Red Hat OpenStack Platform 17.1

Deploying a hyperconverged infrastructure

Understand and configure Hyperconverged Infrastructure on the Red Hat OpenStack Platform overcloud

Last Updated: 2024-05-23

Red Hat OpenStack Platform 17.1 Deploying a hyperconverged infrastructure

Understand and configure Hyperconverged Infrastructure on the Red Hat OpenStack Platform overcloud

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

This document describes the Red Hat OpenStack Platform implementation of hyperconvergence, which colocates Compute and Ceph Storage services on the same host.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Providing documentation feedback in Jira

Use the Create Issue form to provide feedback on the documentation. The Jira issue will be created in the Red Hat OpenStack Platform Jira project, where you can track the progress of your feedback.

1. Ensure that you are logged in to Jira. If you do not have a Jira account, create an account to submit feedback.

2. Click the following link to open a the **Create Issue** page: Create Issue

3. Complete the **Summary** and **Description** fields. In the **Description** field, include the documentation URL, chapter or section number, and a detailed description of the issue. Do not modify any other fields in the form.

4. Click **Create**.

CHAPTER 1. CONFIGURING AND DEPLOYING A RED HAT OPENSTACK PLATFORM HYPERCONVERGED INFRASTRUCTURE

1.1. HYPERCONVERGED INFRASTRUCTURE OVERVIEW

Red Hat OpenStack Platform (RHOSP) hyperconverged infrastructures (HCI) consist of hyperconverged nodes. In RHOSP HCI, the Compute and Storage services are colocated on these hyperconverged nodes for optimized resource use. You can deploy an overcloud with only hyperconverged nodes, or a mixture of hyperconverged nodes with normal Compute and Red Hat Ceph Storage nodes.

NOTE

You must use Red Hat Ceph Storage as the storage provider.

TIP

Use BlueStore as the back end for HCI deployments to make use of the BlueStore memory handling features.

Hyperconverged infrastructures are built using a variation of the deployment process described in Deploying Red Hat Ceph and OpenStack together with director . In this deployment scenario, RHOSP director deploys your cloud environment, which director calls the overcloud, and Red Hat Ceph Storage. You manage and scale the Ceph cluster itself separate from the overcloud configuration.

IMPORTANT

Instance HA is not supported on RHOSP HCI environments. To use Instance HA in your RHOSP HCI environment, you must designate a subset of the Compute nodes with the ComputeInstanceHA role to use the Instance HA. Red Hat Ceph Storage services must not be hosted on the Compute nodes that host Instance HA.

Red Hat OpenStack Platform 17.1 only supports Red Hat Ceph Storage 6 for new deployments. Red Hat Ceph Storage 5 is not supported in new deployment scenarios.

IMPORTANT

All HCI nodes in supported Hyperconverged Infrastructure environments must use the same version of Red Hat Enterprise Linux as the version used by the Red Hat OpenStack Platform controllers. If you wish to use multiple Red Hat Enterprise versions in a hybrid state on HCI nodes in the same Hyperconverged Infrastructure environment, contact the Red Hat Customer Experience and Engagement team to discuss a support exception.

For HCI configuration guidance, see Configuration guidance.

CHAPTER 2. DEPLOYING HCI HARDWARE

This section contains procedures and information about the preparation and configuration of hyperconverged nodes.

Prerequisites

- You have read Deploying an overcloud and Red Hat Ceph Storage in *Deploying Red Hat Ceph and OpenStack together with director*.

2.1. CLEANING CEPH STORAGE NODE DISKS

Ceph Storage OSDs and journal partitions require factory clean disks. All data and metadata must be erased by the Bare Metal Provisioning service (ironic) from these disks before installing the Ceph OSD services.

You can configure director to delete all disk data and metadata by default by using the Bare Metal Provisioning service. When director is configured to perform this task, the Bare Metal Provisioning service performs an additional step to boot the nodes each time a node is set to **available**.

> **WARNING**
>
> The Bare Metal Provisioning service uses the **wipefs --force --all** command. This command deletes all data and metadata on the disk but it does not perform a secure erase. A secure erase takes much longer.

Procedure

1. Open /**home/stack/undercloud.conf** and add the following parameter:

```
clean_nodes=true
```

2. Save /**home/stack/undercloud.conf**.

3. Update the undercloud configuration.

```
openstack undercloud install
```

2.2. REGISTERING NODES

Register the nodes to enable communication with director.

Procedure

1. Create a node inventory JSON file in /**home/stack**.

2. Enter hardware and power management details for each node. For example:

```
{
  "nodes":[
    {
      "mac":[
        "b1:b1:b1:b1:b1:b1"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.205"
    },
    {
      "mac":[
        "b2:b2:b2:b2:b2:b2"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.206"
    },
    {
      "mac":[
        "b3:b3:b3:b3:b3:b3"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.207"
    },
    {
      "mac":[
        "c1:c1:c1:c1:c1:c1"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"ipmi",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.208"
    },
    {
      "mac":[
            "c2:c2:c2:c2:c2:c2"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.209"
    },
    {
        "mac":[
            "c3:c3:c3:c3:c3:c3"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.210"
    },
    {
        "mac":[
            "d1:d1:d1:d1:d1:d1"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.211"
    },
    {
        "mac":[
            "d2:d2:d2:d2:d2:d2"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.0.2.212"
    },
    {
        "mac":[
            "d3:d3:d3:d3:d3:d3"
        ],
        "cpu":"4",
        "memory":"6144",
            "disk":"40",
            "arch":"x86_64",
            "pm_type":"ipmi",
            "pm_user":"admin",
            "pm_password":"p@55w0rd!",
            "pm_addr":"192.0.2.213"
        }
    ]
}
```

3. Save the new file.

4. Initialize the stack user:

```
$ source ~/stackrc
```

5. Import the JSON inventory file into director and register nodes

```
$ openstack overcloud node import <inventory_file>
```

   Replace **<inventory_file>** with the name of the file created in the first step.

6. Assign the kernel and ramdisk images to each node:

```
$ openstack overcloud node configure <node>
```

2.3. VERIFYING AVAILABLE RED HAT CEPH STORAGE PACKAGES

Verify all required packages are available to avoid overcloud deployment failures.

2.3.1. Verifying cephadm package installation

Verify the **cephadm** package is installed on at least one overcloud node. The **cephadm** package is used to bootstrap the first node of the Ceph Storage cluster.

The **cephadm** package is included in the **overcloud-hardened-uefi-full.qcow2** image. The **tripleo_cephadm** role uses the Ansible package module to ensure it is present in the image.

2.4. DEPLOYING THE SOFTWARE IMAGE FOR AN HCI ENVIRONMENT

Nodes configured for an HCI environment must use the **overcloud-hardened-uefi-full.qcow2** software image. Using this software image requires a Red Hat OpenStack Platform (RHOSP) subscription.

