DigitalOcean

Ceph Ops at Scale

Cephalocon 2025

Matt Vandermeulen, Storage Systems

Contents

- Who are we?
- Ceph use at DO
- OS+Ceph
- Cluster Deployment
- Lifecycle Maintenance
- Reflection
- Q&A

Quick Stats

75 Cephs

64 Production clusters
11 Staging clusters

270+ PB

Total raw Ceph capacity

12+ PB in our biggest cluster

~34k

OSDs in the fleet across **>2k nodes**

We use one OSD per disk

Ceph+OS: Packages to Containers

- Previously: Ubuntu Trusty + Luminous packages + Filestore + EC
- Target: Ubuntu Focal + Nautilus
- Path: Many yikes
- Decoupling the OS and Ceph
- Containers allowed us to focus on facilitate a Ceph upgrade
- The OS upgrade could be tackled in a separate effort
- Today: Ubuntu Whatever + Reef Containers + stability

Containers: Gotchas

- **We have a cloud operations team we send many alerts to**
  - These folks operate independently of us
- **We want to treat containers as an implementation detail**
- **External teams do not need to know about packages vs containers**
- **Existing scripts should not need to consider it**
- **We want containers, we don’t want to change habits**

Containers: Getchas

- **We have a `ceph-tools` container we use to shim commands**
  - `for i in ceph rados radosgw-admin …`
  - `/usr/bin/$i == docker exec ceph-tools $i “$@”`
- **`systemd` units to wrap container lifecycle**
  - `ExecStart=/usr/bin/docker start -a ceph-osd-%i`
  - `ExecStop=/usr/bin/docker stop -t 300 ceph-osd-%i`
- **`systemd` target for convenience**
  - `systemctl stop ceph-osd.target`
  - `systemctl restart ceph-mon.target`
- **Operators can now interact with the cluster with no change**

Containers: Today

```
host:~# grep ^NAME /etc/os-release

NAME="Ubuntu"

host:~# docker exec ceph-tools grep ^NAME /etc/os-release

NAME="CentOS Stream"

host:~# apt list --installed 2>/dev/null | grep ceph | wc -l

0

host:~# which ceph

/usr/bin/ceph
```

Automation: Solutions

- **So many options!**
- **cephadm, orch weren't available**
- **rook is available (2016+)**
    - We just adopted containers! We weren't ready for k8s
- **ceph-ansible is available (2016+)**
    - It has to handle everything upstream, very wide scope
    - Hooking into it for our needs might be tricky
    - A lot to evaluate as it changes
- **In house option: storage-cm**
    - Moves as fast as we do
    - Narrow focus on the things that we care about

Automation: Deployment

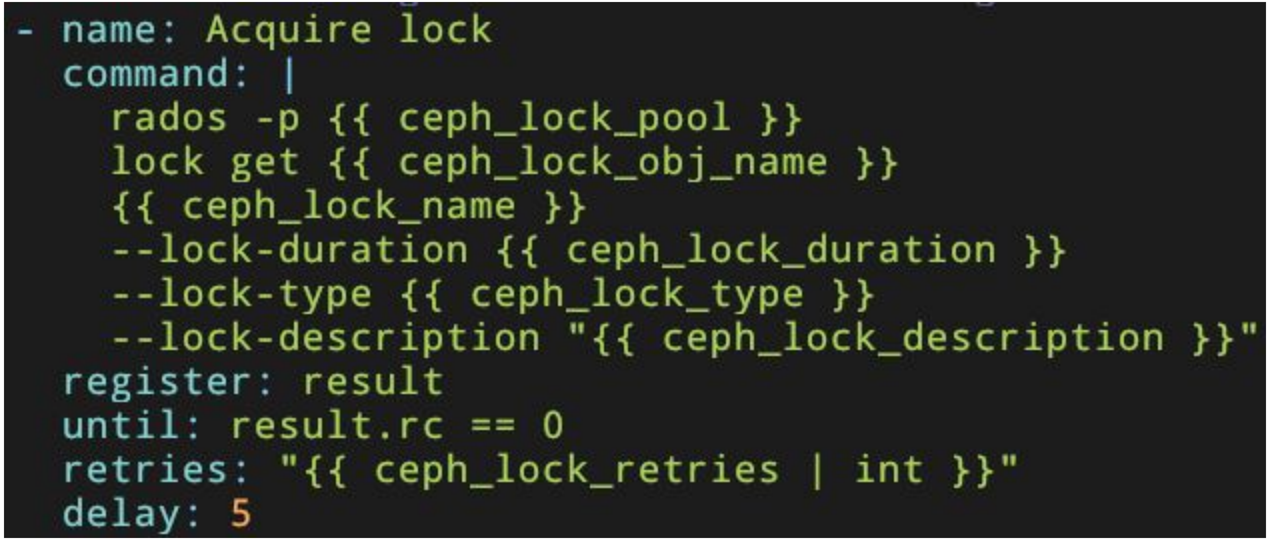
- **We need to support a limited scope, narrow domain**
- **Deployment automation works great for our use case**
  - It does not need to consider anyone else's needs
- **We can enforce checks specific to us before proceeding**
  - Ensure the OS is a specific version
  - We can ensure the kernel is acceptable, NIC FW, etc...
- **CRUSH tree generated for us**
  - Placement data is sourced from our inventory
  - Our inventory is generated from in-house tooling
- **Keyrings managed in house**
- **Deployment is just the start of a cluster lifecycle**

