Red Hat Ceph Storage 5

Installation Guide

Installing Red Hat Ceph Storage on Red Hat Enterprise Linux

Last Updated: 2021-10-06

Red Hat Ceph Storage 5 Installation Guide

Installing Red Hat Ceph Storage on Red Hat Enterprise Linux

Legal Notice

Abstract

This document provides instructions on installing Red Hat Ceph Storage on Red Hat Enterprise Linux 8 running on AMD64 and Intel 64 architectures. 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.

Table of Contents

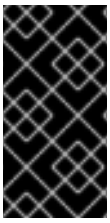
CHAPTER 1. RED HAT CEPH STORAGE

Red Hat Ceph Storage is a scalable, open, software-defined storage platform that combines an enterprise-hardened version of the Ceph storage system, with a Ceph management platform, deployment utilities, and support services.

Red Hat Ceph Storage is designed for cloud infrastructure and web-scale object storage. Red Hat Ceph Storage clusters consist of the following types of nodes:

Ceph Monitor

Each Ceph Monitor node runs the **ceph-mon** daemon, which maintains a master copy of the storage cluster map. The storage cluster map includes the storage cluster topology. A client connecting to the Ceph storage cluster retrieves the current copy of the storage cluster map from the Ceph Monitor, which enables the client to read from and write data to the storage cluster.



IMPORTANT

The storage cluster can run with only one Ceph Monitor; however, to ensure high availability in a production storage cluster, Red Hat will only support deployments with at least three Ceph Monitor nodes. Red Hat recommends deploying a total of 5 Ceph Monitors for storage clusters exceeding 750 Ceph OSDs.

Ceph Manager

The Ceph Manager daemon, **ceph-mgr**, co-exists with the Ceph Monitor daemons running on Ceph Monitor nodes to provide additional services. The Ceph Manager provides an interface for other monitoring and management systems using Ceph Manager modules. Running the Ceph Manager daemons is a requirement for normal storage cluster operations.

Ceph OSD

Each Ceph Object Storage Device (OSD) node runs the **ceph-osd** daemon, which interacts with logical disks attached to the node. The storage cluster stores data on these Ceph OSD nodes.

Ceph can run with very few OSD nodes, of which the default is three, but production storage clusters realize better performance beginning at modest scales. For example, 50 Ceph OSDs in a storage cluster. Ideally, a Ceph storage cluster has multiple OSD nodes, allowing for the possibility to isolate failure domains by configuring the CRUSH map accordingly.

Ceph MDS

Each Ceph Metadata Server (MDS) node runs the **ceph-mds** daemon, which manages metadata related to files stored on the Ceph File System (CephFS). The Ceph MDS daemon also coordinates access to the shared storage cluster.

Ceph Object Gateway

Ceph Object Gateway node runs the **ceph-radosgw** daemon, and is an object storage interface built on top of **librados** to provide applications with a RESTful access point to the Ceph storage cluster. The Ceph Object Gateway supports two interfaces:

- S3
  Provides object storage functionality with an interface that is compatible with a large subset of the Amazon S3 RESTful API.

- Swift

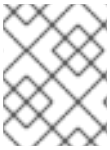
Provides object storage functionality with an interface that is compatible with a large subset of the OpenStack Swift API.

Additional Resources

- For details on the Ceph architecture, see the *Red Hat Ceph Storage Architecture Guide*.
- For the minimum hardware recommendations, see the *Red Hat Ceph Storage Hardware Selection Guide*.

CHAPTER 2. RED HAT CEPH STORAGE CONSIDERATIONS AND RECOMMENDATIONS

As a storage administrator, you can have a basic understanding about what things to consider before running a Red Hat Ceph Storage cluster. Understanding such things as, the hardware and network requirements, understanding what type of workloads work well with a Red Hat Ceph Storage cluster, along with Red Hat's recommendations. Red Hat Ceph Storage can be used for different workloads based on a particular business need or set of requirements. Doing the necessary planning before installing a Red Hat Ceph Storage is critical to the success of running a Ceph storage cluster efficiently, achieving the business requirements.



NOTE

Want help with planning a Red Hat Ceph Storage cluster for a specific use case? Please contact your Red Hat representative for assistance.

2.1. PREREQUISITES

- Time to understand, consider, and plan a storage solution.

2.2. BASIC RED HAT CEPH STORAGE CONSIDERATIONS

The first consideration for using Red Hat Ceph Storage is developing a storage strategy for the data. A storage strategy is a method of storing data that serves a particular use case. If you need to store volumes and images for a cloud platform like OpenStack, you can choose to store data on faster Serial Attached SCSI (SAS) drives with Solid State Drives (SSD) for journals. By contrast, if you need to store object data for an S3- or Swift-compliant gateway, you can choose to use something more economical, like traditional Serial Advanced Technology Attachment (SATA) drives. Red Hat Ceph Storage can accommodate both scenarios in the same storage cluster, but you need a means of providing the fast storage strategy to the cloud platform, and a means of providing more traditional storage for your object store.

One of the most important steps in a successful Ceph deployment is identifying a price-to-performance profile suitable for the storage cluster's use case and workload. It is important to choose the right hardware for the use case. For example, choosing IOPS-optimized hardware for a cold storage application increases hardware costs unnecessarily. Whereas, choosing capacity-optimized hardware for its more attractive price point in an IOPS-intensive workload will likely lead to unhappy users complaining about slow performance.

Red Hat Ceph Storage can support multiple storage strategies. Use cases, cost versus benefit performance tradeoffs, and data durability are the primary considerations that help develop a sound storage strategy.

Use Cases

Ceph provides massive storage capacity, and it supports numerous use cases, such as:

- The Ceph Block Device client is a leading storage backend for cloud platforms that provides limitless storage for volumes and images with high performance features like copy-on-write cloning.

- The Ceph Object Gateway client is a leading storage backend for cloud platforms that provides a RESTful S3-compliant and Swift-compliant object storage for objects like audio, bitmap, video and other data.

- The Ceph File System for traditional file storage.

Cost vs. Benefit of Performance

Faster is better. Bigger is better. High durability is better. However, there is a price for each superlative quality, and a corresponding cost versus benefit trade off. Consider the following use cases from a performance perspective: SSDs can provide very fast storage for relatively small amounts of data and journaling. Storing a database or object index can benefit from a pool of very fast SSDs, but proves too expensive for other data. SAS drives with SSD journaling provide fast performance at an economical price for volumes and images. SATA drives without SSD journaling provide cheap storage with lower overall performance. When you create a CRUSH hierarchy of OSDs, you need to consider the use case and an acceptable cost versus performance trade off.

Data Durability

In large scale storage clusters, hardware failure is an expectation, not an exception. However, data loss and service interruption remain unacceptable. For this reason, data durability is very important. Ceph addresses data durability with multiple replica copies of an object or with erasure coding and multiple coding chunks. Multiple copies or multiple coding chunks present an additional cost versus benefit tradeoff: it is cheaper to store fewer copies or coding chunks, but it can lead to the inability to service write requests in a degraded state. Generally, one object with two additional copies, or two coding chunks can allow a storage cluster to service writes in a degraded state while the storage cluster recovers.

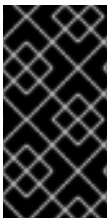
Replication stores one or more redundant copies of the data across failure domains in case of a hardware failure. However, redundant copies of data can become expensive at scale. For example, to store 1 petabyte of data with triple replication would require a cluster with at least 3 petabytes of storage capacity.

Erasure coding stores data as data chunks and coding chunks. In the event of a lost data chunk, erasure coding can recover the lost data chunk with the remaining data chunks and coding chunks. Erasure coding is substantially more economical than replication. For example, using erasure coding with 8 data chunks and 3 coding chunks provides the same redundancy as 3 copies of the data. However, such an encoding scheme uses approximately 1.5x of the initial data stored compared to 3x with replication.

The CRUSH algorithm aids this process by ensuring that Ceph stores additional copies or coding chunks in different locations within the storage cluster. This ensures that the failure of a single storage device or node does not lead to a loss of all of the copies or coding chunks necessary to preclude data loss. You can plan a storage strategy with cost versus benefit tradeoffs, and data durability in mind, then present it to a Ceph client as a storage pool.

IMPORTANT

ONLY the data storage pool can use erasure coding. Pools storing service data and bucket indexes use replication.



IMPORTANT

Ceph's object copies or coding chunks make RAID solutions obsolete. Do not use RAID, because Ceph already handles data durability, a degraded RAID has a negative impact on performance, and recovering data using RAID is substantially slower than using deep copies or erasure coding chunks.

Additional Resources

- See the *Minimum hardware considerations for Red Hat Ceph Storage* section of the *Red Hat Ceph Storage Installation Guide* for more details.

2.3. RED HAT CEPH STORAGE WORKLOAD CONSIDERATIONS

One of the key benefits of a Ceph storage cluster is the ability to support different types of workloads within the same storage cluster using performance domains. Different hardware configurations can be associated with each performance domain. Storage administrators can deploy storage pools on the appropriate performance domain, providing applications with storage tailored to specific performance and cost profiles. Selecting appropriately sized and optimized servers for these performance domains is an essential aspect of designing a Red Hat Ceph Storage cluster.

To the Ceph client interface that reads and writes data, a Ceph storage cluster appears as a simple pool where the client stores data. However, the storage cluster performs many complex operations in a manner that is completely transparent to the client interface. Ceph clients and Ceph object storage daemons, referred to as Ceph OSDs, or simply OSDs, both use the Controlled Replication Under Scalable Hashing (CRUSH) algorithm for storage and retrieval of objects. Ceph OSDs can run in containers within the storage cluster.

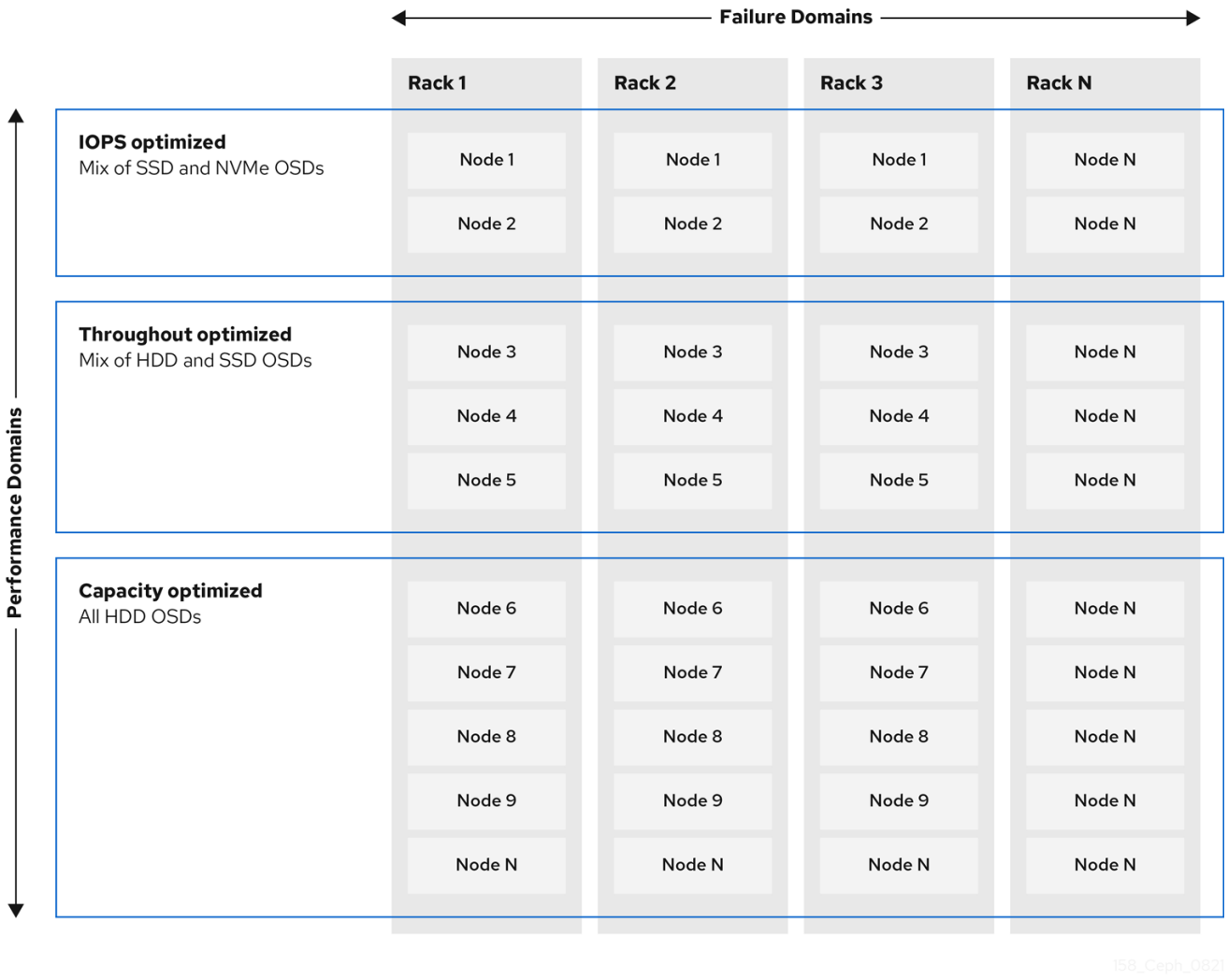
A CRUSH map describes a topography of cluster resources, and the map exists both on client nodes as well as Ceph Monitor nodes within the cluster. Ceph clients and Ceph OSDs both use the CRUSH map and the CRUSH algorithm. Ceph clients communicate directly with OSDs, eliminating a centralized object lookup and a potential performance bottleneck. With awareness of the CRUSH map and communication with their peers, OSDs can handle replication, backfilling, and recovery—allowing for dynamic failure recovery.

Ceph uses the CRUSH map to implement failure domains. Ceph also uses the CRUSH map to implement performance domains, which simply take the performance profile of the underlying hardware into consideration. The CRUSH map describes how Ceph stores data, and it is implemented as a simple hierarchy, specifically an acyclic graph, and a ruleset. The CRUSH map can support multiple hierarchies to separate one type of hardware performance profile from another. Ceph implements performance domains with device "classes".

For example, you can have these performance domains coexisting in the same Red Hat Ceph Storage cluster:

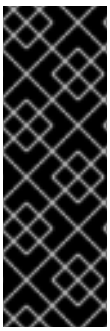
- Hard disk drives (HDDs) are typically appropriate for cost- and capacity-focused workloads.
- Throughput-sensitive workloads typically use HDDs with Ceph write journals on solid state drives (SSDs).
- IOPS-intensive workloads such as MySQL and MariaDB often use SSDs.

Figure 2.1. Performance and Failure Domains



Workloads

Red Hat Ceph Storage is optimized for three primary workloads.



IMPORTANT

Carefully consider the workload being run by Red Hat Ceph Storage clusters **BEFORE** considering what hardware to purchase, because it can significantly impact the price and performance of the storage cluster. For example, if the workload is capacity-optimized and the hardware is better suited to a throughput-optimized workload, then hardware will be more expensive than necessary. Conversely, if the workload is throughput-optimized and the hardware is better suited to a capacity-optimized workload, then the storage cluster can suffer from poor performance.

- **IOPS optimized:** Input, output per second (IOPS) optimization deployments are suitable for cloud computing operations, such as running MYSQL or MariaDB instances as virtual machines on OpenStack. IOPS optimized deployments require higher performance storage such as 15k RPM SAS drives and separate SSD journals to handle frequent write operations. Some high IOPS scenarios use all flash storage to improve IOPS and total throughput.
  An IOPS-optimized storage cluster has the following properties:
  - Lowest cost per IOPS.

- Highest IOPS per GB.
- 99th percentile latency consistency.

Uses for an IOPS-optimized storage cluster are:

- Typically block storage.
- 3x replication for hard disk drives (HDDs) or 2x replication for solid state drives (SSDs).
- MySQL on OpenStack clouds.

- **Throughput optimized:** Throughput-optimized deployments are suitable for serving up significant amounts of data, such as graphic, audio and video content. Throughput-optimized deployments require high bandwidth networking hardware, controllers and hard disk drives with fast sequential read and write characteristics. If fast data access is a requirement, then use a throughput-optimized storage strategy. Also, if fast write performance is a requirement, using Solid State Disks (SSD) for journals will substantially improve write performance.
  A throughput-optimized storage cluster has the following properties:
  - Lowest cost per MBps (throughput).
  - Highest MBps per TB.
  - Highest MBps per BTU.
  - Highest MBps per Watt.
  - 97th percentile latency consistency.

  Uses for an throughput-optimized storage cluster are:
  - Block or object storage.
  - 3x replication.
  - Active performance storage for video, audio, and images.
  - Streaming media, such as 4k video.

- **Capacity optimized:** Capacity-optimized deployments are suitable for storing significant amounts of data as inexpensively as possible. Capacity-optimized deployments typically trade performance for a more attractive price point. For example, capacity-optimized deployments often use slower and less expensive SATA drives and co-locate journals rather than using SSDs for journaling.
  A cost- and capacity-optimized storage cluster has the following properties:
  - Lowest cost per TB.
  - Lowest BTU per TB.
  - Lowest Watts required per TB.

  Uses for an cost- and capacity-optimized storage cluster are:
  - Typically object storage.
  - Erasure coding for maximizing usable capacity

- Object archive.
- Video, audio, and image object repositories.

2.4. NETWORK CONSIDERATIONS FOR RED HAT CEPH STORAGE

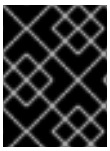
An important aspect of a cloud storage solution is that storage clusters can run out of IOPS due to network latency, and other factors. Also, the storage cluster can run out of throughput due to bandwidth constraints long before the storage clusters run out of storage capacity. This means that the network hardware configuration must support the chosen workloads in order to meet price versus performance requirements.

Storage administrators prefer that a storage cluster recovers as quickly as possible. Carefully consider bandwidth requirements for the storage cluster network, be mindful of network link oversubscription, and segregate the intra-cluster traffic from the client-to-cluster traffic. Also consider that network performance is increasingly important when considering the use of Solid State Disks (SSD), flash, NVMe, and other high performing storage devices.