Procedure

1. Open your **/home/stack/templates/overcloud-baremetal-deploy.yaml** file.

2. Add or update the **image** property for nodes that require the **overcloud-hardened-uefi-full** image. You can set the image to be used on specific nodes, or for all nodes that use a specific role:

   #### Specific nodes

```
- name: Ceph
  count: 3
  instances:
  - hostname: overcloud-ceph-0
    name: node00
    image:
      href: file:///var/lib/ironic/images/overcloud-minimal.qcow2
  - hostname: overcloud-ceph-1
    name: node01
    image:
      href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2
  - hostname: overcloud-ceph-2
    name: node02
    image:
      href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2
```

All nodes configured for a specific role

```
- name: ComputeHCI
  count: 3
  defaults:
    image:
      href: file:///var/lib/ironic/images/overcloud-hardened-uefi-full.qcow2
  instances:
  - hostname: overcloud-ceph-0
    name: node00
  - hostname: overcloud-ceph-1
    name: node01
  - hostname: overcloud-ceph-2
    name: node02
```

3. In the **roles_data.yaml** role definition file, set the **rhsm_enforce** parameter to **False**.

   ```
   rhsm_enforce: False
   ```

4. Run the provisioning command:

   ```
   (undercloud)$ openstack overcloud node provision \
   --stack overcloud \
   --output /home/stack/templates/overcloud-baremetal-deployed.yaml \
   /home/stack/templates/overcloud-baremetal-deploy.yaml
   ```

5. Pass the **overcloud-baremetal-deployed.yaml** environment file to the **openstack overcloud deploy** command.

2.5. DESIGNATING NODES FOR HCI

To designate nodes for HCI, you must create a new role file to configure the **ComputeHCI** role, and configure the bare metal nodes with a resource class for **ComputeHCI**.

Procedure

1. Log in to the undercloud as the **stack** user.

2. Source the **stackrc** credentials file:

```
[stack@director ~]$ source ~/stackrc
```

3. Generate a new roles data file named **roles_data.yaml** that includes the **Controller** and **ComputeHCI** roles:

```
(undercloud)$ openstack overcloud roles generate Controller ComputeHCI -o
~/roles_data.yaml
```

4. Open **roles_data.yaml** and ensure that it has the following parameters and sections:

Section/Parameter	Value
Role comment	**Role: ComputeHCI**
Role name	**name: ComputeHCI**
description	**HCI role**
HostnameFormatDefault	**%stackname%-novaceph-%index%**
deprecated_nic_config_name	**ceph.yaml**

5. Register the ComputeHCI nodes for the overcloud by adding them to your node definition template, **node.json** or **node.yaml**.

6. Inspect the node hardware:

```
(undercloud)$ openstack overcloud node introspect --all-manageable --provide
```

7. Tag each bare metal node that you want to designate for HCI with a custom HCI resource class:

```
(undercloud)$ openstack baremetal node set \
 --resource-class baremetal.HCI <node>
```

Replace **<node>** with the ID of the bare metal node.

8. Add the **ComputeHCI** role to your **/home/stack/templates/overcloud-baremetal-deploy.yaml** file, and define any predictive node placements, resource classes, or other attributes that you want to assign to your nodes:

```
- name: Controller
  count: 3
- name: ComputeHCI
  count: 1
  defaults:
    resource_class: baremetal.HCI
```

9. Open the **baremetal.yaml** file and ensure that it contains the network configuration necessary for HCI. The following is an example configuration:

```
- name: ComputeHCI
  count: 3
  hostname_format: compute-hci-%index%
  defaults:
    profile: ComputeHCI
    network_config:
      template: /home/stack/templates/three-nics-vlans/compute-hci.j2
    networks:
    - network: ctlplane
      vif: true
    - network: external
      subnet: external_subnet
    - network: internalapi
      subnet: internal_api_subnet01
    - network: storage
      subnet: storage_subnet01
    - network: storage_mgmt
      subnet: storage_mgmt_subnet01
    - network: tenant
      subnet: tenant_subnet01
```

> **NOTE**
>
> Network configuration in the **ComputeHCI** role contains the **storage_mgmt** network. CephOSD nodes use this network to make redundant copies of data. The network configuration for the **Compute** role does not contain this network.

See Configuring the Bare Metal Provisioning service for more information.

10. Run the provisioning command:

```
(undercloud)$ openstack overcloud node provision \
--stack overcloud \
--output /home/stack/templates/overcloud-baremetal-deployed.yaml \
/home/stack/templates/overcloud-baremetal-deploy.yaml
```

11. Monitor the provisioning progress in a separate terminal.

```
(undercloud)$ watch openstack baremetal node list
```

> **NOTE**
>
> The **watch** command renews every 2 seconds by default. The **-n** option sets the renewal timer to a different value.

12. To stop the **watch** process, enter **Ctrl-c**.

13. Verification: When provisioning is successful, the node state changes from **available** to **active**.

Additional resources

- For more information about registering nodes, see Registering nodes for the overcloud in the *Installing and managing Red Hat OpenStack Platform with director* guide.

- For more information about inspecting node hardware, see Creating an inventory of the bare-metal node hardware in the *Installing and managing Red Hat OpenStack Platform with director* guide.

- For more information on network configuration in the **ComputeHCI** role and the **storage_mgmt** network, see Configuring the Bare Metal Provisioning service .

2.6. DEFINING THE ROOT DISK FOR MULTI-DISK CEPH CLUSTERS

Ceph Storage nodes typically use multiple disks. Director must identify the root disk in multiple disk configurations. The overcloud image is written to the root disk during the provisioning process.

Hardware properties are used to identify the root disk. For more information about properties you can use to identify the root disk, see Properties that identify the root disk .

Procedure

1. Verify the disk information from the hardware introspection of each node:

```
(undercloud)$ openstack baremetal introspection data save <node_uuid> --file
<output_file_name>
```

- Replace **<node_uuid>** with the UUID of the node.

- Replace **<output_file_name>** with the name of the file that contains the output of the node introspection.
  For example, the data for one node might show three disks:

```
[
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sda",
    "wwn_vendor_extension": "0x1ea4dcc412a9632b",
    "wwn_with_extension": "0x61866da04f3807001ea4dcc412a9632b",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f380700",
    "serial": "61866da04f3807001ea4dcc412a9632b"
  }
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
    "name": "/dev/sdb",
    "wwn_vendor_extension": "0x1ea4e13c12e36ad6",
    "wwn_with_extension": "0x61866da04f380d001ea4e13c12e36ad6",
    "model": "PERC H330 Mini",
    "wwn": "0x61866da04f380d00",
    "serial": "61866da04f380d001ea4e13c12e36ad6"
  }
  {
    "size": 299439751168,
    "rotational": true,
    "vendor": "DELL",
        "name": "/dev/sdc",
        "wwn_vendor_extension": "0x1ea4e31e121cfb45",
        "wwn_with_extension": "0x61866da04f37fc001ea4e31e121cfb45",
        "model": "PERC H330 Mini",
        "wwn": "0x61866da04f37fc00",
        "serial": "61866da04f37fc001ea4e31e121cfb45"
    }
]
```

2. Set the root disk for the node by using a unique hardware property:

   ```
   (undercloud)$ openstack baremetal node set --property root_device='{<property_value>}'
   <node-uuid>
   ```

   - Replace **<property_value>** with the unique hardware property value from the introspection data to use to set the root disk.
   - Replace **<node_uuid>** with the UUID of the node.

