OSD CONFIG REFERENCE

You can configure OSDs in the Ceph configuration file, but OSDs can use the default values and a very minimal configuration. A minimal OSD configuration sets `osd journal size` and `osd host`, and uses default values for nearly everything else.

OSDs are numerically identified in incremental fashion, beginning with `0` using the following convention.

```
osd.0
osd.1
osd.2
```

In a configuration file, you may specify settings for all OSDs in the cluster by adding configuration settings to the `[osd]` section of your configuration file. To add settings directly to a specific OSD (e.g., `osd host`), enter it in an OSD-specific section of your configuration file. For example:

```
[osd]
        osd journal size = 1024

[osd.0]
        osd host = osd-host-a

[osd.1]
        osd host = osd-host-b
```

GENERAL SETTINGS

The following settings provide an OSD's ID, and determine paths to data and journals. Ceph deployment scripts typically generate the UUID automatically. We **DO NOT** recommend changing the default paths for data or journals, as it makes it more problematic to troubleshoot Ceph later.

The journal size should be at least twice the product of the expected drive speed multiplied by `filestore max sync interval`. However, the most common practice is to partition the journal drive (often an SSD), and mount it such that Ceph uses the entire partition for the journal.

`osd uuid`

Description: The universally unique identifier (UUID) for the OSD.
Type: UUID
Default: The UUID.
Note: The `osd uuid` applies to a single OSD. The `fsid` applies to the entire cluster.

`osd data`

Description: The path to the OSDs data. You must create the directory when deploying Ceph. You should mount a drive for OSD data at this mount point. We do not recommend changing the default.
Type: String
Default: `/var/lib/ceph/osd/$cluster-$id`

`osd max write size`

Description: The maximum size of a write in megabytes.
Type: 32-bit Integer
Default: `90`

`osd client message size cap`

Description: The largest client data message allowed in memory.
Type: 64-bit Integer Unsigned
Default: 500MB default. `500*1024L*1024L`

`osd class dir`

Description: The class path for RADOS class plug-ins.
Type: String
Default: `$libdir/rados-classes`

JOURNAL SETTINGS

By default, Ceph expects that you will store an OSDs journal with the following path:

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

Without performance optimization, Ceph stores the journal on the same disk as the OSDs data. An OSD optimized for performance may use a separate disk to store journal data (e.g., a solid state drive delivers high performance journaling).

Ceph’s default `osd journal size` is 0, so you will need to set this in your `ceph.conf` file. A journal size should find the product of the `filestore max sync interval` and the expected throughput, and multiply the product by two (2):

```
osd journal size = {2 * (expected throughput * filestore max sync interval)}
```

The expected throughput number should include the expected disk throughput (i.e., sustained data transfer rate), and network throughput. For example, a 7200 RPM disk will likely have approximately 100 MB/s. Taking the `min()` of the disk and network throughput should provide a reasonable expected throughput. Some users just start off with a 10GB journal size. For example:

```
osd journal size = 10000
```

`osd journal`

Description: The path to the OSD’s journal. This may be a path to a file or a block device (such as a partition of an SSD). If it is a file, you must create the directory to contain it. We recommend using a drive separate from the `osd data` drive.
Type: String
Default: `/var/lib/ceph/osd/$cluster-$id/journal`

`osd journal size`

Description: The size of the journal in megabytes. If this is 0, and the journal is a block device, the entire block device is used. Since v0.54, this is ignored if the journal is a block device, and the entire block device is used.
Type: 32-bit Integer
Default: `5120`
Recommended: Begin with 1GB. Should be at least twice the product of the expected speed multiplied by `filestore max sync interval`.

See Journal Config Reference for additional details.

MONITOR OSD INTERACTION

OSDs check each other’s heartbeats and report to monitors periodically. Ceph can use default values in many cases. However, if your network has latency issues, you may need to adopt longer intervals. See Configuring Monitor/OSD Interaction for a detailed discussion of heartbeats.

DATA PLACEMENT

See Pool & PG Config Reference for details.

