

# Dell EMC SC Series with OpenStack Cinder

## Abstract

This paper presents an introduction to using Dell EMC™ SC Series storage with OpenStack® Cinder.

April 2018

## Revisions

Date	Description
April 2014	Initial release
July 2015	Dell Storage SC Cinder driver introduced
December 2015	Updated for OpenStack Liberty
July 2016	Updated for OpenStack Mitaka
October 2016	Updated for OpenStack Newton
April 2017	Updated for OpenStack Ocata
April 2018	Updated for OpenStack Pike and OpenStack Queens

## Acknowledgements

Author: Steven Lemons

The information in this publication is provided "as is." Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

© 2014-2018 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

Dell believes the information in this document is accurate as of its publication date. The information is subject to change without notice.

# Table of contents

Revisions.....	2
Acknowledgements.....	2
Executive summary.....	5
<b>1 Overview.....</b>	<b>6</b>
1.1 What is OpenStack?.....	6
1.2 Why use OpenStack?.....	7
<b>2 SC Series with OpenStack Cinder .....</b>	<b>8</b>
2.1 OpenStack Cinder setup and configuration.....	8
2.2 What is Cinder? .....	9
2.3 Connecting SC Series to OpenStack Cinder.....	10
2.3.1 Cinder direct connect to SC Series .....	10
2.3.2 Cinder connect to Dell Storage Manager .....	11
2.3.3 Multipath enforcement .....	11
2.3.4 Use with DevStack .....	12
2.3.5 Use with OpenStack .....	13
2.3.6 Use with multiple back-end definitions .....	14
2.4 Volume management and snapshots.....	15
2.5 Replication and Live Volume .....	17
2.6 Creating volumes using OpenStack Horizon.....	19
2.7 Attaching volumes with Horizon .....	20
<b>3 Ocata updates .....</b>	<b>23</b>
3.1 Live Volume with Auto Failover (AFO) .....	23
3.2 Data deduplication and compression .....	23
3.2.1 Deduplication with compression .....	24
3.2.2 Compression.....	24
3.3 Volume QoS profiles.....	24
<b>4 Newton updates.....</b>	<b>25</b>
4.1 Live Volume .....	25
<b>5 Mitaka updates .....</b>	<b>26</b>
5.1 Replication .....	26
5.1.1 Replication target.....	27
5.2 Snapshot profiles (snapshot schedules) .....	27
<b>6 Liberty updates.....</b>	<b>29</b>
6.1 Storage profiles.....	29

6.2	Managing existing volumes .....	33
6.3	Consistency groups .....	35
6.3.1	Creating a consistency group .....	35
6.3.2	Creating a volume into a consistency group .....	36
6.3.3	Creating, viewing, and removing snapshots (replays) of a consistency group .....	36
6.3.4	Displaying the consistency group .....	37
6.3.5	Removing a consistency group .....	37
7	Volume management with logical volume manager (LVM) and CLI .....	38
7.1	Managing volumes with Cinder .....	38
7.1.1	Preparing volumes.....	39
7.1.2	Presenting volumes to instances or VMs .....	40
8	SAN in the cloud.....	42
A	Configuration details.....	43
B	Additional resources .....	44
B.1	Technical support and resources .....	44
B.2	Related documentation.....	44

## Executive summary

This paper presents an introduction to using Dell EMC™ SC Series storage with OpenStack® Cinder. Because there are various methods in which to accomplish the tasks discussed, this paper is intended as a starting point for end users and system administrators.

This guide focuses on methods using both the OpenStack Horizon interface, as well as the CLI because it is the most universally applicable method across UNIX® and Linux®.

# 1 Overview

SC Series storage provides SCSI-3 compliant disk volumes that remove the complexity of allocating, administering, using, and protecting mission-critical data. A properly configured SC Series system removes the need for cumbersome physical disk configuration exercises and management along with complex RAID configuration mathematics. The SC Series additionally provides RAID 10 speed and reliability at the storage-array layer across all disk volumes.

The use of SC Series with OpenStack further elevates the functionality and scalability of SC Series storage by leveraging its strengths and capitalizing on the OpenStack framework. This combination positions SC Series storage into the software-defined storage evolution.

## 1.1 What is OpenStack?

OpenStack is an open-source project that was jointly developed by Rackspace® and NASA® in July 2010. It is a collection of software projects that when used in unison creates and provides a framework for quickly provisioning, delivering, and managing a private cloud infrastructure, and delivering Infrastructure as a Service (IaaS).

The [OpenStack Foundation](#) was founded in September 2012 to promote the development, distribution, and adoption of the OpenStack project.

OpenStack consists of seven core code modules:

- Nova: Compute service
- Neutron: Network service
- Cinder: Block store
- Glance: Image store
- Swift: Object store
- Keystone: Identity
- Horizon: Dashboard

In addition to the core modules, the OpenStack framework continues to grow with the following modules that will eventually become part of future stable releases:

- Manila: NFS
- Ceilometer: Telemetry
- Heat: Orchestration
- Ironic: Bare metal deployment
- Sahara: Apache™ Hadoop® as a service
- Magnum: Container orchestration
- Kolla: OpenStack as a service

The OpenStack framework relies on a native QEMU KVM for hypervisor functionality. However, alternative hypervisors (such as VMware® ESX® and Microsoft® Hyper-V®) can also be plugged into the OpenStack framework using the appropriate vendor-provided drivers that comply with the OpenStack API standards.

Figure 1 depicts a logical OpenStack diagram showing how the various core modules interact and function.

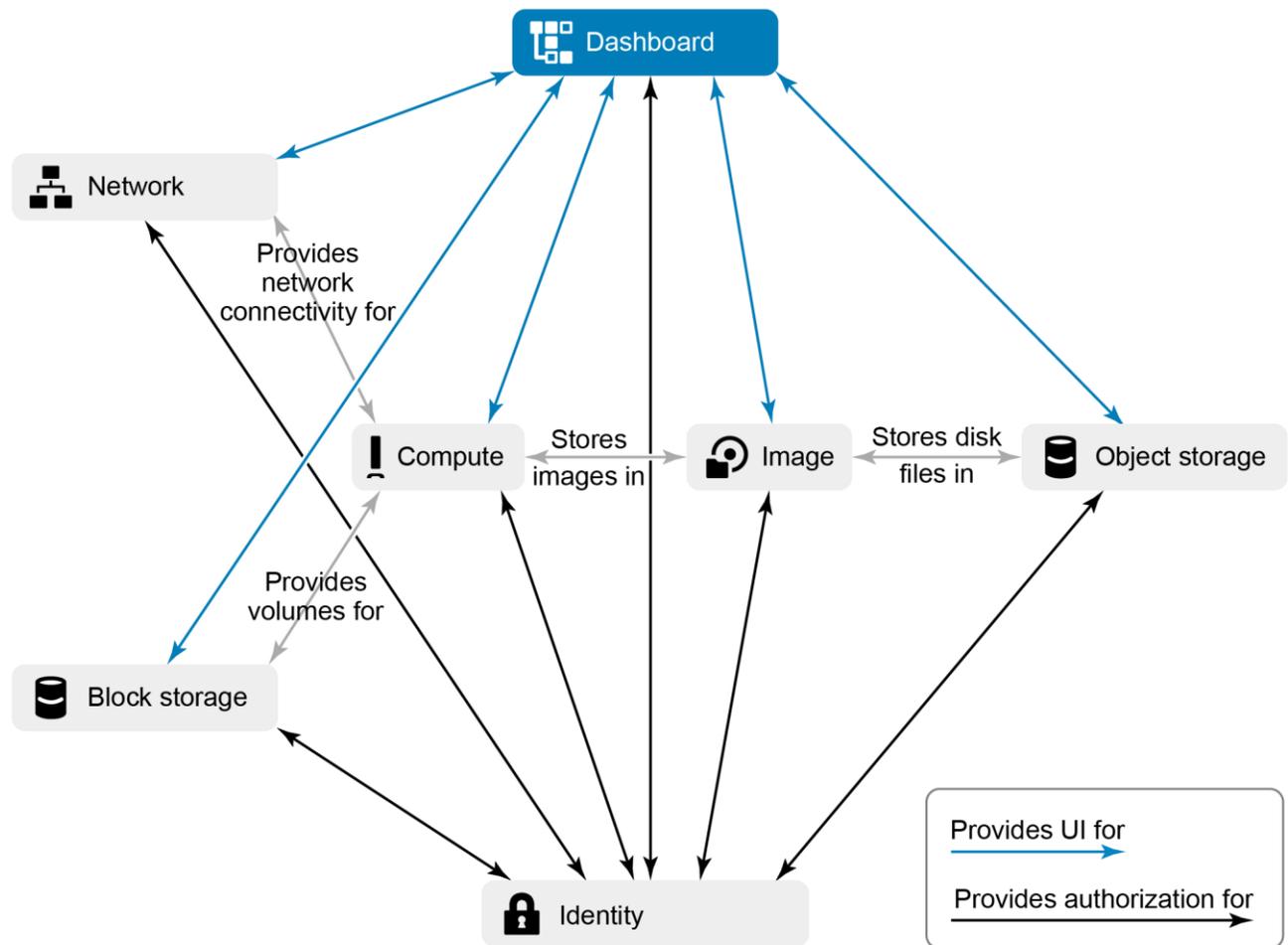


Figure 1 OpenStack logical diagram

## 1.2 Why use OpenStack?

The OpenStack framework allows an organization to leverage existing commodity hardware to build out a private cloud infrastructure. It is available at no cost under the terms of the Apache license model and is managed using open APIs, open formats, and freely accessible source code. The use of OpenStack alleviates and removes concerns of any proprietary vendor lock-in for cloud customers.

