

Upgrading a Ceph Multi-Site S3 Platform from Squid to Tentacle

Julien Laurenceau | Published 2026-07-17 | Updated 2026-09-30

<https://pepitedata.com/ceph-squid-to-tentacle-multisite-s3-upgrade/>

PepiteData | LinkedIn: Julien Laurenceau | X: @Julien_Lau



If your platform faces the same release jump, this is one of the Ceph engagements I run: [Ceph storage consulting](#).

Ceph Squid is approaching the end of its supported life. For an RGW platform serving S3 across several sites, the move to Tentacle is not a routine package refresh. It is a compatibility, replication and client-behaviour change that deserves a controlled rollout.

The reassuring part is that Tentacle brings useful Object Gateway improvements. The risky part is assuming that a healthy-looking cluster proves a healthy multi-site S3 service. It does not. A safe upgrade treats the realm, periods, sync state and critical S3 workflows as first-class acceptance criteria.

What changes on the RGW and S3 side

Tentacle 20.2 adds S3 GetObjectAttributes support, truncates LastModified timestamps to the second for AWS S3 compatibility, and moves most of the bucket-resharding work ahead of the phase that blocks writes, which shrinks the client-visible pause on large buckets. Beyond those headline items the release changes IAM policy conditions, Object Lock enablement and bucket-policy safety, and its multipart and multi-site sync fixes are good reasons to upgrade and to test.

The less visible changes matter just as much on a multi-site platform:

- **IAM policy conditions.** ArnEquals and ArnLike are now supported, with their Not and IfExists variants. Replication policies are validated against the dedicated s3:ReplicateObject,

s3:ReplicateDelete and s3:ReplicateTags actions, so bucket-replication permissions can be scoped precisely instead of granting broad rights.

- **Object Lock.** PutObjectLockConfiguration can now enable S3 Object Lock on an existing versioning-enabled bucket. Useful for compliance workloads, and worth testing before anyone promises it to an auditor.
- **Public-access blocking.** The RestrictPublicBuckets property is now honoured, completing the PublicAccessBlock set.
- **Bucket policy safety.** PutBucketPolicy supports the x-amz-confirm-remove-self-bucket-access header, the AWS guard against locking yourself out of your own bucket.

The release line also carries fixes around multipart responses and multi-site sync. These are good reasons to upgrade. They are also reasons to test. Timestamp-sensitive consumers can observe LastModified values moving backwards during the upgrade. A change in resharding behaviour is welcome, but should be exercised on representative large buckets before it is trusted in production.

What changes at the load-balancing layer

Tentacle also reworks how traffic reaches the gateways: the Beast frontend gains so_reuseport so several RGW instances on one host can share a single TCP port, and cephadm can deploy a per-node HAProxy ingress concentrator with no keepalived and no virtual IP to fail over. That deserves its own line in the upgrade plan, because the zone endpoint list is committed in the period and ingress is realm topology, not local plumbing.

- **Port sharing per host.** The Beast frontend gains a so_reuseport option, so several RGW instances on the same host can share a single TCP port. On large gateway nodes this scales throughput without inventing extra ports or a local proxy.
- **Ingress concentrators.** cephadm can now deploy a per-node HAProxy in front of each node's RGW daemons, declared in the RGW service spec with a frontend port and a monitor port. Traffic is concentrated locally on every gateway node, and scaling the RGW service reconfigures the concentrators automatically. There is no keepalived and no virtual IP to fail over.

The Ceph project has published a two-part series showing where this leads. [Part 1](#) registers each concentrator in HashiCorp Consul with an HTTP health check against the HAProxy monitor port, turning the gateway fleet into a health-aware service catalogue instead of a static backend list. [Part 2](#) puts CoreDNS in front of Consul so that the S3 endpoint, including virtual-host-style bucket subdomains, resolves only to healthy ingress nodes. Failover happens at DNS resolution time, with no VIP to move and no central load balancer to size.

