Dell Compellent Storage Center

Redhat Enterprise Linux (RHEL) 6x

Best Practices Guide

Dell Compellent Technical Solutions Group

July 2013

THIS BEST PRACTICES GUIDE IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

Table of Contents

Document Revisions

Date	Revision	Author	Comments
05/22/2013	1.0	Daniel Tan	Refresh including rewording & new template
10/2/2013	1.1	Daniel Tan	Added “ext4 and SCSI UNMAP” section

1 Preface

1.1 Audience

The audience for this document is system administrators who are responsible for the setup, maintenance and management of *NIX based platforms which are used with Dell Compellent Storage Center products. Readers should have basic conceptual and working knowledge of *NIX platforms and the Dell Compellent Storage Center.

1.2 Purpose

This document provides an overview of Linux servers and introduces best practice guidelines for configuring volume discovery, multipath, queue depth management and more on Linux when using the Dell Compellent Storage Center.

Due to the wide variety of Linux distributions and the variances of each, the information discussed in this document may vary slightly. Users should be aware of these differences and are encouraged to consult specific vendor documentation available for their distribution of Linux. This guide will address Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES).

It is important to note that as is common with Linux platforms, there are multiple ways in which to accomplish what is discussed in this document. This guide does not contain every possible way nor is it intended to be the definitive guide for all scenarios. This document is intended as a starting frame of reference for end users.

Also note that this guide will focus almost exclusively on the command line interface (CLI). Many Linux distributions have created graphical tools to achieve many of these tasks. This guide simply focuses on the CLI because it is the most universal.

1.3 Customer Support

Dell Compellent provides live support 1-866-EZSTORE (866.397.8673), 24 hours a day, 7 days a week, 365 days a year. For additional support, email Dell Compellent at support@compellent.com. Dell Compellent responds to emails during normal business hours.

2 Introduction

The goal of this paper is to provide guidance to Linux system administrators who will be utilizing storage presented from the Dell Compellent Storage Center SAN. It is also intended to be useful for storage administrators who manage the Dell Compellent SAN in an environment that has Linux- based hosts connecting to the SAN.

3 Managing Volumes

Understanding how volumes are managed in Linux systems requires a basic understand of the /sys pseudo file system. The /sys file system is a structure of files that allow for interaction with various elements of the kernel and modules. Many of the files can be read to discover current values, while others can be written to trigger events. This is generally done making use of the commands “cat” and “echo” with a redirect (verses opening them with a traditional text editor).

To interact with the HBAs (including virtual software iSCSI HBAs) values are written to files in /sys/class/scsi_host/ folder. Each HBA (each port on a multiport card counting as a unique HBA) has its own hostX folder containing files for issuing scans and reading HBA parameters. Unless otherwise noted, the ones discussed below will exist on QLogic and Emulex cards, as well as software iSCSI.

4 Scanning for New Volumes

Starting with kernel version 2.6, the modules needed for the QLogic 24xx series cards and the Emulex cards were included in the base kernel. Red Hat version 4 forward and SLES version 10 included both of these drivers by default. The following instructions apply to the default modules included. If the proprietary driver has been installed from either QLogic or Emulex, consult the specific documentation for instructions.

Between 2.6.9 and 2.6.11 a major overhaul of the SCSI stack was implemented. As a result, instructions are different between pre-2.6.11 and 2.6.11 and later kernels.

There are no negatives effects from rescanning an HBA, therefore it is not necessary to explicitly know which host needs to be rescanned. It is just as easy to rescan all of them when mapping a new volume.

Note

Linux systems cannot discover LUN 0 on the fly. LUN 0 can only be discovered at boot time and is thus reserved for the OS Volume in boot from SAN environments. All other volumes should be mapped at LUN 1 or greater.

4.1 Kernel Version 2.6-2.6.9 (RHEL 4, SLES 9)

The following applies to QLogic and Emulex fibre channel HBAs. This will rescan host0 and discover any new volumes presented to host0 only. To rescan the other hosts, simply substitute '0' for the number of the host.

```
# for i in `ls /sys/class/scsi_host/`; do echo 1 >> /sys/class/scsi_host/$i/issue_lip; echo "- - -"
>> /sys/class/scsi_host/$i/scan; done
```

There will be no output from either of the commands. Any new LUNs will be logged in dmesg and to the system messages.

4.2 Kernel Versions 2.6.11+ (RHEL 6/5, SLES 11/10)

The following applies to QLogic and Emulex HBAs, as well as software iSCSI.

```
# for i in `ls /sys/class/scsi_host/`; do echo "- - -" >> /sys/class/scsi_host/$i/scan; done
```

Again, there will be no output from the command. Any new LUNs will be logged in dmesg and to the system messages.

5 Partitions and Filesystems

As a block level SAN, the Dell Compellent Storage Center will take any partition and filesystem scheme supported by the OS. However, there are some things to take into consideration when designing a scheme

5.1 Partitions

For volumes other than the primary boot drive, partition tables are unnecessary. As a result, in many situations where only one partition would be required it is better not to use one. Not using a partition table makes expanding volumes at a later time significantly easier. In order to resize a volume with a partition table, the existing table must be deleted and the new table must be carefully recreated using the same starting point.

This process can result in unreadable filesystems. By not using a partition table, volumes can be expanded in fewer steps, and more recent systems can do the expansion online. Consult the appendix on expanding a volume for instructions and limitations.

The following example shows creating an ext3 file system on a device without a partition table. Note the prompt to proceed, this can be avoided by adding –F to the command.

```
[root@local ~]# mkfs.ext3 -L dataVol /dev/sdc mke2fs 1.39 (29-May-2006)
/dev/sdc is entire device, not just one partition!
Proceed anyway? (y,n) y
Filesystem label=dataVol
OS type: Linux
Block size=4096 (log=2) Fragment size=4096 (log=2)
10485760 inodes, 20971520 blocks
```

5.2 LVM

When deciding whether to use LVM, a few things should be considered. For most systems it is not possible to mount a View Volume of an LVM back to the same server for recovery with the original volume still mounted without complicated manual tasks. Many of the benefits of LVM are already provided for at the Compellent level. Due to the complication of View Volumes, LVM are generally not recommended. LVMs should be used in the case where a specific benefit is desired that is not provided by the SAN.

Currently Red Hat Enterprise Server 5.4 is the only release that can manage the duplicate LVM signatures with built-in tools.

5.3 Disk Labels and UUIDs for Persistence

All modern Linux operating systems are capable of discovering multiple volumes from the

Dell Compellent Storage Center. These new disks are given a device designation of

/dev/sda, /dev/sdb, etc depending upon how they are discovered by the Linux operating system via the various interfaces connecting the server to the storage.

The /dev/sdx names are used to designate the volumes for a myriad of things, but most importantly, mount commands including /etc/fstab. In a static disk environment, the /dev/sdx name works well for entries in the /etc/fstab file.

However, in the dynamic environment of fiber channel or iSCSI connectivity the Linux operating system lacks the ability to track these disk designations persistently through reboots and dynamic additions of new volumes via rescans of the storage subsystems.

There are multiple ways to ensure that disks are referenced by persistent names. This guide will cover using Disk Labels and UUIDs. Disk Labels or UUIDs should be used with all single pathed volumes.

Disk labels are also exceptionally useful when scripting Replay recovery. In the example where a view of a production volume is mapped to a backup server, it is not necessary to know what drive letter the View Volume is assigned. Since the label is written to the filesystem, the label goes with the view and can easily be mounted or manipulated.

Note

Disk labels will not work in a multipathed environment, and should not be used; multipath device names are persistent by default and will not change. Multipathing does support aliasing the multipath device names for human readable names. Consult the Alias section under Multipath Configuration for more information.

5.4 New Filesystem Volume Label Creation

The mke2fs and mkfs.reiserfs commands with the –L and –l LabelName added to the standard file system creation commands, erases any previous filesystem tables, **destroys** the pointers to existing files, creates a new filesystem and a new label on the disk.

The examples below create a new file system with the label FileShare for the various major filesystems types.

Caution

The process below will format the volume destroying all data on that volume.

```
# mke2fs -j -L FileShare /dev/sdc
# mkfs -t ext3 -L FileShare /dev/sdc
# mkfs.reiserfs -l FileShare /dev/sdc
```

5.5 Existing Filesystem Volume Label Creation

To add or change the volume label without destroying data on the disk, use the following command. These commands can be performed while the filesystem is mounted.

```
# e2label /dev/sdb FileShare
```

It is also possible to set the filesystem label using the -L option of tune2fs.

```
# tune2fs -L FileShare /dev/sdb
```

5.6 Discover Existing Labels

To discover the label of an existing partition the following simple command can be used.

```
# e2label /dev/sde
FileShare
```

In this output, 'FileShare' is the volume label.

5.7 /etc/fstab Example

```
LABEL=root          /        ext3       defaults     1    1
LABEL=boot          /boot ext3          defaults     1    2
LABEL=FileShare     /share ext3         defaults     1    2
```

The LABEL= syntax can be used in a variety of places including mount commands and Grub configuration. Disk labels can also be referenced as a path for applications that do not recognize the LABEL= syntax. For example, the volume designated by the label FileShare can be accessed at the path '/dev/disk/by-label/FileShare'.

5.8 Swap Space

Swap space can also be labeled, however only at the time of creation. This isn't a problem since no static data is stored in swap. To label an existing swap partition, follow these steps.

```
# swapoff /dev/sda1
# mkswap -L swapLabel /dev/sda1
# swapon LABEL=swapLabel
```

The new swap label can be used in /etc/fstab just like any volume label.

5.9 UUIDs

An alternative to disk labels is UUIDs. They are static and safe for use anywhere, however, their long length can make them awkward to work with. UUID is assigned at filesystem creation.

A UUID for a specific filesystem can be discovered using 'tune2fs –l'.

```
[root@local ~]# tune2fs -l /dev/sdc tune2fs 1.39 (29-May-2006) Filesystem volume name:
          dataVol
Last mounted on:        <not available>
Filesystem UUID:        5458d975-8f38-4702-9df2-46a64a638e07 [Truncate]
```

Another simple way to discover the UUID of a device or partition is to do a long list on the /dev/disk/by-uuid directory.

```
[root@local ~]# ls -l /dev/disk/by-uuid total 0
lrwxrwxrwx 1 root root 10 Sep 15 14:11 5458d975-8f38-4702-9df2-
46a64a638e07 -> ../../sdc
```

From the output above, we discover that the UUID is '5458d975-8f38-4702-9df2-46a64a638e07'

Disk UUIDs can be used in /etc/fstab or any place were persistent mappings is required. Below is an example of its use in /etc/fstab.

```
          /dev/VolGroup00/LogVol00    /          ext3      defaults     1 1
          LABEL=/boot                 /boot      ext3      defaults     1 2
UUID=8284393c-18aa-46ff-9dc4-0357a5ef742d       swap swap defaults     0 0
```

As with disk labels, if an application requires an absolute path, the links created in /dev/disk/by-uuid should work in almost all situations.

5.10 GRUB

In addition to /etc/fstab, GRUBs config file should also be reconfigured to reference LABEL or UUID. The example below shows using a label for the root volume, UUID can be used the same way. Labels or UUIDs can also be used for "resume" if needed.

```
title Linux 2.6 Kernel root (hd0,0)
kernel (hd0,0)/vmlinuz ro root=LABEL=RootVol rhgb quiet initrd (hd0,0)/initrd.img
```

6 Unmapping Volumes

A Linux system stores information on each volume presented to it. Even if a volume is unmapped on the Dell Compellent storage array, the Linux system will retain information about that volume until the next reboot. If the Linux system is presented with a volume from the same target using the same LUN number again, it will reuse the old data on the volume. This can result in complications and misinformation.

Therefore, it is a best practice to always delete the volume information on the Linux side after the volume has been unmapped. This will not delete any data stored on the volume itself, just the information about the volume stored by the OS (volume size, type, etc.).

Determine the drive letter of the volume that will be unmapped. For example, /dev/sdc.

Note

In a multipath environment, it is a best practice to flush the multipath device entry prior to removing the volume information. This can be done in a couple of different ways. Refer to the multipath section of this document for more details.