Ceph supports a public network and a storage cluster network. The public network handles client traffic and communication with Ceph Monitors. The storage cluster network handles Ceph OSD heartbeats, replication, backfilling and recovery traffic. At a **minimum**, a single 10 GB Ethernet link should be used for storage hardware, and you can add additional 10 GB Ethernet links for connectivity and throughput.

IMPORTANT

Red Hat recommends allocating bandwidth to the storage cluster network, such that it is a multiple of the public network using **osd_pool_default_size** as the basis for the multiple on replicated pools. Red Hat also recommends running the public and storage cluster networks on separate network cards.



IMPORTANT

Red Hat recommends using 10 GB Ethernet for Red Hat Ceph Storage deployments in production. A 1 GB Ethernet network is not suitable for production storage clusters.

In the case of a drive failure, replicating 1 TB of data across a 1 GB Ethernet network takes 3 hours, and 3 TB takes 9 hours. Using 3 TB is the typical drive configuration. By contrast, with a 10 GB Ethernet network, the replication times would be 20 minutes and 1 hour respectively. Remember that when a Ceph OSD fails, the storage cluster will recover by replicating the data it contained to other Ceph OSDs within the pool.

The failure of a larger domain such as a rack means that the storage cluster will utilize considerably more bandwidth. When building a storage cluster consisting of multiple racks, which is common for large storage implementations, consider utilizing as much network bandwidth between switches in a "fat tree" design for optimal performance. A typical 10 GB Ethernet switch has 48 10 GB ports and four 40 GB ports. Use the 40 GB ports on the spine for maximum throughput. Alternatively, consider aggregating unused 10 GB ports with QSFP+ and SFP+ cables into more 40 GB ports to connect to other rack and spine routers. Also, consider using LACP mode 4 to bond network interfaces. Additionally, use jumbo frames, maximum transmission unit (MTU) of 9000, especially on the backend or cluster network.

Before installing and testing a Red Hat Ceph Storage cluster, verify the network throughput. Most performance-related problems in Ceph usually begin with a networking issue. Simple network issues like a kinked or bent Cat-6 cable could result in degraded bandwidth. Use a minimum of 10 GB ethernet for the front side network. For large clusters, consider using 40 GB ethernet for the backend or cluster network.

IMPORTANT

For network optimization, Red Hat recommends using jumbo frames for a better CPU per bandwidth ratio, and a non-blocking network switch back-plane. Red Hat Ceph Storage requires the same MTU value throughout all networking devices in the communication path, end-to-end for both public and cluster networks. Verify that the MTU value is the same on all nodes and networking equipment in the environment before using a Red Hat Ceph Storage cluster in production.

2.5. CONSIDERATIONS FOR USING A RAID CONTROLLER WITH OSD NODES

Optionally, you can consider using a RAID controller on the OSD nodes. Here are some things to consider:

- If an OSD node has a RAID controller with 1-2GB of cache installed, enabling the write-back cache might result in increased small I/O write throughput. However, the cache must be non-volatile.
- Most modern RAID controllers have super capacitors that provide enough power to drain volatile memory to non-volatile NAND memory during a power-loss event. It is important to understand how a particular controller and its firmware behave after power is restored.
- Some RAID controllers require manual intervention. Hard drives typically advertise to the operating system whether their disk caches should be enabled or disabled by default. However, certain RAID controllers and some firmware do not provide such information. Verify that disk level caches are disabled to avoid file system corruption.
- Create a single RAID 0 volume with write-back for each Ceph OSD data drive with write-back cache enabled.
- If Serial Attached SCSI (SAS) or SATA connected Solid-state Drive (SSD) disks are also present on the RAID controller, then investigate whether the controller and firmware support *pass-through* mode. Enabling *pass-through* mode helps avoid caching logic, and generally results in much lower latency for fast media.

2.6. TUNING CONSIDERATIONS FOR THE LINUX KERNEL WHEN RUNNING CEPH

Production Red Hat Ceph Storage clusters generally benefit from tuning the operating system, specifically around limits and memory allocation. Ensure that adjustments are set for all nodes within the storage cluster. You can also open a case with Red Hat support asking for additional guidance.

Reserving Free Memory for Ceph OSDs

To help prevent insufficient memory-related errors during Ceph OSD memory allocation requests, set the specific amount of physical memory to keep in reserve. Red Hat recommends the following settings based on the amount of system RAM.

- For 64 GB, reserve 1 GB:

```
vm.min_free_kbytes = 1048576
```

- For 128 GB, reserve 2 GB:

```
vm.min_free_kbytes = 2097152
```

- For 256 GB, reserve 3 GB:

```
vm.min_free_kbytes = 3145728
```

Increase the File Descriptors

The Ceph Object Gateway can hang if it runs out of file descriptors. You can modify the **/etc/security/limits.conf** file on Ceph Object Gateway nodes to increase the file descriptors for the Ceph Object Gateway.

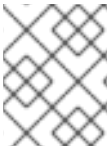
```
ceph      soft    nofile     unlimited
```

Adjusting the ulimit value for Large Storage Clusters

When running Ceph administrative commands on large storage clusters, for example, with 1024 Ceph OSDs or more, create an **/etc/security/limits.d/50-ceph.conf** file on each node that runs administrative commands with the following contents:

```
USER_NAME      soft    nproc     unlimited
```

Replace *USER_NAME* with the name of the non-root user account that runs the Ceph administrative commands.



NOTE

The root user's **ulimit** value is already set to **unlimited** by default on Red Hat Enterprise Linux.

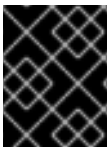
2.7. OPERATING SYSTEM REQUIREMENTS FOR RED HAT CEPH STORAGE

Red Hat Enterprise Linux entitlements are included in the Red Hat Ceph Storage subscription.

The initial release of Red Hat Ceph Storage 5 is supported on Red Hat Enterprise Linux 8.4 or higher.

Red Hat Ceph Storage 5 is supported on container-based deployments only.

Use the same operating system version, architecture, and deployment type across all nodes. For example, do not use a mixture of nodes with both AMD64 and Intel 64 architectures, a mixture of nodes with Red Hat Enterprise Linux 8 operating systems, or a mixture of nodes with container-based deployments.



IMPORTANT

Red Hat does not support clusters with heterogeneous architectures, operating system versions, or deployment types.

SELinux

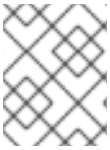
By default, SELinux is set to **Enforcing** mode and the **ceph-selinux** packages are installed. For additional information on SELinux please see the *Data Security and Hardening Guide* , and *Red Hat Enterprise Linux 8 Using SELinux Guide*.

Additional Resources

- The documentation set for Red Hat Enterprise Linux 8.

2.8. MINIMUM HARDWARE CONSIDERATIONS FOR RED HAT CEPH STORAGE

Red Hat Ceph Storage can run on non-proprietary commodity hardware. Small production clusters and development clusters can run without performance optimization with modest hardware.



NOTE

Disk space requirements are based on the Ceph daemons' default path under **/var/lib/ceph/** directory.

Table 2.1. Containers

Process	Criteria	Minimum Recommended
ceph-osd-container	Processor	1x AMD64 or Intel 64 CPU CORE per OSD container
	RAM	Minimum of 5 GB of RAM per OSD container
	OS Disk	1x OS disk per host
	OSD Storage	1x storage drive per OSD container. Cannot be shared with OS Disk.
	block.db	Optional, but Red Hat recommended, 1x SSD or NVMe or Optane partition or lvm per daemon. Sizing is 4% of **block.data** for BlueStore for object, file and mixed workloads and 1% of **block.data** for the BlueStore for Block Device, Openstack cinder, and Openstack cinder workloads.
	block.wal	Optionally, 1x SSD or NVMe or Optane partition or logical volume per daemon. Use a small size, for example 10 GB, and only if it's faster than the **block.db** device.
	Network	2x 10 GB Ethernet NICs, 10 GB Recommended
ceph-mon-container	Processor	1x AMD64 or Intel 64 CPU CORE per mon-container
	RAM	3 GB per **mon-container**
	Disk Space	10 GB per **mon-container**, 50 GB Recommended

Process	Criteria	Minimum Recommended
	Monitor Disk	Optionally, 1x SSD disk for **Monitor rocksdb** data
	Network	2x 1 GB Ethernet NICs, 10 GB Recommended
ceph-mgr-container	Processor	1x AMD64 or Intel 64 CPU CORE per **mgr-container**
	RAM	3 GB per **mgr-container**
	Network	2x 1 GB Ethernet NICs, 10 GB Recommended
ceph-radosgw-container	Processor	1x AMD64 or Intel 64 CPU CORE per radosgw-container
	RAM	1 GB per daemon
	Disk Space	5 GB per daemon
	Network	1x 1 GB Ethernet NICs
ceph-mds-container	Processor	1x AMD64 or Intel 64 CPU CORE per mds-container
	RAM	3 GB per **mds-container** This number is highly dependent on the configurable MDS cache size. The RAM requirement is typically twice as much as the amount set in the **mds_cache_memory_limit** configuration setting. Note also that this is the memory for your daemon, not the overall system memory.
	Disk Space	2 GB per **mds-container**, plus taking into consideration any additional space required for possible debug logging, 20GB is a good start.
	Network	2x 1 GB Ethernet NICs, 10 GB Recommended Note that this is the same network as the OSD containers. If you have a 10 GB network on your OSDs you should use the same on your MDS so that the MDS is not disadvantaged when it comes to latency.

2.9. ADDITIONAL RESOURCES

- If you want to take a deeper look into Ceph's various internal components, and the strategies around those components, see the *Red Hat Ceph Storage Storage Strategies Guide* for more details.

CHAPTER 3. RED HAT CEPH STORAGE INSTALLATION

As a storage administrator, you can use the **cephadm** utility to deploy new Red Hat Ceph Storage clusters.

The **cephadm** utility manages the entire life cycle of a Ceph cluster. Installation and management tasks comprise two types of operations:

- Day One operations involve installing and bootstrapping a bare-minimum, containerized Ceph storage cluster, running on a single node. Day One also includes deploying the Monitor and Manager daemons and adding Ceph OSDs.
- Day Two operations use the Ceph orchestration interface, **cephadm orch**, or the Red Hat Ceph Storage Dashboard to expand the storage cluster by adding other Ceph services to the storage cluster.

3.1. PREREQUISITES

- At least one running virtual machine (VM) or bare-metal server with an active internet connection.
- Red Hat Enterprise Linux 8.4 or later.
- Ansible 2.9 or later.
- A valid Red Hat subscription with the appropriate entitlements.
- Root-level access to all nodes.
- An active Red Hat Network (RHN) or service account to access the Red Hat Registry.

3.2. THE CEPHADM UTILITY

The **cephadm** utility deploys and manages a Ceph storage cluster. It is tightly integrated with both the command-line interface (CLI) and the Red Hat Ceph Storage Dashboard web interface, so that you can manage storage clusters from either environment. **cephadm** uses SSH to connect to hosts from the manager daemon to add, remove, or update Ceph daemon containers. It does not rely on external configuration or orchestration tools such as Ansible or Rook.

The **cephadm** utility consists of two main components:

- The **cephadm** shell.
- The **cephadm** orchestrator.

The cephadm shell

The **cephadm** shell launches a **bash** shell within a container. This enables you to perform "Day One" cluster setup tasks, such as installation and bootstrapping, and to invoke **ceph** commands.

There are two ways to invoke the **cephadm** shell:

- Enter **cephadm shell** at the system prompt:

  Example

  ■

```
[root@node00 ~]# cephadm shell
[cephadm@cephadm ~]# ceph -s
```

- At the system prompt, type **cephadm shell** and the command you want to execute:

  Example

```
[root@node00 ~]# cephadm shell ceph -s
```

NOTE

If the node contains configuration and keyring files in /**etc**/**ceph**/, the container environment uses the values in those files as defaults for the **cephadm** shell. However, if you execute the **cephadm** shell on a Ceph Monitor node, the **cephadm** shell inherits its default configuration from the Ceph Monitor container, instead of using the default configuration.

The cephadm orchestrator

The **cephadm** orchestrator enables you to perform "Day Two" Ceph functions, such as expanding the storage cluster and provisioning Ceph daemons and services. You can use the **cephadm** orchestrator through either the command-line interface (CLI) or the web-based Red Hat Ceph Storage Dashboard. Orchestrator commands take the form **ceph orch**.

The **cephadm** script interacts with the Ceph orchestration module used by the Ceph Manager.

3.3. HOW CEPHADM WORKS

The **cephadm** command manages the full lifecycle of a Red Hat Ceph Storage cluster. The **cephadm** command can perform the following operations:

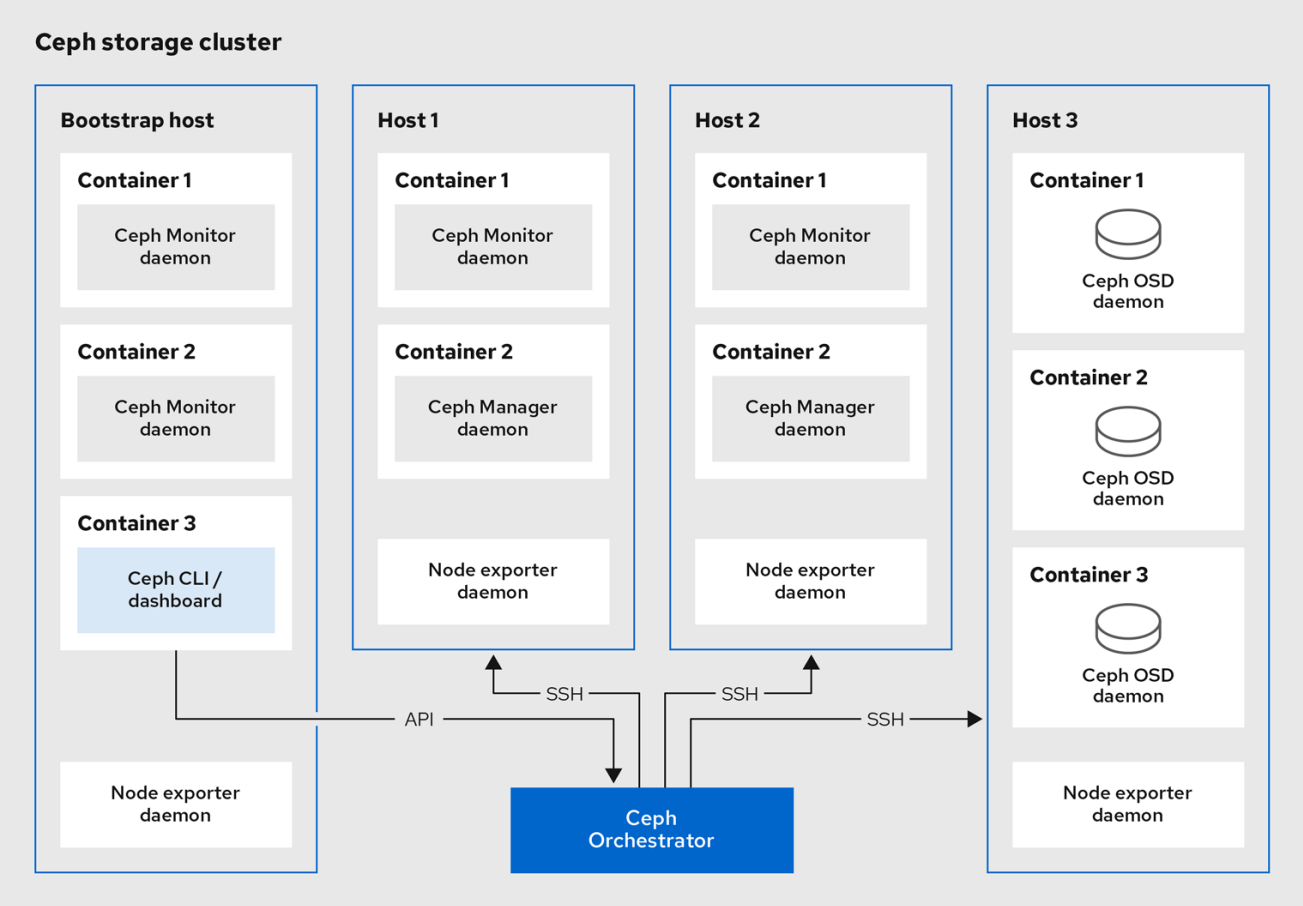
- Bootstrap a new Red Hat Ceph Storage cluster.
- Launch a containerized shell that works with the Red Hat Ceph Storage command-line interface (CLI).
- Aid in debugging containerized daemons.

The **cephadm** command uses **ssh** to communicate with the nodes in the storage cluster. This allows you to add, remove, or update Red Hat Ceph Storage containers without using external tools. Generate the **ssh** key pair during the bootstrapping process, or use your own **ssh** key.

The **cephadm** bootstrapping process creates a small storage cluster on a single node, consisting of one Ceph Monitor and one Ceph Manager, as well as any required dependencies. You then use the orchestrator CLI or the Red Hat Ceph Storage Dashboard to expand the storage cluster to include nodes, and to provision all of the Red Hat Ceph Storage daemons and services. You can perform management functions through the CLI or from the Red Hat Ceph Storage Dashboard web interface.

NOTE

The **cephadm** utility is a new feature in Red Hat Ceph Storage 5.0. It does not support older versions of Red Hat Ceph Storage.



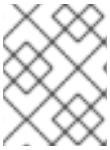
3.4. REGISTERING THE RED HAT CEPH STORAGE NODES TO THE CDN AND ATTACHING SUBSCRIPTIONS

NOTE

Red Hat Ceph Storage supports Red Hat Enterprise Linux 8.4 and later.

Prerequisites

- At least one running virtual machine (VM) or bare-metal server with an active internet connection.
- Red Hat Enterprise Linux 8.4 or later.
- A valid Red Hat subscription with the appropriate entitlements.
- Root-level access to all nodes.
- An active Red Hat Network (RHN) or service account to access the Red Hat Registry.



NOTE

The Red Hat Registry is located at https://registry.redhat.io/. Nodes require connectivity to the registry.

