COMMON SETTINGS

The Hardware Recommendations section provides some hardware guidelines for configuring a Ceph Storage Cluster. It is possible for a single Ceph Node to run multiple daemons. For example, a single node with multiple drives may run one `ceph-osd` for each drive. Ideally, you will have a node for a particular type of process. For example, some nodes may run `ceph-osd` daemons, other nodes may run `ceph-mds` daemons, and still other nodes may run `ceph-mon` daemons.

Each node 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 monitor daemons. For example:

```
[global]
mon_initial_members = ceph1
mon_host = 10.0.0.1
```

Important: The `host` setting is the short name of the node (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 node. Do not use `host` settings for anything other than initial monitors unless you are deploying Ceph manually. You **MUST NOT** specify `host` under individual daemons when using deployment tools like `chef` or `ceph-deploy`, as those tools will enter the appropriate values for you in the cluster map.

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 Ceph Monitor daemons to ensure high availability should a monitor instance crash. At least three (3) monitors ensures that the Paxos algorithm can determine which version of the Ceph Cluster Map is the most recent from a majority of Ceph Monitors in the quorum.

Note: You may deploy Ceph with a single monitor, but if the instance fails, the lack of other monitors 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 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 Ceph OSD Daemons where one node has one OSD daemon running a filestore on one storage drive. A typical deployment specifies a journal size. For example:

```
[osd]
osd journal size = 10000

[osd.0]
host = {hostname} #manual deployments only.
```

By default, Ceph expects that you will store a Ceph OSD Daemon'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 when running **mkfs**. (`btrfs` and `ext4` are not recommended and no longer tested.)

See the OSD Config Reference for additional configuration details.

HEARTBEATS

During runtime operations, Ceph OSD Daemons check up on other Ceph OSD Daemons and report their findings to the Ceph 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

Sometimes you may encounter issues with Ceph that require modifying logging output and using Ceph's debugging. See Debugging and Logging for details on log rotation.

EXAMPLE CEPH.CONF

```
[global]
fsid = {cluster-id}
mon initial members = {hostname}[, {hostname}]
mon host = {ip-address}[, {ip-address}]

#All clusters have a front-side public network.
#If you have two NICs, you can configure a back side cluster
#network for OSD object replication, heart beats, backfilling,
#recovery, etc.
public network = {network}[, {network}]
#cluster network = {network}[, {network}]

#Clusters require authentication by default.
auth cluster required = cephx
auth service required = cephx
auth client required = cephx

#Choose reasonable numbers for your journals, number of replicas
#and placement groups.
osd journal size = {n}
osd pool default size = {n}  # Write an object n times.
osd pool default min size = {n} # Allow writing n copy in a degraded state.
osd pool default pg num = {n}
osd pool default pgp num = {n}

#Choose a reasonable crush leaf type.
#0 for a 1-node cluster.
#1 for a multi node cluster in a single rack
#2 for a multi node, multi chassis cluster with multiple hosts in a chassis
#3 for a multi node cluster with hosts across racks, etc.
osd crush chooseleaf type = {n}
```

RUNNING MULTIPLE CLUSTERS

With Ceph, you can run multiple Ceph Storage 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 rules. 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.

See ceph-deploy new for details. .. _ceph-deploy new:../ceph-deploy-new

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 creating default directories or files, you should 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 `-c {filename}.conf` option with the ceph command. For example:

```
ceph -c {cluster-name}.conf health
ceph -c openstack.conf health
```