   NOTE

   A unique hardware property is any property from the hardware introspection step that uniquely identifies the disk. For example, the following command uses the disk serial number to set the root disk:

   ```
   (undercloud)$ openstack baremetal node set --property
   root_device='{"serial": "61866da04f380d001ea4e13c12e36ad6"}'
   1a4e30da-b6dc-499d-ba87-0bd8a3819bc0
   ```

3. Configure the BIOS of each node to first boot from the network and then the root disk.

Director identifies the specific disk to use as the root disk. When you run the **openstack overcloud node provision** command, director provisions and writes the overcloud image to the root disk.

2.6.1. Properties that identify the root disk

There are several properties that you can define to help director identify the root disk:

- **model** (String): Device identifier.
- **vendor** (String): Device vendor.
- **serial** (String): Disk serial number.
- **hctl** (String): Host:Channel:Target:Lun for SCSI.
- **size** (Integer): Size of the device in GB.
- **wwn** (String): Unique storage identifier.
- **wwn_with_extension** (String): Unique storage identifier with the vendor extension appended.
- **wwn_vendor_extension** (String): Unique vendor storage identifier.
- **rotational** (Boolean): True for a rotational device (HDD), otherwise false (SSD).
- **name** (String): The name of the device, for example: /dev/sdb1.

IMPORTANT

Use the **name** property for devices with persistent names. Do not use the **name** property to set the root disk for devices that do not have persistent names because the value can change when the node boots.

CHAPTER 3. CONFIGURING THE RED HAT CEPH STORAGE CLUSTER FOR HCI

This chapter describes how to configure and deploy the Red Hat Ceph Storage cluster for HCI environments.

3.1. DEPLOYMENT PREREQUISITES

Confirm the following has been performed before attempting to configure and deploy the Red Hat Ceph Storage cluster:

- Provision of bare metal instances and their networks using the Bare Metal Provisioning service (ironic). For more information about the provisioning of bare metal instances, see Configuring the Bare Metal Provisioning service.

3.2. THE OPENSTACK OVERCLOUD CEPH DEPLOY COMMAND

If you deploy the Ceph cluster using director, you must use the **openstack overcloud ceph deploy** command. For a complete listing of command options and parameters, see openstack overcloud ceph deploy in the *Command line interface reference*.

The command **openstack overcloud ceph deploy --help** provides the current options and parameters available in your environment.

3.3. CEPH CONFIGURATION OVERRIDES FOR HCI

A standard format initialization file is an option for Ceph cluster configuration. This initialization file is then used to configure the Ceph cluster with either the **cephadm bootstap --config <file_name>** or **openstack overcloud ceph deploy --config <file_name>** commands.

Colocating Ceph OSD and Compute services on hyperconverged nodes risks resource contention between Red Hat Ceph Storage and Compute services. This occurs because the services are not aware of the colocation. Resource contention can result in service degradation, which offsets the benefits of hyperconvergence.

Resource allocation can be tuned using an initialization file to manage resource contention. The following creates an initialization file called **initial-ceph.conf** and then uses the **openstack overcloud ceph deploy** command to configure the HCI deployment.

```
$ cat <<EOF > initial-ceph.conf
[osd]
osd_memory_target_autotune = true
osd_numa_auto_affinity = true
[mgr]
mgr/cephadm/autotune_memory_target_ratio = 0.2
EOF
$ openstack overcloud ceph deploy --config initial-ceph.conf
```

The **osd_memory_target_autotune** option is set to **true** so that the OSD daemons adjust their memory consumption based on the **osd_memory_target config** option. The **autotune_memory_target_ratio** defaults to **0.7**. This indicates 70% of the total RAM in the system is the starting point from which any memory consumed by non-autotuned Ceph daemons are subtracted. Then the remaining memory is divided by the OSDs, assuming all OSDs have **osd_memory_target_autotune** set to **true**. For HCI deployments, set the **mgr/cephadm/autotune_memory_target_ratio** to 0.2 to ensure more memory is available for the Compute service. The **0.2** value is a cautious starting point. After deployment, use the **ceph** command to change this value if necessary.

A two NUMA node system can host a latency sensitive Nova workload on one NUMA node and a Ceph OSD workload on the other NUMA node. To configure Ceph OSDs to use a specific NUMA node not used by the Compute workload, use either of the following Ceph OSD configurations:

- **osd_numa_node** sets affinity to a numa node
- **osd_numa_auto_affinity** automatically sets affinity to the NUMA node where storage and network match

If there are network interfaces on both NUMA nodes and the disk controllers are NUMA node 0, use a network interface on NUMA node 0 for the storage network and host the Ceph OSD workload on NUMA node 0. Host the Nova workload on NUMA node 1 and have it use the network interfaces on NUMA node 1. Setting **osd_numa_auto_affinity** to **true** to achieve this configuration. Alternatively, the **osd_numa_node** could be set directly to **0** and a value would not be set for **osd_numa_auto_affinity** so that it defaults to **false**.

When a hyperconverged cluster backfills as a result of an OSD going offline, the backfill process can be slowed down. In exchange for a slower recovery, the backfill activity has less of an impact on the collocated Compute workload. Red Hat Ceph Storage has the following defaults to control the rate of backfill activity:

- **osd_recovery_op_priority = 3**
- **osd_max_backfills = 1**
- **osd_recovery_max_active_hdd = 3**
- **osd_recovery_max_active_ssd = 10**

NOTE

It is not necessary to pass these defaults in an initialization file as they are the default values. If values other than the defaults are desired for the inital configuration, add them to the initialization file with the required values before deployment. After deployment, use the command 'ceph config set osd'.

3.4. CONFIGURING TIME SYNCHRONIZATION

The Time Synchronization Service (chrony) is enabled for time synchronization by default. You can perform the following tasks to configure the service.

- Configuring time synchronization with a delimited list
- Configuring time synchronization with an environment file
- Disabling time synchronization

NOTE

Time synchronization is configured using either a delimited list or an environment file. Use the procedure that is best suited to your administrative practices.

3.4.1. Configuring time synchronization with a delimited list

You can configure the Time Synchronization Service (chrony) to use a delimited list to configure NTP servers.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Configure NTP servers with a delimited list:

   ```
   openstack overcloud ceph deploy \
           --ntp-server "<ntp_server_list>"
   ```

   Replace **<ntp_server_list>** with a comma delimited list of servers.

   ```
   openstack overcloud ceph deploy \
           --ntp-server "0.pool.ntp.org,1.pool.ntp.org"
   ```

3.4.2. Configuring time synchronization with an environment file

You can configure the Time Synchronization Service (chrony) to use an environment file that defines NTP servers.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Create an environment file, such as **/home/stack/templates/ntp-parameters.yaml**, to contain the NTP server configuration.
3. Add the **NtpServer** parameter. The **NtpServer** parameter contains a comma delimited list of NTP servers.

   ```
   parameter_defaults:
     NtpServer: 0.pool.ntp.org,1.pool.ntp.org
   ```

4. Configure NTP servers with an environment file:

   ```
   openstack overcloud ceph deploy \
           --ntp-heat-env-file "<ntp_file_name>"
   ```

   Replace **<ntp_file_name>** with the name of the environment file you created.

