Ceph RGW Accounts in Tentacle: Plan the Identity Model

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

https://pepitedata.com/ceph-rgw-accounts-identity-model/

PepiteData | LinkedIn: Julien Laurenceau | X: @Julien_Lau



Ceph Tentacle does not introduce RGW tenants or accounts. Tenants have existed since Jewel. Accounts arrived in Squid. Tentacle clarifies the direction of travel by deprecating tenant-level IAM in favor of accounts. That distinction matters if you run a multi-tenant S3 service and plan to change its identity model during the upgrade.

An RGW version upgrade and an identity migration are separate changes. The first changes binaries and service behavior. The second can transfer bucket ownership, change effective permissions and alter how notification topics appear through the API. I would never combine them in one production window.

Start with the correct RGW hierarchy

The correct RGW hierarchy separates topology from identity: realms, zonegroups and zones define the multi-site topology, tenants give namespace separation for users and buckets with the same names, and accounts own resources, IAM users and roles, aggregate usage and quotas. The terms sound interchangeable until a failure exposes the boundary. They are not interchangeable: RGW topology and RGW identity solve different problems, and a safe design keeps those boundaries explicit.

Layer	What it controls	What it does not provide
Realm	The RGW multi-site replication domain and its periods	Application-level tenant isolation

Layer	What it controls	What it does not provide
Zonegroup	A group of zones, endpoints and supported multi-site features	An IAM or billing boundary
Zone	An RGW deployment that stores and serves data	A customer account
Tenant	Namespace separation for users and buckets with the same names	A first-class IAM organization
Account	Account-owned resources, IAM users and roles, aggregate usage and quotas	A replacement for realms, zonegroups or zones

RGW topology and RGW identity solve different problems. A safe design keeps those boundaries explicit.

Ceph documents tenant isolation as a namespace feature introduced in Jewel. Every user and bucket belongs to a tenant, including the empty legacy tenant. Accounts add a different model. Resources created by account users belong to the account, appear to other authorized users in that account and contribute to account-level statistics and quotas.

What changes when a user joins an account

When a user joins an account, bucket ownership moves to the account, membership becomes permanent, and the user starts with no permissions until identity policy is added. The Ceph account documentation lists consequences that belong in the migration plan, not in release notes that nobody reads during the change window: notification topics need separate work, names are validated differently, and tenant placement still matters.

- **Bucket ownership moves.** Adopting an existing user into an account transfers ownership of all that user's buckets to the account.
- **Membership is permanent.** Ceph states that a user cannot be removed from an account after adoption.
- **Permissions change.** Account users start with no permissions by default. You must add identity policy before the migrated user can resume normal API operations.
- **Notification topics need separate work.** Existing topics keep working, but ownership does not transfer. The user can lose visibility through the SNS Topic APIs until topics and bucket notification references are migrated.
- **Names are validated differently.** IAM user naming rules apply during adoption. A display name that worked for a legacy RGW user can fail account migration.
- **Tenant placement still matters.** A tenanted account can contain only users from the same tenant.

These are data-model changes. Replication health alone cannot prove that the new ownership and policy model is correct.

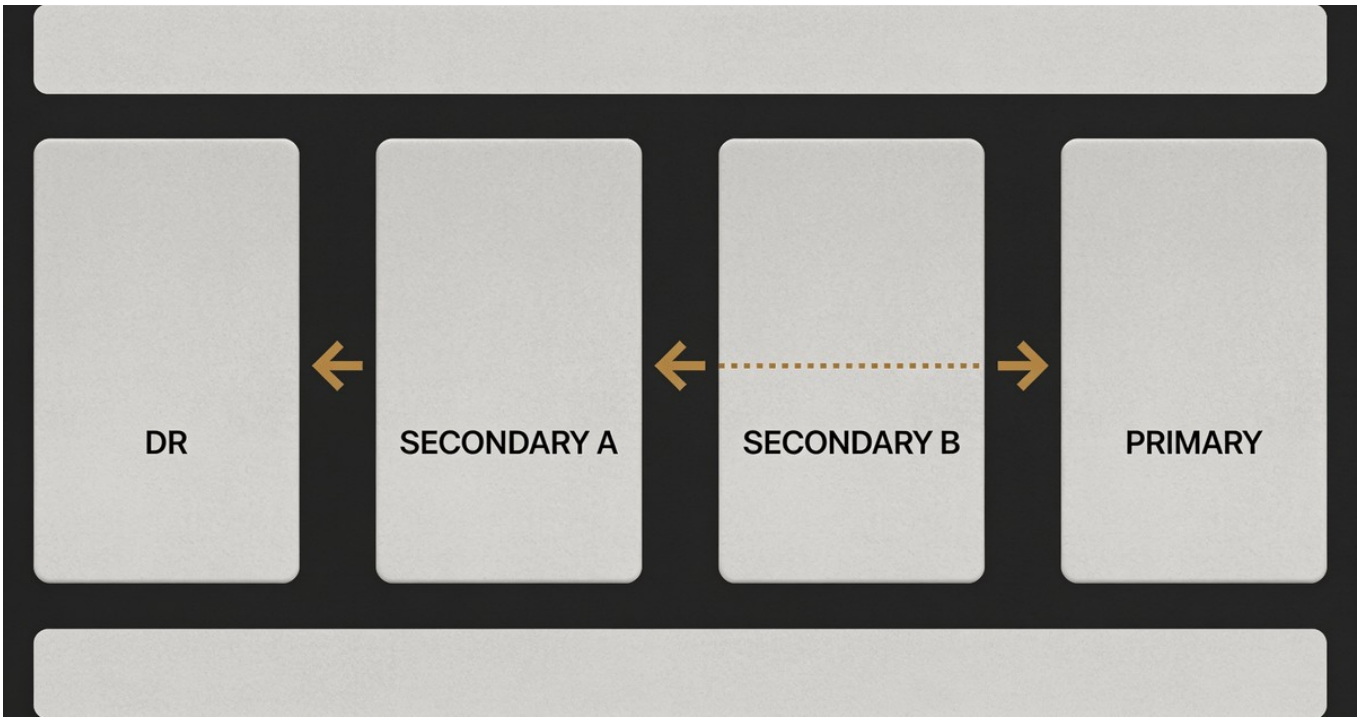
Separate the Tentacle upgrade from account adoption

