POOLS

When you first deploy a cluster without creating a pool, Ceph uses the default pools for storing data. A pool provides you with:

- **Resilience**: You can set how many OSD are allowed to fail without losing data. For replicated pools, it is the desired number of copies/replicas of an object. A typical configuration stores an object and one additional copy (i.e., `size = 2`), but you can determine the number of copies/replicas. For erasure coded pools, it is the number of coding chunks (i.e. `m=2` in the **erasure code profile**)
- **Placement Groups**: You can set the number of placement groups for the pool. A typical configuration uses approximately 100 placement groups per OSD to provide optimal balancing without using up too many computing resources. When setting up multiple pools, be careful to ensure you set a reasonable number of placement groups for both the pool and the cluster as a whole.
- **CRUSH Rules**: When you store data in a pool, placement of the object and its replicas (or chunks for erasure coded pools) in your cluster is governed by CRUSH rules. You can create a custom CRUSH rule for your pool if the default rule is not appropriate for your use case.
- **Snapshots**: When you create snapshots with `ceph osd pool mksnap`, you effectively take a snapshot of a particular pool.

To organize data into pools, you can list, create, and remove pools. You can also view the utilization statistics for each pool.

LIST POOLS

To list your cluster's pools, execute:

```
ceph osd lspools
```

CREATE A POOL

Before creating pools, refer to the Pool, PG and CRUSH Config Reference. Ideally, you should override the default value for the number of placement groups in your Ceph configuration file, as the default is NOT ideal. For details on placement group numbers refer to setting the number of placement groups

Note: Starting with Luminous, all pools need to be associated to the application using the pool. See Associate Pool to Application below for more information.

For example:

```
osd pool default pg num = 100
osd pool default pgp num = 100
```

To create a pool, execute:

```
ceph osd pool create {pool-name} {pg-num} [{pgp-num}] [replicated] \
     [crush-rule-name] [expected-num-objects]
ceph osd pool create {pool-name} {pg-num}  {pgp-num}   erasure \
     [erasure-code-profile] [crush-rule-name] [expected_num_objects]
```

Where:

`{pool-name}`

Description: The name of the pool. It must be unique.
Type: String
Required: Yes.

`{pg-num}`

Description: The total number of placement groups for the pool. See Placement Groups for details on calculating a suitable number. The default value 8 is NOT suitable for most systems.

Type: Integer
Required: Yes.
Default: 8

`{pgp-num}`

Description: The total number of placement groups for placement purposes. This **should be equal to the total number of placement groups**, except for placement group splitting scenarios.
Type: Integer
Required: Yes. Picks up default or Ceph configuration value if not specified.
Default: 8

`{replicated|erasure}`

Description: The pool type which may either be **replicated** to recover from lost OSDs by keeping multiple copies of the objects or **erasure** to get a kind of generalized RAID5 capability. The **replicated** pools require more raw storage but implement all Ceph operations. The **erasure** pools require less raw storage but only implement a subset of the available operations.
Type: String
Required: No.
Default: replicated

`[crush-rule-name]`

Description: The name of a CRUSH rule to use for this pool. The specified rule must exist.
Type: String
Required: No.
Default: For **replicated** pools it is the rule specified by the `osd pool default crush rule` config variable. This rule must exist. For **erasure** pools it is `erasure-code` if the `default` erasure code profile is used or `{pool-name}` otherwise. This rule will be created implicitly if it doesn’t exist already.

`[erasure-code-profile=profile]`

Description: For **erasure** pools only. Use the erasure code profile. It must be an existing profile as defined by **osd erasure-code-profile set**.
Type: String
Required: No.

When you create a pool, set the number of placement groups to a reasonable value (e.g., `100`). Consider the total number of placement groups per OSD too. Placement groups are computationally expensive, so performance will degrade when you have many pools with many placement groups (e.g., 50 pools with 100 placement groups each). The point of diminishing returns depends upon the power of the OSD host.

See Placement Groups for details on calculating an appropriate number of placement groups for your pool.

