RED HAT CEPH STORAGE CHEAT SHEET

Summary of Certain Operations-oriented Ceph Commands

Note: Certain command outputs in the Example column were edited for better readability.

Monitoring and Health

Command	Purpose	Example
ceph -s	Show status summary	# ceph -s cluster 1c528497-24e0-4af7-bb18-d43a8d31cecc health HEALTH_OK
ceph -w	Watch ongoing status	
rados df	Show per pool and total usage	# rados df **pool_name** rbd **used** 700M **objects** 176 **clones** 0 **copie** 528 **missing_on_primary** 0 **unfound** 0 **degraded** 0 **rd_ops** 0 **rd** 0 **wr_ops** 351 **wr** 700M **total_objects** 176 **total_used** 9220M **total_avail** 4824G **total_space** 4833G
ceph df	Show disk usage overview, global and per pool	# ceph df **GLOBAL** **size** 11172G **avail** 11172G **raw used** 501M **%raw** 0 **POOLS** **name** rbd **ID** 0 **used** 0 **%used** 0 **max avail** 3724G **objects** 0
ceph health detail	Show details about health issues	# ceph health detail HEALTH_WARN mon.ceph4 low disk space; mon.ceph5 low disk space; mon.ceph6 low disk space mon.ceph4 low disk space -- 18% avail mon.ceph5 low disk space -- 22% avail mon.ceph6 low disk space -- 16% avail

ceph osd df tree	Show disk usage linked to the CRUSH tree	# ceph osd df tree

ID	-1	-2	0	3	4	8
weight	10.91034	3.63678	0.90919	0.90919	0.90919	0.90919
reweight	- 11172G	- 3724G	1.00000	1.00000	1.00000	1.00000
size	501M	168M	931G	931G	931G	931G
use	11172G	3724G	44760k	42752k	42804k	42616k
avail	0.00	0.00	931G	931G	931G	931G
%use	1.00	1.01	0.00	0.00	0.00	0.00
var	0	0	1.05	1.0	1.0	1.0
type	root	host	69	63	62	62
name	default	ceh3	ods.0	ods.3	ods.4	ods.8

Working With Pools and OSDs

Subcommands of the "**ceph osd**" command

Command	Purpose	Example
ceph osd tree	Lits hosts, their OSDs, up and down status, OSD weight, local reweight	# ceph osd tree

ID	-1	-10	0	3	6
class			hdd	hdd	hdd
weight	4.72031	2.71317	0.90439	0.90439	0.90439
type name	root default	host	osd.0	osd.3	osd.6
status			up	up	up
reweight	1.00000	1.00000	1.00000	1.00000	1.00000
PRI-AFF	1.00000	1.00000	1.00000	1.00000	1.00000

Command	Purpose	Example
ceph osd stat	Print a summary of the OSD map	# ceph osd stat 9 osds: 9 up, 9 in
ceph osd deep-scrub *<id>*	Instruct Ceph to perform a deep scrubbing process (consistency check) on an OSD.	# ceph osd deep-scrub osd.0 osd.0 instructed to deep-scrub
ceph osd find *<id>*	Display location of a given OSD (host name, port, and CRUSH details)	# ceph osd find 0

```
{
  "osd": 0,
  "ip": "10.12.xxx.xxx:6804/61412",
  "crush_location": {
    "host": "ceph4",
    "root": "default"
  }
```

Command	Purpose	Example
ceph osd map *pool object*	Locate an object from a pool. Displays primary and replica placement groups for the object	# ceph osd map rbd benchmark_data_ceph1_268097_object865 osdmap e115 pool 'rbd' (3) object 'benchmark_data_ceph1_268097_object865' -> pg 3.c9f193ff (3.7f) -> up ([4,6,8], p4) acting ([4,6,8], p4)

ceph osd metadata <*id*>	Display OSD metadata (host and host info)	# ceph osd metadata 0
		{
		  "id": 0,
		  "arch": "x86_64",
		  "back_addr": "10.12.xxx.xxx:6805/61412",
		  "back_iface": "eno1",

