Upgrading Red Hat Ceph Storage

Upgrade to an Red Hat Ceph Storage cluster running Red Hat Enterprise Linux on AMD64 and Intel 64 architectures.

Supported upgrade paths

Upgrade to Red Hat Ceph Storage based on the support that is provided by Red Hat.

Red Hat upgrade paths

Supported upgrade paths details the upgrade paths that are supported to upgrade from older Red Hat Ceph Storage versions to Red Hat Ceph Storage 9.

Supported Red Hat Ceph Storage upgrade paths	
Pre-upgrade	**Post-upgrade**
Red Hat Ceph Storage 7[1]	Red Hat Ceph Storage 9
Red Hat Ceph Storage 8	Red Hat Ceph Storage 9
Red Hat Ceph Storage 9.0	Red Hat Ceph Storage 9.1

[1]To upgrade Ceph Object Gateway (RGW) with multi-site to use`cephadm` self-signed certificates from Red Hat Ceph Storage 7.1, see "Updating Ceph Object Gateway to use cephadm self-signed certificates in a multi-site deployment" on page 8.

For more information and the latest supported configurations, see Red Hat Ceph Storage: Supported configurations.

Upgrading a Red Hat Ceph Storage cluster using `cephadm`

As a storage administrator, you can use the `cephadm` Orchestrator to upgrade to the latest version of Red Hat Ceph Storage by using the **`ceph orch upgrade`** command.

The automated upgrade process follows Ceph best practices.

- 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.

For information about upgrade paths, see "Supported upgrade paths" on page 1.

Before starting to upgrade Red Hat Ceph Storage, see Installation and upgrade requirements.

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:** 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.

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.

Red Hat recommends to use the `podman` version shipped with the corresponding Red Hat Enterprise Linux version for Red Hat Ceph Storage. See the Red Hat Ceph Storage: Supported configurations knowledge base article for more details. See the Contacting Red Hat support for service section in "Red Hat Ceph Storage Troubleshooting" on page 0 for additional assistance.

The following table shows version compatibility between Red Hat Ceph Storage 9 and versions of `podman`.

Important: You must use a version of Podman that is 2.0.0 or higher.

Red Hat Ceph Storage 9 and podman version compatibility						
Ceph	**Podman**					
	1.9	**2.0**	**2.1**	**2.2**	**3.0**	**>3.0**
Red Hat Ceph Storage 9	false	true	true	false	true	true

Upgrading the Red Hat Ceph Storage cluster

Use the **ceph orch upgrade** command to upgrade an Red Hat Ceph Storage cluster. Alternatively, update the cluster through the dashboard.

PREREQUISITE

Before you begin, make sure that you have the following prerequisites in place:

- A supported Red Hat Ceph Storage cluster version. For more information, see Red Hat Ceph Storage: Supported configurations.
- Root-level access to all the nodes.
- At least two Ceph Manager nodes in the storage cluster: one active and one standby.
- Ansible user with sudo and passwordless ssh access to all nodes in the storage cluster.

Note: Red Hat Ceph Storage includes a health check function that returns a DAEMON_OLD_VERSION warning if it detects 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 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 by using the **ceph health mute** command.

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

After the upgrade has finished, unmute the health warning, by using the **ceph health unmute** command.

```
ceph health unmute DAEMON_OLD_VERSION
```

Use either the command-line interface (CLI) or the Ceph Dashboard to upgrade a cluster. To update the cluster through the dashboard, see Upgrading a cluster. To upgrade through the CLI, continue with the instructions provided here.
The Ceph Orchestrator coordinates cluster upgrades by upgrading daemons in a controlled sequence while ensuring service availability. For OSD upgrades, the Orchestrator can limit upgrades to specific CRUSH failure domains, such as a rack, chassis, or host, enabling phased upgrades aligned with physical infrastructure boundaries.

For more information, see Troubleshooting upgrade error messages.

If a cluster host is or becomes unavailable the upgrade will be paused until it is restored.

To run a staggered upgrade, see “Running a staggered upgrade” on page 16.

1. Register the system, and when prompted, enter your Red Hat Customer Portal credentials.
   For example,

```
[root@admin ~]# subscription-manager register
```

2. Pull the latest subscription.

```
subscription-manager refresh
```

> **Note:** Red Hat customers now use Simple Content Access (SCA), as a result, attaching a subscription to the system is no longer necessary. Registering the system and enabling the required repositories is sufficient. For more information, see How to register and subscribe a RHEL system to the Red Hat Customer Portal using Red Hat Subscription-Manager? on the Red Hat Customer Portal.

3. Enable the Ceph Ansible repositories on the Ansible administration node.

```
subscription-manager repos --enable=rhceph-9-tools-for-rhel-9-x86_64-rpms
```

4. Update the system.

```
dnf update
```

5. Update the `cephadm`, `cephadm-ansible`, and `crun` packages on the admin node where `cephadm-ansible` is installed.

```
dnf update cephadm
```

```
dnf update cephadm-ansible
```

```
dnf update crun
```

For example,

```
[root@admin ~]# dnf update cephadm
[root@admin ~]# dnf update cephadm-ansible
[root@admin ~]# dnf update crun
```

6. Navigate to the `/usr/share/cephadm-ansible/` directory.

```
cd /usr/share/cephadm-ansible
```

7. Run the preflight playbook with the **upgrade_ceph_packages** parameter set to `true` on the bootstrapped host in the storage cluster.
This package upgrades `cephadm` on all the nodes.

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

For example,

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i hosts cephadm-preflight.yml --
extra-vars "ceph_origin=rhcs upgrade_ceph_packages=true"
```

8. Log in to the `cephadm` shell.