   ```
   openstack overcloud ceph deploy \
           --ntp-heat-env-file "/home/stack/templates/ntp-parameters.yaml"
   ```

3.4.3. Disabling time synchronization

The Time Synchronization Service (chrony) is enabled by default. You can disable the service if you do not want to use it.

Procedure

1. Log in to the undercloud node as the **stack** user.

2. Disable the Time Synchronization Service (chrony):

```
openstack overcloud ceph deploy \
         --skip-ntp
```

3.5. CONFIGURING A TOP LEVEL DOMAIN SUFFIX

You can configure a top level domain (TLD) suffix. This suffix is added to the short hostname to create a fully qualified domain name for overcloud nodes.

> **NOTE**
>
> A fully qualified domain name is required for TLS-e configuration.

Procedure

1. Log in to the undercloud node as the **stack** user.

2. Configure the top level domain suffix:

```
openstack overcloud ceph deploy \
        --tld "<domain_name>"
```

Replace **<domain_name>** with the required domain name.

```
openstack overcloud ceph deploy \
        --tld "example.local"
```

3.6. CONFIGURING THE RED HAT CEPH STORAGE CLUSTER NAME

You can deploy the Red Hat Ceph Storage cluster with a name that you configure. The default name is **ceph**.

Procedure

1. Log in to the undercloud node as the **stack** user.

2. Configure the name of the Ceph Storage cluster by using the following command:
openstack overcloud ceph deploy \ --cluster <cluster_name>

**$ openstack overcloud ceph deploy \ --cluster central **

NOTE

Keyring files are not created at this time. Keyring files are created during the overcloud deployment. Keyring files inherit the cluster name configured during this procedure. For more information about overcloud deployment see Section 5.8, “Initiating overcloud deployment for HCI”

In the example above, the Ceph cluster is named **central**. The configuration and keyring files for the **central** Ceph cluster would be created in **/etc/ceph** during the deployment process.

```
[root@oc0-controller-0 ~]# ls -l /etc/ceph/
total 16
-rw-------. 1 root root  63 Mar 26 21:49 central.client.admin.keyring
-rw-------. 1  167  167 201 Mar 26 22:17 central.client.openstack.keyring
-rw-------. 1  167  167 134 Mar 26 22:17 central.client.radosgw.keyring
-rw-r--r--. 1 root root 177 Mar 26 21:49 central.conf
```

Troubleshooting

The following error may be displayed if you configure a custom name for the Ceph Storage cluster:

monclient: get_monmap_and_config cannot identify monitors to contact because

If this error is displayed, use the following command after Ceph deployment:

cephadm shell --config <configuration_file> --keyring <keyring_file>

For example, if this error was displayed when you configured the cluster name to **central**, you would use the following command:

```
cephadm shell --config /etc/ceph/central.conf \
          --keyring /etc/ceph/central.client.admin.keyring
```

The following command could also be used as an alternative:

```
cephadm shell --mount /etc/ceph:/etc/ceph
export CEPH_ARGS='--cluster central'
```

3.7. CONFIGURING NETWORK OPTIONS WITH THE NETWORK DATA FILE

The network data file describes the networks used by the Red Hat Ceph Storage cluster.

Procedure

1. Log in to the undercloud node as the **stack** user.

2. Create a YAML format file that defines the custom network attributes called **network_data.yaml**.

IMPORTANT

Using network isolation, the standard network deployment consists of two storage networks which map to the two Ceph networks:

- The storage network, **storage**, maps to the Ceph network, **public_network**. This network handles storage traffic such as the RBD traffic from the Compute nodes to the Ceph cluster.
- The storage network, **storage_mgmt**, maps to the Ceph network, **cluster_network**. This network handles storage management traffic such as data replication between Ceph OSDs.

3. Use the **openstack overcloud ceph deploy** command with the **--crush-hierarchy** option to deploy the configuration.

```
openstack overcloud ceph deploy \
        deployed_metal.yaml \
        -o deployed_ceph.yaml \
        --network-data network_data.yaml
```

IMPORTANT

The **openstack overcloud ceph deploy** command uses the network data file specified by the **--network-data** option to determine the networks to be used as the **public_network** and **cluster_network**. The command assumes these networks are named **storage** and **storage_mgmt** in network data file unless a different name is specified by the **--public-network-name** and **--cluster-network-name** options.

You must use the **--network-data** option when deploying with network isolation. The default undercloud (192.168.24.0/24) will be used for both the **public_network** and **cluster_network** if you do not use this option.

3.8. CONFIGURING NETWORK OPTIONS WITH A CONFIGURATION FILE

Network options can be specified with a configuration file as an alternative to the network data file.

IMPORTANT

Using this method to configure network options overwrites automatically generated values in **network_data.yaml**. Ensure you set all four values when using this network configuration method.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Create a standard format initialization file to configure the Ceph cluster. If you have already created a file to include other configuration options, you can add the network configuration to it.
3. Add the following parameters to the **[global]** section of the file:
    - **public_network**

- **public_network**
- **cluster_network**
- **ms_bind_ipv4**

> IMPORTANT
>
> Ensure the **public_network** and **cluster_network** map to the same networks as **storage** and **storage_mgmt**.

The following is an example of a configuration file entry for a network configuration with multiple subnets and custom networking names:

```
[global]
public_network = 172.16.14.0/24,172.16.15.0/24
cluster_network = 172.16.12.0/24,172.16.13.0/24
ms_bind_ipv4 = True
ms_bind_ipv6 = False
```

4. Use the command **openstack overcloud ceph deploy** with the **--config** option to deploy the configuration file.

```
$ openstack overcloud ceph deploy \
  --config initial-ceph.conf --network-data network_data.yaml
```

3.9. CONFIGURING A CRUSH HIERARCHY FOR AN OSD

You can configure a custom Controlled Replication Under Scalable Hashing (CRUSH) hierarchy during OSD deployment to add the OSD **location** attribute to the Ceph Storage cluster **hosts** specification. The **location** attribute configures where the OSD is placed within the CRUSH hierarchy.

> NOTE
>
> The **location** attribute sets only the initial CRUSH location. Subsequent changes of the attribute are ignored.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Source the **stackrc** undercloud credentials file:
   $ source ~/stackrc
3. Create a configuration file to define the custom CRUSH hierarchy, for example, **crush_hierarchy.yaml**.
4. Add the following configuration to the file:

```
<osd_host>:
  root: default
  rack: <rack_num>
<osd_host>:
  root: default
  rack: <rack_num>
<osd_host>:
  root: default
  rack: <rack_num>
```

- Replace **<osd_host>** with the hostnames of the nodes where the OSDs are deployed, for example, **ceph-0**.
- Replace **<rack_num>** with the number of the rack where the OSDs are deployed, for example, **r0**.

5. Deploy the Ceph cluster with your custom OSD layout:

```
openstack overcloud ceph deploy \
        deployed_metal.yaml \
        -o deployed_ceph.yaml \
        --osd-spec osd_spec.yaml \
        --crush-hierarchy crush_hierarchy.yaml
```

The Ceph cluster is created with the custom OSD layout.