Additionally, OpenStack allows an organization to consolidate and manage heterogeneous infrastructure through a homogenous framework, platform, and interface while elevating infrastructure into becoming cloud-based, redundant, widely scalable, and self-serviceable.

Dell EMC is partnered with Red Hat® to co-engineer, deliver, and support enterprise-grade OpenStack private cloud solutions. Details about this partnership are provided here: [www.redhat.com/en/partners/strategic-alliance/dell](http://www.redhat.com/en/partners/strategic-alliance/dell).

## 2 SC Series with OpenStack Cinder

This section outlines the setup, configuration, and connectivity of SC Series storage into an OpenStack framework that presents volumes to OpenStack through the Cinder module and makes these volumes available to OpenStack instances and virtual machines.

Dell EMC storage offers a Cinder driver in conjunction with the OpenStack Kilo release or newer.

The two Cinder volume drivers to consider when configuring Cinder to work with SC Series storage arrays include the following:

### SCFCDriver

- Implements management commands for SC Series storage using Fibre Channel
- Current version: 4.1.0
- Driver name: **cinder.volume.drivers.dell\_emc.sc.storagecenter\_fc.SCFCDriver**

### SCISCSIDriver

- Implements management commands for SC Series storage using iSCSI
- Current version: 4.1.0
- Driver name: **cinder.volume.drivers.dell\_emc.sc.storagecenter\_iscsi.SCISCSIDriver**

Additional volume driver information can be found in the [OpenStack Cinder available volume drivers page](#).

---

**Note:** While the former volume\_driver name for Cinder and SC Series storage (**cinder.volume.drivers.dell.dell\_storagecenter**) will still work on new Cinder versions, it is recommended to use this new volume\_driver name for all Cinder and SC Series configuration.

---

## 2.1 OpenStack Cinder setup and configuration

The OpenStack framework used in this environment spans two Dell EMC PowerEdge™ servers; one server operates as the controller node and the other as the compute node. OpenStack (Queens release) is used and installed per the [OpenStack Installation Guide](#).

Figure 2 depicts a diagram of the test environment. To focus on validating SC Series storage with OpenStack Cinder, networking information and other redundancies have been omitted for the sake of simplicity and setup.

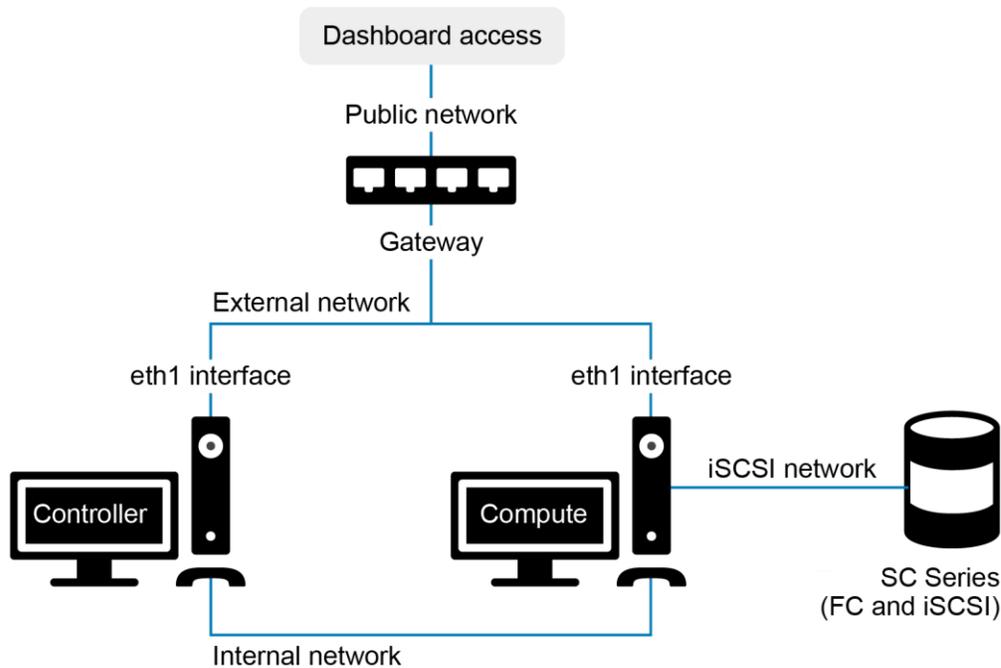


Figure 2 SC Series with OpenStack Cinder

## 2.2 What is Cinder?

Cinder is one of the core modules of the OpenStack framework. It provides block storage to the compute nodes of the OpenStack framework. Cinder block storage represents the ability to provision and provide data and volume storage facilities to the instances and virtual machines that run on the compute node.

Cinder includes multiple processes that run on both the controller and compute nodes of the OpenStack framework. These components are outlined in this section and further discussed in the OpenStack article, [Block Storage service](#).

- **cinder-api:** Installed to the controller node, it accepts API requests and routes them to cinder-volume for action.
- **cinder-volume:** Installed to the compute node, it responds to requests from cinder-api to read or write to the block storage database and maintain state awareness. This component interacts with cinder-scheduler through RabbitMQ™ (or another message-queue method) and operates on the storage itself (using the appropriate drivers).
- **cinder-scheduler:** Installed to the controller node, it routes any cinder-api requests to the most optimal node for servicing.
- **cinder-backup:** It provides an interface to allow the backup of volumes to a variety of back-end storage providers through a driver API.

## 2.3 Connecting SC Series to OpenStack Cinder

SC Series storage is best leveraged when connected with Cinder into the OpenStack framework. Physical connectivity, Fibre Channel (FC) or iSCSI, is established by connecting the SC Series array to the compute nodes of the framework and applying required fabric zoning.

The SC Series system can be configured to support either multipath FC or multipath iSCSI-based protocols, and complies with the Cinder API specifications and respective supported operations as outlined in the OpenStack article, [Dell Storage Center Fibre Channel and iSCSI drivers](#). This is the recommended method for connecting SC Series storage into the OpenStack framework.

### 2.3.1 Cinder direct connect to SC Series

With the release of SCOS version 7.1.x and higher, combined with OpenStack versions Ocata and later, Cinder can be configured to initiate block storage API calls directly against the virtual management IP of the SC Series array rather than directing those API calls through Dell Storage Manager (DSM).

---

**Note:** If replication is needed, direct connect of Cinder to SC Series is not supported and the use of DSM as Cinder block storage target is required.

---

The following screenshot shows the Cinder configuration snippet for iSCSI between Cinder API and the virtual management IP (outlined in blue) of the target SC Series array. This example uses a Dell EMC SC5020 array with a SSN of 351396 (outlined in red).

```
[delliscsi]
volume_backend_name = delliscsi
volume_driver = cinder.volume.drivers.dell_emc.sc.storagecenter_iscsi.SCISCSIDriver
san_ip = 100.80.83.200
san_login = Admin
san_password = mmmm
dell_sc_ssn = 351396
iscsi_ip_address = 10.10.10.200
iscsi_port = 3260
dell_sc_api_port = 3033
dell_sc_server_folder = OSP_Servers/delliscsi/queens
dell_sc_volume_folder = OpenStackVolumes/delliscsi/queens
image_volume_cache_enabled = True
volume_clear = zero
```

Figure 3 Cinder configuration snippet for SC Series direct connect

The following screenshot shows the virtual management IP of the SC5020 (outlined in blue) used as the target for the prior Cinder iSCSI configuration.

The screenshot shows the 'Edit Storage Center Settings' window for 'Storage Center 351396'. On the left is a navigation menu with options like General, License, Preferences, Storage, etc. The main area is titled 'Storage Center Information' and contains several fields: Name (Storage Center 351396), Serial Number (351396), Host or IP Address (100.80.83.200), Version (7.2.11.4), Model (CT-SC5020), Operation Mode (Normal), and Notes. A blue box highlights the 'Storage Center Management IP Address' section, which includes 'Management IPv4 Address' (100.80.83.200) and 'IPv6 Management IP' (::). Below this is the 'Login Banner' section with a 'Login Banner Message' field.

Figure 4 Virtual management IP of SC Series array

### 2.3.2 Cinder connect to Dell Storage Manager

Dell Storage Manager (DSM) 2015 R3 or newer is required for connecting SC Series storage with Cinder into the OpenStack framework when replication is needed.

The Dell Storage Manager platform is used to proxy OpenStack API calls to the configured SC Series arrays.

