



Red Hat Ceph Storage 8

Getting Started Guide

Guide on getting started with Red Hat Ceph Storage

Red Hat Ceph Storage 8 Getting Started Guide

Guide on getting started with Red Hat Ceph Storage

Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document provides information on getting started with Object storage, Block storage, and File storage for a Red Hat Ceph Storage cluster. 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. GETTING STARTED	3
CHAPTER 2. OBJECT STORAGE	4
2.1. OBJECT STORAGE COMMON WORKLOADS	4
2.2. OBJECT STORAGE INTERFACES	4
2.3. GETTING STARTED WITH OBJECT STORAGE	5
CHAPTER 3. BLOCK STORAGE	6
3.1. BLOCK STORAGE COMMON WORKLOADS	6
3.2. GETTING STARTED WITH BLOCK STORAGE	6
CHAPTER 4. FILE STORAGE	9
4.1. FILE STORAGE COMMON WORKLOADS	9
4.2. GETTING STARTED WITH FILE STORAGE	9
4.3. DEPLOYING THE CEPH FILE SYSTEM	11

CHAPTER 1. 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 sections are provided.

Use this information to understand how to use Red Hat Ceph Storage basic workflows. 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 8.0](#)
- [Minimum hardware considerations](#)
- [Network considerations for Red Hat Ceph Storage](#)
- [Basic Red Hat Ceph Storage considerations](#)

CHAPTER 2. 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.

Common use cases

The following are some of the most common uses cases for CephFS.

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.

Backup and archive or large amounts of unstructured data

A unique new way to architect 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, streaming the overall IoT ecosystem.

2.1. OBJECT STORAGE COMMON WORKLOADS

Understand the most common workloads for object storage.

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.

2.2. OBJECT STORAGE INTERFACES

Learn about the three 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](#).

2.3. GETTING STARTED WITH OBJECT STORAGE

This section lists the relevant tasks required for working with object storage.

Before you begin

See [Compatibility Matrix for Red Hat Ceph Storage 7.0](#) for a list of backup vendors that are certified with Red Hat Ceph Storage as an S3 target.

Prerequisites

There are specific network and hardware requirements to work with Ceph object storage. For more information, see [Red Hat Ceph Storage considerations and recommendations](#).

Setting up S3 server-side security

For detailed information, see [Server-Side Encryption \(SSE\)](#).

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](#).

Managing Object Gateway through the dashboard

For detailed information, see [Management of Ceph Object Gateway using the dashboard](#).

Multi-site replication to enable Disaster Recovery of backup

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](#).

CHAPTER 3. 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 flash storage (SSDs and HDDs).

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 [Introduction to Ceph block devices](#).

3.1. BLOCK STORAGE COMMON WORKLOADS

Understand the most common workloads for Ceph Block Device.

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.

3.2. GETTING STARTED WITH BLOCK STORAGE

This section lists the relevant tasks required for working with block storage.

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 [Management of block devices using the Ceph dashboard](#).

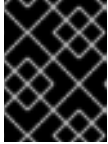
Common block storage CLI commands

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 [Introduction to Ceph block devices](#).

- Creating images

Syntax

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



IMPORTANT

A pool must be created before creating a block image. For details, see [Creating a block device pool](#).

- Listing images

Syntax

```
rbd ls POOL_NAME
```

- Retrieving image information from a particular image in the default pool

Syntax

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

- Retrieving information from an image within a pool.

Syntax

```
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.

Syntax

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

- Increasing the maximum size of a Ceph Block Device image for a specific pool.

Syntax

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

- Decreasing the maximum size of a Ceph Block Device image for the default **rbd** pool.

Syntax

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

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

Syntax

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

- Moving images to the trash.

Syntax

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

- Restoring an image from the trash.

Syntax

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

- Ensuring the **rbd_support** Ceph Manager module is enabled.

Syntax

```
ceph mgr module ls
```

CHAPTER 4. FILE STORAGE

Ceph File System (CephFS) is a file system compatible with POSIX standards that is built on top of Ceph's distributed object store, called RADOS (Reliable Autonomic Distributed Object Storage).

Ceph File System

CephFS provides file access to a Red Hat Ceph Storage cluster and uses the POSIX semantics wherever possible.

4.1. FILE STORAGE COMMON WORKLOADS

The most common workload for using CephFS is for data security.

For more information about securing your data by using CephFS, see [Ceph File System](#).

4.2. GETTING STARTED WITH FILE STORAGE

This section lists the relevant tasks required for working with file storage.

Limitations

To know about the limitations and POSIX standards to consider when working with Ceph File System, see [Ceph File System limitations and the POSIX standards](#).

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](#).
2. [Mounting the Ceph File System as a kernel client](#) or [Mounting the Ceph File System as a FUSE client](#).
3. [Client features](#).

Basic CephFS 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 [Management of Ceph File System volumes, sub-volume groups, and sub-volumes](#).

- Creating a Ceph File System volume.

Syntax

```
ceph fs volume create VOLUME_NAME
```

- Creating a File System subvolume.

Syntax

```
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 File System subvolume group.

Syntax

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

- Listing Ceph File System volumes.

Syntax

```
ceph fs volume ls
```

- Listing Ceph File System subvolumes.

Syntax

```
ceph fs subvolume ls VOLUME_NAME
```

- Listing Ceph File System subvolume groups.

Syntax

```
ceph fs subvolume group ls VOLUME_NAME
```

- Viewing information about a Ceph File system volume.

Syntax

```
ceph fs volume info VOLUME_NAME
```

- Removing a Ceph File System volume.

Syntax

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

- Removing a file system subvolume.

Syntax

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

- Removing a file system subvolume group.

Syntax

```
ceph fs subvolume group rm VOLUME_NAME GROUP_NAME [--force]
```

- Creating a snapshot of a file system subvolume.

Syntax

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

4.3. DEPLOYING THE CEPH FILE SYSTEM

Understand the basic deployment procedures for Ceph File System.

Detailed deployment instructions for Ceph File System is found in [Deployment of the Ceph File System](#).