Red Hat OpenStack Platform 10

CephFS Back End Guide for the Shared File System Service

Deploying a CephFS Back End for the Shared File System Service in a Red Hat OpenStack Platform Overcloud

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

Deploying a CephFS Back End for the Shared File System Service in a Red Hat OpenStack Platform Overcloud

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

This document describes a test deployment of OpenStack Shared File System Service that uses a CephFS back end. The back end used by this deployment is enabled through the manila.share.drivers.cephfs.cephfs_native driver. Red Hat CephFS integration is currently available as a Technology Preview, and as such this deployment should not be used in production. For more information about Technology Previews, see:

Table of Contents

1. INTRODUCTION

IMPORTANT

Red Hat Ceph file system (CephFS) is available only as a *Technology Preview*, and therefore not fully supported by Red Hat. The deployment scenario described in this document should only be used for testing, and should not be deployed in a production environment.

For more information about Technology Preview features, see Scope of Coverage Details.

The OpenStack Shared File Systems 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 Shared File Systems 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 includes a technology preview of the necessary driver for Red Hat CephFS (namely, **manila.share.drivers.cephfs.cephfs_native.CephFSNativeDriver**). This driver allows the Shared File System service to use CephFS 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 CephFS back end on the overcloud. This document explains how to do so.

2. REQUIREMENTS

The test scenario described in this document requires the following:

1. A Red Hat Ceph Storage cluster, with at least one Metadata Server (MDS) running. See the Red Hat Ceph Storage Administration Guide for more information.

2. A Red Hat Ceph file system (CephFS) must already exist on the Red Hat Storage cluster. This file system is available as a Technology Preview on Red Hat Ceph 2.0. See Red Hat Ceph File System Guide (Technology Preview) for instructions.

    NOTE

    The Red Hat Ceph features and packages required for this test scenario are available in Red Hat Ceph 2.0. See the Red Hat Ceph Storage Release Notes for subscription details.

3. The Red Hat Ceph Storage cluster hosting CephFS must already be integrated into the OpenStack deployment. See Integrating an Existing Ceph Storage Cluster with an Overcloud (from the Red Hat Ceph Storage for the Overcloud guide) for information.

4. Access to the *director installation user account*, which is created as part of the overcloud deployment. This account is used to deploy the overcloud. See Creating a Director Installation User (from Director Installation and Usage) for more information.

In addition, this scenario assumes that:

- The Shared File System service will still be installed on the Controller nodes, as is the default behavior; and

- You intend to only use a single instance of the Ceph File Sysem as the only back end for your Shared File System Service.

Finally, you also need to integrate the existing Red Hat Ceph Storage cluster with the overcloud. This requires a separate environment file; the director provides a pre-made one at **/usr/share/openstack-tripleo-heat-templates/environments/puppet-ceph-external.yaml**. For more information on how to configure this, see Integrating an Existing Ceph Storage Cluster with an Overcloud (from the Red Hat Ceph Storage for the Overcloud guide).

2.1. Limitations and Restrictions

Given the current state of the involved components, the test scenario in this document has the following limitations and restrictions:

1. Untrusted instance users pose a security risk to the Ceph Storage cluster, as they would have direct access to the public network of the Ceph Storage cluster. Ensure that the cluster you are using is quarantined from the production environment, and that only trusted users have access to the test environment.

2. This release only allows **read-write** access to shares.

3. CONFIGURE ACCESS TO THE CEPH CLUSTER

To integrate an existing CephFS back end to your Shared File System service, you must first retrieve its *pool name* and *metadata pool name*. Log in to any MON node within the Ceph cluster and run:

```
# ceph fs ls
```

This will provide information you will need later on in Section 4, "Edit the Environment File". For example:

```
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data]
```

Afterwards, perform the following steps to provide the Shared File System service with access to the CephFS share:

1. Create a *client* that the Shared File System service can use to access the CephFS share. Run the following on the MON node to create a client named **client.manila**:

```
$ ceph auth add client.manila mds 'allow *' \
  mon 'allow r, allow command="auth del", \
    allow command="auth caps", \
    allow command="auth get", \
    allow command="auth get-or-create"' \
  osd 'allow class-read object_prefix rbd_children, \
    allow rwx pool=cephfs_data, allow rwx pool=cephfs_metadata'
```

2. Use the **ceph auth list** command to retrieve the details of **client.manila**. In particular, note the **key** value (*CLIENT_KEY*):

   ```
   $ ceph auth list | grep -A4 manila

   installed auth entries:

   client.manila
    key: CLIENT_KEY
    caps: [mds] allow *
    caps: [mon] allow r, allow command="auth del", allow command="auth caps", allow
   command="auth get", allow command="auth get-or-create"
    caps: [osd] allow class-read object_prefix rbd_children, allow rwx pool=cephfs_data, allow rwx
   pool=cephfs_metadata
   ```

   The *CLIENT_KEY* is a long string of alphanumeric characters, such as **324a54+5322b6b312==a4f7a6c4d3e775d4839AA**.

