Red Hat Ceph Storage Troubleshooting

Troubleshoot and resolve common problems with Red Hat Ceph Storage.

Initial troubleshooting

As a storage administrator, you can do the initial troubleshooting of a Red Hat Ceph Storage cluster before contacting Red Hat Support.

Initial troubleshooting requires a running Red Hat Ceph Storage cluster.

For more information about contacting Red Hat Support, see .

Identifying problems

Identify any possible causes of errors within the Red Hat Ceph Storage cluster.

PREREQUISITE

- A running Red Hat Ceph Storage cluster.

Determine possible causes of the error with the Red Hat Ceph Storage cluster, by answering the following questions.

1. Certain problems can arise when using unsupported configurations.
   Ensure that your configuration is supported. For more information, see Compatibility matrix.
2. Do you know which Ceph component caused the problem?
   - No.
     See “Diagnosing health” on page 1.
   - Ceph Monitors.
     See “Troubleshooting Ceph Monitors” on page 21.
   - Ceph OSDs.
     See “Troubleshooting Ceph OSDs” on page 34.
   - Ceph placement groups.
     See “Troubleshooting Ceph placement groups” on page 52.
   - Multi-site Ceph Object Gateway.
     See “Troubleshooting multi-site Ceph Object Gateway” on page 48.

AFTER COMPLETING THE TASK

For more information, see Red Hat Ceph Storage: Supported configurations, on Red Hat Customer Portal.

Diagnosing health

Learn how to run a basic diagnostic of the health of an Red Hat Ceph Storage cluster.

PREREQUISITE

- A running Red Hat Ceph Storage cluster.

1. Log in to the `cephadm` shell.

```
cephadm shell
```

2. Check the overall status of the storage cluster.
   For example,

```
[ceph: root@host01 /]# ceph health detail
```

If the command returns a HEALTH_WARN or HEALTH_ERR message, see “Understanding Ceph health” on page 2.

3. Monitor the logs of the storage cluster.
   For example,

   ```
   [ceph: root@host01 /]# ceph -w cephadm
   ```

4. Run the following commands to capture the logs of the cluster to a file.

   ```
   ceph config set global log_to_file true
   ```

   ```
   ceph config set global mon_cluster_log_to_file true
   ```

   By default, the logs are located in the `/var/log/ceph/` directory.

5. Check the Ceph logs for any error messages that are listed in “Understanding Ceph logs” on page 4.
6. If the logs do not include enough information, increase the debugging level and try to reproduce the action that failed.
   For more information, see “Configuring logging at runtime” on page 7.

Understanding Ceph health

Learn about the different Ceph health messages.

The **`ceph health`** command returns information about the status of the Red Hat Ceph Storage cluster.

HEALTH_OK

Indicates that the cluster is healthy.

HEALTH_WARN

Indicates a warning. Sometimes the Ceph status returns to HEALTH_OK automatically such as when Red Hat Ceph Storage cluster finishes the rebalancing process.
Consider further troubleshooting if a cluster is in the HEALTH_WARN state for an extended time.

HEALTH_ERR

Indicates a more serious problem that requires your immediate attention.

Use the `ceph health detail` and `ceph -s` commands to get a more detailed output.

> **Note:** A health warning is displayed if there is no `mgr` daemon running. In case the last `mgr` daemon of an Red Hat Ceph Storage cluster was removed, you can manually deploy a `mgr` daemon on a random host of the Red Hat Ceph Storage cluster.
> For more information, see Manually deploying a mgr daemon.

For more information, see “Ceph Monitor error messages” on page 21, “Ceph OSD error messages” on page 34, and “Placement group error messages” on page 53.

Muting health alerts

In certain scenarios, users might want to temporarily mute some warnings because they are already aware of the warning and cannot act on it immediately. You can mute health checks so that they do not affect the overall reported status of the Ceph cluster.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Root-level of access to the nodes.
- A health warning message.

Alerts are specified by using the health check codes. One example is, when an OSD is brought down for maintenance, OSD_DOWN warnings are expected. You can choose to mute the warning until the maintenance is over because those warnings put the cluster in HEALTH_WARN instead of HEALTH_OK for the entire duration of maintenance.

Most health mutes also disappear if the extent of an alert gets worse. For example, if there is one OSD down, and the alert is muted, the mute disappears if one or more additional OSDs go down. This is true for any health alert that involves a count indicating how much or how many of something is triggering the warning or error.

For more information, see "Health messages of a Ceph cluster" on page 79.

1. Log in to the `cephadm` shell.

   ```
   cephadm shell
   ```

2. Check the health of the Red Hat Ceph Storage cluster by running the **`ceph health detail`** command. For example,

   ```
   [ceph: root@host01 /]# ceph health detail

    HEALTH_WARN 1 osds down; 1 OSDs or CRUSH {nodes, device-classes} have
   {NOUP,NODOWN,NOIN,NOOUT} flags set
    [WRN] OSD_DOWN: 1 osds down
        osd.1 (root=default,host=host01) is down
    [WRN] OSD_FLAGS: 1 OSDs or CRUSH {nodes, device-classes} have
   {NOUP,NODOWN,NOIN,NOOUT} flags set
        osd.1 has flags noup
   ```

   In this example, the storage cluster is in *HEALTH_WARN* status, indicated as one of the OSDs is down.

3. Mute the alert.

   ```
   ceph health mute HEALTH_MESSAGE
   ```

   The following example mutes the OSD_DOWN message.

   ```
   [ceph: root@host01 /]# ceph health mute OSD_DOWN
   ```

   Note: A health check mute can have a time to live (TTL) associated with it, such that the mute automatically expires after the specified time is over. You can optionally specify the TTL as an optional duration argument in the command.

   ```
   ceph health mute HEALTH_MESSAGE DURATION
   ```

   DURATION can be specified in any of the following value types: `s`, `sec`, `m`, `min`, `h`, or `hour`.
   The following example mutes the OSD_DOWN message for 10 minutes:

   ```
   [ceph: root@host01 /]# ceph health mute OSD_DOWN 10m
   ```

4. Verify that the Red Hat Ceph Storage cluster status has changed to HEALTH_OK.
   The following example shows that the alerts OSD_DOWN and OSD_FLAG are muted and the mute is active for nine minutes.

```
[ceph: root@host01 /]# ceph -s
   cluster:
     id:     81a4597a-b711-11eb-8cb8-001a4a000740
     health: HEALTH_OK
             (muted: OSD_DOWN(9m) OSD_FLAGS(9m))

   services:
     mon: 3 daemons, quorum host01,host02,host03 (age 33h)
     mgr: host01.pzhfuh(active, since 33h), standbys: host02.wsnngf, host03.xwzphg
     osd: 11 osds: 10 up (since 4m), 11 in (since 5d)

   data:
     pools:   1 pools, 1 pgs
     objects: 13 objects, 0 B
     usage:   85 MiB used, 165 GiB / 165 GiB avail
     pgs:     1 active+clean
```

5. Optional: Retain the mute even after the alert is cleared by making it **sticky**.

```
ceph health mute HEALTH_MESSAGE DURATION --sticky
```

For example,

```
[ceph: root@host01 /]# ceph health mute OSD_DOWN 1h --sticky
```

AFTER COMPLETING THE TASK

Remove the mute by running the following command:

```
ceph health unmute HEALTH_MESSAGE
```

For example,

```
[ceph: root@host01 /]# ceph health unmute OSD_DOWN
```

Understanding Ceph logs

Ceph stores its logs in the `/var/log/ceph/` directory after the logging is enabled.

The `CLUSTER_NAME.log` is the main storage cluster log file that includes global events. By default, the log file name is `ceph.log`. Only the Ceph Monitor nodes include the main storage cluster log.

Each Ceph OSD and Monitor has its own log file named `CLUSTER_NAME-osd.NUMBER.log` and `CLUSTER_NAME-mon.HOSTNAME.log` respectively.

When you increase debugging level for Ceph subsystems, Ceph generates new log files for those subsystems as well.

For more information about logging, see “Configuring logging” on page 5.

For more information about error messages, see “Common Ceph Monitor error messages in Ceph logs” on page 21 and “Common Ceph OSD error messages in Ceph logs” on page 35.

Generating an `sos report`

Run the **`sos report`** command to collect the configuration details, system information, and diagnostic information of an Red Hat Ceph Storage cluster from a Red Hat Enterprise Linux system.

PREREQUISITE

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

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

Red Hat Support uses the logs that are generated for further troubleshooting of the storage cluster.

For more information, see the *What is an sos report and how to create one in Red Hat Enterprise Linux?* KnowledgeBase article.

1. Install the `sos` package.

For example,

```
[root@host01 ~]# dnf install sos
```

2. Run the **sos report** command to get the system information of the storage cluster.

```
sos report -a --all-logs
```

For example,

```
[root@host01 ~]# sos report -a --all-logs
```

The report is saved in the `/var/tmp` file.
Run the following command for specific Ceph daemon information.

```
sos report --all-logs -e CEPH_SERVICE1,CEPH_SERVICE2
```

For example,

```
[root@host01 ~]# sos report --all-logs -e
ceph_mgr,ceph_common,ceph_mon,ceph_osd,ceph_ansible,ceph_mds,ceph_rgw
```

Configuring logging

Learn how to configure logging for various Ceph subsystems.

> **Important:** Logging is resource-intensive. Also, verbose logging can generate a huge amount of data in a relatively short time. If you are encountering problems in a specific subsystem of the cluster, enable logging only of that subsystem. For more information, see "Ceph subsystems" on page 5.
> Consider also setting up a rotation of log files. For more information, see "Accelerating log rotation" on page 9.
>
> Once you fix any problems you encounter, change the subsystems log and memory levels to their default values. For a list of all Ceph subsystems and their default values, see "Ceph subsystems default logging level values" on page 77.

You can configure Ceph logging by:

- Using the **ceph** command at runtime. This is the most common approach. For more information, see "Configuring logging at runtime" on page 7.
- Updating the Ceph configuration file. Use this approach if you are encountering problems when starting the cluster. For more information, see "Configuring logging in configuration file" on page 8.

> **Note:** When configuring logging, be sure to have a running Red Hat Ceph Storage cluster.

Ceph subsystems

This section contains information about Ceph subsystems and their logging levels.

Understanding Ceph subsystems and their logging levels

Ceph consists of several subsystems.

Each subsystem has a logging level:

- Output logs that are stored by default in `/var/log/ceph/` directory (log level)
- Logs that are stored in a memory cache (memory level)

In general, Ceph does not send logs stored in memory to the output logs unless:

- A fatal signal is raised

- An assert in source code is triggered
- You request it

You can set different values for each of these subsystems. Ceph logging levels operate on a scale of `1` to `20`, where `1` is terse and `20` is verbose.

Use a single value for the log level and memory level to set them both to the same value. For example, `debug_osd = 5` sets the debug level for the `ceph-osd` daemon to `5`.

To use different values for the output log level and the memory level, separate the values with a forward slash (`/`). For example, `debug_mon = 1/5` sets the debug log level for the `ceph-mon` daemon to `1` and its memory log level to `5`.

Ceph Subsystems and the Logging Default Values

Default logging values of sub systems			
Subsystem	**Log Level**	**Memory Level**	**Description**
`asok`	1	5	The administration socket
`auth`	1	5	Authentication
`client`	0	5	Any application or library that uses `librados` to connect to the cluster.
`bluestore`	1	5	The BlueStore OSD backend.
`journal`	1	5	The OSD journal
`mds`	1	5	The Metadata Servers
`monc`	0	5	The Monitor client handles communication between most Ceph daemons and Monitors
`mon`	1	5	Monitors
`ms`	0	5	The messaging system between Ceph components
`osd`	0	5	The OSD Daemons
`paxos`	0	5	The algorithm that Monitors use to establish a consensus
`rados`	0	5	Reliable Autonomic Distributed Object Store, a core component of Ceph
`rbd`	0	5	The Ceph Block Devices
`rgw`	1	5	The Ceph Object Gateway

Example Log Outputs

The following examples show the type of messages in the logs when you increase the verbosity for the Monitors and OSDs.

Monitor Debug Settings

```
debug_ms = 5
debug_mon = 20
debug_paxos = 20
debug_auth = 20
```

Example Log Output of Monitor Debug Settings

```
2022-05-12 12:37:04.278761 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2
up, 2 in
2022-05-12 12:37:04.278792 7f45a9afc700 10 mon.cephn2@0(leader).osd e322
 min_last_epoch_clean 322
2022-05-12 12:37:04.278795 7f45a9afc700 10 mon.cephn2@0(leader).log v1010106 log
2022-05-12 12:37:04.278799 7f45a9afc700 10 mon.cephn2@0(leader).auth v2877 auth
2022-05-12 12:37:04.278811 7f45a9afc700 20 mon.cephn2@0(leader) e1 sync_trim_providers
2022-05-12 12:37:09.278914 7f45a9afc700 11 mon.cephn2@0(leader) e1 tick
2022-05-12 12:37:09.278949 7f45a9afc700 10 mon.cephn2@0(leader).pg v8126 v8126: 64 pgs: 64
active+clean; 60168 kB data, 172 MB used, 20285 MB / 20457 MB avail
2022-05-12 12:37:09.278975 7f45a9afc700 10 mon.cephn2@0(leader).paxosservice(pgmap
7511..8126) maybe_trim trim_to 7626 would only trim 115 < paxos_service_trim_min 250
2022-05-12 12:37:09.278982 7f45a9afc700 10 mon.cephn2@0(leader).osd e322 e322: 2 osds: 2
up, 2 in
2022-05-12 12:37:09.278989 7f45a9afc700  5 mon.cephn2@0(leader).paxos(paxos active c
1028850..1029466) is_readable = 1 - now=2021-08-12 12:37:09.278990 lease_expire=0.000000
has v0 lc 1029466
....
2022-05-12 12:59:18.769963 7f45a92fb700  1 -- 192.168.0.112:6789/0 <== osd.1
192.168.0.114:6800/2801 5724 ==== pg_stats(0 pgs tid 3045 v 0) v1 ==== 124+0+0 (2380105412
0 0) 0x5d96300 con 0x4d5bf40
2022-05-12 12:59:18.770053 7f45a92fb700  1 -- 192.168.0.112:6789/0 -->
192.168.0.114:6800/2801 -- pg_stats_ack(0 pgs tid 3045) v1 -- ?+0 0x550ae00 con 0x4d5bf40
2022-05-12 12:59:32.916397 7f45a9afc700  0 mon.cephn2@0(leader).data_health(1) update_stats
avail 53% total 1951 MB, used 780 MB, avail 1053 MB
....
2022-05-12 13:01:05.256263 7f45a92fb700  1 -- 192.168.0.112:6789/0 -->
192.168.0.113:6800/2410 -- mon_subscribe_ack(300s) v1 -- ?+0 0x4f283c0 con 0x4d5b440
```

Example log output of OSD debug settings

```
debug_ms = 5
debug_osd = 20
```

```
2022-05-12 11:27:53.869151 7f5d55d84700  1 -- 192.168.17.3:0/2410 -->
192.168.17.4:6801/2801 -- osd_ping(ping e322 stamp 2021-08-12 11:27:53.869147) v2 -- ?+0
0x63baa00 con 0x578dee0
2022-05-12 11:27:53.869214 7f5d55d84700  1 -- 192.168.17.3:0/2410 -->
192.168.0.114:6801/2801 -- osd_ping(ping e322 stamp 2021-08-12 11:27:53.869147) v2 -- ?+0
0x638f200 con 0x578e040
2022-05-12 11:27:53.870215 7f5d6359f700  1 -- 192.168.17.3:0/2410 <== osd.1
192.168.0.114:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp 2021-08-12
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x63c1a00 con 0x578e040
2022-05-12 11:27:53.870698 7f5d6359f700  1 -- 192.168.17.3:0/2410 <== osd.1
192.168.17.4:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp 2021-08-12
11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x6313200 con 0x578dee0
....
2022-05-12 11:28:10.432313 7f5d6e71f700  5 osd.0 322 tick
2022-05-12 11:28:10.432375 7f5d6e71f700 20 osd.0 322 scrub_random_backoff lost coin flip,
randomly backing off
2022-05-12 11:28:10.432381 7f5d6e71f700 10 osd.0 322 do_waiters -- start
2022-05-12 11:28:10.432383 7f5d6e71f700 10 osd.0 322 do_waiters -- finish
```

Reference

For more information, see “Configuring logging at runtime” on page 7 and “Configuring logging in configuration file” on page 8.

Configuring logging at runtime

Configure the logging of Ceph subsystems at system runtime to help troubleshoot any issues that might occur.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Access to Ceph debugger.

For more information, see:

- “Ceph subsystems” on page 5

- “Configuring logging in configuration file” on page 8
- Debugging and logging configuration options
- Activate the Ceph debugging output `dout()`, at runtime.

  ```
  ceph tell TYPE.ID injectargs --debug-SUBSYSTEM VALUE [--NAME VALUE]
  ```

  Replace as follows:

  - *TYPE* with the type of Ceph daemons (`osd`, `mon`, or `mds`).
  - *ID* with a specific ID of the Ceph daemon. Alternatively, use `*` to apply the runtime setting to all daemons of a particular type.
  - *SUBSYSTEM* with a specific subsystem.
  - *VALUE* with a number from `1` to `20`, where `1` is terse and `20` is verbose.

  The following example shows setting the log level for the OSD subsystem on the OSD name `osd.0` to 0 and the memory level to 5.

  ```
  # ceph tell osd.0 injectargs --debug-osd 0/5
  ```

Seeing the configuration settings at runtime

1. Log in to the host with a running Ceph daemon.
   For example, `ceph-osd` or `ceph-mon`.
2. Display the configuration.

   ```
   ceph daemon NAME config show | less
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph daemon osd.0 config show | less
   ```

Configuring logging in configuration file

Configure Ceph subsystems to log information, warning, and error messages to the log file.

You can specify the debugging level in the Ceph configuration file, which is found by default in `/etc/ceph/ceph.conf`.

For more information, see:

- “Ceph subsystems” on page 5
- “Configuring logging at runtime” on page 7
- Debugging and logging configuration options

  To activate the Ceph debugging output `dout()` at boot time, add the debugging settings to the Ceph configuration file.

1. For subsystems common to each daemon, add the settings under the `[global]` section.
2. For subsystems for particular daemons, add the settings under a daemon section, such as `[mon]`, `[osd]`, or `[mds]`.
   For example,

```
[global]
            debug_ms = 1/5

    [mon]
            debug_mon = 20
            debug_paxos = 1/5
            debug_auth = 2

    [osd]
            debug_osd = 1/5
            debug_monc = 5/20

    [mds]
            debug_mds = 1
```

Accelerating log rotation

Accelerate log rotation by modifying the Ceph log rotation file.

PREREQUISITE

- A running Red Hat Ceph Storage cluster.
- Root-level access to the node.

Increasing debugging level for Ceph components might generate a huge amount of data. If you have almost full disks, you can accelerate log rotation by modifying the Ceph log rotation file at `/etc/logrotate.d/ceph-<fsid>`. The Cron job scheduler uses this file to schedule log rotation.

1. Add the size setting after the rotation frequency to the log rotation file.

```
rotate 7
 weekly
 size SIZE
 compress
 sharedscripts
```

   The following is an example to rotate a log file when it reaches 500 MB.

```
rotate 7
 weekly
 size 500 MB
 compress
 sharedscripts
 size 500M
```

> **Note:** The *SIZE* value can be expressed as *500 MB* or *500M*.

2. Open the `crontab` editor.
   For example,

```
[root@mon ~]# crontab -e
```

3. Add an entry to check the `/etc/logrotate.d/ceph-<fsid>` file.
   The following example shows how to instruct Cron to check the `/etc/logrotate.d/ceph-<fsid>` file every 30 minutes.

```
30 * * * * /usr/sbin/logrotate /etc/logrotate.d/ceph-d3bb5396-
c404-11ee-9e65-002590fc2a2e >/dev/null 2>&1
```

Creating and collecting operation logs for Ceph Object Gateway

Track user identities reliably by S3 request in all versions of the Ceph Object Gateway operation log.

User identity information is added to the operation log output. This is used to enable customers to access this information for auditing of S3 access.

1. Find where the logs are located.

```
logrotate -f
```

For example,

```
[root@host01 ~]# logrotate -f
 /etc/logrotate.d/ceph-12ab345c-1a2b-11ed-b736-fa163e4f6220
```

2. List the logs within the specified location.

```
ll LOG_LOCATION
```

For example,

```
[root@host01 ~]# ll /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220
 -rw-r--r--. 1 ceph ceph    412 Sep 28 09:26 opslog.log.1.gz
```

3. Create a bucket.

```
/usr/local/bin/s3cmd mb s3://NEW_BUCKET_NAME
```

For example,

```
[root@host01 ~]# /usr/local/bin/s3cmd mb s3://bucket1
 Bucket `s3://bucket1` created
```

4. Collect the logs.

```
tail -f LOG_LOCATION/opslog.log
```

For example,

```
[root@host01 ~]# tail -f /var/log/ceph/12ab345c-1a2b-11ed-b736-fa163e4f6220/opslog.log

 {"bucket":"","time":"2022-09-29T06:17:03.133488Z","time_local":"2022-09-
 29T06:17:03.133488+0000","remote_addr":"10.0.211.66","user":"test1",
 "operation":"list_buckets","uri":"GET /
 HTTP/1.1","http_status":"200","error_code":"","bytes_sent":232,
 "bytes_received":0,"object_size":0,"total_time":9,"user_agent":"","referrer":
 "","trans_id":"tx00000c80881a9acd2952a-006335385f-175e5-primary",
 "authentication_type":"Local","access_key_id":"1234","temp_url":false}

 {"bucket":"cn1","time":"2022-09-29T06:17:10.521156Z","time_local":"2022-09-
 29T06:17:10.521156+0000","remote_addr":"10.0.211.66","user":"test1",
 "operation":"create_bucket","uri":"PUT /cn1/
 HTTP/1.1","http_status":"200","error_code":"","bytes_sent":0,
 "bytes_received":0,"object_size":0,"total_time":106,"user_agent":"",
 "referrer":"","trans_id":"tx0000058d60c593632c017-0063353866-175e5-primary",
 "authentication_type":"Local","access_key_id":"1234","temp_url":false}
```

Troubleshooting networking issues

Learn basic troubleshooting procedures connected with networking and chrony for Network Time Protocol (NTP).

In order to troubleshoot networking issues, a running Red Hat Ceph Storage cluster is required.

