CL'SO

Powered by ceph

Beyond Backup: S3 Data Management with Ceph RGW tiering and Chorus

Manage object data like real teams do: hot, warm, cold, archive and Chorus

Sirisha Guduru
Senior Ceph Engineer

Artem Torubarov
Software Engineer

CL'SO

Backups alone are not enough

- Object data grows, access patterns age. Cost, latency, compliance all matter.
- Snapshots and backups do not answer day two needs like lifecycle migrations, partial recalls, or selective restore
- Teams need policy driven flows that move data automatically and bring it back when apps need it

CL'SO

What will we cover

- Multisite in one slide for context
- Ceph RGW tiering model: placement targets, storage classes, lifecycle transitions
- Tape and external archive via S3 compatible gateways (PoINT) with cache tiering
- Retrieval patterns: read through GET, S3 RestoreObject, policy driven prefetch
- Chorus for large S3 moves and cutovers with resumable parallel transfers

CL'SO

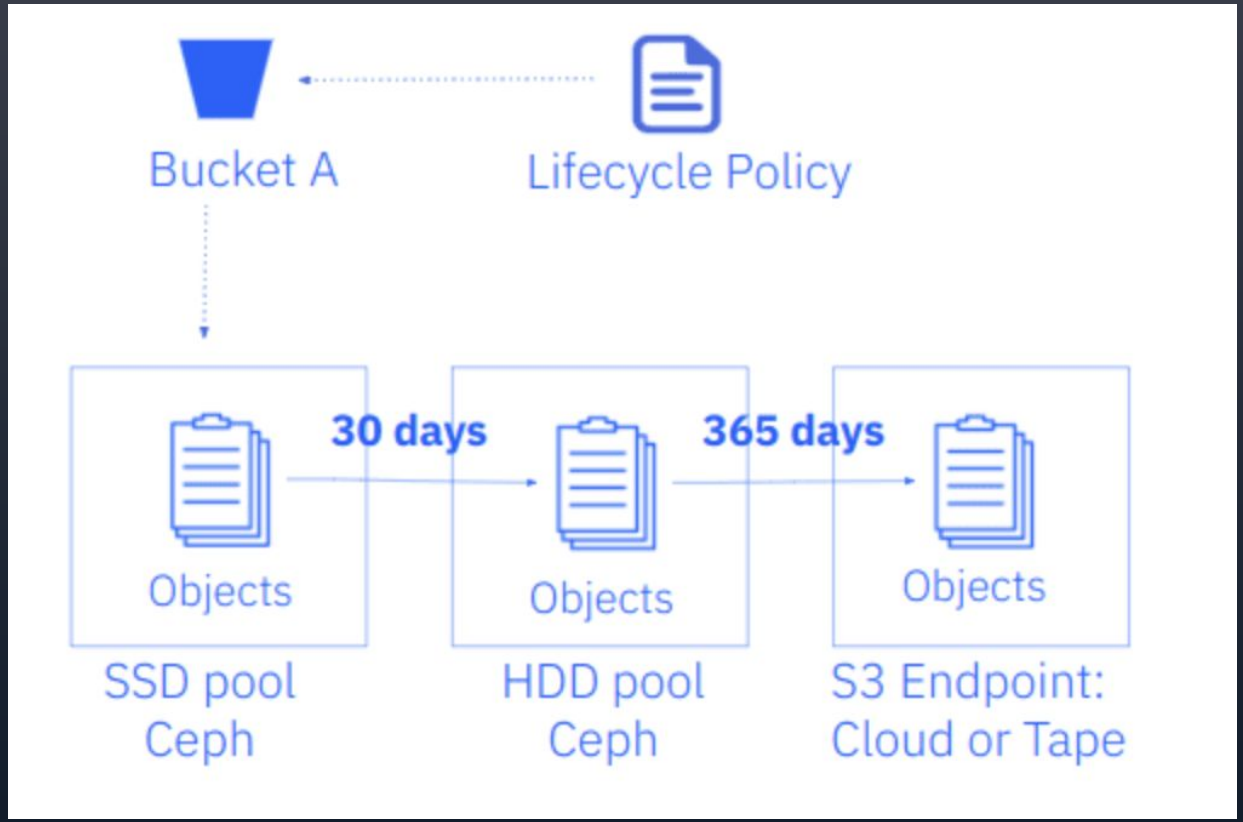
Multisite

- Use multisite for geo isolation and DR
- Lifecycle and placement apply within a zone; multisite replicates object metadata and data across zones
- For this session, focus is on in zone tiering plus external archive target

RGW Tiering Overview

- Placement targets map a bucket or object class to pools
- Example: HOT on NVMe, WARM on HDD, COLD on HDD with erasure coding
- Lifecycle policies drive transitions by age, tags, or prefixes
- Storage classes give clients explicit control on PUT or through lifecycle transitions

RGW Tiering Overview



Design a practical Tier plan

- Tiers can be on-cluster (NVMe, HDD, EC) and off-cluster (cloud or tape S3 endpoints)
- HOT: .rgw.buckets.data.ssd + .rgw.buckets.index.ssd (replicated)
- WARM: .rgw.buckets.data.hdd + index on HDD
- COLD: Erasure coded pool for large objects, larger stripe size
- ARCHIVE: External S3 endpoint backed by tape library emulator (PoINT Archival Gateway) or actual tape

External Archive with Tape (PoINT)

- Use a PoINT S3 compatible archival gateway in front of tape library or emulator