`[expected-num-objects]`

Description: The expected number of objects for this pool. By setting this value (together with a negative **filestore merge threshold**), the PG folder splitting would happen at the pool creation time, to avoid the latency impact to do a runtime folder splitting.
Type: Integer
Required: No.
Default: 0, no splitting at the pool creation time.

ASSOCIATE POOL TO APPLICATION

Pools need to be associated with an application before use. Pools that will be used with CephFS or pools that are automatically created by RGW are automatically associated. Pools that are intended for use with RBD should be initialized using the `rbd` tool (see Block Device Commands for more information).

For other cases, you can manually associate a free-form application name to a pool.:

```
ceph osd pool application enable {pool-name} {application-name}
```

Note: CephFS uses the application name cephfs, RBD uses the application name rbd, and RGW uses the application name rgw.

SET POOL QUOTAS

You can set pool quotas for the maximum number of bytes and/or the maximum number of objects per pool.

```
ceph osd pool set-quota {pool-name} [max_objects {obj-count}] [max_bytes {bytes}]
```

For example:

```
ceph osd pool set-quota data max_objects 10000
```

To remove a quota, set its value to 0.

DELETE A POOL

To delete a pool, execute:

```
ceph osd pool delete {pool-name} [{pool-name} --yes-i-really-really-mean-it]
```

To remove a pool the mon_allow_pool_delete flag must be set to true in the Monitor's configuration. Otherwise they will refuse to remove a pool.

See Monitor Configuration for more information.

If you created your own rules for a pool you created, you should consider removing them when you no longer need your pool:

```
ceph osd pool get {pool-name} crush_rule
```

If the rule was “123”, for example, you can check the other pools like so:

```
ceph osd dump | grep "^pool" | grep "crush_rule 123"
```

If no other pools use that custom rule, then it's safe to delete that rule from the cluster.

If you created users with permissions strictly for a pool that no longer exists, you should consider deleting those users too:

```
ceph auth ls | grep -C 5 {pool-name}
ceph auth del {user}
```

RENAME A POOL

To rename a pool, execute:

```
ceph osd pool rename {current-pool-name} {new-pool-name}
```

If you rename a pool and you have per-pool capabilities for an authenticated user, you must update the user's capabilities (i.e., caps) with the new pool name.

Note: Version 0.48 Argonaut and above.

SHOW POOL STATISTICS

To show a pool's utilization statistics, execute:

```
rados df
```

MAKE A SNAPSHOT OF A POOL

To make a snapshot of a pool, execute:

```
ceph osd pool mksnap {pool-name} {snap-name}
```

Note: Version 0.48 Argonaut and above.

REMOVE A SNAPSHOT OF A POOL

To remove a snapshot of a pool, execute:

```
ceph osd pool rmsnap {pool-name} {snap-name}
```

Note: Version 0.48 Argonaut and above.

SET POOL VALUES

To set a value to a pool, execute the following:

```
ceph osd pool set {pool-name} {key} {value}
```

You may set values for the following keys:

`compression_algorithm`

Description: Sets inline compression algorithm to use for underlying BlueStore. This setting overrides the global setting of `bluestore compression algorithm`.

Type: String

Valid Settings: `lz4`, `snappy`, `zlib`, `zstd`

`compression_mode`

Description: Sets the policy for the inline compression algorithm for underlying BlueStore. This setting overrides the global setting of `bluestore compression mode`.

Type: String

Valid Settings: `none`, `passive`, `aggressive`, `force`

`compression_min_blob_size`

Description: Chunks smaller than this are never compressed. This setting overrides the global setting of `bluestore compression min blob *`.

Type: Unsigned Integer

`compression_max_blob_size`

Description: Chunks larger than this are broken into smaller blobs sizing `compression_max_blob_size` before being compressed.

Type: Unsigned Integer

`size`

Description: Sets the number of replicas for objects in the pool. See Set the Number of Object Replicas for further details. Replicated pools only.
Type: Integer

`min_size`

Description: Sets the minimum number of replicas required for I/O. See Set the Number of Object Replicas for further details. Replicated pools only.
Type: Integer
Version: `0.54` and above

`pg_num`

Description: The effective number of placement groups to use when calculating data placement.
Type: Integer
Valid Range: Superior to `pg_num` current value.