```
cephadm shell
```

For example,

```
[root@host01 ~]# cephadm shell
[ceph: root@host01 /]#
```

9. Ensure all the hosts are online and that the storage cluster is healthy, by using the **ceph -s** command.

For example,

```
[ceph: root@host01 /]# ceph -s
```

10. Set the OSD `noout`, `noscrub`, and `nodeep-scrub` flags.
Setting these flags prevents OSDs from getting marked out during upgrade and to avoid unnecessary load on the cluster.

```
ceph osd set noout
```

```
ceph osd set noscrub
```

```
ceph osd set nodeep-scrub
```

For example,

```
[ceph: root@host01 /]# ceph osd set noout
[ceph: root@host01 /]# ceph osd set noscrub
[ceph: root@host01 /]# ceph osd set nodeep-scrub
```

11. Check service versions and the available target containers.

> **Note:** The following procedure provides syntax and examples for installing with Red Hat Enterprise Linux 9.7. For installing with Red Hat Enterprise Linux 10.1, replace `rhel-9` with `rhel-10`, in each instance.

```
ceph orch upgrade check IMAGE_NAME
```

For example,

```
[ceph: root@host01 /]# ceph orch upgrade check registry.redhat.io/rhceph/
rhceph-9-rhel9:latest
```

12. Upgrade the storage cluster.

> **Note:** The following procedure provides syntax and examples for installing with Red Hat Enterprise Linux 9.7. For installing with Red Hat Enterprise Linux 10.1, replace `rhel-9` with `rhel-10`, in each instance.

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

For staggered upgrades where OSDs can be upgraded efficiently at specific CRUSH bucket scopes (such as rack, chassis, or host), see “Running a staggered upgrade” on page 16.

You can scope OSD upgrades to a specific failure domain, such as a rack. For more information, see “Upgrading OSDs by rack” on page 11.

For example,

```
ceph orch upgrade start --image registry.redhat.io/rhceph/rhceph-9-rhel9:latest
```

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

```
[ceph: root@host01 /]# ceph status
[...]
progress:
    Upgrade to 20.1.0-145.el9cp (1s)
      [.........................]
```

For more information about monitoring the upgrade status, see “Monitoring and managing upgrade” on page 21.

13. Verify the new *IMAGE_ID* and *VERSION* of the Ceph cluster.

```
ceph versions
```

```
ceph orch ps
```

> **Note:** If you are not using the `cephadm-ansible` playbooks, after upgrading your Ceph cluster, you must upgrade the `ceph-common` package and client libraries on your client nodes and then verify that you have the latest version.
> For example,
>
> ```
> [root@client01 ~] dnf update ceph-common
> [root@client01 ~] ceph --version
> ```

> **Important:** If you are upgrading from Red Hat Ceph Storage versions earlier than 6.1, the `ceph-exporter` service is not deployed automatically during the upgrade. The `ceph-exporter` service is required for the Dashboard and Prometheus to display Ceph cluster metrics. If the service is not deployed, the Dashboard shows `No Data`.

14. If required, deploy the `ceph-exporter` service.
    For example,

```
[ceph: root@host01 /]# ceph orch apply ceph-exporter --placement='*'
```

    a. Verify that the `ceph-exporter` is running.

```
ceph orch ls
```

15. When the upgrade is complete, unset the `noout`, `noscrub`, and `nodeep-scrub` flags.
    For example,

```
[ceph: root@host01 /]# ceph osd unset noout
[ceph: root@host01 /]# ceph osd unset noscrub
[ceph: root@host01 /]# ceph osd unset nodeep-scrub
```

AFTER COMPLETING THE TASK

To complete the upgrade, rotate CephX keys. For more information, see Rotating CephX keys.

Upgrading cluster in a disconnected (air-gapped) environment

Upgrade the Red Hat Ceph Storage cluster in a disconnected environment.

PREREQUISITE

Before you begin, make sure that you have the following prerequisites in place:

- 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. For more information, see Changing configurations of custom container images for disconnected installations.

Upgrade the storage cluster in a disconnected environment by using the **`--image`** tag.

By default, the monitoring stack components are deployed based on the primary Ceph image. For disconnected environment of the storage cluster, you have to use the latest available monitoring stack component images.

"Custom image details for the monitoring stack" on page 6 details the custom image details for each monitoring stack component.

Custom image details for the monitoring stack

Monitoring stack component	Image details
Ceph image	`registry.redhat.io/rhceph/rhceph-9-rhel9:latest`
Ceph image (image mode)	`registry.redhat.io/rhceph/alloy-rhel10:latest`
Prometheus	`registry.redhat.io/openshift4/ose-prometheus:v4.15`
Grafana	`registry.redhat.io/rhceph/grafana-rhel10:latest`
Node-exporter	`registry.redhat.io/openshift4/ose-prometheus-nodeexporter:v4.15`
AlertManager	`registry.redhat.io/openshift4/ose-prometheusalertmanager:v4.15`
HAProxy	`registry.redhat.io/rhceph/rhceph-haproxy-rhel10:latest`
Keepalived	`registry.redhat.io/rhceph/keepalived-rhel10:latest`
SNMP Gateway	`registry.redhat.io/rhceph/snmp-notifier-rhel10:latest`
OAuth2 Proxy	`registry.redhat.io/rhceph/oauth2-proxy-rhel10:v7.14`

Use the **`ceph orch upgrade`** command to start and check an upgrade to a specific tagged version.

1. Update the `cephadm` and `cephadm-ansible` package.

```
dnf update cephadm
dnf update cephadm-ansible
```

For example,

```
[root@admin cephadm-ansible]# dnf update cephadm
 [root@admin cephadm-ansible]# dnf update cephadm-ansible
```

2. Navigate to the `/usr/share/cephadm-ansible/` directory.

```
cd /usr/share/cephadm-ansible
```

3. Run the preflight playbook with the **`upgrade_ceph_packages`** parameter set to `true` and the **`ceph_origin`** parameter set to `custom` on the bootstrapped host in the storage cluster.

