

Toward One-Click Ceph

Engineering the Future of Autonomous Storage

Dan van der Ster | CTO @ CLYSO

Ceph Days Raleigh - 25th March 2026



Agenda

- **Motivation:**
 - Why we need better tools to get to "One-Click Ceph"
- **Defining Safety:**
 - Updates to the Ceph Durability Calculator (and new math!)
- **Seamless Maintenance:**
 - Painless scaling with upmap-remapped (and its future)
- **Safety Nets:**
 - Community tooling for rescuing CephFS metadata
- **The Future:** Putting the puzzle pieces together



From Autonomic to Autonomous

- Ceph's internal "brain" – R.A.D.O.S. – is already *Autonomic* !

Reliable, **A**utonomic **D**istributed **O**bject **S**tore

- *Autonomic* systems feature involuntary, self-regulating processes.
 - They're biological ... they *breathe*. (Our cephalopod metaphor goes deep).
- Ceph is becoming *Autonomous*:
 - Mechanical, independent, self-governing, and acting without external control.
- I want *Full Self-Drive Ceph* – today we have *95% Self-Drive Ceph*.
- This talk is about (part of) that missing 5% – open source tools that we use daily to make our work easier and safer.



Planning for Failure (Without Guessing)

The Durability Calculator Update

- Very large disks mean we must re-evaluate Ceph constants (e.g., PGs per OSD).
- These changes have unexpected consequences for reliability.

20TB+ Disks → New PG Targets → Unknown Risk

We need a way to calculate realistic risks, not just guess.



What's New in the Math?

Simulating the Worst-Case Scenarios

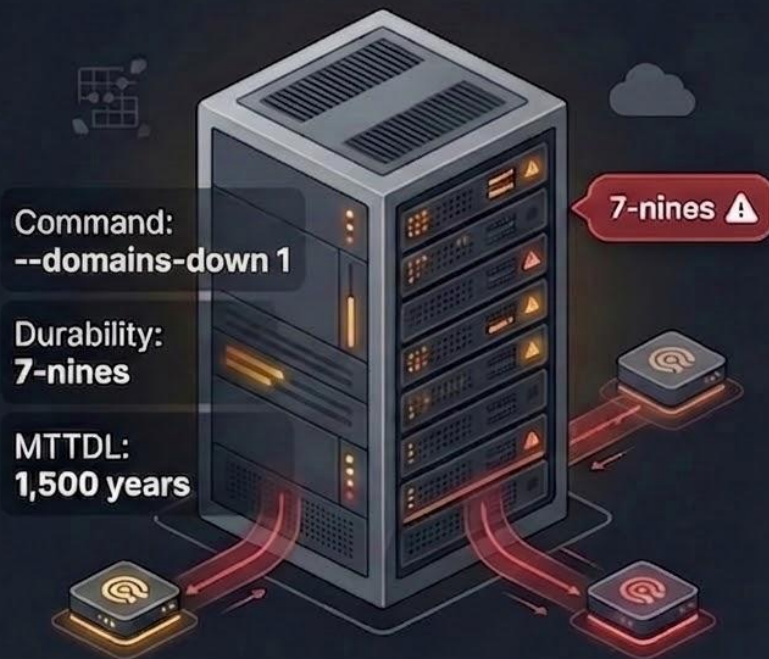
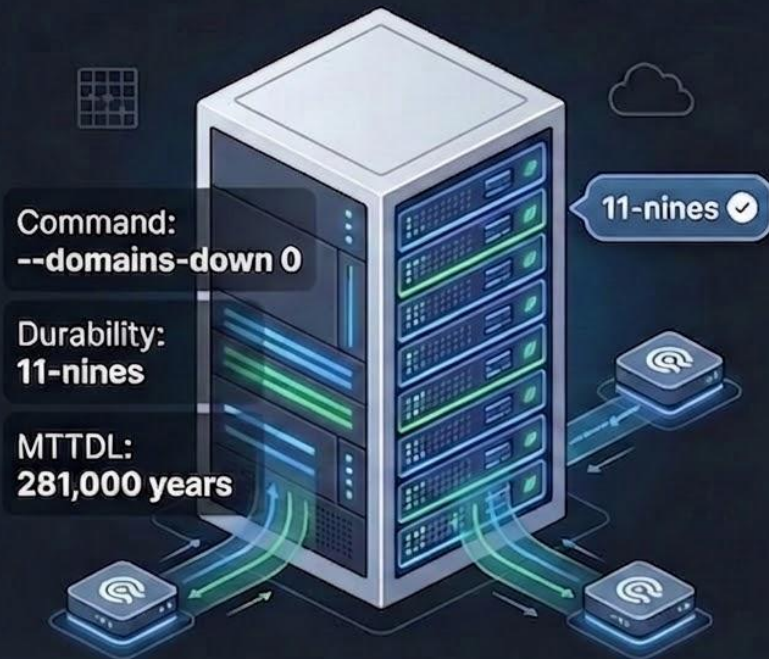
- **Monte Carlo Simulations:** Since Cephalocon, I've overhauled the underlying math, reconciling the differences between theoretical predictions and real-world simulations.
- **The "Already Down" State:** The calculator now accurately models your durability risk if a host or a whole rack is already offline when a secondary failure strikes.
- **Testing Real-World Scenarios:**
 - **Drive Size:** What happens during a correlated rack failure with 20TB vs. 60TB drives?
 - **PG Density:** What is the exact mathematical impact of tripling your pg_num per OSD?



```
$ ./clyso-nines simulate --domains
Simulating a 3x replicated pool with
PGs: 4096 total, averaging 76.8 MB
Data per PG: ~133.3GB, ~34.1K obj
Calculated PG Loss Rate (with 1 pr
Monte Carlo Data Loss Rate: 11.1%
MTTDL: 1.5 thousand years
Durability: 7-nines
```

