



# **Red Hat OpenStack Platform 10 NetApp Back End Guide for the Shared File System Service**

---

Deploying Multiple NetApp Back Ends for the Shared File System Service  
in a Red Hat OpenStack Platform Overcloud

OpenStack Team



# Red Hat OpenStack Platform 10 NetApp Back End Guide for the Shared File System Service

---

## Deploying Multiple NetApp Back Ends for the Shared File System Service 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 configure and deploy the OpenStack Shared File System Service using a NetApp storage controller (running Data ONTAP) as a back end. The scenario described herein uses the ``manila.share.drivers.netapp.common.NetAppDriver`` in a custom environment file to enable two NetApp back ends.

## Table of Contents

<b>1. INTRODUCTION</b>	<b>2</b>
<b>2. REQUIREMENTS</b>	<b>2</b>
<b>3. CREATE THE ENVIRONMENT FILE</b>	<b>2</b>
<b>4. DEPLOY THE SHARED FILE SYSTEM SERVICE WITH NETAPP BACK ENDS</b>	<b>4</b>
<b>5. CREATE A BASIC SHARE TYPE</b>	<b>4</b>
<b>A. CONFIGURING A DEFAULT SHARE TYPE (OPTIONAL)</b>	<b>5</b>

## 1. INTRODUCTION

The OpenStack File Share Service (**openstack-manila**) provides the means to easily provision shared file systems that can be consumed by multiple instances. In the past, OpenStack users needed to manually deploy shared file systems before mounting them on instances. The OpenStack File Share Service, on the other hand, allows users to easily provision shares from a pre-configured storage pool, ready to be mounted securely. This pool, in turn, can be independently managed and scaled to meet demand.

This release supports the use of the NetApp unified driver (`manila.share.drivers.netapp.common.NetAppDriver`). This driver allows the Shared File System service to use NetApp storage controllers (running Data ONTAP) as a back end.

While you can manually configure the Shared File System service by directly editing its node's **/etc/manila/manila.conf** file, any settings can be overwritten by the Red Hat OpenStack Platform director in future overcloud updates. As such, the recommended method for configuring a Shared File System back end is through the director. Doing so involves writing a *custom environment file*.

With this release, the director can now deploy the Shared File System with a NetApp back end on the overcloud. This document explains how to do so.

## 2. REQUIREMENTS

The following sections assume that:

- ✱ A NetApp storage controller has already been deployed, and is ready to be used as a back end.
- ✱ You intend to only use one NetApp storage controller as a back end for your Shared File System service.
- ✱ You can use the director installation user account, which is created as part of the overcloud deployment. See [Creating a Director Installation User](#) (from [Director Installation and Usage](#)) for more information.
- ✱ The Shared File System service will still be installed on the Controller nodes, as is the default behavior.

This document will not discuss the different deployment configurations possible for your NetApp back end. Rather, to learn more about possible NetApp storage deployment configurations suitable for the Shared File System service, consult the upstream [NetApp documentation](#) (in particular, see [Theory of Operation and Deployment Choices](#)).

After mapping out your target configuration (as in, the settings you want for each NetApp back end), you can then translate your configuration to a custom environment file. The director will then use this file to orchestrate the configuration of your back ends, and make them persistent across overcloud updates.

## 3. CREATE THE ENVIRONMENT FILE

The director already includes Heat templates to configure most of the necessary settings to integrate a NetApp back end. An *environment file* allows you to define settings specific to your deployment.

To start, log in as the *stack* user on the undercloud. Then, create an environment file with the following contents:

```
/home/stack/templates/netapp-config.yaml
```

```
parameter_defaults:
  ManilaNetappLogin: 'ADMIN' # 1
  ManilaNetappPassword: 'ADMINPW'
  ManilaNetappServerHostname: 'HOSTNAME' # 2
  ManilaNetappVserver: 'SVM' # 3
  ManilaNetappRootVolumeAggr: 'ROOTVAGGR' # 4
  ManilaNetappTraceFlags: 'TRFLAGS' # 5
  ManilaNetappDriverHandlesShareServers: 'false' # 6
```

1

Replace *ADMIN* and *ADMINPW* with the credentials of the administrative account used to access the storage system (specifically, *HOSTNAME*).