The example file above would result in the following OSD layout.

```
ID  CLASS  WEIGHT      TYPE NAME               STATUS  REWEIGHT  PRI-AFF
-1         0.02939     root default
-3         0.00980     rack r0
-2         0.00980         host ceph-node-00
 0    hdd  0.00980             osd.0            up   1.00000   1.00000
-5         0.00980     rack r1
-4         0.00980         host ceph-node-01
 1    hdd  0.00980             osd.1            up   1.00000   1.00000
-7         0.00980     rack r2
-6         0.00980         host ceph-node-02
 2    hdd  0.00980             osd.2            up   1.00000   1.00000
```

NOTE

Device classes are automatically detected by Ceph but CRUSH rules are associated with pools. Pools are still defined and created using the **CephCrushRules** parameter during the overcloud deployment.

Additional resources

See Red Hat Ceph Storage workload considerations in the *Red Hat Ceph Storage Installation Guide* for additional information.

3.10. CONFIGURING CEPH SERVICE PLACEMENT OPTIONS

You can define what nodes run what Ceph services using a custom roles file. A custom roles file is only necessary when default role assignments are not used because of the environment. For example, when deploying hyperconverged nodes, the predeployed compute nodes should be labeled as **osd** with a service type of **osd** to have a placement list containing a list of compute instances.

Service definitions in the **roles_data.yaml** file determine which bare metal instance runs which service.

By default, the Controller role has the CephMon and CephMgr service while the CephStorage role has the CephOSD service. Unlike most composable services, Ceph services do not require heat output to determine how services are configured. The **roles_data.yaml** file always determines Ceph service placement even though the deployed Ceph process occurs before Heat runs.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Create a YAML format file that defines the custom roles.
3. Deploy the configuration file:

```
$ openstack overcloud ceph deploy \
      deployed_metal.yaml \
      -o deployed_ceph.yaml \
      --roles-data custom_roles.yaml
```

3.11. CONFIGURING SSH USER OPTIONS FOR CEPH NODES

The **openstack overcloud ceph deploy** command creates the user and keys and distributes them to the hosts so it is not necessary to perform the procedures in this section. However, it is a supported option.

Cephadm connects to all managed remote Ceph nodes using SSH. The Red Hat Ceph Storage cluster deployment process creates an account and SSH key pair on all overcloud Ceph nodes. The key pair is then given to Cephadm so it can communicate with the nodes.

3.11.1. Creating the SSH user before Red Hat Ceph Storage cluster creation

You can create the SSH user before Ceph cluster creation with the **openstack overcloud ceph user enable** command.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Create the SSH user:
   $ openstack overcloud ceph user enable <specification_file>
   - Replace **<specification_file>** with the path and name of a Ceph specification file that describes the cluster where the user is created and the public SSH keys are installed. The specification file provides the information to determine which nodes to modify and if the private keys are required.
     For more information on creating a specification file, see Generating the service specification.

NOTE

The default user name is **ceph-admin**. To specify a different user name, use the **--cephadm-ssh-user** option to specify a different one.

openstack overcloud ceph user enable --cephadm-ssh-user <custom_user_name>

It is recommended to use the default name and not use the **--cephadm-ssh-user** parameter.

If the user is created in advance, use the parameter **--skip-user-create** when executing **openstack overcloud ceph deploy**.

3.11.2. Disabling the SSH user

Disabling the SSH user disables **cephadm**. Disabling **cephadm** removes the ability of the service to administer the Ceph cluster and prevents associated commands from working. It also prevents Ceph node overcloud scaling operations. It also removes all public and private SSH keys.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Use the command **openstack overcloud ceph user disable --fsid <FSID> <specification_file>** to disable the SSH user.
   - Replace **<FSID>** with the File System ID of the cluster. The FSID is a unique identifier for the cluster. The FSID is located in the **deployed_ceph.yaml** environment file.
   - Replace **<specification_file>** with the path and name of a Ceph specification file that describes the cluster where the user was created.

   IMPORTANT

   The **openstack overcloud ceph user disable** command is not recommended unless it is necessary to disable **cephadm**.

   IMPORTANT

   To enable the SSH user and Ceph orchestrator service after being disabled, use the **openstack overcloud ceph user enable --fsid <FSID> <specification_file>** command.

NOTE

This command requires the path to a Ceph specification file to determine:

- Which hosts require the SSH user.
- Which hosts have the **_admin** label and require the private SSH key.
- Which hosts require the public SSH key.

For more information about specification files and how to generate them, see Generating the service specification.

3.12. ACCESSING CEPH STORAGE CONTAINERS

Preparing container images in the *Installing and managing Red Hat OpenStack Platform with director* guide contains procedures and information on how to prepare the registry and your undercloud and overcloud configuration to use container images. Use the information in this section to adapt these procedures to access Ceph Storage containers.

There are two options for accessing Ceph Storage containers from the overcloud.

- Downloading containers directly from a remote registry
- Cacheing containers on the undercloud

3.12.1. Cacheing containers on the undercloud

The procedure Modifying images during preparation describes using the following command:

```
sudo openstack tripleo container image prepare \
  -e ~/containers-prepare-parameter.yaml \
```

If you do not use the **--container-image-prepare** option to provide authentication credentials to the **openstack overcloud ceph deploy** command and directly download the Ceph containers from a remote registry, as described in Downloading containers directly from a remote registry, you must run the **sudo openstack tripleo container image prepare** command before deploying Ceph.

3.12.2. Downloading containers directly from a remote registry

You can configure Ceph to download containers directly from a remote registry.

The **cephadm** command uses the credentials that are configured in the **containers-prepare-parameter.yaml** file to authenticate to the remote registry and download the Red Hat Ceph Storage container.

Procedure

1. Create a **containers-prepare-parameter.yaml** file using the procedure Preparing container images in the *Installing and managing Red Hat OpenStack Platform with director* guide.
2. Add the remote registry credentials to the **containers-prepare-parameter.yaml** file using the **ContainerImageRegistryCredentials** parameter as described in Obtaining container images from private registries.

3. When you deploy Ceph, pass the **containers-prepare-parameter.yaml** file using the **openstack overcloud ceph deploy** command.

```
openstack overcloud ceph deploy \
        --container-image-prepare containers-prepare-parameter.yaml
```

NOTE

If you do not cache the containers on the undercloud, as described in Cacheing containers on the undercloud, then you should pass the same **containers-prepare-parameter.yaml** file to the **openstack overcloud ceph deploy** command when you deploy Ceph. This will cache containers on the undercloud.

CHAPTER 4. CUSTOMIZING THE RED HAT CEPH STORAGE CLUSTER FOR HCI

Red Hat OpenStack Platform (RHOSP) director uses a default configuration to deploy containerized Red Hat Ceph Storage. You can customize Ceph Storage by overriding the default settings.

Prerequisites

- The servers are deployed and their storage networks configured.
- The deployed bare metal file as output by **openstack overcloud node provision -o ~/deployed_metal.yaml ...**.

4.1. CONFIGURATION OPTIONS

There are several options for configuring the Red Hat Ceph Storage cluster.