- RGW Cloud Sync to PoINT endpoint for objects that reach ARCHIVE class
- Retrieval
  - RestoreObject to stage from tape, then read through GET back into Ceph cache placement

CL'SO

Steps for Archival tier configuration

- Create new storage class for Cloud-S3 tiering
- Configure new storage class with tier configurations
- Apply lifecycle policies

Lifecycle Policies in use

- Expiration
- noncurrent version expiration
- abort incomplete multipart
- transitions
- size filters
- prefix/tag filters

Lifecycle Policy Examples (S3 JSON)

Bucket Lifecycle:

```
{
"Rules": [
        {"ID": "to-warm-30d", "Filter": {"Prefix": ""}, "Status": "Enabled", "Transitions": [{"Days": 30, "StorageClass": "WARM"}]},
        {"ID": "to-cold-90d", "Filter": {"Prefix": ""}, "Status": "Enabled", "Transitions": [{"Days": 90, "StorageClass": "COLD"}]},
        {"ID": "to-archive-180d", "Filter": {"Prefix": ""}, "Status": "Enabled", "Transitions": [{"Days": 180, "StorageClass": "ARCHIVE"}]}
        ]
}
```

Tag-based:

```
{
"Rules": [
        {"ID": "logs-fast", "Filter": {"Tag": {"Key": "tier", "Value": "hot"}}, "Status": "Enabled"},
        {"ID": "logs-cold-30d", "Filter": {"Tag": {"Key": "tier", "Value": "cold"}}, "Status": "Enabled", "Transitions": [{"Days": 30,
        "StorageClass": "COLD"}]}
        ]
}
```

CL'SO

Tier target for Archival (Tape simulator)

```
"tier_targets": [
    {
        "key": "CLOUDTIER",
        "val": {
            "tier_type": "cloud-s3",
            "storage_class": "CLOUDTIER",
            "retain_head_object": true,
            "s3": {
                "endpoint": "http://172.16.2.6:4080",
                "access_key": "FA7FAF9532E61C6367CC",
                "secret": "Vo/XP0VEKKh9jELE1WGMyohQkHsAYk1ms7XOSg8I",
                "region": "",
                "host_style": "path",
                "target_storage_class": "",
                "target_path": "",
                "acl_mappings": [],
                "multipart_sync_threshold": 44432,
                "multipart_min_part_size": 44432
            }
        }
    }
]
```

