



# Red Hat OpenStack Platform

## 9

# Dell Storage Center Back End Guide

---

A Guide to Using Dell Storage Center Storage in a Red Hat OpenStack Platform Overcloud

OpenStack Team



# Red Hat OpenStack Platform 9 Dell Storage Center Back End Guide

---

## A Guide to Using Dell Storage Center Storage in a Red Hat OpenStack Platform Overcloud

OpenStack Team  
rhos-docs@redhat.com

## Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document describes how to deploy a single Dell Storage Center device as a back end to the Red Hat OpenStack Platform 9 Overcloud.

---

## Table of Contents

1. INTRODUCTION .....	2
2. PROCESS DESCRIPTION .....	2
3. DEFINE A SINGLE BACK END .....	3
4. DEPLOY THE CONFIGURED BACK END .....	5
5. TEST THE CONFIGURED BACK END .....	6

## 1. INTRODUCTION

This document describes how to configure OpenStack to use one or more Dell Storage Center back ends. The following sections assume that:

- ✦ You intend to use only Dell Storage Center devices and drivers for Block Storage back ends
- ✦ The OpenStack Overcloud has already been deployed through Director, with a properly-functional Block Storage service
- ✦ The Dell storage device has already been deployed and configured as a storage repository
- ✦ You have the necessary credentials for connecting to the Enterprise Manager and Dell Storage Center Group
- ✦ You have the username and password of an account with elevated privileges. You can use the same account that was created to deploy the Overcloud; in [Creating a Director Installation User](#), we create and use the **stack** user for this purpose.

When RHEL OpenStack Platform is deployed through the Director, all major Overcloud settings (in particular, the Block Storage service back end) must be defined and orchestrated through the Director as well. This ensures that the settings will persist through any further Overcloud updates. For more information about deploying OpenStack through the Director, see [Director Installation and Usage](#).

The purpose of this document is to explain how to orchestrate your desired Dell Storage Center back end configuration to the Overcloud's Block Storage service. This document will not discuss the different deployment configurations possible with the back end. Rather, to learn more about the different available deployment configurations, see your device's product documentation.

Once you are familiar with the resulting back end configuration you want to deploy (and its corresponding settings), refer to this document for instructions on how to orchestrate it through the Director.



### Note

At present, the Director only has the integrated components to deploy a **single** instance of a Dell Storage Center back end. As such, this document only describes the deployment of a single back end.

Deploying multiple instances of a Dell Storage Center back end requires a *custom back end configuration*. See the [Custom Block Storage Back End Deployment Guide](#) for instructions.

## 2. PROCESS DESCRIPTION

RHEL OpenStack Platform includes all the drivers required for all Dell devices supported by the Block Storage service. In addition, the Director also has the puppet manifests, environment files, and Orchestration templates necessary for integrating the device as a back end to the Overcloud.

Configuring [a single Dell device as a back end](#) involves editing the default **environment file** and including it in the Overcloud deployment. This file is available locally on the Undercloud, and can be edited to suit your environment.

After editing this file, invoke it through the Director. Doing so ensures that it will persist through future Overcloud updates. The following sections describe this process in greater detail. In addition, the default

environment file already contains enough information to call the necessary puppet manifests and Orchestration (Heat) templates that will configure the rest of the required Block Storage settings.

### 3. DEFINE A SINGLE BACK END

#### Important

This section describes the deployment of a single back end. Deploying multiple instances of a Dell Storage Center back end requires a *custom back end configuration*. See the [Custom Block Storage Back End Deployment Guide](#) for instructions.

With a Director deployment, the easiest way to define a **single** Dell Storage Center back end is through the integrated environment file. This file is located in the following path of the Undercloud node:

```
/usr/share/openstack-tripleo-heat-templates/environments/cinder-dellsc-config.yaml
```

Copy this file to a local path where you can edit and invoke it later. For example, to copy it to `~/templates/`:

```
$ cp /usr/share/openstack-tripleo-heat-templates/environments/cinder-dellsc-config.yaml ~/templates/
```

Afterwards, open the copy (`~/templates/cinder-dellsc-config.yaml`) and edit it as you see fit. The following snippet displays the default contents of this file:

```
# A Heat environment file which can be used to enable a
# a Cinder Dell Storage Center ISCSI backend, configured via puppet
resource_registry:
  OS::TripleO::ControllerExtraConfigPre:
  ../puppet/extraconfig/pre_deploy/controller/cinder-dellsc.yaml # 1

parameter_defaults: # 2
  CinderEnableDellScBackend: true # 3
  CinderDellScBackendName: 'tripleo_dellsc'
  CinderDellScSanIp: ''
  CinderDellScSanLogin: 'Admin'
  CinderDellScSanPassword: ''
  CinderDellScSsn: '64702'
  CinderDellScIscsiIpAddress: ''
  CinderDellScIscsiPort: '3260'
  CinderDellScApiPort: '3033'
  CinderDellScServerFolder: 'dellsc_server'
  CinderDellScVolumeFolder: 'dellsc_volume'
```

1

The **OS::TripleO::ControllerExtraConfigPre:** parameter in the **resource\_registry** section refers to a Heat template named **cinder-dellsc.yaml**. This is the template that the Director should use to load the necessary resources for configuring the back end. By default, the parameter

specifies the path to `cinder-dellsc.yaml` relatively. As such, update this parameter with the absolute path to the file:

```
resource_registry:
  OS::TripleO::ControllerExtraConfigPre: /usr/share/openstack-tripleo-heat-templates/puppet/extraconfig/pre_deploy/controller/cinder-dellsc.yaml
```

2

The `parameter_defaults` section contains your back end definition. Specifically, it contains the parameters that the Director should pass to the resources defined in `cinder-dellsc.yaml`.