Separate the Tentacle upgrade from account adoption: complete the binary upgrade first with a canary, cross-zone S3 tests and explicit rollout gates, then begin account adoption only after every zone is stable, as a second project with its own rollback boundary. Mixing the two removes the cleanest diagnostic boundary you have, because a failure can no longer be attributed to the version change or to the identity change.



Tenant and account are different identity doors; users queue at the wrong one unless the model is planned.

I would first complete the binary upgrade using a canary, cross-zone S3 tests and explicit rollout gates. I cover that process in Upgrading a Ceph Multi-Site S3 Platform from Squid to Tentacle. Only after every zone is stable would I begin account adoption as a second project with its own rollback boundary.



Finish and validate the version rollout before changing the identity model. Mixing the two removes the cleanest diagnostic boundary.

A migration sequence that keeps evidence

A migration sequence that keeps evidence runs in order: inventory every tenant, user, bucket, policy, quota and notification, create the target account in the correct tenant, build the identity policies before

adoption, and rehearse with a representative non-production user. Then adopt one low-risk production user, apply the planned policies, run client-level S3 acceptance tests, migrate notification topics where used, and verify metadata and data sync from every zone before the next migration.

1. Inventory tenants, users, buckets, ACL owners, IAM policies, quotas, notification topics and bucket notification configurations.
2. Create the target account in the correct tenant. Record its account ID and root user.
3. Build the identity policies that the adopted user will need. Do not wait for the migration to reveal missing rights.
4. Rehearse the procedure with a representative non-production user, including bucket listing, object access and notification delivery.
5. Adopt one low-risk production user. Confirm that bucket ownership moved to the account and that the account can see the expected resources.
6. Apply the planned identity policies and run client-level S3 acceptance tests.
7. If notifications are used, migrate topics and bucket references according to Ceph's procedure. The `notification_v2` zone feature must be enabled where required.
8. Verify metadata and data sync from every zone, then hold the next migration until the observation window closes.

The gates I would put in the runbook

The gates I would put in the runbook are five: inventory, ownership, authorization, notifications and multi-site, each with client-visible evidence to collect and a stop condition that halts the migration. A healthy Ceph status is necessary, but it is not an identity acceptance test: each gate proves the change with real client credentials and real S3 results, not with cluster health alone.

Gate	Evidence to collect	Stop condition
Inventory	User, bucket, policy, quota and notification export	An owner or dependency cannot be mapped
Ownership	Bucket owner and `ListBuckets` results before and after adoption	Unexpected bucket visibility or ACL owner
Authorization	Positive and negative API tests using real client credentials	A required action fails or a forbidden action succeeds
Notifications	Topic list, bucket configuration and delivered canary event	A topic becomes invisible or delivery duplicates
Multi-site	Metadata sync, data sync and cross-zone client tests	Lag exceeds baseline or account metadata diverges

Each gate needs client-visible evidence. A healthy Ceph status is necessary, but it is not an identity acceptance test.

The design decision comes before the command

The design decision comes before the command: decide how customers, business units, billing and operational ownership map to accounts and tenants before adopting the first user. Accounts are the better foundation when you need delegated IAM, account-owned resources and aggregate quotas, and tenants still matter when you need namespace separation. Account membership is permanent and ownership changes propagate across the service.

If you are planning a Ceph RGW upgrade or account migration, I can review the topology, identity model, acceptance tests and rollout gates through a Flash Architecture Review or Data Platform Audit. Book a 15-min intro call to discuss the platform and the scope.

Primary documentation

- Ceph Tentacle release notes
- Ceph RGW user accounts
- Ceph RGW multi-tenancy

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

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