`pgp_num`

Description: The effective number of placement groups for placement to use when calculating data placement.
Type: Integer
Valid Range: Equal to or less than `pg_num`.

`crush_rule`

Description: The rule to use for mapping object placement in the cluster.
Type: Integer

`allow_ec_overwrites`

Description: Whether writes to an erasure coded pool can update part of an object, so cephfs and rbd can use it. See Erasure Coding with Overwrites for more details.
Type: Boolean
Version: `12.2.0` and above

`hashpspool`

Description: Set/Unset HASHPSPOOL flag on a given pool.
Type: Integer
Valid Range: 1 sets flag, 0 unsets flag
Version: Version `0.48` Argonaut and above.

`nodelete`

Description: Set/Unset NODELETE flag on a given pool.
Type: Integer
Valid Range: 1 sets flag, 0 unsets flag
Version: Version FIXME

`nopgchange`

Description: Set/Unset NOPGCHANGE flag on a given pool.
Type: Integer
Valid Range: 1 sets flag, 0 unsets flag
Version: Version FIXME

`nosizechange`

Description: Set/Unset NOSIZECHANGE flag on a given pool.
Type: Integer
Valid Range: 1 sets flag, 0 unsets flag
Version: Version FIXME

`write_fadvise_dontneed`

Description: Set/Unset WRITE_FADVISE_DONTNEED flag on a given pool.
Type: Integer
Valid Range: 1 sets flag, 0 unsets flag

`noscrub`

Description: Set/Unset NOSCRUB flag on a given pool.
Type: Integer
Valid Range: 1 sets flag, 0 unsets flag

`nodeep-scrub`

Description: Set/Unset NODEEP_SCRUB flag on a given pool.
Type: Integer
Valid Range: 1 sets flag, 0 unsets flag

`hit_set_type`

Description: Enables hit set tracking for cache pools. See Bloom Filter for additional information.
Type: String
Valid Settings: `bloom`, `explicit_hash`, `explicit_object`
Default: `bloom`. Other values are for testing.

`hit_set_count`

Description: The number of hit sets to store for cache pools. The higher the number, the more RAM consumed by the `ceph-osd` daemon.
Type: Integer
Valid Range: 1. Agent doesn't handle > 1 yet.

`hit_set_period`

Description: The duration of a hit set period in seconds for cache pools. The higher the number, the more RAM consumed by the `ceph-osd` daemon.
Type: Integer
Example: `3600` 1hr

`hit_set_fpp`

Description: The false positive probability for the `bloom` hit set type. See Bloom Filter for additional information.
Type: Double
Valid Range: 0.0 - 1.0
Default: `0.05`

`cache_target_dirty_ratio`

Description: The percentage of the cache pool containing modified (dirty) objects before the cache tiering agent will flush them to the backing storage pool.
Type: Double
Default: `.4`

`cache_target_dirty_high_ratio`

Description: The percentage of the cache pool containing modified (dirty) objects before the cache tiering agent will flush them to the backing storage pool with a higher speed.
Type: Double
Default: `.6`

`cache_target_full_ratio`

Description: The percentage of the cache pool containing unmodified (clean) objects before the cache tiering agent will evict them from the cache pool.

Type: Double
Default: `.8`

`target_max_bytes`

Description: Ceph will begin flushing or evicting objects when the `max_bytes` threshold is triggered.
Type: Integer
Example: `1000000000000` #1-TB

`target_max_objects`

Description: Ceph will begin flushing or evicting objects when the `max_objects` threshold is triggered.
Type: Integer
Example: `1000000` #1M objects

`hit_set_grade_decay_rate`

Description: Temperature decay rate between two successive hit_sets
Type: Integer
Valid Range: 0 - 100
Default: `20`

`hit_set_search_last_n`

Description: Count at most N appearance in hit_sets for temperature calculation
Type: Integer
Valid Range: 0 - hit_set_count
Default: `1`

