

CONFIGURING CEPH

When you start the Ceph service, the initialization process activates a series of daemons that run in the background. All Ceph clusters run two types of daemons:

- Monitor (ceph-mon)
- Object Storage Device (ceph-osd)

Clusters that support CephFS run Metadata Servers (ceph-mds). Clusters that support a RESTful object store run Ceph Gateways (radosgw). For your convenience, each daemon has a series of default values (*i.e.*, many are set by `ceph/src/common/config_opts.h`). You may override these settings with a Ceph configuration file.

THE CONFIGURATION FILE

When you start a Ceph cluster, each daemon looks for a Ceph configuration file (*i.e.*, `ceph.conf` by default) that provides the cluster's configuration settings. For manual deployments, you need to create a Ceph configuration file. For tools that create configuration files for you (*e.g.*, `ceph-deploy`, Chef, etc.), you may use the information contained herein as a reference. The Ceph Configuration file defines:

- Cluster Identity
- Authentication settings
- Cluster membership
- Host names
- Host addresses
- Paths to keyrings
- Paths to journals
- Paths to data
- Other runtime options

The default `ceph.conf` locations in sequential order include:

1. `$CEPH_CONF` (*i.e.*, the path following the `$CEPH_CONF` environment variable)
2. `-c path/path` (*i.e.*, the `-c` command line argument)
3. `/etc/ceph/ceph.conf`
4. `~/.ceph/config`
5. `./ceph.conf` (*i.e.*, in the current working directory)

The Ceph configuration file uses an *ini* style syntax. You can add comments by preceding comments with a pound sign (`#`) or a semi-colon (`;`). For example:

```
# <--A number (#) sign precedes a comment.  
; A comment may be anything.  
# Comments always follow a semi-colon (;) or a pound (#) on each line.  
# The end of the line terminates a comment.  
# We recommend that you provide comments in your configuration file(s).
```

CONFIG SECTIONS

The configuration file can configure all daemons in a cluster, or all daemons of a particular type. To configure a series of daemons, the settings must be included under the processes that will receive the configuration as follows:

[global]

Description: Settings under [global] affect all daemons in a Ceph cluster.

Example: auth supported = cephx

[osd]

Description: Settings under [osd] affect all ceph-osd daemons in the cluster, and override the same setting in [global].

Example: osd journal size = 1000

[mon]

Description: Settings under [mon] affect all ceph-mon daemons in the cluster, and override the same setting in [global].

Example: mon addr = 10.0.0.101:6789

[mds]

Description: Settings under [mds] affect all ceph-mds daemons in the cluster, and override the same setting in [global].

Example: host = myserver01

[client]

Description: Settings under [client] affect all clients (e.g., mounted CephFS filesystems, mounted block devices, etc.).

Example: log file = /var/log/ceph/radosgw.log

Global settings affect all instances of all daemon in the cluster. Use the [global] setting for values that are common for all daemons in the cluster. You can override each [global] setting by:

1. Changing the setting in a particular process type (e.g., [osd], [mon], [mds]).
2. Changing the setting in a particular process (e.g., [osd.1]).

Overriding a global setting affects all child processes, except those that you specifically override in a particular daemon.

A typical global setting involves activating authentication. For example:

```
[global]
#Enable authentication between hosts within the cluster.
#v 0.54 and earlier
auth supported = cephx

#v 0.55 and after
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
```

You can specify settings that apply to a particular type of daemon. When you specify settings under [osd], [mon] or [mds] without specifying a particular instance, the setting will apply to all OSDs, monitors or metadata daemons respectively.

A typical daemon-wide setting involves setting journal sizes, filestore settings, etc. For example:

```
[osd]
osd journal size = 1000
filestore xattr use omap = true
```

You may specify settings for particular instances of a daemon. You may specify an instance by entering its type, delimited by a period (.) and by the instance ID. The instance ID for an OSD is always numeric, but it may be alphanumeric for monitors and metadata servers.

```
[osd.1]
# settings affect osd.1 only.

[mon.a]
# settings affect mon.a only.

[mds.b]
# settings affect mds.b only.
```

METAVARIABLES

Metavariables simplify cluster configuration dramatically. When a metavariable is set in a configuration value, Ceph expands the metavariable into a concrete value. Metavariables are very powerful when used within the [global], [osd], [mon] or

[`mds`] sections of your configuration file. Ceph metavariables are similar to Bash shell expansion.

Ceph supports the following metavariables:

`$cluster`

Description: Expands to the cluster name. Useful when running multiple clusters on the same hardware.
Example: `/etc/ceph/$cluster.keyring`
Default: `ceph`

`$type`

Description: Expands to one of `mds`, `osd`, or `mon`, depending on the type of the current daemon.
Example: `/var/lib/ceph/$type`

`$id`

Description: Expands to the daemon identifier. For `osd.0`, this would be `0`; for `mds.a`, it would be `a`.
Example: `/var/lib/ceph/$type/$cluster-$id`

`$host`

Description: Expands to the host name of the current daemon.