CL'SO

Lifecycle Policy to move to tape

```
{
  "Rules": [
    {"ID": "Transition objects from Ceph to cloud storage tier that are older than 60 days",
      "Prefix": "",
      "Status": "Enabled",
      "Transitions": [
        {
          "Days": 60,
          "StorageClass": "CLOUDTIER"
        }
      ]
    }
  ]
}
```

CL'SO

Retrieval Patterns

- S3 RestoreObject API
  - Temporary Restore
  - Permanent Restore

- Read through GET: App issues GET; if object is in archive, RGW or gateway recalls and rehydrates into HOT or a configured cache placement

Retrieval Patterns

- S3 RestoreObject API
  - Temporary Restore
    - aws --profile ceph --endpoint http://10.0.12.16:80 s3api restore-object --bucket testbucket --key testobject --restore-request Days=3

Retrieval Patterns

- S3 RestoreObject API
  - Permanent Restore
    - aws --profile ceph --endpoint http://10.0.12.16:80 s3api restore-object --bucket testbucket --key testobject --restore-request {}

Retrieval Patterns

- Read through GET:
  - --tier-config=allow_read_through=true,read_through_restore_days=3

Chorus

Use-cases beyond S3 lifecycle policies

What is chorus

1. Open source [Apache 2.0]

What is chorus

1. Open source [Apache 2.0]
2. Vendor-agnostic

What is chorus

1. Open source [Apache 2.0]
2. Vendor-agnostic
3. Data management software

github.com/clyso/chorus

What is chorus

1. Open source [Apache 2.0]
2. Vendor-agnostic
3. Data management software
4. For Object storage

CLYSO

What is chorus

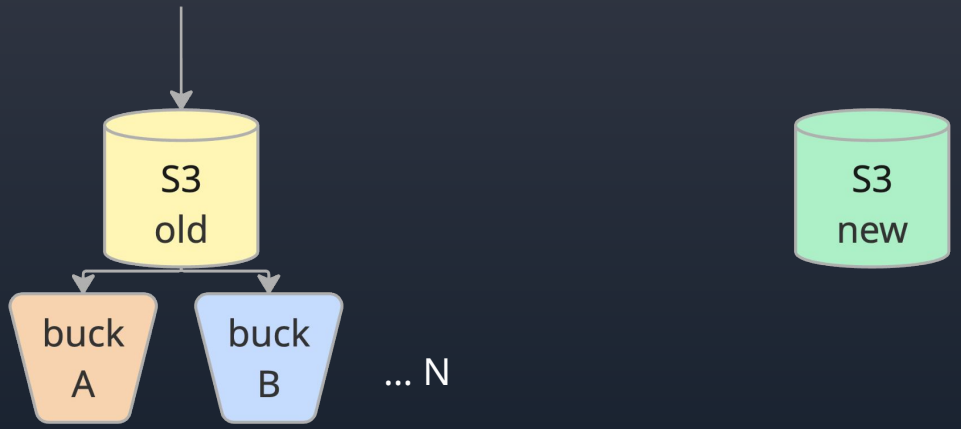
1. Open source [Apache 2.0]
2. Vendor-agnostic
3. Data management software
4. For Object storage