My advice for the upgrade window itself is unchanged: do not redesign ingress in the same change as the version upgrade. But plan for it. In a multi-site deployment the zone endpoint list is committed in the period, so how clients and peer zones reach the gateways is realm topology, not local plumbing. Moving from a VIP-based ingress to concentrators with DNS-based discovery is a topology change with its own baseline, canary and rollback, ideally a few quiet weeks after Tentacle is validated.

Why multi-site makes the upgrade different

Multi-site makes the upgrade different: RGW multi-site is driven by a realm, zonegroups, zones and committed periods, and those configuration objects and the data-sync pipeline are part of the service. Updating gateway binaries without validating them can leave a platform that answers S3 requests locally while replication is delayed, stalled or semantically wrong.

- Keep the topology and period history documented before the change.

- Check metadata and data sync in every zone, not only the master zone.
- Use one controlled write path for the canary, then confirm visibility and reads from remote zones.
- Do not enable a zonegroup feature until every zone declares support for it.
- Keep new behaviour separate from the binary upgrade. A feature gate is a second change.

This matters for features introduced after an older deployment was built. Ceph documents that multi-site zone features must be supported by all zones before a zonegroup can enable them. Existing deployments may need explicit feature enablement after an upgrade. Treat that as a post-upgrade decision, not as an incidental side effect.

A practical Squid to Tentacle upgrade plan

A practical Squid to Tentacle upgrade plan runs in four steps: establish a clean baseline with recorded cluster, realm, period and sync state, build an S3 acceptance suite against real clients before touching production, upgrade in Ceph's supported order with a canary RGW tier, and delay irreversible feature enablement until every daemon and zone is ready. Each step exists to make a replication anomaly attributable to the change that caused it.

1. Establish a clean baseline

Do not begin while the cluster is recovering, backfilling, undersized or incomplete. Record `ceph -s`, `ceph health detail`, `ceph versions`, the active RGW service inventory and the current realm, zonegroup and zone configuration. Capture sync status from every zone and take a database/configuration backup that you have tested restoring.

Also freeze the operational variables that often obscure an upgrade result: planned resharding, large lifecycle changes, bulk ingest, client SDK upgrades and DNS or load-balancer changes. The aim is to make a replication anomaly attributable.

If the cluster still carries unmanaged OSDs from an older deployment, resolve that before the upgrade, not during it. The supported path is documented in [Migrating unmanaged OSDs to managed OSDs in Ceph Squid](#).

2. Build an S3 acceptance suite before touching production

Use real clients and credentials, not only administrative commands. At a minimum, test a versioned bucket, a multipart upload, server-side encryption if used, object tags, copy operations, deletes and delete markers, bucket policies, and lifecycle behaviour. Write through the preferred zone, then read and list from each secondary zone. Verify ETags, object body, metadata, tags and object versions.

For each test, define an expected replication window. Measure the lag before the change so that a post-upgrade result has a meaningful baseline. Keep the test objects and logs long enough to investigate a late sync rather than overwriting the evidence with the next test run.

3. Upgrade in Ceph's supported order

Ceph's Tentacle guide starts from a healthy cluster and upgrades monitors, managers, OSDs, MDS daemons where applicable, then RGW. For `cephadm` deployments, use the orchestrated upgrade and monitor it with `ceph orch upgrade status`, `ceph -s` and `ceph versions`. For package-managed deployments, follow the documented daemon order and verify each stage before moving on.

Use a canary RGW tier where the topology allows it. Drain a small, known client population to the upgraded gateway group, run the S3 acceptance suite across zones, and watch sync queues and error logs. Only then progress to the remaining gateways.

4. Delay irreversible feature enablement

After every daemon is on Tentacle and the service tests pass, you may complete the cluster upgrade with `ceph osd require-osd-release tentacle`. This is an explicit gate, not a command to run early. For multi-site capabilities, verify `supported_features` in every zone and enable a zonegroup feature only after all zones are ready, followed by a committed period update.