```
ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --extra-vars
"ceph_origin=custom upgrade_ceph_packages=true"
```

For example,

```
[ansible@admin cephadm-ansible]$ ansible-playbook -i /etc/ansible/hosts cephadm-
preflight.yml --extra-vars "ceph_origin=custom upgrade_ceph_packages=true"
```

This package upgrades `cephadm` on all the nodes.

4. Log into the `cephadm` shell.

```
cephadm shell
```

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

```
ceph -s
```

6. Check service versions and the available target containers.

```
ceph orch upgrade check IMAGE_NAME
```

For example,

```
[ceph: root@host01 /]# ceph orch upgrade check LOCAL_NODE_FQDN:5000/rhceph/
rhceph-9-rhel9
```

7. Set the OSD `noout`, `noscrub`, and `nodeep-scrub` flags to prevent OSDs from getting marked out during upgrade and to avoid unnecessary load on the cluster.

```
ceph osd set noout
ceph osd set noscrub
ceph osd set nodeep-scrub
```

```
[ceph: root@host01 /]# ceph osd set noout
 [ceph: root@host01 /]# ceph osd set noscrub
 [ceph: root@host01 /]# ceph osd set nodeep-scrub
```

8. Upgrade the storage cluster.

> **Note:** The following procedure provides syntax and examples for installing with Red Hat Enterprise Linux 9.7. For installing with Red Hat Enterprise Linux 10.1, replace `rhel-9` with `rhel-10`, in each instance.

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

For staggered upgrades where OSDs can be upgraded efficiently at specific CRUSH bucket scopes (such as rack, chassis, or host), see “Running a staggered upgrade” on page 16.

You can scope OSD upgrades to a specific failure domain, such as a rack. For more information, see “Upgrading OSDs by rack” on page 11.

For example,

```
ceph orch upgrade start --image registry.redhat.io/rhceph/rhceph-9-rhel9:latest
```

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

```
[ceph: root@host01 /]# ceph status
[...]
progress:
    Upgrade to 20.1.0-145.el9cp (1s)
      [.........................]
```

For more information about monitoring the upgrade status, see “Monitoring and managing upgrade” on page 21.

9. Verify the new *IMAGE_ID* and *VERSION* of the cluster.

```
ceph version
ceph versions
ceph orch ps
```

For example,

```
[ceph: root@host01 /]# ceph version
[ceph: root@host01 /]# ceph versions
[ceph: root@host01 /]# ceph orch ps
```

10. When the upgrade is complete, unset the `noout`, `noscrub`, and`nodeep-scrub` flags.

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

```
[ceph: root@host01 /]# ceph osd unset noout
[ceph: root@host01 /]# ceph osd unset noscrub
[ceph: root@host01 /]# ceph osd unset nodeep-scrub
```

AFTER COMPLETING THE TASK

To complete the upgrade, rotate CephX keys. For more information, see Rotating CephX keys.

Related information

Registering Red Hat Ceph Storage nodes to the CDN and attaching subscriptions

Configuring a private registry for a disconnected installation

Updating Ceph Object Gateway to use `cephadm` self-signed certificates in a multi-site deployment

Starting from Red Hat Ceph Storage 8.1, Red Hat Ceph Storage introduced a new root certificate authority (CA) format. As a result, when updating the Ceph Object Gateway in multi-site deployments from Red Hat Ceph Storage versions 8.0 or earlier, you must complete additional steps to enable `cephadm` self-signed certificates.

PREREQUISITE

Before you begin, make sure that your Red Hat Ceph Storage 9 storage system is using the earlier old-style root CA.

```
ceph orch certmgr cert ls
```

- If the root CA `commonName` has the `cephadm-root` format, it is in the old style. Continue with the procedure documented here.
- If the root CA `commonName` has the `cephadm-root-`*FSID* format, your cluster has already been upgraded to the new format and no further actions are required.

Use this procedure in any of the following scenarios:

- You are upgrading from Red Hat Ceph Storage versions 8.0 or earlier and are planning on using `cephadm` self-signed certificates.

– `cephadm` self-signed certificates are expired.

If you do not plan on using `cephadm` self-signed certificates or the root CA `commonName` has the `cephadm-root-FSID` format and the certificates are valid, this procedure is not required.

1. Remove the previous root CA.

   ```
   ceph orch certmgr rm cephadm_root_ca_cert
   ```

2. Reload the `certmgr`, to force creating the new root CA.

   ```
   ceph orch certmgr reload
   ```

3. Verify that the `certmgr` has been loaded correctly.

   ```
   ceph orch certmgr cert ls
   ```

   Ensure that the `cephadm_root_ca_cert` subject has the following format: `cephadm-root-FSID`
   The root CA is upgraded.

4. Reissue new certificates for all services that use `cephadm` self-signed certificates.
5. Refresh any existing Grafana services.
   a. Remove all Grafana certificates.
   This step must be done for each Grafana service that uses self-signed certificates.

      ```
      ceph orch certmgr cert rm grafana_cert --hostname=HOSTNAME
      ```

      Only continue once all certificates are removed.

   b. Redeploy Grafana.

      ```
      ceph orch redeploy grafana
      ```

   c. Verify that all Grafana services are running correctly.

      ```
      ceph orch ps --service-name grafana
      ```

   d. Verify that new `cephadm`-signed certificates have been created for Grafana.

      ```
      ceph orch certmgr cert ls
      ```

6. Refresh any existing Ceph Object Gateway (RGW) services.
   This step must be done for each Ceph Object Gateway service that uses self-signed certificates.

   ```
   ceph orch certmgr cert rm rgw_frontend_ssl_cert --service-name RGW-SERVICE-NAME
   ceph orch redeploy RGW-SERVICE-NAME
   ```

7. Verify that the Ceph Object Gateway (RGW) service is deployed and running as expected.

   ```
   ceph orch ps --service-name RGW_SERVICE_NAME
   ```

8. Verify the certificates.

   ```
   ceph orch certmgr cert ls
   ```

9. Upgrade the Ceph Dashboard certificates.
   a. Generate the `cephadm`-signed certificates for the dashboard.
   From the `cephadm` command-line interface (CLI), run the following commands:

      ```
      json="$(ceph orch certmgr generate-certificates dashboard)"
      printf '%s' "$json" | jq -r '.cert' > dashboard.crt
      printf '%s' "$json" | jq -r '.key'  > dashboard.key
      ```

   b. Set the new generated certificate and key.

From the Ceph command-line interface (CLI), run the following commands:

```
ceph dashboard set-ssl-certificate -i dashboard.crt
ceph dashboard set-ssl-certificate-key -i dashboard.key
```

The `dashboard.crt` and `dashboard.key` pair files are generated.

AFTER COMPLETING THE TASK

Enable and then disable the Ceph Dashboard to load the new certificates.

```
ceph mgr module disable dashboard
ceph mgr module enable dashboard
```

Checking whether OSDs are safe to upgrade

Determines whether Object Storage Daemons (OSDs) can be safely upgraded using the `ceph osd ok-to-upgrade` command. This command is primarily used by the Ceph Orchestrator to automate upgrade batching.

The `ceph osd ok-to-upgrade` command replaces the older `ok-to-stop` command, improving upgrade efficiency and reducing upgrade times.

The `mgr_osd_upgrade_check_convergence_factor` configuration option lets you tune the balance between response time and the size of the safe set. Adjust this value based on observed command performance.

How the command works

Before upgrading Object Storage Daemons (OSDs), you can use the `ceph osd ok-to-upgrade` command to determine how many OSDs can be upgraded at the same time without impacting access to data.

The command checks the current cluster state and identifies a set of OSDs within a specified CRUSH bucket that can be upgraded simultaneously. When a safe set is found, all data remains readable and writeable. However, data redundancy might be temporarily reduced and some placement groups (PGs) might appear as degraded but remain active.

This behavior is used when scoping upgrades to a CRUSH failure domain, such as a rack, to determine which OSDs in that domain can be upgraded in parallel. For more information, see “Upgrading OSDs by rack” on page 11.

Only OSDs that still require an upgrade to the specified Ceph version are considered. If a safe set cannot be determined, the command returns an error message explaining the reason.

> **Note:** Use this command as part of upgrade planning. Ensure that the cluster is healthy before upgrading OSDs.

During orchestrated cluster upgrades, the Ceph Orchestrator uses this command to determine how many OSDs can be upgraded concurrently. For more information, see “Upgrading the Red Hat Ceph Storage cluster” on page 2.

Usage

Use the following syntax and parameters to use the **`ceph osd ok-to-upgrade`** command.

```
ceph osd ok-to-upgrade CRUSH_BUCKET_NAME NEW_CEPH_VERSION_SHORT [--max NUM]
```

`CRUSH_BUCKET_NAME`

The CRUSH bucket that contains the OSDs to evaluate. Supported bucket types are `rack`, `chassis`, `host`, and `osd`.

`NEW_CEPH_VERSION_SHORT`

The target Ceph version in short format, for example `20.3.0-3803-g63ca1ffb5a2`, `20.1.0-144.el9cp`. This format matches the `NEW_CEPH_VERSION_SHORT` value shown by the `ceph osd metadata id` command.

`--max NUM`

Optional. Limits the number of OSDs returned in the safe set to the specified value. Use this option to upgrade OSDs in smaller batches.

Example

```
# ceph osd ok-to-upgrade my_crush_bucket my_ceph_version
{"ok_to_upgrade":true,"all_osds_upgraded":false,"osds_in_crush_bucket":
[9,21,2,15,17,10,4,22],"osds_ok_to_upgrade":[9,21,2,15,17,10,4,22],"osds_upgraded":
[],"bad_no_version":[]}
```

The command returns a JSON object with the following fields.

`ok_to_upgrade`

Indicates whether one or more OSDs in the specified CRUSH bucket can be safely upgraded to the target Ceph version. Values are `true` or `false`.

`osds_in_crush_bucket`

Lists all OSD IDs that belong to the specified CRUSH bucket.

`osds_ok_to_upgrade`

Lists the OSD IDs within the specified CRUSH bucket that are currently eligible for upgrade to the target Ceph version.

`osds_upgraded`

Lists the OSD IDs in the specified CRUSH bucket that have already been upgraded to the target Ceph version.

`bad_no_version`

Lists OSD IDs for which the current Ceph version could not be determined.

Upgrading OSDs by rack

Describes how to upgrade Object Storage Daemons (OSDs) within a rack by scoping upgrades to a CRUSH failure domain.

You can upgrade Object Storage Daemons (OSDs) by rack by limiting the upgrade to a CRUSH failure domain. This method upgrades multiple hosts in the same rack in parallel while maintaining cluster availability.

> **Important:** Use rack-based upgrades only in clusters that are configured with CRUSH rules that distribute replicas or erasure-coded chunks across racks. Using this method in other configurations can lead to pool unavailability.
> This method is supported only for multi-rack deployments. Do not use it in single-rack clusters.

How rack-based upgrades work

When you specify a CRUSH bucket type and name in the `ceph orch upgrade start` command, the Ceph Orchestrator limits the upgrade scope to the selected failure domain.

The related CRUSH bucket types include `rack`, `chassis`, and `host`.

For more information, see “Upgrading the Red Hat Ceph Storage cluster” on page 2.

For OSD upgrades, the Orchestrator uses the `ceph osd ok-to-upgrade` command to determine a safe set of OSDs within the selected rack that can be upgraded in parallel without affecting data availability. For more information, see “Checking whether OSDs are safe to upgrade” on page 10.

During the upgrade, OSDs in the selected rack are upgraded in batches. Data remains accessible, although data redundancy might be temporarily reduced and some placement groups (PGs) might appear degraded but remain active.

Before you begin

Before you begin, make sure that you have the following prerequisites in place:

- The cluster must have a multi-rack architecture.
- CRUSH rules must be configured to distribute data across racks.

- The cluster must be in a HEALTH_OK state before starting the upgrade.
- Network bandwidth must be sufficient to handle parallel OSD upgrades.

Usage

Use the **--crush_bucket_type** and **--crush_bucket_name** options with the `ceph orch upgrade start` command to target a specific CRUSH failure domain.