2

Replace *HOSTNAME* with the storage system or proxy server. 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.

3

*SVM* specifies the storage virtual machine (previously called a *vserver*) name on the storage cluster on which provisioning of shared file systems should occur. This parameter is required if the driver should operate without managing share servers (that is, be limited to the scope of a single SVM).

4

*ROOTVAGGR* specifies the name of the aggregate upon which the root volume should be placed when a new storage virtual machine (SVM) is created to correspond to a Manila share server.

5

Replace *TRFLAGS* with a comma-separated list of options that control which trace info is written to the Shared File System service logs when the debug level is set to **True**. Supported values include **method** and **api**.

6

The **ManilaNetappDriverHandlesShareServers** parameter sets whether the driver should handle the lifecycle of the share server (**false** means it should not).

For example:

**/home/stack/templates/netapp-config.yaml**

■

```
parameter_defaults:
  ManilaNetappLogin: 'admin'
  ManilaNetappPassword: 'netapp123'
  ManilaNetappServerHostname: '10.8.18.108'
  ManilaNetappVserver: 'vserver_1'
  ManilaNetappRootVolumeAggr: 'aggr1_n1'
  ManilaNetappTraceFlags: 'method,api'
  ManilaNetappDriverHandlesShareServers: 'false'
```

The next section describes how to use the `/home/stack/templates/netapp-config.yaml` environment file to orchestrate the configuration of your NetApp back end.

## 4. DEPLOY THE SHARED FILE SYSTEM SERVICE WITH NETAPP BACK ENDS

Once you create `/home/stack/templates/netapp-config.yaml`, log in as the **stack** user on the undercloud. Then, deploy the configured back end by running:

```
$ source ~/stackrc
$ openstack overcloud deploy --templates -e /usr/share/openstack-tripleo-heat-
templates/environments/manila-netapp-config.yaml -e
/home/stack/templates/netapp-config.yaml
```

The `/usr/share/openstack-tripleo-heat-templates/environments/manila-netapp-config.yaml` used here is the environment file provided with the director for deploying NetApp back ends for the Shared File System service. The `/home/stack/templates/netapp-config.yaml` file we created in the previous section allows you to override the default settings to suit your deployment.

### 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 [Modifying the Overcloud Environment](#) (from [Director Installation and Usage](#)).

## 5. CREATE A BASIC SHARE TYPE

Whenever you create a new share, you need to specify a *share type*. If you don't specify one, the share creation will fail.

To create a basic share type named **default**, run the following as the **stack** user on the undercloud:

```
$ source ~/overcloudrc
$ manila type-create default false
```

Here, we specify **false** since there is no need for the NetApp driver to handle the lifecycle of share servers. This is because we set **ManilaNetappDriverHandlesShareServers** to **false** in [Section 3, “Create the Environment File”](#). Otherwise, if **ManilaNetappDriverHandlesShareServers** is set to **true** you can match the default share type to this. For more information about share types, see [Create and Manage Shares](#) (from the [Red Hat OpenStack Platform Storage Guide](#)).



## A. CONFIGURING A DEFAULT SHARE TYPE (OPTIONAL)

At present, the director does not automatically specify a *default share type* for the Shared File System service. This is why you need to specify a share type every time you create a share.

You can, however, manually define a *default share type*. To do so, open the `/etc/manila/manila.conf` after the Shared File System service is deployed. Then, add the following line to the **[DEFAULT]** section:

```
default_share_type = default
```

If the Shared File System service is deployed on multiple nodes, apply this same change to the `/etc/manila/manila.conf` of each of those nodes. Then, restart the service by performing the following steps:

1. Log in to any Controller node as heat-admin.
2. Run the following command:

```
$ sudo systemctl restart *manila*
```