Delete the volume information on the Linux OS with the following command replacing sdc with the correct device name.

```
echo 1 > /sys/block/sdc/device/delete
```

7 Useful Tools

Determining which Dell Compellent volume correlates to a specific Linux device can be tricky, but the following tools can be useful and many are included in the base install.

7.1 lsscsi

lsscsi is a tool that parses information from the /proc and /sys psudofilesystems into a simple human readable output. Although not currently included in the base installs for either Red Hat 5 or SLES 10, it is in the base repository and can be easily installed.

```
[root@local ~]# lsscsi
                [0:0:0:0]    disk    COMPELNT  Compellent  Vol   0402   -
                [0:0:1:0]    disk    COMPELNT  Compellent  Vol   0402   -
                [0:0:2:0]    disk    COMPELNT  Compellent  Vol   0401   /dev/sda
                [0:0:3:0]    disk    COMPELNT  Compellent  Vol   0402   -
                [0:0:3:5]    disk    COMPELNT  Compellent  Vol   0401   /dev/sdc
```

This output shows two drives from Dell Compellent, and it also shows that three front end ports are visible but are not presenting a LUN 0. This is the expected behavior. There are multiple modifies for lsscsi that provide even more detailed information.

The first column above shows the [host:channel:target:lun] designation for the volume. The first number corresponds to the local HBA hostX that the volume is mapped to. Channel is the SCSI bus address which will always be zero. The third number correlates to the Compellent front end ports (targets). The last number is the LUN that the volume is mapped on.

7.2 scsi_id

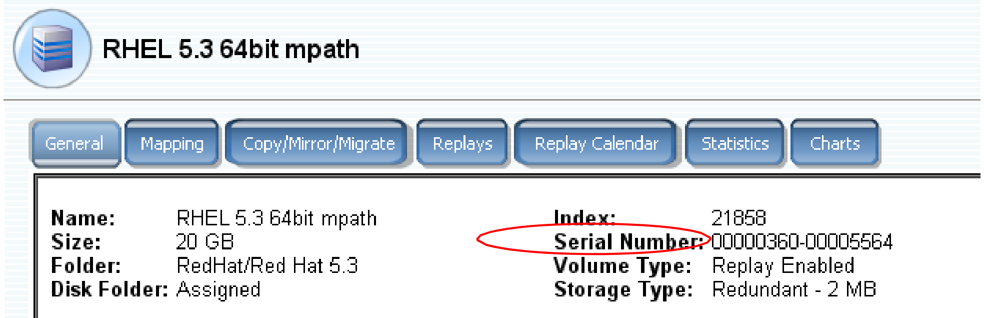
scsi_id can be used to report the WWID of a volume and is available in all base installations. This wwid can be matched to the volume serial number reported in the Dell Compellent System Manager GUI for accurate correlation.

7.2.1 Redhat Linux 6 and Newer

```
for i in `cat /proc/partitions | awk {'print $4'} | grep sd`
    do
        echo "Device: $i WWID: `scsi_id --page=0x83 --whitelisted --device=/dev/$i`"
    done | sort -k4
```

7.2.2 Redhat Linux 5 and Older

```
for i in `cat /proc/partitions | awk {'print $4'} | grep sd`
    do
        echo "Device: $i WWID: `scsi_id -g -u -s /block/$i`"
    done | sort -k4
```



RHEL 5.3 64bit mpath

```
Name:        RHEL 5.3 64bit mpath         Index:          21858
Size:        20 GB                        Serial Number:  00000360-00005564
Folder:      RedHat/Red Hat 5.3           Volume Type:    Replay Enabled
Disk Folder: Assigned                     Storage Type:   Redundant - 2 MB
```

Figure 1: Observer the Volume Serial Number in the Volume Properties General Tab

The first part of the WWID is Dell Compellent's unique ID, the middle part is made up of the controller number in hex and the last part is the serial number of the volume. To ensure correct correlation in environments with multiple Dell Compellent Storage Centers, be sure to check the controller number as well.

The only situation where the two numbers would not correlate is if a Copy Migrate had been performed. In this case, a new serial number is assigned on the Dell Compellent side, but the old WWID is used to present to the server so that the server is not disrupted.

7.3 /proc/scsi/scsi

Viewing the contents of this file can provide information about LUNs and targets on systems that do not have lsscsi installed. However, it is not easy to correlate to a specific device.

```
[root@local ~]# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
Vendor: COMPELNT Model: Compellent Vol    Rev: 0402
Type:   Direct-Access   ANSI SCSI revision: 04
Host: scsi0 Channel: 00 Id: 01 Lun: 00
                 Vendor: COMPELNT Model: Compell ent Vol Rev:    0402
                 Type:    Direct-Access          ANSI            SCSI  revision:   04
Host: scsi0 Channel: 00 Id: 02 Lun: 00
Vendor: COMPELNT Model: Compellent Vol    Rev: 0401
Type:   Direct-Access   ANSI SCSI revision: 04
Host: scsi0 Channel: 00 Id: 03 Lun: 00
              Vendor: COMPELNT Model: Compell    ent Vol Rev:    0402
              Type:        Direct-Access         ANSI            SCSI  revision:   04
              Host: scsi0 Channel: 00 Id: 03 Lun: 05
              Vendor       : COMPELNT Model:     ent Vol Rev:    0401
              Type:        Compell               ANSI            SCSI  revision:   04
```

7.4 dmesg

The output from dmesg can be useful for discovering what device name was assigned to a recently discovered volume.

```
SCSI device sdf: 587202560 512-byte hdwr sectors (300648 MB)
sdf: Write Protect is off sdf: Mode Sense: 87 00 00 00
SCSI device sdf: drive cache: write through
SCSI device sdf: 587202560 512-byte hdwr sectors (300648 MB)
sdf: Write Protect is off sdf: Mode Sense: 87 00 00 00
SCSI device sdf: drive cache: write through sdf: unknown partition table
sd 0:0:3:15: Attached scsi disk sdf
sd 0:0:3:15: Attached scsi generic sg13 type 0
```

The above output is taken just after a host rescan and shows that a 300 GB volume has been discovered and assigned as /dev/sdf.

8 Software iSCSI

Most major Linux distributions have been including a software iSCSI initiator for at least a few releases. Red Hat includes it in both versions 4 and 5, and SUSE includes it in versions 9, 10 and 11. The package can be installed using the respective package management systems.

Both Red Hat Enterprise Linux (RHEL) and SuSE Linux Enterprise Server (SLES) utilize the open-iscsi implementation of software iSCSi on the Linux platform. RHEL has included iSCSI support since version 4.2 (dating back to October 2005) and SLES has included open-iscsi since version 10.1 (dating back to May 2006). Because iSCSI has now been included in several releases and has been refined for several years, it is now considered to be a mature technology.

While iSCSI is considered to be a mature technology that allows organizations to economically scale into the world of enterprise storage, it has grown in complexity at both the hardware and software layers. The scope of this documented is limited to the default Linux iSCSI software initiator (`open- iscsi`). For more advanced implementations (for example leveraging iSCSI HBAs, or drivers that make use of iSCSI offload engines) please consult the associated vendor's documentation and support services.

For instructions on setting up an iSCSI network topology, please consult the Storage Center Connectivity Guide.

This guide first covers some common elements, and then will walk through configuring an iSCSI volume first on Red Hat, then on SUSE, finishing with some more common elements. For all other distributions, please consult the documentation from that distribution.

8.1 Network Configuration

The system being configured will require a network port that can communicate with the iSCSI ports on the Dell Compellent Storage Center. This does not necessarily have to be a dedicated port, but is highly recommended and is generally considered to be a best practice.

The most important thing to consider when configuring an iSCSI volume is the network path. If it is important that iSCSI traffic be dedicated a distinct port, or if multipathing is involved, controlling what traffic is carried on which ports is important. This can be achieved at multiple different levels, which is mostly a matter of choice for the administrator and a function of what network infrastructure is available.

It is a best practice to separate traffic by subnet. In general, most administrators will dedicate a second network port to iSCSI traffic. This port will be in a different subnet than the rest of the network traffic. This way, the TCP/IP layer handles the proper routing out the dedicated port.

This is also a best practice for multipathing. In a fully redundant multipath environment, one switch fabric and corresponding ports should be in one subnet and the other in a different subnet. This isolates the iSCSI traffic to the proper ports on the server side.

If distinct subnets are not an option, two other options are available:

- Route traffic at the network layer by defining static routes
- Route traffic at the iSCSI level via configuration

Deciding on which option to use is an administrator's choice.

The following directions assume that a network port has already been configured and can communicate with the Dell Compellent iSCSI ports.

8.2 Red Hat Configuration

The necessary tools for Red Hat servers are contained in the package 'iscsi-initiator-utils' and can be installed with yum with the following command.

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

The iSCSI software initiator consists of two main components, the daemon (which runs in the background and handles connections and traffic), and the administration utility (which is used to configure and modify connections). Before anything can be configured the daemon needs to be started. It should also be configured to start automatically in most cases.

```
[root@local ~]# /etc/init.d/iscsi start
Turning off network shutdown. Starting iSCSI daemon:     [ OK ] Setting up iSCSI targets:
iscsiadm: No records found      [ OK ] [root@local ~]#
[root@local ~]#
[root@local ~]# chkconfig iscsi on
```

The next step is to discover the iqn for the Dell Compellent ports. For Dell Compellent Storage Center 4.x the discovery command needs to be run against each primary iSCSI port on the system. Starting with Storage Center 5.0 running with virtual ports enabled, the discovery command only needs to be run against the control port. It will report back all the iqns on the system.

In the example below, the iSCSI ports on the Dell Compellent system have the IP addresses 10.10.3.1 and 10.10.3.2

```
[root@local ~]# iscsiadm -m discovery -t sendtargets -p 10.10.3.1
10.10.3.1:3260,0 iqn.2002-03.com.compellent:5000d3100000670c
[root@local ~]# iscsiadm -m discovery -t sendtargets -p 10.10.3.2
10.10.3.2:3260,0 iqn.2002-03.com.compellent:5000d3100000670d
```

The iSCSI daemon saves the nodes in /var/lib/iscsi and will automatically log into them when the daemon starts. The below command instructs the software to log into all known nodes.

```
[root@local iscsi]# iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100000670c, portal: 10.10.3.1,3260] Logging in to [iface:
default, target: iqn.2002-
03.com.compellent:5000d3100000670d, portal: 10.10.3.2,3260] Login to [iface: default,
target: iqn.2002-
03.com.compellent:5000d3100000670c, portal: 10.10.3.1,3260]:
successful
Login to [iface: default, target: iqn.2002-
03.com.compellent:5000d3100000670d, portal: 10.10.3.2,3260]:
successful
```

After running this command, a server object can now be created on the Dell Compellent Storage Center.

After creating the server object and mapping a volume to the initiator, the virtual HBA can be rescanned to discover the new LUN.

```
# echo "- - -" >> /sys/class/scsi_host/host6/scan
```

As long as the iscsi daemon is set to start on boot, the system will automatically login to the Dell Compellent targets and discover all volumes.

8.3 SuSE Configuration

For SUSE systems, the package that provides the iSCSI initiator is named “open-iscsi” (it is not necessary to install the “iscsitarget” package).

The iSCSI software initiator consists of two main components, the daemon (which runs in the background and handles connections and traffic), and the administration utility (which is used to configure and modify connections). Before anything can be configured the daemon needs to be started. It should also be configured to start automatically in most cases.

```
local:~ # /etc/init.d/open-iscsi start
Starting iSCSI initiator service:done iscsiadm: no records found!
Setting up iSCSI targets:         unused local:~ #
local:~ #
local:~ # chkconfig open-iscsi on

local:~ # iscsiadm -m discovery -t sendtargets -p 10.10.3.1
10.10.3.1:3260,0 iqn.2002-03.com.compellent:5000d3100000670c local:~ # iscsiadm -m
discovery -t sendtargets -p 10.10.3.2
10.10.3.2:3260,0 iqn.2002-03.com.compellent:5000d3100000670d
```

The system stores information on each target. After the targets have been discovered, they can be logged into. This creates the virtual HBAs as well as any disks devices for volumes mapped at login time.

```
local:~ # iscsiadm -m node --login
```

The last step is to configure the system to automatically login to the targets when the initiator starts, which should be configured to start at boot time.

```
[root@local ~]# iscsiadm -m node --op=update --name=node.startup\
--value=automatic
```

8.4 Scanning for New Volumes

Volumes are discovered on the fly (the same as for physical HBAs).

```
[root@local ~]# echo “- - -“ >> /sys/class/scsi_host/host3/scan
```

The host number just needs to be replaced with the correct host for the target connection.