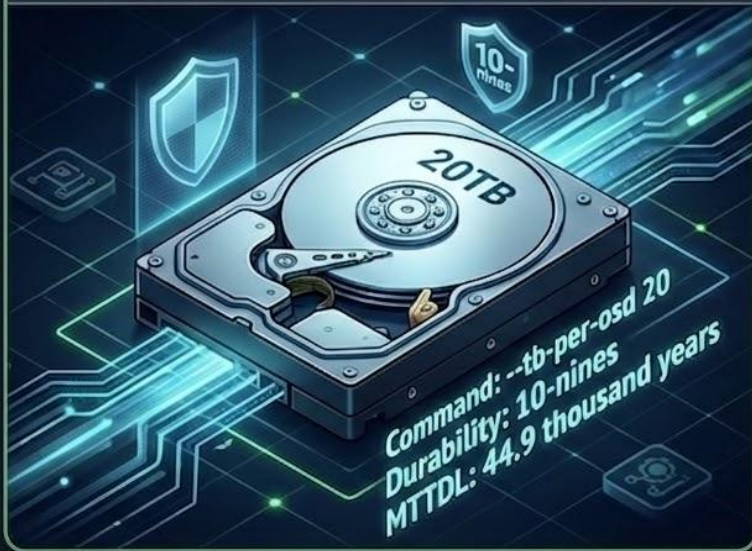
```
$ ./clyso-nines simulate --
Simulating a 3x replicated po
PGs: 4096 total, averaging
Data per PG: ~133.3GB, ~34
Calculated PG Loss Rate (with
Monte Carlo Data Loss Rate:
MTTDL: 281.6 thousand years
Durability: 11-nines
```

cliso-nines comparison: Healthy State vs. Degraded State



The Hidden Cost of Density: Drive Size vs. Recovery Time

The Standard (20TB)



The Ultra-Dense (60TB)



Command: --tb-per-osd 20	Command: --tb-per-osd 60
Durability: 10-nines	Durability: 9-nines
MTTDL: 44.9 thousand years	MTTDL: 9.2 thousand years

Scaling Constraints: Density + Higher PG Num

Simulating a 160 OSD cluster (3x replication, seed 42)

The Baseline: 20TB / 100 PGs



The Challenge: 60TB / 300 PGs



Simulating a 180 OSD cluster (3x replication, seed 42)

Command: --tb-per-osd 20	Command: --tb-per-osd 60
Durability: 10-nines	Durability: 9-nines
MTDL: 49.4 thousand years	MTDL: 9.8 thousand years

Get the Calculator

Prove Your 9s

- Connect the simulator to your live cluster to evaluate the true impact of concurrent failures.
- Test your own architecture and make decisions based on data, not gut feelings.
- Available now:

<https://github.com/clyso/otto/pull/49>

```
$ ./clyso-nines simulate --
```

```
Simulating a 3x replicated pool with  
PGs: 4096 total, averaging  
Data per PG: ~133.3GB, ~34.1K obj  
Calculated PG Loss Rate (with 1 pr  
Monte Carlo Data Loss Rate: 11.1%  
MTTDL: 281.6 thousand years  
Durability: 11-nines
```

```
$ ./clyso-nines simulate --domains
```

```
Simulating a 3x replicated pool with  
PGs: 4096 total, averaging 76.8  
Data per PG: ~133.3GB, ~34.1K obj  
Calculated PG Loss Rate (with 1 pr  
Monte Carlo Data Loss Rate: 11.1%  
MTTDL: 1.5 thousand years  
Durability: 7-nines
```



