

Migrating Unmanaged OSDs to Managed OSDs in Ceph Squid

Julien Laurenceau | Published 2026-02-05 | Updated 2026-10-01

<https://pepitedata.com/ceph-squid-migrate-unmanaged-osds/>

PepiteData | LinkedIn: Julien Laurenceau | X: @Julien_Lau



Migrating unmanaged OSDs is one of the operations I run for Ceph clients: [Ceph storage consulting](#).

If you are running **Ceph Squid** with a mix of managed and unmanaged OSDs, there is one thing worth stating clearly upfront:

There is **no in-place adoption mechanism** in Squid.

You cannot “import” an existing OSD into the orchestrator. If an OSD was deployed manually or predates your orchestrator setup, the only supported path is **remove and redeploy**.

This is not obvious from the documentation, and it regularly surprises operators during upgrades or cleanup phases. The good news is that the migration is straightforward if you are methodical and patient.

This post documents the exact process I usually follow in production.

Why this matters

Unmanaged OSDs create long-term operational friction:

- inconsistent lifecycle management
- drift between hosts
- harder upgrades
- partial visibility in ceph orch

If you want reproducibility and predictable operations, **all OSDs must be orchestrator-managed**. There is no shortcut.

High-level approach

The migration consists of:

1. Identifying unmanaged OSDs and their backing devices
2. Gracefully evacuating data
3. Removing OSDs cleanly from the cluster
4. Fully wiping the underlying devices
5. Redeploying OSDs using a DriveGroup
6. Verifying cluster health and placement

This is destructive by design. Data safety depends entirely on waiting for rebalancing to complete before moving on.

1. Identify unmanaged OSDs

Start by listing OSD services and daemons:

```
ceph orch ls --service-type osd
ceph orch ps --daemon-type osd
```

Cross-check with the CRUSH tree:

```
ceph osd tree
```

To map OSD IDs to devices and hosts:

```
ceph device ls
ceph osd metadata <osd-id> | grep -e '"hostname"' -e '"devices"'
```

On a given host:

```
listosd=$(ceph device ls |grep $h |grep osd | awk -F '.' '{print $NF}' | awk '{print $1}' |
sort -n)
osd_devices=$(ceph device ls |grep $h |grep osd | awk -F "${h}:" '{print $NF}' | awk '{print
$1}' | sort -n)

for osdid in ${listosd}; do
    ceph osd metadata ${osdid} | grep -e '"hostname"' -e '"devices"'
done
```

Take your time here. Any mistake propagates later.

2. Mark OSDs out and wait

Mark OSDs out **and wait** until the cluster is fully healthy again:

```
for osdid in ${listosd}; do
    ceph osd out ${osdid}
done
```

Do not proceed until:

- no degraded objects
- no reduced data availability

Patience here saves incidents later.

3. Mark OSDs down

Once data has migrated:

```
for osdid in ${listosd}; do
    ceph osd down ${osdid}
done
```

4. Stop OSD daemons on the host

On each affected host:

```
systemctl list-units --no-legend --no-pager "ceph-*@osd.*"
```

```
for osdid in ${listosd}; do
    systemctl stop ceph-${clusterid}@osd.${osdid}
done
```

5. Remove OSDs from the cluster

Remove the OSDs definitively:

```
for osdid in ${listosd}; do
    ceph osd purge ${osdid} --yes-i-really-mean-it
done
```

```
for osdid in ${listosd}; do
    ceph orch daemon rm osd.${osdid} --force
done
```

At this point, the cluster no longer knows about these OSDs.

6. Clean the devices (properly)

This step is where most redeployments fail.

Basic zap:

```
ceph-volume lvm zap /dev/sdX --destroy
# or
sgdisk --zap-all /dev/sdX
```

For multiple devices:

```
for d in ${osd_devices}; do
    sgdisk --zap-all /dev/${d}
done
```

If the zap is not sufficient, **nuke the drives**:

```
for dm in $(dmsetup ls | grep '^ceph--' | awk '{print $1}'); do
    dmsetup remove ${dm}
done
```

```
for vg in $(vgdisplay -s | grep '"ceph-' | awk '{print $1}' | tr -d '"'); do
    vgremove --force $vg
done
```

```
for d in ${osd_devices}; do
    pvremove /dev/${d}
done
```

```
wipefs -af /dev/${d}
sfdisk --delete /dev/${d}
partprobe /dev/${d}
done
```

If Ceph sees any leftover metadata, the redeploy may silently fail.

7. Redeploy using a DriveGroup

Create a DriveGroup YAML per host:

```
cat << EOF > ceph-osd-${h}.yaml
service_type: osd
service_id: osd-${h}
placement:
  hosts:
    - ${h}
data_devices:
  paths:
    EOF

for dev in ${osd_devices}; do
  echo "    - /dev/${dev}" >> ceph-osd-${h}.yaml
done
```

Apply it:

```
ceph orch apply -i ceph-osd-${h}.yaml
```

Optional: set a device class if needed:

```
mydevclass=nvme-all
listosd=$(ceph device ls |grep $h |grep osd | awk -F '.' '{print $NF}' | awk '{print $1}' |
sort -n)

for osdid in ${listosd}; do
  ceph osd crush set-device-class ${mydevclass} ${osdid}
done
```

8. Verify

Final checks:

```
ceph orch ps --daemon-type osd
ceph osd tree
ceph -s
```

You should now have fully managed OSDs with consistent lifecycle control.

Final notes

- There is no shortcut in Squid today
- Plan the maintenance window
- Double-check devices before wiping
- Wait for full rebalance at every step

This process is boring, repetitive, and safe: exactly what you want when touching storage.

If the next step on your roadmap is the major version upgrade itself, see [Upgrading a Ceph multi-site S3 platform from Squid to Tentacle](#).

[Comments on linkedin](#)

Update 2026-09-29. This post is now part of [Ceph in Production: RBD, RGW and CephFS](#), the field guide to Ceph in production.

Related posts

- [Ceph in Production: RBD, RGW and CephFS](#)
- [Upgrading a Ceph multi-site S3 platform from Squid to Tentacle](#)
- [Ceph security update: four CVEs and key rotation](#)
- [Ceph RBD mirroring for Kubernetes storage in production](#)

PDF snapshot of the public article. The online version is authoritative; links remain clickable.