8.5 /etc/fstab Configuration

Since iSCSI is dependent on the network connection being up, any volumes that are added to /etc/fstab need to be designated as network dependant. The example below will mount the volume labeled iscsiVol. The important part is adding “_netdev” to the options.

```
LABEL=iscsiVOL    /mnt/iscsi     ext3    _netdev        0 0
```

8.6 iSCSI Timeout Values

In the event of a failure in the SAN environment that causes a controller failover event (in a dual-controller system), the iSCSI daemon needs to be configured to wait for a sufficient amount of time to allow the failure recovery to occur. A fail over between Storage Center controllers takes approximately 30 seconds to complete, so it is a best practice to configure the iSCSI initiator to queue for 60 seconds before failing.

When using iSCSI in a multipath environment, the iSCSI daemon can be configured to fail a path very quickly. It will then pass outstanding I/O back to the multipath layer. If dm-multipath still has an available route, the I/O will be resubmitted to the live route. If all available routes are down, dm-multipath will queue I/O until a route becomes available. This allows an environment to sustain failures at the network and storage levels.

For the iSCSI daemon, the following configuration settings directly affect iSCSI connection timeouts:

To control how often a NOP-Out request is sent to each target, the following value can be set:

```
node.conn[0].timeo.noop_out_interval = X
```

Where X is in seconds and the default is 10 seconds.

To control the time out for the NOP-Out, the `noop_out_timeout` value can be used:

```
node.conn[0].timeo.noop_out_timeout = X
```

Again X is in seconds and the default is 15 seconds. The next iSCSI timer that will need to be modified is:

```
node.session.timeo.replacement_timeout = X
```

Again X is in seconds.

`replacement_timeout` will control how long to wait for session re-establishment before failing pending SCSI commands and commands that are being operated on by the SCSI layer's error handler up to a higher level like multipath, or to an application if multipath is not being used.

Remember, from the NOP-Out section that if a network problem is detected, the running commands are failed immediately. There is one exception to this and that is when the SCSI layer's error handler is running. To check if the SCSI Error Handler is running, `iscsiadm` can be run as:

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

With the following output:

```
Host Number: X State: Recovery
```

When the SCSI Error Handler is running, commands will not be failed until node.session.timeo.replacement_timeout seconds is modified.

To modify the timer that starts the SCSI Error Handler, “echo” X into the following device's sysfs file:

```
echo X > /sys/block/sdX/device/timeout
```

where X is in seconds, or depending on the Linux distribution, this can also be achieved by

modifying the respective udev rule. To modify the udev rule, open /etc/udev/rules.d/60-raw.rules, and add the following
lines:

```
ACTION=="add", SUBSYSTEM=="scsi" , SYSFS{type}=="0|7|14", \ RUN+="/bin/sh -c 'echo 60 >
/sys$$DEVPATH/timeout'"
```

8.7 Multipath Timeout Values

The following line will tell dm-multipath to queue I/O in the event that all paths are down. This line allows multipath to wait for the Storage Center to recover in the event of a fail over event:

```
features          "1 queue_if_no_path"
```

Timeout and other connection settings are statically created during the discovery step and written to config files in /var/lib/iscsi/*.

There is no specific timeout value that is appropriate for every environment. In a multipath fibre channel environment, it is recommended to set timeout values on the FC HBA to five (5) seconds. However, additional caution should be taken when determining the appropriate value to use in an iSCSI configuration. Since iSCSI is often used on shared network switches, it is extremely important that necessary consideration be made to avoid inadvertent non-iSCSI network traffic from interfering with the iSCSI storage traffic.

It is important to take into consideration all the different variables that go into the environment's configuration and thoroughly test failover scenarios (port, switch, controller, etc) before deploying into a production environment.

9 Server Configuration

9.1 Server Timeout Values

These settings need to be configured for Linux systems that are connected to Dell Compellent systems **without multipath**. Systems that do not have these settings could have volumes go read-only during controller failover. Do not set these values on multipath systems. Consult the multipath configuration section for the correct settings.

This section covers configuration of QLogic 2xxx HBAs that utilize the qla2xxx module as well as the Emulex LightPulse HBAs that utilize the lpfc module. This section will cover both the open source default qla2xxx module and the proprietary QLogic release version.

HBA BIOS settings should be configured to the specifications recommended by Dell Compellent for the specific HBA and Storage Center version. These documents can be located at the Dell Compellent Knowledge Center portal at http://kc.compellent.com.

9.2 Module Settings

Depending on the version of Linux, the method for setting the module parameter will vary. This guide will explicitly cover Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server 10. The setting should be the same on any Linux system using a 2.6.11 or later kernel. For other distributions, please consult the specific documentation for that distribution. It has only been tested on Red Hat and SuSE systems.

The important module parameter for the QLogic cards is qlport_down_retry, and for the Emulex cards it is lpfc_devloss_tmo. These settings determine how long the system waits to destroy a connection after losing connectivity with the port. During a Storage Center controller failover, the WWN for the active port will disappear from the fabric momentarily before resuming on the reserve port on the other controller. This process can take anywhere from 5 to 60 seconds to fully propagate through a fabric. As a result, the default timeout of 30 seconds is too short and therefore the value needs to be changed to 60 seconds.

9.2.1 Redhat Linux 6 and Newer

For RHEL 6 using the default open source driver, add or update the following file inside of the /etc/modprobe.d directory.

For Qlogic

Create the file **qla2xxx.conf**

Append this line into the file if it already exists

```
options qla2xxx qlport_down_retry=60
```

For Emulex

Create the file **lpfc.conf**
Append this line into the file if it already exists

```
options lpfc lpfc_devloss_tmo=60
```

9.2.2 Redhat Linux 5 and Older

For RHEL 5 using the default open source driver, add or update the following line in /etc/modprobe.conf with the qlport_down_retry or lpfc_devloss_tmo variable.

For Qlogic

```
options qla2xxx qlport_down_retry=60
```

For Emulex

```
options lpfc lpfc_devloss_tmo=60
```

9.3 Reloading modprobe and RAM Disk (mkinitrd)

Other module options such as queue depth can be left as is. The module will need to be reloaded for the settings to take effect.

For local boot systems, unmount all SAN volumes and reload the module needed. For QLogic, run the commands below (for Emulex substitute lpfc).

```
# modprobe -r qla2xxx
# modprobe qla2xxx
```

The volumes can now be remounted.

For boot-from-SAN systems, the initial RAM disk needs to be rebuilt so that the setting will take effect on boot. This will rebuild a new initrd file, overwriting the same file at its existing location. Copying the existing one to a safe location is recommended.

```
# mkinitrd -f -v /boot/initrd-<kernel version>.img \
<kernel version>
```

Watch the output from the command and make sure that the “Adding module” line for the applicable module has the options added.

```
[root@local ~]# mkinitrd -f -v /boot/initrd $(uname -r) [SNIP]
Adding module qla2xxx with options qlport_down_retry=60 [SNIP]
```

The system will then need to be rebooted. Ensure that the Grub entry points to the correct initrd.

9.4 SuSE Linux Enterprise Server 11/10

SLES 11 loads module parameters through Grub at boot time for boot from SAN systems, and through /etc/modprobe.d files for regular system.

For non-boot from SAN systems, create/edit the file /etc/modprobe.d/qla2xxx and add the qlport_down_retry to the options line for QLogic cards. For Emulex cards, edit /etc/modprobe.d/lpfc and add the lpfc_devloss_tmo option. Below is an example for the QLogic card.

```
options qla2xxx qlport_down_retry=60
```

Or for Emulex

```
options lpfc_devloss_tmo=60
```

For boot from SAN systems using QLogic, append the following to the kernel line in /boot/grub/menu.lst for each desired kernel.

```
qla2xxx.qlport_down_retry=60
```

Or for Emulex:

```
lpfc.lpfc_devloss_tmo=60
```

An example entry would then look like this:

```
title SUSE Linux Enterprise Server 10 SP2 root (hd0,1)
kernel /boot/vmlinuz-2.6.16.60.21-smp root=LABEL=sysRoot\ vga=0x317 splash=silent
showopts \ qla2xxx.ql2xmaxqdepth=64 qla2xxx.qlport_down_retry=60
initrd /boot/initrd-2.6.16.60-0.21-smp
```

9.5 QLogic Proprietary Driver

If using the proprietary driver from QLogic, the option can be set by the qlinstall scrip t in the install package from QLogic. The command below will set the option on both Red Hat and SUSE systems.

```
# ./qlinstall -o qlport_down_retry=60
```

This will rebuild the initial RAM disk as well. Local boot systems can unload and reload the qla2xxx module immediately. Boot from SAN systems will have to be rebooted for the setting to take effect.

9.6 Verifying Parameter

To verify that the parameter has taken effect, run the appropriate command and check that the output is 60.

For QLogic:

```
# cat /sys/module/qla2xxx/parameters/qlport_down_retry
60
```

For Emulex:

```
# cat /sys/class/scsi_host/host0/lpfc_devloss_tmo
60
```

If possible, failover should be tested while running I/O to ensure that the configuration is correct and functional.

9.7 Disk Time Out

By default, the disk time out is set to 60 seconds. This value should not require changing. However, the setting should be verified.

```
# cat /sys/block/sdc/device/timeout
60
```

Do this for the correct Dell Compellent block device. If the value returned is not 60, consult the documentation for the specific distribution in use.

9.8 Queue Depth Settings

Queue depth for Fibre Channel HBAs is set in two places. First in the HBA BIOS for the card. This value can be modified at boot time, or using the tools provided by the HBA manufacturer. Second, it is controlled in the module for the card at the OS level. If these two numbers differ, the lower of the two numbers takes precedence.

To change the value on the HBA, consult the documentation for the HBA. To configure the modules, follow the documentation below.

For the QLogic cards being controlled by the qla2xxx module, the parameter that needs to be set is ql2xmaxqdepth. By default it is set to 32. For Emulex cards there are two parameters, lpfc_lun_queue_depth and lpfc_hba_queue_depth.

These values are set using the same procedure as the timeout configuration above. For example, on a Red Hat system using a QLogic card, the file /etc/modprobe.conf would be edited to contain a line like the following:

```
options qla2xxx qlport_down_retry=60 ql2xmaxqdepth=128
```

Follow the specific instructions for setting the module parameters that correspond to the system being configured.

10 Multipath Configuration

Though the default multipath configuration will generally appear to work and provide path failure, key values must be set in order for the system to survive a Storage Center controller failover. It is recommended to configure a volume as a multipath device whenever possible. Even in situations where a single path is used initially, configuring the volume as a multipath device brings with it some advantages. Multipath device names are persistent across reboots, device name aliases are supported to help with correlating volumes/LUNs to business function and is useful when used with advanced Storage Center features such as Live Volume and availability with controller failover/upgrades.

When a Storage Center controller fails, the system will lose connectivity with the storage for a period of time. During this time, it will often fail all paths. The default configuration is that once all paths are failed to immediately fail the disk. This results in the filesystem going read-only. By telling the system to wait before failing the disk, it can resume traffic as soon as one or more of the paths have been restored.

Starting with Red Hat Enterprise Linux 5.4, the Dell Compellent device definition is already in the default table. Therefore, it is not necessary to add the below device definition to the multipath configuration file.

10.1 Pre-configuration

Set all HBA settings per spec for the given card and Storage Center version. DO NOT follow the generic Linux timeout value documentation.

