Red Hat Ceph Storage Getting started

Learn about how to get started with Red Hat Ceph Storage and its basic features. This information is designed for customers that are new to Red Hat Ceph Storage or established customers wanting an overview of how Red Hat Ceph Storage works and where to start in the workflow.

This information provides basic workflows for Red Hat Ceph Storage. For detailed instructions links to the correct documentation section are provided.

Before you begin to start working with Red Hat Ceph Storage, familiarize yourself with the following information:

- “Compatibility matrix for Red Hat Ceph Storage 9.0” on page 0
- “Minimum hardware considerations” on page 0
- “Network considerations for Red Hat Ceph Storage” on page 0
- “Basic considerations” on page 0

Object storage

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. Ceph Object Gateway, also known as RADOS Gateway (RGW), is an object storage interface built on top of the librados library to provide applications with a RESTful gateway to Ceph storage clusters.

Go through the following sections to get started with Ceph Object Gateways (object storage)

Object storage interfaces

The following are the object storage interfaces:

Administrative API

Provides an administrative interface for managing the Ceph Object Gateways. For more information, see Ceph Object Gateway administrative API.

S3

Provides object storage functionality with an interface that is compatible with a large subset of the Amazon S3 RESTful API. For more information, see Ceph Object Gateway and the S3 API.

Swift

Provides object storage functionality with an interface that is compatible with a large subset of the OpenStack Swift API. The Ceph Object Gateway is a service interacting with a Ceph storage cluster. For more information, see Ceph Object Gateway and the Swift API.

Common workloads

The following are the most common workloads for using Ceph Object Gateway:

Data efficiency

Use for erasure coding, thin provisioning, lifecycle management, and compression.

Data security

Use for object lock, key management, at rest and inflight encryption, and WORM.

Data resilience

Use for backup, snapshots, cloning, and business continuity.

Common use cases

The following are some of the most common uses cases for Ceph Object Gateway.

Storage as a Service (SaaS)

Provides scalability and performance for both small and large object stores.

AI, Analytics and Big Data including Data Lake and Data Lake House

Cloud native data lake, with massive scalability and high availability to support demanding workloads. For example, you can use Red Hat Ceph Storage to provide an external S3 object store for IBM watsonx.data.

Backup and archive or large amounts of unstructured data

A unique new way of architecting the dataflow in applications which is through event driven architectures.

Data intensive workloads like Cloud Native (S3) object data

Back up and recover into and from an object storage helps improve recovery time objectives (RTO) and recovery point objectives (RPO).

Internet of Things (IoT)

Object gateways serve as intermediaries in IoT systems, aggregating data from various devices, translating communication protocols, and enabling edge processing. They enhance security, facilitate device management, and ensure interoperability, streamlining the overall IoT ecosystem.

Getting started with object storage

Get to know the relevant tasks that are required for working with Ceph Object Gateway.

See Compatibility matrix for a list of backup vendors that are certified with Red Hat Ceph Storage as an S3 target.

There are specific network and hardware requirements to work with Ceph object storage. For more information, see Considerations and recommendations.

Managing Object Gateway through the dashboard

For detailed information, see “Managing Ceph Object Gateway” on page 0.

Deploying the Ceph Object Gateway using CLI

For detailed information, see Deploying the Ceph Object Gateway using CLI.

Setting up Ingress service

For detailed information, see High availability service.

Setting up S3 server-side security

For detailed information, see Server side encryption.

Multi-site Asynchronous replication for High Availability and Disaster Recovery

For detailed information, see Failover and disaster recovery.

Deploying Ceph Object Gateway with Ceph Orchestrator

Ceph Object Gateway is deployed by either using the Ceph Orchestrator with the command line interface or by using the service specification. You can also configure multi-site Ceph Object Gateways, and remove the Ceph Object Gateway using the Ceph Orchestrator. The **`cephadm`** command deploys the Ceph Object Gateway as a collection of daemons that manages a single-cluster deployment or a particular realm and zone in a multi-site deployment.

For full Ceph Object Gateway deployment information and instructions, see Deployment.

Alternatively, you can deploy Ceph Object Gateway using the command-line interface. For more information, see Deploying the Ceph Object Gateway using the command line interface.

Creating S3 users and testing S3 access

- For detailed information on creating S3 users, see Create an S3 user.
- For detailed information on testing S3 access, see Test S3 access.
- For detailed information about managing Ceph Object Gateway S3 users, see Management of Ceph Object Gateway users.