Procedure

1. Register the node, and when prompted, enter your Red Hat Customer Portal credentials:

   Syntax

   ```
   subscription-manager register
   ```

2. Pull the latest subscription data from the CDN:

   Syntax

   ```
   subscription-manager refresh
   ```

3. List all available subscriptions for Red Hat Ceph Storage:

   Syntax

   ```
   subscription-manager list --available --matches 'Red Hat Ceph Storage'
   ```

4. Identify the appropriate subscription and retrieve its Pool ID.

5. Attach a pool ID to gain access to the software entitlements. Use the Pool ID you identified in the previous step.

   Syntax

   ```
   subscription-manager attach --pool=POOL_ID
   ```

6. Disable the default software repositories, and then enable the server and the extras repositories on the respective version of Red Hat Enterprise Linux:

   Syntax

   ```
   subscription-manager repos --disable=*
   subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
   subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
   ```

7. Update the system to receive the latest packages for Red Hat Enterprise Linux 8:

   Syntax

   ```
   # dnf update
   ```

8. Subscribe to Red Hat Ceph Storage 5.0 content. Follow the instructions in How to Register Ceph with Red Hat Satellite 6.

9. Enable the **ceph-tools** repository:

   Syntax

   ```
   subscription-manager repos --enable=rhceph-5-tools-for-rhel-8-x86_64-rpms
   ```

10. Enable the ansible repository:

    Syntax

    ```
    subscription-manager repos --enable=ansible-2.9-for-rhel-8-x86_64-rpms
    ```

11. Install **cephadm-ansible**:

    Syntax

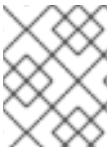
    ```
    dnf install cephadm-ansible
    ```

Additional Resources

- Red Hat Registry

3.5. CONFIGURING ANSIBLE INVENTORY LOCATION

You can configure inventory location files for the **cephadm-ansible** staging and production environments.



NOTE

cephadm-ansible only recognizes two groups of hosts in its inventory **hosts** file: [admin] and [clients]. If other host types appear in the file, **cephadm-ansible** ignores them.

Prerequisites

- An Ansible administration node.
- Root-level access to the Ansible administration node.
- The **cephadm-ansible** package is installed on the node.

Procedure

1. Navigate to the **/usr/share/cephadm-ansible** directory:

   ```
   [root@admin ~]# cd /usr/share/cephadm-ansible
   ```

2. Optional: Create subdirectories for staging and production:

   ```
   [root@admin cephadm-ansible]# mkdir -p inventory/staging inventory/production
   ```

3. Optional: Edit the **ansible.cfg** file and add the following line to assign a default inventory location:

   ```
   [defaults]
   inventory = ./inventory/staging
   ```

4. Optional: Create an inventory **hosts** file for each environment:

```
[root@admin cephadm-ansible]# touch inventory/staging/hosts
[root@admin cephadm-ansible]# touch inventory/production/hosts
```

5. Open and edit each **hosts** file and add the admin and client nodes:

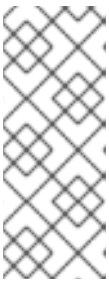
```
[admin]
ADMIN_NODE_NAME_1

[clients]
CLIENT_NAME_1
CLIENT_NAME_2
CLIENT_NAME_3
```

Example

```
[admin]
node00

[clients]
client01
client02
client03
```



NOTE

By default, playbooks run in the staging environment. To run the playbook in the production environment:

```
[root@admin cephadm-ansible]# ansible-playbook -i inventory/production playbook.yml
```

3.6. ENABLING PASSWORD-LESS SSH FOR ANSIBLE

Generate an SSH key pair on the Ansible administration node and distribute the public key to each node in the storage cluster so that Ansible can access the nodes without being prompted for a password.

Prerequisites

- Access to the Ansible administration node.

Procedure

1. Generate the SSH key pair, accept the default file name and leave the passphrase empty:

```
[ansible@admin ~]$ ssh-keygen
```

2. Copy the public key to all nodes in the storage cluster:

```
ssh-copy-id USER_NAME@HOST_NAME
```

Replace

- *USER_NAME* with the new user name for the Ansible user

- *USER_NAME* with the new user name for the Ansible user.
- *HOST_NAME* with the host name of the Ceph node.

Example

```
[ansible@admin ~]$ ssh-copy-id ceph-admin@ceph-mon01
```

3. Create the user's SSH **config** file:

```
[ansible@admin ~]$ touch ~/.ssh/config
```

4. Open for editing the **config** file. Set values for the **Hostname** and **User** options for each node in the storage cluster:

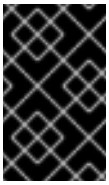
```
Host node1
  Hostname HOST_NAME
  User USER_NAME
Host node2
  Hostname HOST_NAME
  User USER_NAME
...
```

Replace

- *HOST_NAME* with the host name of the Ceph node.
- *USER_NAME* with the new user name for the Ansible user.

Example

```
Host node1
  Hostname monitor
  User admin
Host node2
  Hostname osd
  User admin
Host node3
  Hostname gateway
  User admin
```



IMPORTANT

By configuring the **~/.ssh/config** file you do not have to specify the **-u *USER_NAME*** option each time you execute the **ansible-playbook** command.

5. Set the correct file permissions for the **~/.ssh/config** file:

```
[admin@admin ~]$ chmod 600 ~/.ssh/config
```

Additional Resources

- The **ssh_config(5)** manual page.
- See Using secure communications between two systems with OpenSSH .

3.7. RUNNING THE PREFLIGHT PLAYBOOK

This Ansible playbook configures the Ceph repository and prepares the storage cluster for bootstrapping. It also installs some prerequisites, such as **podman**, **lvm2**, **chronyd**, and **cephadm**. The default location for **cephadm-ansible** and **cephadm-preflight.yml** is **/usr/share/cephadm-ansible**.

The preflight playbook uses the **cephadm-ansible** inventory file to identify all the admin and client nodes in the storage cluster.

The default location for the inventory file is **/usr/share/cephadm-ansible/hosts**. The following example shows the structure of a typical inventory file:

+

```
[admin]
node00

[clients]
client01
client02
client03
```

The [admin] group in the inventory file contains the name of the node where the admin keyring is stored.

NOTE

Run the preflight playbook before you bootstrap the initial host.

Prerequisites

- Ansible is installed on the host.
- Root-level access to all nodes in the storage cluster.

Procedure

1. Navigate to the **/usr/share/cephadm-ansible** directory.
2. Run the preflight playbook on the initial host in the storage cluster:

   Syntax

   ```
   ansible-playbook -i INVENTORY-FILE cephadm-preflight.yml --extra-vars
   "ceph_origin=rhcs"
   ```

   Example

   ```
   [root@admin ~]# ansible-playbook -i /usr/share/cephadm-ansible/hosts/ cephadm-
   preflight.yml --extra-vars "ceph_origin=rhcs"
   ```

After installation is complete, **cephadm** resides in the **/usr/sbin/** directory.

- Use the **--limit** option to run the preflight playbook on a selected set of hosts in the storage cluster:

  Syntax

  ```
  ansible-playbook -i INVENTORY-FILE cephadm-preflight.yml --extra-vars
  "ceph_origin=rhcs" --limit OSD_GROUP|NODE_NAME
  ```

  Example

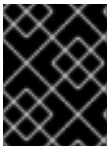
  ```
  [root@admin ~]# ansible-playbook -i /usr/share/cephadm-ansible/hosts/ cephadm-
  preflight.yml --extra-vars "ceph_origin=rhcs" --limit my-osd-group
  [root@admin ~]# ansible-playbook -i /usr/share/cephadm-ansible/hosts/ cephadm-
  preflight.yml --extra-vars "ceph_origin=rhcs" --limit my-nodes
  ```

- When you run the preflight playbook, **cephadm-ansible** automatically installs **chronyd** and **ceph-common** on the client nodes.

3.8. BOOTSTRAPPING A NEW STORAGE CLUSTER

The **cephadm** utility performs the following tasks during the bootstrap process:

- Installs and starts a Ceph Monitor daemon and a Ceph Manager daemon for a new Red Hat Ceph Storage cluster on the local node as containers.
- Creates the **/etc/ceph** directory.
- Writes a copy of the public key to **/etc/ceph/ceph.pub** for the Red Hat Ceph Storage cluster and adds the SSH key to the root user's **/root/.ssh/authorized_keys** file.
- Writes a minimal configuration file needed to communicate with the new cluster to **/etc/ceph/ceph.conf**.
- Writes a copy of the **client.admin** administrative secret key to **/etc/ceph/ceph.client.admin.keyring**.
- Deploys a basic monitoring stack with prometheus, grafana, and other tools such as **node-exporter** and **alert-manager**.



IMPORTANT

If you are performing a disconnected installation, see Performing a disconnected installation.

NOTE

If you have existing prometheus services that you want to run with the new storage cluster, or if you are running Ceph with Rook, use the **--skip-monitoring-stack** option with the **cephadm bootstrap** command. This option bypasses the basic monitoring stack so that you can manually configure it later.

IMPORTANT

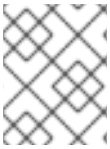
Bootstrapping provides the default user name and password for initial login to the Dashboard. Bootstrap requires you to change the password after you log in.

IMPORTANT

Before you begin the bootstrapping process, make sure that the container image that you want to use has the same version of Red Hat Ceph Storage as **cephadm**. If the two versions do not match, bootstrapping fails at the **Creating initial admin user** stage.

Prerequisites

- An IP address for the first Ceph Monitor container, which is also the IP address for the first node in the storage cluster.
- Login to **registry.redhat.io** on all the nodes of the storage cluster.
- A minimum of 10 GB of free space for /**var**/**lib**/**containers**/.
- Root-level access to all nodes.

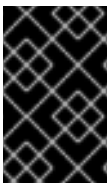


NOTE

If the storage cluster includes multiple networks and interfaces, be sure to choose a network that is accessible by any node that uses the storage cluster.

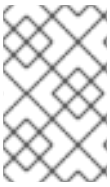
NOTE

If the local node uses fully-qualified domain names (FQDN), then add the **--allow-fqdn-hostname** option to **cephadm bootstrap** on the command line.



IMPORTANT

Run **cephadm bootstrap** on the node that you want to be the initial Monitor node in the cluster. The *IP_ADDRESS* option should be the IP address of the node you are using to run **cephadm bootstrap**.



NOTE

If you want to deploy a storage cluster using IPV6 addresses, then use the IPV6 address format for the **--mon-ip *IP-ADDRESS* option. For example: `cephadm bootstrap --mon-ip 2620:52:0:880:225:90ff:fefc:2536 --registry-json /etc/mylogin.json**

Procedure

1. Bootstrap a storage cluster:

   Syntax

   ```
   cephadm bootstrap --mon-ip IP_ADDRESS --registry-url registry.redhat.io --registry-username USER_NAME --registry-password PASSWORD
   ```

   Example

```
[root@vm00 ~]# cephadm bootstrap --mon-ip 10.10.128.68 --registry-url registry.redhat.io --
registry-username myuser1 --registry-password mypassword1
```

The script takes a few minutes to complete. Once the script completes, it provides the credentials to the Red Hat Ceph Storage Dashboard URL, a command to access the Ceph command-line interface (CLI), and a request to enable telemetry.

```
Ceph Dashboard is now available at:

             URL: https://rh8-3.storage.lab:8443/
            User: admin
        Password: i8nhu7zham

You can access the Ceph CLI with:

        sudo /usr/sbin/cephadm shell --fsid 266ee7a8-2a05-11eb-b846-5254002d4916 -c
/etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring

Please consider enabling telemetry to help improve Ceph:

        ceph telemetry on

For more information see:

        https://docs.ceph.com/docs/master/mgr/telemetry/

Bootstrap complete.
```

Additional Resources

- For more information about the recommended bootstrap command options, see Recommended bootstrap command options .
- For more information about the options available for the bootstrap command, see Bootstrap command options.
- For information about using a JSON file to contain login credentials for the bootstrap process, see Using a JSON file to protect login information .

3.8.1. Using a JSON file to protect login information

As a storage administrator, you might choose to add login and password information to a JSON file, and then refer to the JSON file for bootstrapping. This protects the login credentials from exposure.

NOTE

You can also use a JSON file with the **cephadm --registry-login** command.

Prerequisites

- An IP address for the first Ceph Monitor container, which is also the IP address for the first node in the storage cluster.
- Login access to **registry.redhat.io**.

- A minimum of 10 GB of free space for **/var/lib/containers/**.
- Root-level access to all nodes.

Procedure

1. Create the JSON file. In this example, the file is named **mylogin.json**.

   Syntax

   ```
   {
    "url":"REGISTRY-URL",
    "username":"USER-NAME",
    "password":"PASSWORD"
   }
   ```

   Example

   ```
   {
    "url":"registry.redhat.io",
    "username":"myuser1",
    "password":"mypassword1"
   }
   ```

2. Bootstrap a storage cluster:

   Syntax

   ```
   cephadm bootstrap --mon-ip IP_ADDRESS --registry-json /etc/mylogin.json
   ```

   Example

   ```
   [root@vm00 ~]# cephadm bootstrap --mon-ip 10.10.128.68 --registry-json /etc/mylogin.json
   ```

3.8.2. Bootstrapping a storage cluster using a service configuration file

As a storage administrator, you can use a service configuration file and the **--apply-spec** option to bootstrap the storage cluster and configure additional hosts and daemons. The configuration file is a **.yaml** file that contains the service type, placement, and designated nodes for services that you want to deploy.

NOTE

If you want to use a non-default realm or zone for applications such as multisite, configure your RGW daemons after you bootstrap the storage cluster, instead of adding them to the configuration file and using the **--apply-spec** option. This gives you the opportunity to create the realm or zone you need for the Ceph Object Gateway daemons before deploying them. Refer to the *Red Hat Ceph Storage Operations Guide* for more information.

Prerequisites

- At least one running virtual machine (VM) or server.

- Red Hat Enterprise Linux 8.4 or later.
- Root-level access to all nodes.
- Passwordless **ssh** is set up on all hosts in the storage cluster.
- **cephadm** is installed on the node that you want to be the initial Monitor node in the storage cluster.

Procedure

1. Log in to the bootstrap host.
2. Create the service configuration **.yaml** file for your storage cluster. The example file directs **cephadm bootstrap** to configure the initial host and two additional hosts, and it specifies that OSDs be created on all available disks.

   Example

   ```
   service_type: host
   addr: node-00
   hostname: node-00
   ---
   service_type: host
   addr: node-01
   hostname: node-01
   ---
   service_type: host
   addr: node-02
   hostname: node-02
   ---
   service_type: osd
   placement:
     host_pattern: "*"
   data_devices:
     all: true
   ```

3. Bootstrap the storage cluster with the **--apply-spec** option:

   Syntax

   ```
   cephadm bootstrap --apply-spec CONFIGURATION_FILE_NAME --mon-ip MONITOR-IP-ADDRESS
   ```

   Example

   ```
   [root@vm00 ~]# cephadm bootstrap --apply-spec initial-config.yaml --mon-ip 10.10.128.68
   ```

   The script takes a few minutes to complete. Once the script completes, it provides the credentials to the Red Hat Ceph Storage Dashboard URL, a command to access the Ceph command-line interface (CLI), and a request to enable telemetry.

4. Once your storage cluster is up and running, refer to the *Red Hat Ceph Storage Operations Guide* for more information about configuring additional daemons and services.

Additional Resources

- For more information about configuring additional daemons and services, refer to the *Red Hat Ceph Storage Operations Guide*
- For more information about the options available for the bootstrap command, see Bootstrap command options.

3.8.3. Bootstrap command options

The **cephadm bootstrap** command bootstraps a Ceph storage cluster on the local host. It deploys a MON daemon and a MGR daemon on the bootstrap node, automatically deploys the monitoring stack on the local host, and calls **ceph orch host add *HOSTNAME***.

The following table lists the available options for **cephadm bootstrap**.

cephadm bootstrap option	Description
--config *CONFIG-FILE*, -c *CONFIG-FILE*	*CONFIG-FILE* is the **ceph.conf** file to use with the bootsrap command
--mon-id *MON-ID*	Bootstraps on the host named *MON-ID*. Default value is the local host.
--mon-addrv *MON-ADDRV*	mon IPs (e.g., [v2:localipaddr:3300,v1:localipaddr:6789])
--mon-ip *IP-ADDRESS*	IP address of the node you are using to run **cephadm bootstrap**.
--mgr-id *MGR_ID*	Host ID where a MGR node should be installed. Default: randomly generated.
--fsid *FSID*	cluster FSID
--output-dir *OUTPUT_DIR*	Use this directory to write config, keyring, and pub key files.
--output-keyring *OUTPUT_KEYRING*	Use this location to write the keyring file with the new cluster admin and mon keys.
--output-config *OUTPUT_CONFIG*	Use this location to write the configuration file to connect to the new cluster.
--output-pub-ssh-key *OUTPUT_PUB_SSH_KEY*	Use this location to write the public SSH key for the cluster.
--skip-ssh	Skip the setup of the ssh key on the local host.
--initial-dashboard-user *INITIAL_DASHBOARD_USER*	Initial user for the dashboard.

cephadm bootstrap option	Description
--initial-dashboard-password *INITIAL_DASHBOARD_PASSWORD*	Initial password for the initial dashboard user.
--ssl-dashboard-port *SSL_DASHBOARD_PORT*	Port number used to connect with the dashboard using SSL.
--dashboard-key *DASHBOARD_KEY*	Dashboard key.
--dashboard-crt *DASHBOARD_CRT*	Dashboard certificate.
--ssh-config *SSH_CONFIG*	SSH config.
--ssh-private-key *SSH_PRIVATE_KEY*	SSH private key.
--ssh-public-key *SSH_PUBLIC_KEY*	SSH public key.
--ssh-user *SSH_USER*	sets the user for SSH connections to cluster hosts. Passwordless sudo is needed for non-root users.
--skip-mon-network	Sets mon public_network based on the bootstrap mon ip.
--skip-dashboard	Do not enable the Ceph Dashboard.
--dashboard-password-noupdate	Disable forced dashboard password change.
--no-minimize-config	Do not assimilate and minimize the configuration file.
--skip-ping-check	Do not verify that the mon IP is pingable.
--skip-pull	Do not pull the latest image before bootstrapping.
--skip-firewalld	Do not configure firewalld.
--allow-overwrite	Allow the overwrite of existing –output-* config/keyring/ssh files.
--allow-fqdn-hostname	Allow fully qualified host name.
--skip-prepare-host	Do not prepare host.
--orphan-initial-daemons	Do not create initial mon, mgr, and crash service specs.

cephadm bootstrap option	Description
--skip-monitoring-stack	Do not automatically provision the monitoring stack] (prometheus, grafana, alertmanager, node-exporter).
--apply-spec *APPLY_SPEC*	Apply cluster spec file after bootstrap (copy ssh key, add hosts and apply services).
--registry-url *REGISTRY_URL*	Specifies the URL of the custom registry to log into. For example: **registry.redhat.io**.
--registry-username *REGISTRY_USERNAME*	User name of the login account to the custom registry.
--registry-password *REGISTRY_PASSWORD*	Password of the login account to the custom registry.
--registry-json *REGISTRY_JSON*	JSON file containing registry login information.