ceph osd out <*id*>	Take an OSD out of the cluster, rebalancing its data to other OSDs.	# ceph osd out 0 marked out osd.0
ceph osd pool create <*pool-name*> <*pg-number*> <*pgs-number*>	Create a new replicated pool with a number of placement groups. Use the [Ceph Placement Groups (PGs) per Pool Calculator]() to determine the number of placement groups. Also, see [Pools]() in the Administration Guide.	# ceph osd pool create test 64 64 pool 'test' created
ceph osd pool delete <*pool-name*> <*pool-name*> *--yes-i-really-really-mean-it*	Delete a pool. Specify the pool name twice followed by confirmation. Be careful when deleting pools because this action cannot be reverted.	# ceph osd pool delete test test --yes-i-really-really-mean-it pool 'test' removed
ceph osd pool get <*pool*> all	Get all parameters for a pool. Specify pool name for specific pool.	# ceph osd pool get rbd all size: 3 min_size: 2...
ceph osd pool ls *detail*	List pools and details of pools.	# ceph osd pool ls detail pool 1 'rbd' replicated size 3 min_size 2 crush_rule 0 object_hash rjenkins pg_num 128 pgp_num 128 last_change 65 flags hashpspool stripe_width 0
ceph osd pool set <*parameter*> <*value*>	Set a pool parameter, for example, “size”, “min_size”, or “pg_num”	# ceph osd pool set rbd min_size 1 set pool 1 min_size to 1
ceph osd reweight <*id*> <*weight*>	Temporarily override weight for an OSD.	# ceph osd reweight 0 0.5 # use 50% of default space on osd.0

ceph osd reweight-by-utilization <*percent*>	Change the weight of OSDs based on their utilization. See Set an OSD's Weight by Utilization in the Storage Strategies Guide.	# ceph osd reweight-by-utilization 110 moved 7 / 576 (1.21528%) avg 64 stddev 26.7623 -> 26.8328 (expected baseline 7.54247) min osd.1 with 18 -> 18 pgs (0.28125 -> 0.28125 * mean) max osd.4 with 102 -> 102 pgs (1.59375 -> 1.59375 * mean)
ceph osd scrub <*id*>	Initiate a "light" scrub on an OSD.	# ceph osd scrub osd.0 osd.0 instructed to scrub
ceph osd test-reweight-by-utilization <*percent*>	Test how setting an OSD weight based on utilization will reflect data movement.	# ceph osd test-reweight-by-utilization 110 no change moved 3 / 576 (0.520833%) avg 64
ceph osd set <*flag*>	Set various flags on the OSD subsystem. See Overrides in the Administration Guide.	# ceph osd set noout

Working With Placement Group

Subcommands of the "**ceph pg**" command

Command	Purpose	Example
ceph pg *pg-id* query	Query statistics and other metadata about a placement group. Often valuable info for troubleshooting, for example the state of replicas, past events, and other.	# ceph pg 1.c query { "state": "active+clean", "snap_trimq": "[]", "epoch": 72, "up": [7, 3, 8],
ceph pg *pg-id* list_missing	List unfound objects. The "ceph pg *pg-id* query" command lists more information about which OSDs contain unfound objects. See Unfound Objects in the Troubleshooting Guide.	# ceph pg 1.c list_missing { "num_missing": 0, "num_unfound": 0, "objects": [],
ceph pg dump [--format *format*]	Show statistics and metadata for all placement groups including information about scrub processes, last replication, current OSDs, blocking OSDs, and so on. Format can be plain or json.	# ceph pg dump dumped all version 1409550 stamp 2017-10-24 08:51:54.763931 last_osdmap_epoch 0 last_pg_scan 0 full_ratio 0 nearfull_ratio 0...

ceph pg dump_stuck inactive | unclean | stale | undersized | degraded	Show stuck placement groups (PGs).See [Identifying Troubled Placement Groups]() in the Administration Guide.	# ceph pg dump_stuck unclean ok **pg_stat** 3.6 3.6 **stat** active+undersized+degraded active+undersized+degraded **up** [7,8] [8,4] **up_primary** 7 8 **acting** [7,8] [8,4] **acting_primary** 7 8
ceph pg scrub ***pg-id***	Initiate the scrub process on the placement groups contents.	# ceph pg scrub 3.0 instructing pg 3.0 on osd.1 to scrub
ceph deep-scrub ***pg-id***	Initiate the deep scrub process on the placement groups contents.	# ceph pg deep-scrub 3.0 instructing pg 3.0 on osd.1 to deep-scrub
ceph pg repair {pg-id}	Fix inconsistent placement groups. See [Repairing Inconsistent Placement Groups]() in the Troubleshooting Guide.	# ceph pg repair 3.0 instructing pg 3.0 on osd.1 to repair

Interaction With Individual Daemons

Subcommands of the "**ceph daemon <*daemon-name*>**" command. These commands interact with individual daemons on the current host. Typically, they are used for low-level investigation and troubleshooting. Specify the target daemon by its name, for example "osd.1", or by using a path to the daemon's socket file. For example, "/var/run/ceph/ceph-osd.0.asok".