Block storage

Red Hat Ceph Storage uses block storage, and refers to this as Ceph Block Devices. Block-based storage interfaces are the most common way to store data with rotating media such as hard drives and both SSD and NVMe flash storage.

Ceph Block Devices interact with OSDs by using the `librbd` library.

Ceph Block Devices deliver high performance with infinite scalability to Kernel Virtual Machines (KVMs), such as Quick Emulator (QEMU), and cloud-based computing systems, like OpenStack, that rely on the `libvirt` and QEMU utilities to integrate with Ceph Block Devices. You can use the same storage cluster to operate the Ceph Object Gateway and Ceph Block Devices simultaneously.

Ceph Block Devices can easily be managed through either the Ceph dashboard or command-line interface (CLI) commands. For detailed information about Ceph Block Devices, see Ceph Block Devices.

Common workloads

The following are the most common workloads for using Ceph Block Devices:

Database store

Use for data protection application database backup.

Device mirroring

Use to protect against data loss or site failures.

Data resiliency

Use for replication and erasure coding.

Common use cases

The following are some of the most common uses cases for Ceph Block Devices:

- Storage for Virtualization, Hypervisors, and private clouds.
- Persistent storage for containers (Kubernetes).
- Storage for applications that require low-latency and high performance, such as databases.
- Virtual machines for VMware, OpenStack and OpenShift.
- Unified file and object storage.
  - Ingest data via file protocols from existing (legacy) enterprise applications into a common repository based on an object store.
  - Once it is in the repository, access the data via S3 for analytics, machine-learning, and so on.

Example workflow for using Ceph Block Device snapshots and clones

Learn the generic working configurations that can be configured within the context of your environment. For detailed information about the procedures see the relevant sections within the documentation.

This information provides an example workflow for using Ceph Block Device snapshots and clones. This workflow includes creating, listing, renaming, layering, protecting, and cloning Ceph Block Device snapshots.

For detailed information, see Managing snapshots.

1. Create, list, and rename Ceph Block Device snapshots.

```
[root@rbd-client ~]# rbd --pool pool1 snap create --snap snap1 image1
[root@rbd-client ~]# rbd --pool pool1 --image image1 snap ls
[root@rbd-client ~]# rbd snap ls pool1/image1
[root@rbd-client ~]# rbd snap rename data/dataset@snap1 data/dataset@snap2
```

2. Layer Ceph Block Device snapshots.
   Within the `ceph.conf` file, enable this feature by adding `rbd_clone_copy_on_read = true` into either the `[global]` or `[client]` section.

   For detailed information, see Ceph Block Device layering.

3. Protect and clone Ceph Block Device snapshots.

```
[root@rbd-client ~]# ceph osd set-require-min-compat-client mimic
[root@rbd-client ~]# rbd clone --pool pool1 --image image1 --snap snap2 --dest-pool
pool2 --dest childimage1
[root@rbd-client ~]# rbd clone pool1/image1@snap1 pool1/childimage1
```

Getting started with block storage

Get to know the relevant tasks and most common commands that are required for working with Ceph Block Devices.

Managing Ceph Block Devices with the dashboard

Manage Ceph Block Devices by using the Red Hat Ceph Storage dashboard. As a storage administrator, you can manage and monitor block device images on the Red Hat Ceph Storage dashboard. The functionality is divided between generic image functions and mirroring functions. For example, you can create new images, view the state of images that are mirrored across clusters, and set IOPS limits on an image.

For detailed information, see Managing block devices.

Common block image CLI commands

This information is for a quick reference of basic block image CLI commands. For a full list and more detailed information about each command, see Ceph Block Devices.

Creating images

```
rbd create IMAGE_NAME --size MEGABYTES --pool POOL_NAME
```

> **Important:** A pool must be created before creating a block image. For details, see “Creating block device pools” on page 0.