If your platform uses server-side encryption and originated before Reef, include Ceph's encrypted-multipart resync guidance in the runbook after all zones are upgraded. It is a multi-site data-correctness concern, not an optional cleanup task.

The rollout gates I would use

The rollout gates I would use are five: cluster health with no degraded placement groups, stable quorum and expected daemon versions, topology matching the recorded realm, period, zonegroup and endpoint baseline, and replication proven healthy from every zone with a cross-zone canary write. An S3 gate covers multipart, versioning, encryption, tags, copy, deletion and policy tests from your application SDKs, and an operations gate covers dashboards, alerts, logs and on-call handover.

- **Cluster gate:** no degraded placement groups, stable quorum, expected daemon versions.
- **Topology gate:** realm, period, zonegroup and endpoints match the recorded baseline.
- **Replication gate:** metadata and data sync are healthy from every zone, including a cross-zone canary write.
- **S3 gate:** multipart, versioning, encryption, tags, copy, deletion and policy tests pass from the SDKs your applications use.
- **Operations gate:** dashboards, alerts, logs and on-call handover are ready before the next rollout step.

Plan rollback honestly

Plan rollback honestly around canary isolation and client re-routing, not version downgrade: Ceph's orchestrated upgrade can be paused, resumed or stopped, but stopping it does not downgrade the cluster back to Squid. That changes the right rollback question. Before starting, decide how you will isolate an unhealthy RGW canary, route clients to a known-good gateway group, pause further rollout and preserve the evidence needed to repair sync.

Keep package, image and configuration rollback procedures documented, but do not present a version downgrade as a guaranteed escape hatch.

The practical takeaway

A Squid to Tentacle upgrade is worth doing for a multi-site S3 platform, but it should be run as a service migration. Upgrade the cluster in the supported order. Prove cross-zone replication with the S3 semantics your workloads depend on. Enable new multi-site features only after every zone is ready. And make the canary and acceptance suite strong enough that the rollout decision is based on evidence, not optimism.

FAQ

Can you roll back from Tentacle to Squid?

No. The orchestrated upgrade can be paused, resumed or stopped, but stopping it does not downgrade daemons that already run Tentacle. Plan rollback as canary isolation and client re-routing to a known-good gateway group, not as a version downgrade.

When should you run `ceph osd require-osd-release tentacle`?

Only after every daemon runs Tentacle and the cross-zone S3 acceptance tests pass. Treat it as the final gate of the upgrade, not an early step.

Do new multi-site features activate automatically after the upgrade?

No. A zonegroup feature can be enabled only when every zone declares support for it, and the change takes effect through a committed period update. Treat feature enablement as a separate change after the binary upgrade.

If your Ceph, MinIO or S3 platform is carrying a costly migration or reliability risk, request a data platform performance audit. I help CTOs and platform teams turn upgrade uncertainty into a focused technical plan, validation criteria and a prioritized risk register.

References

The references behind this upgrade guidance are the Ceph Tentacle release notes and upgrade guidance, the Ceph Object Gateway multi-site documentation, and the Ceph Reef release notes with the encrypted multipart replication guidance. Two Ceph blog posts show where ingress is heading: health-aware load balancing for RGW with Consul, and DNS routing with CoreDNS and virtual-host style bucket access.

- [Ceph Tentacle release notes and upgrade guidance](#)
- [Ceph Object Gateway multi-site documentation](#)
- [Ceph Reef release notes, encrypted multipart replication guidance](#)
- [Ceph blog, part 1: health-aware load balancing for RGW with Consul](#)
- [Ceph blog, part 2: DNS routing with CoreDNS and virtual-host style bucket access](#)

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](#)
- [Migrating unmanaged OSDs to managed OSDs in Ceph Squid](#)
- [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.