



Ceph operations

A DigitalOcean journey

Ceph Days Silicon Valley - 2025/03/25

Alex Marangone - Storage Systems

Contents

- Ceph at DO
- Operations journey
- Community involvement
- Q&A

Quick Stats

65 Ceph clusters

57 Production clusters

8 Staging clusters

250+ PB

Total raw Ceph capacity

12+ PB in our biggest cluster

30,000+

OSDs in the fleet across **1,700+ nodes**



Containers!

- **Initial goal: detach OS release from Ceph version**
 - Removes upgrade headaches
- **No noticeable overhead**
- **Everything still works as if it was non-containerized**

```
#!/bin/bash
TOOLS_CONTAINER="ceph-tools"
if [ -e /usr/bin/ceph ] && [ ! -L /usr/bin/ceph ]; then
    /usr/bin/ceph "$@"
else
    docker exec "$TOOLS_CONTAINER" ceph "$@"
fi
```

```
ExecStartPre=/usr/bin/install -d -m0770 -o ceph -g ceph /var/run/ceph
ExecStartPre=/root/osd_mount_script.sh "%i"
ExecStart=/usr/bin/docker start -a ceph-osd-%i
ExecStop=/usr/bin/docker stop -t 300 ceph-osd-%i
```



```
playbooks
├── common
│   ├── ceph_augment.yml
│   ├── ceph_deploy.yml
│   ├── ceph_post_repave.yml
│   ├── ceph_reconfigure.yml
│   ├── ceph_upgrade.yml
│   ├── ceph_wait_healthy.yml
│   ├── ceph-backfill-storageclass.yml
│   ├── ceph-lock-clear.yml
│   ├── ceph-lock-test.yml
│   ├── ceph-update-services.yml
│   ├── convert-omap-data.yml
│   ├── deploy-ceph-tools.yml
│   ├── filestore-destroy.yml
│   ├── filestore-recondition.yml
│   ├── fio-pre-check.yml
│   ├── hpw-node-maintenance-down.yml
│   ├── import-keyring-from-cmdb.yml
│   ├── install_droplet.yml
│   ├── install_units.yml
│   ├── mute_health.yml
│   ├── node-maintenance-down.yml
│   ├── node-maintenance-up.yml
│   ├── node-poweroff.yml
│   └── osd-compact.yml
```

Deployment

- **Build your own vs use upstream? We built our own**
- **Our use cases are limited vs what Ceph has to support**
- **Allows for tight integration with our internal systems**
 - This allows us to deploy and release a cluster to prod within the same day with time to spare
- **Allows for quicker iteration and deployment**
- **Same tooling can be leveraged for daily operations**
- **Our stack is mostly ansible orchestrated by AWX**



Operations goal: no ssh

```
args := []string{
    "rados",
    "-p",
    poolName,
    "lock",
    "get",
    cephLockObject,
    cephLockKey,
    "--lock-duration",
    "600",
    "--lock-type",
    "exclusive",
    "--lock-description",
    description,
}

_, err := cephtools.Command(args[0], args[1:]...).Output()
if err != nil {
    lockInfo, _ := GetCephLockInfo(poolName)
    lockDesc := lockInfo.Lockers[0].Description
    if lockDesc == "" {
        lockDesc = "Unknown"
    }
    log.Fatalf("Cluster Lock is currently held. Reason: %s", lockDesc)
}
```

- **How do you run any sort of create/update/delete operation safely?**
 - Two persons operating on the same cluster can conflict and create an outage
- **Create automation for all common tasks**
 - Automation check health of the cluster
 - If healthy attempts to grab a rados lock on a uniquely named object
 - Does its thing → release the lock
 - Other tasks will wait for the lock to release before operating



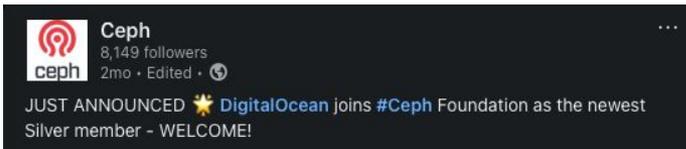
```
2025/03/18 18:03:15 nvme15n1 is already running firmware version [REDACTED]; no
OSD ID: 15
OSD Type: bluestore
OSD Status: down
OSD Created: 17 Mar 25 21:28 UTC
Vendor/Model: [REDACTED]
Serial Number: [REDACTED]
Firmware Version: [REDACTED]
Capacity: 8.0 TB
SMART Status: PASS / 97
Written: 311 TB
Power On Hours: 41145
Mountpoints:
  nvme15n1      disk
  dm-21         lvm
  dm-29         crypt
continue [y/n]?: y
osd.15: marking as destroyed
osd.15: waiting for OSD to be marked destroyed
nvme15n1: stopping Services
nvme15n1: unmounting: /var/lib/ceph/osd/ceph-15
nvme15n1: stopping device mapper dm-29
nvme15n1: wiping Physical Drive
running: ceph-volume lvm zap /dev/nvme15n1 --destroy
nvme15n1: starting Services
running: udevadm trigger --sysname-match=nvme15n1
Checking that the best LBA format size is in use on nvme15n1
executing: nvme id-ns -o json /dev/nvme15n1
nvme15n1: recreating OSDs: [15]
2025/03/18 18:03:35 Running: [ceph-volume lvm batch --yes --bluestore --osds-
osd.15: setting device class based on model [REDACTED] (attempt 0/10)
2025/03/18 18:03:53 Running: [ceph osd crush rm-device-class osd.15]
2025/03/18 18:03:54 Running: [ceph osd crush set-device-class S [REDACTED]]
osd.15: Starting service
osd.15: Waiting For OSD to Come Up
importing 3 upmap entries
```

Operations: disks

- With 30k+ drives proper automation becomes critical
- Storman (now a service) handles the LC of every OSD
 - Wraps ceph-volume
 - Manages firmware updates
 - Does slack alerts
 - Diagnoses failed OSDs
- Crawl → Walk → Run
 - At *first* disk failure would require someone to follow a tedious process
 - Then disk failure would require someone to SSH and run a couple of commands + ticket creation
 - Now disk failure only require human intervention if the disk needs to be physically replaced



Community



- **We're getting more involved and want to do more**
 - We're BACK in the Ceph foundation
 - Ceph performance meetings
 - Crimson testing

- **On Github**
 - [digitalocean/ceph_exporter](https://github.com/digitalocean/ceph_exporter) : prom exporter for ceph
 - [digitalocean/pgremapper](https://github.com/digitalocean/pgremapper) : Make node operations drama-free
 - We're looking at more



Thank You!

Hiring do.co/jobs - amarangone@digitalocean.com

Q&A?