Automation: Maintenance

- **We have routine maintenance to perform through the cluster life**
- **Spiritual goal: No SSH… but, reality… we try**
- **Scaling reboots across the fleet**
  - AMD errata requires <1044d uptime for affected chips
  - When was the last time we read superblocks?
  - Find out about inability to reboot before an incident
- **Automation checks cluster health before proceeding**
  - storage-cm is cluster-aware
  - /usr/sbin/reboot is not
- **If the cluster isn’t healthy - it will wait or fail the job**

Automation: Safety

- If the cluster isn't operationally locked - it will wait or fail the job

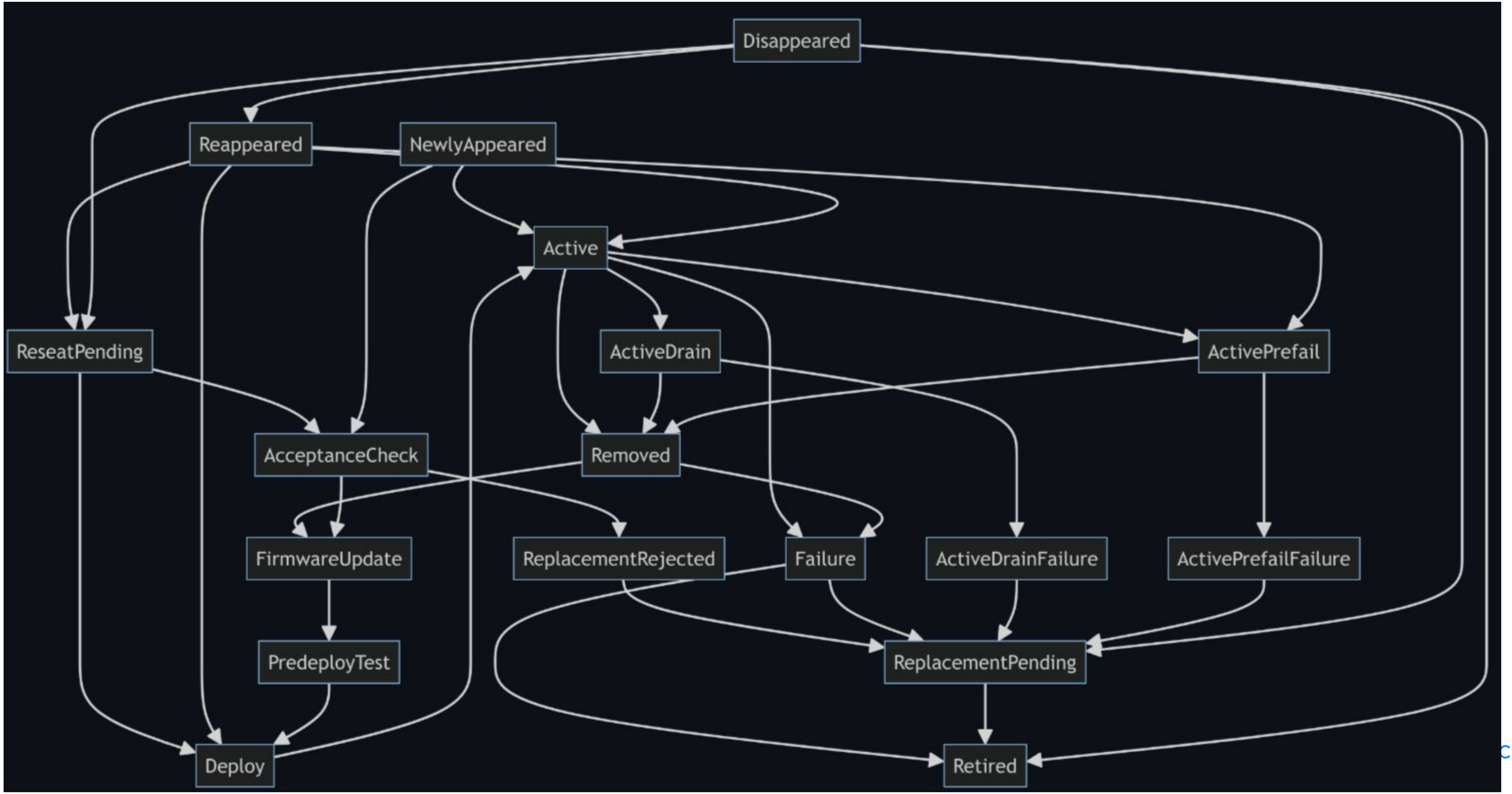


```
- name: Acquire lock
  command: |
    rados -p {{ ceph_lock_pool }}
    lock get {{ ceph_lock_obj_name }}
    {{ ceph_lock_name }}
    --lock-duration {{ ceph_lock_duration }}
    --lock-type {{ ceph_lock_type }}
    --lock-description "{{ ceph_lock_description }}"
  register: result
  until: result.rc == 0
  retries: "{{ ceph_lock_retries | int }}"
  delay: 5
```