Automates backup and migration

ANY ⇔ ANY*

CL'SO

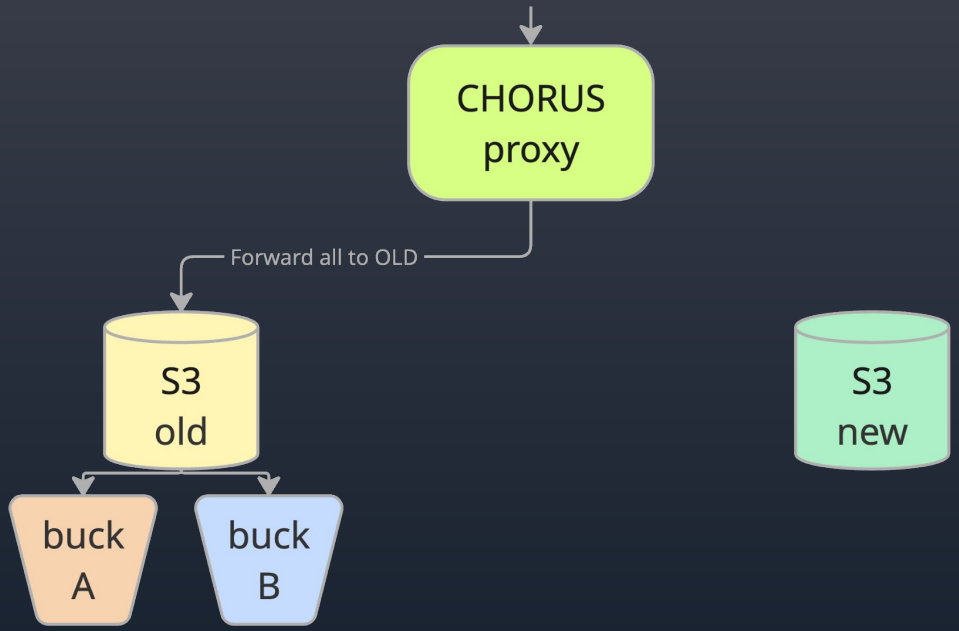
S3 migration with chorus



- Live traffic
- 100*X buckets
- 1M*X objects
- Pb*X data

CL'SO

S3 migration with chorus

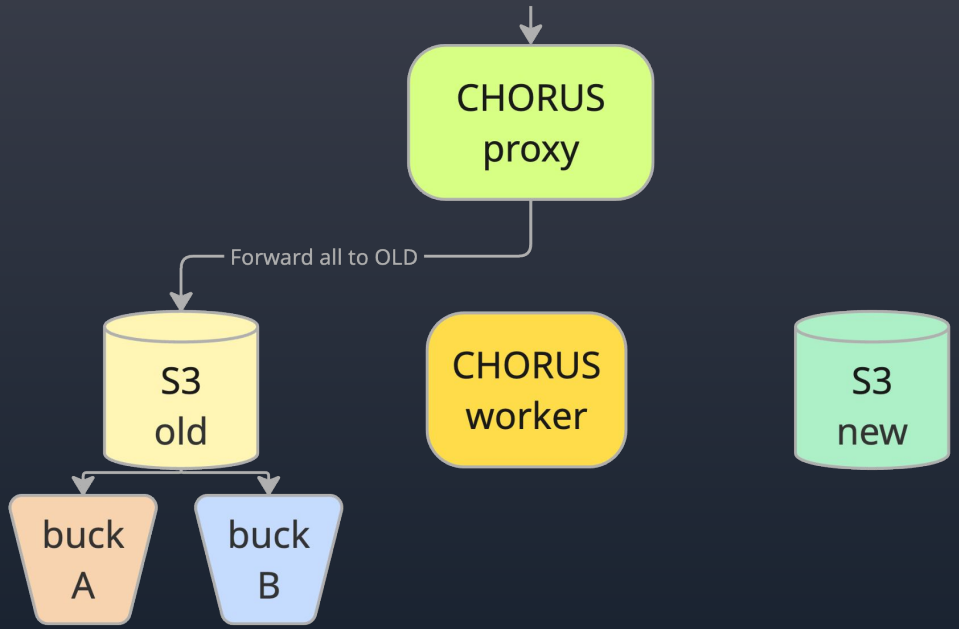


Chorus S3 proxy:

- Route to Old | New
- Block request
- ^ All by Bucket ^

CL'SO

S3 migration with chorus

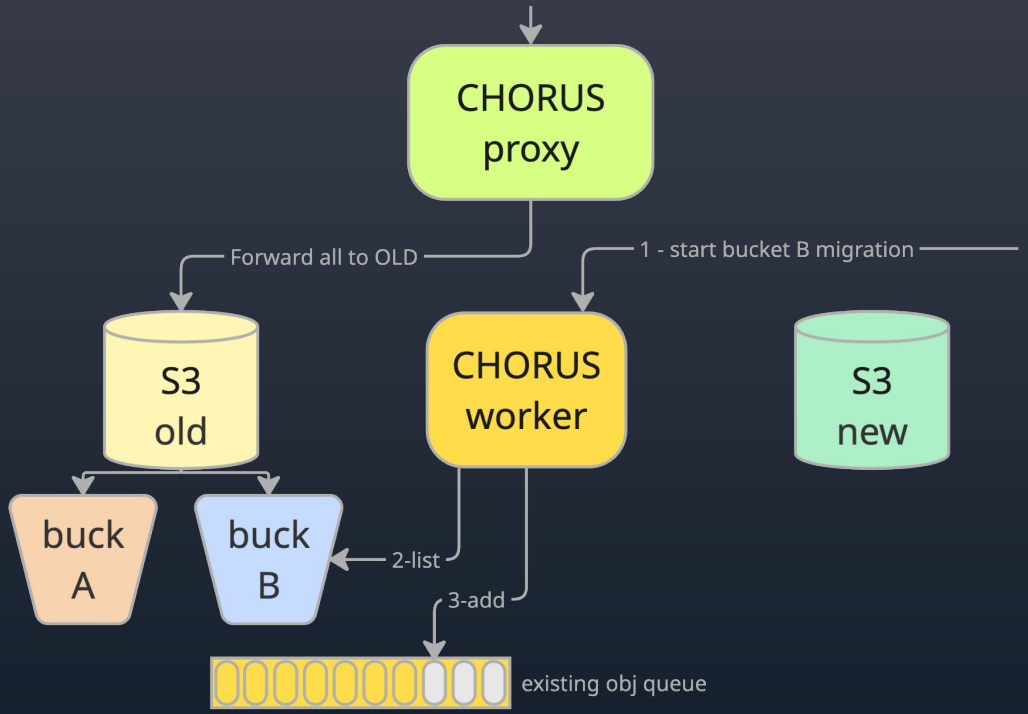


Chorus worker

- Multiple machines
- Checkpoints progress
- Horizontally scalable

CL'SO

S3 migration with chorus

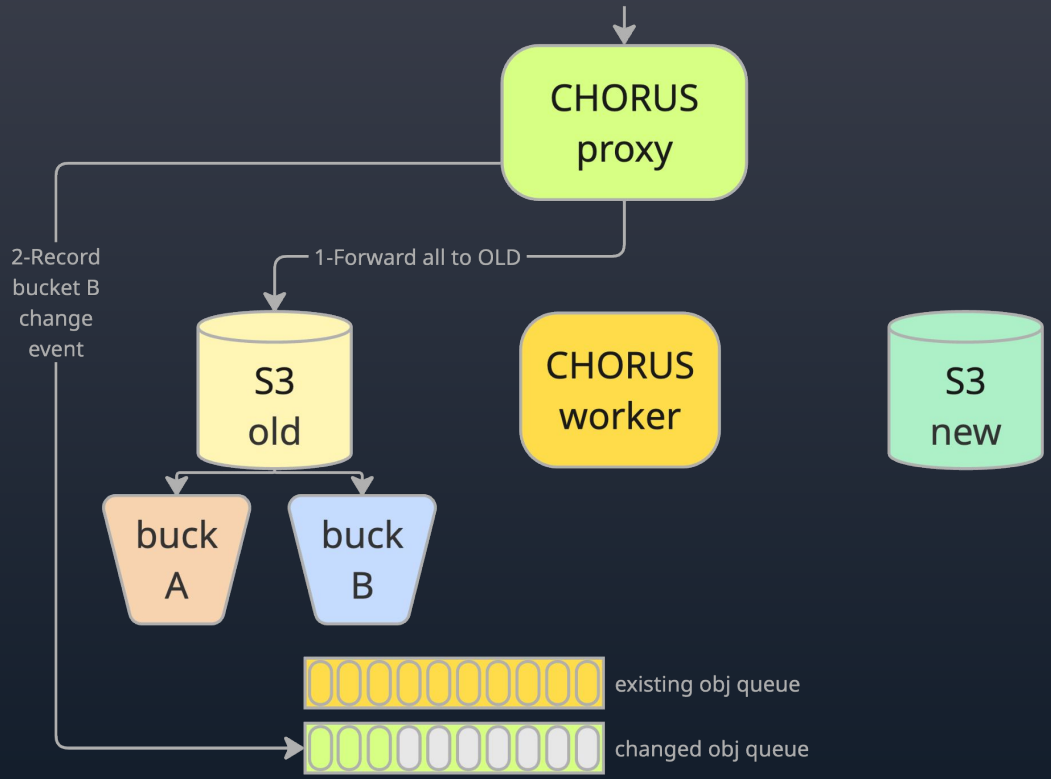


Manage with:

- API | CLI | Web-UI
- Single OR All user buckets
- Pause/resume
- Restart
- Get status