### 2.3.3 Multipath enforcement

The following statements should be included in **local.conf/cinder.conf** or **nova.conf** to ensure that multipath is used and enforced for all data transfer events. These statements must always exist in the configuration to ensure proper interoperability between the SC Series array and OpenStack.

```
# cat ~/devstack/local.conf
```

```
[snip]
```

```
# Dell SC Series
#-----
```

```
# enable multiple Cinder storage backends
CINDER_ENABLED_BACKENDS=DellStorageCenterFCDriver:dellfc,
DellStorageCenterFCDriver:dellfc_repl,DellStorageCenteriSCSIDriver:delliscsi
CINDER_DEFAULT_VOLUME_TYPE=dellfc
```

```
[[post-config|$CINDER_CONF]]
```

```
[default]
use_multipath_for_image_xfer = True
enforce_multipath_for_image_xfer = True

[snip]

[[post-config|$NOVA_CONF]]
[libvirt]
iscsi_use_multipath = True
```

### 2.3.4 Use with DevStack

When used with DevStack, the recommended configuration approach is to append the following syntax to the end of the `~/devstack/local.conf` file. Append the syntax to the file before stacking up the OpenStack environment with the `~/devstack/stack.sh` command. The following configuration contains comments in **red**.

```
# cat ~/devstack/local.conf

[snip]

# Dell SC Series
#-----

# enable multiple Cinder storage backends
CINDER_ENABLED_BACKENDS=DellStorageCenterFCDriver:dellfc,
DellStorageCenterFCDriver:dellfc_repl,DellStorageCenteriSCSIDriver:delliscsi
CINDER_DEFAULT_VOLUME_TYPE=dellfc

[[post-config|$CINDER_CONF]]

[default]
use_multipath_for_image_xfer = True
enforce_multipath_for_image_xfer = True

[dellfc]
volume_backend_name = dellfc
volume_driver = cinder.volume.drivers.dell_emc.sc.storagecenter_FC.SCFCDriver
san_ip = <IP address> # IP of SC MGMT or DSM
san_login = <ID> # ID for SC MGMT or DSM
san_password = <password> # Password for SC MGMT or DSM
dell_sc_ssn = 350032 # System Serial Number
dell_sc_api_port = 3033 # API port for DSM
dell_sc_server_folder = OSP_Servers/dellfc/queens
dell_sc_volume_folder = OpenStackVolumes/dellfc/queens

[[post-config|$NOVA_CONF]]
[libvirt]
iscsi_use_multipath = True
```

### 2.3.5 Use with OpenStack

When used with OpenStack, the recommended configuration approach is to append the following syntax to the end of the **/etc/cinder/cinder.conf** file. Restart the OpenStack environment, including the respective Cinder-based processes, to apply these changes. The following configuration contains comments in **red**.

```
# cat ~/devstack/local.conf

[snip]

# Dell SC Series
#-----

# enable multiple Cinder storage backends
CINDER_ENABLED_BACKENDS=DellStorageCenterFCDriver:dellfc,
DellStorageCenterFCDriver:dellfc_repl,DellStorageCenteriSCSIDriver:delliscsi
CINDER_DEFAULT_VOLUME_TYPE=dellfc

[[post-config|$CINDER_CONF]]

[default]
use_multipath_for_image_xfer = True
enforce_multipath_for_image_xfer = True

[dellfc]
volume_backend_name = dellfc
volume_driver = cinder.volume.drivers.dell_emc.sc.storagecenter_fc.SCFCDriver
san_ip = <IP address>           # IP of SC MGMT or DSM
san_login = <ID>                 # ID for SC MGMT or DSM
san_password = <password>       # Password for SC MGMT or DSM
dell_sc_ssn = 350032             # System Serial Number
dell_sc_api_port = 3033         # API port for SC MGMT or DSM
dell_sc_server_folder = OSP_Servers/dellfc/queens
dell_sc_volume_folder = OpenStackVolumes/dellfc/queens

[[post-config|$NOVA_CONF]]
[libvirt]
iscsi_use_multipath = True
```

When used with OpenStack, the recommended configuration approach is to append the following syntax to the end of the **/etc/nova/nova.conf** file. Restart the OpenStack environment, including the respective Nova-based processes, to apply these changes.

```
# cat /etc/nova/nova.conf

[snip]

[libvirt]
iscsi_use_multipath = True
```

### 2.3.6 Use with multiple back-end definitions

The SC Series driver supports the use of multiple back-end configurations. This means that volumes can be provisioned and managed over Fibre Channel or iSCSI simultaneously, and from single or multiple SC Series arrays.

This configuration is accomplished with multiple schemas, where each volume type would be defined within its own schema definition. An additional Dell EMC iSCSI-based schema would look similar to the following example with comments in red.

```
# cat ~/devstack/local.conf

[snip]

# Dell SC Series
#-----

# enable multiple Cinder storage backends
CINDER_ENABLED_BACKENDS=DellStorageCenterFCDriver:dellfc,
DellStorageCenterFCDriver:dellfc_repl,DellStorageCenteriSCSIDriver:delliscsi
CINDER_DEFAULT_VOLUME_TYPE=dellfc

[[post-config|$CINDER_CONF]]

[dellfc]
volume_backend_name = dellfc
volume_driver = cinder.volume.drivers.dell_emc.sc.storagecenter_fc.SCFCDriver
san_ip = <IP address>           # IP of SC MGMT or DSM
san_login = <ID>                 # ID for SC MGMT or DSM
san_password = <password>       # Password for SC MGMT or DSM
dell_sc_ssn = 350032             # System Serial Number
dell_sc_api_port = 3033         # API port for SC MGMT or DSM
dell_sc_server_folder = OSP_Servers/dellfc/queens
dell_sc_volume_folder = OpenStackVolumes/dellfc/queens

[dellfc_repl]
volume_backend_name = dellfc_repl
volume_driver = cinder.volume.drivers.dell_emc.sc.storagecenter_fc.SCFCDriver
san_ip = <IP address>           # IP of SC MGMT or DSM
san_login = <ID>                 # ID for SC MGMT or DSM
san_password = <password>       # Password for SC MGMT or DSM
dell_sc_ssn = 350032             # System Serial Number
dell_sc_api_port = 3033         # API port for SC MGMT or DSM
dell_sc_server_folder = OSP_Servers/dellfc_repl/queens
dell_sc_volume_folder = OpenStackVolumes/dellfc_repl/queens
replication_device = target_device_id:101,qosnode:'FC - 4 GBPS'
```

```
[delliscsi]
volume_backend_name = delliscsi
volume_driver =
cinder.volume.drivers.dell_emc.sc.storagecenter_iscsi.SCISCSIDriver
san_ip = <IP address>           # IP of SC MGMT or DSM
san_login = <ID>                 # ID for SC MGMT or DSM
san_password = <password>       # Password for SC MGMT or DSM
dell_sc_ssn = 351396             # System Serial Number
iscsi_ip_address = 10.10.10.200 # SC iSCSI target IP
iscsi_port = 3260                # SC iSCSI target port
dell_sc_api_port = 3033          # API port for SC MGMT or DSM
dell_sc_server_folder = OSP_Servers/delliscsi/queens
dell_sc_volume_folder = OpenStackVolumes/delliscsi/queens

[[post-config|$NOVA_CONF]]
[libvirt]
iscsi_use_multipath = True
```

---

**Note:** OpenStack establishes iSCSI login only on paths which are presenting volumes on the active controller. This may result in SC Series array server object mapping, showing certain paths in a Partial Connectivity state. This is proper functionality by design and the SC Series array is operating correctly and nominally.

---

## 2.4 Volume management and snapshots

In the DSM client interface, OpenStack volumes/server objects are created in the folders defined by the `dell_sc_server_folder` and `dell_sc_volume_folder` attributes within the `local.conf` or `cinder.conf` files, respectively. In the previous iSCSI examples, the `dell_sc_volume_folder` attribute is defined as `OpenStackVolumes/delliscsi/queens`.

Volumes created on SC Series storage with Cinder are automatically named by their Cinder volume IDs as referenced in Figure 5 (outlined in red). The volume ID is correlated to the respective volume type and name at time of creation. Any Cinder-created volumes on SC Series storage are also marked with a note stating **Created by Dell EMC Cinder Driver** as referenced in Figure 5 (outlined in blue).