Automation: OSDs

- **Concurrency safety becomes important with automated services**
- **The lifecycle of the cluster largely revolves around OSDs**
- **We have a service that manages the OSD lifecycle**
- **Expectation of that state machine:**
  - New disk → OSD → Happy disk noises → ☠️🔥💥 lol rip
- **Reality of the state machine**
  - 👉👈 brace yourself...

Disk Lifecycle State Machine



Automation: Fruits of our Labour

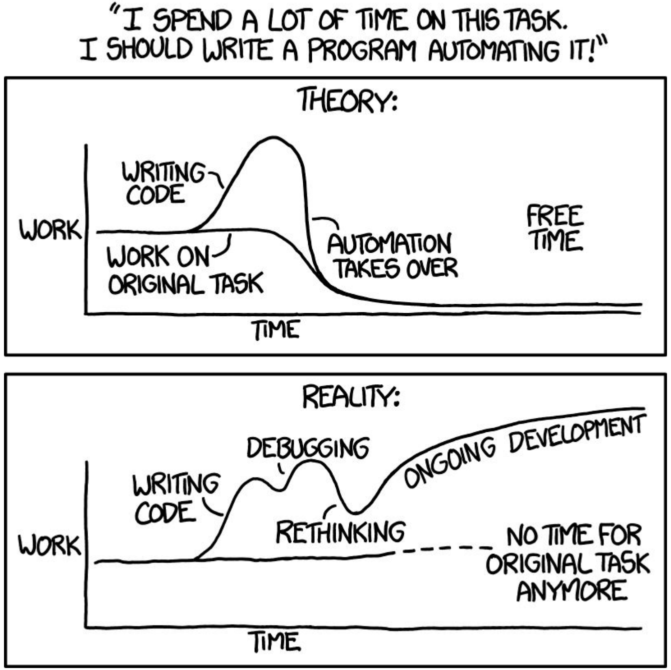
- More than just Ceph
- State transition safety, gated transitions
- CLI tool lets us remotely list OSDs