Basic networking troubleshooting

Red Hat Ceph Storage depends heavily on a reliable network connection. Ceph Storage nodes use the network for communicating with each other. Networking issues can cause many problems with Ceph OSDs, such as them flapping, or being incorrectly reported as `down`. Networking issues can also cause the Ceph Monitor's clock skew errors. In addition, packet loss, high latency, or limited bandwidth can impact the cluster performance and stability.

PREREQUISITE

Before you begin, be sure that you have root-level access to the node.
For more information, see the following resources on the Red Hat Customer Portal:

- *Basic Network troubleshooting*
- What is the *ethtool* command and how can I use it to obtain information about my network devices and interfaces
- *RHEL network interface dropping packets*
- *What are the performance benchmarking tools available for Red Hat Ceph Storage?*.
- *Knowledgebase articles and solutions* related to troubleshooting networking issues

1. Install the `net-tools` and `telnet` packages.
   The `net-tools` and `telnet` packages can help troubleshoot network issues that can occur in a Ceph storage cluster.
   For example,

   ```
   [root@host01 ~]# dnf install net-tools
   [root@host01 ~]# dnf install telnet
   ```

2. Log in to the `cephadm` shell and verify that the **`public_network`** parameters in the Ceph configuration file include the correct values.
   For example,

   ```
   [ceph: root@host01 /]# cat /etc/ceph/ceph.conf
   # minimal ceph.conf for 57bddb48-ee04-11eb-9962-001a4a000672
   [global]
           fsid = 57bddb48-ee04-11eb-9962-001a4a000672
           mon_host = [v2:10.74.249.26:3300/0,v1:10.74.249.26:6789/0]
   [v2:10.74.249.163:3300/0,v1:10.74.249.163:6789/0]
   [v2:10.74.254.129:3300/0,v1:10.74.254.129:6789/0]
   [mon.host01]
   public network = 10.74.248.0/21
   ```

3. Exit the shell and verify that the network interfaces are up.
   For example,

   ```
   [root@host01 ~]# ip link list
   1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
   default qlen 1000
       link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT
   group default qlen 1000
       link/ether 00:1a:4a:00:06:72 brd ff:ff:ff:ff:ff:ff
   ```

4. Verify that the Ceph nodes are able to reach each other using their short hostnames .
   Do this check on each node in the storage cluster.

   ```
   ping SHORT_HOST_NAME
   ```

   For example,

   ```
   [root@host01 ~]# ping host02
   ```

5. If you use a firewall, ensure that that Ceph nodes are able to reach each other on their appropriate ports. If you do not use a firewall, continue to step “6” on page 12. The `firewall-cmd` tool validates the port status and the `telnet` tool validates if the port is open.

   ```
   firewall-cmd --info-zone=ZONE
   telnet IP_ADDRESS PORT
   ```

   For example,

```
[root@host01 ~]# firewall-cmd --info-zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens3
  sources:
  services: ceph ceph-mon cockpit dhcpv6-client ssh
  ports: 9283/tcp 8443/tcp 9093/tcp 9094/tcp 3000/tcp 9100/tcp 9095/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@host01 ~]# telnet 192.168.0.22 9100
```

6. Verify that there are no errors on the interface counters.
   Check that the network connectivity between nodes has expected latency, and that there is no packet loss.

   a. Use the **ethtool** command.

      ```
      ethtool -S INTERFACE
      ```

      For example,

      ```
      [root@host01 ~]# ethtool -S ens3 | grep errors
      NIC statistics:
           rx_fcs_errors: 0
           rx_align_errors: 0
           rx_frame_too_long_errors: 0
           rx_in_length_errors: 0
           rx_out_length_errors: 0
           tx_mac_errors: 0
           tx_carrier_sense_errors: 0
           tx_errors: 0
           rx_errors: 0
      ```

   b. Use the **ifconfig** command.
      For example,

      ```
      [root@host01 ~]# ifconfig
      ens3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
              inet 10.74.249.26  netmask 255.255.248.0  broadcast 10.74.255.255
              inet6 fe80::21a:4aff:fe00:672  prefixlen 64  scopeid 0x20<link>
              inet6 2620:52:0:4af8:21a:4aff:fe00:672  prefixlen 64  scopeid 0x0<global>
              ether 00:1a:4a:00:06:72  txqueuelen 1000  (Ethernet)
              RX packets 150549316  bytes 56759897541 (52.8 GiB)
              RX errors 0  dropped 176924  overruns 0  frame 0
              TX packets 55584046  bytes 62111365424 (57.8 GiB)
              TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

      lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
              inet 127.0.0.1  netmask 255.0.0.0
              inet6 ::1  prefixlen 128  scopeid 0x10<host>
              loop  txqueuelen 1000  (Local Loopback)
              RX packets 9373290  bytes 16044697815 (14.9 GiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 9373290  bytes 16044697815 (14.9 GiB)
              TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
      ```

   c. Use the **netstat** command.
      For example,

      ```
      [root@host01 ~]# netstat -ai
      Kernel Interface table
      Iface  MTU    RX-OK      RX-ERR RX-DRP RX-OVR TX-OK      TX-ERR TX-DRP TX-OVR Flg
      ens3   1500   311847720  0      364903 0      114341918  0      0      0
       BMRU
      lo     65536  19577001   0      0      0      19577001   0      0      0      LRU
      ```

7. For performance issues, use the `iperf3` tool to verify the network bandwidth between all nodes of the storage cluster.
   The `iperf3` tool does a simple point-to-point network bandwidth test between a server and a client.

a. Install the `iperf3` package on the Red Hat Ceph Storage nodes that you want to check the bandwidth.
For example,

```
[root@host01 ~]# dnf install iperf3
```

b. On an Red Hat Ceph Storage, start the `iperf3` server.

> **Note:** The default port is 5201, but can be set by using the `-P` command argument.

For example,

```
[root@host01 ~]# iperf3 -s
 Server listening on 5201
```

c. On a different Red Hat Ceph Storage node, start the `iperf3` client.

```
iperf3 -c mon
```

For example,

```
[root@host02 ~]# iperf3 -c mon
 Connecting to host mon, port 5201
 [  4] local xx.x.xxx.xx port 52270 connected to xx.x.xxx.xx port 5201
 [ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
 [  4]   0.00-1.00   sec   114 MBytes   954 Mbits/sec    0    409 KBytes
 [  4]   1.00-2.00   sec   113 MBytes   945 Mbits/sec    0    409 KBytes
 [  4]   2.00-3.00   sec   112 MBytes   943 Mbits/sec    0    454 KBytes
 [  4]   3.00-4.00   sec   112 MBytes   941 Mbits/sec    0    471 KBytes
 [  4]   4.00-5.00   sec   112 MBytes   940 Mbits/sec    0    471 KBytes
 [  4]   5.00-6.00   sec   113 MBytes   945 Mbits/sec    0    471 KBytes
 [  4]   6.00-7.00   sec   112 MBytes   937 Mbits/sec    0    488 KBytes
 [  4]   7.00-8.00   sec   113 MBytes   947 Mbits/sec    0    520 KBytes
 [  4]   8.00-9.00   sec   112 MBytes   939 Mbits/sec    0    520 KBytes
 [  4]   9.00-10.00  sec   112 MBytes   939 Mbits/sec    0    520 KBytes
 - - - - - - - - - - - - - - - - - - - - - - - -
 [ ID] Interval           Transfer     Bandwidth       Retr
 [  4]   0.00-10.00  sec  1.10 GBytes   943 Mbits/sec    0             sender
 [  4]   0.00-10.00  sec  1.10 GBytes   941 Mbits/sec                  receiver

 iperf Done.
```

This output shows a network bandwidth of 1.1 Gbits/second between the Red Hat Ceph Storage nodes, along with no retransmissions (`Retr`) during the test. It is recommended to validate the network bandwidth between all the nodes in the storage cluster.

8. Ensure that all nodes have the same network interconnect speed.
Slower attached nodes might slow down the faster connected ones. Also, ensure that the inter-switch links can handle the aggregated bandwidth of the attached nodes.

```
ethtool INTERFACE
```

For example,

```
[root@host01 ~]# ethtool ens3
 Settings for ens3:
 Supported ports: [ TP ]
 Supported link modes:   10baseT/Half 10baseT/Full
                         100baseT/Half 100baseT/Full
                         1000baseT/Half 1000baseT/Full
 Supported pause frame use: No
 Supports auto-negotiation: Yes
 Supported FEC modes: Not reported
 Advertised link modes:  10baseT/Half 10baseT/Full
                         100baseT/Half 100baseT/Full
                         1000baseT/Half 1000baseT/Full
 Advertised pause frame use: Symmetric
 Advertised auto-negotiation: Yes
 Advertised FEC modes: Not reported
 Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                      100baseT/Half 100baseT/Full
                                      1000baseT/Full
 Link partner advertised pause frame use: Symmetric
 Link partner advertised auto-negotiation: Yes
 Link partner advertised FEC modes: Not reported
 Speed: 1000Mb/s
 Duplex: Full
 Port: Twisted Pair
 PHYAD: 1
 Transceiver: internal
 Auto-negotiation: on
 MDI-X: off
 Supports Wake-on: g
 Wake-on: d
 Current message level: 0x000000ff (255)
        drv probe link timer ifdown ifup rx_err tx_err
 Link detected: yes
```

Basic chrony NTP troubleshooting

Use this information to run basic chrony NTP troubleshooting steps.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Root-level access to the Ceph Monitor node.

For more information, see the following:

- “Clock skew” on page 24
- Checking if chrony is synchronized on Red Hat Documentation.
- How to troubleshoot chrony issues solution on the Red Hat Customer Portal for advanced chrony NTP troubleshooting steps.

1. Verify that the `chrony` daemon is running on the Ceph Monitor hosts:
   For example,

   ```
   [root@mon ~]# systemctl status chrony
   ```

   a. If `chrony` is not running, enable and start it.
      For example,

      ```
      [root@mon ~]# systemctl enable chrony
      [root@mon ~]# systemctl start chrony
      ```

2. Ensure that `chrony` is synchronizing the clocks correctly.
   For example,

   ```
   [root@mon ~]# chronyc sources
   [root@mon ~]# chronyc sourcestats
   [root@mon ~]# chronyc tracking
   ```

3. To configure NTP. see NTP section in Installation Guide for RHEL (x86_64)

4. To configure using chrony, open the NTP configuration file `/etc/chrony.con` in a text editor such as vi or nano, or create a new one if it does not already exist.
5. Add the NTP server entry details `server <NTP_Server> iburst`
6. Restart the chrony service `systemctl restart chrony`.
7. Repeat the same steps on all nodes to configure NTP and to sync time across all nodes.

`cephadm` troubleshooting

As a storage administrator, you can troubleshoot the Red Hat Ceph Storage cluster. Sometimes there is a need to investigate why a `cephadm` command failed or why a specific service does not run properly.

Before troubleshooting `cephadm`, be sure that you have a running Red Hat Ceph Storage cluster

Powering down and starting the cluster

Use this information to troubleshoot issues when powering down and starting the IBM Storage Ceph cluster by using `cephadm` commands.

Stale state file

If a previous shutdown did not complete successfully, a state file might remain on the administration host and prevent the cluster from starting.

Rename or remove the state file, and then retry the operation.

```
mv /var/lib/ceph/<fsid>/cluster-shutdown-state.json /var/lib/ceph/<fsid>/cluster-shutdown-state.json.bak
```

Hosts fail to start

If one or more hosts fail to start during cluster startup:

- Verify SSH connectivity from the administration host
- Ensure that the `ceph.target` systemd unit exists and is enabled
- Check system logs for errors

```
journalctl -u ceph.target
```

Cluster does not return to a healthy state

If the cluster does not reach a healthy state after startup:

- Verify that all OSDs are running
- Check placement group states
- Review logs for errors

```
ceph osd tree
```

```
journalctl -u ceph-<fsid>@osd.*
```

Pause or disable cephadm

If `cephadm` does not behave as expected, you can pause most of the background activity or disable it completely.

Pause `cephadm` with the following commands:

Example

```
[ceph: root@host01 /]# ceph orch pause
```

This stops any changes, but `cephadm` periodically checks hosts to refresh it's inventory of daemons and devices.

If you want to disable `cephadm` completely, run the following commands:

Example

```
[ceph: root@host01 /]# ceph orch set backend ''
[ceph: root@host01 /]# ceph mgr module disable cephadm
```

Note that previously deployed daemon containers continue to exist and start as they did before.

To re-enable `cephadm` in the cluster, run the following commands:

Example

```
[ceph: root@host01 /]# ceph mgr module enable cephadm
[ceph: root@host01 /]# ceph orch set backend cephadm
```

Per service and per daemon event

`cephadm` stores events per service and per daemon in order to aid in debugging failed daemon deployments. These events often contain relevant information.

Per service

Syntax

```
ceph orch ls --service_name SERVICE_NAME --format yaml
```

Example

```
[ceph: root@host01 /]# ceph orch ls --service_name alertmanager --format yaml
service_type: alertmanager
service_name: alertmanager
placement:
  hosts:
  - unknown_host
status:
  ...
  running: 1
  size: 1
events:
- 2021-02-01T08:58:02.741162 service:alertmanager [INFO] "service was created"
- '2021-02-01T12:09:25.264584 service:alertmanager [ERROR] "Failed to apply: Cannot
  place <AlertManagerSpec for service_name=alertmanager> on unknown_host: Unknown hosts"'
```

Per daemon

Syntax

```
ceph orch ps --service-name SERVICE_NAME --daemon-id DAEMON_ID --format yaml
```

Example

```
[ceph: root@host01 /]# ceph orch ps --service-name mds --daemon-id cephfs.hostname.ppdhsz
--format yaml
daemon_type: mds
daemon_id: cephfs.hostname.ppdhsz
hostname: hostname
status_desc: running
...
events:
- 2021-02-01T08:59:43.845866 daemon:mds.cephfs.hostname.ppdhsz [INFO] "Reconfigured
  mds.cephfs.hostname.ppdhsz on host 'hostname'"
```

Check `cephadm` logs

Use the `cephadm` logs for monitoring and troubleshooting cephadm.

You can monitor the `cephadm` log in real time with the following command:

Example

```
[ceph: root@host01 /]# ceph -W cephadm
```

You can see the last few messages with the following command:

Example

```
[ceph: root@host01 /]# ceph log last cephadm
```

If you have enabled logging to files, you can see a `cephadm` log file called `ceph.cephadm.log` on the monitor hosts.

Gather log files

Gather log files to help troubleshoot `cephadm`.

Note:

- Be sure to run all log file commands outside the `cephadm` shell.
- By default, `cephadm` stores logs in `journald`.

Gathering `cephadm` ogs

Action needed	Command	Example	Notes
Gather log files for all daemons	**`journalctl`**		
Read the log file of a specific daemon	`cephadm logs --name DAEMON_NAME`	`[root@host01 ~]# cephadm logs --name cephfs.hostname.ppdhsz`	This command works when run on the same hosts where the daemon is running.
Read the log file of a specific daemon running on a different host	`cephadm logs --fsid FSID --name DAEMON_NAME`	`[root@host01 ~]# cephadm logs --fsid 2d2fd136-6df1-11ea-ae74-002590e526e8 --name cephfs.hostname.ppdhsz`	*FSID* is the cluster ID provided by the **`ceph status`** command.
Fetch all log files of all the daemons on a given host	`for name in $(cephadm ls \| python3 -c "import sys, json; [print(i[name]) for i in json.load(sys.stdin)]") ; do cephadm logs --fsid FSID_OF_CLUSTER --name "$name" > $name; done`	`[root@host01 ~]# for name in $(cephadm ls \| python3 -c "import sys, json; [print(i['name']) for i in json.load(sys.stdin)]") ; do cephadm logs --fsid 57bddb48-ee04-11eb-9962-001a4a000672 --name "$name" > $name; done`	

Collect systemd status

Collect the systemd status for `cephadm` troubleshooting.

To print the state of a systemd unit, run the following command:

Example

```
[root@host01 ~]$ systemctl status ceph-a538d494-fb2a-48e4-82c8-
b91c37bb0684@mon.host01.service
```

List all downloaded container images

List all the container images that are downloaded on a host.

To list all the container images that are downloaded on a host, run the following command:

```
[ceph: root@host01 /]# podman ps -a --format json | jq '.[].Image'
"docker.io/library/rhel9"
"registry.redhat.io/rhceph-alpha/
rhceph-9-rhel9@sha256:9aaea414e2c263216f3cdcb7a096f57c3adf6125ec9f4b0f5f65fa8c43987155"
```

Manually run containers

`cephadm` writes small wrappers that runs a container. Refer to `/var/lib/ceph/`*`CLUSTER_FSID`*`/`*`SERVICE_NAME`*`/unit` to run the container command.

Analyzing SSH errors

If you get the following error:

Example

```
    execnet.gateway_bootstrap.HostNotFound: -F /tmp/cephadm-conf-73z09u6g -i /tmp/cephadm-
identity-ky7ahp_5 root@10.10.1.2

    ...

    raise OrchestratorError(msg) from e
    orchestrator._interface.OrchestratorError: Failed to connect to 10.10.1.2 (10.10.1.2).

    Please make sure that the host is reachable and accepts connections using the cephadm
SSH key
```

Try the following options to troubleshoot the issue:

- To ensure `cephadm` has a SSH identity key, run the following command:

  Example

  ```
  [ceph: root@host01 /]# ceph config-key get mgr/cephadm/ssh_identity_key > ~/
  cephadm_private_key
  INFO:cephadm:Inferring fsid f8edc08a-7f17-11ea-8707-000c2915dd98
  INFO:cephadm:Using recent ceph image docker.io/ceph/ceph:v15 obtained 'mgr/cephadm/
  ssh_identity_key'
  [root@mon1 ~] # chmod 0600 ~/cephadm_private_key
  ```

  If the command fails, `cephadm` does not have a key. To generate a SSH key, run the following command:

  Example

  ```
  [ceph: root@host01 /]# chmod 0600 ~/cephadm_private_key
  ```

  Or

  Example

  ```
  [ceph: root@host01 /]# cat ~/cephadm_private_key | ceph cephadm set-ssk-key -i-
  ```

- To ensure that the SSH configuration is correct, run the following command:

Example

```
[ceph: root@host01 /]# ceph cephadm get-ssh-config
```

- To verify the connection to the host, run the following command:

  Example

  ```
  [ceph: root@host01 /]# ssh -F config -i ~/cephadm_private_key root@host01
  ```

Verify public key is in `authorized_keys`.

To verify that the public key is in the `authorized_keys` file, run the following commands:

Example

```
[ceph: root@host01 /]# ceph cephadm get-pub-key
[ceph: root@host01 /]# grep "`cat ~/ceph.pub`" /root/.ssh/authorized_keys
```

CIDR network error

Classless inter domain routing (CIDR) also known as supernetting, is a method of assigning Internet Protocol (IP) addresses, the `cephadm` log entries shows the current state that improves the efficiency of address distribution and replaces the previous system based on Class A, Class B and Class C networks.

If you see one of the following errors:

```
ERROR: Failed to infer CIDR network for mon ip ***; pass --skip-mon-network to configure it
later
```

Or

```
Must set public_network config option or specify a CIDR network, ceph addrvec, or plain IP
```

You need to run the following command:

Example

```
[ceph: root@host01 /]# ceph config set host public_network hostnetwork
```

Access admin socket

Each Ceph daemon provides an admin socket that bypasses the MONs.

To access the admin socket, enter the daemon container on the host:

Example

```
[ceph: root@host01 /]# cephadm enter --name cephfs.hostname.ppdhsz
[ceph: root@mon1 /]# ceph --admin-daemon /var/run/ceph/ceph-cephfs.hostname.ppdhsz.asok
config show
```

Manually deploying a `mgr` daemon

`cephadm` requires a `mgr` daemon in order to manage the Red Hat Ceph Storage cluster. In case the last `mgr` daemon of an Red Hat Ceph Storage cluster was removed, you can manually deploy a `mgr` daemon, on a random host of the Red Hat Ceph Storage cluster.

Prerequisites

- A running Red Hat Ceph Storage cluster.
- Root-level access to all the nodes.
- Hosts are added to the cluster.

Procedure

1. Log into the `cephadm` shell:

   Example

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

2. Disable the `cephadm` scheduler to prevent `cephadm` from removing the new MGR daemon, with the following command:

   Example

   ```
   [ceph: root@host01 /]# ceph config-key set mgr/cephadm/pause true
   ```

3. Get or create the `auth` entry for the new MGR daemon:

   Example

   ```
   [ceph: root@host01 /]# ceph auth get-or-create mgr.host01.smfvfd1 mon "profile mgr"
   osd "allow *" mds "allow *"
   [mgr.host01.smfvfd1]
   key = AQDhcORgW8toCRAAlMzlqWXnh3cGRjqYEa9ikw==
   ```

4. Open `ceph.conf` file:

   Example

   ```
   [ceph: root@host01 /]# ceph config generate-minimal-conf
   # minimal ceph.conf for 8c9b0072-67ca-11eb-af06-001a4a0002a0
   [global]
   fsid = 8c9b0072-67ca-11eb-af06-001a4a0002a0
   mon_host = [v2:10.10.200.10:3300/0,v1:10.10.200.10:6789/0]
   [v2:10.10.10.100:3300/0,v1:10.10.200.100:6789/0]
   ```

5. Get the container image:

   Example

   ```
   [ceph: root@host01 /]# ceph config get "mgr.host01.smfvfd1" container_image
   ```

6. Create a `config-json.json` file and add the following:

   > **Note:** Use the values from the output of the **`ceph config generate-minimal-conf`** command.

   Example

   ```
   {
     {
     "config": "# minimal ceph.conf for 8c9b0072-67ca-11eb-af06-001a4a0002a0\n[global]
   \n\tfsid = 8c9b0072-67ca-11eb-af06-001a4a0002a0\n\tmon_host =
    [v2:10.10.200.10:3300/0,v1:10.10.200.10:6789/0]
   [v2:10.10.10.100:3300/0,v1:10.10.200.100:6789/0]\n",
     "keyring": "[mgr.Ceph5-2.smfvfd1]\n\tkey =
   AQDhcORgW8toCRAAlMzlqWXnh3cGRjqYEa9ikw==\n"
   }
   }
   ```

7. Exit from the `cephadm` shell:

   Example

   ```
   [ceph: root@host01 /]# exit
   ```

8. Deploy the MGR daemon:

   Example

```
[root@host01 ~]# cephadm --image registry.redhat.io/rhceph-alpha/
rhceph-9-rhel9:latest  deploy --fsid  8c9b0072-67ca-11eb-af06-001a4a0002a0 --name
mgr.host01.smfvfd1 --config-json config-json.json
```

Verification

- In the `cephadm` shell, run the following command:

  Example

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

  You can see a new `mgr` daemon has been added.

