NEC Cinder Backend Deployment Guide for Red Hat OpenStack Platform 17

2024/10

NEC Corporation

First Edition

1. Overview

This document describes how to deploy NEC Block Storage services in a Red Hat OpenStack Platform Overcloud. This assumes that the RHOSP installation is through RHOSP director toolset which is based primarily on the upstream TripleO project.

- This mainly covers NEC storage backend like
  - NEC Storage M series driver

2. Prerequisites

NEC Storage Backend configured as storage repository.

3. Limitations

- Extending volumes with snapshots
  NEC Storage does not support extension of a logical disks which has one or more snapshot volumes. The Cinder driver returns an error when performing an expansion operation on a volume which has snapshots.
  To extend a volume with snapshots, all the snapshots must be deleted before extension. You can use Create Volume from Snapshot to make backup of snapshot data. The procedure is as follows.
  1. Create a new volume from each snapshot which data need to be preserved.
  2. Delete all snapshots.
  3. Expand the capacity of the volume.
- Creating the encrypted volume
  NEC Storage does not support creating the encrypted volumes.
- Showing the volume with snapshots and the snapshots in the Manageable-list
  NEC storage does not support showing the volume with snapshots and the snapshots in the Manageable-list.

4. Deployment Steps

4.1. Prepare the Environment File

The environment file is a RHOSP director environment file. The environment file contains the settings for each back end you want to define. Using the environment file will ensure that the back end settings persist through future Overcloud updates and upgrades.

Create the environment file that will orchestrate the back end settings. Use the sample file provided below for your specific backend.

Note: LVM driver is enabled by default in TripleO, you want to set the CinderEnableIscsiBackend to false in one of your environment files to turn it off.

4.1.1. NEC FC and iSCSI drivers

For full detailed instruction of all options please refer to NEC Storage M series driver Configuration.

Environment sample

With a director deployment, backend can be deployed using the heat template. Create the template file in a local path where you can edit and invoke it later. For example, to create it in ~/templates/:

```
$ vi ~/templates/nec-storage-cinder.yaml
```

Afterwards, edit it as you see fit. The following shows a sample content of the file.

FC Environment sample

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  ControllerExtraConfig:
    cinder::config::cinder_config:
      Storage1/volume_driver:
        value: cinder.volume.drivers.nec.volume.MStorageFCDriver
      Storage1/volume_backend_name:
        value: Storage1
      Storage1/nec_ismcli_fip:
        value: 192.168.100.120
      Storage1/nec_ismcli_user:
        value: cliusername
      Storage1/nec_ismcli_password:
        value: clipassword
      Storage1/nec_pools:
        value: 0,1
      Storage1/nec_backup_pools:
        value: 2
      Storage1/nec_diskarray_name:
        value: NECstorage1
    cinder_user_enabled_backends: ['Storage1']
```

iSCSI Environment sample

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  ControllerExtraConfig:
    cinder::config::cinder_config:
      Storage1/volume_driver:
        value: cinder.volume.drivers.nec.volume.MStorageISCSIDriver
      Storage1/volume_backend_name:
        value: Storage1
      Storage1/nec_ismcli_fip:
        value: 192.168.100.120
      Storage1/nec_ismcli_user:
        value: cliusername
      Storage1/nec_ismcli_password:
        value: clipassword
      Storage1/nec_pools:
        value: 0,1
      Storage1/nec_backup_pools:
        value: 2
      Storage1/nec_diskarray_name:
        value: NECstorage1
    cinder_user_enabled_backends: ['Storage1']
```

4.2. Deploy the configured backend

When you have created the file nec-storage-cinder.yaml file with appropriate backend, deploy the backend configuration by running the openstack overcloud deploy command using the --templates option and the -e option with the created file. If you passed any extra environment files when you created the overcloud, pass them again here using the -e option.

```
(undercloud) $ openstack overcloud deploy --templates ¥
-e /home/stack/templates/overcloud_images.yaml ¥
-e <other templates> ¥
.....
-e /home/stack/templates/nec-storage-cinder.yaml
```

4.3. Multiple backend Deployment

Multiple backend can be deployed at once (same kind or different kinds), the following sample environment file defines two NEC back ends, Storage1 using iSCSI driver and Storage2 using the FC driver:

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  ControllerExtraConfig:
    cinder::config::cinder_config:
      Storage1/volume_driver:
        value: cinder.volume.drivers.nec.volume.MStorageISCSIDriver
      Storage1/volume_backend_name:
        value: Storage1
      Storage1/nec_ismcli_fip:
        value: 192.168.100.120
      Storage1/nec_ismcli_user:
        value: cliusername
      Storage1/nec_ismcli_password:
        value: clipassword
      Storage1/nec_pools:
        value: 0,1
      Storage1/nec_backup_pools:
        value: 2
      Storage1/nec_diskarray_name:
        value: NECstorage1
      Storage2/volume_driver:
        value: cinder.volume.drivers.nec.volume.MStorageFCDriver
      Storage2/volume_backend_name:
        value: Storage2
      Storage2/nec_ismcli_fip:
        value: 192.168.100.120
      Storage2/nec_ismcli_user:
        value: cliusername
      Storage2/nec_ismcli_password:
        value: clipassword
      Storage2/nec_pools:
        value: 4,5
      Storage2/nec_backup_pools:
        value: 6
      Storage2/nec_diskarray_name:
        value: NECstorage1
    cinder_user_enabled_backends: ['Storage1', 'Storage2']
```

4.4. Verify the configured changes

When the director completes the overcloud deployment, check that the volume services are up using the openstack cli command. You can also verify that the cinder.conf in the cinder container and it should reflect changes made above.

```
$openstack volume service list
```

4.5. Testing the configured Backend

After you deploy the back ends to the overcloud, create a volume-type per backend and test if you can successfully create and attach volumes of that type.

5. References

- Deploying a custom Block Storage back end | Red Hat Product Documentation