The Churn is Real

When routine maintenance feels like an outage

- **The Reality of Scaling:** Adding capacity, replacing nodes, or changing CRUSH tunables can trigger massive data movement.
- **The "Leap of Faith":** You type the command, and suddenly 25% of your cluster is misplaced. Your client I/O tanks, recovery takes weeks, and you just have to wait.
- **The Goal:** Maintenance shouldn't cause a client-facing I/O storm. We need a way to skip the panic phase.



upmap-remapped (And a New Trick)

Skipping the data storm

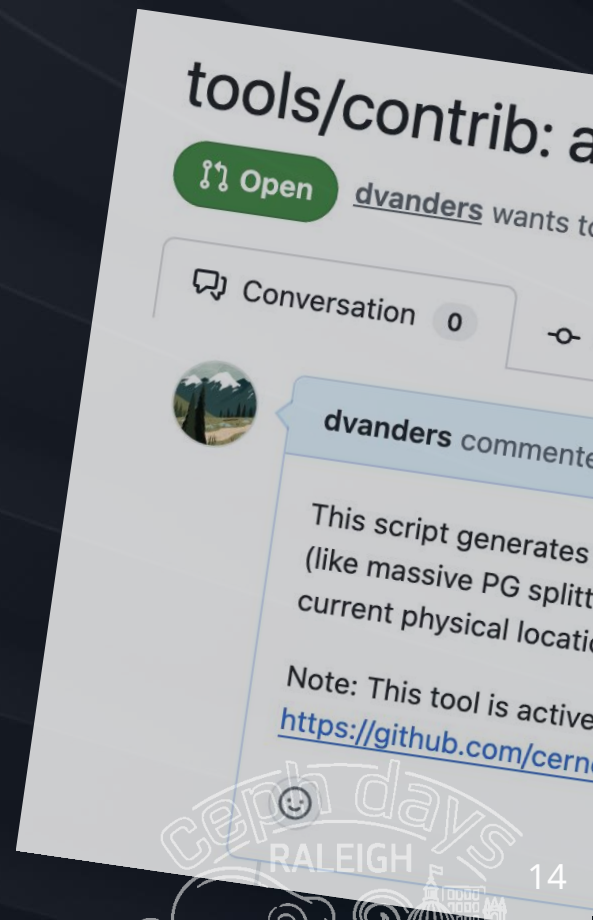
- **How it works:** Instead of letting the cluster instantly try to move everything, upmap-remapped pins the PGs exactly where the data currently lives.
- **Instant HEALTH_OK:** The cluster goes green instantly. You avoid the I/O storm, and the background balancer slowly moves the data over time safely.
- **The New Superpower (PG Splitting):** At Cephalocon, operators told me they use this for PG splitting.
 - I tested it in production: We added **10s of thousands of PGs to multi-PB clusters in minutes**, instead of weeks. It's a massive win.

```
cluster:  
  id:      e7681812-f2b2-41d1-9  
  health: HEALTH_OK  
  
data:  
  pools:   2 pools, 8448 pgs  
  objects: 34.29M objects, 130  
  usage:   391TiB used, 812TiB  
  pgs:     8444 active+clean  
           2   active+clean+s  
           2   active+clean+s
```

Moving Upstream

From a script to a core feature

- **No More "Magic Scripts":** Relying on a downloaded Python script for critical cluster maintenance isn't the final answer.
- **The Upstream PR:** I've submitted upmap-remapped as a PR to be contributed upstream directly into Ceph.
- **The Balancer Roadmap:** This is just step one. We are looking at a roadmap for a completely new balancer that will natively incorporate these ideas, simplifying maintenance even further.



When CephFS Breaks

The manual, high-stress minefield of metadata recovery

- **The Nightmare Scenario:** Your CephFS metadata pool gets corrupted or loses a critical PG. The filesystem is down, and users are yelling.
- **The "Playbook":** Recovery involves a highly complex, multi-step disaster recovery process detailed in the Ceph docs.
- **The Human Element:** People panic. When operators are stressed, they miss crucial steps, execute commands in the wrong order, or skip safety checks.
- **The Result:** One wrong command during metadata recovery can significantly complicate recovery efforts.

