MONITORING A CLUSTER

Once you have a running cluster, you may use the ceph tool to monitor your cluster. Monitoring a cluster typically involves checking OSD status, monitor status, placement group status and metadata server status.

USING THE COMMAND LINE

INTERACTIVE MODE

To run the ceph tool in interactive mode, type ceph at the command line with no arguments. For example:

```
ceph
ceph> health
ceph> status
ceph> quorum_status
ceph> mon_status
```

NON-DEFAULT PATHS

If you specified non-default locations for your configuration or keyring, you may specify their locations:

```
ceph -c /path/to/conf -k /path/to/keyring health
```

CHECKING A CLUSTER'S STATUS

After you start your cluster, and before you start reading and/or writing data, check your cluster's status first.

To check a cluster's status, execute the following:

```
ceph status
```

Or:

```
ceph -s
```

In interactive mode, type `status` and press **Enter**.

```
ceph> status
```

Ceph will print the cluster status. For example, a tiny Ceph demonstration cluster with one of each service may print the following:

```
cluster:
  id:     477e46f1-ae41-4e43-9c8f-72c918ab0a20
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum a,b,c
  mgr: x(active)
  mds: cephfs_a-1/1/1 up  {0=a=up:active}, 2 up:standby
  osd: 3 osds: 3 up, 3 in

data:
  pools:   2 pools, 16 pgs
  objects: 21 objects, 2.19K
```

```
  usage:   546 GB used, 384 GB / 931 GB avail
  pgs:     16 active+clean
```

How Ceph Calculates Data Usage

The usage value reflects the *actual* amount of raw storage used. The `xxx GB / xxx GB` value means the amount available (the lesser number) of the overall storage capacity of the cluster. The notional number reflects the size of the stored data before it is replicated, cloned or snapshotted. Therefore, the amount of data actually stored typically exceeds the notional amount stored, because Ceph creates replicas of the data and may also use storage capacity for cloning and snapshotting.

WATCHING A CLUSTER

In addition to local logging by each daemon, Ceph clusters maintain a *cluster log* that records high level events about the whole system. This is logged to disk on monitor servers (as `/var/log/ceph/ceph.log` by default), but can also be monitored via the command line.

To follow the cluster log, use the following command

```
ceph -w
```

Ceph will print the status of the system, followed by each log message as it is emitted. For example:

```
cluster:
  id:     477e46f1-ae41-4e43-9c8f-72c918ab0a20
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum a,b,c
  mgr: x(active)
  mds: cephfs_a-1/1/1 up  {0=a=up:active}, 2 up:standby
  osd: 3 osds: 3 up, 3 in

data:
  pools:   2 pools, 16 pgs
  objects: 21 objects, 2.19K
  usage:   546 GB used, 384 GB / 931 GB avail
  pgs:     16 active+clean


2017-07-24 08:15:11.329298 mon.a mon.0 172.21.9.34:6789/0 23 : cluster [INF] osd.0 172.21.9.3
2017-07-24 08:15:14.258143 mon.a mon.0 172.21.9.34:6789/0 39 : cluster [INF] Activating manag
2017-07-24 08:15:15.446025 mon.a mon.0 172.21.9.34:6789/0 47 : cluster [INF] Manager daemon x
```

In addition to using `ceph -w` to print log lines as they are emitted, use `ceph log last [n]` to see the most recent n lines from the cluster log.

MONITORING HEALTH CHECKS

Ceph continously runs various *health checks* against its own status. When a health check fails, this is reflected in the output of `ceph status` (or `ceph health`). In addition, messages are sent to the cluster log to indicate when a check fails, and when the cluster recovers.

For example, when an OSD goes down, the `health` section of the status output may be updated as follows:

```
health: HEALTH_WARN
        1 osds down
        Degraded data redundancy: 21/63 objects degraded (33.333%), 16 pgs unclean, 16 pgs de
```

At this time, cluster log messages are also emitted to record the failure of the health checks:

```
2017-07-25 10:08:58.265945 mon.a mon.0 172.21.9.34:6789/0 91 : cluster [WRN] Health check fai
2017-07-25 10:09:01.302624 mon.a mon.0 172.21.9.34:6789/0 94 : cluster [WRN] Health check fai
```

When the OSD comes back online, the cluster log records the cluster's return to a health state:

```
2017-07-25 10:11:11.526841 mon.a mon.0 172.21.9.34:6789/0 109 : cluster [WRN] Health check up
2017-07-25 10:11:13.535493 mon.a mon.0 172.21.9.34:6789/0 110 : cluster [INF] Health check cl
2017-07-25 10:11:13.535577 mon.a mon.0 172.21.9.34:6789/0 111 : cluster [INF] Cluster is now
```

DETECTING CONFIGURATION ISSUES

In addition to the health checks that Ceph continuously runs on its own status, there are some configuration issues that may only be detected by an external tool.

Use the ceph-medic tool to run these additional checks on your Ceph cluster's configuration.

CHECKING A CLUSTER'S USAGE STATS

To check a cluster's data usage and data distribution among pools, you can use the `df` option. It is similar to Linux `df`. Execute the following:

```
ceph df
```

The **GLOBAL** section of the output provides an overview of the amount of storage your cluster uses for your data.