```
ceph orch upgrade start --image IMAGE_NAME --daemon-types osd --crush_bucket_type TYPE --
crush_bucket_name NAME
```

Where *TYPE* can be `rack`, `chassis`, or `host`.

Example for upgrading OSDs by rack

The following example upgrades all OSDs in rack `rackA` in controlled batches.

```
[ceph: root@host01 /]# ceph orch upgrade start --image registry.redhat.io/rhceph/
rhceph-9-rhel9:latest --daemon-types osd --crush_bucket_type rack --crush_bucket_name
rackA
```

Example for upgrading OSDs by host

The following example upgrades all OSDs on a single host `saraut-fullrack-node13`.

```
[ceph: root@host01 /]# ceph orch upgrade start registry.redhat.io/rhceph/
rhceph-9-rhel9:latest --daemon_types=osd --crush_bucket_type=host --
crush_bucket_name=saraut-fullrack-node13
```

Example for upgrading OSDs by chassis

The following example upgrades all OSDs within chassis `chassis-2`.

```
[cephcp.icr.io/cp/ibm-ceph/ceph-9-rhel9  --daemon-types osd --crush_bucket_type=chassis
--crush_bucket_name=chassis-2
```

```
[ceph: root@host01 /]# ceph orch upgrade start cp.icr.io/cp/ibm-ceph/ceph-9-rhel9 --
daemon-types osd --crush_bucket_type=chassis --crush_bucket_name=chassis-2
```

Upgrading a host operating system from RHEL 9 to RHEL 10

You can perform an Red Hat Ceph Storage host operating system upgrade from Red Hat Enterprise Linux (RHEL) 9 to Red Hat Enterprise Linux 10 using the Leapp utility.

PREREQUISITE

Be sure that you have a running Red Hat Ceph Storage cluster that is supported for upgrade. For more information, see“Supported upgrade paths” on page 1.

> **Important:** This host operating system upgrade *must* be performed before upgrading the Red Hat Ceph Storage cluster.

The following are the supported combinations of containerized Ceph daemons. For more information, see “Colocation” on page 0.

- Ceph Metadata Server (`ceph-mds`), Ceph OSD (`ceph-osd`), and Ceph Object Gateway (`radosgw`)
- Ceph Monitor (`ceph-mon`) *or* Ceph Manager (`ceph-mgr`), Ceph OSD (`ceph-osd`), and Ceph Object Gateway (`radosgw`)
- Ceph Monitor (`ceph-mon`), Ceph Manager (`ceph-mgr`), Ceph OSD (`ceph-osd`), and Ceph Object Gateway (`radosgw`)

1. Deploy your existingRed Hat Ceph Storage on the latest Red Hat Enterprise Linux 9 supported version with service.

> **Note:** Verify that the cluster contains two admin nodes, so that while performing host upgrade in one admin node (with `_admin` label), the second admin node can be used for managing clusters.

For full instructions, see Initial installation and "Deploying the Ceph daemons using the service specification" on page 0.

2. Set the `noout` flag on the Ceph OSD.
   For example,

   ```
   [ceph: root@host01 /]# ceph osd set noout
   ```

3. Run the host upgrade one node at a time using the Leapp utility.
   a. Put the respective node in maintenance mode before performing host upgrade using Leapp.

      ```
      ceph orch host maintenance enter HOST
      ```

      For example,

      ```
      ceph orch host maintenance enter host01
      ```

   b. Enable the Ceph tools repo manually when executing the Leapp command with the `--enablerepo` parameter.
      For example,

      ```
      leapp upgrade --enablerepo rhceph-9-tools-for-rhel-10-x86_64-rpms
      ```

   c. Upgrade Red Hat Enterprise Linux from version 9 to 10.
      For full instructions, follow the instructions provided within the Upgrading from RHEL 9 to RHEL 10 guide within the Red Hat Enterprise Linux product documentation on the Red Hat Customer Portal.

      > **Important:** After performing an in-place upgrade from Red Hat Enterprise Linux 9 to Red Hat Enterprise Linux 10, manually enable and start the `logrotate.timer` service.
      >
      > ```
      > systemctl start logrotate.timer
      > systemctl enable logrotate.timer
      > ```

4. Verify the new *IMAGE_ID* and *VERSION* of the Ceph cluster.

   ```
   [ceph: root@node0 /]# ceph version
   [ceph: root@node0 /]# ceph orch ps
   ```

AFTER COMPLETING THE TASK

Continue with the existing Red Hat Ceph Storage version to Red Hat Ceph Storage 9 upgrade, by following "Upgrading a Red Hat Ceph Storage cluster using cephadm" on page 1.

Upgrading to 9 with RHEL 10 with stretch mode enabled from previous versions

You can perform an upgrade from Red Hat Ceph Storage 7.1.0 or 8.1.0 to Red Hat Ceph Storage 9 involving Red Hat Enterprise Linux 9 to Red Hat Enterprise Linux 10 with the stretch mode enabled.

PREREQUISITE

Before you begin, make sure that you have the following prerequisites in place:

- Red Hat Ceph Storage on Red Hat Enterprise Linux 9 with necessary hosts and daemons that run with stretch mode enabled.

- Backup of Ceph binary (`/usr/sbin/cephadm`), `ceph.pub` (`/etc/ceph`), and the Ceph cluster's public SSH keys from the admin node.

> **Important:** Upgrade to the latest version of Red Hat Ceph Storage 7.1.0 or 8.1.0 before upgrading to the latest version of Red Hat Ceph Storage 9.

1. Log in to the `cephadm` shell.

   ```
   cephadm shell
   ```

2. Label a second node as the admin in the cluster to manage the cluster when the admin node is reprovisioned.

   ```
   ceph orch host label add HOSTNAME_admin
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph orch host label add host02_admin
   ```

3. Set the `noout` flag.

   ```
   ceph osd set noout
   ```

4. Drain all daemons from the host.

   ```
   ceph orch host drain HOSTNAME --force
   ```

   The `_no_schedule` label is automatically applied to the host that blocks deployment.

5. Verify that all daemons are removed from the storage cluster.

   ```
   ceph orch ps HOSTNAME
   ```

6. Zap the devices.
   Zapping the devices helps ensure that the hosts being drained have OSDs present. After the hosts are drained, they can be used to redeploy OSDs when the host is added back.

   ```
   ceph orch device zap HOSTNAME DISK --force
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph orch device zap ceph-host02 /dev/vdb --force
   zap successful for /dev/vdb on ceph-host02
   ```

7. Verify the status of the OSD removal.

   ```
   ceph orch host rm HOSTNAME --force
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph orch host rm host02 --force
   ```

8. Reprovision the respective hosts from RHEL 9 to RHEL 10.
   Use the instructions, as detailed in *Upgrading from RHEL 9 to RHEL 10*, on Red Hat Documentation.

9. Go to the `/usr/share/cephadm-ansible/` directory.

   ```
   cd /usr/share/cephadm-ansible
   ```

10. Run the preflight playbook with the `--limit` option.

    ```
    ansible-playbook -i INVENTORY_FILE cephadm-preflight.yml --limit NEW_HOSTNAME
    ```

For example,