Procedure

1. Log in to the undercloud node as the **stack** user.
2. Optional: Use a standard format initialization (ini) file to configure the Ceph cluster.
   a. Create the file with configuration options.
   The following is an example of a simple configuration file:

   ```
   [global]
   osd_crush_chooseleaf type = 0
   log_file = /var/log/ceph/$cluster-$type.$id.log

   [mon]
   mon_cluster_log_to_syslog = true
   ```

   b. Save the configuration file.

   c. Use the **openstack overcloud ceph deploy --config <configuration_file_name>** command to deploy the configuration.
   Replace **<configuration_file_name>** with the name of the file you created.

   ```
   $ openstack overcloud ceph deploy --config initial-ceph.conf
   ```

3. Optional: Send configuration values to the **cephadm bootstrap** command:
   **openstack overcloud ceph deploy --force \ --cephadm-extra-args '<optional_arguments>' **

   Replace **<optional_arguments>** with the configuration values to provide to the underlying command.

> NOTE
>
> When using the arguments **--log-to-file** and **--skip-prepare-host**, the command **openstack overcloud ceph deploy --force \ --cephadm-extra-args '--log-to-file --skip-prepare-host' ** is used.

4.2. GENERATING THE SERVICE SPECIFICATION (OPTIONAL)

The Red Hat Ceph Storage cluster service specification is a YAML file that describes the deployment of Ceph Storage services. It is automatically generated by **tripleo** before the Ceph Storage cluster is deployed. It does not usually have to be generated separately.

A custom service specification can be created to customize the Red Hat Ceph Storage cluster.

Procedure

1. Log in to the undercloud node as the **stack** user.

2. Generate the specification file:
   openstack overcloud ceph spec deployed_metal.yaml -o <specification_file>

   - Replace **<specification_file>** with the name of the file to generate with the current service specification.

     NOTE

     The **deployed_metal.yaml** comes from the output of the **openstack overcloud node provision** command.

3. Edit the generated file with the required configuration.

4. Deploy the custom service specification:

   ```
   openstack overcloud ceph deploy \
     deployed_metal.yaml \
     -o deployed_ceph.yaml \
     --ceph-spec <specification_file>
   ```

   - Replace **<specification_file>** with the name of the custom service specification file.

4.3. CEPH CONTAINERS FOR RED HAT OPENSTACK PLATFORM WITH RED HAT CEPH STORAGE

You must have a Ceph Storage container to configure Red Hat Openstack Platform (RHOSP) to use Red Hat Ceph Storage with NFS Ganesha. You do not require a Ceph Storage container if the external Ceph Storage cluster only provides Block (through RBD), Object (through RGW), or File (through native CephFS) storage.

RHOSP 17.1 will deploy Red Hat Ceph Storage 6.x (Ceph package 17.x). The Ceph Storage 6.x containers are hosted at **registry.redhat.io**, a registry that requires authentication. For more information, see Container image preparation parameters.

4.4. CONFIGURING ADVANCED OSD SPECIFICATIONS

Configure an advanced OSD specification when the default specification does not provide the necessary functionality for your Ceph Storage cluster.

Procedure

1. Log in to the undercloud node as the **stack** user.

2. Create a YAML format file that defines the advanced OSD specification.
   The following is an example of a custom OSD specification.

   ```
   data_devices:
     rotational: 1
   db_devices:
     rotational: 0
   ```

   This example would create an OSD specification where all rotating devices will be data devices and all non-rotating devices will be used as shared devices. When the dynamic Ceph service specification is built, whatever is in the specification file is appended to the section of the specification if the **service_type** is **osd**.

3. Save the specification file.

4. Deploy the specification:
   openstack overcloud ceph deploy \ --osd-spec <osd_specification_file>

   Replace **<osd_specification_file>** with the name of the specification file you created.

   **$ openstack overcloud ceph deploy \ --osd-spec osd_spec.yaml **

Additional resources

For a list of OSD-related attributes used to configure OSDs in the service specification, see Advanced service specifications and filters for deploying OSDs in the *Red Hat Ceph Storage Operations Guide* .

4.5. MIGRATING FROM NODE-SPECIFIC OVERRIDES

Node-specific overrides were used to manage non-homogenous server hardware before Red Hat OpenStack Platform 17.0. This is now done with a custom OSD specification file. See Configuring advanced OSD specifications for information on how to create a custom OSD specification file.

4.6. ENABLING CEPH ON-WIRE ENCRYPTION

Enable encryption for all Ceph Storage traffic using the **secure mode** of the messenger version 2 protocol. Configure Ceph Storage as described in Encryption and Key Management in the Red Hat Ceph Storage *Data Hardening Red Hat OpenStack Platform* to enable Ceph on-wire encryption.

Additional resources

For more information about Ceph on-wire encryption, see Ceph on-wire encryption in the Red Hat Ceph Storage *Architecture Guide*.

CHAPTER 5. CUSTOMIZING THE STORAGE SERVICE FOR HCI

Red Hat OpenStack Platform (RHOSP) director provides the necessary heat templates and environment files to enable a basic Ceph Storage configuration.

Director uses the **/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml** environment file to add additional configuration to the Ceph cluster deployed by **openstack overcloud ceph deploy**.

For more information about containerized services in RHOSP, see Configuring a basic overcloud with the CLI tools in *Installing and managing Red Hat OpenStack Platform with director* .

5.1. CONFIGURING COMPUTE SERVICE RESOURCES FOR HCI

Colocating Ceph OSD and Compute services on hyperconverged nodes risks resource contention between Red Hat Ceph Storage and Compute services. This occurs because the services are not aware of the colcation. Resource contention can result in service degradation, which offsets the benefits of hyperconvergence.

Configuring the resources used by the Compute service mitigates resource contention and improves HCI performance.

Procedure

1. Log in to the undercloud host as the stack user.
2. Source the stackrc undercloud credentials file:

```
$ source ~/stackrc
```

3. Add the **NovaReservedHostMemory** parameter to the **ceph-overrides.yaml** file. The following is a usage example.

```
parameter_defaults:
  ComputeHCIParameters:
    NovaReservedHostMemory: 75000
```

The **NovaReservedHostMemory** parameter overrides the default value of **reserved_host_memory_mb** in **/etc/nova/nova.conf**. This parameter is set to stop Nova scheduler giving memory, that a Ceph OSD needs, to a virtual machine.

The example above reserves 5 GB per OSD for 10 OSDs per host in addition to the default reserved memory for the hypervisor. In an IOPS-optimized cluster, you can improve performance by reserving more memory per OSD. The 5 GB number is provided as a starting point that you can further refine as necessary.

IMPORTANT

Include this file when you use the **openstack overcloud deploy** command.

5.2. CONFIGURING A CUSTOM ENVIRONMENT FILE

Director applies basic, default settings to the deployed Red Hat Ceph Storage cluster. You must define additional configuration in a custom environment file.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Create a file to define the custom configuration.
   vi /home/stack/templates/storage-config.yaml
3. Add a **parameter_defaults** section to the file.
4. Add the custom configuration parameters. For more information about parameter definitions, see Overcloud parameters.