`$name`

Description: Expands to `$type.$id`.
Example: `/var/run/ceph/$cluster-$name.asok`

COMMON SETTINGS

The [Hardware Recommendations](#) section provides some hardware guidelines for configuring the cluster. It is possible for a single host to run multiple daemons. For example, a single host with multiple disks or RAID's may run one `ceph-osd` for each disk or RAID. Ideally, you will have a host for a particular type of process. For example, one host may run `ceph-osd` daemons, another host may run a `ceph-mds` daemon, and other hosts may run `ceph-mon` daemons.

Each host has a name identified by the `host` setting. Monitors also specify a network address and port (i.e., domain name or IP address) identified by the `addr` setting. A basic configuration file will typically specify only minimal settings for each instance of a daemon. For example:

```
[mon.a]
    host = hostName
    mon addr = 150.140.130.120:6789

[osd.0]
    host = hostName
```

Important: The `host` setting is the short name of the host (i.e., not an fqdn). It is **NOT** an IP address either. Enter `hostname -s` on the command line to retrieve the name of the host. Also, this setting is **ONLY** for `mkcephfs` and manual deployment. It **MUST NOT** be used with `chef` or `ceph-deploy`, as those tools will enter the appropriate values for you.

NETWORKS

See the [Network Configuration Reference](#) for a detailed discussion about configuring a network for use with Ceph.

MONITORS

Ceph production clusters typically deploy with a minimum 3 monitors to ensure high availability should a monitor instance crash. An odd number of monitors (3) ensures that the Paxos algorithm can determine which version of the cluster map is the most recent from a quorum of monitors.

Note: You may deploy Ceph with a single monitor, but if the instance fails, the lack of a monitor may interrupt data service availability.

Ceph monitors typically listen on port 6789. For example:

```
[mon.a]
host = hostName
mon addr = 150.140.130.120:6789
```

By default, Ceph expects that you will store a monitor's data under the following path:

```
/var/lib/ceph/mon/$cluster-$id
```

You or a deployment tool (e.g., ceph-deploy) must create the corresponding directory. With metavariables fully expressed and a cluster named "ceph", the foregoing directory would evaluate to:

```
/var/lib/ceph/mon/ceph-a
```

For additional details, see the [Monitor Config Reference](#).

AUTHENTICATION

New in version Bobtail: 0.56

For Bobtail (v 0.56) and beyond, you should expressly enable or disable authentication in the [global] section of your Ceph configuration file.

```
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
```

Additionally, you should enable message signing. See [Cephx Config Reference](#) and [Cephx Authentication](#) for details.

Important: When upgrading, we recommend expressly disabling authentication first, then perform the upgrade. Once the upgrade is complete, re-enable authentication.

OSDS

Ceph production clusters typically deploy OSDs where one host has one OSD daemon running a filestore on one data disk. A typical deployment specifies a journal size and whether the file store's extended attributes (XATTRs) use an object map (i.e., when running on the ext4 filesystem). For example:

```
[osd]
osd journal size = 10000
filestore xattr use omap = true #enables the object map. Only if running ext4.

[osd.0]
host = {hostname}
```

By default, Ceph expects that you will store an OSD's data with the following path:

```
/var/lib/ceph/osd/$cluster-$id
```

You or a deployment tool (e.g., ceph-deploy) must create the corresponding directory. With metavariables fully expressed and a cluster named "ceph", the foregoing directory would evaluate to:

```
/var/lib/ceph/osd/ceph-0
```

You may override this path using the `osd_data` setting. We don't recommend changing the default location. Create the default directory on your OSD host.

```
ssh {osd-host}
sudo mkdir /var/lib/ceph/osd/ceph-{osd-number}
```

The `osd_data` path ideally leads to a mount point with a hard disk that is separate from the hard disk storing and running the operating system and daemons. If the OSD is for a disk other than the OS disk, prepare it for use with Ceph, and mount it to the directory you just created:

```
ssh {new-osd-host}
sudo mkfs -t {fstype} /dev/{disk}
sudo mount -o user_xattr /dev/{hdd} /var/lib/ceph/osd/ceph-{osd-number}
```

We recommend using the `xfs` file system or the `btrfs` file system when running `mkfs`.

See the [OSD Config Reference](#) for additional configuration details.

HEARTBEATS

During runtime operations, OSDs check up on other OSDs and report their findings to the monitor. You do not have to provide any settings. However, if you have network latency issues, you may wish to modify the settings.

See [Configuring Monitor/OSD Interaction](#) for additional details.

LOGS / DEBUGGING

Ceph is still on the leading edge, so you may encounter situations that require modifying logging output and using Ceph's debugging. See [Debugging and Logging](#) for details on log rotation.