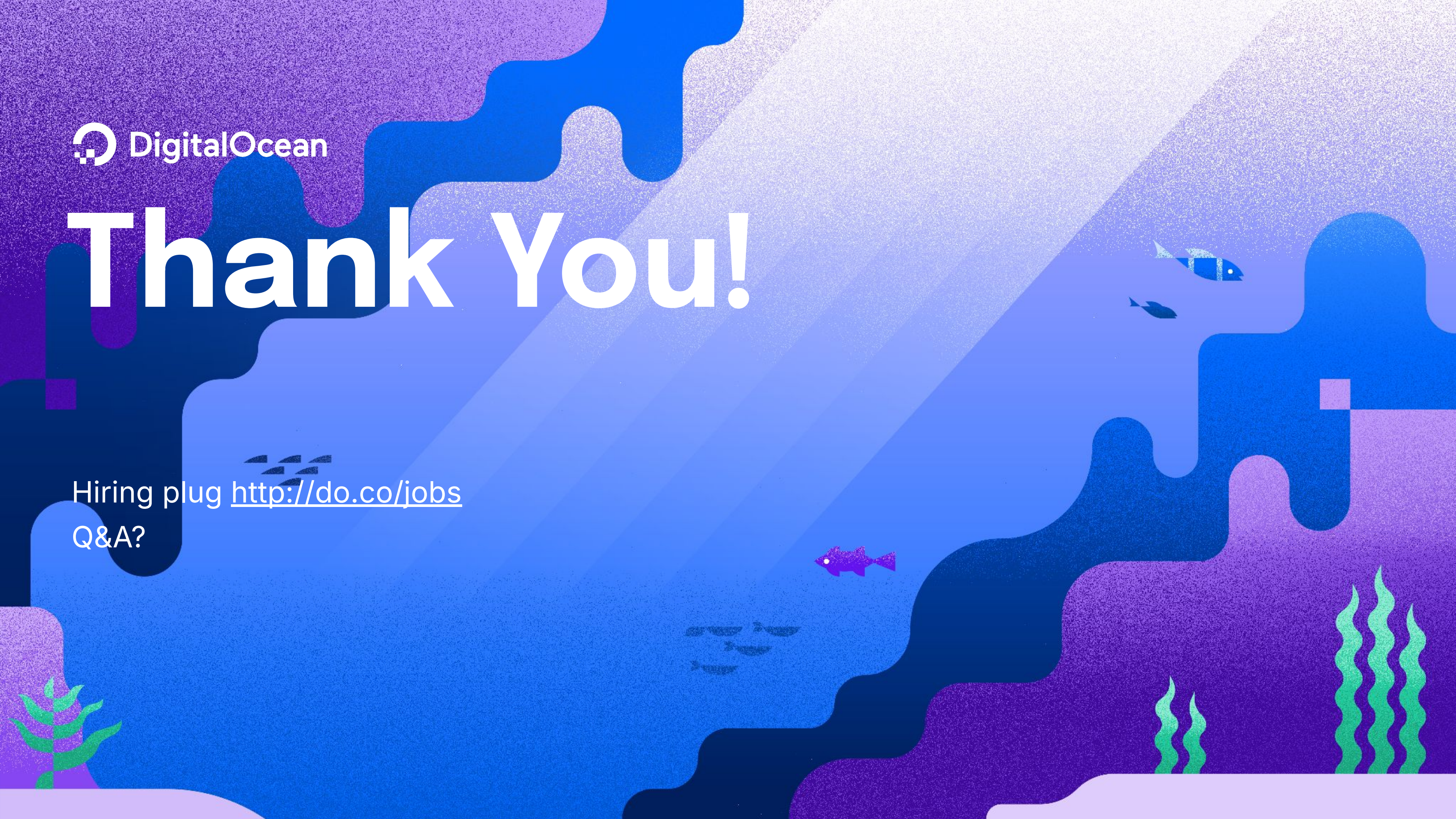
```
$ stormanctl inv list --format json $host | jq '.inventory[] |
  select(.application == "storage-ceph-osd" and .disk.slot == "Slot 5") |
  {
    size: .disk.capacityGib,
    osd: "osd." + ((.appMetadata | @base64d | fromjson |
      .osd_metadata[0].id) | tostring)
  }'

{
  "size": "28615",
  "osd": "osd.7"
}
```

Automation: Rewards

- **This automation buys us time**
- **Allows us to do these talks!**
- **Automation reduces human error**
- **It forces consistency, reduces snowflakes**
  - Produces consistent results
  - Works best on consistent inputs
  - Positive feedback loop





DigitalOcean

Thank You!

Hiring plug http://do.co/jobs

Q&A?