



Red Hat OpenStack Platform 9 NetApp Block Storage Back End Guide

A guide to using a NetApp appliance as a Block Storage back end in Red Hat OpenStack Platform 9

OpenStack Team

Red Hat OpenStack Platform 9 NetApp Block Storage Back End Guide

A guide to using a NetApp appliance as a Block Storage back end in Red Hat OpenStack Platform 9

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 use Director to deploy a NetApp storage appliance as a back end for the Block Storage service in Red Hat OpenStack Platform 9.

Table of Contents

1. INTRODUCTION	2
2. PROCESS DESCRIPTION	2
3. DEFINE THE BACK END	3
4. DEPLOY THE CONFIGURED BACK END	10
5. TEST THE CONFIGURED BACK END	10

1. INTRODUCTION

This document describes how to use the Director to deploy a NetApp appliance as a back end to the Overcloud's Block Storage service. The following sections assume that:

- You intend to use only NetApp appliance and drivers for Block Storage back ends
- The OpenStack Overcloud has already been deployed through Director
- The NetApp appliance has already been configured and is ready to be used as a storage repository
- You have the necessary credentials for connecting to the NetApp storage system or proxy server
- 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](#), a **stack** user is created for this purpose.

When Red Hat 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.



Note

For manual instructions on configuring the Block Storage service to use a NetApp appliance as a back end, see [Chapter 4. OpenStack Block Storage Service \(from the NetApp OpenStack Deployment and Operations Guide\)](#). Manually-configured Block Storage settings will need to be re-applied during updates to the Overcloud, as the Director will overwrite any settings it did not orchestrate.

This document explains how to orchestrate your desired NetApp back end configuration to the Overcloud's Block Storage service. This document will not discuss the different deployment configurations possible with the NetApp back end. Rather, to learn more about the different available NetApp deployment choices, see [Theory of Operation & Deployment Choices \(from the NetApp OpenStack Deployment and Operations Guide\)](#).

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 NetApp back end. As such, this document only describes the deployment of a single back end.

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

2. PROCESS DESCRIPTION

Red Hat OpenStack Platform includes all the drivers required for all NetApp appliances supported by the Block Storage service. In addition, the Director also has the puppet manifests, environment files, and Orchestration templates necessary for integrating the NetApp appliance as a back end to the Overcloud.

Configuring the NetApp appliance 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.

3. DEFINE THE BACK END

Important

This section describes the deployment of a single back end. Deploying multiple instances of a NetApp 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 the NetApp appliance as a Block Storage back end is through the integrated NetApp environment file. This file is located in the following path of the Undercloud node:

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

Afterwards, open the copy (`~/templates/cinder-netapp-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 NetApp backend, configured via puppet
resource_registry:
  OS::TripleO::ControllerExtraConfigPre:
    ../puppet/extraconfig/pre_deploy/controller/cinder-netapp.yaml # 1

parameter_defaults: # 2
  CinderEnableNetappBackend: true # 3
  CinderNetappBackendName: 'tripleo_netapp'
  CinderNetappLogin: ''
  CinderNetappPassword: ''
  CinderNetappServerHostname: ''
  CinderNetappServerPort: '80'
  CinderNetappSizeMultiplier: '1.2'
  CinderNetappStorageFamily: 'ontap_cluster'
  CinderNetappStorageProtocol: 'nfs'
  CinderNetappTransportType: 'http'
  CinderNetappVfiler: ''
```