10.2 Dell Compellent Device Definition

For pre-RHEL 5.4 releases, to properly configure multipath settings for the Dell Compellent SAN, add the following devices section to the /etc/multipath.conf file:

```
 devices {
          device {
                  vendor                      COMPELNT
                  product                     "Compellent Vol"
                  path_checker                tur
                  no_path_retry               queue
         }
}
```

10.3 Fibre Channel with iSCSI Multipathing

When using the hardware or software iSCSI as a backup path for faster Fibre Channel, the configuration needs to be changed from round-robin (multibus) to active/passive (failover). This configuration insures that the backup (iSCSI) path is only used when the primary (Fibre

Channel) path fails. If left at multibus then multipath will attempt to round-robin between the faster Fibre Channel and slower iSCSI paths.

In that case, the Compellent device section needs to be changed in /etc/multipath.conf so that the “path_grouping_policy” is “failover”.

```
devices {
        device {
               vendor                      COMPELNT
               product                     "Compellent Vol"
               path_grouping_policy        failover
               path_checker                tur
               no_path_retry               queue
       }

               }
```

Multipath will automatically select the SCSI host adaptor with the lowest ID number to be the active path. By design, the HBAs will always get lower ID numbers than the software initiators, therefore there is no need to configure anything special to set the priorities.

10.4 PortDown Timeout

When running in a multipath configuration, it is desirable to have the system fail faulty links quickly. By default, the system will fail the link after 30 seconds. This means that if a cable is unplugged, I/O will be halted for 30 seconds before the link is failed. Instead, the port down timeout should be reduced to between 1 and 5 seconds.

For QLogic cards, this is achieved by setting the qlport_down_retry parameter for the qla2xxx module. If using the supplied qla2xxx module, consult the documentation for the specific distribution. If using the QLogic source version, use the included script to configure the parameter.
For the Emulex LightPulse cards the setting is lpfc_devloss_tmo for the lpfc module. For example, with Red Hat based systems using the QLogic card, adding the following
line to /etc/modprobe.conf would set the timeout to five seconds:

```
options qla2xxx qlport_down_retry=5
```

> Note
>
> Note that with boot-from-SAN systems, rebuilding the initial RAM disk and rebooting may be required depending on the system.

The instructions for modifying the timeout for single path systems shown earlier in this document can be referenced for more detailed instructions substituting.

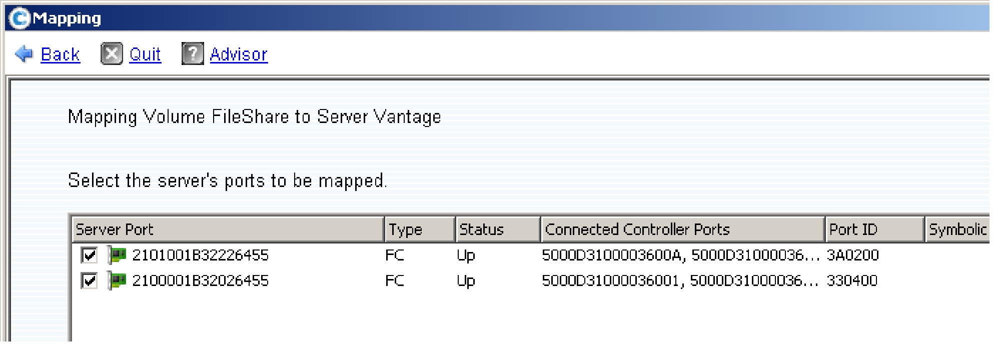
10.5 Multipathing a Volume

It is recommended that multipath be used whenever possible. Even if a volume is to be initially mapped with a single path, the advantages to configuring the path using the multipath subsystem is recommended as a best practice.

The first step in configuring multipath for a volume is to create the necessary mappings. While a volume can be configured as multipath with only one path, obviously in order to achieve the benefits, it is necessary to have at least two paths.

In this example, the server has two Fibre Channel ports and the Dell Compellent has two front end ports on each controller. They are zoned in two separate “VSANs” to create a dual switch fabric.

After selecting the server to map the volume too, the wizard will prompt for which ports on the server to map to.



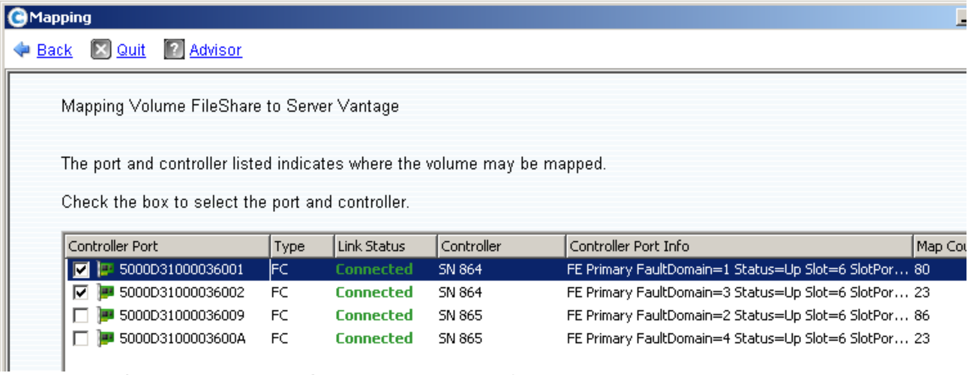
Mapping Volume FileShare to Server Vantage

Select the server's ports to be mapped.

Server Port	Type	Status	Connected Controller Ports	Port ID	Symbolic
☑ 2101001B32226455	FC	Up	5000D3100003600A, 5000D31000036...	3A0200	
☑ 2100001B32026455	FC	Up	5000D31000036001, 5000D31000036...	330400	

Figure 2: Select the Server Ports' identifying the Server Object

The next screen selects the front end ports on the Dell Compellent to use for the mapping. Select both ports from one controller.



Mapping Volume FileShare to Server Vantage

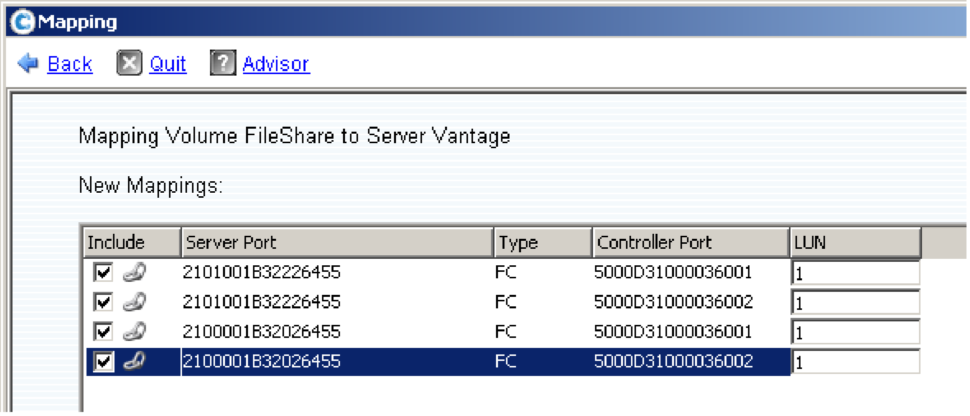
The port and controller listed indicates where the volume may be mapped.

Check the box to select the port and controller.

Controller Port	Type	Link Status	Controller	Controller Port Info	Map Cou
☑ 5000D31000036001	FC	Connected	SN 864	FE Primary FaultDomain=1 Status=Up Slot=6 SlotPor...	80
☑ 5000D31000036002	FC	Connected	SN 864	FE Primary FaultDomain=3 Status=Up Slot=6 SlotPor...	23
□ 5000D31000036009	FC	Connected	SN 865	FE Primary FaultDomain=2 Status=Up Slot=6 SlotPor...	86
□ 5000D3100003600A	FC	Connected	SN 865	FE Primary FaultDomain=4 Status=Up Slot=6 SlotPor...	23

Figure 3: Select the Dell Compellent FEPs' to use for this Volume Mapping

Finally, assign the LUN. In this case, Dell Compellent presents four pairs of mappings. In reality there are only two valid paths, however the Dell Compellent system cannot determine which ones are valid. It is therefore best to create all the mappings unless the correct WWN pairings are known.



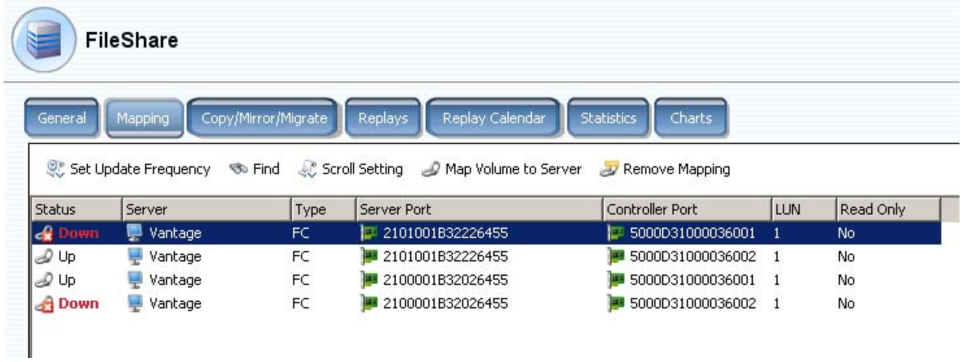
Mapping Volume FileShare to Server Vantage

New Mappings:

Include	Server Port	Type	Controller Port	LUN
☑	2101001B32226455	FC	5000D31000036001	1
☑	2101001B32226455	FC	5000D31000036002	1
☑	2100001B32026455	FC	5000D31000036001	1
☑	2100001B32026455	FC	5000D31000036002	1

Figure 4: Select the Server Ports'

There will now be two up and two down mappings to the server. The down ones can be deleted for clarity.



Status	Server	Type	Server Port	Controller Port	LUN	Read Only
Down	Vantage	FC	2101001B32226455	5000D31000036001	1	No
Up	Vantage	FC	2101001B32226455	5000D31000036002	1	No
Up	Vantage	FC	2100001B32026455	5000D31000036001	1	No
Down	Vantage	FC	2100001B32026455	5000D31000036002	1	No

Figure 5: Observe the port Status

Next, rescan the HBAs on the server to detect the new volume.

```
[root@vantage ~]# echo "- - -" >> /sys/class/scsi_host/host0/scan
[root@vantage ~]# echo "- - -" >> /sys/class/scsi_host/host1/scan
```

The new paths to the disk should have been discovered. The output from ‘lsscsi’ verifies this.

```
[root@vantage ~]# lsscsi
[0:0:0:0]    disk    COMPELN  Compellent  Vol    0402  -
[0:0:1:0]    disk    COMPELN  Compellent  Vol    0402  -
[0:0:2:0]    disk    COMPELN  Compellent  Vol    0401  /dev/sda
[0:0:3:0]    disk    COMPELN  Compellent  Vol    0402  -
[0:0:3:1]    disk    COMPELN  Compellent  Vol    0402  /dev/sdc
[1:0:0:0]    disk    COMPELN  Compellent  Vol    0402  -
[1:0:1:0]    disk    COMPELN  Compellent  Vol    0402  -
[1:0:1:1]    disk    COMPELN  Compellent  Vol    0402  /dev/sdd
[1:0:2:0]    disk    COMPELN  Compellent  Vol    0402  -
[1:0:3:0]    disk    COMPELN  Compellent  Vol    0401  /dev/sdb
                     T
```

This output shows that /dev/sdc and /dev/sdd are both LUN 1, which was the LUN used for mapping the volume. In order to add the blacklist exception, collect the WWID from the volume. It will be the same on all paths, so it can be a good sanity check.

```
[root@vantage ~]# scsi_id -g -u -s /block/sdc
36000d310000360000000000000075c8 [root@vantage ~]# scsi_id -g
-u -s /block/sdd
36000d310000360000000000000075c8
```

Add the WWID to the “blacklist_exception” sections of /etc/multipath.conf.

```
blacklist_exceptions {
            wwid "36000d3100003600000000000000005564" wwid
            "36000d310000360000000000000075c8"
}
```

To test that the configuration is correct, a dry run of the multipath command shows what configuration changes would be made if the command was run.

```
[root@vantage ~]# multipath -v2 -d
create: mpath3 (36000d310000360000000000000000075c8)
COMPELNT,Compellent Vol
[size=500G][features=0][hwhandler=0][n/a]
\_ round-robin 0 [prio=2][undef]
 \_ 0:0:3:1 sdc 8:32                [undef][ready]
 \_ 1:0:1:1 sdd 8:48                [undef][ready]
```

This shows that a multipath device, mpath3, would be created from sdc and sdd, which is what is expected. Run the command again without "-d" to create the new device. Remember that a name for the device can be supplied (instead of the automatically generated 'mpath3') using Aliases, see the section below.

```
[root@vantage ~]# multipath -v2
create: mpath3 (36000d310000360000000000000000075c8)
COMPELNT,Compellent Vol [size=500G][features=0][hwhandler=0][n/a]
\_ round-robin 0 [prio=2][undef]
 \_ 0:0:3:1 sdc 8:32                [undef][ready]
 \_ 1:0:1:1 sdd 8:48                [undef][ready]
```

The new device is ready to be formatted.

```
[root@vantage ~]# mkfs.ext3 /dev/mapper/mpath3 mke2fs 1.39 (29-
May-2006)
Filesystem label= OS type: Linux
Block size=4096 (log=2) Fragment size=4096
(log=2)
65536000 inodes, 131072000 blocks
6553600 blocks (5.00%) reserved for the super user
[TRUNCATE]
```

The multipath device is now ready to be used.

```
[root@vantage ~]# mount /dev/mapper/mpath3 /share/
```

10.6 Multipath Aliases

The multipath utility will automatically generate a new name for the multipath device. Unlike the sdX names assigned to drives, these names are persistent over reboots and/or reconfiguration. This means that they are safe to use in fstab, mount commands, and scripts. Additionally, an alias can be defined which renames the device to a user defined string. This is very useful for naming volumes to align with business function/usage. It is recommended to use multipath aliases whenever possible.