`cache_min_flush_age`

Description: The time (in seconds) before the cache tiering agent will flush an object from the cache pool to the storage pool.
Type: Integer
Example: `600` 10min

`cache_min_evict_age`

Description: The time (in seconds) before the cache tiering agent will evict an object from the cache pool.
Type: Integer
Example: `1800` 30min

`fast_read`

Description: On Erasure Coding pool, if this flag is turned on, the read request would issue sub reads to all shards, and waits until it receives enough shards to decode to serve the client. In the case of jerasure and isa erasure plugins, once the first K replies return, client’s request is served immediately using the data decoded from these replies. This helps to tradeoff some resources for better performance. Currently this flag is only supported for Erasure Coding pool.
Type: Boolean
Defaults: `0`

`scrub_min_interval`

Description: The minimum interval in seconds for pool scrubbing when load is low. If it is 0, the value osd_scrub_min_interval from config is used.
Type: Double
Default: `0`

`scrub_max_interval`

Description: The maximum interval in seconds for pool scrubbing irrespective of cluster load. If it is 0, the value osd_scrub_max_interval from config is used.
Type: Double
Default: `0`

`deep_scrub_interval`

Description: The interval in seconds for pool “deep” scrubbing. If it is 0, the value osd_deep_scrub_interval from config is used.
Type: Double
Default: `0`

GET POOL VALUES

To get a value from a pool, execute the following:

```
ceph osd pool get {pool-name} {key}
```

You may get values for the following keys:

`size`

Description: see size
Type: Integer

`min_size`

Description: see min_size
Type: Integer
Version: `0.54` and above

`pg_num`

Description: see pg_num
Type: Integer

`pgp_num`

Description: see pgp_num
Type: Integer
Valid Range: Equal to or less than `pg_num`.

`crush_rule`

Description: see crush_rule

`hit_set_type`

Description: see hit_set_type
Type: String
Valid Settings: `bloom`, `explicit_hash`, `explicit_object`

`hit_set_count`

Description: see hit_set_count
Type: Integer

`hit_set_period`

Description: see hit_set_period
Type: Integer

`hit_set_fpp`

Description: see hit_set_fpp
Type: Double

`cache_target_dirty_ratio`

Description: see cache_target_dirty_ratio
Type: Double

`cache_target_dirty_high_ratio`

Description: see cache_target_dirty_high_ratio
Type: Double

`cache_target_full_ratio`

Description: see cache_target_full_ratio
Type: Double

`target_max_bytes`

Description: see target_max_bytes
Type: Integer

`target_max_objects`

Description: see target_max_objects
Type: Integer

`cache_min_flush_age`

Description: see cache_min_flush_age
Type: Integer

`cache_min_evict_age`

Description: see cache_min_evict_age
Type: Integer

`fast_read`

Description: see fast_read
Type: Boolean

`scrub_min_interval`

Description: see scrub_min_interval
Type: Double

`scrub_max_interval`

Description: see scrub_max_interval
Type: Double

`deep_scrub_interval`

Description: see deep_scrub_interval
Type: Double

SET THE NUMBER OF OBJECT REPLICAS

To set the number of object replicas on a replicated pool, execute the following:

```
ceph osd pool set {poolname} size {num-replicas}
```

Important: The `{num-replicas}` includes the object itself. If you want the object and two copies of the object for a total of three instances of the object, specify 3.

For example:

```
ceph osd pool set data size 3
```

You may execute this command for each pool. **Note:** An object might accept I/Os in degraded mode with fewer than `pool size` replicas. To set a minimum number of required replicas for I/O, you should use the `min_size` setting. For example:

```
ceph osd pool set data min_size 2
```

This ensures that no object in the data pool will receive I/O with fewer than `min_size` replicas.

GET THE NUMBER OF OBJECT REPLICAS

To get the number of object replicas, execute the following:

```
ceph osd dump | grep 'replicated size'
```

Ceph will list the pools, with the `replicated size` attribute highlighted. By default, ceph creates two replicas of an object (a total of three copies, or a size of 3).