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  CinderEnableRbdBackend: true
  CinderBackupBackend: ceph
  NovaEnableRbdBackend: true
  GlanceBackend: rbd
```

> **NOTE**
>
> Parameters defined in a custom configuration file override any corresponding default settings in **/usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml**.

5. Save the file.

Additional resources

The custom configuration is applied during overcloud deployment.

5.3. ENABLING CEPH METADATA SERVER

The Ceph Metadata Server (MDS) runs the **ceph-mds** daemon. This daemon manages metadata related to files stored on CephFS. CephFS can be consumed natively or through the NFS protocol.

> **NOTE**
>
> Red Hat supports deploying Ceph MDS with the native CephFS and CephFS NFS back ends for the Shared File Systems service (manila).

Procedure

- To enable Ceph MDS, use the following environment file when you deploy the overcloud:

```
/usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml
```

> **NOTE**
>
> By default, Ceph MDS is deployed on the Controller node. You can deploy Ceph MDS on its own dedicated node.

Additional resources

- Red Hat Ceph Storage *File System Guide*

5.4. CEPH OBJECT GATEWAY OBJECT STORAGE

The Ceph Object Gateway (RGW) provides an interface to access object storage capabilities within a Red Hat Ceph Storage cluster.

When you use director to deploy Ceph, director automatically enables RGW. This is a direct replacement for the Object Storage service (swift). Services that normally use the Object Storage service can use RGW instead without additional configuration. The Object Storage service remains available as an object storage option for upgraded Ceph clusters.

There is no requirement for a separate RGW environment file to enable it. For more information about environment files for other object storage options, see Section 5.5, "Deployment options for Red Hat OpenStack Platform object storage".

By default, Ceph Storage allows 250 placement groups per Object Storage Daemon (OSD). When you enable RGW, Ceph Storage creates the following six additional pools required by RGW:

- **.rgw.root**
- **<zone_name>.rgw.control**
- **<zone_name>.rgw.meta**
- **<zone_name>.rgw.log**
- **<zone_name>.rgw.buckets.index**
- **<zone_name>.rgw.buckets.data**

NOTE

In your deployment, **<zone_name>** is replaced with the name of the zone to which the pools belong.

Additional resources

- For more information about RGW, see the Red Hat Ceph Storage Object Gateway Guide .
- For more information about using RGW instead of Swift, see the Backuping up BLock Storage volumes guide.

5.5. DEPLOYMENT OPTIONS FOR RED HAT OPENSTACK PLATFORM OBJECT STORAGE

There are three options for deploying overcloud object storage:

- Ceph Object Gateway (RGW)
  To deploy RGW as described in Section 5.4, "Ceph Object Gateway object storage" , include the following environment file during overcloud deployment:

```
-e  environments/cephadm/cephadm.yaml
```

This environment file configures both Ceph block storage (RBD) and RGW.

- Object Storage service (swift)
  To deploy the Object Storage service (swift) instead of RGW, include the following environment file during overcloud deployment:

  ```
  -e  environments/cephadm/cephadm-rbd-only.yaml
  ```

  The **cephadm-rbd-only.yaml** file configures Ceph RBD but not RGW.

  NOTE

  If you used the Object Storage service (swift) before upgrading your Red Hat Ceph Storage cluster, you can continue to use the Object Storage service (swift) instead of RGW by replacing the **environments/ceph-ansible/ceph-ansible.yaml** file with the **environments/cephadm/cephadm-rbd-only.yaml** during the upgrade. For more information, see Performing a minor update of Red Hat OpenStack Platform.

  Red Hat OpenStack Platform does not support migration from the Object Storage service (swift) to Ceph Object Gateway (RGW).

- No object storage
  To deploy Ceph with RBD but not with RGW or the Object Storage service (swift), include the following environment files during overcloud deployment:

  ```
  -e  environments/cephadm/cephadm-rbd-only.yaml
  -e  environments/disable-swift.yaml
  ```

  The **cephadm-rbd-only.yaml** file configures RBD but not RGW. The **disable-swift.yaml** file ensures that the Object Storage service (swift) does not deploy.

5.6. CONFIGURING THE BLOCK STORAGE BACKUP SERVICE TO USE CEPH

The Block Storage Backup service (cinder-backup) is disabled by default. It must be enabled to use it with Ceph.

Procedure

To enable the Block Storage Backup service (cinder-backup), use the following environment file when you deploy the overcloud:

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

5.7. CONFIGURING MULTIPLE BONDED INTERFACES FOR CEPH NODES

Use a bonded interface to combine multiple NICs and add redundancy to a network connection. If you have enough NICs on your Ceph nodes, you can create multiple bonded interfaces on each node to expand redundancy capability.

Use a bonded interface for each network connection the node requires. This provides both redundancy and a dedicated connection for each network.

See Provisioning the overcloud networks in the *Installing and managing Red Hat OpenStack Platform with director* guide for information and procedures.

5.8. INITIATING OVERCLOUD DEPLOYMENT FOR HCI

To implement the changes you made to your Red Hat OpenStack Platform (RHOSP) environment, you must deploy the overcloud.

Prerequisites

- Before undercloud installation, set **generate_service_certificate=false** in the **undercloud.conf** file. Otherwise, you must configure SSL/TLS on the overcloud as described in Enabling SSL/TLS on overcloud public endpoints in the *Hardening Red Hat OpenStack Platform*.

NOTE

If you want to add Ceph Dashboard during your overcloud deployment, see Adding the Red Hat Ceph Storage Dashboard to an overcloud deployment in *Deploying Red Hat Ceph Storage and Red Hat OpenStack Platform together with director.*

Procedure

- Deploy the overcloud. The deployment command requires additional arguments, for example:

```
$ openstack overcloud deploy --templates -r /home/stack/templates/roles_data_custom.yaml
\
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml \
  -e /home/stack/templates/storage-config.yaml \
  -e /home/stack/templates/deployed-ceph.yaml \
  --ntp-server pool.ntp.org
```

  The example command uses the following options:

  - **--templates** - Creates the overcloud from the default heat template collection, **/usr/share/openstack-tripleo-heat-templates/**.
  - **-r /home/stack/templates/roles_data_custom.yaml** - Specifies a customized roles definition file.
  - **-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/cephadm.yaml** - Sets the director to finalize the previously deployed Ceph Storage cluster. This environment file deploys RGW by default. It also creates pools, keys, and daemons.
  - **-e /usr/share/openstack-tripleo-heat-templates/environments/cephadm/ceph-mds.yaml** - Enables the Ceph Metadata Server.
  - **-e /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml** - Enables the Block Storage Backup service.

  - **-e /home/stack/templates/storage-config.yaml** - Adds the environment file that contains your custom Ceph Storage configuration.
  - **-e /home/stack/templates/deployed-ceph.yaml** - Adds the environment file that contains your Ceph cluster settings, as output by the **openstack overcloud ceph deploy** command run earlier.
  - **--ntp-server pool.ntp.org** - Sets the NTP server.

> NOTE
>
> For a full list of options, run the **openstack help overcloud deploy** command.

Additional resources

- For more information, see Configuring a basic overcloud with the CLI tools in the *Installing and managing Red Hat OpenStack Platform with director* guide.

CHAPTER 6. VERIFYING HCI CONFIGURATION

After deployment is complete, verify the HCI environment is properly configured.

6.1. VERIFYING HCI CONFIGURATION

After the deployment of the HCI environment, verify that the deployment was successful with the configuration specified.

Procedure

1. Start a ceph shell.
2. Confirm NUMA and memory target configuration:

```
[ceph: root@oc0-controller-0 /]# ceph config dump | grep numa
  osd                                 advanced  osd_numa_auto_affinity              true
[ceph: root@oc0-controller-0 /]# ceph config dump | grep autotune
  osd                                 advanced  osd_memory_target_autotune              true
[ceph: root@oc0-controller-0 /]# ceph config get mgr
mgr/cephadm/autotune_memory_target_ratio
0.200000
```

3. Confirm specific OSD configuration:

```
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_memory_target
4294967296
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_memory_target_autotune
true
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_numa_auto_affinity
true
```

4. Confirm specific OSD backfill configuration:

```
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_recovery_op_priority
3
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_max_backfills
1
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_recovery_max_active_hdd
3
[ceph: root@oc0-controller-0 /]# ceph config get osd.11 osd_recovery_max_active_ssd
10
```

5. Confirm the **reserved_host_memory_mb** configuration on the Compute node.

```
$ sudo podman exec -ti nova_compute /bin/bash
bash-5.1$ grep reserved_host_memory_mb /etc/nova/nova.conf
```

CHAPTER 7. SCALING HYPERCONVERGED NODES

To scale HCI nodes up or down, the same principles and methods for scaling Compute nodes or Red Hat Ceph Storage nodes apply.

7.1. SCALING UP HYPERCONVERGED NODES IN HCI ENVIRONMENTS

To scale up hyperconverged nodes in HCI environments follow the same procedure for scaling up non-hyperconverged nodes. For more information, see Adding nodes to the overcloud .

NOTE

When you tag new nodes, remember to use the right flavor.

7.2. SCALING DOWN HYPERCONVERGED NODES IN HCI ENVIRONMENTS

To scale down hyperconverged nodes in HCI environments you must rebalance the Ceph OSD services on the HCI node, migrate instances from the HCI nodes, and remove the Compute nodes from the overcloud.

Procedure

1. Disable and rebalance the Ceph OSD services on the HCI node. This step is necessary because director does not automatically rebalance the Red Hat Ceph Storage cluster when you remove HCI or Red Hat Ceph Storage nodes. For more information, see Scaling the Ceph Storage cluster in *Deploying Red Hat Ceph Storage and Red Hat OpenStack Platform together with director* for more information.

2. Migrate the instances from the HCI nodes. For more information, see Migrating virtual machines between Compute nodes in the *Configuring the Compute service for instance creation* guide.

3. Remove the Compute nodes from the overcloud. For more information, see Removing Compute nodes.

APPENDIX A. ADDITIONAL INFORMATION

A.1. CONFIGURATION GUIDANCE

The following configuration guidance is intended to provide a framework for creating Hyperconverged Infrastructure environments. This guidance is not intended to provide definitive configuration parameters for every Red Hat OpenStack Platform installation. Contact the Red Hat Customer Experience and Engagement team for specific guidance and suggestions that fit your specific environment.

- Cluster sizing and scale out
- Capacity planning and sizing

A.1.1. Cluster sizing and scale out

The Red Hat Ceph Storage Hardware Guide provides recommendations for IOPS optimized, throughput optimized, and cost and capacity optimized Ceph deployment scenarios. Follow the recommendation that best represents your deployment scenario and add the NICs, CPUs, and RAM required to support the Compute workload.

An optimal, small footprint configuration consists of seven nodes. Unless you have a requirement for IOPS optimized performance in your environment and you are using all flash storage, the throughput optimized deployment scenario should be used.

Three node Ceph Storage cluster configurations are possible. In this configuration, you should:

- use all flash storage.
- set the **replica_count** parameter to 3 in the **ceph.conf** file.
- set the **min_size** parameter to 2 in the **ceph.conf** file.

If a node leaves service in this configuration, IOPS continue. To retain 3 copies of the data, replication to the third node is queued until it returns to service. Data is then backfilled to the third node.

NOTE

HCI configurations of up to 64 nodes have been tested. Some HCI environment examples have been documented up to 128 nodes. Large clusters such as these can be considered with a Support Exception and Consulting Services engagement. Contact the Red Hat Customer Experience and Engagement team for guidance.

A deployment with two NUMA nodes can host a latency sensitive Compute workload on one NUMA node and Ceph OSDs services on the other. If there are network interfaces on both nodes, and the disk controllers are on node 0, use a network interface on node 0 for the Storage network and host the Ceph OSD workload on node 0. Host the Compute workload on node 1 and configure it to use the network interfaces on node 1. When acquiring hardware for your deployment, be mindful of which NICs will use which nodes and attempt to split them between storage and workload.

A.1.2. Capacity planning and sizing

The throughput optimized Ceph solution defined in the Red Hat Ceph Storage Hardware Guide provides a balanced solution for most deployments that do not require optimization for IOPS. In addition to the configuration guidelines provided with the solution, note the following when creating your environment:

- The allotment of 5 GB of RAM per OSD ensures OSDs have sufficient operational memory. Ensure your hardware can support this requirement.
- CPU speed should match the storage medium in use. The advantages of faster storage mediums such as SSDs can be negated by CPUs too slow to support them. Similarly, a fast CPU can be more efficiently used by faster storage mediums. Balance CPU and storage medium speed so that neither becomes a bottleneck for the other.

A.2. GUIDES AND RESOURCES FOR THE CONFIGURATION OF YOUR HYPERCONVERGED INFRASTRUCTURE ENVIRONMENT

The following guides contain additional information and procedures that can aid in the configuration of your hyperconverged infrastructure environment.

- Deploying Red Hat Ceph and OpenStack together with director
  - This guide provides information about using the Red Hat OpenStack Platform director to create an overcloud with a Red Hat Ceph Storage cluster. This includes instructions for customizing your Ceph cluster through the director.
- Installing and managing Red Hat OpenStack Platform with director
  - This guide provides guidance on the end-to-end deployment of a Red Hat OpenStack Platform environment. This includes installing the director, planning your environment, and creating an OpenStack environment with the director.
- Configuring Red Hat OpenStack Platform networking
  - This guide provides details on Red Hat OpenStack Platform networking tasks.
- Configuring persistent storage
  - This guide details the different procedures for using and managing persistent storage in a Red Hat OpenStack Platform environment. It also includes procedures for configuring and managing the respective OpenStack service of each persistent storage type.
- Configuring the Bare Metal Provisioning service
  - This guide provides details on the installation and configuration of the Bare Metal Provisioning service in the overcloud of a Red Hat OpenStack Platform environment to provision and manage physical machines for cloud users.
- Hardening Red Hat OpenStack Platform
  - This guide provides good practice advice and conceptual information about hardening the security of a Red Hat OpenStack Platform environment.
- Release Notes
  - This document outlines the major features, enhancements, and known issues in this release of Red Hat OpenStack Platform.