Additional Resources

- For more information about the **--skip-monitoring-stack** option, see Adding hosts.
- For more information about logging into the registry with the **registry-json** option, see help for the **registry-login** command.
- For more information about **cephadm** options, see help for **cephadm**.

3.8.4. Configuring a custom registry for disconnected installation

You can use a disconnected installation procedure to install **cephadm** and bootstrap your cluster on a private network. A disconnected installation uses a custom container registry for installation.

Prerequisites

- At least one running virtual machine (VM) or server.
- Red Hat Enterprise Linux 8.4 or later.
- Root-level access to all nodes.
- Passwordless **ssh** is set up on all hosts in the storage cluster.
- A Red Hat Ceph Storage container image.
- The container image resides in the custom registry.
- Docker for Red Hat Enterprise Linux 7 or podman for Red Hat Enterprise Linux 8 is installed. For Red Hat Enterprise Linux 7, the docker service is running.

Procedure

Use this procedure when the Red Hat Ceph Storage nodes do NOT have access to the Internet during deployment. Perform these steps on a node that has both Internet access and access to the local cluster.

1. Log in to the node that has access to both public network and the cluster nodes.
2. Register the node, and when prompted, enter the appropriate Red Hat Customer Portal credentials:

   Syntax

   ```
   subscription-manager register
   ```

3. Pull the latest subscription data:

   Syntax

   ```
   subscription-manager refresh
   ```

4. List all available subscriptions for Red Hat Ceph Storage:

   ```
   subscription-manager list --available --all --matches="*Ceph*"
   ```

   Copy the Pool ID from the list of available subscriptions for Red Hat Ceph Storage.

5. Attach the subscription to get access to the software entitlements.:

   Syntax

   ```
   subscription-manager attach --pool=POOL_ID
   ```

   Replace

   - *POOL_ID* with the Pool ID identified in the previous step.

6. Disable the default software repositories, and enable the server and the extras repositories on the respective version of Red Hat Enterprise Linux:

   Red Hat Enterprise Linux 7

   ```
   subscription-manager repos --disable=*
   subscription-manager repos --enable=rhel-7-server-rpms
   subscription-manager repos --enable=rhel-7-server-extras-rpms
   ```

   Red Hat Enterprise Linux 8

   ```
   subscription-manager repos --disable=*
   subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
   subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
   ```

7. Install the container runtimes:

   Red Hat Enterprise Linux 7:

```
yum install docker
```

Red Hat Enterprise Linux 8:

```
dnf install podman
```

8. Update the system to receive the latest packages.

   Red Hat Enterprise Linux 7:

   ```
   yum update
   ```

   Red Hat Enterprise Linux 8:

   ```
   dnf update
   ```

9. Start a local registry. Use docker for Red Hat Enterprise Linux 7 or podman for Red Hat Enterprise Linux 8:

   Red Hat Enterprise Linux 7

   ```
   docker run -d -p 5000:5000 --restart=always --name registry registry:2
   ```

   Red Hat Enterprise Linux 8

   ```
   podman run -d -p 5000:5000 --restart=always --name registry registry:2
   ```

10. Verify that **registry.redhat.io** is in the container registry search path.

    a. Edit the **/etc/containers/registries.conf** file:

       Example

       ```
       [registries.search]
       registries = [ 'registry.access.redhat.com', 'registry.fedoraproject.org',
       'registry.centos.org', 'docker.io']
       ```

    b. If **registry.redhat.io** is not included in the file, add it:

       Example

       ```
       [registries.search]
       registries = ['registry.redhat.io', 'registry.access.redhat.com', 'registry.fedoraproject.org',
       'registry.centos.org', 'docker.io']
       ```

11. Pull the Red Hat Ceph Storage 5.0 image, Prometheus image, and Dashboard image from the Red Hat Customer Portal:

    Red Hat Enterprise Linux 7

    ```
    # docker pull registry.redhat.io/rhceph/rhceph-5-rhel8:latest
    ```

```
# docker pull registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
# docker pull registry.redhat.io/rhceph/rhceph-5-dashboard-rhel8:latest
# docker pull registry.redhat.io/openshift4/ose-prometheus:v4.6
# docker pull registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
```

Red Hat Enterprise Linux 8

```
# podman pull registry.redhat.io/rhceph/rhceph-5-rhel8:latest
# podman pull registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
# podman pull registry.redhat.io/rhceph/rhceph-5-dashboard-rhel8:latest
# podman pull registry.redhat.io/openshift4/ose-prometheus:v4.6
# podman pull registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
```

12. Tag the image. Replace ***LOCAL_NODE_FQDN*** with your local host Fully Qualified Domain Name (FQDN):

Red Hat Enterprise Linux 7

```
# docker tag registry.redhat.io/rhceph/rhceph-5-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-rhel8:latest
# docker tag registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
# docker tag registry.redhat.io/rhceph/rhceph-5-dashboard-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-dashboard-rhel8:latest
# docker tag registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
# docker tag registry.redhat.io/openshift4/ose-prometheus:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

Red Hat Enterprise Linux 8

```
# podman tag registry.redhat.io/rhceph/rhceph-5-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-rhel8:latest
# podman tag registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
# podman tag registry.redhat.io/rhceph/rhceph-5-dashboard-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-dashboard-rhel8:latest
# podman tag registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
# podman tag registry.redhat.io/openshift4/ose-prometheus:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

13. Edit the **/etc/containers/registries.conf** file. Add the node's FQDN with the port into the file in place of LOCAL_NODE_FQDN, and then save it:

Syntax

```
[registries.insecure]
registries = ['LOCAL_NODE_FQDN:5000']
```

NOTE

You must perform this step on all storage cluster nodes that access the local container registry.

14. Push the image to the local container registry you started. Use the local node's FQDN in place of LOCAL_NODE_FQDN:

Red Hat Enterprise Linux 7

```
# docker push LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-rhel8
# docker push LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
# docker push LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-dashboard-rhel8
# docker push LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
# docker push LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

Red Hat Enterprise Linux 8

```
# podman push LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-rhel8
# podman push LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
# podman push LOCAL_NODE_FQDN:5000/rhceph/rhceph-5-dashboard-rhel8
# podman push LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
# podman push LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

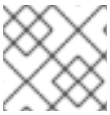
15. For Red Hat Enterprise Linux 7, restart the docker service:

Syntax

```
systemctl restart docker
```

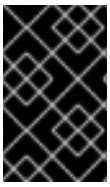
3.8.5. Performing a disconnected installation

Before you can perform the installation, you must obtain a Red Hat Ceph Storage container image, either from a proxy host that has access to the Red Hat registry or by copying the image to your local registry.



NOTE

Red Hat Ceph Storage supports Red Hat Enterprise Linux 8.4 and later.



IMPORTANT

Before you begin the bootstrapping process, make sure that the container image that you want to use has the same version of Red Hat Ceph Storage as **cephadm**. If the two versions do not match, bootstrapping fails at the **Creating initial admin user** stage.

Prerequisites

- At least one running virtual machine (VM) or server.
- Red Hat Enterprise Linux 8.4 or later.
- Root-level access to all nodes.

- Passwordless **ssh** is set up on all hosts in the storage cluster.
- A Red Hat Ceph Storage container image.
- The container image resides in the custom registry.

Procedure

1. Log in to the bootstrap host.
2. Bootstrap the storage cluster:

Syntax

```
cephadm --image CUSTOM-CONTAINER-REGISTRY-NAME:PORT:_CUSTOM-IMAGE-NAME_:_IMAGE-TAG_ bootstrap --mon-ip IP-ADDRESS
```

Example

```
[root@vm00 ~]# cephadm --image my-private-registry.com:5000:myimage:mytag1 bootstrap --mon-ip 10..0.127.0
```

The script takes a few minutes to complete. Once the script completes, it provides the credentials to the Red Hat Ceph Storage Dashboard URL, a command to access the Ceph command-line interface (CLI), and a request to enable telemetry.

```
Ceph Dashboard is now available at:
+
             URL: https://rh8-3.storage.lab:8443/
            User: admin
        Password: i8nhu7zham

You can access the Ceph CLI with:

        sudo /usr/sbin/cephadm shell --fsid 266ee7a8-2a05-11eb-b846-5254002d4916 -c /etc/ceph/ceph.conf -k /etc/ceph/ceph.client.admin.keyring

Please consider enabling telemetry to help improve Ceph:

        ceph telemetry on

For more information see:

        https://docs.ceph.com/docs/master/mgr/telemetry/

Bootstrap complete.
```

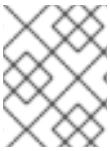
After the bootstrap process is complete, see Changing configurations of custom container images for disconnected installations to configure the container images.

Additional Resources

- Once your storage cluster is up and running, refer to the *Red Hat Ceph Storage Operations Guide* for more information about configuring additional daemons and services.

3.8.6. Changing configurations of custom container images for disconnected installations

After you perform the initial bootstrap for disconnected nodes, you must specify custom container images for monitoring stack daemons. You can override the default container images for monitoring stack daemons, since the nodes do not have access to the default container registry.



NOTE

Make sure that the bootstrap process on the initial host is complete before making any configuration changes.

Prerequisites

- At least one running virtual machine (VM) or server.
- Red Hat Enterprise Linux 8.4 or later.
- Root-level access to all nodes.
- Passwordless **ssh** is set up on all hosts in the storage cluster.

Procedure

1. Set the custom container images with the **ceph config** command:

   Syntax

   ```
   ceph config set mgr mgr/cephadm/OPTION-NAME CUSTOM-REGISTRY-NAME/CONTAINER-NAME
   ```

   Use the following options for OPTION-NAME:

   ```
   container_image_prometheus
   container_image_grafana
   container_image_alertmanager
   container_image_node_exporter
   ```

   Example

   ```
   [root@vm00 ~]# ceph config set mgr mgr/cephadm/container_image_prometheus myregistry/mycontainer
   [root@vm00 ~]# ceph config set mgr mgr/cephadm/container_image_grafana myregistry/mycontainer
   [root@vm00 ~]# ceph config set mgr mgr/cephadm/container_image_alertmanager myregistry/mycontainer
   [root@vm00 ~]# ceph config set mgr mgr/cephadm/container_image_node_exporter myregistry/mycontainer
   ```

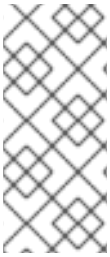
2. Redeploy **node_exporter**:

   Syntax

```
ceph orch redeploy node_exporter
```

NOTE

If any of the services do not deploy, you can redeploy them with the **ceph orch redeploy** command.



NOTE

By setting a custom image, the default values for the configuration image name and tag will be overridden, but not overwritten. The default values change when updates become available. By setting a custom image, you will not be able to configure the component for which you have set the custom image for automatic updates. You will need to manually update the configuration image name and tag to be able to install updates.

- If you choose to revert to using the default configuration, you can reset the custom container image. Use **ceph config rm** to reset the configuration option:

  Syntax

  ```
  ceph config rm mgr mgr/cephadm/OPTION-NAME
  ```

  Example

  ```
  ceph config rm mgr mgr/cephadm/container_image_prometheus
  ```

Additional Resources

- For more information about performing a disconnected installation, see Performing a disconnected installation.

3.8.7. Verifying the cluster installation

Once the cluster installation is complete, you can verify that the Red Hat Ceph Storage 5 installation is running properly.

Prerequisites

- Root-level access to all nodes in the storage cluster.

Procedure

1. Use **podman** to verify that the installation is up and running:

   Example

   ```
   [root@vm00 ~]# podman ps
   CONTAINER ID  IMAGE                                                COMMAND
   CREATED         STATUS             PORTS   NAMES
   2189a046ee8f  registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.5  --no-
   collector.ti...  40 seconds ago  Up 40 seconds ago
   ceph-266ee7a8-2a05-11eb-b846-5254002d4916-node-exporter.rh8-3
   ```

```
4fcd70e36789  registry.redhat.io/openshift4/ose-prometheus:v4.6
--config.file=/et...  37 seconds ago  Up 36 seconds ago
ceph-266ee7a8-2a05-11eb-b846-5254002d4916-prometheus.rh8-3
6fee9fb1a0b8  registry.redhat.io/rhceph-alpha/rhceph-5-rhel8:latest            -n mgr.rh8-
3.stor...  2 minutes ago   Up 2 minutes ago
ceph-266ee7a8-2a05-11eb-b846-5254002d4916-mgr.rh8-3.storage.lab.byr
gqt
99efe0deaf6e  registry.redhat.io/rhceph-alpha/rhceph-5-dashboard-rhel8:latest
23 seconds ago  Up 22 seconds ago
ceph-266ee7a8-2a05-11eb-b846-5254002d4916-grafana.rh8-3
a31e731a60e0  registry.redhat.io/rhceph-alpha/rhceph-5-rhel8:latest
-n mon.rh8-3.stor...  2 minutes ago   Up 2 minutes ago
ceph-266ee7a8-2a05-11eb-b846-5254002d4916-mon.rh8-3.storage.lab
a8897bf4b654  registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.5   --
web.listen-addr...  26 seconds ago  Up 25 seconds ago
ceph-266ee7a8-2a05-11eb-b846-5254002d4916-alertmanager.rh8-3
b443935032d0  registry.redhat.io/rhceph-alpha/rhceph-5-rhel8:latest
```

NOTE

In Red Hat Ceph Storage 5, the format of the **systemd** units has changed. In the **NAMES** column, the unit files now include the **FSID**.

3.9. LAUNCHING THE CEPHADM SHELL

The **cephadm shell** command launches a **bash** shell in a container with all of the Ceph packages installed. This enables you to perform "Day One" cluster setup tasks, such as installation and bootstrapping, and to invoke **ceph** commands.

Prerequisites

- A storage cluster that has been installed and bootstrapped.
- Root-level access to all nodes in the storage cluster.

Procedure

There are two ways to launch the **cephadm** shell:

- Enter **cephadm shell** at the system prompt. This example invokes the **ceph -s** command from within the shell.

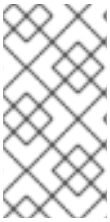
  #### Example

  ```
  [root@vm00 ~]# cephadm shell
  [cephadm@cephadm /~]# ceph -s
  ```

- At the system prompt, type **cephadm shell** and the command you want to execute:

  #### Example

  ```
  [root@vm00 ~]# cephadm shell ceph -s
  ```



NOTE

If the node contains configuration and keyring files in **/etc/ceph/**, the container environment uses the values in those files as defaults for the **cephadm** shell. If you execute the **cephadm** shell on a MON node, the **cephadm** shell inherits its default configuration from the MON container, instead of using the default configuration.

3.10. ADDING HOSTS

Bootstrapping the Red Hat Ceph Storage installation creates a working storage cluster, consisting of one Monitor daemon and one Manager daemon within the same container. As a storage administrator, you can add additional hosts to the storage cluster and configure them.

NOTE

Running the preflight playbook installs **podman**, **lvm2**, **chronyd**, and **cephadm** on all hosts listed in the Ansible inventory file.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Root-level access to all nodes in the storage cluster.
- Login to **registry.redhat.io** on all the nodes of the storage cluster.

Procedure

1. Add the new host to the Ansible inventory file. The default location for the file is **/usr/share/cephadm-ansible/hosts/**.
2. Switch to root user and install the storage cluster's public SSH key in the root user's **authorized_keys** file on the new host:

   Syntax

   ```
   ssh-copy-id -f -i /etc/ceph/ceph.pub root@NEWHOST
   ```

   Example

   ```
   [root@node00 /]# ssh-copy-id -f -i /etc/ceph/ceph.pub root@node01
   [root@node00 /]# ssh-copy-id -f -i /etc/ceph/ceph.pub root@node02
   ```

3. Run the preflight playbook with the **--limit** option:

   Syntax

   ```
   ansible-playbook -i INVENTORY-FILE cephadm-preflight.yml --limit NEWHOST
   ```

   Example

   ```
   [root@admin ~]# ansible-playbook -i /usr/share/cephadm-ansible/hosts/ cephadm-
   preflight.yml --limit host01
   ```

The preflight playbook installs **podman**, **lvm2**, **chronyd**, and **cephadm** on the new host. After installation is complete, **cephadm** resides in the **/usr/sbin/** directory.

4. Use the **cephadm** orchestrator to add the new host to the storage cluster:

```
ceph orch host add _NEWHOST_ _IP-ADDRESS_
```

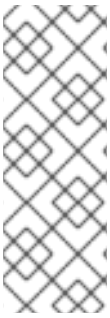
Example

```
[ceph: root@host01 /]# ceph orch host add host02 10.0.127.0
Added host 'host02'
[ceph: root@host01 /]# ceph orch host add host03 10.0.127.1
Added host 'host03'
```

5. Use the **ceph orch host ls** command to view the status of the storage cluster, and to verify that the new host has been added.

NOTE

The *STATUS* of the hosts is blank, in the output of the **ceph orch host ls** command.



NOTE

You can also add nodes by IP address. If you do not have DNS configured in your storage cluster environment, you can add the hosts by IP address, along with the host names.