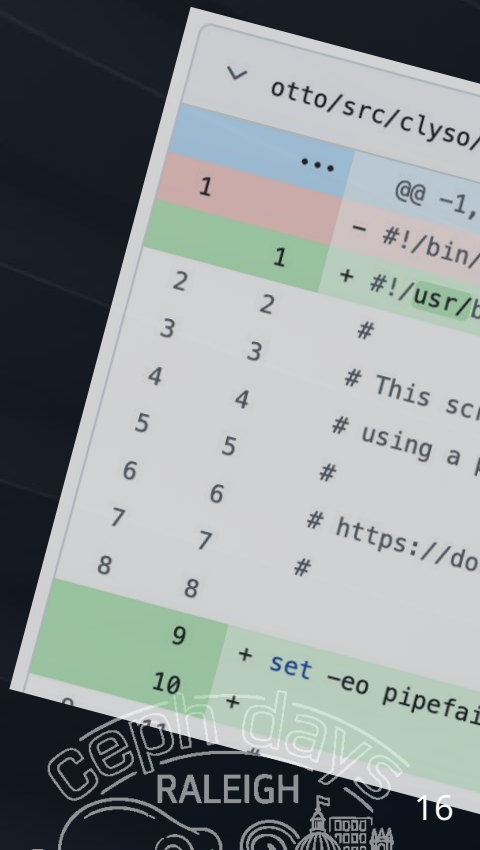
```
# ceph health detail
HEALTH_ERR 1 filesystem is degraded
mds daemon damaged
[WRN] FS_DEGRADED: 1 filesystem is degraded
fs cephfs is degraded
[ERR] MDS_ALL_DOWN: 1 filesystem is offline because no MDS is available
fs cephfs is offline because no MDS is available
[ERR] MDS_DAMAGE: 1 mds daemon damaged
fs cephfs mds.0 is damaged
```



Scripting the Rescue

clyso-cephfs-recover-journal/-metadata

- **Automated Guardrails:** Mykola @ CLYSO maintains these tools which take the error-prone, manual disaster playbooks and wraps them in strict safety checks.
- **Sequence is Everything:** It forces operations to happen in the correct order. It aborts the recovery if anything doesn't go as expected.
- **Structured Recovery:** It turns a multi-hour panic attack into a methodical, step-by-step rescue operation.



Get the Recovery Tools

Don't wait for an outage to figure this out

- **Prepare Now:** Download it, read the code, and understand how it works before you have an emergency.
- **Test it:** Run it in a lab environment to see the recovery sequence in action.
- **Contribute:** We are actively updating this tool (check out PR #68!). Help us make it bulletproof.

Available now:

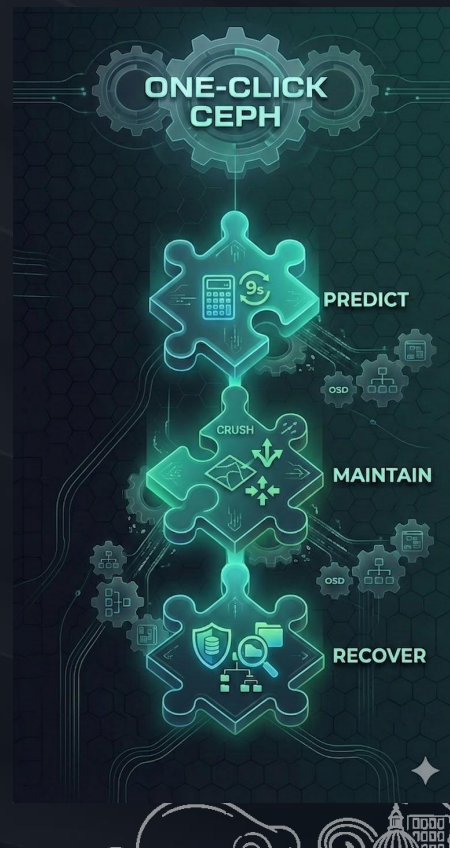
github.com/clyso/otto



Putting the Puzzle Together

The path to "One-Click" Ceph

- **The Goal is an Evolution:** True autonomy isn't a single software update; it's an iterative process.
- **The Building Blocks:** We can't have a self-driving cluster until we can script the math, the maintenance, and the recovery. These tools are those foundational blocks.
- **The Community Effort:** Our job as operators is to find the manual pain points, build the open-source tools to solve them, and eventually teach the cluster to do it itself.



Thank You!

Dan van der Ster | CTO, CLYSO

- **Predict (Durability Calculator):**
github.com/clyso/otto/pull/49
- **Maintain (upmap-remapped):**
github.com/ceph/ceph &
github.com/cernceph/ceph-scripts
- **Recover (CephFS Rescue):**
github.com/clyso/otto/pull/68
- **Let's Talk:** Come find me after to chat about balancer ideas!
- **Email:** dan.vanderster@clyso.com
- **Web:** clyso.com
- **GitHub:** @dvanders / @clyso

