Ceph Quick Reference Guide

Pre-Flight

```
echo "cephuser ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/cephuser
sudo chmod 0440 /etc/sudoers.d/cephuser
ceph-deploy new <hostname> --public-network <xxx.yyy.zzz.0/24> --cluster-network <aaa.bbb.ccc.0/24>
ceph-deploy install --release=mimic <nodename(s)>
ceph-deploy mon create-initial
ceph-deploy admin <nodename>
ceph-deploy mgr create <mgrnode>
ceph mgr module ls
ceph mgr module enable dashboard
ceph dashboard create-self-signed-cert (mimic)
ceph dashboard set-login-credentials <username> <password>
ceph mgr services
        example "dashboard": https://<node>:8080
yum install ceph-deploy-1.5.39 (old version of ceph-deploy)(if needed)
```

Status and health related

```
ceph -s
ceph -w
ceph health detail
ceph osd tree
ceph osd df tree
ceph osd df plain
ceph df
ceph osd stat
ceph osd find <osd number>
ceph osd scrub <osd number>
ceph osd deep-scrub <osd number>
ceph osd map <poolname> <objectname>
ceph osd metadata < osd number>
ceph-volume lvm zap /dev/sdx
ceph-deploy disk zap <hostname> <device name>
ceph-deploy osd create <nodename> --data /dev/sdx (V2 ceph-deploy, creates logical volumes with Mimic)
# lvremove /dev/ceph*
ceph-volume lvm create -bluestore --data /dev/sda1
ceph-volume lvm create --osd-id 0 --bluestore --data /dev/sdc --block.db sdb-vg/block-lv --block.wal sdb-vg/wal-lv
```

Stop/remove an OSD

```
ceph osd out <n> (Do on the MON node)
ssh <osdnode> systemctl stop ceph-osd@<n> (Do this on the OSD node)
ceph osd purge <n> --yes-i-really-mean-it (Do this on the MON node)
```

Pool related

```
ceph osd crush rule create-replicated <ssdrule> default <host> <ssd>
ceph osd erasure-code-profile set nvmeprofile0 ruleset k=2 m=1 crush-device-class=<hdd|nvme> crush-failure-domain=host
ceph osd erasure-code-profile get nvmeprofile0
ceph osd pool create < pool-name> <pg-number> <pgs-number> <ruleset>
ceph osd pool create <ECpoolname> <pg> <pgs> erasure <ECruleset>
ceph osd crush rule dump
ceph osd pool application enable <poolname> <appname>
ceph osd pool delete < pool-name> < pool-name> --yes-i-really-really-mean-it
ceph osd pool get < poolname> all
ceph osd pool ls detail
ceph osd pool rename <sourcepool> <destpoolname>
ceph osd pool set <poolname> key value pair
        example ceph osd pool set <poolname> min_size 1
ceph osd reweight-by-utilization < percent>
ceph osd test-reweight-by-utilization < percent>
ceph osd set < flag>
        example ceph osd set noout
ceph tell osd.<n> bench
ceph osd metadata
ceph tell mon.* injectargs --mon-allow-pool-delete=true
```

Pool management

```
rados -p <poolname> put <destobjectname> <sourceobjectname>
rados -p <poolname> ls
rados -p <poolname> get <poolfilename> <localfilename>
rados -p <poolname> rm <objectname>
rados bench -p <poolname> -t <threadcount> -b <objectsize in bytes> <timeinseconds> write|rand|seq
```

Note If –t or –b is not specified then the default is 16 threads at 4MB object size.Prior to reading (rand or seq) - -no-cleanup qualifier must be used

```
rados cppool <source pool> <destination pool>
for i in `rados -p <poolname> ls`; do echo $i; rados -p <poolname> rm $i; done
```

Advanced pg related commands and authentication

```
ceph pg dump
ceph pg <pg id> query
ceph pg ls-by-pool <poolname>
ceph pg ls-by-osd <osd number>
ceph pg dump pools_json
ceph pg map <pg id>
ceph pg <pg-id> list_missing
ceph pg dump_stuck
ceph pg repair <pg-id>
ceph daemon < daemon-name> help
ceph -show-config (warning large output, filter by grep?)
ceph auth ls
ceph auth delete
ceph auth caps
        example ceph auth caps <user> mon 'allow *'osd 'allow *' mds 'allow *'
```

Crushmap

```
-----------------------------------------------------------
# This script sets the default replication ruleset to be OSD on a fresh crushmap
ceph osd getcrushmap -o binarycrushmap
crushtool -d binarycrushmap -o crushmapdecompiled
sed -i 's/firstn 0 type host/firstn 0 type osd/' crushmapdecompiled
crushtool -c crushmapdecompiled -o newbinarycrushmap
ceph osd setcrushmap -i newbinarycrushmap
-------------------------------------------
```

or to leave the replication ruleset untouched

```
ceph osd crush rule create-simple <rulename> default osd firstn # Creates rule on the fly
ceph osd pool create <poolname> <pg> <pgs> <rulename>
```

RBD

```
sudo mkfs.xfs -L iscsiimage0 /dev/rbd0
rbd info -p iscsipool iscsiimage0
sudo fio --filename=/dev/rbd0 --direct=1 --rw=randwrite --bs=4096k --numjobs=16 --iodepth=16 --
runtime=60 --name=OSD-test --nrfiles=30 --output=rbd0.out --size=32G
sudo rbd unmap /dev/rbd0
rbd rm -p iscsipool iscsimage0
```

Cephfs

```
ceph-deploy mds create <mdsnode>
ceh osd pool create <metadatapool> <pg-num> <pgp-num>
ceph osd pool create <datapool> <pg-num> <pgp-num>
ceph fs new mycephfs <metadatapool> <datapool>
ceph mds stat
sudo mount -t ceph <monnode>:6789:/ .mnt/cephfs -o name=admin,secret=`ceph-authtool -p
/etc/ceph/ceph.client.admin.keyring`
```

RGW related

```
ceph-deploy rgw create <rgwnode>
radosgw-admin user create --uid=<username> --key-type=s3 --access-key <accesskey> --secret-key
<secretkey> --display-name="<displayname>"
```

SUSE Enterprise Storage 5

```
salt-run state.orch ceph.stage.0
salt-run state.orch ceph.stage.1
```

```
salt-run state.orch ceph.stage.2
salt-run state.orch ceph.stage.3
salt-run state.orch ceph.stage.4
salt-run disengage.safety
salt-run state.orch ceph.purge
deepsea monitor
```

Parted commands

```
parted -a optimal /dev/nvme3n1 mkpart primary <start> <end>
```

SAR commands

```
sar -b <time interval> <count>
```

Memory stats :

```
sar -r <time interval> <count>
```

Network stats :

```
sar -n DEV <time interval> <count>
```

CPU stats :

```
sar -P ALL <time interval> <count>
```

Disk stats

```
sar -p -d 1
```