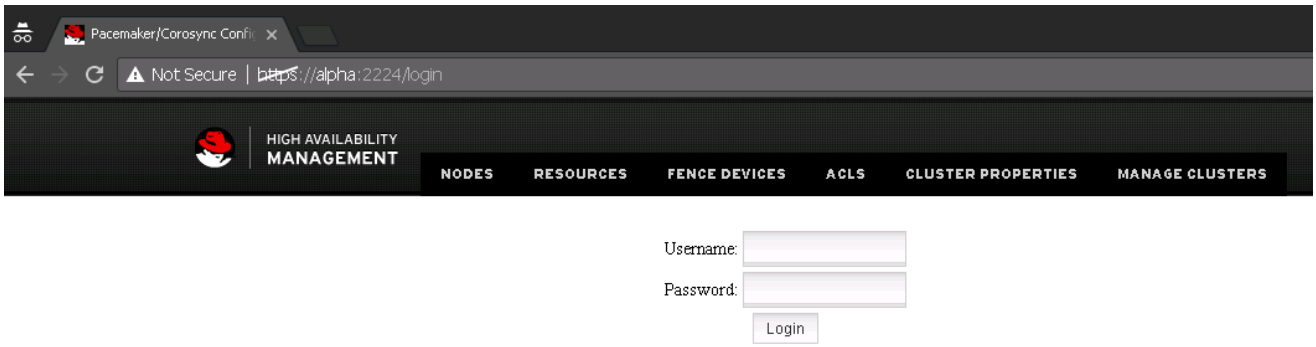

Figure 25 PCSD Web UI

The login credentials used to access this Web UI are the same as defined in Section 3.4; the administration account for the pcs service. This will be username and password of *hacluster*.

6.1 Add existing cluster

Upon initial login to the PCSD Web UI, the cluster will need to be added to the Manage Cluster screen for any further actions through the Web UI. From the below screen, add the first node in the cluster (alpha).

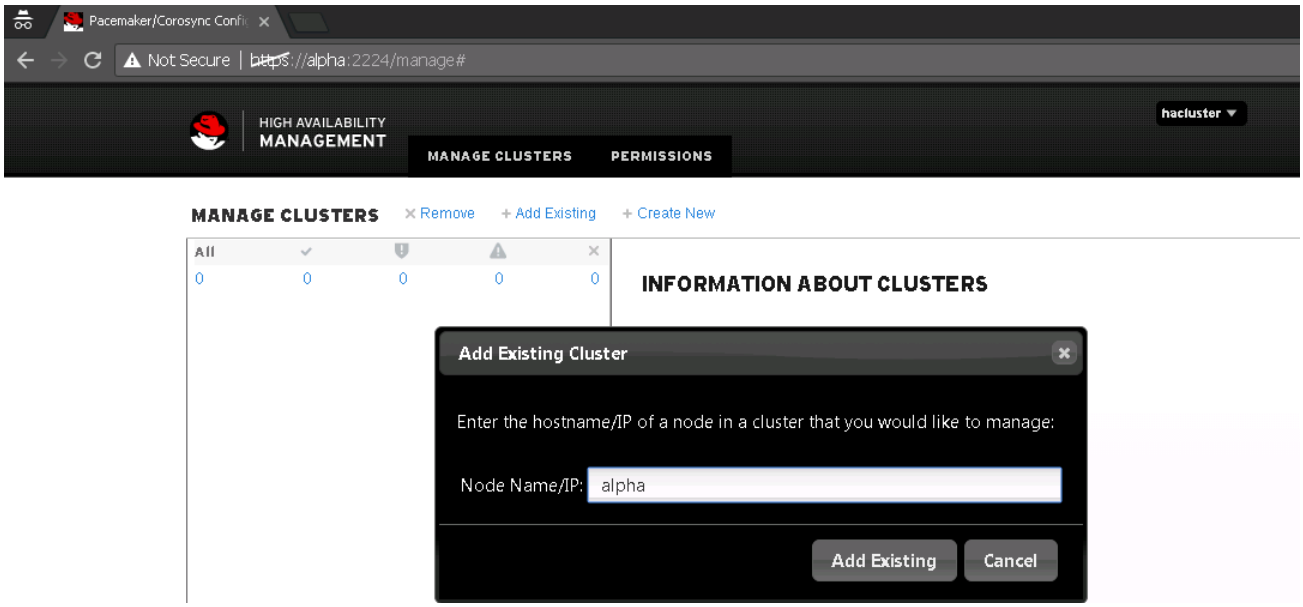

Figure 26 Add existing cluster

6.2 Information about managed cluster

Once the alpha node has been added, the PCSD Web UI manage cluster screen provides additional information pertaining to the cluster where the alpha node is configured. This is done by simply checking the cluster name phonetic_cluster and reviewing the node, resources and fence device information to the right.