```
[ceph: root@host01 cephadm ansible]# ansible-playbook -i hosts cephadm-preflight.yml
--extra-vars "ceph_origin={storage-product}" --limit host02
```

The preflight playbook installs `podman`, `lvm2`, `chrony`, and `cephadm` on the new host. After installation is complete, `cephadm` is found in the `/usr/sbin/` directory.

11. Extract the cluster's public SSH keys to a folder.

```
ceph cephadm get-pub-key ~/PATH
```

For example,

```
[ceph: root@host01 /]# ceph cephadm get-pub-key ~/ceph.pub
```

12. Copy the Ceph cluster public SSH keys to the reprovisioned node.

```
ssh-copy-id -f -i ~/PATH root@HOSTNAME_2
```

For example,

```
[ceph: root@host01 /]# ssh-copy-id -f -i ~/ceph.pub root@host02
```

13. Optional: If the removed host has a monitor daemon, before adding the host to the cluster, add the `--unmanaged` flag to monitor deployment.

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

14. Re-add the host to the cluster and re-add the labels that were present earlier.

```
ceph orch host add HOSTNAME IP_ADDRESS --labels=LABELS
```

The following examples give different ways to re-add the host to the cluster.

- ```
  ceph mon add HOSTNAME IP_LOCATION
  ```

  For example,

  ```
  [ceph: root@host01 /]# ceph mon add ceph-host02 10.0.211.62 datacenter=DC2
  ```

- ```
  ceph orch daemon add mon HOSTNAME
  ```

  For example,

  ```
  [ceph: root@host01 /]# ceph orch daemon add mon ceph-host02
  ```

15. Verify the daemons on the reprovisioned host running successfully with the same Ceph version.

```
ceph orch ps
```

16. Set the monitor daemon placement to `managed`.