Troubleshooting Ceph Monitors

Use this information to learn how to fix the most common errors that are related to Ceph Monitors.

Before troubleshooting Ceph Monitors verify all network connections.

Most common Ceph Monitor errors

Learn the most common error messages that are returned by the **`ceph health detail`** command and that are included in the Ceph logs.

Use the **`ceph health detail`** command on a running Red Hat Ceph Storage cluster.

Ceph Monitor error messages

Understand Ceph Monitor error messages and how to fix them.

"Ceph Monitor error messages" on page 21 provides links to corresponding sections that explain the errors and point to specific procedures to fix the problems.

A table of Ceph Monitor error messages, and links in the documentation of potential fixes.

Ceph Monitor error messages	
Error message	**See**
HEALTH_WARN	
`mon.X is down (out of quorum)`	"Ceph Monitor is out of quorum" on page 22
`clock skew`	"Clock skew" on page 24
`store is getting too big!`	"Ceph Monitor store is getting too big" on page 24

Common Ceph Monitor error messages in Ceph logs

Understand Ceph Monitor error messages in the Ceph logs and how to fix them.

"Ceph Monitor error messages in the Ceph logs" on page 21 provides links to corresponding sections that explain the errors and point to specific procedures to fix the problems.

A table of Ceph Monitor error messages in the Ceph logs and links in the documentation of potential fixes.

Ceph Monitor error messages in the Ceph logs		
Error message	**Log file**	**See**
clock skew	Main cluster log	"Clock skew" on page 24
clocks not synchronized	Main cluster log	"Clock skew" on page 24

Error message	Log file	See
Corruption: error in middle of record	Monitor log	– "Ceph Monitor is out of quorum" on page 22 – "Recovering Ceph Monitor store when using BlueStore" on page 31
Corruption: 1 missing files	Monitor log	– "Ceph Monitor is out of quorum" on page 22 – "Recovering Ceph Monitor store when using BlueStore" on page 31
Caught signal (Bus error)	Monitor log	"Ceph Monitor is out of quorum" on page 22

Ceph Monitor is out of quorum

Understand and troubleshoot Ceph Monitors out of quorum.

A Ceph Monitor is out of quorum if one or more Ceph Monitors are marked as *down* but the other Ceph Monitors are still able to form a quorum. In addition, the **`ceph health detail`** command returns an error message similar to the following example:

```
HEALTH_WARN 1 mons down, quorum 1,2 mon.b,mon.c, mon.a (rank 0) addr 127.0.0.1:6789/0 is
down (out of quorum)
```

What this means

Ceph marks a Ceph Monitor as *down* due to various reasons.

If the `ceph-mon` daemon is not running, it might have a corrupted store or some other error is preventing the daemon from starting. Also, the `/var/` partition might be full. As a consequence, `ceph-mon` is not able to run any operations to the store located by default at `/var/lib/ceph/mon-SHORT_HOST_NAME/store.db` and ends.

If the `ceph-mon` daemon is running but the Ceph Monitor is out of quorum and marked as *down*, the cause of the problem depends on the Ceph Monitor state:

- If the Ceph Monitor is in the *probing* state longer than expected, it cannot find the other Ceph Monitors. This problem can be caused by networking issues, or the Ceph Monitor can have an outdated Ceph Monitor map (`monmap`) and be trying to reach the other Ceph Monitors on incorrect IP addresses. Alternatively, if the `monmap` is up-to-date, the Ceph Monitor clock might not be synchronized.
- If the Ceph Monitor is in the *electing* state longer than expected, the Ceph Monitor's clock might not be synchronized.
- If the Ceph Monitor changes its state from *synchronizing* to *electing* and back, the cluster state is advancing. This means that it is generating new maps faster than the synchronization process can handle.
- If the Ceph Monitor marks itself as the *leader* or a *peon*, then it believes to be in a quorum, while the remaining cluster is sure that it is not. This problem can be caused by failed clock synchronization.

For more information, see:

- Understanding Ceph Monitor status
- Starting, stopping, and restarting all Ceph daemons using the systemctl command
- Using Ceph administration socket

Troubleshooting this problem

Verify that the `ceph-mon` daemon is running. If needed, start the daemon.

```
systemctl status ceph-mon@HOST_NAME
systemctl start ceph-mon@HOST_NAME
```

Replace *HOST_NAME* with the short name of the host where the daemon is running.

> **Note:** If the hostname is not known, use the **hostname -s** command.

- If you are not able to start the `ceph-mon` daemon, go to "The ceph-mon daemon cannot start" on page 23.
- If you are able to start the `ceph-mon` daemon but it is marked as *down*, go to "The ceph-mon daemon is running, but still marked as down" on page 23.

The `ceph-mon` daemon cannot start

1. Check the corresponding Ceph Monitor log located in the `/var/log/ceph/CLUSTER_FSID/` directory.

   > **Note:** By default, the monitor logs are not present in the log folder. You need to enable logging to files for the logs to appear in the folder. See the Ceph daemon logs to enable logging to files.

2. If the log contains error messages similar to the following ones, the Ceph Monitor might have a corrupted store.

   ```
   Corruption: error in middle of record
   Corruption: 1 missing files; example: /var/lib/ceph/mon/mon.0/store.db/1234567.ldb
   ```

   To fix this problem, replace the Ceph Monitor. For more information, see Replacing a failed monitor.

3. If the log contains an error message similar to the following one, the `/var/` partition might be full. Delete any unnecessary data from `/var/`.

   ```
   Caught signal (Bus error)
   ```

   > **Important:** Do not delete any data from the Monitor directory manually. Instead, use the `ceph-monstore-tool` to compact it.

4. If you see any other error messages, open a support ticket. For more information, contact Red Hat Support.

The `ceph-mon` daemon is running, but still marked as `down`

From the Ceph Monitor host that is out of the quorum, use the **mon_status** command to check its state.

```
ceph daemon ID mon_status
```

Replace *ID* with the ID of the Ceph Monitor

For example,

```
[ceph: root@host01 /]# ceph daemon mon.host01 mon_status
```

- If the status is *probing*, verify the locations of the other Ceph Monitors in the **mon_status** output.
  - If the addresses are incorrect, the Ceph Monitor has an incorrect Ceph Monitor map (`monmap`). To fix this problem, see "Injecting a monmap" on page 26.
  - If the addresses are correct, verify that the Ceph Monitor clocks are synchronized. For more information, see "Clock skew" on page 24. To troubleshoot any networking issues, see "Troubleshooting networking issues" on page 10.
- If the status is *electing*, verify that the Ceph Monitor clocks are synchronized. For more information, see "Clock skew" on page 24.
- If the status changes from *electing* to *synchronizing*, open a support ticket. For more information, contact Red Hat Support.

- If the Ceph Monitor is the *leader* or a *peon*, verify that the Ceph Monitor clocks are synchronized. Open a support ticket if synchronizing the clocks does not solve the problem. For more information, contact Red Hat Support.

Clock skew

Understand and troubleshoot clock skew.

A Ceph Monitor is out of quorum, and the **`ceph health detail`** command output contains error messages similar to the following examples:

```
mon.a (rank 0) addr 127.0.0.1:6789/0 is down (out of quorum)
mon.a addr 127.0.0.1:6789/0 clock skew 0.08235s > max 0.05s (latency 0.0045s)
```

In addition, Ceph logs contain error messages similar to the following examples:

```
2022-05-04 07:28:32.035795 7f806062e700 0 log [WRN] : mon.a 127.0.0.1:6789/0 clock skew
0.14s > max 0.05s
2022-05-04 04:31:25.773235 7f4997663700 0 log [WRN] : message from mon.1 was stamped
0.186257s in the future, clocks not synchronized
```

What this means

The clock skew error message indicates that Ceph Monitors' clocks are not synchronized. Clock synchronization is important because Ceph Monitors depend on time precision and behave unpredictably if their clocks are not synchronized.

The **`mon_clock_drift_allowed`** parameter determines what disparity between the clocks is tolerated. By default, this parameter is set to 0.05 seconds.

> **Important:** Do not change the default value of **`mon_clock_drift_allowed`** without previous testing. Changing this value might affect the stability of the Ceph Monitors and the Ceph Storage Cluster in general.

Possible causes of the clock skew error include network problems or problems with chrony Network Time Protocol (NTP) synchronization if that is configured. In addition, time synchronization does not work properly on Ceph Monitors deployed on virtual machines.

For more information, see:

- "Understanding Ceph Monitor status" on page 25
- "Ceph Monitor is out of quorum" on page 22

Troubleshooting this problem

> **Note:** Ceph evaluates time synchronization every five minutes, resulting in delays between fixing the problem and clearing the clock skew messages.

1. Verify that your network works correctly. For more information, see "Troubleshooting networking issues" on page 10. If you use chrony for NTP, see "Basic chrony NTP troubleshooting" on page 14.
2. If you use a remote NTP server, consider deploying your own chrony NTP server on your network. For more information, see **Using the Chrony Suite to Configure NTP** chapter within the *Configuring basic system settings* guide within the Product Documentation for Red Hat Enterprise Linux for your OS version, on Red Hat Documentation.

Ceph Monitor store is getting too big

Understand and troubleshoot Ceph Monitors that are getting too big.

The **`ceph health`** command returns an error message similar to the following example when the Ceph Monitor is getting too big:

```
mon.ceph1 store is getting too big! 48031 MB >= 15360 MB -- 62% avail
```

What this means

Red Hat Ceph Storage Monitor store is a RocksDB database that stores entries as key–values pairs. The database includes a cluster map and is located by default at `/var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME/store.db`.

Querying a large Monitor store can take time, which can cause the Ceph Monitor to be delayed in responding to client queries.

In addition, if the `/var/` partition is full, the Ceph Monitor cannot perform any write operations to the store and terminates. For more information about troubleshooting this issue, see "Ceph Monitor is out of quorum" on page 22.

Troubleshooting this problem

1. Check the size of the database.

   ```
   du -sch /var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME/store.db
   ```

   Specify the name of the cluster and the short host name of the host where the `ceph-mon` is running. For example,

   ```
   # du -sch /var/lib/ceph/mon/ceph-host1/store.db
   47G     /var/lib/ceph/mon/ceph-ceph1/store.db/
   47G     total
   ```

2. Compact the Ceph Monitor store. For more information, see "Compacting Monitor store" on page 28.

Understanding Ceph Monitor status

Understand the different Ceph Monitor statuses.

The **`mon_status`** command returns information about a Ceph Monitor. Information provided includes:

- State
- Rank
- Elections epoch
- Monitor map (`monmap`)

If Ceph Monitors are able to form a quorum, use the **`mon_status`** command with the `ceph` command-line utility.

If Ceph Monitors are not able to form a quorum, but the `ceph-mon` daemon is running, use the administration socket to run the **`mon_status`** command.

For more information, see Using Ceph administration socket

Example output of `mon_status`

```
{
    "name": "mon.3",
    "rank": 2,
    "state": "peon",
    "election_epoch": 96,
    "quorum": [
        1,
        2
    ],
    "outside_quorum": [],
    "extra_probe_peers": [],
    "sync_provider": [],
    "monmap": {
        "epoch": 1,
        "fsid": "d5552d32-9d1d-436c-8db1-ab5fc2c63cd0",
        "modified": "0.000000",
        "created": "0.000000",
        "mons": [
            {
                "rank": 0,
                "name": "mon.1",
                "addr": "172.25.1.10:6789\/0"
            },
            {
                "rank": 1,
                "name": "mon.2",
                "addr": "172.25.1.12:6789\/0"
            },
            {
                "rank": 2,
                "name": "mon.3",
                "addr": "172.25.1.13:6789\/0"
            }
        ]
    }
}
```

Ceph Monitor states

Leader

During the electing phase, Ceph Monitors are electing a leader. The leader is the Ceph Monitor with the highest rank, that is the rank with the lowest value. In the example provided, the leader is `mon.1`.

Peon

Peons are the Ceph Monitors in the quorum that are not leaders. If the leader fails, the peon with the highest rank becomes a new leader.

Probing

A Ceph Monitor is in the probing state if it is looking for other Ceph Monitors. For example, after you start the Ceph Monitors, they are *probing* until they find enough Ceph Monitors specified in the Ceph Monitor map (`monmap`) to form a quorum.

Electing

A Ceph Monitor is in the electing state if it is in the process of electing the leader. Usually, this status changes quickly.

Synchronizing

A Ceph Monitor is in the synchronizing state if it is synchronizing with the other Ceph Monitors to join the quorum. The smaller the Ceph Monitor store it, the faster the synchronization process. Therefore, if you have a large store, synchronization takes a longer time.

Injecting a `monmap`

If a Ceph Monitor has an outdated or corrupted Ceph Monitor map (`monmap`), it cannot join a quorum because it is trying to reach the other Ceph Monitors on incorrect IP addresses. The safest way to fix this problem is to obtain and inject the actual Ceph Monitor map from other Ceph Monitors.

PREREQUISITE

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

- Access to the Ceph Monitor Map.
- Root-level access to the Ceph Monitor node.

> **Note:** This action overwrites the existing Ceph Monitor map that is kept by the Ceph Monitor.

Use this procedure to inject the Ceph Monitor map when the other Ceph Monitors are able to form a quorum, or when at least one Ceph Monitor has a correct Ceph Monitor map. If all Ceph Monitors have corrupted stores, and therefore also a corrupted Ceph Monitor map, see "Recovering Ceph Monitor store when using BlueStore" on page 31.

For more information, see "Ceph Monitor is out of quorum" on page 22.

1. If the remaining Ceph Monitors are able to form a quorum, get the Ceph Monitor map by using the **ceph mon getmap** command.
   If the remaining Ceph Monitors are not able to form the quorum, go to step "2" on page 27.
   For example,

   ```
   [ceph: root@host01 /]# ceph mon getmap -o /tmp/monmap
   ```

2. If the remaining Ceph Monitors are not able to form the quorum and you have at least one Ceph Monitor with a correct Ceph Monitor map, copy it from that Ceph Monitor.
   a. Stop the Ceph Monitor which you want to copy the Ceph Monitor map from.

      ```
      systemctl stop ceph-mon@HOST_NAME
      ```

      For example, to stop the Ceph Monitor running on a host with the `host01` short hostname, run the following command:

      ```
      [root@mon ~]# systemctl stop ceph-mon@host01
      ```

   b. Copy the Ceph Monitor map.

      ```
      ceph-mon -i ID --extract-monmap /tmp/monmap
      ```

      Replace *ID* with the ID of the Ceph Monitor which you want to copy the Ceph Monitor map from.
      For example,

      ```
      [ceph: root@host01 /]#  ceph-mon -i mon.a  --extract-monmap /tmp/monmap
      ```

3. Stop the Ceph Monitor which you want to copy the Ceph Monitor map from.

   ```
   systemctl stop ceph-mon@HOST_NAME
   ```

   For example, to stop the Ceph Monitor running on a host with the `host01` short hostname, run the following command:

   ```
   [root@mon ~]# systemctl stop ceph-mon@host01
   ```

4. Inject the Ceph Monitor map.

   ```
   ceph-mon -i ID --inject-monmap /tmp/monmap
   ```

   Replace *ID* with the ID of the Ceph Monitor with the corrupted or outdated Ceph Monitor map.
   For example,

   ```
   [root@mon ~]# ceph-mon -i mon.host01 --inject-monmap /tmp/monmap
   ```

5. Start the Ceph Monitor.
   For example,

   ```
   [root@mon ~]# systemctl start ceph-mon@host01
   ```

> **Note:** If you copied the Ceph Monitor map from another Ceph Monitor, also start that Ceph Monitor.

Replacing a failed Monitor

When a Ceph Monitor has a corrupted store, replace the Monitor in the storage cluster.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster
- Able to form a quorum
- Root-level access to Ceph Monitor node

For more information, see "Ceph Monitor is out of quorum" on page 22.

1. From the Monitor host, remove the Monitor store.
   The Monitor store is located, by default, in `/var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME`.

   ```
   rm -rf /var/lib/ceph/mon/CLUSTER_NAME-SHORT_HOST_NAME
   ```

   Specify the short hostname of the Monitor host and the cluster name.
   For example, to remove the Monitor store of a Monitor running on `host1` from a cluster called `remote`, run the following command:

   ```
   [root@mon ~]# rm -rf /var/lib/ceph/mon/remote-host1
   ```

2. Remove the Monitor from the Monitor map (`monmap`).

   ```
   ceph mon remove SHORT_HOST_NAME --cluster CLUSTER_NAME
   ```

   Specify the short hostname of the Monitor host and the cluster name.
   For example, to remove the Monitor running on `host1` from a cluster called `remote`, run the following command:

   ```
   [ceph: root@host01 /]# ceph mon remove host01 --cluster remote
   ```

3. Troubleshoot and fix any problems that are related to the underlying file system or hardware of the Monitor host.

Compacting Monitor store

Compact a Monitor store when it has grown too large.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Root-level access to the Ceph Monitor node.

The Monitor store can be compacted in any of the following ways:

- Dynamically by using the **`ceph tell`** command.
- Upon the start of the `ceph-mon` daemon.
- By using the `ceph-monstore-tool` when the `ceph-mon` daemon is not running. Use this method when the other methods fail to compact the Monitor store or when the Monitor is out of quorum and its log contains the Caught signal (Bus error) error message.

Important: Monitor store size changes when the cluster is not in the `active+clean` state or during the rebalancing process. For this reason, compact the Monitor store when rebalancing is completed. Also, ensure that the placement groups are in the `active+clean` state.

For more information, see "Ceph Monitor store is getting too big" on page 24 and "Ceph Monitor is out of quorum" on page 22.

1. While the `ceph-mon` daemon is running, compact the Monitor store.

   ```
   ceph tell mon.HOST_NAME compact
   ```

   Replace *HOST_NAME* with the short hostname of the host where the `ceph-mon` is running.

   Note: If the hostname is not known, use the **`hostname -s`** command.

   For example,

   ```
   [ceph: root@host01 /]# ceph tell mon.host01 compact
   ```

2. Add the **`mon_compact_on_start = true`** parameter to the Ceph configuration under the `[mon]` section.
   For example,

   ```
   [mon]
   mon_compact_on_start = true
   ```

3. Restart the `ceph-mon` daemon.

   ```
   systemctl restart ceph-mon@HOST_NAME
   ```

   Replace *HOST_NAME* with the short name of the host where the daemon is running.

   Note: If the hostname is not known, use the **`hostname -s`** command.

   For example,

   ```
   [root@mon ~]# systemctl restart ceph-mon@host01
   ```

4. Ensure that Monitors have formed a quorum.

   ```
   ceph mon stat
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph mon stat
   ```

5. Optional: If needed, repeat steps "1" on page 29 through "4" on page 29 on any other Monitors.

   Note: Before starting, be sure that the `ceph-test` package is installed.

6. Verify that the `ceph-mon` daemon with the large store is not running.
   If needed, stop the daemon.

   ```
   systemctl status ceph-mon@HOST_NAME
   ```

   ```
   systemctl stop ceph-mon@HOST_NAME
   ```

Replace *HOST_NAME* with the short name of the host where the daemon is running.

> **Note:** If the hostname is not known, use the **`hostname -s`** command.

For example,

```
[root@mon ~]# systemctl status ceph-mon@host01
[root@mon ~]# systemctl stop ceph-mon@host01
```

7. Compact the Monitor store.

```
ceph-monstore-tool /var/lib/ceph/mon/mon.HOST_NAME compact
```

Replace *HOST_NAME* with a short hostname of the Monitor host.
For example,

```
[ceph: root@host01 /]# ceph-monstore-tool /var/lib/ceph/mon/mon.host01 compact
```

8. Start `ceph-mon` again.

```
systemctl start ceph-mon@HOST_NAME
```

For example,

```
[root@mon ~]# systemctl start ceph-mon@host01
```

Opening port for Ceph Manager

Understand how to open ports for Ceph Manager. The ports must be open to keep the clusters working correctly.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Root-level access to Ceph Manager.

The `ceph-mgr` daemons receive placement group information from OSDs on the same range of ports as the `ceph-osd` daemons. If these ports are not open, a cluster devolves from HEALTH_OK to HEALTH_WARN and indicates that placement groups (PGs) are *unknown* with a percentage count of the PGs unknown.

Use this procedure to open ports.

1. For each host running `ceph-mgr` daemons, open ports 6800-7300.
For example,

```
[root@ceph-mgr] # firewall-cmd --add-port 6800-7300/tcp
[root@ceph-mgr] # firewall-cmd --add-port 6800-7300/tcp --permanent
```

2. Restart the `ceph-mgr` daemons.

Recovering Ceph Monitor store

If the Ceph Monitors experience corruption, you can recover them with information that is stored on the OSD nodes.

Ceph Monitors store the cluster map in a key-value store such as LevelDB. If the store is corrupted on a Monitor, the Monitor stops unexpectedly and fails to start again. The Ceph logs might include the following errors:

```
Corruption: error in middle of record
Corruption: 1 missing files; e.g.: /var/lib/ceph/mon/mon.0/store.db/1234567.ldb
```

The Red Hat Ceph Storage clusters use at least three Ceph Monitors so that if one fails, it can be replaced with another one. However, under certain circumstances, all Ceph Monitors can have corrupted stores. For example,

when the Ceph Monitor nodes have incorrectly configured disk or file system settings, a power outage can corrupt the underlying file system.

If there is corruption on all Ceph Monitors, you can recover it with information that is stored on the OSD nodes by using the `ceph-monstore-tool` and `ceph-objectstore-tool` utilities.

Important:

- These procedures cannot recover the following information:
  - Metadata Daemon Server (MDS) keyrings and maps.
  - Placement group (PG) settings.
    `full ratio`

    `full ratio` set, by using the **`ceph pg set_full_ratio`** command.

    `nearfull ratio`

    `nearfull ratio` set, by using the **`ceph pg set_nearfull_ratio`** command.
- Never restore the Ceph Monitor store from an old backup. Rebuild the Ceph Monitor store from the current cluster state by using the steps detailed in the following sections, and restore from that.

Recovering Ceph Monitor store when using BlueStore

Use this information if the Ceph Monitor store is corrupted on all Ceph Monitors and you use the BlueStore back end.