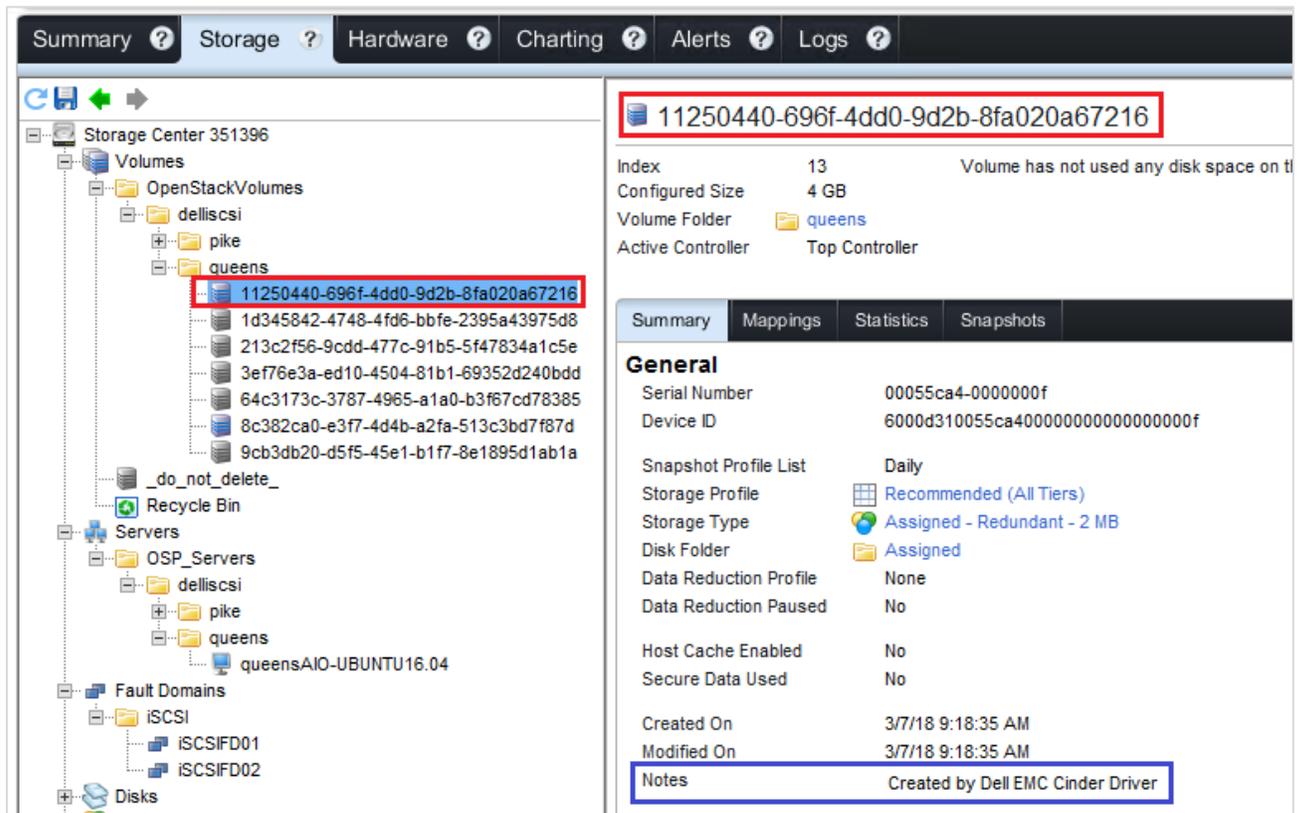


Figure 5 DSM view of Cinder-created volumes

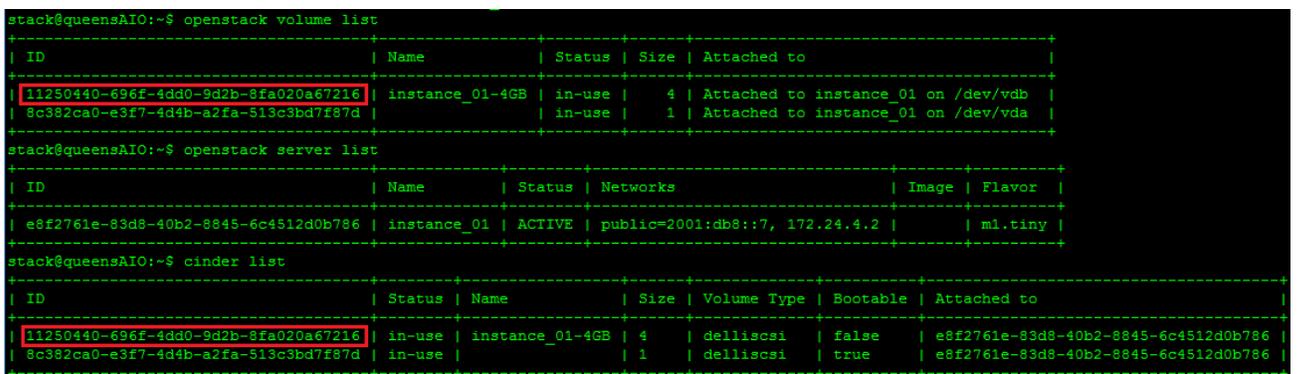


Figure 6 CLI of Cinder-created SC Series volumes

Keep the default names and locations of these SC Series volumes. These SC Series volumes are created and managed through OpenStack, and their naming convention provides a unique and accurate method towards referencing the proper volume during volume operations. If these Cinder-created SC Series volumes are moved or renamed, through DSM rather than through OpenStack CLI or Horizon dashboard, these volumes will become unavailable to Cinder, thus causing service disruption within the OpenStack platform.

The Create Snapshot feature in the Horizon interface creates SC Series snapshots. These are always created as never-expiring snapshots as referenced in Figure 7 (outlined in red). Manage these snapshots using the OpenStack Horizon interface and not with DSM.

Freeze Time	Expire Time	Size	% of Actual
Active		0 MB	0%
3/21/18 9:12:17 AM	Never Expires	0 MB	0%
3/21/18 12:01:05 AM	3/28/18 12:01:05 AM	0 MB	0%
3/20/18 12:01:05 AM	3/27/18 12:01:05 AM	0 MB	0%
3/19/18 12:01:05 AM	3/26/18 12:01:05 AM	0 MB	0%
3/18/18 12:01:05 AM	3/25/18 12:01:05 AM	0 MB	0%
3/17/18 12:01:05 AM	3/24/18 12:01:05 AM	0 MB	0%
3/16/18 12:01:05 AM	3/23/18 12:01:05 AM	0 MB	0%
3/15/18 12:01:05 AM	3/22/18 12:01:05 AM	0 MB	0%

Figure 7 Never-expiring snapshot of Cinder volume created in Horizon

## 2.5 Replication and Live Volume

This section presents the creation of a multi-attribute-based replication relationship as well as a Live Volume relationship with the use of the Cinder CLI interface.

The following volume type definitions are created, **repl\_async** and **repl\_livevol**, and each is bound to a volume named **testVol**. The following declarations define the parameters for all existing or newly created volumes bound to this volume type definition.

### Volume Type: repl\_async

- **Storage profile:** Recommended (All Tiers)
- **Replay profile** (snapshot schedule): Daily
- **Replication:** Async

```
# cinder type-create "repl_async"
# cinder type-key "repl_async" set volume_backend_name='dellfc_repl'
# cinder type-key "repl_async" set storagetype:storageprofile='Recommended (All Tiers)'
# cinder type-key "repl_async" set storagetype:replayprofile='Daily'
# cinder type-key "repl_async" set replication_enabled='<is> True'
# cinder type-key "repl_async" set replication:activereplay='<is> True'
# cinder manage --volume-type repl_async william@dellfc#dellfc "testVol"
```

Alternately, creating a display-name **testVol** --volume-type **repl\_async** --availability-zone **nova** new volume is shown as follows.

```
# cinder create 5 --
```

#### Volume Type: repl\_livevol

- **Storage profile:** Recommended (All Tiers)
- **Replay profile** (snapshot schedule): Daily
- **Replication:** Async

```
# cinder type-create "repl_livevol"
# cinder type-key "repl_livevol" set volume_backend_name='dellfc_repl'
# cinder type-key "repl_livevol" set storagetype:storageprofile='Recommended
(All Tiers) '
# cinder type-key "repl_livevol" set storagetype:replayprofile='Daily'
# cinder type-key "repl_livevol" set replication_enabled='<is> True'
# cinder type-key "repl_livevol" set replication:livevolume='<is> True'
# cinder manage --volume-type repl_livevol william@dellfc#dellfc "testVol"
```

Alternately, creating a new volume is shown as follows.

```
# cinder create 5 --display-name testVol --volume-type repl_livevol --
availability-zone nova
```

## 2.6 Creating volumes using OpenStack Horizon

This method explains how to use the OpenStack Horizon interface to create volumes.

1. In the Horizon interface, use either the Admin or another defined user ID.
2. On the **Volumes** tab, click **Create Volume** to display the dialog box. The following screenshot provides an example of populated fields.

**Note:** The Type field shows **dellfc** and correlates to the `volume_backend_name` attribute as configured in the `local.conf` or `cinder.conf` files.

3. Click **Create Volume** in the dialog box to create the 10 GB volume on the SC Series array. The 10 GB volume is created and shown in an **Available** state.

Name	Description	Size	Status	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
dellsc_10G	10G demo volume	10GB	Available	dellfc		nova	No	No	Edit Volume

The **Extend Volume**, **Create Snapshot**, and **Delete Volume** functions are enabled as well.

The screenshot shows the 'Volumes' tab in the OpenStack dashboard. A table lists one volume: 'dellsc\_10G' (10G demo volume, 10GB size, Available status, dellfc type, attached to nova). The 'Actions' column for this volume has a dropdown menu open, showing options: 'Extend Volume', 'Manage Attachments', 'Create Snapshot', 'Change Volume Type', 'Upload to Image', 'Create Transfer', and 'Delete Volume'. The 'Extend Volume' option is highlighted with a mouse cursor.

## 2.7 Attaching volumes with Horizon

In the Horizon interface, attaching an SC Series volume to an instance or virtual machine can be accomplished by multiple methods. One of these methods is demonstrated in this section with an instance named **dellsc\_10G\_instance**.

1. In the **Volumes** tab > **Actions** drop-down menu, select **Manage Attachments**.

This screenshot is similar to the one above, but the 'Manage Attachments' option in the actions dropdown menu is highlighted with a mouse cursor. The table data remains the same: 'dellsc\_10G' (10G demo volume, 50GB size, Available status, dellfc type, attached to nova).

- Identify and select the instance to attach the volume and click **Attach Volume**.

### Manage Volume Attachments ✕

Instance	Device	Actions
No items to display.		
Displaying 0 items		

### Attach To Instance

Attach to Instance \* 🔍

dellsc\_10G\_instance (5d1a1bcc-402c-46fa-ab71-0f779c5f19f0)
▼

Cancel
Attach Volume

The volume is attached to the instance as device **/dev/vdb**.