```
ceph orch apply mon managed
```

17. Repeat all previous steps for each host.
18. Reprovision the arbiter monitor.
    The arbiter monitor cannot be drained or removed from the host. As a result, the arbiter monitor must be reprovisioned to another tie-breaker node, and then drained or removed from host. For more information, see Replacing a tiebreaker with a new monitor.
19. Reprovision the administrator nodes.
    Use a second administrator node to manage clusters. For more information, see Replacing a tiebreaker with a new monitor.

20. Re-add the backup files to the node.
21. Re-add the administrator noes to the cluster, by using the second administrator node.
    Set the `mon` deployment to `unmanaged`.

    ```
    ceph orch apply mon unmanaged
    ```

22. Re-add the original arbiter monitor and remove the temporary monitor that was created in step "18" on page 15.
    For more information, see Replacing a tiebreaker with a new monitor.
23. Unset the `noout` flag.

    ```
    ceph osd unset noout
    ```

AFTER COMPLETING THE TASK

1. Verify the Ceph version and the cluster status to help ensure that all demons are working as expected after the OS upgrade.
2. Follow the "Upgrading a Red Hat Ceph Storage cluster using cephadm" on page 1 to upgrade from Red Hat Ceph Storage 7.1.0 or 8.1.0 to 9.

Running a staggered upgrade

As a storage administrator, you can upgrade Red Hat Ceph Storage components in phases rather than all at once. You can use the **`ceph orch upgrade`** options to limit which daemons are upgraded with a single upgrade command.

PREREQUISITE

Before you begin, make sure that you have the following prerequisites in place:

- The latest version of a supported Red Hat Ceph Storage cluster. For a full list of supported upgrade paths, see "Supported upgrade paths" on page 1.
- Root-level access to all the nodes.
- At least two Ceph Manager nodes in the storage cluster: one active and one standby.
- Topological labels set on a host. Use topological labels to upgrade specific hosts. For more information, see Adding topological labels to a host.

> **Note:** If you want to upgrade from a version that does not support staggered upgrades, you must first manually upgrade the Ceph Manager (`ceph-mgr`) daemons.

The **`ceph orch upgrade`** command supports several options to upgrade cluster components in phases. "Staggered upgrade options" on page 16 lists the staggered upgrade options.

Staggered upgrade options	
Command option	**Description**
`--daemon_types`	The **`--daemon_types`** option takes a comma-separated list of daemon types and only upgrades daemons of those types. Valid daemon types for this option include `mgr`, `mon`, `crash`, `osd`, `mds`, `rgw`, `rbd-mirror`, and `cephfs-mirror`.
`--crush_bucket_type`	The **`--crush_bucket_type`** option specifies the CRUSH bucket type for scoping OSD upgrades. You must use this option with **`--daemon_types=osd`** and **`--crush_bucket_name`** to limit upgrades to a specific CRUSH failure domain, such as rack, chassis, or host.

Command option	Description
--crush_bucket_name	The **--crush_bucket_name** option specifies the name of the CRUSH bucket to upgrade. You must use this option with**--crush_bucket_type** and**--daemon_types=osd**. For example, `--crush_bucket_type rack --crush_bucket_name rack01` upgrades only the OSDs in rack01.
--services	The **--services** option is mutually exclusive with **--daemon_types**. **--services** takes only services of one type at a time, and upgrades only daemons belonging to those services. For example, you cannot provide an OSD and RGW service simultaneously.
--hosts	You can combine the **--hosts** option with**--daemon_types**,**--services**, or use it on its own. The **--hosts** option parameter follows the same format as the command line options for orchestrator CLI placement specification.
--limit	The **--limit** option takes an integer greater than zero and provides a numerical limit on the number of daemons `cephadm` upgrades. You can combine the **--limit** option with **--daemon_types**, **--services**, or **--hosts**. For example, if you specify to upgrade daemons of type `osd` on `host01` with a limit set to `3`, `cephadm` upgrades up to three OSD daemons on host01.
--topological-labels	The **--topological-labels** option upgrades a specific set of hosts that have the specified topological label. For example, if the label `datacenter=A` is used on a set of hosts, you can target these specific hosts for upgrade, without needing to manually input each host.

`cephadm` strictly enforces an order for the upgrade of daemons that is still present in staggered upgrade scenarios. The current upgrade order is as follows:

1. Ceph Manager (`mgr`) nodes
2. Ceph Monitor (`mon`) nodes
3. Ceph `crash` daemons
4. Ceph `osd` nodes
5. Ceph Metadata Server (`mds`) nodes
6. Ceph Object Gateway (`rgw`) nodes
7. Ceph Block Device mirror (`rbd-mirror`) node
8. Ceph File System mirror (`cephfs-mirror`) node

When you specify CRUSH bucket parameters for OSD upgrades, the Orchestrator determines OSD readiness using `ok-to-upgrade` safety checks to identify OSDs that can be upgraded in parallel without downtime. If you do not specify CRUSH bucket parameters, the upgrade behaves as before and uses `ok-to-stop` checks. This enables phased upgrades aligned with physical infrastructure boundaries and provides better OSD upgrade performance compared to upgrades without bucket parameters.

Note:

- If you specify parameters that upgrade daemons out of order, the upgrade command blocks and notes which daemons you need to upgrade before you proceed.

- Restarting instances does not require a specific order. It is recommended to restart the instance pointing to the pool with primary images followed by the instance pointing to the mirrored pool.

The following example shows a staggered upgrade that was started out of order:

```
[ceph: root@host01 /]# ceph orch upgrade start --image registry.redhat.io/rhceph-9/
rhceph-9-rhel9:latest --hosts host02

Error EINVAL: Cannot start upgrade. Daemons with types earlier in upgrade order than
daemons on given host need upgrading.
Please first upgrade mon.ceph-host01
NOTE: Enforced upgrade order is: mgr -> mon -> crash -> osd -> mds -> rgw -> rbd-
mirror -> cephfs-mirror
```

1. Log in to the `cephadm` shell and check that all hosts are online and that the storage cluster is in a healthy state.
   For more information about starting the `cephadm` shell, see “Using the cephadm shell” on page 0.
   For example,

   ```
   [root@host01 ~]# cephadm shell
   [ceph: root@host01 /]# ceph -s
   ```

2. Set the OSD `noout`, `noscrub`, and `nodeep-scrub` flags to prevent OSDs from getting marked out during upgrade and to avoid unnecessary load on the cluster.

   ```
   ceph osd set noout
   ```

   ```
   ceph osd set noscrub
   ```

   ```
   ceph osd set nodeep-scrub
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph osd set noout
    [ceph: root@host01 /]# ceph osd set noscrub
    [ceph: root@host01 /]# ceph osd set nodeep-scrub
   ```

3. Check service versions and the available target containers.

   ```
   ceph orch upgrade check IMAGE_NAME
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph orch upgrade check registry.redhat.io/rhceph-9/
   rhceph-9-rhel9:latest
   ```

4. Upgrade the storage cluster.
   Upgrade clusters in one of the following ways:
   - Upgrade specific daemon types on specific hosts.

     ```
     ceph orch upgrade start --image IMAGE_NAME --daemon-types
     DAEMON_TYPE1,DAEMON_TYPE2 --hosts HOST1,HOST2
     ```

     For example,

     ```
     [ceph: root@host01 /]# ceph orch upgrade start --image registry.redhat.io/
     rhceph-9/rhceph-9-rhel9:latest --daemon-types mgr,mon --hosts host02,host03
     ```

   - Upgrade OSD daemons across the cluster. You can scope the upgrade to a specific CRUSH failure domain, such as a rack, chassis, or host, by using CRUSH bucket parameters with **`--daemon-types=osd`**.