Syntax

```
ceph orch host add HOSTNAME IP-ADDRESS LABELS
```

NOTE

You can also add nodes by IP address after you run the preflight playbook. If you do not have DNS configured in your storage cluster environment, you can add the hosts by IP address, along with the host names.

```
ceph orch host add _HOSTNAME_ _IP-ADDRESS_ _LABELS_
```

3.10.1. Using the addr option to identify hosts

The **addr** option offers an additional way to contact a host. Add the IP address of the host to the **addr** option. If **ssh** cannot connect to the host by its hostname, then it uses the value stored in **addr** to reach the host by its IP address.

Prerequisites

- A storage cluster that has been installed and bootstrapped.
- Root-level access to all nodes in the storage cluster.

Procedure

Run this procedure from inside the **cephadm** shell.

1. Add the IP address:

   Syntax

   ```
   ceph orch host add HOSTNAME ADDR
   ```

   Example

   ```
   [cephadm@cephadm /]# ceph orch host add node00 192.168.1.128
   ```

NOTE

If adding a host by hostname results in that host being added with an IPv6 address instead of an IPv4 address, use **ceph orch host** to specify the IP address of that host:

```
ceph orch host set-addr _HOSTNAME_ _IP-ADDR_
```

To convert the IP address from IPv6 format to IPv4 format for a host you have added, use the following command:

```
ceph orch host set-addr _HOSTNAME_ IPV4-ADDRESS
```

3.10.2. Labeling Hosts

The Ceph orchestrator supports assigning labels to hosts. Labels are free-form and have no specific meanings. This means that you can use **mon**, **monitor**, **mycluster_monitor**, or any other text string. Each host can have multiple labels.

For example, apply the **mon** label to all hosts on which you want to deploy Monitor daemons, **mgr** for all hosts on which you want to deploy Manager daemons, **rgw** for RADOS gateways, and so on.

Labeling all the hosts in the storage cluster helps to simplify system management tasks by allowing you to quickly identify the daemons running on each host. In addition, you can use the Ceph orchestrator or a YAML file to deploy or remove daemons on hosts that have specific host labels.

Prerequisites

- A storage cluster that has been installed and bootstrapped.

Procedure

1. Launch the **cephadm** shell:

   ```
   [root@vm00 ~]# cephadm shell
   [cephadm@cephadm ~]#
   ```

2. Add a label to a host:

   Syntax

   ```
   ceph orch host label add HOSTNAME LABEL
   ```

Example

```
[cephadm@cephadm ~]# ceph orch host label add node00 mon
```

3.10.2.1. Removing a label from a host

1. Use the ceph orchestrator to remove a label from a host:

Syntax

```
ceph orch host label rm HOSTNAME LABEL
```

Example

```
[cephadm@cephadm ~]# ceph orch host label rm node00 mon
```

3.10.2.2. Using host labels to deploy daemons on specific hosts

There are two ways to use host labels to deploy daemons on specific hosts: by using the **--placement** option from the command line, and by using a YAML file.

- Use the **--placement** option to deploy a daemon from the command line:

Example

```
[cephadm@cephadm ~]# ceph orch apply prometheus --placement="label:mylabel"
```

- To assign the daemon to a specific host label in a YAML file, specify the service type and label in the YAML file:

Example

```
service_type: prometheus
placement:
  label: "mylabel"
```

3.10.3. Adding multiple hosts

Use a YAML file to add multiple hosts to the storage cluster at the same time.

NOTE

Be sure to create the **hosts.yaml** file within a host container, or create the file on the local host and then use the **cephadm** shell to mount the file within the container. The **cephadm** shell automatically places mounted files in **/mnt**. If you create the file directly on the local host and then apply the **hosts.yaml** file instead of mounting it, you might see a **File does not exist** error.

Prerequisites

- A storage cluster that has been installed and bootstrapped.

- Root-level access to all nodes in the storage cluster.

Procedure

1. Copy over the public **ssh** key to each of the hosts that you want to add.
2. Use a text editor to create a **hosts.yaml** file.
3. Add the host descriptions to the **hosts.yaml** file, as shown in the following example. Include the labels to identify placements for the daemons that you want to deploy on each host. Separate each host description with three dashes (---).

   Example

   ```
   service_type: host
   addr:
   hostname: host00
   labels:
   - mon
   - osd
   - mgr
   ---
   service_type: host
   addr:
   hostname: host01
   labels:
   - mon
   - osd
   - mgr
   ---
   service_type: host
   addr:
   hostname: host02
   labels:
   - mon
   - osd
   ```

4. If you created the **hosts.yaml** file within the host container, invoke the **ceph orch apply** command:

   Example

   ```
   [root@vm00 ~]# ceph orch apply -i hosts.yaml
   Added host 'host00'
   Added host 'host01'
   Added host 'host02'
   ```

5. If you created the **hosts.yaml** file directly on the local host, use the **cephadm** shell to mount the file:

   Example

   ```
   [root@vm00 ~]# cephadm shell --mount hosts.yaml -- ceph orch apply -i /mnt/hosts.yaml"
   ```

6. View the list of hosts and their labels:

   Example

```
[root@vm00 ~]# ceph orch host ls
HOST     ADDR     LABELS          STATUS
host00   host00   mon osd mgr
host01   host01   mon osd mgr
host02   host02   mon osd
```

NOTE

If a host is online and operating normally, its status is blank. An offline host shows a status of OFFLINE, and a host in maintenance mode shows a status of MAINTENANCE.

3.10.4. Adding hosts in disconnected deployments

If you are running a storage cluster on a private network and your host domain name server (DNS) cannot be reached through private IP, you must include both the host name and the IP address for each host you want to add to the storage cluster.

Prerequisites

- A running storage cluster.
- Root-level access to all hosts in the storage cluster.

Procedure

1. Invoke the **cephadm** shell.

   Syntax

```
[root@vm00 ~]# cephadm shell
```

2. Add the host:

   Syntax

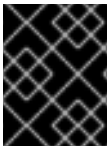
```
ceph orch host add HOST_NAME HOST_ADDRESS
```

   Example

```
[ceph:root@node00 /]# ceph orch host add node03 172.20.20.9
```

3.10.5. Removing hosts

There are two ways to remove hosts from a storage cluster. The method that you use depends upon whether the host is running the node-exporter or crash services.



IMPORTANT

If the host that you want to remove is running OSDs, remove them from the host before removing the host.

Prerequisites

- A storage cluster that has been installed and bootstrapped.
- Root-level access to all nodes in the storage cluster.

Procedure

1. If the host is not running node-exporter or the crash service, edit the placement specification file and remove all instances of the host name. By default, the placement specification file is named **cluster.yml**.

   Example

   ```
   Update:

   service_type: rgw
   placement:
     hosts:
     - host01
     - host02

   To:

   service_type: rgw
   placement:
     hosts:
     - host01
   ```

2. Remove the host from the **cephadm** environment`:

   Example

   ```
   [root@vm00 ~]# ceph orch host rm host2
   ```

   - If the host that you want to remove is running node-exporter or crash services, run the following command on the host to remove them:

     Syntax

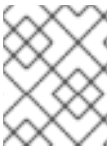
     ```
     cephadm rm-daemon --fsid CLUSTER-ID --name SERVICE-NAME
     ```

     Example

     ```
     [root@host02 ~]# cephadm rm-daemon --fsid cluster00 --name node-exporter
     ```

3.11. ADDING MONITOR SERVICE

A typical Red Hat Ceph Storage storage cluster has three or five monitor daemons deployed on different hosts. If your storage cluster has five or more hosts, Red Hat recommends that you deploy five Monitor nodes.



NOTE

The bootstrap node is the initial monitor of the storage cluster. Be sure to include the bootstrap node in the list of hosts to which you want to deploy.

NOTE

If you want to apply Monitor service to more than one specific host, be sure to specify all of the host names within the same **ceph orch apply** command. If you specify **ceph orch apply mon --placement host1** and then specify **ceph orch apply mon --placement host2**, the second command removes the Monitor service on host1 and applies a Monitor service to host2.

If your Monitor nodes or your entire cluster are located on a single subnet, then **cephadm** automatically adds up to five Monitor daemons as you add new hosts to the cluster. **cephadm** automatically configures the Monitor daemons on the new hosts. The new hosts reside on the same subnet as the first (bootstrap) host in the storage cluster. **cephadm** can also deploy and scale monitors to correspond to changes in the size of the storage cluster.

Prerequisites

- Root-level access to all hosts in the storage cluster.
- A running storage cluster.

Procedure

1. Apply the five Monitor daemons to five random hosts in the storage cluster:

```
ceph orch apply mon 5
```

- Disable automatic Monitor deployment:

```
ceph orch apply mon --unmanaged
```

3.11.1. Adding Monitor nodes to specific hosts

Use host labels to identify the hosts that contain Monitor nodes.

Prerequisites

- Root-level access to all nodes in the storage cluster.
- A running storage cluster.

Procedure

1. Assign the **mon** label to the host:

Syntax

```
ceph orch host label add HOSTNAME mon
```

Example

```
[ceph: root@host01 ~]# ceph orch host label add host01 mon
```

2. View the current hosts and labels:

Syntax

```
ceph orch host ls
```

Example

```
[ceph: root@host01 ~]# ceph orch host label add host01 mon
[ceph: root@host01 ~]# ceph orch host label add host02 mon
[ceph: root@host01 ~]# ceph orch host ls
HOST   ADDR   LABELS  STATUS
host01        mon
host02        mon
host03
host04
host05
```

3. Deploy monitors based on the host label:

Syntax

```
ceph orch apply mon label:mon
```

4. Deploy monitors on a specific set of hosts:

Example

```
[root@mon ~]# ceph orch apply mon _host01,host02,host03,..._
```

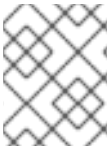
3.12. SETTING UP THE ADMIN NODE

Use an admin node to administer the storage cluster.

An admin node contains both the cluster configuration file and the admin keyring. Both of these files are stored in the directory **/etc/ceph** and use the name of the storage cluster as a prefix.

For example, the default ceph cluster name is **ceph**. In a cluster using the default name, the admin keyring is named **/etc/ceph/ceph.client.admin.keyring**. The corresponding cluster configuration file is named **/etc/ceph/ceph.conf**.

To set up a host in the storage cluster as the admin node, apply the _admin label to the host you want to designate as the admin node.



NOTE

Make sure that you copy the **ceph.conf** file and admin keyring to the admin node after you apply the _admin label.

Prerequisites

- A running storage cluster with **cephadm** installed.
- The storage cluster has running Monitor and Manager nodes.
- Root-level access to all nodes in the cluster.

Procedure

1. Use **ceph orch host ls** to view the hosts in your storage cluster:

   Example

   ```
   [cephadm@cephadm /]# ceph orch host ls
   HOST   ADDR   LABELS  STATUS
   host01         mon
   host02         mon,mgr
   host03
   host04
   host05
   ```

2. Use the **_admin** label to designate the admin host in your storage cluster. For best results, this host should have both Monitor and Manager daemons running.

   Syntax

   ```
   ceph orch host label add HOSTNAME _admin
   ```

   Example

   ```
   [cephadm@cephadm /]# ceph orch host label add host02 _admin
   ```

3. Verify that the admin host has the _admin label.

   Example

   ```
   [cephadm@cephadm /]# ceph orch host ls
   HOST   ADDR   LABELS  STATUS
   host01         mon
   host02         mon,mgr,_admin
   host03
   host04
   host05
   ```

4. Log in to the admin node to manage the storage cluster.

3.12.1. Deploying Ceph monitor nodes using host labels

A typical Red Hat Ceph Storage storage cluster has three or five Ceph Monitor daemons deployed on different hosts. If your storage cluster has five or more hosts, Red Hat recommends that you deploy five Ceph Monitor nodes.

If your Ceph Monitor nodes or your entire cluster are located on a single subnet, then **cephadm** automatically adds up to five Ceph Monitor daemons as you add new nodes to the cluster. **cephadm** automatically configures the Ceph Monitor daemons on the new nodes. The new nodes reside on the same subnet as the first (bootstrap) node in the storage cluster. **cephadm** can also deploy and scale monitors to correspond to changes in the size of the storage cluster.

NOTE

Use host labels to identify the hosts that contain Ceph Monitor nodes.

Prerequisites

- Root-level access to all nodes in the storage cluster.
- A running storage cluster.

Procedure

1. Assign the mon label to the host:

   Syntax

   ```
   ceph orch host label add HOSTNAME mon
   ```

   Example

   ```
   [ceph: root@host01 ~]# ceph orch host label add host01 mon
   [ceph: root@host01 ~]# ceph orch host label add host02 mon
   ```

2. View the current hosts and labels:

   Syntax

   ```
   ceph orch host ls
   ```

   ```
   [ceph: root@host01 ~]# ceph orch host ls
   HOST   ADDR   LABELS  STATUS
   host01         mon
   host02         mon
   host03
   host04
   host05
   ```

   - Deploy Ceph Monitor daemons based on the host label:

     Syntax

     ```
     ceph orch apply mon label:mon
     ```

- Deploy Ceph Monitor daemons on a specific set of hosts:

  Example

  ```
  [ceph: root@host01 ~]# ceph orch apply mon _host01,host02,host03,..._
  ```

  NOTE

  Be sure to include the bootstrap node in the list of hosts to which you want to deploy.

3.12.2. Adding Ceph Monitor nodes by IP address or network name

A typical Red Hat Ceph Storage storage cluster has three or five monitor daemons deployed on different hosts. If your storage cluster has five or more hosts, Red Hat recommends that you deploy five Monitor nodes.

If your Monitor nodes or your entire cluster are located on a single subnet, then **cephadm** automatically adds up to five Monitor daemons as you add new nodes to the cluster. You do not need to configure the Monitor daemons on the new nodes. The new nodes reside on the same subnet as the first node in the storage cluster. The first node in the storage cluster is the bootstrap node. **cephadm** can also deploy and scale monitors to correspond to changes in the size of the storage cluster.

Prerequisites

- Root-level access to all nodes in the storage cluster.
- A running storage cluster.

Procedure

1. To deploy each additional Ceph Monitor node:

   Syntax

   ```
   ceph orch apply mon NODE:IP-ADDRESS-OR-NETWORK-NAME [NODE:IP-ADDRESS-OR-NETWORK-NAME...]
   ```

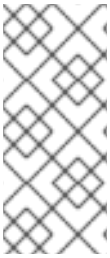
   Example

   ```
   [ceph: root@node00 ~]# ceph orch apply mon node01:10.1.2.0 node02:mynetwork
   ```

3.13. ADDING MANAGER SERVICE

cephadm automatically installs a Manager daemon on the bootstrap node during the bootstrapping process. Use the Ceph orchestrator to deploy additional Manager daemons.

The Ceph orchestrator deploys two Manager daemons by default. To deploy a different number of Manager daemons, specify a different number. If you do not specify the hosts where the Manager daemons should be deployed, the Ceph orchestrator randomly selects the hosts and deploys the Manager daemons to them.



NOTE

If you want to apply Manager daemons to more than one specific host, be sure to specify all of the host names within the same **ceph orch apply** command. If you specify **ceph orch apply mgr --placement host1** and then specify **ceph orch apply mgr --placement host2**, the second command removes the Manager daemon on host1 and applies a Manager daemon to host2.

Red Hat recommends that you use the **--placement** option to deploy to specific hosts.

Prerequisites

- A running storage cluster.

Procedure

1. To specify that you want to apply a certain number of Manager daemons to randomly selected hosts:

   Syntax

   ```
   ceph orch apply mgr NUMBER-OF-DAEMONS
   ```

   Example

   ```
   [ceph: root@node01 ~]# ceph orch apply mgr 3
   ```

   - To add Manager daemons to specific hosts in your storage cluster:

     Syntax

     ```
     ceph orch apply mgr --placement "HOSTNAME1 HOSTNAME2 HOSTNAME3"
     ```

     Example

     ```
     [ceph: root@node01 /]# ceph orch apply mgr --placement "node01 node02 node03"
     ```

3.14. ADDING OSDS

Cephadm will not provision an OSD on a device that is not available. A storage device is considered available if meets all of the following conditions:

- The device must have no partitions.
- The device must not have any LVM state.
- The device must not be mounted.
- The device must not contain a file system.
- The device must not contain a Ceph BlueStore OSD.
- The device must be larger than 5 GB.

Prerequisites

- A running Red Hat Ceph Storage cluster.

Procedure

1. List the available devices to deploy OSDs:

   Syntax

   ```
   ceph orch device ls [--hostname=HOSTNAME_1 HOSTNAME_2] [--wide] [--refresh]
   ```

   Example

   ```
   [ceph: root@host01 /]# ceph orch device ls --wide --refresh
   ```

2. You can either deploy the OSDs on specific hosts or on all the available devices:

   - To create an OSD from a specific device on a specific host:

     Syntax

     ```
     ceph orch daemon add osd HOSTNAME:_DEVICE-PATH_
     ```

     Example

     ```
     [ceph: root@host01 /]# ceph orch daemon add osd host01:/dev/sdb
     ```

   - To deploy OSDs on any available and unused devices, use the **--all-available-devices** option.

     Example

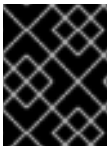
     ```
     [ceph: root@host01 /]# ceph orch apply osd --all-available-devices
     ```

Additional Resources

- For more information about drive specifications for OSDs, see the *Advanced service specifications and filters for deploying OSDs* section in the *Red Hat Ceph Storage Operations Guide*.
- For more information on zapping devices to clear data on devices, see the *Zapping devices for Ceph OSD deployment* section in the *Red Hat Ceph Storage Operations Guide*.

3.15. PURGING THE CEPH STORAGE CLUSTER

Purging the Ceph storage cluster clears any data or connections that remain from previous deployments on your server. This Ansible script removes all daemons, logs, and data that belong to the fsid passed to the script from all hosts in the storage cluster.



IMPORTANT

This process works only if the **cephadm** binary is installed on all hosts in the storage cluster.

The Ansible inventory file lists all the hosts in your cluster and what roles each host plays in your Ceph storage cluster. The default location for an inventory file is /etc/ansible hosts, but this file can be placed anywhere.

The following example shows the structure of an inventory file:

Example

```
[root@node00 ~]# cat hosts

[admin]
node1

[mons]
node1
node2
node3

[mgrs]
node1
node2
node3

[osds]
node1
node2
node3
```

Prerequisites

- A running bootstrap node.
- Ansible 2.9 or later is installed on the bootstrap node.
- Root-level access to all nodes in the cluster.
- The **[admin]** group is defined in the inventory file with a node where the admin keyring is present at **/etc/ceph/ceph.client.admin.keyring**.