PREREQUISITE

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

- All OSDs containers are stopped.
- Enable Ceph repositories on the Ceph nodes based on their roles.
- The `ceph-test` and `rsync` packages are installed on the OSD and Monitor nodes.
- The `ceph-mon` package is installed on the Monitor nodes.
- The `ceph-osd` package is installed on the OSD nodes.

In containerized environments, this method requires attaching Ceph repositories and restoring to a non-containerized Ceph Monitor first.

Warning: This procedure can cause data loss. If you are unsure about any step in this procedure, contact Red Hat Support for assistance with the recovering process.

For more information about registering Ceph nodes to the Content Delivery Network (CDN), see Registering the Red Hat Ceph Storage nodes to the CDN and attaching subscriptions.

1. Mount all disks with Ceph data to a temporary location.
   Repeat this step for all OSD nodes.
   a. List the data partitions using the **`ceph-volume`** command.
      For example,

      ```
      [ceph: root@host01 /]# ceph-volume lvm list
      ```

   b. Mount the data partitions to a temporary location.

      ```
      mount -t tmpfs tmpfs /var/lib/ceph/osd/ceph-$i
      ```

   c. Restore the SELinux context.

```
for i in {OSD_ID}; do restorecon /var/lib/ceph/osd/ceph-$i; done
```

Replace *OSD_ID* with a numeric, space-separated list of Ceph OSD IDs on the OSD node.

d. Change the owner and group to `ceph:ceph`.

```
for i in {OSD_ID}; do chown -R ceph:ceph /var/lib/ceph/osd/ceph-$i; done
```

Replace *OSD_ID* with a numeric, space-separated list of Ceph OSD IDs on the OSD node.

2. Due to a bug that causes the **update-mon-db** command to use additional db and `db.slow` directories for the Monitor database, you must also copy these directories.

a. Prepare a temporary location outside the container to mount and access the OSD database and extract the OSD maps needed to restore the Ceph Monitor.

```
ceph-bluestore-tool --cluster=ceph prime-osd-dir --dev OSD_DATA --path /var/lib/
ceph/osd/ceph-OSD_ID
```

- Replace *OSD_DATA* with the Volume Group (VG) or Logical Volume (LV) path to the OSD data.
- Replace *OSD_ID* with the ID of the OSD.

b. Create a symbolic link between the BlueStore database and `block.db`.

```
ln -snf BLUESTORE_DATABASE /var/lib/ceph/osd/ceph-OSD_ID/block.db
```

- Replace *BLUESTORE_DATABASE* with the Volume Group (VG) or Logical Volume (LV) path to the BlueStore database.
- Replace *OSD_ID* with the ID of the OSD.

3. From the Ceph Monitor node with the corrupted store, complete the following.

> **Important:** Repeat this procedure for all OSDs on the node.

a. Collect the cluster map from all OSD nodes.
For example,

```
[root@host01 ~]# cd /root/
[root@host01 ~]# ms=/tmp/monstore/
[root@host01 ~]# db=/root/db/
[root@host01 ~]# db_slow=/root/db.slow/

[root@host01 ~]# mkdir $ms
[root@host01 ~]# for host in $osd_nodes; do
                 echo "$host"
                 rsync -avz $ms $host:$ms
                 rsync -avz $db $host:$db
                 rsync -avz $db_slow $host:$db_slow

                 rm -rf $ms
                 rm -rf $db
                 rm -rf $db_slow

                 sh -t $host <<EOF
                     for osd in /var/lib/ceph/osd/ceph-*; do
                     ceph-objectstore-tool --type bluestore --data-path \$osd --op
update-mon-db --mon-store-path $ms

                     done
                 EOF

                         rsync -avz $host:$ms $ms
                         rsync -avz $host:$db $db
                         rsync -avz $host:$db_slow $db_slow
                 done
```

b. Set the appropriate capabilities.
For example,

```
[ceph: root@host01 /]# ceph-authtool /etc/ceph/ceph.client.admin.keyring -n mon.
--cap mon 'allow *' --gen-key
[ceph: root@host01 /]# cat /etc/ceph/ceph.client.admin.keyring
    [mon.]
    key = AQCleqldWqm5IhAAgZQbEzoShkZV42RiQVffnA==
    caps mon = "allow *"
    [client.admin]
    key = AQCmAKld8J05KxAArOWeRAw63gAwwZO5o75ZNQ==
    auid = 0
    caps mds = "allow *"
    caps mgr = "allow *"
    caps mon = "allow *"
    caps osd = "allow *"
```

c. Move all SST files from the `db` and `db.slow` directories to the temporary location.

```
mv /root/db/*.sst /root/db.slow/*.sst /tmp/monstore/store.db
```

d. Rebuild the Monitor store from the collected map.

> **Note:** After using this command, only keyrings extracted from the OSDs and the keyring specified on the `ceph-monstore-tool` command line are present in Ceph's authentication database. You have to recreate or import all other keyrings, such as clients, Ceph Manager, Ceph Object Gateway, and others, so those clients can access the cluster.

For example,

```
[ceph: root@host01 /]# ceph-monstore-tool /tmp/monstore rebuild -- --keyring /
etc/ceph/ceph.client.admin
```

e. Back up the corrupted store.
Repeat this step for all Ceph Monitor nodes.

```
mv /var/lib/ceph/mon/ceph-HOST_NAME/store.db /var/lib/ceph/mon/ceph-HOST_NAME/
store.db.corrupted
```

Replace *HOST_NAME* with the hostname of the Ceph Monitor node.

f. Replace the corrupted store.
Repeat this step for all Ceph Monitor nodes.

```
scp -r /tmp/monstore/store.db HOST_NAME:/var/lib/ceph/mon/ceph-HOST_NAME/
```

Replace *HOST_NAME* with the hostname of the Ceph Monitor node.

4. Unmount all the temporary mounted OSDs on all nodes.
For example,

```
[root@host01 ~]# umount /var/lib/ceph/osd/ceph-*
```

5. Start all the Ceph Monitor daemons.
For example,

```
[root@host01 ~]# systemctl start ceph-mon *
```

6. Ensure that the Monitors are able to form a quorum.

```
ceph -s
```

7. Import the Ceph Manager keyring and start all Ceph Manager processes.

```
ceph auth import -i /etc/ceph/ceph.mgr.HOST_NAME.keyring
systemctl start ceph-mgr@HOST_NAME
```

Replace *HOST_NAME* with the hostname of the Ceph Monitor node.

8. Start all OSD processes across all OSD nodes.

For example,

```
[root@host01 ~]# systemctl start ceph-osd *
```

9. Ensure that the OSDs are returning to service.
   For example,

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

Troubleshooting Ceph OSDs

Use this information to learn how to fix the most common errors that are related to Ceph OSDs.

Before you begin troubleshooting Ceph OSDs do the following:

- Verify your network connection. For more information, see “Troubleshooting networking issues” on page 10.
- Verify that Monitors have a quorum by using the **ceph health** command. If the command returns a health status (`HEALTH_OK`, `HEALTH_WARN`, or `HEALTH_ERR`), the Monitors are able to form a quorum. If not, address any Monitor problems first.
  - For more information about Ceph Monitors, see “Troubleshooting Ceph Monitors” on page 21.
  - For more information about `ceph health`, see “Understanding Ceph health” on page 2.
- Optionally, stop the rebalancing process to save time and resources. For more information, see “Stopping and starting rebalancing” on page 43.

Most common Ceph OSD errors

Learn the most common Ceph OSD errors that are returned by the **ceph health detail** command and that are included in the Ceph logs.

Use the `ceph health detail` command with root-level access to the Ceph OSD nodes.

Ceph OSD error messages

Understand Ceph OSD error messages and how to fix them.

This section lists Ceph OSD errors and warnings with links to troubleshooting topics. Each topic includes an explanation and procedures to resolve the problem.

OSD errors (HEALTH_ERR)

full osds

For information and troubleshooting , see “Full OSDs” on page 35.

OSD warnings (HEALTH_WARN)

nearfull osds

For information and troubleshooting see, “Nearfull OSDs” on page 37.

osds are down

For information and troubleshooting, see the following:

- “Down OSDs” on page 38
  “Flapping OSDs” on page 40

requests are blocked

For information and troubleshooting, see “Slow requests or requests are blocked” on page 42.

slow requests

For information and troubleshooting, see “Slow requests or requests are blocked” on page 42.

Common Ceph OSD error messages in Ceph logs

Understand Ceph OSD error messages in the Ceph logs and how to fix them.

"Ceph OSD error messages in the Ceph logs" on page 35 provides links to corresponding sections that explain the errors and point to specific procedures to fix the problems.

This table lists Ceph OSD error messages in the Ceph logs and links in the documentation of potential fixes.

Ceph OSD error messages in the Ceph logs		
Error message	**Log file**	**See**
heartbeat_check: no reply from osd.X	Main cluster log	"Flapping OSDs" on page 40
wrongly marked me down	Main cluster log	"Flapping OSDs" on page 40
osds have slow requests	Main cluster log	"Slow requests or requests are blocked" on page 42
FAILED assert(0 == "hit suicide timeout")	OSD log	"Down OSDs" on page 38

Full OSDs

Understand and troubleshoot full OSDs.

When OSD nodes are considered *full*, the **`ceph health detail`** command returns an error message similar to the following example:

```
HEALTH_ERR 1 full osds
osd.3 is full at 95%
```

What this means

Ceph prevents clients from running I/O operations on full OSD nodes to avoid losing data. It returns the `HEALTH_ERR full osds` message when the cluster reaches the capacity set by the **`mon_osd_full_ratio`** parameter. By default, this parameter is set to `0.95`, which means 95% of the cluster capacity.

Troubleshooting this problem

To diagnose and resolve full OSDs, complete the following steps:

1. Run **`ceph osd df`** to check how full each OSD is. When invoked without arguments, it reports on the entire cluster. You can also filter by device class or a single OSD.

   ```
   ceph osd df# ceph osd df ssd# ceph osd df osd.1701
   ```

2. Review the `%USE` column to see how full each OSD is. The `VAR` column shows how each OSD compares to the average.
   The `STDDEV` value indicates how evenly OSDs are filled.

   If **`ceph osd df ssd`** or **`ceph osd df nvme`** reports a standard deviation greater than 2.0, the balancer might not be enabled or functioning correctly.

3. If **`ceph df`** shows that the device class of the full OSD is much less full on average than the full OSD, this is likely a balancer issue. For more information, see "Using the Ceph Manager balancer module" on page 0.

4. If OSDs within the device class are balanced, use the following approaches to reduce usage:

   a. Check for benchmark detritus.
      If **`rados bench`** was run against this pool, orphaned data might remain. Check by running the following command:

      ```
      rados -p mypoolname ls | grep -c bench
      ```

      If this command returns a large number, consult Red Hat Support for safe removal by using the **`rados rm`** command.

   b. Request that users remove temporary, outdated, or non-essential data.

c. Run **fstrim** on Ceph Block Device clients that use conventional file systems on volumes in a pool that uses the full OSD. This discards unused blocks and frees capacity.

d. Add nodes or OSDs of the appropriate device class so the balancer can redistribute data and reduce fullness.

5. As an emergency measure, you can temporarily raise the full ratio.

```
ceph osd set-full-ratio 0.98
```

> **Important:** Raising the full ratio is a temporary emergency measure. Restore the default setting as soon as possible to reduce the risk of cluster instability.

For more information, see "Nearfull OSDs" on page 37 and "Deleting data from a full storage cluster" on page 47.

Backfillfull OSDs

Understand and troubleshoot backfillfull OSDs.

If OSDs are considered *backfillfull*, the **ceph health detail** command returns an error message similar to the following example:

```
health: HEALTH_WARN
3 backfillfull osd(s)
Low space hindering backfill (add storage if this doesn't resolve itself): 32 pgs
backfill_toofull
```

What this means

When one or more OSDs has exceeded the backfillfull threshold, Ceph prevents data from rebalancing to this device. This is an early warning that rebalancing might not be complete and that the cluster is approaching full. The default for the backfullfull threshold is 90%.

Troubleshooting this problem

Determine what percent of raw storage (`%RAW USED`) is used, by using the **ceph df** command.

If `%RAW USED` is higher than 70-75%, troubleshoot by using any of the following options:

- For a long-term solution, scale the cluster by adding an OSD node.
- Delete unnecessary data.

  > **Note:** This is a short-term solution to avoid production downtime.

  For Ceph Block Device pools, also run the **fstrim** command on clients with conventional file systems on volumes served by pools that use the backfillfull OSD.
- Increase the `backfillfull` ratio for the OSDs that contain the PGs stuck in `backfull_toofull` to allow the recovery process to continue. Add new storage to the cluster as soon as possible or remove data to prevent filling more OSDs.

  > **Important:** Increasing the backfillfull ratio is a temporary emergency action. Restore the default setting as soon as possible to reduce the risk of cluster instability.

  ```
  ceph osd set-backfillfull-ratio VALUE
  ```

  The range for *VALUE* is 0.0 to 1.0.

  For example,

```
[ceph: root@host01/]# ceph osd set-backfillfull-ratio 0.92
```

For more information and additional resolution strategies, see the following:

- “Nearfull OSDs” on page 37
- “Full OSDs” on page 35
- “Deleting data from a full storage cluster” on page 47

Nearfull OSDs

Understand and troubleshoot nearfull OSDs.

If OSDs are considered *nearfull*, the **ceph health detail** command returns an error message, similar to the following example:

```
HEALTH_WARN 1 nearfull osds
osd.2 is near full at 95%
```

What this means

Ceph returns the `nearfull osds` message when the cluster reaches the capacity set by the **ceph osd set-nearfull-ratio VALUE** command.

For the latest default parameter settings and percentages, see “Full OSDs” on page 35.

Ceph distributes data based on the CRUSH hierarchy in the best possible way but it cannot guarantee equal distribution. The main causes of the uneven data distribution and the `nearfull osds` messages are:

- The OSDs are not balanced among the OSD nodes in the cluster. That is, some OSD nodes host more aggregate capacity than others.
- The placement group (PG) count is not proper as per the number of the OSDs, use case, target PGs per OSD, and OSD utilization.
- The cluster uses inappropriate CRUSH tunables.
- The back-end storage for OSDs is almost full.
- The OSDs have different sizes.

> **Note:** This is only an issue if the pool has too few PGs or if the balancer module is not enabled. Ensure that the target autoscaler ratio is changed from the default and set to `300`.

For more information, see the following:

- “Full OSDs” on page 35.
- How can I test the impact CRUSH map tunable modifications will have on my PG distribution across OSDs in Red Hat Ceph Storage? within the Red Hat Customer Portal.

Troubleshooting this problem

> **Note:** Ratios can trigger even if the **ceph osd df** command shows OSDs slightly below the thresholds. A buffer of 1 - 1.5% is common.

1. Verify that the PG count is sufficient, by using the **ceph osd df** command.
   The **PG** column output should be between `100` and `300`.

   If needed, increase the count. For more information, see “Increasing the placement group” on page 63.

   > **Note:** When ratio values are changed it can take time to take effect. To expedite the value change, you can restart the affected OSDs in a safe rolling fashion.

2. Verify that you are using `optimal` CRUSH tunables to the cluster version. If needed, adjust the CRUSH tunables. For more information, see "CRUSH tuning" on page 0.
3. Use the Ceph Manager balancer module to balance the OSDs.
   For more information, see"Using the Ceph Manager balancer module" on page 0 .
4. Determine how much space remains on the disks that are used by OSDs.
   - To view how much space OSDs use in general.

     ```
     ceph osd df
     ```

     For example,

     ```
     [ceph: root@host01 /]# ceph osd df
     ```

   - To view how much space OSDs use on a particular node, use the **`df`** command from the node containing `nearfull` OSDs.
     For example,

     ```
     [ceph: root@host01 /]# df
     ```

5. If needed, add an OSD node.

Additional considerations and best practices

- If ratios are raised during an emergency, restore them afterward. Leaving them elevated increases the risk of outages.
- Very large clusters may raise ratios by a few percent to reserve capacity.
- Always check that the balancer is working and that there are no overlapping CRUSH roots. Overlaps often occur when some CRUSH rules specify a device class and others do not.
- Run **`fstrim`** on Ceph Block Device clients weekly by using `cron` or one-time during crisis. Avoid using the **`discard mount`** option as it impacts performance continuously.
- Check for unused RADOS pools, empty the Ceph Block Device trash, and orphaned rados bench objects. Remove carefully by using the **`rados rm`** command.

Down OSDs

Understand and troubleshoot OSDs that are down.

If OSDs are considered *down*, the `ceph health detail` command returns an error similar to the following example:

```
HEALTH_WARN 1/3 in osds are down
```

What this means

One of the `ceph-osd` processes is unavailable due to a possible service failure or problems with communication with other OSDs. As a consequence, the surviving `ceph-osd` daemons reported this failure to the Monitors.

If the `ceph-osd` daemon is not running, the underlying OSD drive or file system is either corrupted, or some other error, such as a missing keyring, is preventing the daemon from starting.

Usually, networking issues cause the situation when the `ceph-osd` daemon is running but still marked as `down`.

For more information, see Stale placement groups. To enable logging files, see Ceph daemon logs.

Troubleshooting this problem

1. Determine which OSD is down, by using the **`ceph health detail`** command.

   ```
   [ceph: root@host01 /]# ceph health detail
   HEALTH_WARN 1/3 in osds are down
   osd.0 is down since epoch 23, last address 192.168.106.220:6800/11080
   ```

2. Restart the OSD daemon, by using one of the following commands:

- `ceph orch daemon restart osd.ID`

  Replace *ID* with the ID of the OSD that is in a `down` state.
  For example,

  ```
  [root@host01 ~]# ceph orch daemon restart osd.0
  ```

- `systemctl restart ceph-FSID@osd.ID.service`

  Replace the *FSID* and *ID* with the FSID and ID respectively of the OSD that is in a `down` state.

  ```
  [root@host01 ~]# systemctl restart ceph-b1f94d70-6a2e-4893-a1c3-
  d5e8b9f74261@osd.0.service
  ```

- If you are not able to start `ceph-osd`, go to "The ceph-osd daemon cannot start" on page 39.
- If you are able to start the `ceph-osd` daemon but it is marked in a `down` state, go to "The ceph-osd is running but still marked as down" on page 40.

The `ceph-osd` daemon cannot start

1. If you have a node containing several OSDs (generally, more than twelve), verify that the default maximum number of threads (PID count) is sufficient. For more information, see "Increasing PID count" on page 47.
2. Verify that the OSD data and journal partitions are mounted properly. You can use the **`ceph-volume lvm list`** command to list all devices and volumes that are associated with the Ceph Storage Cluster and then manually inspect if they are mounted properly. See the `mount(8)` manual page for details.
3. If you got the ERROR: missing keyring, cannot use cephx for authentication error message, the OSD is a missing keyring.
4. If you got the ERROR: unable to open OSD superblock on /var/lib/ceph/osd/ceph-1 error message, the `ceph-osd` daemon cannot read the underlying file system.
   - Check the corresponding log file to determine the cause of the failure. By default, Ceph stores log files in the `/var/log/ceph/` directory.
   - An EIO error message indicates a failure of the underlying disk. To fix this problem, replace the underlying OSD disk. For more information, "Replacing an OSD drive" on page 44.
   - If the log includes any other FAILED assert errors, such as the following one, open a support ticket. For more information, contact Red Hat Support.
     See the following FAILED assert message example:

     ```
     FAILED assert(0 == "hit suicide timeout")
     ```

5. Check the `dmesg` output for the errors with the underlying file system or disk:
   - The error -5 error message similar to the one in the following example, indicates corruption of the underlying XFS file system.

     ```
     xfs_log_force: error -5 returned
     ```

     To resolve this issue, unmount the volume and then recover using the **`xfs_repair`** command tool-set. For more information and help using the **`xfs_repair`** command, contact Red Hat Support, referring to this error and documentation.
   - If the `dmesg` output includes any SCSI error error messages, see the SCSI Error Codes Solution Finder solution to determine the best way to fix the problem.
   - If you are still unable to fix the underlying file system, replace the OSD drive. For more information, see "Replacing an OSD drive" on page 44.
6. If the OSD failed with a segmentation fault, such as the one in the following example, gather the required information and open a support ticket. For more information, contact Red Hat Support.

   ```
   Caught signal (Segmentation fault)
   ```

The `ceph-osd` is running but still marked as `down`

Check the corresponding log file to determine the cause of the failure. By default, Ceph stores log files in the `/var/log/ceph/` directory.

- If the log includes any error messages similar to the following example, see Flapping OSDs.

  ```
  wrongly marked me down
      heartbeat_check: no reply from osd.2 since back
  ```

- If there are any other error types, open a support ticket. For more information, contact Red Hat Support.

Flapping OSDs

Flapping OSDs are when OSDs change repeatedly between down and up within a short period of time. Understand and troubleshoot flapping OSDs.

Flapping OSDs are when the **`ceph -w | grep osds`** command shows OSDs repeatedly as `down` and then `up` again within a short period of time, such as in the following example.

```
ceph -w | grep osds
2022-05-05 06:27:20.810535 mon.0 [INF] osdmap e609: 9 osds: 8 up, 9 in
2022-05-05 06:27:24.120611 mon.0 [INF] osdmap e611: 9 osds: 7 up, 9 in
2022-05-05 06:27:25.975622 mon.0 [INF] HEALTH_WARN; 118 pgs stale; 2/9 in osds are down
2022-05-05 06:27:27.489790 mon.0 [INF] osdmap e614: 9 osds: 6 up, 9 in
2022-05-05 06:27:36.540000 mon.0 [INF] osdmap e616: 9 osds: 7 up, 9 in
2022-05-05 06:27:39.681913 mon.0 [INF] osdmap e618: 9 osds: 8 up, 9 in
2022-05-05 06:27:43.269401 mon.0 [INF] osdmap e620: 9 osds: 9 up, 9 in
2022-05-05 06:27:54.884426 mon.0 [INF] osdmap e622: 9 osds: 8 up, 9 in
2022-05-05 06:27:57.398706 mon.0 [INF] osdmap e624: 9 osds: 7 up, 9 in
2022-05-05 06:27:59.669841 mon.0 [INF] osdmap e625: 9 osds: 6 up, 9 in
2022-05-05 06:28:07.043677 mon.0 [INF] osdmap e628: 9 osds: 7 up, 9 in
2022-05-05 06:28:10.512331 mon.0 [INF] osdmap e630: 9 osds: 8 up, 9 in
2022-05-05 06:28:12.670923 mon.0 [INF] osdmap e631: 9 osds: 9 up, 9 in
```

In addition, the Ceph log contains error messages similar to the following example:

```
2022-05-25 03:44:06.510583 osd.50 127.0.0.1:6801/149046 18992 : cluster [WRN] map e600547
wrongly marked me down

2022-05-25 19:00:08.906864 7fa2a0033700 -1 osd.254 609110 heartbeat_check: no reply from
osd.2 since back 2021-07-25 19:00:07.444113 front 2021-07-25 18:59:48.311935 (cutoff
2021-07-25 18:59:48.906862)
```

What this means

The main causes of flapping OSDs are:

- Certain storage cluster operations, such as scrubbing or recovery, take an abnormal amount of time. For example, if you run these operations on objects with a large index or large placement groups. Usually, after these operations finish, the flapping OSDs' problem is solved.
- Problems with the underlying physical hardware. In this case, the **`ceph health detail`** command also returns the `slow requests` error message.
- Problems with the network.