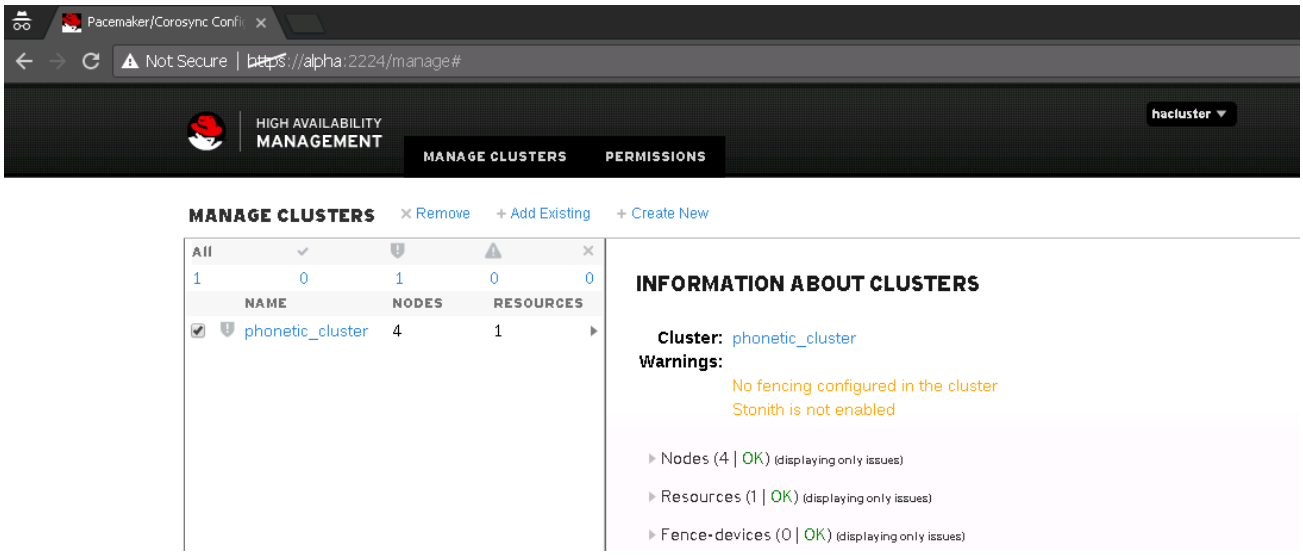

Figure 27 Information about clusters

6.3 Nodes in the managed cluster

To review any of the individual nodes that make up the phonetic_cluster through the PCSD Web UI, click phonetic_cluster to open the Nodes management screen. From here, each node can be selected to drill down into more detail concerning cluster daemons, running resources, node attributes and fence levels. Management actions on the node can also be taken from within this screen such as **Start**, **Stop**, **Restart**, **Standby**, **Maintenance** and **Configure Fencing**.