```

CinderNetappVolumeList: ''
CinderNetappVserver: ''
CinderNetappPartnerBackendName: ''
CinderNetappNfsShares: ''
CinderNetappNfsSharesConfig: '/etc/cinder/shares.conf'
CinderNetappNfsMountOptions: ''
CinderNetappCopyOffloadToolPath: ''
CinderNetappControllerIps: ''
CinderNetappSaPassword: ''
CinderNetappStoragePools: ''
CinderNetappEseriesHostType: 'linux_dm_mp'
CinderNetappWebservicePath: '/devmgr/v2'

```

1

The **OS::TripleO::ControllerExtraConfigPre:** parameter in the **resource_registry** section refers to a Heat template named **cinder-netapp.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-netapp.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-netapp.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-netapp.yaml**.

3

The **CinderEnableNetappBackend: true** line instructs the Director to use the puppet manifests necessary for the default configuration of a NetApp back end. This includes defining the volume driver that the Block Storage service should use (specifically, **cinder.volume.drivers.netapp.common.NetAppDriver**).

To define your NetApp 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.



Note

For more context on each variable, consult your NetApp appliance's corresponding reference in [Configuration \(from the NetApp OpenStack Deployment and Operations Guide\)](#).

Table 1. NetApp universal back end settings

Parameter	/etc/cinder/cinder.conf setting	Description
CinderNetappBackendName	volume_backend_name	(Required) An arbitrary name to identify the volume back end. The cinder-netapp-config.yaml file uses the name tripleo_netapp by default.
CinderNetappLogin	netapp_login	(Required) Administrative account name used to access the back end or its proxy server. For this parameter, you can use an account with cluster-level administrative permissions (namely, admin) or a cluster-scoped account ^[a] with the appropriate privileges.
CinderNetappPassword	netapp_password	(Required) The corresponding password of CinderNetappLogin .
CinderNetappServerHostname	netapp_server_hostname	(Required) The storage system or proxy server (for E-Series). The value of this option should be the IP address or hostname of either the cluster management logical interface (LIF) or Storage Virtual Machine (SVM) LIF.
CinderNetappServerPort	netapp_server_port	(Optional) The TCP port that the Block Storage service should use to communicate with the NetApp back end. If not specified, Data ONTAP drivers will use 80 for HTTP and 443 for HTTPS ; E-Series will use 8080 for HTTP and 8443 for HTTPS .

Parameter	<i>/etc/cinder/cinder.conf</i> setting	Description
CinderNetappSizeMultiplier	netapp_size_multiplier	(Deprecated) During volume creation, the quantity to be multiplied to the requested volume size to ensure that the NetApp back end has enough space.
CinderNetappStorageFamily	netapp_storage_family	(Optional) The storage family type used on the back end device. Use ontap_cluster for clustered Data ONTAP, ontap_7mode for Data ONTAP operating in 7-Mode, or eseries for E-Series.
CinderNetappStorageProtocol	netapp_storage_protocol	(Required) The storage protocol to be used. Use either nfs , iscsi , or fc .
CinderNetappTransportType	netapp_transport_type	(Required) Transport protocol to be used for communicating with the back end. Valid options include http and https .
[a] For more information on cluster-scoped accounts, see Data ONTAP Configuration (from the NetApp OpenStack Deployment and Operations Guide)		

The following setting is only valid for clustered Data ONTAP (as in, with **CinderNetappStorageFamily** set to **ontap_cluster**).

Table 2. NetApp settings for clustered DATA ONTAP

Parameter	<i>/etc/cinder/cinder.conf</i> setting	Description
CinderNetappVserver	netapp_vserver	(Required) Specifies which the name of the SVM where volume provisioning should occur. This refers to a single SVM on the storage cluster.

The following settings are only valid with Data ONTAP operating in 7-Mode (as in, with **CinderNetappStorageFamily** set to **ontap_7mode**).

Table 3. NetApp settings for DATA ONTAP operating in 7-Mode

Parameter	<i>/etc/cinder/cinder.conf</i> setting	Description
CinderNetappVfiler	netapp_vfiler	(Optional) The vFiler unit on which provisioning of volumes will be done. Use this option only when you want to use the MultiStore feature on the NetApp back end.
CinderNetappVolumeList	netapp_volume_list	(Deprecated) Restricts provisioning to the specified comma-separated list of NetApp controller volumes. Backwards compatibility for this option remains for this release.
