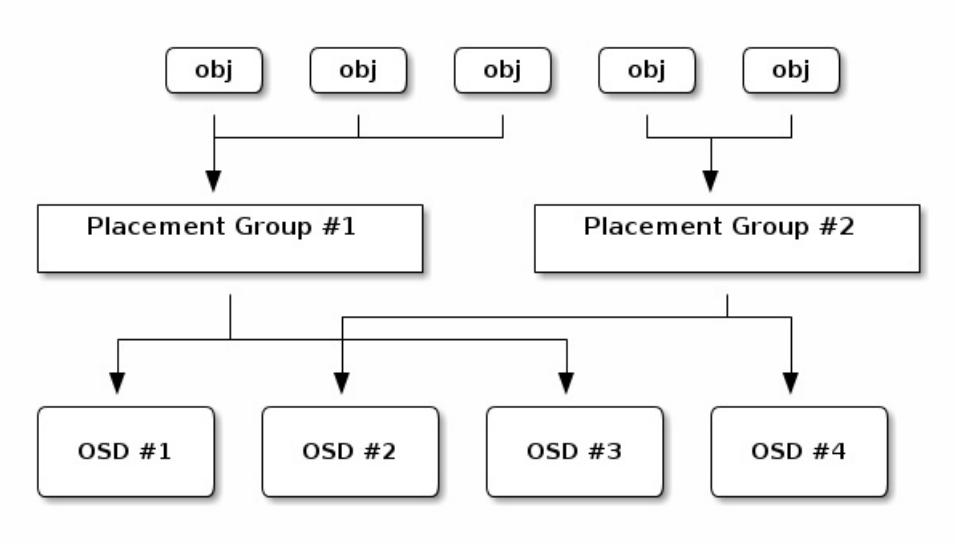
PLACEMENT GROUPS

A Placement Group (PG) aggregates a series of objects into a group, and maps the group to a series of OSDs. Tracking object placement and object metadata on a per-object basis is computationally expensive–i.e., a system with millions of objects cannot realistically track placement on a per-object basis. Placement groups address this barrier to performance and scalability. Additionally, placement groups reduce the number of processes and the amount of per-object metadata Ceph must track when storing and retrieving data.



Each placement group requires some amount of system resources:

- **Directly**: Each PG requires some amount of memory and CPU.
- **Indirectly**: The total number of PGs increases the peering count.

Increasing the number of placement groups reduces the variance in per-OSD load across your cluster. We recommend approximately 50-100 placement groups per OSD to balance out memory and CPU requirements and per-OSD load. For a single pool of objects, you can use the following formula:

```
            (OSDs * 100)
Total PGs = ------------
              Replicas
```

When using multiple data pools for storing objects, you need to ensure that you balance the number of placement groups per pool with the number of placement groups per OSD so that you arrive at a reasonable total number of placement groups that provides reasonably low variance per OSD without taxing system resources or making the peering process too slow.

SET THE NUMBER OF PLACEMENT GROUPS

To set the number of placement groups in a pool, you must specify the number of placement groups at the time you create the pool.

See Create a Pool for details.

GET THE NUMBER OF PLACEMENT GROUPS

To get the number of placement groups in a pool, execute the following:

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

GET A CLUSTER'S PG STATISTICS

To get the statistics for the placement groups in your cluster, execute the following:

```
ceph pg dump [--format {format}]
```

Valid formats are `plain` (default) and `json`.

GET STATISTICS FOR STUCK PGS

To get the statistics for all placement groups stuck in a specified state, execute the following:

```
ceph pg dump_stuck inactive|unclean|stale [--format <format>] [-t|--threshold <seconds>]
```

Inactive Placement groups cannot process reads or writes because they are waiting for an OSD with the most up-to-date data to come up and in.

Unclean Placement groups contain objects that are not replicated the desired number of times. They should be recovering.

Stale Placement groups are in an unknown state - the OSDs that host them have not reported to the monitor cluster in a while (configured by `mon_osd_report_timeout`).

Valid formats are `plain` (default) and `json`. The threshold defines the minimum number of seconds the placement group is stuck before including it in the returned statistics (default 300 seconds).

GET A PG MAP

To get the placement group map for a particular placement group, execute the following:

```
ceph pg map {pg-id}
```

For example:

```
ceph pg map 1.6c
```

Ceph will return the placement group map, the placement group, and the OSD status:

```
osdmap e13 pg 1.6c (1.6c) -> up [1,0] acting [1,0]
```

GET A PGS STATISTICS

To retrieve statistics for a particular placement group, execute the following:

```
ceph pg {pg-id} query
```

SCRUB A PLACEMENT GROUP

To scrub a placement group, execute the following:

```
ceph pg scrub {pg-id}
```

Ceph checks the primary and any replica nodes, generates a catalog of all objects in the placement group and compares them to ensure that no objects are missing or mismatched, and their contents are consistent. Assuming the replicas all match, a final semantic sweep ensures that all of the snapshot-related object metadata is consistent. Errors are reported via logs.

REVERT LOST

If the cluster has lost one or more objects, and you have decided to abandon the search for the lost data, you must mark the unfound objects as `lost`.

If all possible locations have been queried and objects are still lost, you may have to give up on the lost objects. This is possible given unusual combinations of failures that allow the cluster to learn about writes that were performed before the writes themselves are recovered.

Currently the only supported option is “revert”, which will either roll back to a previous version of the object or (if it was a new object) forget about it entirely. To mark the “unfound” objects as “lost”, execute the following:

```
ceph pg {pg-id} mark_unfound_lost revert
```

Important: Use this feature with caution, because it may confuse applications that expect the object(s) to exist.