Listing images

```
rbd ls POOL_NAME
```

Retrieving image information from a particular image in the default pool

```
rbd --image IMAGE_NAME info
```

Retrieving information from an image within a pool

```
rbd --image IMAGE_NAME -p POOL_NAME info
```

Resizing images

Increasing the maximum size of a Ceph Block Device image for the default `rbd` pool

```
rbd resize --image IMAGE_NAME --size SIZE
```

Increasing the maximum size of a Ceph Block Deivce image for a specific pool

```
rbd resize --image POOL_NAME/IMAGE_NAME --size SIZE
```

Decreasing the maximum size of a Ceph Block Device image for the default `rbd` pool

```
rbd resize --image IMAGE_NAME --size SIZE --allow-shrink
```

Decreasing the maximum size of a Ceph Block Device image for a specific pool

```
rbd resize --image POOL_NAME/IMAGE_NAME --size SIZE --allow-shrink
```

Moving images to the trash

```
rbd trash mv POOL_NAME/IMAGE_NAME
```

Restoring an image from the trash

```
rbd trash restore POOL_NAME/IMAGE_NAME
```

Ensuring the `rbd_support` Ceph Manager module is enabled

```
ceph mgr module ls
```

File storage

Red Hat Ceph Storage provides file storage with the Ceph File System (CephFS).

CephFS provides shared file access to a Red Hat Ceph Storage cluster and uses POSIX semantics wherever possible. CephFS is built on top of the Ceph distributed object store, called RADOS (Reliable Autonomic Distributed Object Store). CephFS inherits all of the architectural benefits of RADOS, providing high availability, built-in data redundancy, shared and parallel data access and an infinitely scalable architecture.

Common use cases

The following are some of the most common use cases for CephFS:

- File System as a Service (FSaaS).
- General purpose network file system.
- Home directories.
- Applications that require multiple readers, writers, or both. For example, High Performance Computing (HPC).
- Media and content repositories.

Example use cases for file storage as a service (FSaaS)

Get to know the generic working configurations that can be configured within the context of your environment. For detailed information about the procedures see the relevant sections within the documentation.

See the following workflows to get to know the main use cases for FSaaS:

- “Example workflow for using Ceph File System” on page 5
- “Example workflow for using CephFS snapshot” on page 6

Example workflow for using Ceph File System

This information provides an example workflow for using CephFS.

1. Create a Ceph File system volume or subvolume.
   For detailed information, see “Managing CephFS volumes” on page 0.

   ```
   [root@client01 ~]# subscription-manager repos --enable=rhceph-9-tools-for-rhel-9-
   x86_64-rpms
   [root@client01 ~]# dnf install ceph-common
   [root@host01 ~]# cephadm shell
   [root@client ~]# dnf install ceph-fuse
   [root@client ~]# scp root@192.168.0.1:/etc/ceph/ceph.client.1.keyring /etc/ceph/
   [root@client ~]# scp root@192.168.0.1:/etc/ceph/ceph.conf /etc/ceph/ceph.conf
   [root@client ~]# chmod 644 /etc/ceph/ceph.conf
   [root@mon ~]# ceph fs volume create cephfs01
   ```

2. Set Ceph client credentials for CephFS.
   For detailed information, see Creating client users for a Ceph File System.

Part of setting client credentials is to restrict the client to either the entire `temp` directory or to only write in the `temp` directory of file system `cephfs_a`. This example provides the instructions for restricting the client to the `temp` directory.

```
[ceph: root@host01 ~]# ceph fs authorize cephfs_a client.1 /temp rw
[ceph: root@host01 ~]# ceph auth get client.1

client.1
  key = AQBSdFhcGZFUDRAAcKhG9Cl2HPiDMMRv4DC43A==
  caps mds = "allow r, allow rw path=/temp"
  caps mon = "allow r"
  caps osd = "allow rw tag cephfs data=cephfs_a"
[ceph: root@host01 ~]# ceph auth get client.1 -o ceph.client.1.keyring
exported keyring for client.1
[ceph: root@host01 ~]# scp /ceph.client.1.keyring root@client01:/etc/ceph/
ceph.client.1.keyring
[root@client01 ~]# chmod 644 /etc/ceph/ceph.client.1.keyring
```

3. Mount a CephFS volume on a client. The volume can be mounted as either a kernel or a FUSE client.

   Example for mounting file storage as a kernel client with automatic mounting

   ```
   [root@client01 ~]# ceph auth get-key client.1 > /etc/ceph/1.secreat
   mount -t ceph 10.0.195.41,10.0.195.177,10.0.195.78:/ /mnt/cephfs -o
   name=1,secretfile=/etc/ceph/1.secret,fs=cephfs-ec
   ```

   For detailed information, see Deploying the CephFS.