Procedure

1. Use the **cephadm** orchestrator to halt **cephadm** on the bootstrap node:

   Syntax

   ```
   ceph orch pause
   ```

2. As an ansible user, run the purge script:

Syntax

```
ansible-playbook -i hosts cephadm-purge-cluster.yml -e fsid=FSID -vvv
```

Example

```
[root@node00 cephadm-ansible]# ansible-playbook -i hosts cephadm-purge-cluster.yml -e
fsid=a6ca415a-cde7-11eb-a41a-002590fc2544 -vvv
```

When the script has completed, the entire storage cluster will have been removed from all hosts in the cluster.

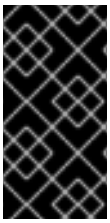
CHAPTER 4. UPGRADING A RED HAT CEPH STORAGE CLUSTER FROM RHCS 4 TO RHCS 5

As a storage administrator, you can upgrade a Red Hat Ceph Storage cluster from Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5. The upgrade process includes the following tasks:

- Upgrade the host OS version on the storage cluster from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8, if your storage cluster is still running Red Hat Enterprise Linux 7.
- Use Ansible playbooks to upgrade a Red Hat Ceph Storage 4 storage cluster to Red Hat Ceph Storage 5.

IMPORTANT

ceph-ansible is currently not supported with Red Hat Ceph Storage 5. This means that once you have migrated your storage cluster to Red Hat Ceph Storage 5, you must use **cephadm** and **cephadm-ansible** to perform subsequent updates.



IMPORTANT

Upgrading a storage cluster with an RGW NFS gateway from Red Hat Ceph Storage 4 to to Red Hat Ceph Storage 5 is currently not supported. The **ceph-ansible** upgrade fails and returns an error message. RGW NFS gateway support will be included in a later version of Red Hat Ceph Storage 5.

IMPORTANT

The option **bluefs_buffered_io** is set to **True** by default for Red Hat Ceph Storage. This option enables BlueFS to perform buffered reads in some cases, and enables the kernel page cache to act as a secondary cache for reads like RocksDB block reads. For example, if the RocksDB block cache is not large enough to hold all blocks during the OMAP iteration, it may be possible to read them from the page cache instead of the disk. This can dramatically improve performance when osd_memory_target is too small to hold all entries in the block cache. Currently, enabling **bluefs_buffered_io** and disabling the system level swap prevents performance degradation.

Red Hat Ceph Storage 5 supports only containerized daemons. It does not support non-containerized storage clusters. If you are upgrading a non-containerized storage cluster from Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5, the upgrade process includes the conversion to a containerized deployment.

4.1. PREREQUISITES

- A running Red Hat Ceph Storage 4 cluster.
- A valid customer subscription.
- Root-level access to the Ansible administration node.
- The Ansible user account for use with the Ansible application.
- Red Hat Ceph Storage tools and Ansible repositories are enabled.

IMPORTANT

You can manually upgrade the Ceph File System (CephFS) Metadata Server (MDS) software on a Red Hat Ceph Storage cluster and the Red Hat Enterprise Linux operating system to a new major release at the same time. The underlying XFS filesystem must be formatted with **ftype=1** or with **d_type** support. Run the command **xfs_info /var** to ensure the **ftype** is set to **1**. If the value of **ftype** is not **1**, attach a new disk or create a volume. On top of this new device, create a new XFS filesystem and mount it on **/var/lib/containers**.

Starting with Red Hat Enterprise Linux 8, **mkfs.xfs** enables **ftype=1** by default.

4.2. COMPATIBILITY CONSIDERATIONS BETWEEN RHCS AND PODMAN VERSIONS

podman and Red Hat Ceph Storage have different end-of-life strategies that might make it challenging to find compatible versions.

If you plan to upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 as part of the Ceph upgrade process, make sure that the version of **podman** is compatible with Red Hat Ceph Storage 5.0.

IMPORTANT

Red Hat Ceph Storage 5.0 is compatible with **podman** versions 2.0.0 and later, except for version 2.2.1. Version 2.2.1 is not compatible with Red Hat Ceph Storage 5.0.

The following table shows version compatibility between Red Hat Ceph Storage 5.0 and versions of **podman**.

Ceph	Podman				
	1.9	2.0	2.1	2.2	3.0
5.0 (Pacific)	false	true	true	false	true

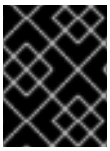
4.3. PREPARING FOR AN UPGRADE

As a storage administrator, you can upgrade your Ceph storage cluster to Red Hat Ceph Storage 5. However, some components of your storage cluster must be running specific software versions before an upgrade can take place. The following list shows the minimum software versions that must be installed on your storage cluster before you can upgrade to Red Hat Ceph Storage 5.

- RHCS 4.2z2 or later.
- Ansible 2.9.
- Ceph-ansible shipped with the latest version of RHCS.
- RHEL 8.4.

- FileStore OSDs must be migrated to BlueStore. For more information about converting OSDs from FileStore to BlueStore, refer to BlueStore.

There is no direct upgrade path from RHCS versions earlier than RHCS 4.2z2. If you are upgrading from RHCS 3, you must first upgrade to RHCS 4.2z2 or later, and then upgrade to RHCS 5.



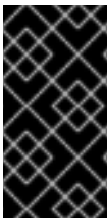
IMPORTANT

You can only upgrade to the latest version of Red Hat Ceph Storage 5. For example, if version 5.1 is available, you cannot upgrade from 4 to 5.0; you must go directly to 5.1.

To upgrade your storage cluster to RHCS 5, Red Hat recommends that your cluster be running RHCS 4.2z2 or later. Refer to the Knowledge Base Article What are the Red Hat Ceph Storage Releases? . This article contains download links to the most recent versions of the Ceph packages and ceph-ansible.

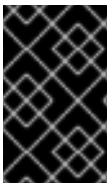
The upgrade process uses Ansible playbooks to upgrade an Red Hat Ceph Storage 4 storage cluster to Red Hat Ceph Storage 5. If your Red Hat Ceph Storage 4 cluster is a non-containerized cluster, the upgrade process includes a step to transform the cluster into a containerized version. Red Hat Ceph Storage 5 does not run on non-containerized clusters.

If you have a mirroring or multisite configuration, upgrade one cluster at a time. Make sure that each upgraded cluster is running properly before upgrading another cluster.



IMPORTANT

leapp does not support upgrades for encrypted OSDs or OSDs that have encrypted partitions. If your OSDs are encrypted and you are upgrading the host OS, disable **dmcrypt** in **ceph-ansible** before upgrading the OS. For more information about using **leapp**, refer to Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 .



IMPORTANT

Perform the first three steps in this procedure *only* if the storage cluster is not already running the latest version of RHCS 4. The latest version of RHCS 4 should be 4.2z2 or later.

Prerequisites

- A running Red Hat Ceph Storage 4 cluster.
- Sudo-level access to all nodes in the storage cluster.
- A valid customer subscription.
- Root-level access to the Ansible administration node.
- The Ansible user account for use with the Ansible application.
- Red Hat Ceph Storage tools and Ansible repositories are enabled.

Procedure

1. Enable the Ceph and Ansible repositories on the Ansible administration node:

   Example

```
[root@admin ceph-ansible]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

2. Use the **--extra-vars** option to update the **infrastructure-playbooks/rolling_update.yml** playbook and to change the **health_osd_check_retries** and **health_osd_check_delay** values to **50** and **30**, respectively:

   Example

   ```
   [root@admin ceph-ansible]# ansible-playbook -i hosts infrastructure-playbooks/rolling_update.yml --extra-vars "health_osd_check_retries=50 health_osd_check_delay=30"
   ```

   For each OSD node, these values cause **ceph-ansible** to check the storage cluster health every 30 seconds, up to 50 times. This means that **ceph-ansible** waits up to 25 minutes for each OSD.

   Adjust the **health_osd_check_retries** option value up or down, based on the used storage capacity of the storage cluster. For example, if you are using 218 TB out of 436 TB, or 50% of the storage capacity, then set the **health_osd_check_retries** option to **50**.

   /etc/ansible/hosts is the default location for the Ansible inventory file.

3. If the storage cluster you want to upgrade contains Ceph Block Device images that use the **exclusive-lock** feature, ensure that all Ceph Block Device users have permissions to create a denylist for clients:

   Syntax

   ```
   ceph auth caps client.ID mon 'allow r, allow command "osd blacklist"' osd 'EXISTING_OSD_USER_CAPS'
   ```

4. If the storage cluster was originally installed using Cockpit, create a symbolic link in the **/usr/share/ceph-ansible** directory to the inventory file where Cockpit created it, at **/usr/share/ansible-runner-service/inventory/hosts**:

   a. Change to the **/usr/share/ceph-ansible** directory:

      ```
      # cd /usr/share/ceph-ansible
      ```

   b. Create the symbolic link:

      ```
      # ln -s /usr/share/ansible-runner-service/inventory/hosts hosts
      ```

5. To upgrade the cluster using **ceph-ansible**, create the symbolic link in the **etc/ansible/hosts** directory to the **hosts** inventory file:

   ```
   # ln -s /etc/ansible/hosts hosts
   ```

6. If the storage cluster was originally installed using Cockpit, copy the Cockpit-generated SSH keys to the Ansible user's **~/.ssh** directory:

   a. Copy the keys:

Syntax

```
cp /usr/share/ansible-runner-service/env/ssh_key.pub
/home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
cp /usr/share/ansible-runner-service/env/ssh_key
/home/ANSIBLE_USERNAME/.ssh/id_rsa
```

Replace *ANSIBLE_USERNAME* with the user name for Ansible. The usual default user name is **admin**.

Example

```
# cp /usr/share/ansible-runner-service/env/ssh_key.pub /home/admin/.ssh/id_rsa.pub
# cp /usr/share/ansible-runner-service/env/ssh_key /home/admin/.ssh/id_rsa
```

b. Set the appropriate owner, group, and permissions on the key files:

Syntax

```
# chown ANSIBLE_USERNAME:_ANSIBLE_USERNAME_
/home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
# chown ANSIBLE_USERNAME:_ANSIBLE_USERNAME_
/home/ANSIBLE_USERNAME/.ssh/id_rsa
# chmod 644 /home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
# chmod 600 /home/ANSIBLE_USERNAME/.ssh/id_rsa
```

Replace *ANSIBLE_USERNAME* with the username for Ansible. The usual default user name is **admin**.

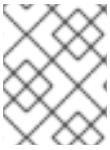
Example

```
# chown admin:admin /home/admin/.ssh/id_rsa.pub
# chown admin:admin /home/admin/.ssh/id_rsa
# chmod 644 /home/admin/.ssh/id_rsa.pub
# chmod 600 /home/admin/.ssh/id_rsa
```

Additional Resources

- What are the Red Hat Ceph Storage Releases?
- For more information about converting from FileStore to BlueStore, refer to BlueStore.

4.4. BACKING UP THE FILES BEFORE THE HOST OS UPGRADE



NOTE

Perform the procedure in this section only if you are upgrading the host OS. If you are not upgrading the host OS, skip this section.

Before you can perform the upgrade procedure, you must make backup copies of the files that you customized for your storage cluster, including keyring files and the **yml** files for your configuration.

Prerequisites

Prerequisites

- A running Red Hat Ceph Storage 4 cluster.
- A valid customer subscription.
- Root-level access to the Ansible administration node.
- The Ansible user account for use with the Ansible application.
- Red Hat Ceph Storage Tools and Ansible repositories are enabled.

Procedure

1. Make a backup copy of the **ceph.client.admin.keyring** file.
2. Make backup copies of the **ceph.conf** files from each node.
3. Make backup copies of the /**etc**/**ganesha**/ folder on each node.
4. If the storage cluster has RBD mirroring defined, then make backup copies of the /**etc**/**ceph** folder and the **group_vars**/**rbdmirrors.yml** file.

Additional Resources

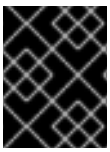
- For information about managing the RBD mirrors through **ceph-ansible**, refer to Migrating a non-containerized Red Hat Ceph Storage cluster to a containerized environment.

4.5. CONVERTING TO A CONTAINERIZED DEPLOYMENT

This procedure is required for non-containerized clusters. If your storage cluster is a non-containerized cluster, this procedure transforms the cluster into a containerized version.

Red Hat Ceph Storage 5 does not run on non-containerized clusters.

If your Red Hat Ceph Storage 4 storage cluster is already containerized, skip this section.



IMPORTANT

This procedure stops and restarts a daemon. If the playbook stops executing during this procedure, be sure to analyze the state of the cluster before restarting.

Prerequisites

- A running Red Hat Ceph Storage 4 cluster.
- Root-level access to all nodes in the storage cluster.
- A valid customer subscription.
- Root-level access to the Ansible administration node.
- The Ansible user account for use with the Ansible application.
- Red Hat Ceph Storage tools and Ansible repositories are enabled.

Procedure

Procedure

1. If you are running a multisite setup, set **rgw_multisite: false** in **all.yml**.
2. Ensure the **group_vars/all.yml** has the following default values for the configuration parameters:

```
ceph_docker_image_tag: "latest"
ceph_docker_registry: "registry.redhat.io"
ceph_docker_image: rhceph/rhceph-4-rhel8
containerized_deployment: true
```

NOTE

These values differ if you use a local registry and a custom image name.

3. If you are using daemons that are not containerized, convert them to containerized format:

Syntax

```
ansible-playbook -vvvv -i INVENTORY-FILE infrastructure-playbooks/switch-from-non-containerized-to-containerized-ceph-daemons.yml
```

The **-vvvv** option collects verbose logs of the conversion process.

Example

```
[ansible@admin ceph-ansible]$ ansible-playbook -vvvv -i hosts infrastructure-playbooks/switch-from-non-containerized-to-containerized-ceph-daemons.yml
```

4. Once the playbook completes successfully, edit the the value of **rgw_multisite: true`in the `all.yml** file and ensure the value of **containerized_deployment** is **true**.

4.6. UPDATING THE HOST OPERATING SYSTEM

Red Hat Ceph Storage 5 supports Red Hat Enterprise Linux 8.4 and later. This procedure enables you to install Red Hat Ceph Storage 5 and Red Hat Enterprise Linux 8 on the nodes in the storage cluster. If you are already running Red Hat Enterprise Linux 8 on your storage cluster, skip this procedure.

You must manually upgrade all other nodes in the cluster to run the most recent versions of Red Hat Enterprise Linux and Red Hat Ceph Storage.

Prerequisites

- A running Red Hat Ceph Storage 4 storage cluster.
- Sudo-level access to all nodes in the storage cluster.
- A valid customer subscription.
- Root-level access to the Ansible administration node.
- The Ansible user account for use with the Ansible application.

- Red Hat Ceph Storage tools and Ansible repositories are enabled.

Procedure

1. Use the **docker-to-podman** playbook to convert docker to podman:

   Example

   ```
   [ansible@admin ceph-ansible]$ ansible-playbook -vvvv -i hosts infrastructure-playbooks/
   docker-to-podman.yml
   ```

Additional Resources

- Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 .

4.6.1. Manually upgrading Ceph Monitor nodes and their operating systems

As a system administrator, you can manually upgrade the Ceph Monitor software on a Red Hat Ceph Storage cluster node and the Red Hat Enterprise Linux operating system to a new major release at the same time.

IMPORTANT

Perform the procedure on only one Monitor node at a time. To prevent cluster access issues, ensure that the current upgraded Monitor node has returned to normal operation *before* proceeding to the next node.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- The nodes are running Red Hat Enterprise Linux 7.8.
- The nodes are using Red Hat Ceph Storage version 4.2z2 or later.
- Access to the installation source is available for Red Hat Enterprise Linux 8.4.

Procedure

1. Stop the monitor service:

   Syntax

   ```
   systemctl stop ceph-mon@MONITOR_ID
   ```

   Replace *MONITOR_ID* with the Monitor node's ID number.

2. If using Red Hat Ceph Storage 4, disable the Red Hat Ceph Storage 4 repositories.

   a. Disable the tools repository:

   ```
   # subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
   ```

b. Disable the mon repository:

```
# subscription-manager repos --disable=rhel-7-server-rhceph-4-mon-rpms
```

3. Install the **leapp** utility. See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.

4. Run through the **leapp** preupgrade checks. See Assessing upgradability from the command line .

5. Set **PermitRootLogin yes** in **/etc/ssh/sshd_config**.

6. Restart the OpenSSH SSH daemon:

```
# systemctl restart sshd.service
```

7. Remove the iSCSI module from the Linux kernel:

```
# modprobe -r iscsi
```

8. Reboot the node.

9. Enable the repositories for Red Hat Ceph Storage 4 for Red Hat Enterprise Linux 8.

a. Enable the tools repository:

```
# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

10. Restore the **ceph-client-admin.keyring** and **ceph.conf** files from a Monitor node which has not been upgraded yet or from a node that has already had those files restored.

11. Verify that the monitor and manager services came back up and that the monitor is in quorum.

Syntax

```
ceph -s
```

On the *mon:* line under *services:*, ensure that the node is listed as in *quorum* and not as *out of quorum*.

Example

```
# ceph -s
mon: 3 daemons, quorum jb-ceph4-mon,jb-ceph4-mon2,jb-ceph4-mon3 (age 2h)
mgr: jb-ceph4-mon(active, since 2h), standbys: jb-ceph4-mon3, jb-ceph4-mon2
```

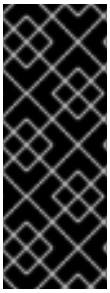
12. Repeat the above steps on all Monitor nodes until they have all been upgraded.

Additional Resources

- See Updating the host operating system for more information.
- See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 for more information.

4.6.2. Upgrading the OSD nodes

As a system administrator, you can manually upgrade the Ceph OSD software on a Red Hat Ceph Storage cluster node and the Red Hat Enterprise Linux operating system to a new major release at the same time.