3

The `CinderEnableDellScBackend: true` line instructs the Director to use the puppet manifests necessary for the default configuration of a Dell Storage Center back end. This includes defining the volume driver that the Block Storage service should use (specifically, `cinder.volume.drivers.dell_storagecenter_iscsi.DellStorageCenterISCSIDriver`).

To define your Dell Storage Center back end, edit the settings in the `parameter_defaults` section as you see fit. The following table explains each parameter, and also lists its corresponding `/etc/cinder/cinder.conf` setting.

**Table 1. Dell Storage Center settings**

Parameter	<code>/etc/cinder/cinder.conf</code> setting	Description
<code>CinderDellScBackendName</code>	<code>volume_backend_name</code>	<b>(Required)</b> An arbitrary name to identify the volume back end.
<code>CinderDellScSanIp</code>	<code>san_ip</code>	<b>(Optional)</b> The IP address used to reach the Dell Enterprise Manager.
<code>CinderDellScSanLogin</code>	<code>san_login</code>	<b>(Required)</b> The user name to login to the Dell Enterprise Manager at the <code>CinderDellScSanIp</code> . The default user name is <b>Admin</b> .
<code>CinderDellScSanPassword</code>	<code>san_password</code>	<b>(Optional)</b> The corresponding password of <code>CinderDellScSanLogin</code> .



Parameter	/etc/cinder/cinder.conf setting	Description
CinderDellScSsn	dell_sc_ssn	<b>(Required)</b> The Dell Storage Center serial number to use.
CinderDellScIscsiIpAddress	iscsi_ip_address	<b>(Optional)</b> The Dell Storage Center iSCSI IP address to be used for creating volumes and snapshots.
CinderDellScIscsiPort	iscsi_port	<b>(Optional)</b> The iSCSI port of the Dell Storage Center array.
CinderDellScApiPort	dell_sc_api_port	<b>(Optional)</b> The Dell Enterprise Manager API port.
CinderDellScServerFolder	dell_sc_server_folder	<b>(Required)</b> The <b>Server</b> folder in Dell Storage Center where the new server definitions are placed.
CinderDellScVolumeFolder	dell_sc_volume_folder	<b>(Required)</b> The <b>Server</b> folder in Dell Storage Center where the new volumes are created.

## 4. DEPLOY THE CONFIGURED BACK END

The Director installation uses a non-root user to execute commands, which includes orchestrating the deployment of the Block Storage back end. In [Creating a Director Installation User](#), we create a user named **stack** for this purpose. This user is configured with elevated privileges.

To deploy the lone back end configured in [Section 3, “Define a Single Back End”](#), first log in as the **stack** user to the Undercloud. Then, deploy the back end (defined in the edited `~/templates/cinder-dellsc-config.yaml`) by running the following:

```
$ openstack overcloud deploy --templates -e ~/templates/cinder-dellsc-config.yaml
```

### Important

If you passed any extra environment files when you created the Overcloud, pass them again here using the `-e` option to avoid making undesired changes to the Overcloud.

For more information, see [Scaling the Overcloud](#) and [Updating the Overcloud Packages](#).

Once the Director completes the orchestration, test the back end. See [Section 5, “Test the Configured Back End”](#) for instructions.

## 5. TEST THE CONFIGURED BACK END

After deploying the back end, test whether you can successfully create volumes on it. Doing so will require loading the necessary environment variables first. These variables are defined in `/home/stack/overcloudrc` by default.

To load these variables, run the following command as the **stack** user:

```
$ source /home/stack/overcloudrc
```



### Note

For more information, see [Accessing the Basic Overcloud](#).

You should now be logged in to the Controller node. From there, you can create a *volume type*, which can be used to specify the back end you want to use (in this case, the newly-defined back end in [Section 3, “Define a Single Back End”](#)). This is required in an OpenStack deployment where you have other back ends enabled (preferably, also through Director).

To create a volume type named **dellsc**, run:

```
$ cinder type-create dellsc
```

Next, map this volume type to the back end defined in [Section 3, “Define a Single Back End”](#). Given the back end name **tripleo\_dellsc** (as defined through the **CinderDellScBackendName** parameter, in [Section 3, “Define a Single Back End”](#)), run:

```
$ cinder type-key dellsc set volume_backend_name=tripleo_dellsc
```

You should now be able to create a 2GB volume on the newly defined back end by invoking its volume type. To do so, run:

```
$ cinder create --volume-type dellsc 2
```