### Volumes

Volumes Volume Snapshots

🔍
+ Create Volume
≡ Accept Transfer
✕ Delete Volumes

<input type="checkbox"/>	Name	Description	Size	Status	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
<input type="checkbox"/>	dellsc_10G	10G demo volume	50GB	In-use	dellfc	Attached to dellsc_10G_instance on /dev/vdb	nova	No	No	Edit Volume ▼

Displaying 1 item

3. View the device from the login shell on the instance by using the **fdisk -l** command.

```
$ sudo fdisk -l
```

```
Disk /dev/vda: 21.5 GB, 21474836480 bytes
16 heads, 63 sectors/track, 41610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

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

```
Disk /dev/vdb: 53.7 GB, 53687091200 bytes
16 heads, 63 sectors/track, 104025 cylinders, total 104857600 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

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

## 3 Ocata updates

The SC Series driver has incorporated a number of feature improvements within the OpenStack Ocata release. This section discusses these improvements, as well as the methods to apply them using the Cinder CLI and any additional syntax required within the **local.conf/cinder.conf** or **nova.conf** configuration files.

### 3.1 Live Volume with Auto Failover (AFO)

The SC Series Live Volume feature with OpenStack is further enhanced with the integration of Auto Failover (AFO). Auto Failover enables the SC Series array to automatically fail over volumes from a failed primary SC Series array to an alternate SC Series array without user intervention while providing seamless and uninterrupted volume access to the OpenStack Cinder driver.

A replication-enabled volume back-end definition is required for all Live Volume AFO-intended volumes.

---

**Note:** When a Live Volume AFO event occurs, volumes presented from the primary SC Series array will no longer be available for use and may cause interruption to the host instance which relies upon use of this volume. However, the volume will be accessible through the alternate SC Series array and available for the creation of new host instances. Live Volume AFO does not automatically fail over from the alternate SC Series array back to the primary SC Series array upon its recovery or availability; this requires manual user intervention.

---

```
# cinder type-create "LiveVolAFO"
# cinder type-key "LiveVolAFO" set volume_backend_name='dellfc_repl'
# cinder type-key "LiveVolAFO" set replication_enabled='<is> True'
# cinder type-key "LiveVolAFO" set replication:livevolume:autofailover='<is>
True'
```

Use the following commands to enable volume snapshots for Live Volume AFO-intended volumes to achieve the proper desired function and behavior.

```
# cinder type-key "LiveVolAFO" set replication:activereplay='<is> True'
# cinder type-key "LiveVolAFO" set storagetype:replayprofiles='Daily'
```

### 3.2 Data deduplication and compression

SC Series deduplication and compression features are integrated with this OpenStack release. These features enable the SC Series array to intelligently apply deduplication and compression algorithms to the data being stored.

Solid-state drives (SSDs) are required to enable deduplication with compression features. The use of compression features alone does not require the use of SSDs.

### 3.2.1 Deduplication with compression

Deduplication algorithms are automatically applied with compression, and are achieved with the following Cinder commands.

```
# cinder type-create "dedupe_vol"
# cinder type-key "dedupe_vol" set volume_backend_name='dellfc'
# cinder type-key "dedupe_vol" set
storagetype:datareductionprofile='Deduplication'
```

### 3.2.2 Compression

Compression algorithms can be applied independently of deduplication, and are achieved with the following Cinder commands.

```
# cinder type-create "compress_vol"
# cinder type-key "compress_vol" set volume_backend_name='dellfc'
# cinder type-key "compress_vol" set
storagetype:datareductionprofile='Compression'
```

## 3.3 Volume QoS profiles

SC Series Quality of Service (QoS) profiles are introduced with OpenStack Ocata. QoS profiles enable the user or administrator to configure volumes with predefined and deterministic usage (IOPS, MB/sec, or both) limits.

QoS profiles or limits need to be enabled and predefined within DSM before they can be used and applied with the following Cinder commands. Volume and group QoS profiles may be used independently or will apply concurrently if both are defined.

```
# cinder type-create "myqos"
# cinder type-key "myqos" set volume_backend_name='dellfc'
# cinder type-key "myqos" set storagetype:volumesqos='IOPS_10k'
```

Group QoS profiles are implemented with the following Cinder command.

```
# cinder type-key "grpqos" set storagetype:groupqos='MBps_5k'
```

## 4 Newton updates

The SC Series driver incorporated a number of feature improvements within the OpenStack Newton release. This section discusses these improvements, as well as the methods to apply them using the Cinder CLI and any additional syntax required within the **local.conf/cinder.conf** or **nova.conf** configuration files.

### 4.1 Live Volume

SC Series Live Volume features build upon the replication features introduced with OpenStack Mitaka. They can be applied to volumes created and managed by Cinder, and enable the replication of a volume from an SC Series array to another in either asynchronous active replay (snapshot) or synchronous mode.

Replication relationships, like storage profiles, are configured and managed through the extra-specs attributes of the volume type definition, and thereafter bind the desired volume type definition to the volume during either volume creation or post-creation.

The creation and definition of the SC Series Live Volume relationship is shown in the following.

```
# cinder type-create "LiveVol"
# cinder type-key "LiveVol" set replication_enabled='<is> True'
# cinder type-key "LiveVol" set replication:livevolume='<is> True'
```

The default replication type for Live Volume relationships is asynchronous with active replay enabled. Apply the additional syntax shown in the following line to place the Live Volume relationship into synchronous, high availability mode.

```
# cinder type-key "LiveVol" set replication_type='<in> sync'
```

In either asynchronous or synchronous modes, failover from the primary SC Series array to the destination SC Series array is **not** automatic. The following command should be issued to initiate a failover (and failback) of an OpenStack Cinder-managed host from the primary SC array to the destination SC array. This command can also be applied to normal (non-Live Volume) volume replication relationships (Mitaka update).

```
# cinder failover-host [--backend_id <backend ID>] <hostname>
```

In this command, **backend\_id** is an optional value that represents the backend ID value of the destination SC array if multiple backend definitions have been created. The default value of this parameter is **None**.

Replication statements inside **local.conf/cinder.conf** are still required to denote that the back-end definition is replication-enabled. An additional parameter **remoteqos** is required to indicate and define the destination SC Series array QoS node profile as shown in the following.

```
# Establish replication relationship from 862 to 101, and QOS Node profile
# for any volume objects created using this volume type definition
replication_device = target_device_id:101,qosnode:'FC - 4
GBPS',remoteqos:'cinderqos'
```

The **remoteqos** QoS node profile will be created with default values of 1 Gbps and will not be bandwidth limited if it has not been previously created or defined.

## 5 Mitaka updates

The SC Series driver has incorporated a number of feature improvements within the OpenStack Mitaka release. This section discusses these improvements, as well as the methods to apply them using the Cinder CLI and any additional syntax required within the **local.conf/cinder.conf** or **nova.conf** configuration files.

### 5.1 Replication

SC Series replication can be applied to volumes created and managed by Cinder, and enables the replication of a volume from an SC Series array to another in either asynchronous active snapshot (replay) or synchronous mode.

Replication relationships, like storage profiles, are configured and managed through the extra-specs attributes of the volume type definition, and thereafter binding the desired volume type definition to the volume during either volume creation or post-creation.

The creation and definition of asynchronous active snapshot replication is shown in the following. Active snapshot is a feature of SC Series arrays which replicates every change from the source volume to the destination volume in real time instead of replicating the last frozen snapshot. Active snapshot is recommended when data integrity between sites/volumes is critical.

```
# cinder type-create "Repl_Async"
# cinder type-key "Repl_Async" set replication_enabled='<is> True'
# cinder type-key "Repl_Async" set replication:activereplay='<is> True'
```

The definition of synchronous replication is shown in the following. The synchronous relationship defaults the replication relationship into high availability mode and cannot be changed at this time.

```
# cinder type-create "Repl_Sync"
# cinder type-key "Repl_Sync" set replication_enabled='<is> True'
# cinder type-key "Repl_Sync" set replication_type='<in> sync'
```

Finally, the Horizon association of a volume type definition to a volume is presented and discussed in section 6.2 while the CLI equivalent of this procedure is shown as follows.

```
# cinder manage --volume-type <volume_type_name>
<hostname>@<volume_type>#<pool_name> "<volume name>"
```

---

**Note:** The `replication_enabled`, `replication:activereplay`, and `replication_type` attributes cannot be predefined through `local.conf` or `cinder.conf` and must be configured through Horizon or the CLI. Multiple extra specs attributes may be defined within a single volume type definition.

---

## 5.1.1 Replication target

An SC Series back-end can be defined in a replication-enabled or replication-disabled state. An SC Series back-end definition in its default state is replication-disabled, as discussed in sections 2.3.3–2.3.6. The back-end replication status can only be preconfigured in **local.conf** or **cinder.conf** files and cannot be modified after configuration through Horizon or the CLI.