3. Log in to any node hosting the Shared File System service and create the following file: **/etc/ceph/ceph.client.manila.keyring**

   This will be the *keyring* file, which provides the Shared File System with access to the CephFS share.

   > **NOTE**
   >
   > By default, the director installs the Shared File System on the Controller nodes.

4. Populate the keyring file with the details of **client.manila**:

   ```
   [client.manila]
     key = CLIENT_KEY
     caps mds = "allow *"
     caps mon = "allow r, allow command=\"auth del\", allow command=\"auth caps\", allow
   command=\"auth get\", allow command=\"auth get-or-create\""
     caps osd = "allow class-read object_prefix rbd_children, allow rwx pool=cephfs_data, allow rwx
   pool=cephfs_metadata"
   ```
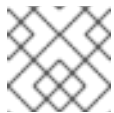
5. Change the permissions of this file to secure it from unauthorized access:

   ```
   $ chmod 600 /etc/ceph/ceph.client.manila.keyring
   ```

6. Copy this file to all other nodes hosting the Shared File System (for example, all other Controller nodes in the overcloud).

7. Determine which node is *running* the Shared File System service. Run:

   ```
   $ pcs status
   [...]
     openstack-manila-share (systemd:openstack-manila-share): Started overcloud-controller-2
   ```

8. Log in to the node running the Shared File System service (in this case, **overcloud-controller-2**).

9. Restart the Shared File System service:

```
$ sudo systemctl restart openstack-manila-service
```

4. EDIT THE ENVIRONMENT FILE

The environment file contains the back end settings you want to define. It also contains other settings relevant to the deployment of the Shared File System service. For more information about environment files, see Environment Files (from the Director Installation and Usage guide).

This release includes an integrated environment file for defining a CephFS back end. This file is located in the following path of the undercloud node:

/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml

This file provides default settings for deploying a Shared File System service.

Create an environment file which will contain the settings necessary for your environment — namely, **~/templates/manila-cephfsnative-config.yaml**. The following snippet shows the default values used by the director when deploying the Shared File System service:

/home/stack/templates/manila-cephfsnative-config.yaml

```
parameter_defaults: #  1
  ManilaCephFSNativeEnableBackend: true
  ManilaCephFSNativeBackendName: cephfsnative
  ManilaCephFSNativeDriverHandlesShareServers: false #  2
  ManilaCephFSNativeCephFSConfPath: '/etc/ceph/cephfs.conf' #  3
  ManilaCephFSNativeCephFSAuthId: 'manila' #  4
  ManilaCephFSNativeCephFSClusterName: 'ceph'
  ManilaCephFSNativeCephFSEnableSnapshots: false
```

1 The **parameter_defaults** header signifies the start of your configuration. Specifically, it allows you to override default values set in **resource_registry**. This includes values set by **OS::Tripleo::Services::ManilaBackendCephFs**, which sets defaults for a CephFS back end.

2 With **ManilaCephFSNativeDriverHandlesShareServers** set to **false**, the driver will not handle the lifecycle of the share server.

3 **ManilaCephFSNativeCephFSConfPath:** sets the path to the configuration file of the Ceph cluster.

4 **ManilaCephFSNativeCephFSAuthId:** is the Ceph auth ID that the director will create for share access.

Add the *pool name*, *metadata pool name*, and *key* you retrieved earlier in Section 3, "Configure Access to the Ceph Cluster" to **/home/stack/templates/manila-cephfsnative-config.yaml**:

```
ManilaCephFSDataPoolName: cephfs_data
ManilaCephFSMetadataPoolName: cephfs_metadata
CephManilaClientKey: CLIENT_KEY
```

5. DEPLOY THE SHARED FILE SYSTEM SERVICE WITH A CEPHFS BACK END

Once you create **/home/stack/templates/manila-cephfsnative-config.yaml**, log in as the **stack** user on the undercloud. Then, deploy the Shared File System service with a CephFS back end by including the following environment files:

- **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml** – deploys the Ceph cluster.

- **/home/stack/templates/manila-cephfsnative-config.yaml** – created earlier in Section 4, "Edit the Environment File", and contains any settings to override defaults set in **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml**.

For example, if your OpenStack and Ceph settings are defined in **/home/stack/templates/ceph-external.yaml** (as in Integrating with the Existing Ceph Storage Cluster from Red Hat Ceph Storage for the Overcloud), run:

```
$ openstack overcloud deploy --templates \
  -e /home/stack/templates/ceph-external.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml \
  -e /home/stack/templates/manila-cephfsnative-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 Modifying the Overcloud Environment (from Director Installation and Usage).