IMPORTANT

Perform this procedure for each OSD node in the Ceph cluster, but typically only for one OSD node at a time. A maximum of one failure domain's worth of OSD nodes may be performed in parallel. For example, if per-rack replication is in use, one entire rack's OSD nodes can be upgraded in parallel. To prevent data access issues, ensure that the OSDs of the current OSD node have returned to normal operation and that all of the cluster PGs are in the **active+clean** state **before** proceeding to the next OSD.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- The nodes are running Red Hat Enterprise Linux 7.9.
- The nodes are using Red Hat Ceph Storage version 4.2z2 or later.
- Access to the installation source for Red Hat Enterprise Linux 8.4 or later.
- FileStore OSDs must be migrated to BlueStore.

Procedure

1. If you have FileStore OSDs that have not been migrated to BlueStore, run the **filestore-to-bluestore** playbook. For more information about converting OSDs from FileStore to BlueStore, refer to BlueStore.
2. Set the OSD **noout** flag to prevent OSDs from getting marked down during the migration:

   Syntax

   ```
   ceph osd set noout
   ```

3. Set the OSD **nobackfill**, **norecover**, **norrebalance**, **noscrub** and **nodeep-scrub** flags to avoid unnecessary load on the cluster and to avoid any data reshuffling when the node goes down for migration:

   Syntax

   ```
   ceph osd set nobackfill
   ceph osd set norecover
   ceph osd set norebalance
   ceph osd set noscrub
   ceph osd set nodeep-scrub
   ```

4. Gracefully shut down all the OSD processes on the node:

   Syntax

   ```
   ```

```
systemctl stop ceph-osd.target
```

5. If using Red Hat Ceph Storage 4, disable the Red Hat Ceph Storage 4 repositories.
   a. Disable the tools repository:

      Syntax

      ```
      subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
      ```

   b. Disable the osd repository:

      Syntax

      ```
      # subscription-manager repos --disable=rhel-7-server-rhceph-4-osd-rpms
      ```

6. Install the **leapp** utility. See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.
7. Run through the **leapp** preupgrade checks. See Assessing upgradability from the command line .
8. Set **PermitRootLogin yes** in **/etc/ssh/sshd_config**.
9. Restart the OpenSSH SSH daemon:

   Syntax

   ```
   systemctl restart sshd.service
   ```

10. Remove the iSCSI module from the Linux kernel:

    Syntax

    ```
    modprobe -r iscsi
    ```

11. Perform the upgrade by following Performing the upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.
    a. Enable the tools repository:

       ```
       # subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
       ```

12. Restore the **ceph.conf** file.
13. Unset the **noout**, **nobackfill**, **norecover**, **norebalance**, **noscrub** and **nodeep-scrub** flags:

    Syntax

    ```
    ceph osd unset noout
    ceph osd unset nobackfill
    ceph osd unset norecover
    ```

```
ceph osd unset norebalance
ceph osd unset noscrub
ceph osd unset nodeep-scrub
```

14. Verify that the OSDs are **up** and **in**, and that they are in the **active+clean** state.

    Syntax

    ```
    ceph -s
    ```

    On the *osd:* line under *services:*, ensure that all OSDs are **up** and **in**:

    Example

    ```
    # ceph-s
    osd: 3 osds: 3 up (since 8s), 3 in (since 3M)
    ```

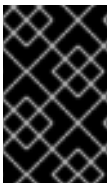
15. Repeat this procedure on all OSD nodes until they have all been upgraded.

Additional Resources

- Refer to BlueStore for more information about converting OSDs from FileStore to BlueStore.
- For more information about the **leapp** utility, see Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.
- For more information about converting docker to podman, see Updating the host operating system.

4.6.3. Upgrading the Ceph Object Gateway nodes

As a system administrator, you can manually upgrade the Ceph Object Gateway (RGW) software on a Red Hat Ceph Storage cluster node and the Red Hat Enterprise Linux operating system to a new major release at the same time.



IMPORTANT

Perform this procedure for each RGW node in the Ceph cluster, but only for one RGW node at a time. To prevent client access issues, ensure that the current upgraded RGW has returned to normal operation before proceeding to upgrade the next node.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- The nodes are running Red Hat Enterprise Linux 7.8 or later
- The nodes are using Red Hat Ceph Storage version 4.2z2 or later.
- Access to the installation source for Red Hat Enterprise Linux 8.4 or later.

Procedure

1. Stop the Ceph Object Gateway service:

Syntax

```
# systemctl stop ceph-radosgw.target
```

2. Disable the Red Hat Ceph Storage 4 tools repository:

```
# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

3. Install the **leapp** utility. See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.

4. Run through the leapp preupgrade checks. See Assessing upgradability from the command line .

5. Set **PermitRootLogin yes** in **/etc/ssh/sshd_config**.

6. Restart the OpenSSH SSH daemon:

```
# systemctl restart sshd.service
```

7. Remove the iSCSI module from the Linux kernel:

```
# modprobe -r iscsi
```

8. Perform the upgrade by following Performing the upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.

9. Enable the tools repository:

Syntax

```
subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

10. Restore the **ceph-client-admin.keyring** and **ceph.conf** files.

11. Verify that the daemon is active:

Syntax

```
ceph -s
```

View the *rgw:* line under *services:* to make sure that the RGW daemon is active.

Example

```
rgw: 1 daemon active (jb-ceph4-rgw.rgw0)
```

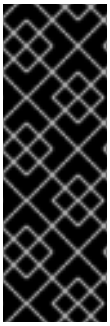
12. Repeat the above steps on all Ceph Object Gateway nodes until they have all been upgraded.

Additional Resources

- See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 for more information about the **leapp** utility.

4.6.4. Upgrading the CephFS Metadata Server nodes

As a storage administrator, you can manually upgrade the Ceph File System (CephFS) Metadata Server (MDS) software on a Red Hat Ceph Storage cluster and the Red Hat Enterprise Linux operating system to a new major release at the same time.



IMPORTANT

Before you upgrade the storage cluster, reduce the number of active MDS ranks to one per file system. This eliminates any possible version conflicts between multiple MDS. In addition, take all standby nodes offline before upgrading.

This is because the MDS cluster does not possess built-in versioning or file system flags. Without these features, multiple MDS might communicate using different versions of the MDS software, and could cause assertions or other faults to occur.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- The nodes are running Red Hat Enterprise Linux 7.8 or later.
- The nodes are using Red Hat Ceph Storage version 4.2z2 or later.
- Access to the installation source for Red Hat Enterprise Linux 8.4 or later.
- Root-level access to all nodes in the storage cluster.

Procedure

1. Reduce the number of active MDS ranks to 1:

   Syntax

   ```
   ceph fs set FILE_SYSTEM_NAME max_mds 1
   ```

   Example

   ```
   [root@mds ~]# ceph fs set fs1 max_mds 1
   ```

2. Wait for the cluster to stop all of the MDS ranks. When all of the MDS have stopped, only rank 0 should be active. The rest should be in standby mode. Check the status of the file system:

   ```
   [root@mds ~]# ceph status
   ```

3. Use **systemctl** to take all standby MDS offline:

   ```
   [root@mds ~]# systemctl stop ceph-mds.target
   ```

4. Confirm that only one MDS is online, and that it has rank 0 for the file system:

   ```
   [root@mds ~]# ceph status
   ```

5. Disable the Red Hat Ceph Storage 4 tools repository:

```
[root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

6. Install the **leapp** utility. For more information about **leapp**, refer to Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.

7. Run through the **leapp** preupgrade checks. For more information, refer to Assessing upgradability from the command line.

8. Edit **/etc/ssh/sshd_config** and set **PermitRootLogin** to **yes**.

9. Restart the OpenSSH SSH daemon:

```
[root@mds ~]# systemctl restart sshd.service
```

10. Remove the iSCSI module from the Linux kernel:

```
[root@mds ~]# modprobe -r iscsi
```

11. Perform the upgrade. See Performing the upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.

12. Enable the tools repository:

    Syntax

```
subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

13. Restore the **ceph-client-admin.keyring** and **ceph.conf** files.

14. Verify that the daemon is active:

```
[root@mds ~]# ceph -s
```

15. Follow the same processes for the standby daemons.

16. When you have finished restarting all of the MDS in standby, restore the previous value of **max_mds** for your cluster:

    Syntax

```
ceph fs set FILE_SYSTEM_NAME max_mds ORIGINAL_VALUE
```

    Example

```
[root@mds ~]# ceph fs set fs1 max_mds 5
```

Additional Resources

- See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 for more information about the **leapp** utility.

4.6.5. Manually upgrading the Ceph Dashboard node and its operating system

As a system administrator, you can manually upgrade the Ceph Dashboard software on a Red Hat Ceph Storage cluster node and the Red Hat Enterprise Linux operating system to a new major release at the same time.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- The node is running Red Hat Enterprise Linux 7.
- The node is running Red Hat Ceph Storage version 4.2z2 or later.
- Access is available to the installation source for Red Hat Enterprise Linux 8.4.

Procedure

1. Disable the Red Hat Ceph Storage 4 tools repository:

```
# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

2. Install the **leapp** utility. See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.
3. Run through the **leapp** preupgrade checks. See Assessing upgradability from the command line .
4. Set **PermitRootLogin yes** in **/etc/ssh/sshd_config**.
5. Restart the OpenSSH SSH daemon:

```
# systemctl restart sshd.service
```

6. Remove the iSCSI module from the Linux kernel:

```
# modprobe -r iscsi
```

7. Perform the upgrade by following Performing the upgrade from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.
8. Enable the tools repository for Red Hat Ceph Storage 4 for Red Hat Enterprise Linux 8:

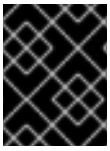
```
# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

Additional Resources

- See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 for more information.

4.6.6. Manually upgrading Ceph Ansible nodes and reconfiguring settings

Manually upgrade the Ceph Ansible software on a Red Hat Ceph Storage cluster node and the Red Hat Enterprise Linux operating system to a new major release at the same time.



IMPORTANT

Before upgrading the host OS on the Ceph Ansible nodes, back up the **group_vars** and **hosts** files. Use the created backups before reconfiguring the Ceph Ansible nodes.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- The node is running Red Hat Enterprise Linux 7.
- The node is running Red Hat Ceph Storage version 4.2z2 or later.
- Access is available to the installation source for Red Hat Enterprise Linux 8.4.

Procedure

1. Disable the tools repository for Red Hat Ceph Storage 4 for Red Hat Enterprise Linux 8:

```
[root@ansible ~]# subscription-manager repos --disable=rhceph-4-tools-for-rhel-8-x86_64-rpms
[root@ansible ~]# subscription-manager repos --disable=ansible-2.9-for-rhel-8-x86_64-rpms
```

2. Install the **leapp** utility. See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.
3. Run through the **leapp** preupgrade checks. See Assessing upgradability from the command line .
4. Edit **/etc/ssh/sshd_config** and set **PermitRootLogin** to **yes**.
5. Restart the OpenSSH SSH daemon:

```
[root@mds ~]# systemctl restart sshd.service
```

6. Remove the iSCSI module from the Linux kernel:

```
[root@mds ~]# modprobe -r iscsi
```

7. Perform the upgrade. See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.

   Syntax

```
subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

8. Restore the **ceph-client-admin.keyring** and **ceph.conf** files.

Additional Resources

- See Updating the host operating system for more information.
- See Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 for more information.

4.7. RESTORING THE BACKUP FILES

After you have completed the host OS upgrade on each node in your storage cluster, restore all the files that you backed up earlier to each node so that your upgraded node uses your preserved settings.

Repeat this process on each host in your storage cluster after the OS upgrade process for that host is complete.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Root-level access to all nodes in the storage cluster.

Procedure

1. Restore the files that you backed up before the host OS upgrade to the host.
2. Restore the **/etc/ceph** folders and their contents to all of the hosts, including the **ceph.client.admin.keyring** and **ceph.conf** files.
3. Restore the **/etc/ganesha/** folder to each node.
4. Check to make sure that the ownership for each of the backed-up files has not changed after the operating system upgrade. The file owner should be **ceph**. If the file owner has been changed to **root**, use the following command on each file to change the ownership back to **ceph**:

   #### Example

   ```
   [root@admin]# chown ceph: ceph.client.rbd-mirror.node01.keyring
   ```

5. If you upgraded from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8 and the storage cluster had RBD mirroring defined, restore the **/etc/ceph** folder from the backup copy.
6. Restore the **group_vars/rbdmirrors.yml** file that you backed up earlier.

4.8. BACKING UP THE FILES BEFORE THE RHCS UPGRADE

Before you run the **rolling_update.yml** playbook to upgrade Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5, make backup copies of all the **yml** files.

Prerequisites

- A Red Hat Ceph Storage 4 cluster running RHCS 4.2z2 or later.
- A valid customer subscription.
- Root-level access to the Ansible administration node.
- The Ansible user account for use with the Ansible application.
- Red Hat Ceph Storage tools and Ansible repositories are enabled.

Procedure

- Make backup copies of all the **yml** files.

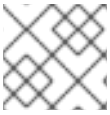
  Example

```
[root@admin ceph-ansible]# cp group_vars/all.yml group_vars/all_old.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml group_vars/osds_old.yml
[root@admin ceph-ansible]# cp group_vars/mdss.yml group_vars/mdss_old.yml
[root@admin ceph-ansible]# cp group_vars/rgws.yml group_vars/rgws_old.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml group_vars/clients_old.yml
```

4.9. THE UPGRADE PROCESS

As a storage administrator, you use Ansible playbooks to upgrade an Red Hat Ceph Storage 4 storage cluster to Red Hat Ceph Storage 5. The **rolling_update.yml** Ansible playbook performs upgrades for deployments of Red Hat Ceph Storage. The **ceph-ansible** upgrades the Ceph nodes in the following order:

- Ceph Monitor
- Ceph Manager
- Ceph OSD nodes
- MDS nodes
- Ceph Object Gateway (RGW) nodes
- Ceph RBD-mirror node
- Ceph NFS nodes
- Ceph iSCSI gateway node
- Ceph client nodes
- Ceph-crash daemons
- Node-exporter on all nodes
- Ceph Dashboard



NOTE

Red Hat Ceph Storage 5 supports only containerized deployments.

ceph-ansible is currently not supported with Red Hat Ceph Storage 5. This means that once you have migrated your storage cluster to Red Hat Ceph Storage 5, you must use **cephadm** to perform subsequent updates.

NOTE

Red Hat Ceph Storage 5 also includes a health check function that returns a DAEMON_OLD_VERSION warning if it detects that any of the daemons in the storage cluster are running multiple versions of Red Hat Ceph Storage. The warning is triggered when the daemons continue to run multiple versions of Red Hat Ceph Storage beyond the time value set in the **mon_warn_older_version_delay** option. By default, the **mon_warn_older_version_delay** option is set to one week. This setting allows most upgrades to proceed without falsely seeing the warning. If the upgrade process is paused for an extended time period, you can mute the health warning:

```
ceph health mute DAEMON_OLD_VERSION --sticky
```

After the upgrade has finished, unmute the health warning:

```
ceph health unmute DAEMON_OLD_VERSION
```

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Root-level access to all hosts in the storage cluster.
- A valid customer subscription.
- Root-level access to the Ansible administration node.
- The latest versions of Ansible and **ceph-ansible** available with Red Hat Ceph Storage 5.
- The **ansible** user account for use with the Ansible application.
- The nodes of the storage cluster is upgraded to Red Hat Enterprise Linux 8.4 or above.

IMPORTANT

The Ansible inventory file must be present in the **ceph-ansible** directory.

Procedure

1. Enable the Ceph and Ansible repositories on the Ansible administration node:

   Syntax

   ```
   subscription-manager repos --enable=rhceph-5-tools-for-rhel-8-x86_64-rpms --
   enable=ansible-2.9-for-rhel-8-x86_64-rpms
   ```

2. On the Ansible administration node, ensure that the latest versions of the **ansible** and **ceph-ansible** packages are installed.

   Syntax

   ```
   dnf update ansible ceph-ansible
   ```

3. Navigate to the **/usr/share/ceph-ansible/** directory:

Example

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

4. If upgrading from Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5, make copies of the **group_vars/osds.yml.sample** and **group_vars/clients.yml.sample** files, and rename them to **group_vars/osds.yml**, and **group_vars/clients.yml** respectively.

   Example

   ```
   [root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
   [root@admin ceph-ansible]# cp group_vars/mdss.yml.sample group_vars/mdss.yml
   [root@admin ceph-ansible]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
   [root@admin ceph-ansible]# cp group_vars/clients.yml.sample group_vars/clients.yml
   ```

5. If upgrading from Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5, edit the **group_vars/all.yml** file to add Red Hat Ceph Storage 5 details.

6. Once you have done the above two steps, copy the settings from the old **yaml** files to the new **yaml** files. Do not change the values of **ceph_rhcs_version**, **ceph_docker_image**, and **grafana_container_image** as the values for these configuration parameters are for Red Hat Ceph Storage 5. This ensures that all the settings related to your cluster are present in the current **yaml** file.

   Example

   ```
   fetch_directory: ~/ceph-ansible-keys
   monitor_interface: eth0
   public_network: 192.168.0.0/24
   ceph_docker_registry_auth: true
   ceph_docker_registry_username: _SERVICE_ACCOUNT_USER_NAME_
   ceph_docker_registry_password: _TOKEN_
   dashboard_admin_user:
   dashboard_admin_password:
   grafana_admin_user:
   grafana_admin_password:
   radosgw_interface: eth0
   ceph_docker_image: "rhceph/rhceph-5-rhel8"
   ceph_docker_image_tag: "latest"
   ceph_docker_registry: "registry.redhat.io"
   node_exporter_container_image: registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
   grafana_container_image: registry.redhat.io/rhceph/rhceph-5-dashboard-rhel8:5
   prometheus_container_image: registry.redhat.io/openshift4/ose-prometheus:v4.6
   alertmanager_container_image: registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
   ```

   NOTE

   Ensure the Red Hat Ceph Storage 5 container images are set to the default values.

7. Edit the **group_vars/osds.yml** file. Add and set the following options:

Syntax