To create a replication-enabled backend, append the following line (shown in **bold**) to the end of the desired backend definition.

```
[dellfc_repl]
# Name the storage backend
volume_backend_name = dellfc_repl
# Load this driver module, DellStorageCenterFCDriver
volume_driver = cinder.volume.drivers.dell_emc.sc.storagecenter_fc.SCFCDriver
# IP address of the DSM installation
san_ip = <IP address>
# DSM login credentials
san_login = <ID>
# DSM login password
san_password = <password>
# DSM default api port
dell_sc_api_port = 3033
# Dell SC array Index
dell_sc_ssn = 350032
# Create new server objects in this folder
dell_sc_server_folder = OSP_Servers/dellfc_repl/queens
# Create new volume objects in this folder
dell_sc_volume_folder = OpenStackVolumes/dellfc_repl/queens
# Establish replication relationship from 862 to 101, and QoS Node profile
# for any volume objects created using this volume type definition
replication_device = target_device_id:101,qosnode:'FC - 4 GBPS'
```

The **target\_device\_id** key represents the destination SC Series controller index and the **qosnode** key represents the QoS relationship binding the source and destination SC Series arrays. The replication relationship between SC Series arrays must be predefined and already exist.

---

**Note:** An SC Series volume is replicated only when there is a replication target defined and the volume is bound to a volume type definition which contains the extra specs attribute **replication\_enabled=<is> True'**. Additionally, the SC Series volume is replicated to the destination SC Series array within the same folder tree as the source volume, and the replicated volume name is prefixed with the text **Cinder repl of**.

---

## 5.2 Snapshot profiles (snapshot schedules)

SC Series snapshot profiles (snapshot schedules) can be applied to volumes created and managed by Cinder, and define the frequency in which snapshots (replays) are automatically created from the defined volume, as well as duration of time in which those snapshots are kept. The **Daily** snapshot profile (default) captures a snapshot daily at 12.01am and has a default expiry of 7 days.

Snapshots are critical to the Data Progression feature. SC Series volumes without any current or captured snapshots **will not** progress data down into more cost-effective storage tiers and will remain in the storage tier in which it was originally written.

Snapshot profiles, like storage profiles, are configured and managed through the extra spec attributes of the volume type definition, and thereafter binding the desired volume type definition to the volume during either volume creation or post-creation.

The creation and definition of a snapshot profile volume type definition is shown as follows.

```
# cinder type-create "Gold_Tier"
# cinder type-key "Gold_Tier" set storagetype:replayprofile=dellfc_tier1
```

or

```
# cinder type-create "Bronze_Tier"
# cinder type-key "Bronze_Tier" set storagetype:replayprofile=dellfc_tier3
```

Finally, the Horizon association of a volume type definition to a volume is presented and discussed in section 6.2 while the CLI equivalent of this procedure is shown in the following.

```
# cinder manage --volume-type <volume_type_name>
<hostname>@<volume_type>#<pool_name> "<volume name>"
```

---

**Note:** The **storagetype:replayprofile** attribute cannot be predefined through local.conf or cinder.conf and must be configured through Horizon or the CLI. Multiple extra spec attributes may be defined within a single volume type definition.

---

## 6 Liberty updates

The SC Series driver has incorporated a number of feature improvements within the OpenStack Liberty release. This section discusses these improvements, as well as the methods to apply them using the Horizon dashboard and the CLI, where applicable.

### 6.1 Storage profiles

SC Series storage profiles can be applied to volumes created and managed by Cinder. They define which disk tier is used to accept initial writes into the SC Series array, as well as how Data Progression moves data between tiers.

---

**Note:** Storage profiles are a supported feature of Dell Storage SC8000 and SC9000 arrays only.

---

Storage profiles are configured and managed through the extra specs attributes of the volume type definition. Use the following procedure to apply storage profiles:

1. In the Horizon dashboard, click the **System** drop-down menu, select **Volumes**, and click the **Volumes Types** tab.

The screenshot shows the OpenStack Horizon dashboard interface. The top navigation bar includes the OpenStack logo, the project name 'alt\_demo', and the user 'admin'. The left sidebar shows a navigation menu with categories like Project, Admin, Compute, Volume, Volumes, Snapshots, Network, System, and Identity. The main content area is titled 'Volume Types' and shows a table with one item: 'delliscsi'. The table has columns for Name, Description, Associated QoS Spec, Encryption, Public, and Actions. The 'Actions' column for 'delliscsi' contains a 'Create Encryption' button. Below the table, there is a section for 'QoS Specs' with a 'Create QoS Spec' button.

Name	Description	Associated QoS Spec	Encryption	Public	Actions
delliscsi	-		-	Yes	Create Encryption

- For the **delliscsi** volume type, click the **Actions** drop-down menu and select the **View Extra Specs** option.

**Volume Types**

Volume Types

Displaying 1 item

<input type="checkbox"/>	Name	Description	Associated QoS Spec	Encryption	Public	Actions
<input type="checkbox"/>	delliscsi	-	-	-	Yes	Create Encryption View Extra Specs Manage QoS Spec Association Edit Volume Type Delete Volume Type Update Metadata

Displaying 1 item

**QoS Specs**

Name	Consumer	Specs	Actions
------	----------	-------	---------

- The dialog box displays a single key value pair where **volume\_backend\_name = delliscsi**. Click **Close**.

**Volume Type Extra Specs**

<input type="checkbox"/>	Key	Value	Actions
<input type="checkbox"/>	volume_backend_name	delliscsi	Edit <input type="button" value="x"/>

Displaying 1 item

- To create a storage profile that pairs the volume type to SC Series storage, click **Create Volume Type**.

## Volumes

Volumes Volume Types Volume Snapshots

Volume Types Filter + Create Volume Type ✖ Delete Volume Types

<input type="checkbox"/>	Name	Description	Associated QoS Spec	Encryption	Actions
<input type="checkbox"/>	dellfc	-		-	Create Encryption ▾
<input type="checkbox"/>	delliscsi	-		-	Create Encryption ▾

Displaying 2 items

QoS Specs + Create QoS Spec

Name	Consumer	Specs	Actions
No items to display.			

Displaying 0 items

- Name the new volume type and click **Create Volume Type**.

### Create Volume Type ✖

**Name \***

**Description**

**Public** ⓘ

Cancel
Create Volume Type

**Description:**

Volume type is a type or label that can be selected at volume creation time in OpenStack. It usually maps to a set of capabilities of the storage back-end driver to be used for this volume. Examples: "Performance", "SSD", "Backup", etc. This is equivalent to the `openstack volume type create` command. Once the volume type gets created, click the "View Extra Specs" button to set up extra specs key-value pair(s) for that volume type.

- For the newly created **delliscsi\_Gold** volume type, click the **Actions** drop-down menu and select the **View Extra Specs** option.

**Volume Types**

Volume Types

Displaying 2 items

<input type="checkbox"/>	Name	Description	Associated QoS Spec	Encryption	Public	Actions
<input type="checkbox"/>	delliscsi_Gold			-	Yes	Create Encryption <input type="button" value="v"/>
<input type="checkbox"/>	delliscsi	-		-	Yes	View Extra Specs Manage QoS Spec Association Edit Volume Type Delete Volume Type Update Metadata

Displaying 2 items

**QoS Specs**

- Create two new key value pairs in which the value of the **storagetype:storageprofile** attribute references the name of the storage profile as preconfigured on the SC Series array.

**Volume Type Extra Specs**

<input type="checkbox"/>	Key	Value	Actions
<input type="checkbox"/>	storagetype:storageprofile	delliscsi_Gold	Edit <input type="button" value="v"/>
<input type="checkbox"/>	volume_backend_name	delliscsi	Edit <input type="button" value="v"/>

Displaying 2 items

**Note:** The value of the **storagetype:storageprofile** attribute is lowercase without spaces to match the pattern of the storage profile name on the SC Series array. The storage profile should be predefined on the target SC Series array before attempting this procedure.

Subsequent volume creation through Horizon should use the **delliscsi\_Gold** volume type if the volume is intended to land in the **delliscsi\_Gold** tier on the destination SC Series array. These methods can be applied when mapping multiple volume types to storage profiles for different use-case scenarios.

This function, through the Horizon dashboard, can also be accomplished using the following CLI statements.

```
# cinder type-create delliscsi_Gold
# cinder type-key delliscsi_Gold set volume_backend_name=delliscsi
# cinder type-key delliscsi_Gold set storagetype:storageprofile=delliscsi_Gold
```

---

**Note:** The **volume\_backend\_name** and **storagetype:storageprofile** attributes cannot be predefined through `local.conf` or `cinder.conf` and have to be configured after creation through Horizon or CLI.

---

## 6.2 Managing existing volumes

This feature allows existing SC Series volumes to be imported into Cinder awareness and placed under OpenStack management. The import function and any subsequent export function from OpenStack management maintains volume and data integrity. In contrast, a volume delete function through OpenStack is a volume- and data-destructive action.

These import and export functions through OpenStack are known as manage and unmanage actions, respectively.

---

**Note:** Any volumes intended for import into OpenStack management must be in an unmapped state between the SC Series array and the intended Cinder host.