SCRUBBING

In addition to making multiple copies of objects, Ceph insures data integrity by scrubbing placement groups. Ceph scrubbing is analogous to `fsck` on the object storage layer. For each placement group, Ceph generates a catalog of all objects and compares each primary object and its replicas to ensure that no objects are missing or mismatched. Light scrubbing (daily) checks the object size and attributes. Deep scrubbing (weekly) reads the data and uses checksums to ensure data integrity.

Scrubbing is important for maintaining data integrity, but it can reduce performance. You can adjust the following settings to increase or decrease scrubbing operations.

`osd max scrubs`

Description: The maximum number of scrub operations for an OSD.
Type: 32-bit Int
Default: `1`

`osd scrub thread timeout`

Description: The maximum time in seconds before timing out a scrub thread.
Type: 32-bit Integer
Default: `60`

`osd scrub finalize thread timeout`

Description: The maximum time in seconds before timing out a scrub finalize thread.
Type: 32-bit Integer
Default: `60*10`

`osd scrub load threshold`

Description: The maximum CPU load. Ceph will not scrub when the CPU load is higher than this number. Default is 50%.
Type: Float
Default: `0.5`

`osd scrub min interval`

Description: The maximum interval in seconds for scrubbing the OSD when the cluster load is low.
Type: Float
Default: 5 minutes. `300`

`osd scrub max interval`

Description: The maximum interval in seconds for scrubbing the OSD irrespective of cluster load.
Type: Float
Default: Once per day. `60*60*24`

`osd deep scrub interval`

Description: The interval for “deep” scrubbing (fully reading all data).
Type: Float
Default: Once per week. `60*60*24*7`

`osd deep scrub stride`

Description: Read size when doing a deep scrub.
Type: 32-bit Int
Default: 512 KB. 524288

OPERATIONS

Operations settings allow you to configure the number of threads for servicing requests. If you set `osd op threads` to `0`, it disables multi-threading. By default, Ceph uses two threads with a 30 second timeout and a 30 second complaint time if an operation doesn’t complete within those time parameters. You can set operations priority weights between client operations and recovery operations to ensure optimal performance during recovery.

`osd op threads`

Description: The number of threads to service OSD operations. Set to `0` to disable it. Increasing the number may increase the request processing rate.

Type: 32-bit Integer

Default: `2`

`osd client op priority`

Description: The priority set for client operations. It is relative to `osd recovery op priority`.

Type: 32-bit Integer

Default: `63`

Valid Range: 1-63

`osd recovery op priority`

Description: The priority set for recovery operations. It is relative to `osd client op priority`.

Type: 32-bit Integer

Default: `10`

Valid Range: 1-63

`osd op thread timeout`

Description: The OSD operation thread timeout in seconds.

Type: 32-bit Integer

Default: `30`

`osd op complaint time`

Description: An operation becomes complaint worthy after the specified number of seconds have elapsed.

Type: Float

Default: `30`

`osd disk threads`

Description: The number of disk threads, which are used to perform background disk intensive OSD operations such as scrubbing and snap trimming.

Type: 32-bit Integer

Default: `1`

`osd op history size`

Description: The maximum number of completed operations to track.

Type: 32-bit Unsigned Integer

Default: `20`

`osd op history duration`

Description: The oldest completed operation to track.

Type: 32-bit Unsigned Integer

Default: `600`

`osd op log threshold`

Description: How many operations logs to display at once.

Type: 32-bit Integer

Default: `5`

BACKFILLING

When you add or remove OSDs to a cluster, the CRUSH algorithm will want to rebalance the cluster by moving placement groups to or from OSDs to restore the balance. The process of migrating placement groups and the objects they contain can reduce the cluster’s operational performance considerably. To maintain operational performance, Ceph performs this migration with ‘backfilling’, which allows Ceph to set backfill operations to a lower priority than requests to read or write data.