Ceph OSDs cannot manage situations where the private network for the storage cluster fails, or significant latency is on the public client-facing network.

Ceph OSDs use the private network for sending heartbeat packets to each other to indicate that they are `up` and `in`. If the private storage cluster network does not work properly, OSDs are unable to send and receive the heartbeat packets. As a consequence, they report each other as being `down` to the Ceph Monitors, while marking themselves as `up`.

“Ceph configuration parameters that can influence the OSD status” on page 41 lists parameters in the Ceph configuration file that can influence this behavior.

Ceph configuration parameters that can influence the OSD status		
Parameter	**Description**	**Default value**
osd_heartbeat_grace_time	How long OSDs wait for the heartbeat packets to return before reporting an OSD as down to the Ceph Monitors.	20 seconds
mon_osd_min_down_reporters	How many OSDs must report another OSD as down before the Ceph Monitors mark the OSD as down	2

"Ceph configuration parameters that can influence the OSD status" on page 41 shows that in the default configuration, the Ceph Monitors mark an OSD as down if only one OSD made three distinct reports about the first OSD being down. In some cases, if one single host encounters network issues, the entire cluster can experience flapping OSDs. This is because the OSDs that are on the host reports other OSDs in the cluster as down.

Note: The flapping OSDs scenario does not include the situation when the OSD processes are started and then immediately stopped.

Troubleshooting this problem

Important: Flapping OSDs can be caused by MTU misconfiguration on Ceph OSD nodes, at the network switch level, or both. To resolve the issue, set MTU to a uniform size on all storage cluster nodes, including on the core and access network switches with a planned downtime. Do not tune `osd heartbeat min size` because changing this setting can hide issues within the network, and it does not solve actual network inconsistency.

1. Check the output of the **`ceph health detail`** command again.

   Note: If the output includes the `slow requests` error message, see "Slow requests or requests are blocked" on page 42 for information on how to troubleshoot this issue.

   ```
   ceph health detail
   HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests
   30 ops are blocked > 268435 sec
   1 ops are blocked > 268435 sec on osd.11
   1 ops are blocked > 268435 sec on osd.18
   28 ops are blocked > 268435 sec on osd.39
   3 osds have slow requests
   ```

2. Determine which OSDs are marked as down and on what nodes they reside.

   ```
   ceph osd tree | grep down
   ```

3. On the nodes containing the flapping OSDs, troubleshoot and fix any networking problems. For more information, see Troubleshooting networking issues.

4. Alternatively, you can temporarily force Monitors to stop marking the OSDs as down and up by setting the noup and nodown flags.

   Important: Using the noup and nodown flags does not fix the root cause of the problem but only prevents OSDs from flapping.

```
ceph osd set noup
ceph osd set nodown
```

Related information

Heartbeat

Slow requests or requests are blocked

Understand and troubleshoot slow requests or if requests are blocked.

When the `ceph-osd` daemon is slow to respond to a request, the `ceph health detail` command returns an error message similar to the following example:

```
HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests
30 ops are blocked > 268435 sec
1 ops are blocked > 268435 sec on osd.11
1 ops are blocked > 268435 sec on osd.18
28 ops are blocked > 268435 sec on osd.39
3 osds have slow requests
```

In addition to the error message, the Ceph logs include an error message similar to the following messages:

```
2022-05-24 13:18:10.024659 osd.1 127.0.0.1:6812/3032 9 : cluster [WRN] 6 slow requests, 6
included below; oldest blocked for > 61.758455 secs

2022-05-25 03:44:06.510583 osd.50 [WRN] slow request 30.005692 seconds old, received at
{date-time}: osd_op(client.4240.0:8 benchmark_data_ceph-1_39426_object7 [write 0~4194304]
0.69848840) v4 currently waiting for subops from [610]
```

What this means

An OSD with slow requests is every OSD that is not able to service the I/O operations per second (IOPS) in the queue within the time defined by the **`osd_op_complaint_time`** parameter. By default, this parameter is set to 30 seconds.

The main causes of OSDs having slow requests are:

- Problems with the underlying hardware, such as disk drives, hosts, racks, or network switches.
- Problems with the network. These problems are usually connected with flapping OSDs. For more information, see “Flapping OSDs” on page 40.
- System load.

“Slow request types” on page 42 shows the types of slow requests. Use the **`dump_historic_ops`** administration socket command to determine the type of a slow request.

Slow request types

Slow request type	Description
`waiting for rw locks`	The OSD is waiting to acquire a lock on a placement group for the operation.
`waiting for subops`	The OSD is waiting for replica OSDs to apply the operation to the journal.
`no flag points reached`	The OSD did not reach any major operation milestone.
`waiting for degraded object`	The OSDs have not yet replicated an object the specified number of times.

For more information about the administration socket, see Using the Ceph administration socket.

Troubleshooting this problem

1. Determine whether the OSDs with slow or block requests share a common piece of hardware, for example, a disk drive, host, rack, or network switch.
2. If the OSDs share a disk:

a. Use the `smartmontools` utility to check the health of the disk or the logs to determine any errors on the disk.

> **Note:** The `smartmontools` utility is included in the `smartmontools` package.

b. Use the `iostat` utility to get the I/O wait report (`%iowai`) on the OSD disk to determine whether the disk is under heavy load.

> **Note:** The `iostat` utility is included in the `sysstat` package.

3. If the OSDs share the node with another service:
   a. Check the RAM and CPU usage.
   b. Use the `netstat` utility to see the network statistics on the Network Interface Controllers (NICs) and troubleshoot any networking issues.
4. If the OSDs share a rack, check the network switch for the rack. For example, if you use jumbo frames, verify that the NIC in the path has jumbo frames set.
5. If you are unable to determine a common piece of hardware shared by OSDs with slow requests, or to troubleshoot and fix hardware and networking problems, open a support ticket with Red Hat Support.

Stopping and starting rebalancing

When an OSD fails or you stop it, the CRUSH algorithm automatically starts the rebalancing process to redistribute data across the remaining OSDs.

PREREQUISITE

Before you begin, be sure that you have root-level access to the Ceph Monitor node.
Rebalancing can take time and resources, therefore, consider stopping rebalancing during troubleshooting or maintaining OSDs.

> **Note:** Placement groups within the stopped OSDs become *degraded* during troubleshooting and maintenance.

For more information, see Rebalancing and recovery.

1. Log in to the `cephadm` shell.
   For example,

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

2. Set the `noout` flag before stopping the OSD.

```
ceph osd set noout
```

   For example,

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

3. When you finish troubleshooting or maintenance, unset the `noout` flag to start rebalancing.
   For example,

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

Mounting the OSD data partition

Learn how to properly mount the OSD data partiion.

PREREQUISITE

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

- Access to the `ceph-osd` daemon.
- Root-level access to the Ceph Monitor node.

If the OSD data partition is not mounted correctly, the `ceph-osd` daemon cannot start. If you discover that the partition is not mounted as expected, use this information to properly mount the partition.

For more information, see Down OSDs.

1. Mount the partition.

   ```
   mount -o noatime PARTITION /var/lib/ceph/osd/CLUSTER_NAME-OSD_NUMBER
   ```

   - Replace *PARTITION* with the path to the partition on the OSD drive that is dedicated to OSD data.
   - Specify the *CLUSTER_NAME* and *OSD_NUMBER*.

   For example,

   ```
   [root@host01 ~]# mount -o noatime /dev/sdd1 /var/lib/ceph/osd/ceph-0
   ```

2. Try to start the failed `ceph-osd` daemon.

   ```
   systemctl start ceph-osd@OSD_ID
   ```

   Replace the *OSD_ID* with the ID of the OSD.
   For example,

   ```
   [root@host01 ~]# systemctl start ceph-osd@0
   ```

Replacing an OSD drive

Ceph is designed for fault tolerance, which means that it can operate in a *degraded* state without losing data. Therefore, Ceph can operate even if a data storage drive fails. In the context of a failed drive, the *degraded* state means that the extra copies of the data that is stored on other OSDs backfill automatically to other OSDs in the cluster. If this occurs, replace the failed OSD drive and re-create the OSD manually.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Root-level access to the Ceph Monitor node.
- At least one OSD is *down*.

When a drive fails, Ceph reports the OSD as *down*.

For example,

```
HEALTH_WARN 1/3 in osds are down
osd.0 is down since epoch 23, last address 192.168.106.220:6800/11080
```

> **Note:** Ceph can mark an OSD as *down* also as a consequence of networking or permissions problems.

Modern servers typically deploy with hot-swappable drives so you can pull a failed drive and replace it with a new one without bringing down the node. The procedure includes these steps:

1. “Removing an OSD from the Ceph cluster” on page 45

2. “Replacing the physical drive” on page 46
3. “Adding an OSD to the Ceph cluster” on page 46

For more information, see:

- Deploying Ceph OSDs on all available devices
- Deploying Ceph OSDs on specific devices and hosts
- “Down OSDs” on page 38
- Installing

Removing an OSD from the Ceph cluster

1. Log in to the `cephadm` shell.

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

2. Determine which OSD is *down*.

   ```
   ceph osd tree | grep -i down
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph osd tree | grep -i down
   ID  CLASS  WEIGHT   TYPE NAME  STATUS  REWEIGHT  PRI-AFF
   0   hdd    0.00999  osd.0      down    1.00000   1.00000
   ```

   Determine which OSD is *down*.

3. Mark the OSD as `out` for the cluster to rebalance and copy its data to other OSDs

   ```
   ceph osd out osd.OSD_ID
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph osd out osd.0
   marked out osd.0.
   ```

   > **Note:** If the OSD is *down*, Ceph marks it as *out* automatically after 600 seconds when it does not receive any heartbeat packet from the OSD based on the **`mon_osd_down_out_interval`** parameter. When this happens, other OSDs with copies of the failed OSD data begin backfilling to ensure that the required number of copies exists within the cluster. While the cluster is backfilling, the cluster is in a *degraded* state.

4. Ensure that the failed OSD is backfilling.

   ```
   ceph -w | grep backfill
   ```

   For example,

```
[ceph: root@host01 /]# ceph -w | grep backfill
2022-05-02 04:48:03.403872 mon.0 [INF] pgmap v10293282: 431 pgs: 1
active+undersized+degraded+remapped+backfilling, 28 active+undersized+degraded, 49
active+undersized+degraded+remapped+wait_backfill, 59 stale+active+clean, 294
active+clean; 72347 MB data, 101302 MB used, 1624 GB / 1722 GB avail; 227 kB/s rd,
1358 B/s wr, 12 op/s; 10626/35917 objects degraded (29.585%); 6757/35917 objects
misplaced (18.813%); 63500 kB/s, 15 objects/s recovering
2022-05-02 04:48:04.414397 mon.0 [INF] pgmap v10293283: 431 pgs: 2
active+undersized+degraded+remapped+backfilling, 75
active+undersized+degraded+remapped+wait_backfill, 59 stale+active+clean, 295
active+clean; 72347 MB data, 101398 MB used, 1623 GB / 1722 GB avail; 969 kB/s rd,
6778 B/s wr, 32 op/s; 10626/35917 objects degraded (29.585%); 10580/35917 objects
misplaced (29.457%); 125 MB/s, 31 objects/s recovering
2022-05-02 04:48:00.380063 osd.1 [INF] 0.6f starting backfill to osd.0 from (0'0,0'0]
MAX to 2521'166639
2022-05-02 04:48:00.380139 osd.1 [INF] 0.48 starting backfill to osd.0 from (0'0,0'0]
MAX to 2513'43079
2022-05-02 04:48:00.380260 osd.1 [INF] 0.d starting backfill to osd.0 from (0'0,0'0]
MAX to 2513'136847
2022-05-02 04:48:00.380849 osd.1 [INF] 0.71 starting backfill to osd.0 from (0'0,0'0]
MAX to 2331'28496
2022-05-02 04:48:00.381027 osd.1 [INF] 0.51 starting backfill to osd.0 from (0'0,0'0]
MAX to 2513'87544
```

The placement group states change from `active+clean` to `active`, some degraded objects, and then changes to `active+clean` when migration completes.

5. Stop the OSD.

```
ceph orch daemon stop OSD_ID
```

For example,

```
[ceph: root@host01 /]# ceph orch daemon stop osd.0
```

6. Remove the OSD from the storage cluster.

> **Note:** The *OSD_ID* is preserved.

```
ceph orch osd rm OSD_ID --replace
```

For example,

```
[ceph: root@host01 /]# ceph orch osd rm 0 --replace
```

Replacing the physical drive

See the documentation for the hardware node for details on replacing the physical drive.

- If the drive is hot-swappable, replace the failed drive with a new one.
- If the drive is not hot-swappable and the node contains multiple OSDs, you might have to shut down the whole node and replace the physical drive. Consider preventing the cluster from backfilling. For more information, see “Stopping and starting rebalancing” on page 43.
- When the drive appears under the `/dev/` directory, make a note of the drive path.
- If you want to add the OSD manually, find the OSD drive and format the disk.

Adding an OSD to the Ceph cluster

1. After the new drive is inserted, deploy the OSDs by using one of the following options.
   - The OSDs are deployed automatically by the Ceph Orchestrator if the **--unmanaged** parameter is not set.

```
ceph orch apply osd --all-available-devices
```

- Deploy the OSDs on all the available devices with the **unmanaged** parameter set to `true`.

```
ceph orch apply osd --all-available-devices --unmanaged=true
```

- Deploy the OSDs on specific devices and hosts.

```
ceph orch daemon add osd HOST:PATH_TO_DRIVE
```

For example,

```
[ceph: root@host01 /]# ceph orch daemon add osd host02:/dev/sdb
```

2. Ensure that the CRUSH hierarchy is accurate.

```
ceph osd tree
```

Increasing PID count

Learn how to increase the maximum number of threads (PID count).

The default PID count can be sufficient with nodes containing more than 12 Ceph OSDs, especially during recovery. As a result, some `ceph-osd` daemons can terminate and fail to start again. If this happens, increase the maximum possible number of threads allowed.

Temporarily increase the maximum number of threads

Temporarily increase the maximum number of threads.

```
sysctl -w kernel.pid.max=PID_COUNT
```

For example,

```
[root@mon ~]# sysctl -w kernel.pid.max=4194303
```

Permanently increase the maximum number of threads

Permanently increase the maximum number of threads by updating the `/etc/sysctl.conf` file.
For example,

```
[root@mon ~]# sysctl -w kernel.pid.max=4194303
```

Deleting data from a full storage cluster

Ceph automatically prevents any I/O operations on OSDs that reached the capacity that is specified by the **`mon_osd_full_ratio`** parameter and returns the full osds error message. To fix this error, delete any unnecessary data.

PREREQUISITE

Before you begin, be sure that you have root-level access to the Ceph Monitor node.

> **Note:** The **`mon_osd_full_ratio`** parameter sets the value of the **`full_ratio`** parameter when creating a cluster. You cannot change the value of **`mon_osd_full_ratio`** afterward. To temporarily increase the **`full_ratio`** value, increase the `set-full-ratio` instead.

1. Log in to the `cephadm` shell.

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

2. Determine the current value of **`full_ratio`**.
By default, the **`full_ratio`** value is set to `0.95`.

```
ceph osd dump | grep -i full
```

For example,

```
[ceph: root@host01 /]# ceph osd dump | grep -i full
full_ratio 0.95
```

3. Temporarily change the value of `set-full-ratio` to `0.97`.

> **Important:** Do not set the **`set-full-ratio`** to a value higher than 0.97. Setting this parameter to a higher value makes the recovery process harder. As a result, you might not be able to recover full OSDs at all.

```
ceph osd set-full-ratio 0.97
```

4. Check that the parameter was successfully changed to `0.97`.
For example

```
[ceph: root@host01 /]# ceph osd dump | grep -i full
full_ratio 0.97
```

5. Monitor the cluster state, by using the **`ceph -w`** command.
As soon as the cluster changes its state from *full* to *nearfull*, delete any unnecessary data.
For more information about these states, see "Full OSDs" on page 35 and "Nearfull OSDs" on page 37.
6. After any unnecessary data is deleted, change the `full_ratio` value back to `0.95`.

```
ceph osd set-full-ratio 0.95
```

7. Check that the parameter was successfully changed to `0.95`.
For example,

```
[ceph: root@host01 /]# ceph osd dump | grep -i full
full_ratio 0.95
```

Troubleshooting multi-site Ceph Object Gateway

Understand how to troubleshoot and learn how to fix the most common errors that are related to multi-site Ceph Object Gateway configuration and operational conditions.

> **Note:** When the **`rados-admin bucket sync status`** command reports that the bucket is behind on shards even if the data is consistent across multi-site, run more writes to the bucket. It synchronizes the `sync status` reports and displays the message bucket is caught up with source.

To troubleshoot multi-site Ceph Object Gateway errors make sure to have:

- A running Red Hat Ceph Storage cluster.
- A running Ceph Object Gateway.

Error code definitions for Ceph Object Gateway

The Ceph Object Gateway logs contain error and warning messages to help troubleshooting your environment.

This information lists the common Ceph Object Gateway common error and warning messages and suggested resolutions. For more information, contact Red Hat Support.

- "Common error messages" on page 49
- "Syncing error messages" on page 49

Common error messages	
Error message	**Description**
data_sync: ERROR: a sync operation returned error	A high-level data sync process is indicating that lower-level bucket sync process returned an error. This message is redundant; the bucket sync error appears previously it in the log.
data sync: ERROR: failed to sync object: BUCKET_NAME:OBJECT_NAME	Either the process failed to fetch the required object over HTTP from a remote gateway or the process failed to write that object to RADOS and it will be tried again.
data sync: ERROR: failure in sync, backing out (sync_status=2)	A low-level message that shows one of the previous conditions, specifically that the data was deleted before it was able to sync and thus showing a `-2 ENOENT` status.
data sync: ERROR: failure in sync, backing out (sync_status=-5)	A low-level message that shows one of the previous conditions, specifically that Ceph failed to write that object to RADOS and thus showing a `-5 EIO`.
ERROR: failed to fetch remote data log info: ret=11	This is the `EAGAIN` generic error code from `libcurl` showing an error condition from another gateway. By default, the system tries again.
meta sync: ERROR: failed to read mdlog info with (2) No such file or directory	The shard of the mdlog was never created so there is nothing to sync.

Syncing error messages	
Error message	**Description**
failed to sync object	Either the process failed to fetch this object over HTTP from a remote gateway or it failed to write that object to RADOS and it will be tried again.
failed to sync bucket instance: (11) Resource temporarily unavailable	A connection issue between primary and secondary zones.
failed to sync bucket instance: (125) Operation canceled	A racing condition exists between writes to the same RADOS object.

Syncing a multi-site Ceph Object Gateway

A multi-site sync reads the change log from other zones.

To get a high-level view of the sync progress from the metadata and the data logs, you can use the following command:

```
radosgw-admin sync status
```

This command lists which log shards, if any, which are behind their source zone.

> **Note:** Sometimes you might observe recovering shards when running the **`radosgw-admin sync status`** command. For data sync, there are 128 shards of replication logs that are each processed independently. If any of the actions triggered by these replication log events result in any error from the network, storage, or elsewhere, those errors get tracked so the operation can retry again later. While a given shard has errors that need a retry, **`radosgw-admin sync status`** command reports that shard as `recovering`. This recovery happens automatically, so the operator does not need to intervene to resolve them.

If the results of the sync status you have previously run reports log shards are behind, run the following command substituting the shard-id for *X*.

Buckets within a multi-site object can also be monitored on the Ceph dashboard. For more information, see Monitoring buckets of a multi-site object.

```
radosgw-admin data sync status --shard-id=X --source-zone=ZONE_NAME
```

For example,

```
[ceph: root@host01 /]# radosgw-admin data sync status --shard-id=27 --source-zone=us-east
{
  "shard_id": 27,
  "marker": {
        "status": "incremental-sync",
        "marker": "1_1534494893.816775_131867195.1",
        "next_step_marker": "",
        "total_entries": 1,
        "pos": 0,
        "timestamp": "0.000000"
   },
   "pending_buckets": [],
   "recovering_buckets": [
        "pro-registry:4ed07bb2-a80b-4c69-aa15-fdc17ae6f5f2.314303.1:26"
   ]
}
```

The output lists which buckets are next to sync and which buckets, if any, are going to be retried due to previous errors.

Inspect the status of individual buckets with the following command, substituting the bucket ID for *X*.

```
radosgw-admin bucket sync status --bucket=X
```

Replace *X* with the ID number of the bucket.

The result shows which bucket index log shards are behind their source zone.

A common error in sync is EBUSY, which means the sync is already in progress, often on another gateway. Read errors that are written to the sync error log, which can be read with the following command:

```
radosgw-admin sync error list
```

The syncing process tries again until it is successful. Errors can still occur that can require intervention.

Performance counters for multi-site Ceph Object Gateway data sync

Use performance counters to measure data sync for multi-site configurations of the Ceph Object Gateway.

The following performance counters are available for multi-site configurations of the Ceph Object Gateway to measure data sync:

- `poll_latency` measures the latency of requests for remote replication logs.
- `fetch_bytes` measures the number of objects and bytes fetched by data sync.

For more information about performance encounters, see “Ceph performance counters” on page 0.

- Use the **`ceph --admin-daemon`** command to view the current metric data for the performance counters.