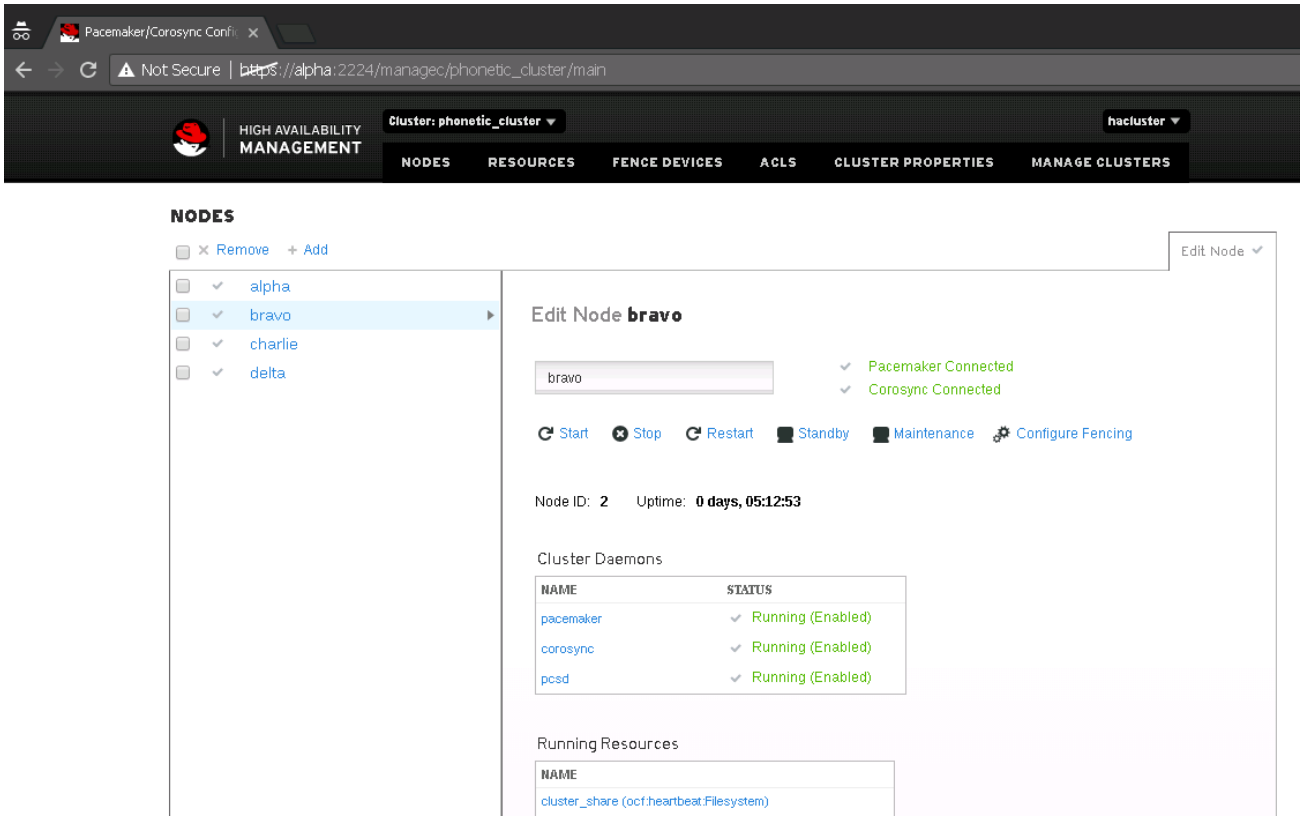

Figure 28 Node configuration

6.4 Resources in the managed cluster

The review any of the defined resource groups and the configured constraints therein through the pcs Web UI, simply click on the Resources tab in the middle of the top navigation bar. This will take you to the defined resources for the currently monitored cluster, which for this example is the phonetic_cluster. You can add additional clusters to this pcs Web UI thus providing you the ability to review multiple cluster configurations and their resources from a simple Web UI.