`osd max backfills`

Description: The maximum number of backfills allowed to or from a single OSD.
Type: 64-bit Unsigned Integer
Default: `10`

`osd backfill scan min`

Description: The scan interval in seconds for backfill operations when cluster load is low.
Type: 32-bit Integer
Default: `64`

`osd backfill scan max`

Description: The maximum scan interval in seconds for backfill operations irrespective of cluster load.
Type: 32-bit Integer
Default: `512`

`osd backfill full ratio`

Description: Refuse to accept backfill requests when the OSD’s full ratio is above this value.
Type: Float
Default: `0.85`

`osd backfill retry interval`

Description: The number of seconds to wait before retrying backfill requests.
Type: Double
Default: `10.0`

OSD MAP

OSD maps reflect the OSD daemons operating in the cluster. Over time, the number of map epochs increases. Ceph provides some settings to ensure that Ceph performs well as the OSD map grows larger.

`osd map dedup`

Description: Enable removing duplicates in the OSD map.
Type: Boolean
Default: `true`

`osd map cache size`

Description: The size of the OSD map cache in megabytes.
Type: 32-bit Integer
Default: `500`

`osd map cache bl size`

Description: The size of the in-memory OSD map cache in OSD daemons.
Type: 32-bit Integer
Default: `50`

`osd map cache bl inc size`

Description: The size of the in-memory OSD map cache incrementals in OSD daemons.
Type: 32-bit Integer
Default: `100`

`osd map message max`

Description: The maximum map entries allowed per MOSDMap message.
Type: 32-bit Integer
Default: `100`

RECOVERY

When the cluster starts or when an OSD crashes and restarts, the OSD begins peering with other OSDs before writes can occur. See Monitoring OSDs and PGs for details.

If an OSD crashed and comes back online, usually it will be out of sync with other OSDs containing more recent versions of objects in the placement groups. When this happens, the OSD goes into recovery mode and seeks to get the latest copy of the data and bring its map back up to date. Depending upon how long the OSD was down, the OSD’s objects and placement groups may be significantly out of date. Also, if a failure domain went down (e.g., a rack), more than one OSD may come back online at the same time. This can make the recovery process time consuming and resource intensive.

To maintain operational performance, Ceph performs recovery with limitations on the number recovery requests, threads and object chunk sizes which allows Ceph perform well in a degraded state.

`osd recovery delay start`

Description: After peering completes, Ceph will delay for the specified number of seconds before starting to recover objects.
Type: Float
Default: 15

`osd recovery max active`

Description: The number of active recovery requests per OSD at one time. More requests will accelerate recovery, but the requests places an increased load on the cluster.
Type: 32-bit Integer
Default: 5

`osd recovery max chunk`

Description: The maximum size of a recovered chunk of data to push.
Type: 64-bit Integer Unsigned
Default: `1 << 20`

`osd recovery threads`

Description: The number of threads for recovering data.
Type: 32-bit Integer
Default: 1

`osd recovery thread timeout`

Description: The maximum time in seconds before timing out a recovery thread.
Type: 32-bit Integer
Default: 30

`osd recover clone overlap`

Description: Preserves clone overlap during recovery. Should always be set to `true`.
Type: Boolean
Default: `true`

MISCELLANEOUS

`osd snap trim thread timeout`

Description: The maximum time in seconds before timing out a snap trim thread.
Type: 32-bit Integer
Default: `60*60*1`

`osd backlog thread timeout`

Description: The maximum time in seconds before timing out a backlog thread.
Type: 32-bit Integer

Default: `60*60*1`

`osd default notify timeout`

Description: The OSD default notification timeout (in seconds).
Type: 32-bit Integer Unsigned
Default: `30`

`osd check for log corruption`

Description: Check log files for corruption. Can be computationally expensive.
Type: Boolean
Default: `false`

`osd remove thread timeout`

Description: The maximum time in seconds before timing out a remove OSD thread.
Type: 32-bit Integer
Default: `60*60`

`osd command thread timeout`

Description: The maximum time in seconds before timing out a command thread.
Type: 32-bit Integer
Default: `10*60`

`osd command max records`

Description: Limits the number of lost objects to return.
Type: 32-bit Integer
Default: `256`

`osd auto upgrade tmap`

Description: Uses `tmap` for `omap` on old objects.
Type: Boolean
Default: `true`

`osd tmapput sets users tmap`

Description: Uses `tmap` for debugging only.
Type: Boolean
Default: `false`

`osd preserve trimmed log`

Description: Preserves trimmed log files, but uses more disk space.
Type: Boolean
Default: `false`