  > **Note:** Run the **`ceph --admin-daemon`** command from the node running the daemon.

  ```
  ceph --admin-daemon /var/run/ceph/cluster-id/ceph-client.rgw.RGW_ID.asok perf dump
  data-sync-from-ZONE_NAME
  ```

  For example,

```
[ceph: root@host01 /]# ceph --admin-daemon  /var/run/ceph/cluster-id/ceph-
client.rgw.host02-rgw0.103.94309060818504.asok perf dump data-sync-from-us-west

{
    "data-sync-from-us-west": {
        "fetch bytes": {
            "avgcount": 54,
            "sum": 54526039885
        },
        "fetch not modified": 7,
        "fetch errors": 0,
        "poll latency": {
            "avgcount": 41,
            "sum": 2.533653367,
            "avgtime": 0.061796423
        },
        "poll errors": 0
    }
}
```

Synchronizing data in a multi-site Ceph Object Gateway configuration

In a multi-site Ceph Object Gateway configuration of a storage cluster, failover and failback causes data synchronization to stop. The **`radosgw-admin sync status`** command reports that the data sync is behind for an extended period of time.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Ceph Object Gateway configured at two sites at least.

You can run the **`radosgw-admin data sync init`** command to synchronize data between the sites and then restart the Ceph Object Gateway. This command does not touch any actual object data and initiates data sync for a specified source zone. It causes the zone to restart a full sync from the source zone.

Important: Contact Red Hat Support before running the **`data sync init`** command. If you are going for a full restart of sync, and if there is a lot of data that needs to be synced on the source zone, then the bandwidth consumption is high and then you have to plan accordingly.

Note: If a user accidentally deletes a bucket on the secondary site, you can use the **`metadata sync init`** command on the site to synchronize data.

1. Check the sync status between the sites.
   For example,

   ```
   [ceph: host04 /]# radosgw-admin sync status
             realm d713eec8-6ec4-4f71-9eaf-379be18e551b (india)
         zonegroup ccf9e0b2-df95-4e0a-8933-3b17b64c52b7 (shared)
             zone 04daab24-5bbd-4c17-9cf5-b1981fd7ff79 (primary)
    current time 2022-09-15T06:53:52Z
    zonegroup features enabled: resharding
    metadata sync no sync (zone is master)
        data sync source: 596319d2-4ffe-4977-ace1-8dd1790db9fb (secondary)
                          syncing
                          full sync: 0/128 shards
                          incremental sync: 128/128 shards
                          data is caught up with source
   ```

2. Synchronize data from the secondary zone.

   ```
   radosgw-admin data sync init --source-zone primary
   ```

   For example,

```
[ceph: root@host04 /]# radosgw-admin data sync init --source-zone primary
```

3. Restart all the Ceph Object Gateway daemons at the site.

```
ceph orch restart rgw.myrgw
```

Troubleshooting `radosgw-admin` commands after upgrading a cluster

Troubleshoot using **`radosgw-admin`** commands inside the `cephadm` shell after upgrading a cluster.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Ceph Object Gateway configured at two sites at least.

The following is an example of errors that could be emitted after trying to run **`radosgw-admin`** commands inside the `cephadm` shell after upgrading a cluster.

```
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to decode obj
from .rgw.root:periods.91d2a42c-735b-492a-bcf3-05235ce888aa.3
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 failed reading current period info:  (5) Input/
output error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to start notify service ((5) Input/
output error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to init services (ret=(5) Input/
output error)
couldn't init storage provider
```

For example,

```
[ceph: root@host01 /]# date;radosgw-admin bucket list
Mon May 13 09:05:30 UTC 2024
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to decode obj
from .rgw.root:periods.91d2a42c-735b-492a-bcf3-05235ce888aa.3
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 failed reading current period info:  (5)
Input/output error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to start notify service ((5)
Input/output error
2024-05-13T09:05:30.607+0000 7f4e7c4ea500  0 ERROR: failed to init services (ret=(5) Input/
output error)
couldn't init storage provider
```

- Repair the connection by running the command again with the **`-- radosgw-admin`** syntax.

```
cephadm shell -- radosgw-admin COMMAND
```

For example,

```
[root@host01 /]# cephadm shell -- radosgw-admin bucket list
```

Troubleshooting Ceph placement groups

Learn to troubleshoot the most common errors that are related to the Ceph Placement groups (PGs).

Before troubleshooting Ceph placement groups:

- Verify your network connection.
- Ensure that Monitors are able to form a quorum.
- Ensure that all healthy OSDs are *up* and *in*, and the backfilling and recovery processes are finished.

Most common Ceph placement groups errors

Understand the most common error messages that are returned by the **`ceph health detail`** command.

Placement groups that are stuck in a state that is not optimal can be listed, as described in “Listing placement groups stuck in stale, inactive, or unclean states” on page 59.

Note: When troubleshooting Ceph placement group errors, be sure you have both a running Red Hat Ceph Storage cluster and Ceph Object Gateway.

Placement group error messages

Understand placement group error messages and how to fix them.

"Placement group error messages" on page 53 provides links to corresponding sections that explain the errors and point to specific procedures to fix the problems.

A table of common placement group error messages, and links in the documentation of potential fixes.

Placement group error messages	
Error message	**See**
HEALTH_ERR	
`pgs down`	"Placement groups are down" on page 56
`pgs inconsistent`	"Inconsistent placement groups" on page 54
`scrub errors`	"Inconsistent placement groups" on page 54
HEALTH_WARN	
`pgs stale`	"Stale placement groups" on page 53
`unfound`	"Unfound objects" on page 57

Stale placement groups

The **`ceph health`** command lists some placement groups (PGs) as `stale`.

The following is an example of a placement group marked as *stale*:

```
HEALTH_WARN 24 pgs stale; 3/300 in osds are down
```

What this means

The Monitor marks a placement group as *stale* when it does not receive any status update from the primary OSD of the placement group's acting set or when other OSDs reported that the primary OSD is *down*.

Usually, PGs enter the *stale* state after you start the storage cluster and until the peering process completes. However, when the PGs remain *stale* for longer than expected, it might indicate that the primary OSD for those PGs is *down* or not reporting PG statistics to the Monitor. When the primary OSD storing *stale* PGs is back *up*, Ceph starts to recover the PGs.

The `mon_osd_report_timeout` setting determines how often OSDs report PGs statistics to Monitors. By default, this parameter is set to `0.5`, which means that OSDs report the statistics every half a second.

For more information, see Monitoring Placement Group Sets and "Down OSDs" on page 38.

Troubleshooting this problem

1. Use the **`ceph health detail`** to identify which PGs are *stale* and on what OSDs they are stored. The error message includes information similar to the following example:

   ```
   [ceph: root@host01 /]# ceph health detail
   HEALTH_WARN 24 pgs stale; 3/300 in osds are down
   ...
   pg 2.5 is stuck stale+active+remapped, last acting [2,0]
   ...
   osd.10 is down since epoch 23, last address 192.168.106.220:6800/11080
   osd.11 is down since epoch 13, last address 192.168.106.220:6803/11539
   osd.12 is down since epoch 24, last address 192.168.106.220:6806/11861
   ```

2. Troubleshoot any problems with the OSDs that are marked as *down*.

Inconsistent placement groups

Understand and troubleshoot inconsistent placement groups (PGs).

Some placement groups are marked as active + clean + inconsistent and the **ceph health detail** command returns an error message similar to the following example:

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

What this means

When Ceph detects inconsistencies in one or more replicas of an object in a placement group, it marks the placement group as *inconsistent*. The most common inconsistencies are:

- Objects have an incorrect size.
- Objects are missing from one replica after a recovery finished.

In most cases, errors during scrubbing cause inconsistency within placement groups.

For more information, see the following:

- "Listing placement group inconsistencies" on page 59
- Data integrity
- Scrubbing the OSD
- "Repairing inconsistent placement groups" on page 62

Troubleshooting this problem

1. Log in to the cephadm shell.

   ```
   cephadm shell
   ```

2. Use the **ceph health detail** command to determine which placement group is in the *inconsistent* state.

   ```
   [ceph: root@host01 /]# ceph health detail
   HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
   pg 0.6 is active+clean+inconsistent, acting [0,1,2]
   2 scrub errors
   ```

3. Determine why the placement group is *inconsistent*.
   a. Start the deep scrubbing process on the placement group.

      ```
      ceph pg deep-scrub ID
      ```

      Replace *ID* with the ID of the *inconsistent* placement group.
      For example,

      ```
      [ceph: root@host01 /]# ceph pg deep-scrub 0.6
      instructing pg 0.6 on osd.0 to deep-scrub
      ```

   b. Use the **ceph -w** command to find messages related to the *inconsistent* placement group.

      ```
      ceph -w | grep ID
      ```

      Replace *ID* with the ID of the *inconsistent* placement group.

      For example,

      ```
      [ceph: root@host01 /]# ceph -w | grep 0.6
      2022-05-26 01:35:36.778215 osd.106 [ERR] 0.6 deep-scrub stat mismatch, got
      636/635 objects, 0/0 clones, 0/0 dirty, 0/0 omap, 0/0 hit_set_archive, 0/0
      whiteouts, 1855455/1854371 bytes.
      2022-05-26 01:35:36.788334 osd.106 [ERR] 0.6 deep-scrub 1 errors
      ```

4. If the output includes any error messages similar to the following ones, you can repair the `inconsistent` placement group.

```
PG.ID shard OSD: soid OBJECT missing attr , missing attr ATTRIBUTE_TYPE
PG.ID shard OSD: soid OBJECT digest 0 != known digest DIGEST, size 0 != known size
SIZE
PG.ID shard OSD: soid OBJECT size 0 != known size SIZE
PG.ID deep-scrub stat mismatch, got MISMATCH
PG.ID shard OSD: soid OBJECT candidate had a read error, digest 0 != known digest
DIGEST
```

– If the output includes any error messages similar to the following ones, you can repair the *inconsistent* placement group.

```
PG.ID shard OSD: soid OBJECT missing attr , missing attr ATTRIBUTE_TYPE
PG.ID shard OSD: soid OBJECT digest 0 != known digest DIGEST, size 0 != known size SIZE
PG.ID shard OSD: soid OBJECT size 0 != known size SIZE
PG.ID deep-scrub stat mismatch, got MISMATCH
PG.ID shard OSD: soid OBJECT candidate had a read error, digest 0 != known digest
DIGEST
```

– If the output includes any error messages similar to the following ones, it is not safe to repair the *inconsistent* placement group because you can lose data.

```
PG.ID shard OSD: soid OBJECT digest DIGEST != known digest DIGEST
PG.ID shard OSD: soid OBJECT omap_digest DIGEST != known omap_digest DIGEST
```

If this occurs, open a support ticket. For more information, contact Red Hat Support.

Unclean placement groups

Understand and troubleshoot unclean placement groups (PGs).

The **ceph health** command returns an error message similar to the following example:

```
HEALTH_WARN 197 pgs stuck unclean
```

What this means

Ceph marks a placement group as *unclean* if it has not achieved the *active+clean* state for the number of seconds specified in the **mon_pg_stuck_threshold** parameter in the Ceph configuration file. The default value of **mon_pg_stuck_threshold** is `300` seconds.

If a placement group is *unclean*, it contains objects that are not replicated the number of times specified in the **osd_pool_default_size** parameter. The default value of **osd_pool_default_size** is `3`, which means that Ceph creates three replicas.

Usually, *unclean* placement groups indicate that some OSDs might be *down*.

For more information, see Listing placement groups stuck in stale inactive or unclean state.

Troubleshooting this problem

1. Use the **ceph osd tree** command to determine which OSDs are in a *down* state.
   For example,

```
[ceph: root@host01 /]# ceph osd tree
```

2. Troubleshoot and fix any problems with the OSDs. For more information, see “Down OSDs” on page 38.

Inactive placement groups

Understand and troubleshoot inactive placement groups (PGs).

The **ceph health** command returns an error message similar to the following example:

```
HEALTH_WARN 197 pgs stuck inactive
```

What this means

Ceph marks a placement group as *inactive* if it has not be active for the number of seconds specified in the **mon_pg_stuck_threshold** parameter in the Ceph configuration file. The default value of **mon_pg_stuck_threshold** is `300` seconds.

Usually, `inactive` placement groups indicate that some OSDs might be *down*.

For more information, see "Listing placement groups stuck in stale, inactive, or unclean states" on page 59 and "Down OSDs" on page 38.

Troubleshooting this problem

1. Use the **ceph osd tree** command to determine which OSDs are in a *down* state.
   For example,

   ```
   [ceph: root@host01 /]# ceph osd tree
   ```

2. Troubleshoot and fix any problems with the OSDs. For more information, see "Down OSDs" on page 38.

Placement groups are down

Understand and troubleshoot placement groups that are in a *down* state.

The **ceph health detail** command reports that some placement groups are *down*. For example,

```
HEALTH_ERR 7 pgs degraded; 12 pgs down; 12 pgs peering; 1 pgs recovering; 6 pgs stuck
unclean; 114/3300 degraded (3.455%); 1/3 in osds are down
...
pg 0.5 is down+peering
pg 1.4 is down+peering
...
osd.1 is down since epoch 69, last address 192.168.106.220:6801/8651
```

What this means

In certain cases, the peering process can be blocked, which prevents a placement group from becoming active and usable. Usually, a failure of an OSD causes the peering failures.

For more information, see Ceph OSD peering.

Troubleshooting this problem

Determine what blocks the peering process.

```
ceph pg ID query
```

Replace *ID* with the ID of the placement group that is *down*.

For example,

```
[ceph: root@host01 /]#  ceph pg 0.5 query

{ "state": "down+peering",
  ...
  "recovery_state": [
        { "name": "Started\/Primary\/Peering\/GetInfo",
          "enter_time": "2021-08-06 14:40:16.169679",
          "requested_info_from": []},
        { "name": "Started\/Primary\/Peering",
          "enter_time": "2021-08-06 14:40:16.169659",
          "probing_osds": [
                0,
                1],
          "blocked": "peering is blocked due to down osds",
          "down_osds_we_would_probe": [
                1],
          "peering_blocked_by": [
                { "osd": 1,
                  "current_lost_at": 0,
                  "comment": "starting or marking this osd lost may let us proceed"}]},
        { "name": "Started",
          "enter_time": "2021-08-06 14:40:16.169513"}
    ]
}
```

The `recovery_state` section includes information on why the peering process is blocked.

- If the output includes the peering is blocked due to down osds error message, see "Down OSDs" on page 38.
- If you see any other error message, open a support ticket with Red Hat Support.

Unfound objects

Understand and troubleshoot unfound objects.

The **ceph health** command returns an error message similar to the following one, containing the `unfound` keyword, as in the following example:

```
HEALTH_WARN 1 pgs degraded; 78/3778 unfound (2.065%)
```

What this means

Ceph marks objects as *unfound* when it knows these objects or their newer copies exist but it is unable to find them. As a result, Ceph cannot recover such objects and proceed with the recovery process.

Example situation

A placement group stores data on `osd.1` and `osd.2`.

1. `osd.1` goes *down*.
2. `osd.2` handles some write operations.
3. `osd.1` comes *up*.
4. A peering process between `osd.1` and `osd.2` starts, and the objects missing on `osd.1` are queued for recovery.
5. Before Ceph copies new objects, `osd.2` goes `down`.

As a result, `osd.1` knows that these objects exist, but there is no OSD that has a copy of the objects.

In this scenario, Ceph is waiting for the failed node to be accessible again, and the *unfound* objects blocks the recovery process.

Troubleshooting this problem

1. Log in to the `cephadm` shell.
   For example,

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

2. Use the **`ceph health detail`** command to determine which placement group contains *unfound* objects.
   For example,

```
[ceph: root@host01 /]# ceph health detail
HEALTH_WARN 1 pgs recovering; 1 pgs stuck unclean; recovery 5/937611 objects degraded
(0.001%); 1/312537 unfound (0.000%)
pg 3.8a5 is stuck unclean for 803946.712780, current state active+recovering, last
acting [320,248,0]
pg 3.8a5 is active+recovering, acting [320,248,0], 1 unfound
recovery 5/937611 objects degraded (0.001%); **1/312537 unfound (0.000%)**
```

3. List more information about the placement group.

```
ceph pg ID query
```

   Replace *ID* with the ID of the placement group containing the *unfound* objects.

   For example,

```
[ceph: root@host01 /]# ceph pg 3.8a5 query
{ "state": "active+recovering",
  "epoch": 10741,
  "up": [
        320,
        248,
        0],
  "acting": [
        320,
        248,
        0],
<snip>
  "recovery_state": [
        { "name": "Started\/Primary\/Active",
          "enter_time": "2021-08-28 19:30:12.058136",
          "might_have_unfound": [
                { "osd": "0",
                  "status": "already probed"},
                { "osd": "248",
                  "status": "already probed"},
                { "osd": "301",
                  "status": "already probed"},
                { "osd": "362",
                  "status": "already probed"},
                { "osd": "395",
                  "status": "already probed"},
                { "osd": "429",
                  "status": "osd is down"}],
          "recovery_progress": { "backfill_targets": [],
              "waiting_on_backfill": [],
              "last_backfill_started": "0\/\/0\/\/-1",
              "backfill_info": { "begin": "0\/\/0\/\/-1",
                  "end": "0\/\/0\/\/-1",
                  "objects": []},
              "peer_backfill_info": [],
              "backfills_in_flight": [],
              "recovering": [],
              "pg_backend": { "pull_from_peer": [],
                  "pushing": []}},
          "scrub": { "scrubber.epoch_start": "0",
              "scrubber.active": 0,
              "scrubber.block_writes": 0,
              "scrubber.finalizing": 0,
              "scrubber.waiting_on": 0,
              "scrubber.waiting_on_whom": []}},
        { "name": "Started",
          "enter_time": "2021-08-28 19:30:11.044020"}],
```

   The `might_have_unfound` section includes OSDs where Ceph tried to locate the *unfound* objects:

   - The `already probed` status indicates that Ceph cannot locate the *unfound* objects in that OSD.
   - The `osd is down` status indicates that Ceph cannot contact that OSD.

4. Troubleshoot the OSDs that are marked as *down*. See “Down OSDs” on page 38 for details.
5. If you are unable to fix the problem that causes the OSD to be *down*, open a support ticket. For more information, see Red Hat Support.

Listing placement groups stuck in *stale*, *inactive*, or *unclean* states

After a failure, placement groups enter states such as *degraded* or *peering*. These states indicate normal progression through the failure recovery process. If a placement group stays in one of these states for a longer time than expected, it can be an indication of a larger problem.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Root-level access to the node.

The Monitors report when placement groups (PGs) get stuck in a state that is not optimal.

The `mon_pg_stuck_threshold` option in the Ceph configuration file determines the number of seconds after which placement groups are considered *inactive*, *unclean*, or *stale*.

"Placement group states" on page 59 lists the `mon_pg_stuck_threshold` states along with their descriptions.

Placement group states			
State	**What it means**	**Most common causes**	**See**
`inactive`	The PG is not able to service read/write requests.	Peering problems	"Inactive placement groups" on page 55
`unclean`	The PG contains objects that are not replicated the desired number of times. Something is preventing the PG from recovering.	– Unfound objects – OSDs are down – Incorrect configuration	"Unclean placement groups" on page 55
`stale`	The status of the PG has not been updated by a `ceph-osd` daemon.	OSDs are down	"Stale placement groups" on page 53

1. Log in to the `cephadm` shell.
   For example,

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

2. List the stuck PGs.
   For example,

   ```
   [ceph: root@host01 /]# ceph pg dump_stuck inactive
   [ceph: root@host01 /]# ceph pg dump_stuck unclean
   [ceph: root@host01 /]# ceph pg dump_stuck stale
   ```

   For for more information, see Placement Group states.

Listing placement group inconsistencies

Find and troubleshoot Ceph placement group (PG) issues by finding inconsistencies.

Use the `rados` utility to list inconsistencies in various replicas of objects. Use the `--format=json-pretty` option to list a more detailed output.

Before listing PG inconsistencies, be sure to have a running a Red Hat Ceph Storage cluster running correctly and root-level access to the node.

For more information, see "Inconsistent placement groups" on page 54 and "Repairing inconsistent placement groups" on page 62.

Troubleshoot placement group inconsistencies, by using the following procedures:

- "Listing inconsistent placement groups in a pool" on page 60
- "Listing inconsistent objects in a placement group" on page 60

– “Listing inconsistent snapshot sets in a placement group” on page 61

Listing inconsistent placement groups in a pool

To list all the inconsistent placement groups in a pool, run the following command:

```
rados list-inconsistent-pg POOL --format=json-pretty
```

For example,

```
[ceph: root@host01 /]# rados list-inconsistent-pg data --format=json-pretty
 [0.6]
```

Listing inconsistent objects in a placement group

To list all inconsistent objects in a placement group with ID, run the following command:

```
rados list-inconsistent-obj PLACEMENT_GROUP_ID
```

For example,

```
[ceph: root@host01 /]# rados list-inconsistent-obj 0.6
 {
     "epoch": 14,
     "inconsistents": [
         {
             "object": {
                 "name": "image1",
                 "nspace": "",
                 "locator": "",
                 "snap": "head",
                 "version": 1
             },
             "errors": [
                 "data_digest_mismatch",
                 "size_mismatch"
             ],
             "union_shard_errors": [
                 "data_digest_mismatch_oi",
                 "size_mismatch_oi"
             ],
             "selected_object_info": "0:602f83fe:::foo:head(16'1 client.4110.0:1 dirty|
data_digest|omap_digest s 968 uv 1 dd e978e67f od ffffffff alloc_hint [0 0 0])",
             "shards": [
                 {
                     "osd": 0,
                     "errors": [],
                     "size": 968,
                     "omap_digest": "0xffffffff",
                     "data_digest": "0xe978e67f"
                 },
                 {
                     "osd": 1,
                     "errors": [],
                     "size": 968,
                     "omap_digest": "0xffffffff",
                     "data_digest": "0xe978e67f"
                 },
                 {
                     "osd": 2,
                     "errors": [
                         "data_digest_mismatch_oi",
                         "size_mismatch_oi"
                     ],
                     "size": 0,
                     "omap_digest": "0xffffffff",
                     "data_digest": "0xffffffff"
                 }
             ]
         }
     ]
 }
```

Use the following fields to determine what is causing the inconsistency.

name

The name of the object with inconsistent replicas.

nspace

The namespace that is a logical separation of a pool. The value is empty by default.

locator

The key that is used as the alternative of the object name for placement.

snap

The snapshot ID of the object. The only writable version of the object is called `head`. If an object is a clone, this field includes its sequential ID.

version

The version ID of the object with inconsistent replicas. Each write operation to an object increments it.

errors

A list of errors that indicate inconsistencies between shards without determining which shard or shards are incorrect. See the `shard` array to further investigate the errors.

data_digest_mismatch

The digest of the replica that is read from one OSD is different from the other OSDs.

size_mismatch

The size of a clone or the `head` object does not match the expectation.

read_error

This error indicates inconsistencies that are most likely caused by disk errors.

union_shard_error

The union of all errors specific to shards. These errors are connected to a faulty shard. The errors that end with `oi` indicate that you must compare the information from a faulty object to information with selected objects. See the `shard` array to further investigate the errors.