To assign an alias to a volume, first find the WWID of the volume by running the following command against one of the devices representing that volume.

```
[root@local ~]# scsi_id -g -u -s /block/sdc
360000d310000360000000000000000837
```

Note that the drive is referenced by /block/sdc not /dev/sdc.

Next, add the following section to the /etc/multipath.conf file using the WWID from the above command.

```
multipaths {
        multipath {
                wwid                        “36000d310000360000000000000000837”
                alias                       “volName”
        }
```

This defines the volume to be named “volName” instead of an assigned mpathX. The new multipath definition will be created at /dev/mapper/volName.

To define multiple multipath aliases, place each one inside of its own multipath { ...} block inside the single multipaths { ... } block.

If the generic multipath definition has already been created, unmount the volume. Then reload the multipath service with the command:

```
{root@orpheus} {~} # /etc/init.d/multipathd reload
     Reloading multipathd:                                    [ OK ]
```

Ths will recreate the definition with the new assigned alias.

The path /dev/mapper/volName can be reference anywhere a path to the device is needed.

11 Expanding a Linux Volume

Attempting to grow a file system that is on a logical or primary partition IS NOT recommended for Linux users. Expanding a file system requires advanced knowledge of the Linux system and should only be done after careful planning and consideration, including making sure valid backups exists of the file system prior to performing any volume expansion steps.

Expanding a file system that resides directly on a physical disk, however, can be done.

Note

A disruption in I/O may be required. Please consult the respective Linux distributions manuals according.

Caution

As always, when modifying partitions and/or file systems, some risk of data loss is inherent. Dell Compellent recommends taking a Replay and ensuring a good backup exists of the volume prior to executing any of the following steps.

11.1 Growing an Existing File System Offline

Volume geometry cannot be updated while the filesystem is mounted on systems prior to kernel version 2.6.18-128. For systems with kernels older than this release, follow the steps below.

These steps can be used to grow a volume that has no partition table on the disk. This does require unmounting the volume, but does not require a server reboot.

1. Expand the volume on the Storage Center
2. Stop services and unmount the volume
3. If running multipath, flush the multipath definition
   ```
   # multipath -f volumeName
   ```
4. Rescan the drive geometry (for each path if multipath)
   ```
   # echo 1 >> /sys/block/sdX/device/rescan
   ```
5. *If multipath, recreate definition*
   ```
   # multipath –v2
   ```
   or
   ```
   # /etc/init.d/multipathd reload
   ```
6. *Run fsck*
   ```
   # fsck -f /dev/sdX
   ```
7. *Grow file system*
   ```
   # resize2fs [-p] /dev/sdX
   ```
8. *Mount the filesystem and resume services*

Note

Note that certain Linux distributions may have the ability to do the resize after the volume has been mounted. This can minimize the downtime, especially on larger volumes. Consult the documentation for the specific release for risks and procedures.

11.2 Growing an Existing File System Online

Starting in Red Hat 5.3 volumes can be expanded without requiring the volume to be unmounted.

1. Expand the volume on the Storage Center
2. Rescan the drive geometry (if multipath, rescan each path)

   # echo 1 >> /sys/block/sdX/device/rescan
3. For multipath volumes, the multipath geometry needs to be resized

   # multipathd -k"resize map *multipath_device*"
4. Grow the filesystem

   # resize2fs [-p] /dev/path

12 Volumes over 2TB

Linux will discover volumes larger than 1PB, but there are limitations to the filesystems and partitions that can be created. The various Linux filesystems (ext3, ext4, xfs, zfs, btfs etc.) have specifications which vary over time, please consult the appropriate Linux distribution documentation to determine the thresholds and limitations of each filesystem type. On x86-64bit machines, the largest ext3 filesystem that is supported is just under 8TB. However, MBR partition tables (the most common and default for most Linux distributions) can only support partitions of just under 2 TB.

The easiest way around this limitation is to not use a partition table. For data disks, no partition table is required; the entire disk can simply be formatted with the file system of choice and mount the drive. This is accomplished by simply running mkfs on the device without a partition.

The alternative is to use a GPT partition table as opposed to the traditional MBR system. GPT support is native in RHEL 6/5, SLES 11/10, and many other modern Linux distributions.

12.1 Creating a GPT Partition

After the volume has been created and mapped, rescan for the new device. Then follow the example below to create a GPT partition on the device. In this case, the volume is 5TB in size and is represented by /dev/sdb.

Invoke the parted command

```
# parted /dev/sdb
```

Run the following two commands inside of parted replacing 5000G with the volume size needed

```
> mklabel gpt
> mkpart primary 0 5000G
```

Finally format and label the new partition

```
# mkfs.ext3 -L VolumeName /dev/sdb1
```

13 ext4 File System

Beginning with kernel version 2.6.18-110.el5, Red Hat introduced support for the EXT4 file system. The EXT4 file system is a modern, journaled, more scalable successor to the tried-and-true EXT3 file system. The Dell Compellent Storage Center SAN supports the use of the EXT4 file system with RHEL 6/5.

In order to use the EXT4 file system a couple of add-on packages need to be installed first. These can be found on the RHEL 6/5 DVD or downloaded from the Red Hat Network support portal. These are the ***e4fsprogs*** and ***e4fsprobs-libs*** packages.

```
{root@dean} {/mnt/loop/Server} # rpm -Uvh e4fsprogs-1.41.12-2.el5.x86_64.rpm
e4fsprogs-libs-1.41.12-2.el5.x86_64.rpm
warning: e4fsprogs-1.41.12-2.el5.x86_64.rpm: Header V3 DSA signature: NOKEY, key
ID 37017186
Preparing...                ########################################### [100%]
1:e4fsprogs-libs          ########################################### [ 50%]
2:e4fsprogs               ########################################### [100%]
```

Once installed, the EXT4 file system can be used to format volumes presented to the RHEL 6/5 host from Storage Center. The example below illustrates the process to create a volume, map it to the RHEL 6/5 host and format it using the EXT4 file system. This example uses a 100GB volume, mapped to the RHEL5 host in a multipathed environment.

First, create the volume using the Storage Center GUI, then map it to the RHEL 6/5 server object. This assumes the RHEL 6/5 server object has been configured with both server HBA ports actively zoned to the Front End HBA ports of the Storage Center SAN.

Once mapped, rescan the scsi_host ports to discover the new volume and its paths. Refer to the previous sections of this document for details on this step.

```
Example: # echo "- - -" > /sys/class/scsi_host/host0/scan
```

Determine the SCSI ID using the scsi_id command (assuming sdc is one of the device paths):

```
# scsi_id -u -g -s /block/sdc
```

Add the scsi_id to the blacklist exception section of the /etc/multipath.conf file and add a multipath section to the multipaths section. Then, reload the multipathd configuration. Refer to the Multipath Configuration section above for details. The end result should be a multipath device pointing to the new volume created above:

```
{root@dean} {~} # multipath -ll testvol1
testvol1 (36000d3100000650000000000000000048) dm-4 COMPELNT,Compellent Vol
[size=100G][features=1 queue_if_no_path][hwhandler=0][rw]
\_ round-robin 0 [prio=2][active]
 \_ 0:0:1:1 sdc 8:32   [active][ready]
 \_ 1:0:1:1 sdd 8:48   [active][ready]
```

Next, format the multipath device using the newly available EXT4 file system option:

```
{root@dean} {~} # mkfs.ext4 /dev/mapper/testvol1 mke4fs
1.41.12 (17-May-2010)
<SNIP>
```

Finally, mount the device and begin using the volume as needed.

```
{root@dean} {~} # mkdir /testvol1
{root@dean} {~} # mount /dev/mapper/testvol1 /testvol1/
```

13.1 Converting an ext3 File System to ext4

It is possible to convert file systems previously formatted as EXT3 to the new EXT4 file system format. In the example below, we have a multipath device formatted as EXT3 and mounted to the /testvol2 path. We then convert that file system to EXT4. To verify the integrity of the file system afile is copied to the file system and an md5sum is generated. Then it is compared after the conversion is completed and fsck is run.

Caution

The steps performed below can result in data loss, therefore Dell Compellent recommends taking a Replay of the volumes and verifying that valid backups exist before performing any of the following steps. This process has not been tested on the root file system and is therefore not recommended.

It is important to note that it is necessary to ***unmount*** the file system and perform an ***fsck*** as part of the conversion process.

First, copy some data to the file system and perform a checksum to be compared after the conversion to ext4 and fsck is complete.

```
{root@dean} {/testvol2} # df -h .
Filesystem             Size   Used Avail Use% Mounted on
/dev/mapper/testvol2 9.9G   151M   9.2G     2% /testvol2

{root@dean} {/testvol2} # mount | grep testvol2
/dev/mapper/testvol2 on /testvol2 type ext3 (rw)

{root@dean} {/testvol2} # cp -v /root/rhel-server-5.7-x86_64-dvd.iso /testvol2/              32
`/root/rhel-server-5.7-x86_64-dvd.iso' -> `/testvol2/rhel-server-5.7-x86_64- dvd.iso'

{root@dean} {/testvol2} # ll
```

Next, perform the conversion to ext4, unmount the file system and run fsck to complete the conversion.

```
{root@dean} {/testvol2} # tune4fs -O flex_bg,uninit_bg /dev/mapper/testvol2 tune4fs
1.41.12 (17-May-2010)

Please run e4fsck on the filesystem.
{root@dean} {~} # umount /testvol2

{root@dean} {~} # e4fsck /dev/mapper/testvol2 e4fsck
1.41.12 (17-May-2010)
One or more block group descriptor checksums are invalid.     Fix<y>? yes

Group descriptor 0 checksum is invalid.    FIXED.
<SNIP>
Adding dirhash hint to filesystem.

/dev/mapper/testvol2 contains a file system with errors, check forced. Pass 1:
Checking inodes, blocks, and sizes
Pass 2: Checking directory structure Pass 3:
Checking directory connectivity Pass 4:
Checking reference counts
Pass 5: Checking group summary information

/dev/mapper/testvol2: ***** FILE SYSTEM WAS MODIFIED *****
/dev/mapper/testvol2: 13/1310720 files (0.0% non-contiguous), 1016829/2621440 blocks
```

