RADOS – RADOS OBJECT STORAGE UTILITY

SYNOPSIS

rados [-m *monaddr*] [mkpool | rmpool *foo*] [-p | –pool *pool*] [-s | –snap *snap*] [-i *infile*] [-o *outfile*] *command* ...

DESCRIPTION

rados is a utility for interacting with a Ceph object storage cluster (RADOS), part of the Ceph distributed storage system.

OPTIONS

-p `pool`, **--pool** `pool`
Interact with the given pool. Required by most commands.

-s `snap`, **--snap** `snap`
Read from the given pool snapshot. Valid for all pool-specific read operations.

-i `infile`
will specify an input file to be passed along as a payload with the command to the monitor cluster. This is only used for specific monitor commands.

-o `outfile`
will write any payload returned by the monitor cluster with its reply to outfile. Only specific monitor commands (e.g. osd getmap) return a payload.

-c `ceph.conf`, **--conf**=`ceph.conf`
Use ceph.conf configuration file instead of the default /etc/ceph/ceph.conf to determine monitor addresses during startup.

-m `monaddress[:port]`
Connect to specified monitor (instead of looking through ceph.conf).

-b `block_size`
Set the block size for put/get/append ops and for write benchmarking.

--striper
Uses the striping API of rados rather than the default one. Available for stat, stat2, get, put, append, truncate, rm, ls and all xattr related operation

GLOBAL COMMANDS

lspools
List object pools

df
Show utilization statistics, including disk usage (bytes) and object counts, over the entire system and broken down by pool.

mkpool *foo*
Create a pool with name foo.

rmpool *foo* [*foo* –yes-i-really-really-mean-it]
Delete the pool foo (and all its data).

list-inconsistent-pg *pool*
List inconsistent PGs in given pool.

list-inconsistent-obj *pgid*
List inconsistent objects in given PG.

list-inconsistent-snapset *pgid*
List inconsistent snapsets in given PG.

POOL SPECIFIC COMMANDS

get *name outfile*

Read object name from the cluster and write it to outfile.

put *name infile* [–offset offset]

Write object name with start offset (default:0) to the cluster with contents from infile.

append *name infile*

Append object name to the cluster with contents from infile.

rm *name*

Remove object name.

listwatchers *name*

List the watchers of object name.

ls *outfile*

List objects in given pool and write to outfile.

lssnap

List snapshots for given pool.

clonedata *srcname dstname* –object-locator *key*

Clone object byte data from *srcname* to *dstname*. Both objects must be stored with the locator key *key* (usually either *srcname* or *dstname*). Object attributes and omap keys are not copied or cloned.

mksnap *foo*

Create pool snapshot named *foo*.

rmsnap *foo*

Remove pool snapshot named *foo*.

bench *seconds mode* [-b *objsize*] [-t *threads*]

Benchmark for *seconds*. The mode can be *write*, *seq*, or *rand*. *seq* and *rand* are read benchmarks, either sequential or random. 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) is 16. The *–run-name <label>* option is useful for benchmarking a workload test from multiple clients. The *<label>* is an arbitrary object name. It is "benchmark_last_metadata" by default, and is used as the underlying object name for "read" and "write" ops. Note: -b *objsize* option is valid only in *write* mode. Note: *write* and *seq* must be run on the same host otherwise the objects created by *write* will have names that will fail *seq*.

cleanup [–run-name *run_name*] [–prefix *prefix*]

Clean up a previous benchmark operation. Note: the default run-name is "benchmark_last_metadata"

listxattr *name*

List all extended attributes of an object.

getxattr *name attr*

Dump the extended attribute value of *attr* of an object.

setxattr *name attr value*

Set the value of *attr* in the extended attributes of an object.

rmxattr *name attr*

Remove *attr* from the extended attributes of an object.

stat *name*

Get stat (ie. mtime, size) of given object

stat2 *name*

Get stat (similar to stat, but with high precision time) of given object

listomapkeys *name*

List all the keys stored in the object map of object name.

listomapvals *name*

List all key/value pairs stored in the object map of object name. The values are dumped in hexadecimal.

getomapval [–omap-key-file *file*] *name key* [*out-file*]

Dump the hexadecimal value of key in the object map of object name. If the optional *out-file* argument is not provided, the value will be written to standard output.

setomapval [–omap-key-file *file*] *name key* [*value*]

Set the value of key in the object map of object name. If the optional *value* argument is not provided, the value will be read from standard input.

rmomapkey [–omap-key-file *file*] *name key*
Remove key from the object map of object name.

getomapheader *name*
Dump the hexadecimal value of the object map header of object name.

setomapheader *name value*
Set the value of the object map header of object name.

EXAMPLES

To view cluster utilization:

```
rados df
```

To get a list object in pool foo sent to stdout:

```
rados -p foo ls -
```

To write an object:

```
rados -p foo put myobject blah.txt
```

To create a snapshot:

```
rados -p foo mksnap mysnap
```

To delete the object:

```
rados -p foo rm myobject
```

To read a previously snapshotted version of an object:

```
rados -p foo -s mysnap get myobject blah.txt.old
```

To list inconsistent objects in PG 0.6:

```
rados list-inconsistent-obj 0.6 --format=json-pretty
```

AVAILABILITY

rados is part of Ceph, a massively scalable, open-source, distributed storage system. Please refer to the Ceph documentation at http://ceph.com/docs for more information.

SEE ALSO

ceph(8)