- **SIZE:** The overall storage capacity of the cluster.
- **AVAIL:** The amount of free space available in the cluster.
- **RAW USED:** The amount of raw storage used.
- **% RAW USED:** The percentage of raw storage used. Use this number in conjunction with the `full ratio` and `near full ratio` to ensure that you are not reaching your cluster's capacity. See Storage Capacity for additional details.

The **POOLS** section of the output provides a list of pools and the notional usage of each pool. The output from this section **DOES NOT** reflect replicas, clones or snapshots. For example, if you store an object with 1MB of data, the notional usage will be 1MB, but the actual usage may be 2MB or more depending on the number of replicas, clones and snapshots.

- **NAME:** The name of the pool.
- **ID:** The pool ID.
- **USED:** The notional amount of data stored in kilobytes, unless the number appends **M** for megabytes or **G** for gigabytes.
- **%USED:** The notional percentage of storage used per pool.
- **MAX AVAIL:** An estimate of the notional amount of data that can be written to this pool.
- **OBJECTS:** The notional number of objects stored per pool.

Note: The numbers in the **POOLS** section are notional. They are not inclusive of the number of replicas, shapshots or clones. As a result, the sum of the **USED** and **%USED** amounts will not add up to the **RAW USED** and **%RAW USED** amounts in the **GLOBAL** section of the output.

Note: The **MAX AVAIL** value is a complicated function of the replication or erasure code used, the CRUSH rule that maps storage to devices, the utilization of those devices, and the configured mon_osd_full_ratio.

CHECKING OSD STATUS

You can check OSDs to ensure they are `up` and `in` by executing:

```
ceph osd stat
```

Or:

```
ceph osd dump
```

You can also check view OSDs according to their position in the CRUSH map.

```
ceph osd tree
```

Ceph will print out a CRUSH tree with a host, its OSDs, whether they are up and their weight.

```
#ID CLASS WEIGHT  TYPE NAME             STATUS REWEIGHT PRI-AFF
 -1       3.00000 pool default
 -3       3.00000 rack mainrack
 -2       3.00000 host osd-host
  0   ssd 1.00000         osd.0             up  1.00000 1.00000
  1   ssd 1.00000         osd.1             up  1.00000 1.00000
  2   ssd 1.00000         osd.2             up  1.00000 1.00000
```

For a detailed discussion, refer to Monitoring OSDs and Placement Groups.

CHECKING MONITOR STATUS

If your cluster has multiple monitors (likely), you should check the monitor quorum status after you start the cluster and before reading and/or writing data. A quorum must be present when multiple monitors are running. You should also check monitor status periodically to ensure that they are running.

To see display the monitor map, execute the following:

```
ceph mon stat
```

Or:

```
ceph mon dump
```

To check the quorum status for the monitor cluster, execute the following:

```
ceph quorum_status
```

Ceph will return the quorum status. For example, a Ceph cluster consisting of three monitors may return the following:

```
{ "election_epoch": 10,
  "quorum": [
        0,
        1,
        2],
  "quorum_names": [
        "a",
        "b",
        "c"],
  "quorum_leader_name": "a",
  "monmap": { "epoch": 1,
      "fsid": "444b489c-4f16-4b75-83f0-cb8097468898",
      "modified": "2011-12-12 13:28:27.505520",
      "created": "2011-12-12 13:28:27.505520",
      "features": {"persistent": [
                        "kraken",
                        "luminous",
                        "mimic"],
        "optional": []
      },
      "mons": [
            { "rank": 0,
              "name": "a",
              "addr": "127.0.0.1:6789/0",
```

```
            "public_addr": "127.0.0.1:6789/0"},
          { "rank": 1,
            "name": "b",
            "addr": "127.0.0.1:6790/0",
            "public_addr": "127.0.0.1:6790/0"},
          { "rank": 2,
            "name": "c",
            "addr": "127.0.0.1:6791/0",
            "public_addr": "127.0.0.1:6791/0"}
        ]
  }
}
```

CHECKING MDS STATUS

Metadata servers provide metadata services for Ceph FS. Metadata servers have two sets of states: `up | down` and `active | inactive`. To ensure your metadata servers are up and `active`, execute the following:

```
ceph mds stat
```

To display details of the metadata cluster, execute the following:

```
ceph fs dump
```

CHECKING PLACEMENT GROUP STATES

Placement groups map objects to OSDs. When you monitor your placement groups, you will want them to be `active` and `clean`. For a detailed discussion, refer to Monitoring OSDs and Placement Groups.

USING THE ADMIN SOCKET

The Ceph admin socket allows you to query a daemon via a socket interface. By default, Ceph sockets reside under `/var/run/ceph`. To access a daemon via the admin socket, login to the host running the daemon and use the following command:

```
ceph daemon {daemon-name}
ceph daemon {path-to-socket-file}
```

For example, the following are equivalent:

```
ceph daemon osd.0 foo
ceph daemon /var/run/ceph/ceph-osd.0.asok foo
```

To view the available admin socket commands, execute the following command:

```
ceph daemon {daemon-name} help
```

The admin socket command enables you to show and set your configuration at runtime. See Viewing a Configuration at Runtime for details.

Additionally, you can set configuration values at runtime directly (i.e., the admin socket bypasses the monitor, unlike ceph `tell {daemon-type}.{id} config set`, which relies on the monitor but doesn't require you to login directly to the host in question).