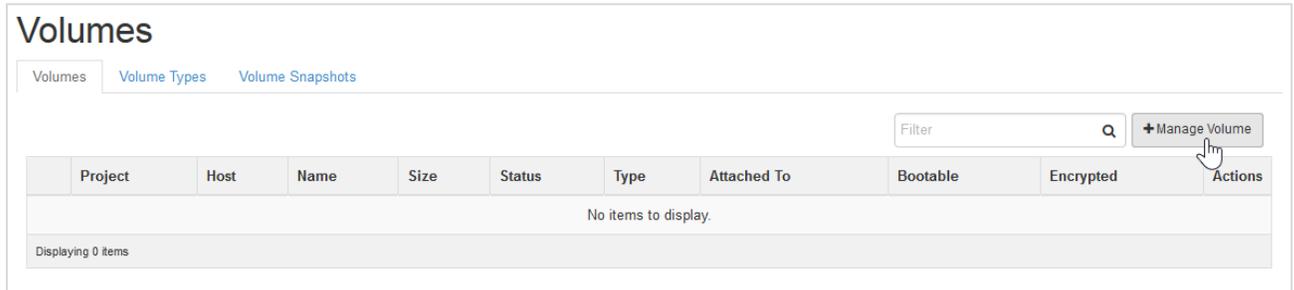
---

Use the following procedure to manage existing volumes:

1. In the Horizon dashboard, click the **System** drop-down menu, click **Volumes**, and click the **Volumes** tab.

The screenshot shows the Horizon dashboard interface. On the left, a navigation sidebar is visible with the 'System' menu expanded, and 'Volumes' is selected. The main content area is titled 'Volumes' and has three tabs: 'Volumes', 'Volume Types', and 'Volume Snapshots'. The 'Volumes' tab is active, showing a table with the following columns: Project, Host, Name, Size, Status, Type, Attached To, and Bo. The table is currently empty, with the message 'No items to display.' and 'Displaying 0 items' shown below the header. A 'Fit' button is visible in the top right corner of the table area.

2. Click **Manage Volume**.



3. Complete the **Identifier** field with the name of the SC Series volume name as shown in DSM.
4. Leave the **Identifier Type** field in its default value of **Name**.
5. In the **Host** field, enter the name of the host and volume type.
6. The **Volume Name** field is optional and should be left blank.
7. In the **Volume Type** field, select the destination volume type of the managed volume.
8. Leave the **Availability Zone** field in its default value of **nova**.
9. Click **Manage**.

The screenshot shows the 'Manage Volume' form. The fields are filled as follows:
 

- Identifier \***: delliscsi\_unmanaged\_vol
- Identifier Type \***: Name
- Host \***: william@delliscsi#delliscsi
- Volume Name**: (empty)
- Description**: (empty text area)
- Metadata**: (empty text area)
- Volume Type**: delliscsi\_Gold
- Availability Zone**: nova
- Bootable**:

 The 'Description' section contains the text: "Manage" an existing volume from a Cinder host. This will make the volume visible within OpenStack. This is equivalent to the `cinder manage` command. At the bottom right, there are 'Cancel' and 'Manage' buttons.

This procedure brings the volume into OpenStack management through Cinder. This volume will be renamed accordingly and assigned an OpenStack manageable id value and interfaced through Horizon similar to traditionally created and managed Cinder volumes.

The previous procedure can also be replicated using the following CLI statements:

```
(keystone_admin)# cinder manage --volume-type delliscsi_Gold
william@delliscsi#delliscsi "delliscsi_unmanaged_vol"
```

## 6.3 Consistency groups

The use of SC Series consistency groups with OpenStack or DevStack can only be configured and managed through the CLI.

Edit the `/etc/cinder/policy.json` file with a text editor. Locate all key value pairs which show:

```
"consistencygroup:*" : "group:nobody"
```

The `*` represents one or multiple sub-key values. For each entry identified, replace the **group:nobody** value with a null value (`" "`). For example:

```
"consistencygroup:*" : ""
```

Restart OpenStack-Cinder-based services and processes or restack the DevStack environment as necessary to apply these changes.

### 6.3.1 Creating a consistency group

The following example shows a new consistency group being created. The name and volume type definitions are required parameters.

```
[stack@queensAIO ~(keystone_admin)]$ cinder consisgroup-create --name ClusterGrp delliscsi
+-----+-----+
| Property          | Value                               |
+-----+-----+
| availability_zone  | nova                                 |
| created_at        | 2018-03-19T19:10:46.000000          |
| description       | None                                 |
| id                | 8601fa13-7b5d-49a7-b367-8361682afffe |
| name              | ClusterGrp                          |
| status            | creating                             |
| volume_types      | [u'19b63f1b-62c1-4983-8fa7-80481de75d60'] |
+-----+-----+
```

### 6.3.2 Creating a volume into a consistency group

The following command creates a new 10 GB volume into the new consistency group. Multiple additional volumes can be created in similar fashion.

```
[stack@queensAIO ~(keystone_admin)]$ cinder create --volume-type delliscsi --name ClusterGrp_Vol01 --consisgroup-id 8601fa13-7b5d-49a7-b367-8361682afffe 10
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	8601fa13-7b5d-49a7-b367-8361682afffe
created_at	2018-03-19T19:12:50.000000
description	None
encrypted	False
id	81c515d6-95e3-4812-8cf1-68b69b58061b
metadata	{}
migration_status	None
multiattach	False
name	ClusterGrp_Vol01
os-vol-host-attr:host	None
os-vol-mig-status-attr:migstat	None
os-vol-mig-status-attr:name_id	None
os-vol-tenant-attr:tenant_id	784baabb1b984edeb40938f48648818f
replication_status	None
size	10
snapshot_id	None
source_vol_id	None
status	creating
updated_at	2018-03-19T19:12:50.000000
user_id	5e6b94871d354c02bda4e8c434ecab62
volume_type	delliscsi

### 6.3.3 Creating, viewing, and removing snapshots (replays) of a consistency group

The following commands are used to manage consistency group snapshots (replays).

Capture and manage consistency group snapshots:

```
[stack@queensAIO ~(keystone_admin)]$ cinder cgsnapshot-create 8601fa13-7b5d-49a7-b367-8361682afffe
```

Property	Value
consistencygroup_id	8601fa13-7b5d-49a7-b367-8361682afffe
created_at	2018-03-19T19:14:29.000000
description	None
id	f8edaf74-52f6-49a5-8d16-89cf8e3997e2
name	None
status	creating

List all consistency group snapshots:

```
[stack@queensAIO ~(keystone_admin)]$ cinder cgsnapshot-list
```

ID	Status	Name
f8edaf74-52f6-49a5-8d16-89cf8e3997e2	available	-

Display the status of a consistency group snapshot:

```
[stack@queensAIO ~(keystone_admin)]$ cinder cgsnapshot-show f8edaf74-52f6-49a5-8d16-89cf8e3997e2
+-----+
| Property          | Value                                     |
+-----+
| consistencygroup_id | 8601fa13-7b5d-49a7-b367-8361682afffe |
| created_at         | 2018-03-19T19:14:29.000000             |
| description        | None                                     |
| id                 | f8edaf74-52f6-49a5-8d16-89cf8e3997e2 |
| name               | None                                     |
| status             | available                                |
+-----+
```

Delete a consistency group snapshot:

```
[stack@queensAIO ~(keystone_admin)]$ cinder cgsnapshot-delete f8edaf74-52f6-49a5-8d16-89cf8e3997e2
```

### 6.3.4 Displaying the consistency group

A consistency group is managed and referenced by its id value as shown in the following example.

```
[stack@queensAIO ~(keystone_admin)]$ cinder consisgroup-list
+-----+
| ID                  | Status   | Name      |
+-----+
| 8601fa13-7b5d-49a7-b367-8361682afffe | available | ClusterGrp |
+-----+
[stack@queensAIO ~(keystone_admin)]$ cinder consisgroup-show 8601fa13-7b5d-49a7-b367-8361682afffe
+-----+
| Property          | Value                                     |
+-----+
| availability_zone | nova                                     |
| created_at         | 2018-03-19T19:10:47.000000             |
| description        | None                                     |
| id                 | 8601fa13-7b5d-49a7-b367-8361682afffe |
| name               | ClusterGrp                             |
| status             | available                                |
| volume_types      | [u'19b63f1b-62c1-4983-8fa7-80481de75d60'] |
+-----+
```

### 6.3.5 Removing a consistency group

A consistency group is removed with the force flag as shown in the following example.

```
[stack@queensAIO ~(keystone_admin)]$ cinder consisgroup-delete --force 8601fa13-7b5d-49a7-b367-8361682afffe
```

## 7 Volume management with logical volume manager (LVM) and CLI

When using SC Series volumes with Cinder (and an LVM-based backend) as defined in the [OpenStack Configuration Reference](#), incorporate the following volume-management methods:

1. Present the SC Series volumes to the OpenStack compute nodes and then scan for new volumes (both FC and iSCSI) on the compute node with the following script:

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

This script and other Linux methods are discussed in the best practices document, [Dell EMC SC Series with Red Hat Enterprise Linux \(RHEL\) 6x](#).

2. After volumes have been discovered and identified, the volumes or devices are required to be part of an LVM volume group named **cinder-volumes**. This is accomplished with the following script.

```
# pvcreate /dev/sdd
# vgcreate cinder-volumes /dev/sdd
```

3. If the **cinder-volumes** volume group already exists, new volumes are added to extend the volume group with the following command.

```
# pvcreate /dev/sdX
# vgextend cinder-volumes /dev/sdX
```

### 7.1 Managing volumes with Cinder

After the volumes have been discovered, identified, and brought into LVM management (as part of the cinder-volumes group), prepare the volume and present it to instances or VMs with Cinder commands on the controller node or the OpenStack dashboard (Horizon).

Basic use cases for Cinder volumes include:

- Carving a volume of desired size and naming it
- Presenting the Cinder volume to an existing instance or VM
- Identifying the Cinder volume on the instance or VM and bringing it into use

Another task Cinder accomplishes is binding new volumes to existing Nova images to create bootable volumes for new instances or VMs. For details, see the *OpenStack End User Guide* available at [OpenStack Documentation](#).

## 7.1.1 Preparing volumes

On the controller node, Cinder commands are issued to the cinder-api and cinder-scheduler to manage Cinder volumes.

1. Create an 8 GB volume with a display name of **8Gb\_Vol**.