In this example, the object replica that is stored on `osd.2` has a different digest than the replicas stored on `osd.0` and `osd.1`. Specifically, the digest of the replica is not `0xffffffff` as calculated from the shard read from `osd.2`, but `0xe978e67f`. In addition, the size of the replica that is read from `osd.2` is 0, while the size reported by `osd.0` and `osd.1` is 968.

Listing inconsistent snapshot sets in a placement group

To list all the inconsistent sets of snapshots, run the following command:

```
rados list-inconsistent-snapset PLACEMENT_GROUP_ID
```

For example,

```
[ceph: root@host01 /]# rados list-inconsistent-snapset 0.23 --format=json-pretty
 {
     "epoch": 64,
     "inconsistents": [
         {
             "name": "obj5",
             "nspace": "",
             "locator": "",
             "snap": "0x00000001",
             "headless": true
         },
         {
             "name": "obj5",
             "nspace": "",
             "locator": "",
             "snap": "0x00000002",
             "headless": true
         },
         {
             "name": "obj5",
             "nspace": "",
             "locator": "",
             "snap": "head",
             "ss_attr_missing": true,
             "extra_clones": true,
             "extra clones": [
                 2,
                 1
             ]
         }
     ]
```

This command returns the following error messages.

ss_attr_missing

One or more attributes are missing. Attributes are information about snapshots encoded into a snapshot set as a list of key-value pairs.

ss_attr_corrupted

One or more attributes fail to decode.

clone_missing

A clone is missing.

snapset_mismatch

The snapshot set is inconsistent by itself.

head_mismatch

The snapshot set indicates that `head` exists or not, but the scrub results report otherwise.

headless

The `head` of the snapshot set is missing.

size_mismatch

The size of a clone or the `head` object does not match the expectation.

Repairing inconsistent placement groups

Due to an error during deep scrubbing, some placement groups (PGs) can include inconsistencies. Ceph reports such placement groups as *inconsistent*.

PREREQUISITE

Before you begin, be sure that you have root-level access to the Ceph Monitor node.
The following is an example of a report with an inconsistent placement group.

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

Important: Only certain inconsistencies can be repaired.
Do not repair the placement groups if the Ceph logs include the following errors:

```
PG.ID shard OSD: soid OBJECT digest DIGEST != known digest DIGEST
PG.ID shard OSD: soid OBJECT omap_digest DIGEST != known omap_digest DIGEST
```

If any of these errors occur, contact Red Hat Support.

For more information, see "Inconsistent placement groups" on page 54 and "Listing placement group inconsistencies" on page 59.

- Repair the inconsistent placement group.

  ```
  ceph pg repair ID
  ```

  Replace *ID* with the ID of the *inconsistent* placement group.

Increasing the placement group

Learn how to increase the placement group.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster in a healthy state.
- Root-level access to the node.

Insufficient placement group (PG) count impacts the performance of the Ceph cluster and data distribution. It is one of the main causes of the nearfull osds error messages.

The recommended ratio is between 100 and 300 PGs per OSD. This ratio can decrease when you add more OSDs to the cluster.

The **pg_num** and **pgp_num** parameters determine the PG count. These parameters are configured per each pool, and therefore, you must adjust each pool with low PG count separately.

Important: Increasing the PG count is the most intensive process that you can run on a Ceph cluster. This process might have a serious performance impact if not done in a slow and methodical way. After you increase **pgp_num**, you will not be able to stop or reverse the process and you must complete it. Consider increasing the PG count outside of business-critical processing time allocation, and alert all clients about the potential performance impact. Do not change the PG count if the cluster is in the HEALTH_ERR state.

For more information, see "Nearfull OSDs" on page 37 and Monitoring placement group sets.

1. Reduce the impact of data redistribution and recovery on individual OSDs and OSD hosts.
   a. Lower the value of the **osd max backfills**, **osd_recovery_max_active**, and **osd_recovery_op_priority** parameters.

      ```
      [ceph: root@host01 /]# ceph tell osd.* injectargs '--osd_max_backfills 1 --
      osd_recovery_max_active 1 --osd_recovery_op_priority 1'
      ```

   b. Disable the shallow and deep scrubbing.

      ```
      ceph osd set noscrub
      ```

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

      For example,

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

2. Calculate the optimal value of the **pg_num** and **pgp_num** parameters.
   Use the Ceph Placement Groups (PGs) per Pool Calculator to calculate the optimal values.
3. Increase the **pg_num** value in small increments until you reach the wanted value.
   a. Determine the starting increment value.
      Use a very low value that is a power of two, and increase it when you determine the impact on the cluster. The optimal value depends on the pool size, OSD count, and client I/O load.
   b. Increment the **pg_num** value.
      Specify the pool name and the new value.

      ```
      ceph osd pool set POOL pg_num VALUE
      ```

      For example,

      ```
      [ceph: root@host01 /]# ceph osd pool set data pg_num 4
      ```

   c. Monitor the status of the cluster by using the **ceph -s** command.
      The PGs state changes from *creating* to *active+clean*. Wait until all PGs are in the *active+clean* state.
4. Increase the **pgp_num** value in small increments until you reach the wanted value.
   a. Determine the starting increment value.
      Use a very low value that is a power of two, and increase it when you determine the impact on the cluster. The optimal value depends on the pool size, OSD count, and client I/O load.
   b. Increment the **pgp_num** value.
      Specify the pool name and the new value.

      ```
      ceph osd pool set POOL pgp_num VALUE
      ```

      For example,

      ```
      [ceph: root@host01 /]# ceph osd pool set data pgp_num 4
      ```

   c. Monitor the status of the cluster by using the **ceph -s** command.
      The PGs state changes through *peering*, *wait_backfill*, *backfilling*, *recover*, and others. Wait until all PGs are in the *active+clean* state.
5. Repeat steps "1" on page 63 though "4" on page 64 for all pools with insufficient PG count.
6. Set **osd max backfills**, **osd_recovery_max_active**, and **osd_recovery_op_priority** to their default values.
   For example,

   ```
   [ceph: root@host01 /]# ceph tell osd.* injectargs '--osd_max_backfills 1 --
   osd_recovery_max_active 3 --osd_recovery_op_priority 3'
   ```

7. Enable the shallow and deep scrubbing.

   ```
   ceph osd unset noscrub
   ```

   ```
   ceph osd unset nodeep-scrub
   ```

   For example,

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

Interpreting placement group dump output

The **ceph pg dump** command displays a wealth of information regarding placement groups.

Because placement groups (PGs) typically range from hundreds to tens of thousands, redirecting the command output to a file or piping it through a pager is advised for easier readability.

See the following output example for the **ceph pg dump** command.

See the following tables for descriptions of each of the output types:

- PG stats
- Pool stats
- OSD stats

```
; PG stats
    PG_STAT  OBJECTS  MISSING_ON_PRIMARY  DEGRADED  MISPLACED  UNFOUND  BYTES  OMAP_BYTES*
 OMAP_KEYS*  LOG    DISK_LOG  STATE         STATE_STAMP                      VERSION
 REPORTED    UP       UP_PRIMARY  ACTING   ACTING_PRIMARY  LAST_SCRUB  SCRUB_STAMP
             LAST_DEEP_SCRUB  DEEP_SCRUB_STAMP                SNAPTRIMQ_LEN
    5.1b           0                   0         0          0        0      0            0
          0      0         0  active+clean  2024-06-05T22:12:41.648713+0000        0'0
255:294  [0,7,8]           0  [0,7,8]              0         0'0
 2024-06-01T16:34:39.253348+0000              0'0  2024-05-30T05:04:41.133279+0000
     0
    4.1a           0                   0         0          0        0      0            0
          0      0         0  active+clean  2024-06-05T22:12:44.045134+0000        0'0
255:291  [3,4,5]           3  [3,4,5]              3         0'0
 2024-06-01T07:43:59.302980+0000              0'0  2024-05-27T16:45:10.130663+0000
     0
    1.1f           0                   0         0          0        0      0            0
          0      0         0  active+clean  2024-06-05T22:12:44.044842+0000        0'0
255:262  [4,1,3]           4  [4,1,3]              4         0'0
 2024-06-01T08:52:22.029352+0000              0'0  2024-05-27T16:44:47.679985+0000
     0
    ....
    ....
    ; Pool stats
    POOLID OBJECTS MISSING_ON_PRIMARY DEGRADED MISPLACED UNFOUND BYTES OMAP_BYTES*
OMAP_KEYS* LOG     DISK_LOG
    5      4       0                  0        0         0       763   0           0
   8       8
    4      8       0                  0        0         0       0     0           0
   1145    1145
    3      209     0                  0        0         0       3702  0           0
   250271  250271
    1      4       0                  0        0         0       1403  0           0
   4       4
    2      0       0                  0        0         0       0     0           0
   0       0
    ...
    ...
    ; OSD stats
    OSD_STAT  USED     AVAIL    USED_RAW  TOTAL   HB_PEERS           PG_SUM  PRIMARY_PG_SUM
    4          70 MiB  9.9 GiB    70 MiB  10 GiB  [0,1,2,3,5,6,7,8]      56              21
    7          44 MiB   10 GiB    44 MiB  10 GiB  [0,1,2,3,4,5,6,8]      39               9
    2          44 MiB   10 GiB    44 MiB  10 GiB  [0,1,3,4,5,6,7,8]      34              11
```

PG stats	
PG stats column name	**Description**
PG_STAT	PG ID
OBJECTS	Number of RADOS objects associated with this PG
MISSING_ON_PRIMARY	Number of missing RADOS objects on primary OSD
DEGRADED	Number of degraded (missing desired replicas) RADOS objects
MISPLACED	Number of misplaced (wrong location in the cluster) RADOS objects
UNFOUND	Number of unfound RADOS objects
BYTES	Total number of bytes stored in RADOS objects
OMAP_BYTES*	Total number of bytes of all omaps
OMAP_KEYS*	Total number of omap keys

PG stats column name	Description
LOG	Number of PG log entries
LOG_DUPS	Number of PG log entries for duplicate ops
DISK_LOG	Number of stored PG log entries
STATE	Current state of the PG
STATE_STAMP	Timestamp of the last state change
VERSION	Version of the most recent write to the placement group
REPORTED	Version of pg_stat reported for this PG
UP	The set of OSDs responsible for transferring data
UP_PRIMARY	The primary OSD for this PG; handles client requests
ACTING	The set of OSDs that currently have a full and working version of the PG
ACTING_PRIMARY	Primary OSD responsible for handling client requests
LAST_SCRUB	Last PG version that completed a shallow scrub
SCRUB_STAMP	Timestamp when this PG last completed a shallow scrub
LAST_DEEP_SCRUB	Last PG version that completed a deep scrub
DEEP_SCRUB_STAMP	Timestamp when this PG last completed a deep-scrub
SNAPTRIMQ_LEN	Size of the snaptrim queue
LAST_SCRUB_DURATION	The duration (in seconds) of the last completed scrub
SCRUB_SCHEDULING	Indicates whether this PG is scheduled/queued/ being scrubbed
OBJECTS_SCRUBBED	The number of RADOS objects scrubbed in a PG after a scrub begins
OBJECTS_TRIMMED	The number of RADOS objects trimmed

Pool stats	
Field	**Description**
POOLID	Numeric pool ID
OBJECTS	Number of RADOS objects stored in this pool
MISSING_ON_PRIMARY	Number of RADOS objects missing from this pool
DEGRADED	Number of degraded (missing desired replicas) RADOS objects in this pool
MISPLACED	Number of misplaced (wrong location in the cluster) RADOS objects in the pool
UNFOUND	Number of unfound RADOS objects in this pool
BYTES	Total number of bytes stored in RADOS objects in this pool
OMAP_BYTES*	Total number of bytes stored in OMAPs in this pool
OMAP_KEYS*	Total number of OMAP keys stored in this pool
LOG	Sum of log entries across all PGs within this pool
DISK_LOG	Sum of all persisted log entries across all PGs within this pool

OSD stats	
Field	**Description**
OSD_STAT	OSD ID
USED	The amount of data stored on the OSD
AVAIL	The amount of free space available on the OSD
USED_RAW	The amount of raw storage consumed by user data, internal overhead, and reserved capacity
TOTAL	The total storage capacity of the OSD
HB_PEERS	Peer OSDs that this OSD heartbeats
PG_SUM	The number of placement group replicas or shards on this OSD
PRIMARY_PG_SUM	The number of placement groups for which this OSD acts as primary

Troubleshooting Ceph objects

Use the `ceph-objectstore-tool` utility to perform high-level or low-level object operations. The `ceph-objectstore-tool` utility can help you troubleshoot problems related to objects within a particular OSD or placement group.

> **Important:** Manipulating objects can cause unrecoverable data loss. Before using the `ceph-objectstore-tool` utility, contact Red Hat Support.

Before troubleshooting Ceph objects, verify that there are no network related issues.

Troubleshooting high-level object operations

Use the `ceph-objectstore-tool` utility to run high-level object operations.

> **Important:** Manipulating objects can cause unrecoverable data loss. Before using the `ceph-objectstore-tool` utility, contact Red Hat Support.

The `ceph-objectstore-tool` utility supports the following high-level object operations:

- List objects
- List lost objects
- Fix lost objects

Before troubleshooting high-level object operations be sure to have root-level access to the Ceph OSD nodes.

Listing objects

The OSD can contain zero to many placement groups, and zero to many objects within a placement group (PG). The `ceph-objectstore-tool` utility allows you to list objects stored within an OSD.

PREREQUISITE

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

- Root-level access to the Ceph OSD node.
- Stopping the `ceph-osd` daemon.

1. Verify that the appropriate OSD is down.

```
systemctl status ceph-osd@OSD_NUMBER
```

For example,

```
[root@host01 ~]# systemctl status ceph-osd@1
```

2. Log in to the OSD container.

```
cephadm shell --name osd.OSD_ID
```

For example,

```
[root@host01 ~]# cephadm shell --name osd.0
```

3. Identify all the objects within an OSD, regardless of their placement group.

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op list
```

For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op
list
```

4. Identify all the objects within a placement group.

```
ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op list
```

For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --
pgid 0.1c --op list
```

5. Identify the PG that an object belongs to.

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op list OBJECT_ID
```

For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op
list default.region
```

Fixing lost objects

Use the `ceph-objectstore-tool` utility to list and fix *lost* and *unfound* legacy objects that are stored within a Ceph OSD.

PREREQUISITE

- Root-level access to the Ceph OSD node.
- Stopping the `ceph-osd` daemon.

> **Note:** This procedure applies only to legacy objects.

1. Verify that the appropriate OSD is down.

```
systemctl status ceph-osd@OSD_NUMBER
```

For example,

```
[root@host01 ~]# systemctl status ceph-osd@1
```

2. Log in to the OSD container.

```
cephadm shell --name osd.OSD_ID
```

For example,

```
[root@host01 ~]# cephadm shell --name osd.0
```

3. List all lost legacy objects.

```
ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost --dry-run
```

For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --op
fix-lost --dry-run
```

4. Fix the *lost* and *unfound* objects by using the `ceph-objectstore-tool` utility.
Use the following procedures, depending on the circumstance.
   - Fix all lost objects.

     ```
     ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost
     ```

     For example,

     ```
     [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0
     --op fix-lost
     ```

   - Fix all lost objects within a placement group.

     ```
     ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID --op fix-lost
     ```

     For example,

     ```
     [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0
     --pgid 0.1c --op fix-lost
     ```

   - Fix a lost object by its identifier.

     ```
     ceph-objectstore-tool --data-path PATH_TO_OSD --op fix-lost OBJECT_ID
     ```

     For example,

     ```
     [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0
     --op fix-lost default.region
     ```

Troubleshooting low-level object operations

Use the `ceph-objectstore-tool` utility to run low-level object operations.

> **Important:** Manipulating objects can cause unrecoverable data loss. Before using the `ceph-objectstore-tool` utility, contact Red Hat Support.

Before using the `ceph-objectstore-tool` utility, verify that you have root-level access to the Ceph OSD nodes.

Manipulating object content

With the `ceph-objectstore-tool` utility, you can get or set bytes on an object.

PREREQUISITE

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

- Root-level access to the Ceph OSD node.
- Stopping the `ceph-osd` daemon.

1. Verify that the appropriate OSD is down.

   ```
   systemctl status ceph-osd@OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# systemctl status ceph-osd@1
   ```

2. Find the object by listing the objects of the OSD or placement group (PG).
3. Log in to the OSD container.

   ```
   cephadm shell --name osd.OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

4. Create a backup and working copy of the object.
   This must be done before setting bytes on an object.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
   OBJECT \
   get-bytes > OBJECT_FILE_NAME
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --
   pgid 0.1c \
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}'  \
   get-bytes > zone_info.default.backup

   [ceph: root@host01 /]#  ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --
   pgid 0.1c \
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}'  \
   get-bytes > zone_info.default.working-copy
   ```

5. Edit the working copy object file and modify the object contents.
6. Set the bytes of the object.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
   OBJECT \
   set-bytes < OBJECT_FILE_NAME
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --
   pgid 0.1c \
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}' \
   set-bytes < zone_info.default.working-copy
   ```

Removing an object

Use the `ceph-objectstore-tool` utility to remove an object. By removing an object, its contents and references are removed from the placement group (PG).

PREREQUISITE

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

- Root-level access to the Ceph OSD node.

– Stopping the `ceph-osd` daemon.

> **Important:** You cannot recreate an object after it is removed.

1. Log in to the OSD container.

   ```
   cephadm shell --name osd.OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

2. Remove an object.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID \
   OBJECT \
   remove
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --
   pgid 0.1c \
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}' \
   remove
   ```

Listing object map

Use the `ceph-objectstore-tool` utility to list the contents of the object map (OMAP).

PREREQUISITE

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

– Root-level access to the Ceph OSD node.

– Stopping the `ceph-osd` daemon.

Listing the contents of the OMAP provides a list of keys.

1. Verify that the appropriate OSD is down.

   ```
   systemctl status ceph-osd@OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# systemctl status ceph-osd@1
   ```

2. Log in to the OSD container.

   ```
   cephadm shell --name osd.OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. List the object map.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD --pgid PG_ID OBJECT list-omap
   ```

   For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 --
pgid 0.1c \
'{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
amespace":""}' \
list-omap
```

Manipulating object map header

The `ceph-objectstore-tool` utility outputs the object map (OMAP) header with the values that are associated with the object's keys.

PREREQUISITE

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

- Root-level access to the Ceph OSD node.
- Stopping the `ceph-osd` daemon.

1. Verify that the appropriate OSD is down.

   ```
   systemctl status ceph-osd@OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# systemctl status ceph-osd@1
   ```

2. Log in to the OSD container.

   ```
   cephadm shell --name osd.OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. Get the object map header.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   get-omaphdr > OBJECT_MAP_FILE_NAME
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
   --pgid 0.1c
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}'  \
   get-omaphdr > zone_info.default.omaphdr.txt
   ```

4. Set the object map header.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   set-omaphdr > OBJECT_MAP_FILE_NAME
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
   --pgid 0.1c
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}'  \
   set-omaphdr > zone_info.default.omaphdr.txt
   ```

Manipulating object map key

Use the `ceph-objectstore-tool` utility to change the object map (OMAP) key.

PREREQUISITE

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

- Root-level access to the Ceph OSD node.
- Stopping the `ceph-osd` daemon.

To change the OMAP, provide the data path, the placement group identifier (PG ID), the object, and the key.

1. Log in to the OSD container.

   ```
   cephadm shell --name osd.OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

2. Get the object map key.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
    --pgid PG_ID OBJECT \
    get-omap KEY > OBJECT_MAP_FILE_NAME
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
    --pgid 0.1c
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}'  \
    get-omap "" > zone_info.default.omap.txt
   ```

3. Set the object map key.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
    --pgid PG_ID OBJECT \
    set-omap KEY > OBJECT_MAP_FILE_NAME
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
    --pgid 0.1c
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}'  \
    set-omap "" > zone_info.default.omap.txt
   ```

4. Remove the object map key.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
    --pgid PG_ID OBJECT \
    rm-omap KEY
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
    --pgid 0.1c
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}'  \
    rm-omap ""
   ```

Listing object attributes

Use the `ceph-objectstore-tool` utility to list object attributes.

PREREQUISITE

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

- Root-level access to the Ceph OSD node.
- Stopping the `ceph-osd` daemon.

The output provides you with the object's keys and values.

1. Verify that the appropriate OSD is down.

   ```
   systemctl status ceph-osd@OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# systemctl status ceph-osd@1
   ```

2. Log in to the OSD container.

   ```
   cephadm shell --name osd.OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. List the object attributes.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   list-attrs
   ```

   For example,

   ```
   [ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
   --pgid 0.1c
   '{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
   amespace":""}' \
   list-attrs
   ```

Manipulating object attribute key

Use the `ceph-objectstore-tool` utility to change object attributes.

PREREQUISITE

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

- Root-level access to the Ceph OSD node.
- Stop the `ceph-osd` daemon.

To manipulate object attributes you need the data paths, the placement group identifier (PG ID), the object, and the key in the object attribute.

1. Verify that the appropriate OSD is down.

   ```
   systemctl status ceph-osd@OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# systemctl status ceph-osd@1
   ```

2. Log in to the OSD container.

   ```
   cephadm shell --name osd.OSD_ID
   ```

   For example,

   ```
   [root@host01 ~]# cephadm shell --name osd.0
   ```

3. Get the attributes of the object.

   ```
   ceph-objectstore-tool --data-path PATH_TO_OSD \
   --pgid PG_ID OBJECT \
   get-attr KEY > OBJECT_ATTRS_FILE_NAME
   ```

For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c
'{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
amespace":""}' \
get-attr "oid" > zone_info.default.attr.txt
```

4. Set the attributes of the object.

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT \
set-attr KEY > OBJECT_ATTRS_FILE_NAME
```

For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c
'{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
amespace":""}' \
set-attr "oid" > zone_info.default.attr.txt
```

5. Remove the attributes of the object.

```
ceph-objectstore-tool --data-path PATH_TO_OSD \
--pgid PG_ID OBJECT  \
rm-attr KEY
```

For example,

```
[ceph: root@host01 /]# ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0 \
--pgid 0.1c
'{"oid":"zone_info.default","key":"","snapid":-2,"hash":235010478,"max":0,"pool":11,"n
amespace":""}' \
rm-attr "oid"
```

Troubleshooting clusters in stretch mode

You can replace and remove the failed tiebreaker monitors. You can also force the cluster into the recovery or healthy mode if needed.