   Example for mounting file storage as a FUSE client

   ```
   [root@client ~]# mkdir /mnt/mycephfs
   [root@client ~]# ceph-fuse /mnt/mycephfs/ -n client.1 --client-fs=cephfs01
   ceph-fuse[555001]: starting ceph client
   2022-05-09T07:33:27.158+0000 7f11feb81200 -1 init, newargv = 0x55fc4269d5d0
   newargc=15
   ceph-fuse[555001]: starting fuse
   ```

Example workflow for using CephFS snapshot

This information provides an example workflow for using CephFS snapshots. For detailed information, see “Creating a snapshot for a Ceph File System” on page 0.

1. Create a snapshot for a CephFS.

   ```
   [ceph: root@host01 ~]# cd /mnt/cephfs
   [ceph: root@host01 ~]# touch before_snapshot
   [ceph: root@host01 cephfs ~]# mkdir .snap/newsnap
   ```

   Note: The CLI command can be used to create subvolume snapshots. While `mkdir` and `rmdir` can still be used to create and delete snapshots, the CLI command is the most current method and is recommended for use.

2. Verify and use the snapshot.

   ```
   [ceph: root@host01 cephfs ~]# touch after_snapshot
   [ceph: root@host01 cephfs ~]# ls -al .snap/newsnap
   ```

3. Delete the snapshots from a CephFS.

   Important: Attempting to delete root-level snapshots, which might contain underlying snapshots, will fail.

   ```
   [ceph: root@host01 ~]# rmdir .snap/newsnap
   [ceph: root@host01 ~]# ls -al .snap/
   ```

Getting started with file storage

Get to know the relevant tasks that are required for working with Ceph File Systems (CephFS).

Limitations

To know about the limitations and POSIX standards to consider when working with Ceph File System, see “CephFS limitations and the POSIX standards” on page 0.

Deploying the Ceph File System

Detailed deployment instructions for Ceph File System is found in “Deploying the CephFS” on page 0.

Managing Ceph File Systems through the dashboard

For detailed information on managing Ceph File Systems from the dashboard, see “Managing Ceph File Systems (CephFS)” on page 0.

Creating client users for a Ceph File System

For detailed information, see “Creating client users for a Ceph File System” on page 0 .

Setting up the Ceph File System

Use the following procedures to setup a Ceph File System.

1. “Creating client users for a Ceph File System” on page 0.
2. “Mounting the Ceph File System as a kernel client” on page 0 or “Mounting the Ceph File System as a FUSE client” on page 0.
3. “Client features” on page 0.

Basic Ceph File System CLI commands

This information is for a quick reference of basic CephFS CLI commands. For a full list and more detailed information about each command, see “Managing CephFS volumes” on page 0.

Creating a CephFS volume

```
ceph fs volume create VOLUME_NAME
```

Creating a CephFS subvolume

```
ceph fs subvolume create VOLUME_NAME SUBVOLUME_NAME [--size SIZE_IN_BYTES --group_name
SUBVOLUME_GROUP_NAME --pool_layout DATA_POOL_NAME --uid UID --gid GID --mode OCTAL_MODE]
[--namespace-isolated]
```

Creating a CephFS subvolume group

```
ceph fs subvolumegroup create VOLUME_NAME GROUP_NAME [--pool_layout DATA_POOL_NAME --uid
UID --gid GID --mode OCTAL_MODE]
```

Listing CephFS volumes

```
ceph fs volume ls
```

Listing CephFS subvolumes

```
ceph fs subvolume ls VOLUME_NAME
```

Listing CephFS subvolume groups

```
ceph fs subvolumegroup ls VOLUME_NAME
```

Viewing information about a CephFS volume

```
ceph fs volume info VOLUME_NAME
```

Removing a CephFS volume

```
ceph fs volume rm VOLUME_NAME [--yes-i-really-mean-it]
```

Removing a CephFS subvolume

```
ceph fs subvolume rm VOLUME_NAME SUBVOLUME_NAME
```

Removing a CephFS subvolume group

```
ceph fs subvolumegroup rm VOLUME_NAME GROUP_NAME [--force]
```

Creating a snapshot of a CephFS subvolume

```
ceph fs subvolume snapshot create VOLUME_NAME SUBVOLUME_NAME SNAP_NAME [--group_name
GROUP_NAME]
```