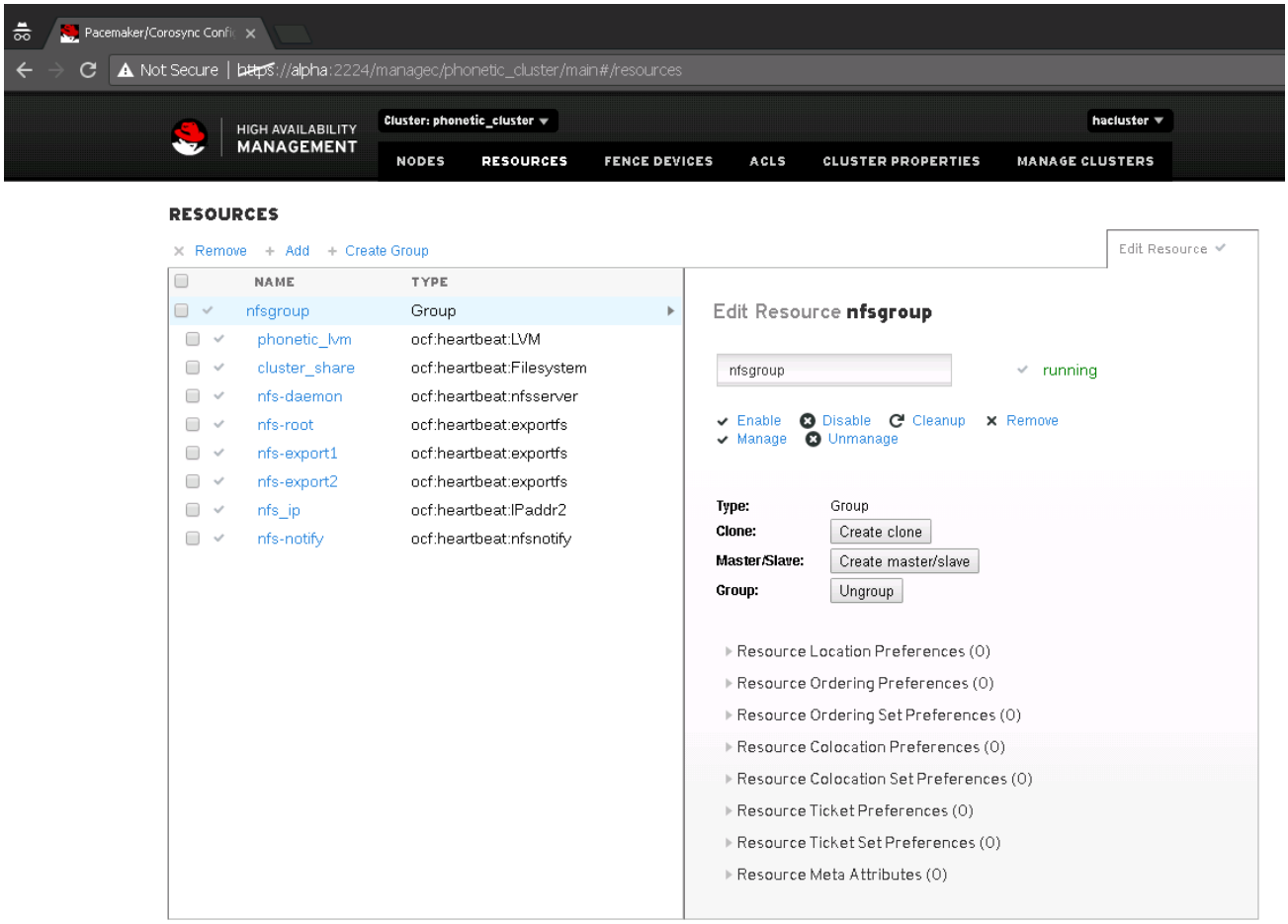

Figure 29 PCS Web UI Resource Screen

More information on the PCSD Web UI can be found within Red Hat High Availability Add-On Configuration and Management Reference Overview.

A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Dell TechCenter is an online technical community where IT professionals have access to numerous resources for Dell EMC software, hardware and services.

Dell.com/StorageResources on Dell TechCenter provide expertise that helps to ensure customer success on Dell EMC Storage platforms.

A.1 Related resources

For RHEL 7.3 documentation from the vendor, see the following:

- Red Hat High Availability Add-On Reference
- Red Hat High Availability Add-On Datasheet
- Red Hat Enterprise Linux 7 Installation Guide
- Red Hat Enterprise Linux Cluster, High Availability, and GFS Deployment Recommended Practices
- Red Hat High Availability Add-On Configuration and Management Reference Overview

See the following referenced or recommended Dell publications:

- *Dell Storage Compatibility Matrix*: http://en.community.dell.com/dell-groups/dtcmedia/m/mediagallery/19856862.aspx
- *Red Hat Enterprise Linux 6 or 7 Host Configuration* http://en.community.dell.com/techcenter/extras/m/white_papers/20438884
- *RHEL 6.x Best Practices for Dell Compellent Storage Center* http://en.community.dell.com/techcenter/extras/m/white_papers/20437964.aspx
- *Switch Configuration Guides for PS Series or SC Series SANs*: http://en.community.dell.com/techcenter/storage/w/wiki/4250.switch-configuration-guides-for-ps-series-or-sc-series-sans
- Integrated Dell Remote Access Controller 8 (iDRAC8) and iDRAC7 Version 2.20.20.20 RACADM Command Line Interface Reference Guide