Finally, mount the device as an ext4 file system and confirm the integrity of the test file.

```
{root@dean} {~} # mount -t ext4 /dev/mapper/testvol2 /testvol2

{root@dean} {~} # cd /testvol2

{root@dean} {/testvol2} # df -h .
Filesystem              Size    Used Avail Use% Mounted on
/dev/mapper/testvol2        9.9G    3.8G    5.7G 40% /testvol2

{root@dean} {/testvol2} # mount | grep testvol2
/dev/mapper/testvol2 on /testvol2 type ext4 (rw)

{root@dean} {/testvol2} # md5sum rhel-server-5.7-x86_64-dvd.iso
1a3c5959e34612e91f4a1840b997b287          rhel-server-5.7-x86_64-dvd.iso

{root@dean} {/testvol2} # cat rhel-server-5.7-x86_64-dvd.md5sum
1a3c5959e34612e91f4a1840b997b287          rhel-server-5.7-x86_64-dvd.iso
```

Update the /etc/fstab file and test by rebooting the system with the ext4 file system type used in the /etc/fstab file to verify the file system mounts without error.

```
{root@dean} {/testvol2}    # cat /etc/fstab
/dev/mapper/mpath0p3       /                   ext3     defaults          1 1
/dev/mapper/mpath0p1       /boot               ext3     defaults          1 2
tmpfs                      /dev/shm            tmpfs    defaults          0 0
devpts                     /dev/pts            devpts   gid=5,mode=620    0 0
sysfs                      /sys                sysfs    defaults          0 0
proc                       /proc               proc     defaults          0 0
/dev/mapper/mpath0p2       swap                swap     defaults          0 0

# Testing
/dev/mapper/testvol2                /testvol2 ext4              defaults0 0
```

Reboot the RHEL 6/5 system.

```
{root@dean} {/testvol2} # shutdown -r now

Broadcast message from root (pts/1) (Fri Oct 28 16:01:04 2011): The

system is going down for reboot NOW!

{root@dean} {~} # df
Filesystem             Size    Used Avail Use% Mounted on
/dev/mapper/mpath0p3   31G      5.9G  24G  21% /
/dev/mapper/mpath0p1     99M  30M     64M          32%
/boot tmpfs            7.9G       0    7.9G     0% /dev/shm
/dev/mapper/testvol2      9.9G   3.8G   5.7G 40% /testvol2

{root@dean} {~} # mount | grep testvol2
/dev/mapper/testvol2 on /testvol2 type ext4 (rw)
```

13.2 ext4 and SCSI UNMAP (Free Space Recovery)

The ext4 file system in RHEL 6 supports the SCSI UNMAP commands, which allows for storage space to be “reclaimed” on a Storage Center (version 5.4 or newer) and to maintain a thinly provisioned volumes. With RHEL 6, the SCSI UNMAP calls traverse the entire storage and IO stack, through Device M multipath and then to the Fiber Channel and/or iSCSI layer. To achieve this functionality, the “discard” flag must be issued when mounting the filesystem or placed in the /etc/fstab file accordingly.

```
# mount -o discard /dev/mapper/mpathb /mnt/mpathb
```

Further documentation is available on the Dell TechCenter page below.
http://en.community.dell.com/techcenter/b/techcenter/archive/2011/06/29/native-free-space-recovery-in-red-hat-linux.aspx

Additional documentation is also available from the Redhat Customer Portal below.
https://access.redhat.com/site/solutions/393643

14 Scripting & Automation

The Dell Compellent Storage Center SAN provides a command line interface (CLI) to accomplish common system/storage administration tasks. Scripting these tasks can be a tremendous time saver to system administrators and can also be used to help keep the creation, mapping and management of volumes consistent. The CLI tool is called the Dell Compellent Command Utility (CompCU).

To use CompCU, the server must have the proper Java release installed. Refer to the Command Utility User Guide for more details. The CompCU.jar object can be downloaded from the Dell Compellent support site. Once installed on the Linux server, this tool can be used to perform Storage Center tasks from the Linux shell prompt, which can be incorporated into new or existing end-user management scripts. Below are some common use cases for using CompCU.

- Creating volumes, mapping volumes to the server.
- Taking replays, recovering replays, etc.

By no means do the examples below cover the full breadth of the usefulness of CompCU. The examples below are meant to provide an initial insight as to the sorts of tasks that can be automated with CompCU. In addition, the examples below are run on a RHEL 6/5 system that is connected to a Storage Center SAN in a multipath Fibre Channel environment. The server object is using both ports of a QLE2562 HBA that has been bound to the server object.

14.1 Using CompCU to Automate Common Tasks

The first example below is used to show the rapid deployment of several volumes and mapping them to a RHEL 6/5 system.

First, install the java package onto the RHEL 6/5 system.

```
{root@dean} {~} # rpm -Uvh jre-6u13-linux-amd64.rpm
Preparing...                ########################################### [100%]
   1:jre                    ########################################### [100%]
Unpacking JAR files...
        rt.jar...
        jsse.jar...
        charsets.jar...
        localedata.jar...
        plugin.jar...
        javaws.jar...
        deploy.jar...
```

Next, download the CompCU zip file from the Dell Compellent support site and install onto the RHEL 6/5 system by extracting the contents. Depending on the details of your RHEL 6/5 installation, the path to the Java binary may need to be updated. Update your system as appropriate. Refer to the Red Hat documentation for further details.

The "-h" switch can be used to get a help listing of available options for CompCU. Again, refer to the Dell Compellent Command Utility User Guide for further details.

```
{root@dean} {~/automation} # java -jar compcu.jar -h
Compellent Command Utility (CompCU) 5.5.1.4


      usage: java -jar CompCU.jar [Options] "<Command>"
 -c <arg>                Run a single command (option must be within quotes)
 -default                 Saves host, user, and password to encrypted file
 -defaultname <arg>      File name to save default host, user, and password encrypted file
                         to
 -file <arg>              Save output to a file
 -h                       Show help message
 -host <arg>              IP Address/Host Name of Storage Center Management
                              IP
 -password <arg>          Password of user
 -s <arg>                 Run a script of commands
 -user <arg>              User to log into Storage Center with
 -verbose                 Show debug output
 -xmloutputfile <arg>    File name to save the CompCU return code in xml format.
                          Default is cu_output.xml.

<SNIP>
```

To facilitate the ease of access for using CompCU, run CompCU with the "-default" switch to initially configure an encrypted password file. This file can then be referenced in other commands to login to the Storage Center and perform the requested actions. Below is an example of this command syntax to use:

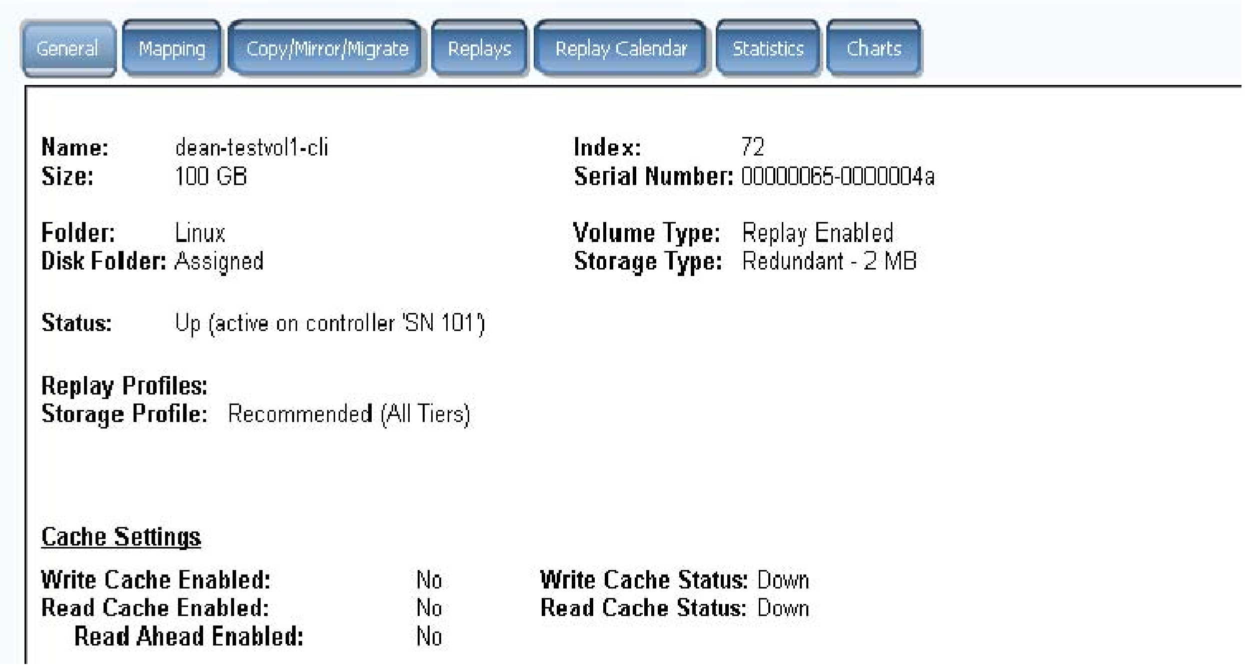
```
{root@dean} {~/automation} #java -jar CompCU.jar -default -host sc9 -user Admin
-password mmm
{root@dean} {~/automation} # java -jar compcu.jar -default -host sc9 -user
Admin -password mmm
Compellent Command Utility (CompCU) 5.5.1.4

=====================================================================
=================
User Name:                  Admin
Host/IP Address:            sc9
=====================================================================
=================
Connecting to Storage Center: sc9 with user: Admin
Saving CompCu Defaults to file [default.cli]...
```

This will create a default file called default.cli. Rename this file to match the Storage Center it refers to and reference it in future commands.

Below is an example command that will create a 100GB volume on the Storage Center SAN and map it to the RHEL 6/5 system:

```
java -jar compcu.jar -defaultname sc9_pwd_file.cli -c "volume create -folder
Linux -lun 100 -name dean-testvol1-cli -server dean-qle2562 -size 100g" Compellent
```

```
Command Utility (CompCU) 5.5.1.4

================================================================================
==================
User Name:                  Admin
Host/IP Address:            sc9
Single Command:             volume create -folder Linux -lun 100 -name dean- testvol1-cli -
server dean-qle2562 -size 100g
================================================================================
==================
Connecting to Storage Center: sc9 with user: Admin
Running Command: volume create -folder Linux -lun 100 -name dean-testvol1-cli - server dean-
qle2562 -size 100g
Creating Volume using StorageType 1: storagetype='Assigned-Redundant-4096',
redundancy=Redundant, pagesize=4096, diskfolder=Assigned.
Successfully mapped Volume 'dean-testvol1-cli' to Server 'dean-qle2562'
Successfully created Volume 'dean-testvol1-cli', mapped it to Server 'dean- qle2562' on
Controller 'SN 101'

Successfully finished running Compellent Command Utility (CompCU) application.
```



Name: dean-testvol1-cli
Size: 100 GB

Index: 72
Serial Number: 00000065-0000004a

Folder: Linux
Disk Folder: Assigned

Volume Type: Replay Enabled
Storage Type: Redundant - 2 MB

Status: Up (active on controller 'SN 101')

Replay Profiles:
Storage Profile: Recommended (All Tiers)

Cache Settings

Write Cache Enabled: No
Read Cache Enabled: No
Read Ahead Enabled: No

Write Cache Status: Down
Read Cache Status: Down

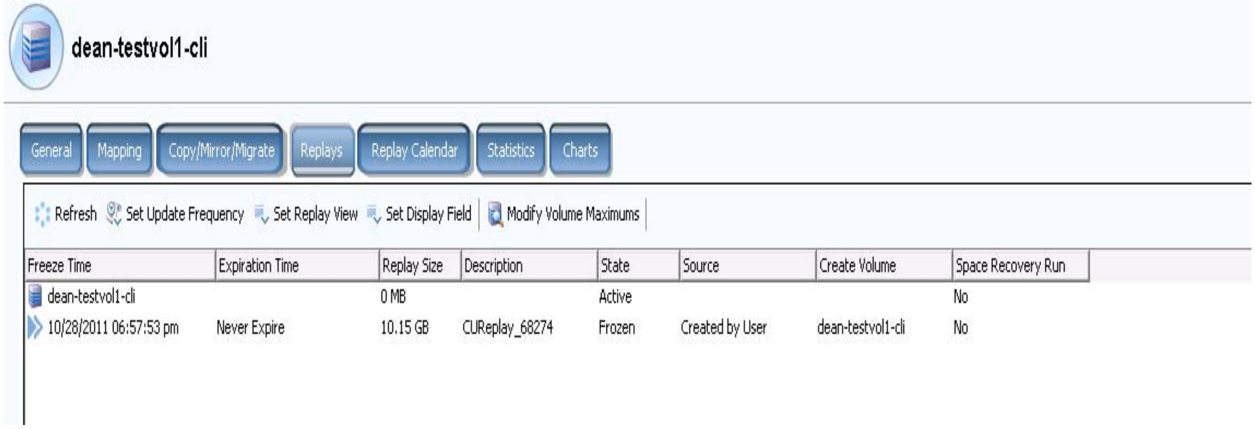
Figure 6: Volume Properties of the newly created volume

Then, rescan the scsi_host paths to discover the new volume. As this is a multipath system, setup the multipath.conf file accordingly. Refer to the Multipath Configuration section of this document for details on this process. After configuring the multipath device, a series of files are placed on the volume and a replay is taken using CompCU.