Forcing stretch cluster into recovery or healthy mode

When in stretch degraded mode, the cluster goes into the recovery mode automatically after the disconnected data center comes back. If that does not happen, or you want to enable recovery mode early, you can force the stretch cluster into the recovery mode.

PREREQUISITE

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

- A running Red Hat Ceph Storage cluster.
- Stretch mode in enabled on a cluster.

1. Force the stretch cluster into the recovery mode.

> **Note:** The recovery state puts the cluster in the *HEALTH_WARN* state.

For example,

```
[ceph: root@host01 /]#  ceph osd force_recovery_stretch_mode --yes-i-really-mean-it
```

When in recovery mode, the cluster should go back into normal stretch mode after the placement groups are healthy. In cases that the cluster does not go back into normal stretch mode you can force the stretch

cluster into the healthy mode. to force the stretch cluster into the healthy mode, go to step "2" on page 76.

2. If needed, force the stretch cluster into the healthy mode.

> **Note:**
> - You can also run this command if you want to force the cross-data-center peering early and you are willing to risk data downtime, or you have verified separately that all the placement groups can peer, even if they are not fully recovered. You might also want to start the healthy mode to remove the `HEALTH_WARN` state, which is generated by the recovery state.
> - The **`force_recovery_stretch_mode`** and **`force_recovery_healthy_mode`** commands are not necessary, as they are included in the process of managing unanticipated situations.

```
ceph osd force_healthy_stretch_mode --yes-i-really-mean-it
```

Troubleshooting scrub and deep-scrub issues

Learn to troubleshoot the scrub and deep-scrub issues.

Addressing scrub slowness issue after upgrading from Red Hat Ceph Storage 6 to Red Hat Ceph Storage 8 and Red Hat Ceph Storage 7 to Red Hat Ceph Storage 8

Learn to troubleshoot the scrub slowness issue which seen after upgrading from Red Hat Ceph Storage 6 to Red Hat Ceph Storage 8 and Red Hat Ceph Storage 7 to Red Hat Ceph Storage 8.

PREREQUISITE

- A running Red Hat Ceph Storage cluster in a healthy state.
- Root-level access to the node.

Scrub slowness is caused by the automated OSD benchmark setting a very low value for **`osd_mclock_max_capacity_iops_hdd`**. Due to this, scrub operations are impacted since the IOPS capacity of an OSD plays a significant role in determining the bandwidth the scrub operation receives. To further increase the problem, scrubs receive only a fraction of the total IOPS capacity based on the QoS allocation defined by the **mClock** profile.

Due to this, the Ceph cluster reports the expected scrub completion time in multiples of days or weeks.

- Detect low measured IOPS reported by OSD bench during OSD boot-up and fallback to default IOPS setting defined for **`osd_mclock_max_capacity_iops_[hdd|ssd]`**. The fallback is triggered if the reported IOPS falls below a threshold determined by **`osd_mclock_iops_capacity_low_threshold_[hdd|ssd]`**. A cluster warning is also logged. Example:

```
$ ceph config rm osd.X osd_mclock_max_capacity_iops_[hdd|ssd]
```

  - Perform the following steps if you have not yet upgraded to Red Hat Ceph Storage 7 from Red Hat Ceph Storage 6 (before the upgrade):

  a. For clusters already affected by the issue, remove the IOPS capacity setting on the OSD(s) before upgrading to the release with the fix by running the following command:
  For example:

```
$ ceph config rm osd.X osd_mclock_max_capacity_iops_[hdd|ssd]
```

  b. Set the **`osd_mclock_force_run_benchmark_on_init`** option for the affected OSD to `true` before the upgrade:

For example:

```
$ ceph config set osd.X osd_mclock_force_run_benchmark_on_init true
```

After upgrading to the release with this fix, the IOPS capacity reflects the default setting or the new one reported by the OSD bench.

– Perform the following steps:

a. If you were unable to perform the above steps before upgrade, you re-run the OSD bench again after upgrading by removing the **osd_mclock_max_capacity_iops_[hdd|ssd]** setting:
Example:

```
$ ceph config rm osd.X osd_mclock_max_capacity_iops_[hdd|ssd]
```

b. Set **osd_mclock_force_run_benchmark_on_init**to `true`.
Example:

```
$ ceph config set osd.X osd_mclock_force_run_benchmark_on_init true
```

c. Restart the OSD.
After the OSD restarts, the IOPS capacity reflects the default setting or the new setting reported by the OSD bench.

Ceph subsystems default logging level values

Ceph subsystems default log and memory levels.

"Ceph subsystems default logging level values" on page 77 details default logging level values for the various Ceph subsystems.

Ceph subsystems default logging level values

Subsystem	Log Level	Memory Level
asok	1	5
auth	1	5
buffer	0	0
client	0	5
context	0	5
crush	1	5
default	0	5
filer	0	5
bluestore	1	5
finisher	1	5
heartbeatmap	1	5
javaclient	1	5
journaler	0	5
journal	1	5
lockdep	0	5

Subsystem	Log Level	Memory Level
mds balancer	1	5
mds locker	1	5
mds log expire	1	5
mds log	1	5
mds migrator	1	5
mds	1	5
monc	0	5
mon	1	5
ms	0	5
objclass	0	5
objectcacher	0	5
objecter	0	0
optracker	0	5
osd	0	5
paxos	0	5
perfcounter	1	5
rados	0	5
rbd	0	5
rgw	1	5
throttle	1	5
timer	0	5
tp	0	5

Troubleshooting upgrade error messages

Use this information to help troubleshoot and understand common `cephadm` upgrade error messages.

"Troubleshooting upgrade error messages" on page 78 shows some `cephadm` upgrade error messages. If the `cephadm` upgrade fails for any reason, an error message appears in the storage cluster health status.

Troubleshooting upgrade error messages

Error Message	Description
UPGRADE_NO_STANDBY_MGR	Ceph requires both active and standby manager daemons to proceed, but there is currently no standby.

Error Message	Description
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.

Health messages of a Ceph cluster

Each health check has a unique identifier. The identifier is a terse pseudo-human-readable string that is intended to enable tools to make sense of health checks, and present them in a way that reflects their meaning.

These tables describe each of the different types of health messages of a Ceph cluster:

- Monitor
- Manager
- OSDs
- Device health
- Pools and placements groups
- Miscellaneous

Monitor	
Health code	**Description**
DAEMON_OLD_VERSION	Warn if old versions of Ceph are running on any daemons. If multiple versions are detected, a health error is generated.
MON_DOWN	One or more Ceph Monitor daemons are currently down.
MON_CLOCK_SKEW	The clocks on the nodes running the `ceph-mon` daemons are not sufficiently synchronized. Resolve it by synchronizing the clocks by using `ntpd` or `chrony`.
MON_NETSPLIT	Monitor nodes are partitioned and cannot communicate, which can affect quorum stability.
MON_MSGR2_NOT_ENABLED	The `ms_bind_msgr2` option is enabled but one or more Ceph Monitors is not configured to bind to a v2 port in the cluster's monmap. Resolve this by running **`ceph mon enable-msgr2`** command.
MON_DISK_LOW	One or more Ceph Monitors are low on disk space.
MON_DISK_CRIT	One or more Ceph Monitors are critically low on disk space.
MON_DISK_BIG	The database size for one or more Ceph Monitors are very large.
AUTH_INSECURE_GLOBAL_ID_RECLAIM	One or more clients or daemons are connected to the storage cluster that are not securely reclaiming their `global_id` when reconnecting to a Ceph Monitor.
AUTH_INSECURE_GLOBAL_ID_RECLAIM_ALLOWED	Ceph is configured to allow clients to reconnect to monitors by using an insecure process to reclaim their previous `global_id` because the setting **`auth_allow_insecure_global_id_reclaim`** is set to `true`.

Manager	
Health code	**Description**
MGR_DOWN	All Ceph Manager daemons are down.
MGR_MODULE_DEPENDENCY	An enabled Ceph Manager module is failing its dependency check.
MGR_MODULE_ERROR	A Ceph Manager module encountered an unexpected error. Typically, this means that an unhandled exception was raised from the module's serve function.

OSDs	
Health code	**Description**
OSD_DOWN	One or more OSDs are marked down.
OSD_CRUSH_TYPE_DOWN	All the OSDs within a particular CRUSH subtree are marked down, for example all OSDs on a host. For example, `OSD_HOST_DOWN` and `OSD_ROOT_DOWN`
OSD_ORPHAN	An OSD is referenced in the CRUSH map hierarchy but does not exist. Remove the OSD by running **ceph osd crush rm osd._OSD_ID** command.
OSD_OUT_OF_ORDER_FULL	The utilization thresholds for `nearfull`, `backfillfull`, `full`, or, `failsafefull` are not ascending. Adjust the thresholds by running **ceph osd set-nearfull-ratio RATIO**, **ceph osd set-backfillfull-ratio RATIO**, and **ceph osd set-full-ratio RATIO** commands.
OSD_FULL	One or more OSDs exceeded the full threshold and is preventing the storage cluster from servicing writes. Restore write availability by raising the full threshold by a small margin **ceph osd set-full-ratio RATIO**.
OSD_BACKFILLFULL	One or more OSDs exceeded the backfillfull threshold, which prevents data from being allowed to rebalance to this device.
OSD_NEARFULL	One or more OSDs exceeded the nearfull threshold.
OSDMAP_FLAGS	One or more storage cluster flags of interest has been set. These flags include`full`,`pauserd`, `pausewr`,`noup`,`nodown`,`noin`,`noout`,`nobackfill`, `norecover`,`norebalance`,`noscrub`, `nodeep_scrub`, and`notieragent`. Except for`full`, the flags can be cleared with the **ceph osd set FLAG** and **ceph osd unset FLAG** commands.
OSD_FLAGS	One or more OSDs or CRUSH has a flag of interest set. These flags includenoup,nodown,`noin`, and `noout`.
OLD_CRUSH_TUNABLES	The CRUSH map is using old settings and should be updated.
OLD_CRUSH_STRAW_CALC_VERSION	The CRUSH map is using an older, non-optimal method for calculating intermediate weight values for `straw` buckets.

Health code	Description
CACHE_POOL_NO_HIT_SET	One or more cache pools are not configured with a hit set to track usage, which prevents the tiering agent from identifying cold objects to flush and evict from the cache. Configure the hit sets on the cache pool with the following commands: – **ceph osd pool set POOL_NAME hit_set_type TYPE** – **ceph osd pool set POOL_NAME hit_set_period PERIOD_IN_SECONDS** – **ceph osd pool set POOL_NAME hit_set_count NUMBER_OF_HIT_SETS** – **ceph osd pool set POOL_NAME hit_set_fpp TARGET_FALSE_POSITIVE_RATE**
OSD_NO_SORTBITWISE	`sortbitwise` flag is not set. Set the flag with **ceph osd set sortbitwise** command.
POOL_FULL	One or more pools reached their quota and is no longer allowing writes. Increase the pool quota with the **ceph osd pool set-quota POOL_NAME max_objects NUMBER_OF_OBJECTS** and **ceph osd pool set-quota POOL_NAME max_bytes BYTES** commands or delete some existing data to reduce utilization.
BLUEFS_SPILLOVER	One or more OSDs that use the BlueStore backend is allocated db partitions but that space is filled, such that metadata has “spilled over” onto the normal slow device. Disable this with the **ceph config set osd bluestore_warn_on_bluefs_spillover false** command.
BLUEFS_AVAILABLE_SPACE	This output gives three values that are `BDEV_DB free`, `BDEV_SLOW free`, and `available_from_bluestore`.
BLUEFS_LOW_SPACE	If the BlueStore File System (BlueFS) is running low on available free space and there is little `available_from_bluestore` one can consider reducing BlueFS allocation unit size.
BLUESTORE_FRAGMENTATION	As BlueStore works, free space on underlying storage gets fragmented. This is normal and unavoidable but excessive fragmentation causes slowdown. Use the `BLUESTORE_WARN_ON_FREE_FRAGMENTATION` and `BLUESTORE_FRAGMENTATION_CHECK_PERIOD` parameters to control and configure the `BLUESTORE_FRAGMENTATION` health warning. The default value for `BLUESTORE_WARN_ON_FREE_FRAGMENTATION` is 0.8 and the default for `BLUESTORE_FRAGMENTATION_CHECK_PERIOD` is 3600 milliseconds.
BLUESTORE_LEGACY_STATFS	BlueStore tracks its internal usage statistics on a per-pool granular basis, and one or more OSDs have BlueStore volumes. Disable the warning with the **ceph config set global bluestore_warn_on_legacy_statfs false** command.

Health code	Description
BLUESTORE_NO_PER_POOL_OMAP	BlueStore tracks omap space utilization by pool. Disable the warning with the **`ceph config set global bluestore_warn_on_no_per_pool_omap false`** command.
BLUESTORE_NO_PER_PG_OMAP	BlueStore tracks omap space utilization by PG. Disable the warning with **`ceph config set global bluestore_warn_on_no_per_pg_omap false`** command.
BLUESTORE_DISK_SIZE_MISMATCH	One or more OSDs using BlueStore has an internal inconsistency between the size of the physical device and the metadata tracking its size.
BLUESTORE_NO_COMPRESSION	One or more OSDs is unable to load a BlueStore compression plug-in. This can be caused by a broken installation, in which the `ceph-osd` binary does not match the compression plug-ins, or a recent upgrade that did not include a restart of the `ceph-osd` daemon.
BLUESTORE_SPURIOUS_READ_ERRORS	One or more OSDs using BlueStore detects spurious read errors on the main device. BlueStore recovers from these errors by retrying disk reads.

Device health	
Health code	**Description**
DEVICE_HEALTH	One or more devices is expected to fail soon, where the warning threshold is controlled by the `mgr/devicehealth/warn_threshold` config option. Mark the device `out` to migrate the data and replace the hardware.
DEVICE_HEALTH_IN_USE	One or more devices is expected to fail soon and has been marked `out` of the storage cluster based on `mgr/devicehealth/mark_out_threshold`, but it is still participating in one more PGs.
DEVICE_HEALTH_TOOMANY	Too many devices are expected to fail soon and the `mgr/devicehealth/self_heal` behavior is enabled, such that marking out all of the ailing devices would exceed the clusters `mon_osd_min_in_ratio` ratio that prevents too many OSDs from being automatically marked `out`.

Pools and placement groups	
Health code	**Description**
PG_AVAILABILITY	Data availability is reduced, meaning that the storage cluster is unable to service potential read or write requests for some data in the cluster.
PG_DEGRADED	Data redundancy is reduced for some data, meaning the storage cluster does not have the desired number of replicas for replicated pools or erasure code fragments.
PG_RECOVERY_FULL	Data redundancy might be reduced or at risk for some data due to a lack of free space in the storage cluster, specifically, one or more PGs has the `recovery_toofull` flag set, which means that the cluster is unable to migrate or recover data because one or more OSDs is over the `full` threshold.

Health code	Description
PG_BACKFILL_FULL	Data redundancy might be reduced or at risk for some data due to a lack of free space in the storage cluster, specifically, one or more PGs has the `backfill_toofull` flag set, which means that the cluster is unable to migrate or recover data because one or more OSDs is over the `backfillfull` threshold.
PG_DAMAGED	Data scrubbing has discovered some problems with data consistency in the storage cluster, specifically, one or more PGs has the inconsistent or `snaptrim_error` flag is set, indicating an earlier scrub operation found a problem, or that the `repair` flag is set, meaning a repair for such an inconsistency is currently in progress.
OSD_SCRUB_ERRORS	Recent OSD scrubs have uncovered inconsistencies.
OSD_TOO_MANY_REPAIRS	When a read error occurs and another replica is available it is used to repair the error immediately, so that the client can get the object data.
LARGE_OMAP_OBJECTS	One or more pools contain large omap objects as determined by `osd_deep_scrub_large_omap_object_key_threshold` or `osd_deep_scrub_large_omap_object_value_sum_threshold` or both. Adjust the thresholds with the **`ceph config set osd osd_deep_scrub_large_omap_object_key_threshold KEYS`** and **`ceph config set osd osd_deep_scrub_large_omap_object_value_sum_threshold BYTES`** commands.
CACHE_POOL_NEAR_FULL	A cache tier pool is nearly full. Adjust the cache pool target size with the **`ceph osd pool set CACHE_POOL_NAME target_max_bytes BYTES`** and **`ceph osd pool set CACHE_POOL_NAME target_max_bytes BYTES`** commands.
TOO_FEW_PGS	The number of PGs in use in the storage cluster is less than the configurable threshold of `mon_pg_warn_min_per_osd` PGs per OSD.
POOL_PG_NUM_NOT_POWER_OF_TWO	One or more pools has a `pg_num` value that is not a power of two. Disable the warning with the **`ceph config set global mon_warn_on_pool_pg_num_not_power_of_two false`** command.
POOL_TOO_FEW_PGS	One or more pools should probably have more PGs, based on the amount of data that is currently stored in the pool. You can either disable auto-scaling of PGs with the **`ceph osd pool set POOL_NAME pg_autoscale_mode off`** command, automatically adjust the number of PGs with the **`ceph osd pool set POOL_NAME pg_autoscale_mode on`** command or manually set the number of PGs with **`ceph osd pool set POOL_NAME pg_num NEW_PG_NUMBER`** command.
TOO_MANY_PGS	The number of PGs in use in the storage cluster is over the configurable threshold of `mon_max_pg_per_osd` PGs per OSD. Increase the number of OSDs in the cluster by adding more hardware.

Health code	Description
POOL_TOO_MANY_PGS	One or more pools should probably have more PGs, based on the amount of data that is currently stored in the pool. You can either disable auto-scaling of PGs with the **ceph osd pool set POOL_NAME pg_autoscale_mode off** command, automatically adjust the number of PGs with the **ceph osd pool set POOL_NAME pg_autoscale_mode on** command or manually set the number of PGs with the **ceph osd pool set POOL_NAME pg_num NEW_PG_NUMBER** command.
POOL_TARGET_SIZE_BYTES_OVERCOMMITTED	One or more pools have a `target_size_bytes` property set to estimate the expected size of the pool, but the values exceed the total available storage. Set the value for the pool to zero with the **ceph osd pool set POOL_NAME target_size_bytes 0** command.
POOL_HAS_TARGET_SIZE_BYTES_AND_RATIO	One or more pools have both `target_size_bytes` and `target_size_ratio` set to estimate the expected size of the pool. Set the value for the pool to zero with `ceph osd pool set POOL_NAME target_size_bytes 0` command.
TOO_FEW_OSDS	The number of OSDs in the storage cluster is less than the configurable threshold of `osd_pool_default_size`.
SMALLER_PGP_NUM	One or more pools has a `pgp_num` value less than `pg_num`. This is normally an indication that the PG count was increased without also increasing the placement behavior. Resolve this by setting `pgp_num` to match with `pg_num` with the **ceph osd pool set POOL_NAME pgp_num PG_NUM_VALUE** command.
MANY_OBJECTS_PER_PG	One or more pools has an average number of objects per PG that is significantly higher than the overall storage cluster average. The specific threshold is controlled by the `mon_pg_warn_max_object_skew` configuration value.
POOL_APP_NOT_ENABLED	A pool exists that contains one or more objects but has not been tagged for use by a particular application. Resolve this warning by labeling the pool for use by an application with **rbd pool init POOL_NAME** command.
POOL_FULL	One or more pools has reached its quota. The threshold to trigger this error condition is controlled by the **mon_pool_quota_crit_threshold** configuration option.
POOL_NEAR_FULL	One or more pools is approaching a configured fullness threshold. Adjust the pool quotas with the **ceph osd pool set-quota POOL_NAME max_objects NUMBER_OF_OBJECTS** and **ceph osd pool set-quota POOL_NAME max_bytes BYTES** commands.
OBJECT_MISPLACED	One or more objects in the storage cluster is not stored on the node that the storage cluster should be stored on. This is an indication that data migration due to some recent storage cluster change has not yet completed.

Health code	Description
OBJECT_UNFOUND	One or more objects in the storage cluster cannot be found, specifically, the OSDs know that a new or updated copy of an object should exist, but a copy of that version of the object has not been found on OSDs that are currently online.
SLOW_OPS	One or more OSD or monitor requests is taking a long time to process. This can be an indication of extreme load, a slow storage device, or a software bug.
PG_NOT_SCRUBBED	One or more PGs have not been scrubbed recently. PGs are normally scrubbed within every configured interval that is specified by `osd_scrub_max_interval` globally. Initiate the scrub with the **`ceph pg scrub PG_ID`** command.
PG_NOT_DEEP_SCRUBBED	One or more PGs have not been deep scrubbed recently. Initiate the scrub with **`ceph pg deep-scrub PG_ID`** command. PGs are normally scrubbed every `osd_deep_scrub_interval` seconds, and this warning triggers when `mon_warn_pg_not_deep_scrubbed_ratio` percentage of interval has elapsed without a scrub since it was due.
PG_SLOW_SNAP_TRIMMING	The snapshot trim queue for one or more PGs has exceeded the configured warning threshold. This indicates that either an extremely large number of snapshots were recently deleted, or that the OSDs are unable to trim snapshots quickly enough to keep up with the rate of new snapshot deletions.

Miscellaneous	
Health code	**Description**
RECENT_CRASH	One or more Ceph daemons has crashed recently, and the crash has not yet been acknowledged by the administrator.
TELEMETRY_CHANGED	Telemetry has been enabled, but the contents of the telemetry report have changed since that time, so telemetry reports will not be sent.
AUTH_BAD_CAPS	One or more authorized users have capabilities that cannot be parsed by the monitor. Update the capabilities of the user with **`ceph auth ENTITY_NAME DAEMON_TYPE CAPS`** command.
OSD_NO_DOWN_OUT_INTERVAL	The `mon_osd_down_out_interval` option is set to zero, which means that the system will not automatically run any repair or healing operations after an OSD fails. Silence the interval with the **`ceph config global mon mon_warn_on_osd_down_out_interval_zero false`** command.
DASHBOARD_DEBUG	The Dashboard debug mode is enabled. This means, if there is an error while processing a REST API request, the HTTP error response contains a Python traceback. Disable the debug mode with the **`ceph dashboard debug disable`** command.