



Backblaze B2 for CEPH

Add an Infrequent Access Tier Without Building the Infrastructure

The Problem

You run CEPH. Your object storage is fast, scalable, and under your control. The problem is that not all of your customers' data belongs on your primary storage tier, and they know it.

AI training datasets, media archives, backup repositories, surveillance footage — this data is accessed unpredictably. Sometimes daily, sometimes never. It does not belong on your highest-performance capacity, but customers still need it available on demand when they do reach for it.

Offering a separate infrequent access tier means building the infrastructure behind it. Separate capacity pools, lifecycle orchestration, a retrieval layer your customers' applications can hit without changes — that is months of engineering and an ongoing operational surface you did not plan to own.

The alternative is to back that tier with Backblaze B2, using CEPH's own lifecycle machinery to move data in and out transparently.

The Solution

Backblaze B2 is a validated S3 endpoint for CEPH's native Cloud Transition and Cloud Restore services. Configure a lifecycle policy, point it at a B2 bucket, and your platform has a working infrequent access tier today — built entirely on CEPH's own lifecycle machinery, with no proprietary agents or sidecars running on your cluster.

Backblaze engineers resolved the S3 compatibility issues that previously blocked CEPH integration and proved the solution end to end in production. You deploy on a path that has already been debugged.

In Production

[A leading AI cloud platform](#) built and shipped their infrequent access storage tier on this solution. Objects transition from CEPH clusters to Backblaze B2 via lifecycle policy, restore transparently on customer access, and usage reconciles cleanly per tenant across multiple regions through one-to-one bucket mapping. They launched a billable infrequent access tier without building separate storage infrastructure.

Built For

- Neocloud and AI cloud platforms that run CEPH and want to offer an infrequent access tier as a product
- Cloud providers whose customers hold large volumes of unpredictably accessed data — AI datasets, media, backups, surveillance
- Engineering teams that need to ship a second storage tier fast without owning additional infrastructure

Key Capabilities

- One-to-one bucket mapping for per-tenant auditing and billing reconciliation
- Read-through restore with configurable wait time

Pricing

Storage	\$6.95/TB/month
Egress	3X storage amount free, then \$0.01/GB/month

Build vs. Use Backblaze

	Backblaze B2 + CEPH	Build Your Own IA Tier
Time to market	Weeks	Months of engineering
Infrastructure to own	None	Separate capacity pools, retrieval layer
Multi-tenant billing	One-to-one bucket mapping, daily usage CSV	Custom instrumentation required
Customer restore experience	Standard S3 GET, no app changes	Depends on implementation
Ongoing ops burden	Backblaze operates the IA tier, your team continues to maintain CEPH	Your team owns it

How It Works

Archive

1. **Discover.** Lifecycle worker scans the bucket index and identifies objects matching your archival policy: age, prefix, tag, or size.
2. **Read.** Eligible objects are read from your RADOS pool.
3. **Send.** Object uploaded to Backblaze B2 (single PUT or multipart for larger files). Backblaze confirms with ETag and HTTP 200.
4. **Replace.** Local object replaced with a zero byte stub. Head metadata kept, data tail removed. Capacity reclaimed.
5. **Index.** Bucket index updated to the new storage class. Client GET and HEAD now serve from stub metadata. Listings unchanged.

Restore

1. **Trigger.** For typical files, a standard S3 GET starts the restore (read through). For larger files where read through can't complete in time, customers use the S3 RestoreObject API to kick off an explicit restore with a single day parameter.
2. **Locate.** CEPH reads the stub manifest, finds the object in Backblaze, and determines whether the restore is temporary (days specified) or permanent.
3. **Fetch.** Object pulled from B2 using ranged reads and streamed back into local RADOS. Head and tail objects rebuilt.
4. **Verify and index.** ETag and size verified against the cloud manifest. Manifest and bucket index updated with the new storage class and restore expiry.
5. **Cleanup.** If temporary, lifecycle reaps the local copy at expiry; the Backblaze copy is retained. If permanent, the object stays local.

Get Started

Talk to our Solutions Engineering team.

We will walk you through how the integration works against your specific CEPH version and cluster setup, help you scope the onboarding, and answer the technical questions that come up before you commit. Start with a conversation.

sales@backblaze.com — backblaze.com

© Backblaze, Inc.