```
ceph orch upgrade start --image IMAGE_NAME --daemon-types osd --crush_bucket_type
BUCKET_TYPE --crush_bucket_name BUCKET_NAME
```

For example,

```
[ceph: root@host01 /]# ceph orch upgrade start --image registry.redhat.io/
rhceph-9/rhceph-9-rhel9:latest --daemon-types osd --crush_bucket_type rack --
crush_bucket_name rack01
```

To upgrade all OSDs within a rack as a failure domain, see "Upgrading OSDs by rack" on page 11.

When you specify CRUSH bucket parameters:

- CRUSH bucket parameters apply only to OSD upgrades and require **`--daemon-types=osd`**.
- You cannot combine bucket parameters with `--services`, `--hosts`, or `--topological-labels` filters.
- The upgrade honors the **`max_parallel_osd_upgrades`** setting within the specified bucket.

  Note: The **`max_parallel_osd_upgrades`** parameter sets the maximum number of OSDs that can be upgraded in parallel. The default value is 16.
  For example,

  ```
  [ceph: root@host01 /]# ceph config get mgr mgr/cephadm/
  max_parallel_osd_upgrades
  16
  [ceph: root@host01 /]# ceph config set mgr mgr/cephadm/
  max_parallel_osd_upgrades 32
  ```

- If you do not specify bucket parameters, the upgrade behaves as before.

Note: Do not change the CRUSH hierarchy while OSD upgrades are ongoing using CRUSH bucket parameters.

– Specify specific services and limit the number of daemons to upgrade.

Note:

- In staggered upgrade scenarios, if using a limiting parameter, the monitoring stack daemons, including Prometheus and `node-exporter`, are refreshed after the upgrade of the Ceph Manager daemons. As a result of the limiting parameter, Ceph Manager upgrades take longer to complete. The versions of monitoring stack daemons might not change between Ceph releases, in which case, they are only redeployed.
- Upgrade commands with limiting parameters validates the options before beginning the upgrade, which can require pulling the new container image. As a result, the **`upgrade start`** command might take a while to return when you provide limiting parameters.

```
ceph orch upgrade start --image IMAGE_NAME --services SERVICE1,SERVICE2 --limit
LIMIT_NUMBER
```

For example,

```
[ceph: root@host01 /]# ceph orch upgrade start --image registry.redhat.io/
rhceph-9/rhceph-9-rhel9:latest --services rgw.example1,rgw1.example2 --limit 2
```

– Specify specific topological labels to run the upgrade on specific hosts.

> **Note:** Using topological labels does not allow changing from the required daemon upgrade order.

```
ceph orch upgrade start --topological-labels TOPOLOGICAL_LABEL
```

For example,

```
[ceph: root@host01 /]# ceph orch upgrade start --topological-labels datacenter=A
```

- Specify specific topological labels to run the upgrade on specific hosts, with the **`--daemon-types`** option.
  Use the **`--daemon-types`** option for hosts with a similar scenario to the following example:

  A host exists with both Monitor and OSD daemons that match `datacenter=A` but nonupgraded Monitor daemons on host that do not match `datacenter=A`. In this case, you only want to upgrade the Monitor daemons on the `datacenter=A` host, to not break the upgrade order. In this scenario, use the following command to upgrade only the Monitor daemons on `datacenter=A`, but not the OSD daemons.

  ```
  ceph orch upgrade start IMAGE_NAME --daemon-types DAEMON_TYPE1 --topological-
  labels TOPOLOGICAL_LABEL
  ```

  For example,

  ```
  [ceph: root@host01 /]# ceph orch upgrade start registry.redhat.io/rhceph-9/
  rhceph-9-rhel9:latest --daemon-types mon --topological-labels datacenter=A
  ```

  Use the **`ceph orch upgrade status`** command to verify what was selected for upgrade. View the `"which"` field in the output. For example,

  ```
  [ceph: root@host01 /]# ceph orch upgrade status
  {
      "in_progress": true,
      "target_image": "registry.redhat.io/rhceph-9/rhceph-9-rhel9:latest",
      "services_complete": [
          "mon"
      ],
      "which": "Upgrading daemons of type(s) mon on host(s) host01",
      "progress": "1/1 daemons upgraded",
      "message": "Currently upgrading mon daemons",
      "is_paused": false
  }
  ```

5. Check which daemons still need to upgrade by running either the **`ceph orch upgrade check`** or **`ceph versions`** command.

   ```
   ceph orch upgrade check --image registry.redhat.io/rhceph-9/rhceph-9-rhel9:latest
   ```

6. Verify the upgrade of all remaining services.

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

   For example,

   ```
   [ceph: root@host01 /]# ceph orch upgrade start --image registry.redhat.io/rhceph-9/
   rhceph-9-rhel9:latest
   ```

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

   ```
   ceph versions
   ceph orch ps
   ```

AFTER COMPLETING THE TASK

After the upgrade is complete, unset the upgrade is complete, unset the `noout`, `noscrub`, and `nodeep-scrub` flags.

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

Monitoring and managing upgrade

You can monitor and manage the upgrade of the storage cluster during the upgrade process.

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. The health of the cluster changes to HEALTH_WARNING during an upgrade.

> **Note:** Upgrade one daemon type after the other.

The upgrade is paused in the following situations:

- The host of the cluster is offline.
- A daemon cannot be upgraded.
- An invalid CRUSH bucket name or type is specified during an OSD upgrade.

> **Note:** If a cluster is or becomes unavailable the upgrade will be paused until it is restored.

Before you begin, make sure that you have the following prerequisites in place:

- A running Red Hat Ceph Storage supported existing.
- 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.

Use the **ceph orch upgrade** command to take you through the upgrade process. "Upgrade management command options" on page 21 provides the different options that can be used to help manage and monitor during the upgrade process.

Upgrade management command options

Command option	Syntax	Notes
status	`ceph orch upgrade status`	– Use the status option to determine whether an upgrade is in progress and the version to which the cluster is upgrading. – You do not get a message when 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.
pause	`ceph orch upgrade pause`	None
resume	`ceph orch upgrade resume`	None

Command option	Syntax	Notes
`stop`	`ceph orch upgrade stop`	None