The command below is an example of how to create a replay of the volume just created above:

```
{root@dean} {~/automation} # java -jar compcu.jar -defaultname sc9_pwd_file.cli
-c "replay create -lun 200 -volume dean-testvol1-cli"
Compellent Command Utility (CompCU) 5.5.1.4
```

```
=======================================================================================
==================
User Name:                    Admin
Host/IP Address:              sc9
Single Command:               replay create -lun 200 -volume dean-testvol1-cli
=======================================================================================
==================
Connecting to Storage Center: sc9 with user: Admin
Running Command: replay create -lun 200 -volume dean-testvol1-cli Creating
replay 'CUReplay_68274' on Volume 'dean-testvol1-cli' with no expiration

Successfully finished running Compellent Command Utility (CompCU) application.
```



Freeze Time	Expiration Time	Replay Size	Description	State	Source	Create Volume	Space Recovery Run
dean-testvol1-cli		0 MB		Active			No
10/28/2011 06:57:53 pm	Never Expire	10.15 GB	CUReplay_68274	Frozen	Created by User	dean-testvol1-cli	No

Figure 7: Observe the Replays tab of the Volume Properties window

The above output shows that a Replay was created with a default description. The Replay size indicates the amount of unique block data that is frozen for Replay recovery if needed.

In the example below, a file is mistakenly removed from the source volume. A View Volume is then created and mapped to the host to recover the file. NOTE: In production it is recommended that View Volumes be mapped to a system other than the source host. This will help avoid additional complexity of realigning disk geometry, partitioning boundaries etc. when more complex volume structures are utilized (e.g. LVM, database volumes, etc).

The command below is used to see a list of the available replays for the given volume:

```
{root@dean} {~/automation} # java -jar compcu.jar -defaultname sc9_pwd_file.cli
-c "replay show -volume dean-testvol1-
cli" Compellent Command Utility
(CompCU) 5.5.1.4

=============================================================================
======
==================
User Name:                    Admin
Host/IP Address:              sc9
Single Command:               replay show -volume dean-testvol1-cli
=============================================================================
======
==================
Connecting to Storage Center: sc9 with user: Admin
Running Command: replay show -volume dean-testvol1-cli

Index                    VolumeIndex Volume
Freeze                                                  Expire
Name                                                             ConsistencyGroup
------------------ ----------- ------------------------------------------------
-- ---------------------------------------------------- -------------------------
-------------------------- ---------------------------------------------------
```

```
------- ------------------------------------------------------------
101-72-1              72            dean-testvol1-cli
10/28/2011 06:57:53 pm                                        Never
CUReplay_68274
101-72-3              72            dean-testvol1-cli
Active                                                        Never

Successfully finished running Compellent Command Utility (CompCU) application.
```

Below is a listing of the volume's files. We will "accidentally" remove testfile.5:

```
{root@dean} {/testvol1-cli} # ll testfile.*
-rw-r--r-- 1 root root 1000M Oct 28 18:55 testfile.0
-rw-r--r-- 1 root root 1000M Oct 28 18:55 testfile.1
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.2
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.3
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.4
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.5
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.6
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.7
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.8
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.9

{root@dean} {/testvol1-cli} # rm testfile.5
{root@dean} {/testvol1-cli} # ll testfile.5
/bin/ls: testfile.5: No such file or directory
```

Now, we can use CompCU to create a View Volume of the Replay we took earlier, map it to the server and recover our file. For the purposes of this example, we are mapping the View Volume back to the source server, and only mounting one of the paths to recover the file:

```
{root@dean} {~/automation} # java -jar compcu.jar -defaultname sc9_pwd_file.cli
-c "replay createview -index 101-72-1 -view dean-testvol1-cli-view -folder
Linux -server dean-qle2562 -lun 205"
Compellent Command Utility (CompCU) 5.5.1.4

=========================================================================
======
==================
User Name:                    Admin
Host/IP Address:              sc9
Single Command:               replay createview -index 101-72-1 -view dean-testvol1- cli-
view -folder Linux -server dean-qle2562 -lun 205
=========================================================================
======
==================
Connecting to Storage Center: sc9 with user: Admin
Running Command: replay createview -index 101-72-1 -view dean-testvol1-cli-view
-folder Linux -server dean-qle2562 -lun 205
Creating View Volume 'dean-testvol1-cli-view' on replay 'CUReplay_68274'
created at '10/28/2011 06:57:53 pm'…
Successfully mapped Volume 'dean-testvol1-cli-view' to Server 'dean-qle2562'

Successfully finished running Compellent Command Utility (CompCU) application.
```

```
{root@dean} {/testvol1-cli} # fdisk -l
<SNIP>
Disk /dev/sdi: 107.3 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdi doesn't contain a valid partition table

Disk /dev/sdj: 107.3 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/sdj doesn't contain a valid partition table
```

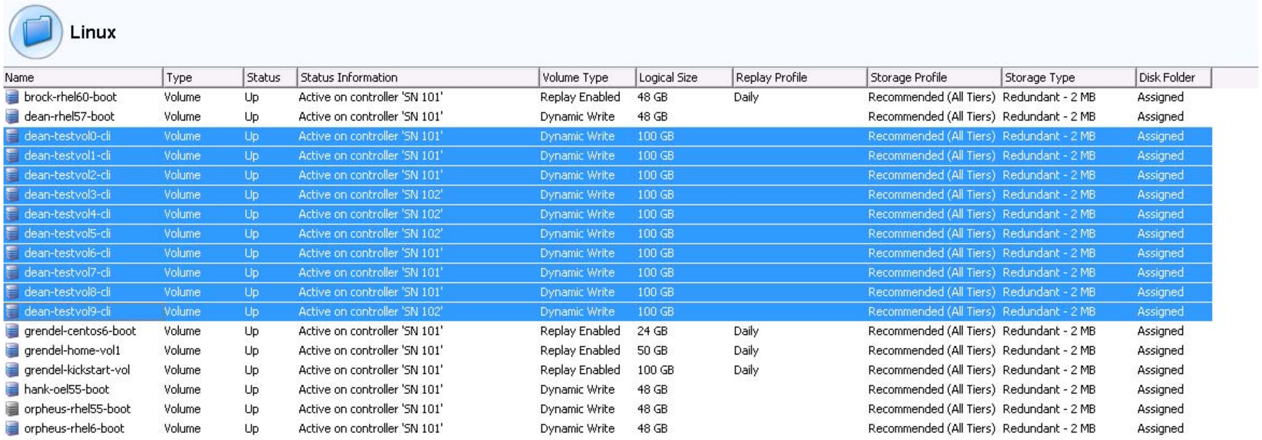
We can use the scsi_id command to verify the correct device to map by comparing the WWID with the Volume SN listed in Storage Center. Refer to the Managing Volumes section of this document for further details. Then we map up the View Volume and recover the file.

```
{root@dean} {~/automation} # mkdir /view
{root@dean} {~/automation} # mount /dev/sdi /view
{root@dean} {~/automation} # cd /view/

{root@dean} {/view} # ll
total 14G
drwx------ 2 root root           16K Oct 28 18:53 lost+found/
---------- 1 root root          3.6G Oct 28 18:54 rhel-server-5.7-x86_64-dvd.iso
-rw-r--r-- 1 root root 1000M Oct 28 18:55 testfile.0
-rw-r--r-- 1 root root 1000M Oct 28 18:55 testfile.1
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.2
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.3
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.4
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.5
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.6
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.7
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.8
-rw-r--r-- 1 root root 1000M Oct 28 18:56 testfile.9

{root@dean} {/view} # cp testfile.5 /testvol1-cli/testfile.5

{root@dean} {/view} # cd /testvol1-cli/

{root@dean} {/testvol1-cli} # ll testfile.5
-rw-r--r-- 1 root root 1000M Oct 28 19:19 testfile.5
```

All of these examples, along with numerous other scenarios, can be integrated into new or existing administration scripts to help system administrators automate common tasks. For example, below we use a simple script to create ten (10) volumes rapidly and map them to a target server:

```
{root@dean} {~/automation} # for a in 0 1 2 3 4 5 6 7 8 9; do java -jar compcu.jar -
defaultname sc9_pwd_file.cli -c "volume create -folder Linux -lun
```

```
10${a} -name dean-testvol${a}-cli -server dean-qle2562 -size 100g"; done

Compellent Command Utility (CompCU) 5.5.1.4

========================================================================
======
==================
User Name:                    Admin
Host/IP Address:              sc9
Single Command:               volume create -folder Linux -lun 100 -name dean-
testvol0-cli -server dean-qle2562 -size 100g
========================================================================
======
==================
Connecting to Storage Center: sc9 with user: Admin
Running Command: volume create -folder Linux -lun 100 -name dean-testvol0-cli - server
dean-qle2562 -size 100g
Creating Volume using StorageType 1: storagetype='Assigned-Redundant-4096',
redundancy=Redundant, pagesize=4096, diskfolder=Assigned.
Successfully mapped Volume 'dean-testvol0-cli' to Server 'dean-qle2562' Successfully
created Volume 'dean-testvol0-cli', mapped it to Server 'dean- qle2562' on Controller
'SN 101'

Successfully finished running Compellent Command Utility (CompCU) application.

<SNIP>

Compellent Command Utility (CompCU) 5.5.1.4

========================================================================
======
==================
User Name:                    Admin
Host/IP Address:              sc9
Single Command:               volume create -folder Linux -lun 109 -name
dean- testvol9-cli -server dean-qle2562 -size 100g
========================================================================
======
==================
Connecting to Storage Center: sc9 with user: Admin
Running Command: volume create -folder Linux -lun 109 -name dean-testvol9-cli
- server dean-qle2562 -size 100g
Creating Volume using StorageType 1: storagetype='Assigned-Redundant-
4096', redundancy=Redundant, pagesize=4096, diskfolder=Assigned.
Successfully mapped Volume 'dean-testvol9-cli' to Server 'dean-qle2562'
Successfully created Volume 'dean-testvol9-cli', mapped it to Server 'dean-
qle2562' on Controller 'SN 102'

Successfully finished running Compellent Command Utility (CompCU) application.
```



Linux

Name	Type	Status	Status Information	Volume Type	Logical Size	Replay Profile	Storage Profile	Storage Type	Disk Folder
brock-rhel60-boot	Volume	Up	Active on controller 'SN 101'	Replay Enabled	48 GB	Daily	Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-rhel57-boot	Volume	Up	Active on controller 'SN 101'	Dynamic Write	48 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol0-cli	Volume	Up	Active on controller 'SN 101'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol1-cli	Volume	Up	Active on controller 'SN 101'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol2-cli	Volume	Up	Active on controller 'SN 101'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol3-cli	Volume	Up	Active on controller 'SN 102'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol4-cli	Volume	Up	Active on controller 'SN 102'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol5-cli	Volume	Up	Active on controller 'SN 102'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol6-cli	Volume	Up	Active on controller 'SN 101'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol7-cli	Volume	Up	Active on controller 'SN 101'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol8-cli	Volume	Up	Active on controller 'SN 101'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
dean-testvol9-cli	Volume	Up	Active on controller 'SN 102'	Dynamic Write	100 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
grendel-centos6-boot	Volume	Up	Active on controller 'SN 101'	Replay Enabled	24 GB	Daily	Recommended (All Tiers)	Redundant - 2 MB	Assigned
grendel-home-vol1	Volume	Up	Active on controller 'SN 101'	Replay Enabled	50 GB	Daily	Recommended (All Tiers)	Redundant - 2 MB	Assigned
grendel-kickstart-vol	Volume	Up	Active on controller 'SN 101'	Replay Enabled	100 GB	Daily	Recommended (All Tiers)	Redundant - 2 MB	Assigned
hank-oel55-boot	Volume	Up	Active on controller 'SN 101'	Dynamic Write	48 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
orpheus-rhel55-boot	Volume	Up	Active on controller 'SN 101'	Dynamic Write	48 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned
orpheus-rhel6-boot	Volume	Up	Active on controller 'SN 101'	Dynamic Write	48 GB		Recommended (All Tiers)	Redundant - 2 MB	Assigned