CinderNetappPartnerBackendName ^[a]	netapp_partner_backend_name	(Required) This specifies another back end that acts as the second half of a high-availability (HA) pair. Both back ends must refer to each other's volume_backend_name in their respective back end definitions.
<p>[a] This option is only valid when using the Fibre Channel protocol (as in, with CinderNetappStorageProtocol set to fc). For more details, see NetApp Unified Driver for Data ONTAP operating in 7-Mode with Fibre Channel (from the NetApp OpenStack Deployment and Operations Guide)</p>		

The following settings are only valid with the E-Series family of devices (as in, with **CinderNetappStorageFamily** set to **eseries**).

Table 4. NetApp settings for E-Series

Parameter	<i>/etc/cinder/cinder.conf</i> setting	Description
-----------	--	-------------

Parameter	/etc/cinder/cinder.conf setting	Description
CinderNetappControllerIps	netapp_controller_ip	(Required) A comma-separated list of controller management IPs/hostnames to which provisioning should be restricted.
CinderNetappSaPassword	netapp_sa_password	(Optional) Password to the NetApp E-Series storage array.
CinderNetappStoragePools	netapp_storage_pools	(Removed) A comma-separated list of disk pools to which provisioning should be restricted. Do not edit this parameter, as it now refers to an unavailable driver option.
CinderNetappEseriesHostType	netapp_eseries_host_type	(Removed) Defines the type of operating system for all initiators that can access a LUN. This information is used when mapping LUNs to individual hosts or groups of hosts. Do not edit this parameter, as it now refers to an unavailable driver option.
CinderNetappWebservicePath	netapp_webservice_path	(Optional) Specifies the path to the E-Series proxy application on a proxy server. To determine the full URL for connecting to the proxy application, the driver combines the CinderNetappTransportType , CinderNetappServerHostname , and CinderNetappServerPort port values.

The following settings are only valid when using the NFS protocol (as in, with **CinderNetappStorageProtocol** set to **nfs**). For more information, see [NetApp Unified Driver for Clustered Data ONTAP with NFS](#) or [NetApp Unified Driver for Data ONTAP operating in 7-Mode with NFS](#) (both from the [NetApp OpenStack Deployment and Operations Guide](#)).

Table 5. NetApp settings for NFS

Parameter	/etc/cinder/cinder.conf setting	Description
CinderNetappNfsShares	None	<p>(Required) Comma-separated list of Data LIFs exported from the NetApp ONTAP device to be mounted by the Controller nodes. This list gets written to the location defined by CinderNetappNfsSharesConfig. For example:</p> <pre>CinderNetappNfsShares : \ '192.168.67.1:/cinder1,192.168.67.2:/cinder2,192.168.67.2:/cinder3,192.168.67.2:/archived_data'</pre>
CinderNetappNfsSharesConfig	nfs_shares_config	<p>(Required) Absolute path to the <i>NFS exports file</i>. This file contains a list of available NFS shares to be used as a back end.</p>
CinderNetappNfsMountOptions	nfs_mount_options	<p>(Optional) Comma-separated list of mount options you want to pass to the NFS client. For more information about valid options, see man mount.</p>
CinderNetappCopyOffloadToolPath ^[a]	netapp_copyoffload_tool_path	<p>(Optional) Specifies the path of the NetApp copy offload tool binary. This binary (available from the NetApp Support portal) must have the Execute permissions set, as the openstack-cinder-volume process will need to execute this file.</p>
<p>[a] This option is only valid with Clustered Data ONTAP (as in, with CinderNetappStorageFamily set to ontap_cluster). For more information, see NetApp Unified Driver for Clustered Data ONTAP with NFS (from the NetApp OpenStack Deployment and Operations Guide).</p>		

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](#), a user named **stack** is created for this purpose. This user is configured with elevated privileges.

Log in as the **stack** user to the Undercloud. Then, deploy the NetApp back end (defined in the edited `~/templates/cinder-netapp-config.yaml`) by running the following:

```
$ openstack overcloud deploy --templates -e ~/templates/cinder-netapp-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 the 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 `netapp`, run:

```
$ cinder type-create netapp
```

Next, map this volume type to the back end defined in [Section 3, “Define the Back End”](#). Given the back end name `tripleo_netapp` (as defined through the `CinderNetappBackendName` parameter, in [Section 3, “Define the Back End”](#)), run:

```
$ cinder type-key netapp set volume_backend_name=tripleo_netapp
```

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 netapp 2
```