Command	Purpose	Example

ceph daemon <*osd.id*> dump_ops_in_flight	Show a list of currently active operations for an OSD. Useful if one or more operations are inactive, stuck or blocked.	# ceph daemon osd.0 dump_ops_in_flight
		{
		  "ops": [
		    {
		      "description": "osd_op(client.24153.0:45 3.33 3:cd6d298e:::benchmark_data_ceph1_268097_object44:head [set-alloc-hint object_size 4194304 write_size 4194304,write 0~4194304] snapc 0=[] ondisk+write+known_if_redirected e115)",
		      "initiated_at": "2017-10-24

ceph daemon <*daemon-name*> help	Print a list of commands a daemon supports	# ceph daemon osd.0 help { "calc_objectstore_db_histogram": "Generate key value histogram of kvdb(rocksdb) which used by bluestore", "compact": "Commpact object store's omap. WARNING: Compaction probably slows your requests"...

ceph daemon ***<daemon-name>*** **mon_status**	Print high level status information for a Monitor	# ceph daemon mon.ceph1 mon_status { "name": "ceph1", "rank": 0, "state": "leader", "election_epoch": 6, "quorum": [0, 1, 2],
ceph daemon ***<osd.id>*** **status**	Print high level status information for an OSD	# ceph daemon osd.0 status { "cluster_fsid": "82282e8f-b8ff-4ec2-b564-e06a3e514fb7", "osd_fsid": "f05ea8f0-df33-440b-8921-511a93f2ec96", "whoami": 0,
ceph daemon ***<daemon-name>*** **perf dump**	Print performance statistics. See Performance Counters in the Administration Guide for details.	# ceph daemon client.radosgw.primary perf dump { "cct": {"total_workers": 16, "unhealthy_workers": 0 }, "client.radosgw.primary": { "req": 1156723,...

Authentication and Authorization

For details, see Managing Users in the Administration Guide.

Command	Purpose	Example
ceph auth list	List users	# ceph auth list installed auth entries: osd.0 key: AQDUIcRZKW5JERAA+DFBSVZLsmd0gjFK6TxS7A== caps: [mgr] allow profile osd caps: [mon] allow profile osd caps: [osd] allow *
ceph auth get-or-create	Get user details, or create the user if it does not exist yet and return details.	# ceph auth get-or-create client.rbd mon 'allow r' osd 'allow rw pool=rbd' [client.rbd] key = Axxxxxxxxxxx==
ceph auth delete	Delete a user	# ceph auth del updated
ceph auth caps	Add or remove permissions for a user. Permissions are grouped per daemon type (mon, osd, mds). Capabilities can be 'r', 'w', 'x' or '*'. See Authorization (Capabilities) in the Administration Guide for details.	# ceph auth caps client.bob mon 'allow *' osd 'allow *' mds 'allow *' updated caps for client.user1

Object Store Utility

The RADOS Object Store utility commands

Command	Purpose	Example
rados -p *pool* put *object file*	Upload a file into a pool, name the resulting object.	# rados -p rbd put myfile myfile.txt
rados -p *pool* ls	List objects in a pool	# rados -p rbd ls
rados -p *pool* get *object file*	Download an object from a pool into a local file. Give '-' as a file name to write to standard output	# rados -p rbd get myfile - new.txt
rados -p *pool* rm *object*	Delete an object from a pool	# rados -p test rm myfile
rados -p *pool* listwatchers *object*	List watchers of an object in pool. For instance, the head object of a mapped rbd volume has its clients as watchers	# rados -p rbd listwatchers benchmark_data_ceph1_268097_object 865watcher=12.10.x.x:0/330978585 client.28223 cookie=1

rados bench *seconds mode* [-b *object-size*] [-t *threads*]	Run the built-in benchmark for given time in seconds. Mode can be write, seq, or rand (latter are read benchmarks). Before running one of the reading benchmarks, run a write benchmark with the –no-cleanup option. The default object size is 4 MB, and the default number of simulated threads (parallel writes operations) is 16. See Benchmarking Performance in the Administration Guide for details.	# rados bench -p rbd 120 write --no-cleanup
		hints = 1
		Maintaining 16 concurrent writes of 4194304 bytes to objects of size 4194304 for up to 120 seconds or 0 objects