Figure 8: 10 new volumes created

```
{root@dean} {/sys/class/scsi_host} # lsscsi
[0:0:0:0]    disk    COMPELNT Compellent Vol   0504 /dev/sda
[0:0:0:100]    disk      COMPELNT  Compellent  Vol     0504
/dev/sdh  [0:0:0:101]    disk      COMPELNT  Compellent  Vol
0504 /dev/sdg [0:0:0:102]  disk    COMPELNT Compellent Vol
0504 /dev/sdf [0:0:0:106]  disk    COMPELNT Compellent Vol
0504 /dev/sde [0:0:0:107]  disk    COMPELNT Compellent Vol
0504 /dev/sdd [0:0:0:108]  disk    COMPELNT Compellent Vol
0504 /dev/sdc [0:0:1:103]  disk    COMPELNT Compellent Vol
0504 /dev/sdl [0:0:1:104]  disk    COMPELNT Compellent Vol
0504 /dev/sdk [0:0:1:105]  disk    COMPELNT Compellent Vol
0504 /dev/sdj [0:0:1:109]  disk    COMPELNT Compellent Vol
0504 /dev/sdi [1:0:0:0]    disk    COMPELNT Compellent Vol
0504 /dev/sdb [1:0:0:100]  disk    COMPELNT Compellent Vol
0504 /dev/sdr [1:0:0:101]  disk    COMPELNT Compellent Vol
0504 /dev/sdq [1:0:0:102]  disk    COMPELNT Compellent Vol
0504 /dev/sdp [1:0:0:106]  disk    COMPELNT Compellent Vol
0504 /dev/sdo [1:0:0:107]  disk    COMPELNT Compellent Vol
0504 /dev/sdn [1:0:0:108]  disk    COMPELNT Compellent Vol
0504 /dev/sdm [1:0:1:103]  disk   COMPELNT Compellent Vol
0504 /dev/sdv [1:0:1:104]  disk    COMPELNT Compellent Vol
0504 /dev/sdu [1:0:1:105]  disk    COMPELNT Compellent Vol
0504 /dev/sdt [1:0:1:109]  disk    COMPELNT Compellent Vol
0504  /dev/sds  [2:0:0:0]     cd/dvd   PLDS       DVD-ROM  DS-
8D3SH HD51 /dev/sr0
```

The above example shows an example in which ten (10) volumes are quickly created from the Linux shell prompt using shell scripting and CompCU.

15 Performance Tuning Considerations

This section provides some general information and guidance pertaining to some of the more common performance tuning options and variables available to Linux, particularly with Red Hat Enterprise Linux 5.x and 6.x installations. This information is not intended to be all encompassing and the values used should not be considered "hard and fast". Rather, the intent of this section is to provide a starting point from which Linux and/or storage administrators' can fine tune their Linux installation to achieve optimal performance.

Prior to making any changes to the following parameters, a good understanding of the environment's current workload should be established. There are numerous methods by which this could be accomplished, in addition to the system and/or storage administrators' perception based on day-to-day experience with supporting the environment. One such tool is the Dell Performance Analysis Collection Kit (DPACK) which is a free download from the following URL:

> http://search.dell.com/results.aspx?s=gen&c=us&l=en&cs=&k=dpack&cat=sup&x=0&y=0

There are some general "rules of thumb" to keep in mind when it comes to performance tuning with
Linux:

1. Performance tuning is as much an art as a science. As there are a number of variables that impact performance (I/O in particular), there are no specific values that can be recommended for every environment. It is best to begin with as few variables as possible and then add more layers as one tunes the system. For example, start with single path, tune and then add multipath.
2. Make one change at a time and test the affect on performance with a performance monitoring tool before making subsequent changes.
3. It is considered a best practice to make sure all original settings are recorded so that changes can be reverted to a known "stable" state.
4. As with other system tuning (e.g. failover), changes should always be made in non-production installations first, and validated with as many environmental conditions as possible before inserting changes into production environments.

There is also another "rule of thumb" that is worth mentioning: if performance needs are being met with the current configuration settings, it is generally a best practice to leave the settings alone to avoid introducing changes that can make the system less stable.

In addition, an understanding of the differences between block and file level data should be established in order to be able to effectively target the tunable(s) that can most effectively impact performance in a positive manner. Although the Dell Compellent Storage Center array is a block-based storage device, the support for the iSCSI transport mechanism introduces performance considerations that are typically associated with network and file level tuning.

When validating whether a change is having an impact on performance, leverage the Charting feature of the Dell Compellent Enterprise Manager to track the performance. In addition, be sure to make singular changes between iterations in order to better track what variables have the most affect (positive or negative) on I/O performance.

15.1 Take Advantage of Multiple Volumes

A volume can be active on only one Storage Center controller at a time. Therefore, when possible, spread volumes evenly across both SAN controllers to most effectively leverage dual I/O processing. A larger number of smaller-sized volumes will, often result in better performance than fewer larger-sized volumes. From a Linux perspective, having multiple target LUNs can result in performance improvements by leveraging the kernel's ability to process I/O in parallel by addressing multiple paths, SCSI devices, etc.

15.2 Host Bus Adapter Queue Depth

Queue depth refers to the I/O "in flight" at any given time. Modifying this value can lead to an improvement in I/O performance in some workloads. Generally, increasing queue depth can increase throughput, but caution should be taken as increasing this value can lead to higher latency. Different applications may benefit from increasing this value, such as environments in which the bulk of I/O is small reads/writes. In environments defined by lower IOPS requirements but needing higher throughput, this may be achieved by lowering this queue depth setting till optimal levels of performance is achieved.

This value can be changed in the HBA's firmware as well as from within the Linux kernel module for the HBA. Keep in mind that if the two settings have different values, the lower value takes precedence. Therefore, one good strategy to consider, given this behavior, would be to set the HBA's firmware setting to a high number, and then tune the value downward from within the Linux kernel module.

Refer to the Server Configuration section of this document for details on modifying this value for the particular HBA model being used.

15.3 Linux SCSI Device Queue Tunable Values

There are several Linux SCSI device settings that can be tuned to affect performance. The most common are listed below, with a brief explanation of what the parameter relates to with regard to I/O. These values are all found within the directory path "/sys/block/<device>/queue", and should be modified for each device comprising the volume being targeted for performance modification.

15.3.1 Kernel I/O Scheduler

This parameter "/sys/block/<device>/queue/schedule" and its contents sets the I/O scheduler in use by the Linux kernel for the SCSI (sd) device. Some application vendors (e.g. Oracle) provide specific recommendations for the I/O scheduler to activate to achieve optimal performance. By default on RHEL 6/5 this is set to "cfq" as denoted by the [ ] parenthesis within the file. This parameter can be dynamically changed by performing the following:

```
{root@orpheus} {~} # cat /sys/block/sda/queue/scheduler noop
anticipatory deadline [cfq]

{root@orpheus} {~} # echo deadline > /sys/block/sda/queue/scheduler

{root@orpheus} {~} # cat /sys/block/sda/queue/scheduler noop
anticipatory [deadline] cfq
```

The above command has changed the I/O scheduler for SCSI device "sda" to use the *"deadline"* option instead of "cfq". This command will change for only the currently running instance. To make this change persistent across system reboots, a boot-time script could be used to make this change on a per device basis, or for system wide I/O scheduler changes, pass in the proper kernel option at boot time, such as:

```
kernel /vmlinuz-2.6.18-238.el5 ro root=/dev/sda3 quiet elevator=deadline
```

Note

In multipath and LVM configurations, this modification should be made to each device used by the device-mapper subsystem.

15.3.2 read_ahead_kb

This parameter is used to tell the Linux kernel how much to read (in kilobytes) at a time when the kernel detects it is sequentially reading from a block device. Modifying this value can have a noticeable effect on performance in heavy sequential read workloads. By default, on RHEL 6/5, this value is set to 128. Increasing this to a larger size may result in higher read throughput performance.

15.3.3 nr_requests

This value is used by the kernel to set the depth of the request queue and is often used in conjunction with changes to the Queue Depth of the HBA. With the “cfq” I/O scheduler, this is set to 128 by default. Increasing this value will set the I/O subsystem to a larger threshold to which it will continue scheduling requests. This keeps the I/O subsystem moving in one direction longer, which can result in more efficient handling of disk I/O. It is a best practice starting point to increase this value to 1024, test and adjust accordingly per the performance results achieved.

15.4 Device-Mapper-Multipath rr_min_io

When taking advantage of multipath configuration in which multiple physical paths can be leveraged to perform I/O operations to a multipathed device, the rr_min_io parameter can be modified to optimize the I/O subsystem. The rr_min_io specifies the number of I/O requests to route to a path before switching to the next path in the current path group.

By default, the rr_min_io is set to 1000. Generally, this is much too high. A general “rule of thumb” is to set this to 2 times the queue depth value and then test the performance. The goal of modifying this setting is to try and create an I/O flow that most efficiently fills up the I/O “buckets” in equal proportions as it is passes through the Linux I/O subsystem.

This value is modified by making changes to the /etc/multipath.conf file in the “Defaults” section. For example:

```
#defaults {
#           udev_dir                    /dev
#           polling_interval            10
#           selector                    "round-robin 0"
#           path_grouping_policy        multibus
#           getuid_callout              "/sbin/scsi_id -g -u -s /block/%n"
#           prio_callout                /bin/true
#           path_checker                readsector0
#           rr_min_io                   100
#           rr_weight                   priorities
#           failback                    immediate
#           no_path_retry               fail
#           user_friendly_name          yes
#}
```

15.5 iSCSI Considerations

Tuning performance for iSCSI is as much an effort in Ethernet network tuning as it is block-level tuning. Many of the common Ethernet kernel tunable parameters should be experimented with in order to determine what settings provide the highest performance gain with iSCSI. Often, just simply increasing the frame size supported to jumbo frames can lead to iSCSI performance improvements when using 1Gb/10Gb Ethernet. As with Fibre Channel, changes should be made incrementally and evaluated against multiple workload types expected in the environment in order to fully understand the effects on overall performance.

In other words, tuning performance for iSCSI is often more time consuming as one must consider the block-level subsystem tuning as well as network (Ethernet) tuning. A solid understanding of the various Linux subsystem layers involved is necessary to effectively tune the system.

Kernel parameters that can be tuned for performance are found in the "/proc/sys/net/core" and "/proc/sys/net/ipv4" kernel parameters. Once optimal values are determined, these can be permanently set in the /etc/sysctl.conf file.

Like most other modern OSes, Linux can do a good job of auto-tuning TCP buffers, however by default some of the settings are set conservatively low. Experimenting with the following kernel parameters can lead to increased network performance, which can then improve iSCSI performance.

- TCP Max Buffer Sizes:
    - net.core.rmem_max
    - net.core.wmem_max
- Linux Auto-tuning buffer limits:
    - net.ipv4.tcp_rmem
    - net.ipv4.tcp_wmem
- net.ipv4.tcp_window_scaling
- net.ipv4.tcp_timestamps
- net.ipv4.tcp_sack

16 Conclusion

The above information is intended to give system and/or storage administrators' a good starting point by which to tune and improve performance on a Linux system using Dell Compellent Storage Center volumes. There are many variables that can be tuned on a Linux host and different combinations of modifications will suit some installations better than others. Incremental changes should be employed following the pattern: change, test and repeat.

17 Additional Resources

Red Hat Enterprise Linux Document Portal
https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/?locale=en-US