EXAMPLE CEPH.CONF

```
[global]
# For version 0.54 and earlier, you may enable
# authentication with the following setting.
# Specifying `cephx` enables authentication;
# and specifying `none` disables authentication.

#auth supported = cephx

# For version 0.55 and beyond, you must explicitly enable
# or disable authentication with "auth" entries in [global].

auth cluster required = cephx
auth service required = cephx
auth client required = cephx

[osd]
osd journal size = 1000
# uncomment the following line if you are mounting with ext4
# filestore xattr use omap = true

# For Bobtail (v 0.56) and subsequent versions, you may
# add settings for mkcephfs so that it will create and mount
# the file system for you. Remove the comment `#` character for
# the following settings and replace the values in parenthesis
# with appropriate values, or leave the following settings commented
# out to accept the default values. You must specify the --mkfs
# option with mkcephfs in order for the deployment script to
```

```

# utilize the following settings, and you must define the 'devs'
# option for each osd instance; see below.

#osd mkfs type = {fs-type}
#osd mkfs options {fs-type} = {mkfs options} # default for xfs is "-f"
#osd mount options {fs-type} = {mount options} # default mount option is "rw, noatime

[mon.a]
host = myserver01
mon addr = 10.0.0.101:6789

[mon.b]
host = myserver02
mon addr = 10.0.0.102:6789

[mon.c]
host = myserver03
mon addr = 10.0.0.103:6789

[osd.0]
host = myserver01
#devs = {path-to-device}

[osd.1]
host = myserver02
#devs = {path-to-device}

[osd.2]
host = myserver03
#devs = {path-to-device}

[mds.a]
host = myserver01
#devs = {path-to-device}

```

RUNTIME CHANGES

Ceph allows you to make changes to the configuration of an ceph-osd, ceph-mon, or ceph-mds daemon at runtime. This capability is quite useful for increasing/decreasing logging output, enabling/disabling debug settings, and even for runtime optimization. The following reflects runtime configuration usage:

```
ceph {daemon-type} tell {id or *} injectargs '--{name} {value} [--{name} {value}]'
```

Replace {daemon-type} with one of osd, mon or mds. You may apply the runtime setting to all daemons of a particular type with *, or specify a specific daemon's ID (i.e., its number or letter). For example, to increase debug logging for a ceph-osd daemon named osd.0, execute the following:

```
ceph osd tell 0 injectargs '--debug-osd 20 --debug-ms 1'
```

In your ceph.conf file, you may use spaces when specifying a setting name. When specifying a setting name on the command line, ensure that you use an underscore or hyphen (_ or -) between terms (e.g., debug osd becomes debug-osd).

VIEWING A CONFIGURATION AT RUNTIME

If your Ceph cluster is running, and you would like to see the configuration settings from a running daemon, execute the following:

```
ceph --admin-daemon {/path/to/admin/socket} config show | less
```

The default path for the admin socket for each daemon is:

```
/var/run/ceph/$cluster-$name.asok
```

At real time, the metavariables will evaluate to the actual cluster name and daemon name. For example, if the cluster name is ceph (it is by default) and you want to retrieve the configuration for `osd.0`, use the following:

```
ceph --admin-daemon /var/run/ceph/ceph-osd.0.asok config show | less
```

RUNNING MULTIPLE CLUSTERS

With Ceph, you can run multiple clusters on the same hardware. Running multiple clusters provides a higher level of isolation compared to using different pools on the same cluster with different CRUSH rulesets. A separate cluster will have separate monitor, OSD and metadata server processes. When running Ceph with default settings, the default cluster name is ceph, which means you would save your Ceph configuration file with the file name `ceph.conf` in the `/etc/ceph` default directory.

When you run multiple clusters, you must name your cluster and save the Ceph configuration file with the name of the cluster. For example, a cluster named `openstack` will have a Ceph configuration file with the file name `openstack.conf` in the `/etc/ceph` default directory.

Important: Cluster names must consist of letters a-z and digits 0-9 only.

Separate clusters imply separate data disks and journals, which are not shared between clusters. Referring to [Metavariables](#), the `$cluster` metavariable evaluates to the cluster name (i.e., `openstack` in the foregoing example). Various settings use the `$cluster` metavariable, including:

- keyring
- admin socket
- log file
- pid file
- mon data
- mon cluster log file
- osd data
- osd journal
- mds data
- rgw data

See [General Settings](#), [OSD Settings](#), [Monitor Settings](#), [MDS Settings](#), [RGW Settings](#) and [Log Settings](#) for relevant path defaults that use the `$cluster` metavariable.

When deploying the Ceph configuration file, ensure that you use the cluster name in your command line syntax. For example:

```
ssh myserver01 sudo tee /etc/ceph/openstack.conf < /etc/ceph/openstack.conf
```

When creating default directories or files, you should also use the cluster name at the appropriate places in the path. For example:

```
sudo mkdir /var/lib/ceph/osd/openstack-0
sudo mkdir /var/lib/ceph/mon/openstack-a
```

Important: When running monitors on the same host, you should use different ports. By default, monitors use port 6789. If you already have monitors using port 6789, use a different port for your other cluster(s).

To invoke a cluster other than the default ceph cluster, use the `--cluster=clustername` option with the `ceph` command. For example:

```
ceph --cluster=openstack health
```