```
[stack@queensAIO ~(keystone_admin)]$ cinder create 8 --display_name 8Gb_Vol
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
consistencygroup_id	None
created_at	2018-03-19T18:35:52.000000
description	None
encrypted	False
id	6b1eed54-f6ec-4628-bfee-c13b294eb143
metadata	{}
migration_status	None
multiattach	False
name	8Gb_Vol
os-vol-host-attr:host	None
os-vol-mig-status-attr:migstat	None
os-vol-mig-status-attr:name_id	None
os-vol-tenant-attr:tenant_id	784baabb1b984edeb40938f48648818f
replication_status	None
size	8
snapshot_id	None
source_volid	None
status	creating
updated_at	2018-03-19T18:35:53.000000
user_id	5e6b94871d354c02bda4e8c434ecab62
volume_type	delliscsi

2. List the newly created volume within this project.

```
[stack@queensAIO ~(keystone_admin)]$ cinder list
```

ID	Status	Name	Size	Volume Type	Bootable	Attached to
6b1eed54-f6ec-4628-bfee-c13b294eb143	available	8Gb_Vol	8	delliscsi	false	

## 7.1.2 Presenting volumes to instances or VMs

After a Cinder volume has been created, it can be mapped to an instance or VM. This is accomplished with either the Nova or OpenStack commands from the controller node. Both Nova and OpenStack commands have been provided as examples.

1. Display all virtual machine instances.

```
[stack@queensAIO ~(keystone_admin)]$ nova list
```

ID	Name	Status	Task State	Power State	Networks
44df1b65-b8b5-4740-956b-7fd4abed3fe8	instance_01	ACTIVE	-	Running	public=2001:db8::a, 172.24.4.6

Figure 8 Nova command to display virtual machine instances

```
[stack@queensAIO ~(keystone_admin)]$ openstack server list
```

ID	Name	Status	Networks	Image	Flavor
44df1b65-b8b5-4740-956b-7fd4abed3fe8	instance_01	ACTIVE	public=2001:db8::a, 172.24.4.6		ml.tiny

Figure 9 OpenStack command to display virtual machine instances

2. Map the Cinder volume ID to the Nova instance ID.

```
[stack@queensAIO ~(keystone_admin)]$ nova volume-attach 44df1b65-b8b5-4740-956b-7fd4abed3fe8 6bleed54-f6ec-4628-bfee-c13b294eb143 auto
```

Property	Value
device	/dev/vdb
id	6bleed54-f6ec-4628-bfee-c13b294eb143
serverId	44df1b65-b8b5-4740-956b-7fd4abed3fe8
volumeId	6bleed54-f6ec-4628-bfee-c13b294eb143

Figure 10 Nova command to attach volume to instance

**Note:** The **auto** CLI parameter instructs Cinder to create a new device file for this volume on the instance or VM to the best of its determination.

```
[stack@queensAIO ~(keystone_admin)]$ openstack server add volume 44df1b65-b8b5-4740-956b-7fd4abed3fe8 6bleed54-f6ec-4628-bfee-c13b294eb143 --device /dev/vdb
```

```
[stack@queensAIO ~(keystone_admin)]$ openstack volume list
```

ID	Name	Status	Size	Attached to
1d5216a6-8103-4a64-8379-c2787a260b61		in-use	1	Attached to instance_01 on /dev/vda
6bleed54-f6ec-4628-bfee-c13b294eb143	8Gb_Vol	attaching	8	

```
[stack@queensAIO ~(keystone_admin)]$ openstack volume list
```

ID	Name	Status	Size	Attached to
1d5216a6-8103-4a64-8379-c2787a260b61		in-use	1	Attached to instance_01 on /dev/vda
6bleed54-f6ec-4628-bfee-c13b294eb143	8Gb_Vol	in-use	8	Attached to instance_01 on /dev/vdb

Figure 11 OpenStack command to attach volume to instance

3. Display the new volumes as `/dev/vdX` devices in the output of either an **fdisk -l** or **lsccsi** command.

```
# fdisk -l
```

```
[snip]
```

```
Disk /dev/vdb: 8589 MB, 8589934592 bytes  
16 heads, 63 sectors/track, 16644 cylinders, total 16777216 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk identifier: 0x00000000
```

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

The `/dev/vdX` device can be partitioned, and then a filesystem can be created and mounted accordingly for later use.

## 8 SAN in the cloud

This section outlines some of the considerations within the evolution of software-defined storage, in which the hosting and delivery of redundant, widely scalable storage is becoming progressively decoupled from the underlying hardware platform. That being said, the traditional hardware-based SAN is not going away, but rather will become the foundation for the cloud-based SAN and infrastructure. Traditional hardware-based SAN infrastructure will continue to operate in environments where geographically localized, low-latency, high-bandwidth, mission-critical applications or databases are required.

The use of OpenStack with SC Series storage is best leveraged by attaching the storage to the OpenStack framework through the Cinder module and absorbed into LVM, for use with instances or VMs deployed on geographically dispersed OpenStack compute nodes.

This architecture, when further coupled with Live Volume replication features, allows seamless deployment, management, and migration of instances or VMs using OpenStack within a cloud-based framework. A diagram of this scenario is depicted in Figure 12.

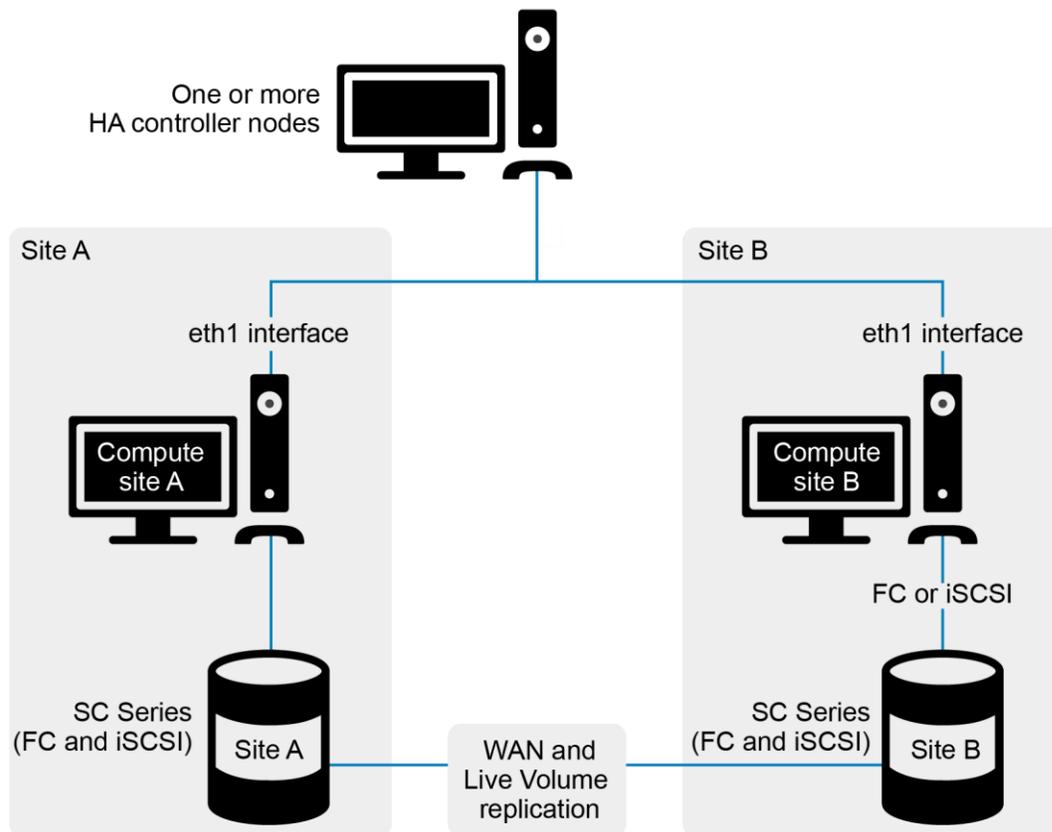


Figure 12 OpenStack using Live Volume replication features

## A Configuration details

Table 1 Component table

Component	Description
OS	Ubuntu 16.04.4 LTS, DevStack with stable/Queens
Driver version	N/A
Firmware version	N/A
Switch	Cisco® MDS 9148
Cabling	Fibre Channel
Server	Dell EMC PowerEdge servers
Storage	Dell EMC SC5020, SCOS 7.2.11.4
Dell Storage Manager	2016 R3.20 build: 16.3.20.18

## B Additional resources

### B.1 Technical support and resources

[Dell.com/support](https://dell.com/support) is focused on meeting customer needs with proven services and support.

[Dell TechCenter](#) is an online technical community where IT professionals have access to numerous resources for Dell software, hardware, and services.

[Storage Solutions Technical Documents](#) on Dell TechCenter provide expertise that helps to ensure customer success on Dell EMC storage platforms.

### B.2 Related documentation

Table 2 lists the referenced or recommended resources related to this document.

Table 2 Referenced or recommended resources

Vendor	Resource
OpenStack	<a href="#">OpenStack Documentation</a>
OpenStack	<a href="#">OpenStack Installation Guide</a>