```
nb_retry_wait_osd_up: 50
delay_wait_osd_up: 30
```

8. Open the **group_vars/all.yml** file and verify the following values are present from the old **all.yml** file.

   a. The **fetch_directory** option is set with the same value from the old **all.yml** file:

      Syntax

      ```
      fetch_directory: FULL_DIRECTORY_PATH
      ```

      Replace *FULL_DIRECTORY_PATH* with a writable location, such as the Ansible user's home directory.

   b. If the cluster you want to upgrade contains any Ceph Object Gateway nodes, add the **radosgw_interface** option:

      ```
      radosgw_interface: INTERFACE
      ```

      Replace *INTERFACE* with the interface to which the Ceph Object Gateway nodes listen.

   c. If your current setup has SSL certificates configured, edit the following:

      Syntax

      ```
      radosgw_frontend_ssl_certificate: /etc/pki/ca-trust/extracted/CERTIFICATE_NAME
      radosgw_frontend_port: 443
      ```

   d. Uncomment the **upgrade_ceph_packages** option and set it to **True**:

      Syntax

      ```
      upgrade_ceph_packages: True
      ```

   e. If the storage cluster has more than one rgw instance per node, then uncomment the **radosgw_num_instances** setting and set it to the number of instances per node in the cluster:

      Syntax

      ```
      radosgw_num_instances : NUMBER-OF-INSTANCES-PER-NODE
      ```

      Example

      ```
      radosgw_num_instances : 2
      ```

   f. If the storage cluster has RGW multisite defined, check the multisite settings in **all.yml** to make sure that they contain the same values as they did in the old **all.yml** file.

9. Log in as **ansible-user** on the Ansible administration node.

10. Execute the **rolling_update.yml** playbook to convert the storage cluster from Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5:

    Syntax

    ```
    ansible-playbook -vvvv infrastructure-playbooks/rolling_update.yml -i INVENTORY-FILE
    ```

    The -vvvv option collects verbose logs of the upgrade process.

    Example

    ```
    [ansible@admin ceph-ansible]$ ansible-playbook -vvvv infrastructure-
    playbooks/rolling_update.yml -i hosts
    ```

11. Review the Ansible playbook log output to verify the status of the upgrade.

Verification

1. List all running containers:

   Example

   ```
   [root@mon ~]# podman ps
   ```

2. Check the health status of the cluster. Replace *MONITOR-ID* with the name of the Ceph Monitor container found in the previous step:

   Syntax

   ```
   podman exec ceph-mon-MONITOR-ID ceph -s
   ```

   Example

   ```
   [root@mon ~]# podman exec ceph-mon-mon01 ceph -s
   ```

3. Verify the Ceph cluster daemon versions to confirm the upgrade of all daemons. Replace *MONITOR-ID* with the name of the Ceph Monitor container found in the previous step:

   Syntax

   ```
   podman exec ceph-mon-MONITOR-ID ceph --cluster ceph versions
   ```

   Example

   ```
   [root@mon ~]# podman exec ceph-mon-mon01 ceph --cluster ceph versions
   ```

4.10. CONVERTING THE STORAGE CLUSTER TO USING CEPHADM

After you have upgraded the storage cluster to Red Hat Ceph Storage 5, run the **cephadm-adopt** playbook to convert the storage cluster daemons to run **cephadm**.

The **cephadm-adopt** playbook adopts the Ceph services, installs all **cephadm** dependencies, enables the **cephadm** Orchestrator backend, generates and configures the **ssh** key on all hosts, and adds the hosts to the Orchestrator configuration.

NOTE

After you run the **cephadm-adopt** playbook, remove the **ceph-ansible** package. The cluster daemons no longer work with **ceph-ansible**. You must use **cephadm** to manage the cluster daemons.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Root-level access to all nodes in the storage cluster.

Procedure

1. Log in to the **ceph-ansible** node and change directory to **/usr/share/ceph-ansible**.
2. Run the **cephadm-adopt** playbook:

   Syntax

   ```
   ansible-playbook infrastructure-playbooks/cephadm-adopt.yml -i INVENTORY-FILE
   ```

   Example

   ```
   [ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/cephadm-adopt.yml -i hosts
   ```

3. Run the following command to enable applications to run on the NFS-Ganesha pool. *POOL-NAME* is **nfs-ganesha**, and *APPLICATION-NAME* is the name of the application you want to enable, such as **cephfs**, **rbd**, or **rgw**.

   Syntax

   ```
   ceph osd pool application enable POOL-NAME APPLICATION_NAME
   ```

   Example

   ```
   [root@host01 ~]# ceph osd pool application enable nfs-ganesha rgw
   ```

IMPORTANT

The **cephadm-adopt** playbook does not bring up rbd-mirroring after migrating the storage cluster from RHCS 4 to RHCS 5.

To work around this issue, add the peers manually:

Syntax

```
rbd mirror pool peer add POOL_NAME CLIENT_NAME@CLUSTER_NAME
```

Example

```
[ceph: root@host01 /]# rbd --cluster site-a mirror pool peer add image-pool
client.rbd-mirror-peer@site-b
```

For more information, see Viewing information about peers.

Additional Resources

- For more information about using **leapp** to upgrade Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8, refer to Upgrading from Red Hat Enterprise Linux 7 to Red Hat Enterprise Linux 8.
- For more information about converting from FileStore to BlueStore, refer to BlueStore.
- For more information about storage peers, see Viewing information about peers.

CHAPTER 5. UPGRADE A RED HAT CEPH STORAGE CLUSTER USING CEPHADM

As a storage administrator, you can use the **cephadm** Orchestrator to upgrade Red Hat Ceph Storage 5.0 and later.

The automated upgrade process follows Ceph best practices. For example:

- The upgrade order starts with Ceph Managers, Ceph Monitors, then other daemons.
- Each daemon is restarted only after Ceph indicates that the cluster will remain available.

The storage cluster health status is likely to switch to **HEALTH_WARNING** during the upgrade. When the upgrade is complete, the health status should switch back to HEALTH_OK.

NOTE

ceph-ansible is currently not supported with Red Hat Ceph Storage 5. This means that once you have migrated your storage cluster to Red Hat Ceph Storage 5, you must use **cephadm** and **cephadm-ansible** to perform subsequent updates.

NOTE

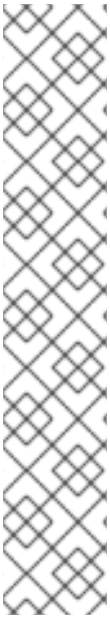
You do not get a message once the upgrade is successful. Run **ceph versions** and **ceph orch ps** commands to verify the new image ID and the version of the storage cluster.

5.1. UPGRADING THE RED HAT CEPH STORAGE CLUSTER

You can use **ceph orch upgrade** command for upgrading a Red Hat Ceph Storage 5.0 cluster.

Prerequisites

- A running Red Hat Ceph Storage cluster 5.0.
- Root-level access to all the nodes.
- At least two Ceph Manager nodes in the storage cluster: one active and one standby.



NOTE

Red Hat Ceph Storage 5 also includes a health check function that returns a DAEMON_OLD_VERSION warning if it detects that any of the daemons in the storage cluster are running multiple versions of RHCS. The warning is triggered when the daemons continue to run multiple versions of RHCS beyond the time value set in the **mon_warn_older_version_delay** option. By default, the **mon_warn_older_version_delay** option is set to 1 week. This setting allows most upgrades to proceed without falsely seeing the warning. If the upgrade process is paused for an extended time period, you can mute the health warning:

```
ceph health mute DAEMON_OLD_VERSION --sticky
```

After the upgrade has finished, unmute the health warning:

```
ceph health unmute DAEMON_OLD_VERSION
```

Procedure

1. Update the **cephadm-ansible** package.

   Example

   ```
   [root@host01 ~] dnf update cephadm-ansible
   ```

2. Run the preflight playbook with the **upgrade_ceph_packages** parameter set to **true** on the bootstrapped host in the storage cluster:

   Syntax

   ```
   ansible-playbook -i INVENTORY-FILE cephadm-preflight.yml --extra-vars "ceph_origin=rhcs
   upgrade_ceph_packages=true"
   ```

   Example

   ```
   [root@host01 ~]# ansible-playbook -i /etc/ansible/hosts/ cephadm-preflight.yml --extra-vars
   "ceph_origin=rhcs upgrade_ceph_packages=true"
   ```

   This package upgrades **cephadm** on all the nodes.

3. Log into the **cephadm** shell:

   Example

   ```
   [root@host01 ~]# cephadm shell
   ```

4. Ensure all the hosts are online and that the storage cluster is healthy:

   Example

   ```
   [ceph: roothost01 /]# ceph -s
   ```

5. Check service versions and the available target containers:

   Syntax

   ```
   ceph orch upgrade check {--image IMAGE_NAME | --ceph-version VERSION}
   ```

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade check --ceph-version 16.2.0-117.el8cp
   ```

6. Upgrade the storage cluster:

   Syntax

   ```
   ceph orch upgrade start {--image IMAGE_NAME | --ceph-version VERSION}
   ```

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade start --ceph-version 16.2.0-117.el8cp
   ```

   While the upgrade is underway, a progress bar appears in the **ceph status** output.

   Example

   ```
   [cephadm@cephadm /~]# ceph status
   [...]
   progress:
      Upgrade to 16.2.0-115.el8cp (1s)
        [.............................]
   ```

7. Verify the new *IMAGE_ID* and *VERSION* of the Ceph cluster:

   Example

   ```
   [ceph: roothost01 /]# ceph versions
   [ceph: roothost01 /]# ceph orch ps
   ```

5.2. UPGRADING THE RED HAT CEPH STORAGE CLUSTER IN A DISCONNECTED ENVIRONMENT

You can upgrade the storage cluster in a disconnected environment by using the **--image** tag.

You can use **ceph orch upgrade** command for upgrading a Red Hat Ceph Storage 5.0 cluster.

Prerequisites

- A running Red Hat Ceph Storage cluster 5.0.
- Root-level access to all the nodes.
- At least two Ceph Manager nodes in the storage cluster: one active and one standby.

- Register the nodes to CDN and attach subscriptions.
- Check for the customer container images in a disconnected environment and change the configuration, if required. See the *Configuring a custom registry for disconnected installation* section in the *Red Hat Ceph Storage Installation Guide* for more details.

Procedure

1. Update the **cephadm-ansible** package.

   Example

   ```
   [root@host01 ~] dnf update cephadm-ansible
   ```

2. Run the preflight playbook with the **upgrade_ceph_packages** parameter set to **true** on the bootstrapped host in the storage cluster:

   Syntax

   ```
   ansible-playbook -i INVENTORY-FILE cephadm-preflight.yml --extra-vars "ceph_origin=rhcs
   upgrade_ceph_packages=true"
   ```

   Example

   ```
   [root@host01 ~]# ansible-playbook -i /etc/ansible/hosts/ cephadm-preflight.yml --extra-vars
   "ceph_origin=rhcs upgrade_ceph_packages=true"
   ```

   This package upgrades **cephadm** on all the nodes.

3. Log into the **cephadm** shell:

   Example

   ```
   [root@host01 ~]# cephadm shell
   ```

4. Ensure all the hosts are online and that the storage cluster is healthy:

   Example

   ```
   [ceph: roothost01 /]# ceph -s
   ```

5. Check service versions and the available target containers:

   Syntax

   ```
   ceph orch upgrade check --image IMAGE_NAME
   ```

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade check --image
   _LOCAL_NODE_FQDN_:5000/rhceph/rhceph-5-rhel8
   ```

6. Upgrade the storage cluster:

   Syntax

   ```
   ceph orch upgrade start --image IMAGE_NAME
   ```

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade start --image
   _LOCAL_NODE_FQDN_:5000/rhceph/rhceph-5-rhel8
   ```

   While the upgrade is underway, a progress bar appears in the **ceph status** output.

   Example

   ```
   [cephadm@cephadm /~]# ceph status
   [...]
   progress:
      Upgrade to 16.2.0-115.el8cp (1s)
        [.............................]
   ```

7. Verify the new *IMAGE_ID* and *VERSION* of the Ceph cluster:

   Example

   ```
   [ceph: roothost01 /]# ceph orch versions
   [ceph: roothost01 /]# ceph orch ps
   ```

Additional Resources

- See the *Registering Red Hat Ceph Storage nodes to the CDN and attaching subscriptions* section in the *Red Hat Ceph Storage Installation Guide*.
- See the *Configuring a custom registry for disconnected installation* section in the *Red Hat Ceph Storage Installation Guide*.

5.3. MONITORING AND MANAGING UPGRADE OF THE STORAGE CLUSTER

After running the **ceph orch upgrade start** command to upgrade the Red Hat Ceph Storage cluster, you can check the status, pause, resume, or stop the upgrade process.

Prerequisites

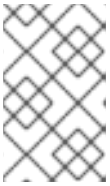
- A running Red Hat Ceph Storage cluster 5.0.
- Root-level access to all the nodes.
- At least two Ceph Manager nodes in the storage cluster: one active and one standby.
- Upgrade for the storage cluster initiated.

Procedure

1. Determine whether an upgrade is in process and the version to which the cluster is upgrading:

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade status
   ```

   NOTE

   You do not get a message once the upgrade is successful. Run **ceph versions** and **ceph orch ps** commands to verify the new image ID and the version of the storage cluster.

2. Optional: Pause the upgrade process:

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade pause
   ```

3. Optional: Resume a paused upgrade process:

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade resume
   ```

4. Optional: Stop the upgrade process:

   Example

   ```
   [ceph: roothost01 /]# ceph orch upgrade stop
   ```

CHAPTER 6. WHAT TO DO NEXT?

As a storage administrator, once you have installed and configured Red Hat Ceph Storage 5, you are ready to perform "Day Two" operations for your storage cluster. These operations include adding metadata servers (MDS) and object gateways (RGW), and configuring services such as iSCSI and NFS.

For more information about how to use the **cephadm** orchestrator to perform "Day Two" operations, refer to the *Red Hat Ceph Storage 5 Operations Guide*.

To deploy, configure, and administer the Ceph Object Gateway on "Day Two" operations, refer to the *Red Hat Ceph Storage 5 Object Gateway Guide*.

6.1. TROUBLESHOOTING UPGRADE ERROR MESSAGES

The following table shows some **cephadm** upgrade error messages. If the **cephadm** upgrade fails for any reason, an error message appears in the storage cluster health status.

Error Message	Description
UPGRADE_NO_STANDBY_MGR	Ceph requires both active and standby manager daemons in order to proceed, but there is currently no standby.
UPGRADE_FAILED_PULL	Ceph was unable to pull the container image for the target version. This can happen if you specify a version or container image that does not exist (e.g., 1.2.3), or if the container registry is not reachable from one or more hosts in the cluster.

Additional Resources

- Stopping and restarting the **cephadm** upgrade process

APPENDIX A. COMPARISON BETWEEN CEPH ANSIBLE AND CEPHADM

The Red Hat Ceph Storage 5.0 introduces a new deployment tool, Cephadm, for containerized deployment of the storage cluster.

The tables compare Cephadm with Ceph-Ansible playbooks for managing the containerized deployment of a Ceph cluster for day one and day two operations.

Table A.1. Day one operations

Description	Ceph-Ansible	Cephadm
Installation of the Red Hat Ceph Storage cluster	Run the **site-container.yml** playbook.	Run **cephadm bootstrap** command to bootstrap the cluster on the admin node.
Addition of hosts	Use the Ceph Ansible inventory.	Run **ceph orch add host *HOST_NAME*** to add hosts to the cluster.
Addition of monitors	Run the **add-mon.yml** playbook.	Run the `ceph orch apply mon ` command.
Addition of managers	Run the **site-container.yml** playbook.	Run the **ceph orch apply mgr** command.
Addition of OSDs	Run the **add-osd.yml** playbook.	Run the **ceph orch apply osd** command to add OSDs on all available devices or on specific hosts.
Addition of OSDs on specific devices	Select the **devices** in the **osd.yml** file and then run the **add-osd.yml** playbook.	Select the **paths** filter under the **data_devices** in the **osd.yml** file and then run **ceph orch apply -i *FILE_NAME*.yml** command.
Addition of MDS	Run the **site-container.yml** playbook.	Run the **ceph orch apply *FILESYSTEM_NAME*** command to add MDS.
Addition of Ceph Object Gateway	Run the **site-container.yml** playbook.	Run the **ceph orch apply rgw** commands to add Ceph Object Gateway.

Table A.2. Day two operations

Description	Ceph-Ansible	Cephadm
Removing hosts	Use the Ansible inventory.	Run **ceph orch host rm *HOST_NAME*** to remove the hosts.
Removing monitors	Run the **shrink-mon.yml** playbook.	Run **ceph orch apply mon** to redeploy other monitors.
Removing managers	Run the **shrink-mon.yml** playbook.	Run **ceph orch apply mgr** to redeploy other managers.
Removing OSDs	Run the **shrink-osd.yml** playbook.	Run **ceph orch osd rm *OSD_ID*** to remove the OSDs.
Removing MDS	Run the **shrink-mds.yml** playbook.	Run **ceph orch rm *SERVICE_NAME*** to remove the specific service.
Exporting Ceph File System over NFS Protocol.	Not supported on Red Hat Ceph Storage 4.	Run **ceph nfs export create** command.
Deployment of Ceph Object Gateway	Run the **site-container.yml** playbook.	Run **ceph orch apply rgw SERVICE_NAME** to to deploy Ceph Object Gateway service.
Removing Ceph Object Gateway	Run the **shrink-rgw.yml** playbook.	Run **ceph orch rm *SERVICE_NAME*** to remove the specific service.
Deployment of iSCSI gateways	Run the **site-container.yml** playbook.	Run **ceph orch apply iscsi** to deploy iSCSI gateway.
Block device mirroring	Run the **site-container.yml** playbook.	Run **ceph orch apply rbd-mirror** command.
Minor version upgrade of Red Hat Ceph Storage	Run the **infrastructure-playbooks/rolling_update.yml** playbook.	Run **ceph orch upgrade start** command.
Upgrading from Red Hat Ceph Storage 4 to Red Hat Ceph Storage 5	Run **infrastructure-playbooks/rolling_update.yml** playbook.	Upgrade using Cephadm is not supported.
Deployment of monitoring stack	Edit the **all.yml** file during installation.	Run the **ceph orch apply -i *FILE*.yml** after specifying the services.

Additional Resources

- For more details on using the Ceph Orchestrator, see the *Red Hat Ceph Storage Operations Guide*.