CL'SO

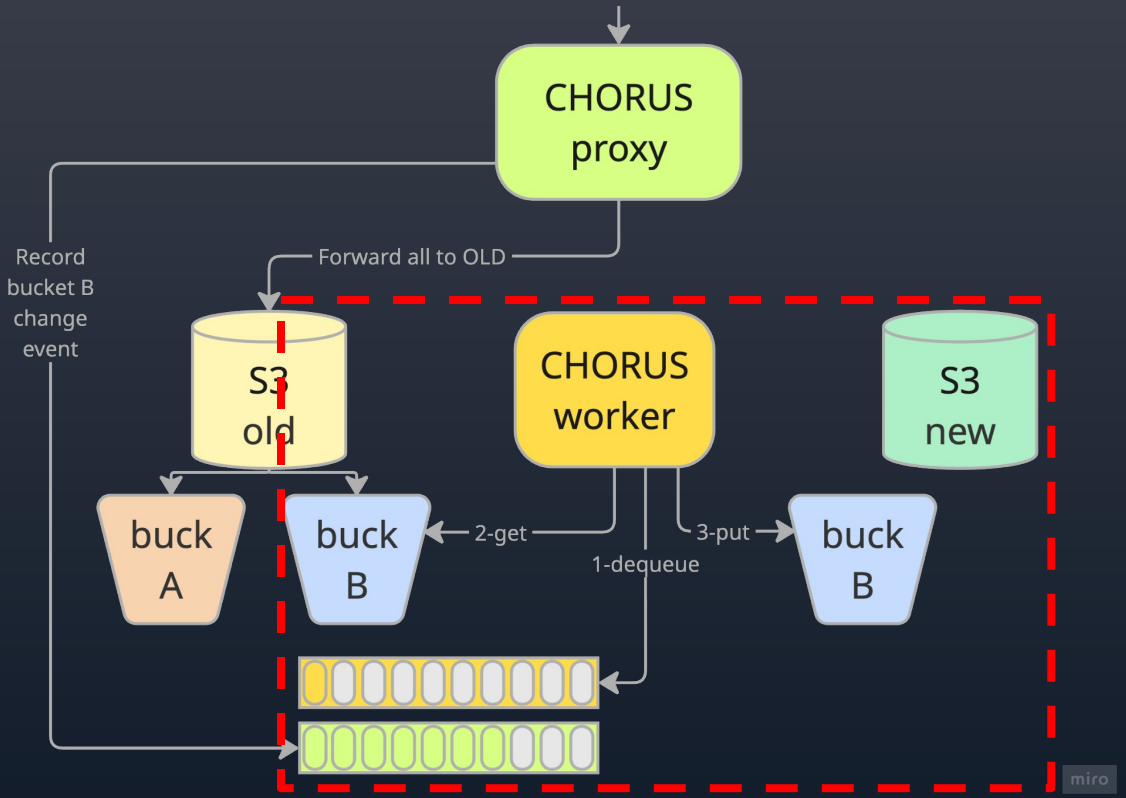
S3 migration with chorus



Track changed obj versions:

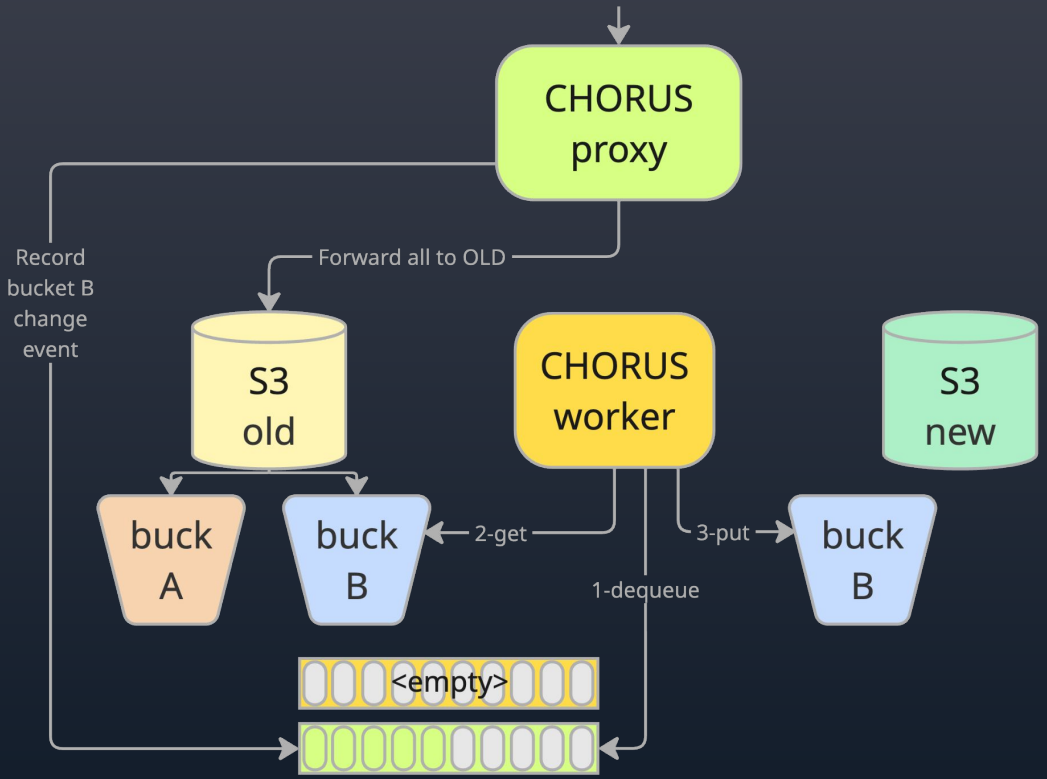
obj-name: <old:3, new:0>

S3 migration with chorus



- Existing, then changed
- N objects in parallel
- Worker rate-limit: RPM by storage

S3 backup with chorus



If Stop here => we have an async replication for bucket B

In this case proxy can be replaced with S3 bucket notifications

Plan Downtime Window

Per migration (Per bucket)

- START: Date (12.11.2025 4:00) | CRON ([0 4 * * *] - try every night at 4:00)
    - start only if change events < N
    - max downtime duration - rollback to old and retry next time

CL'SO

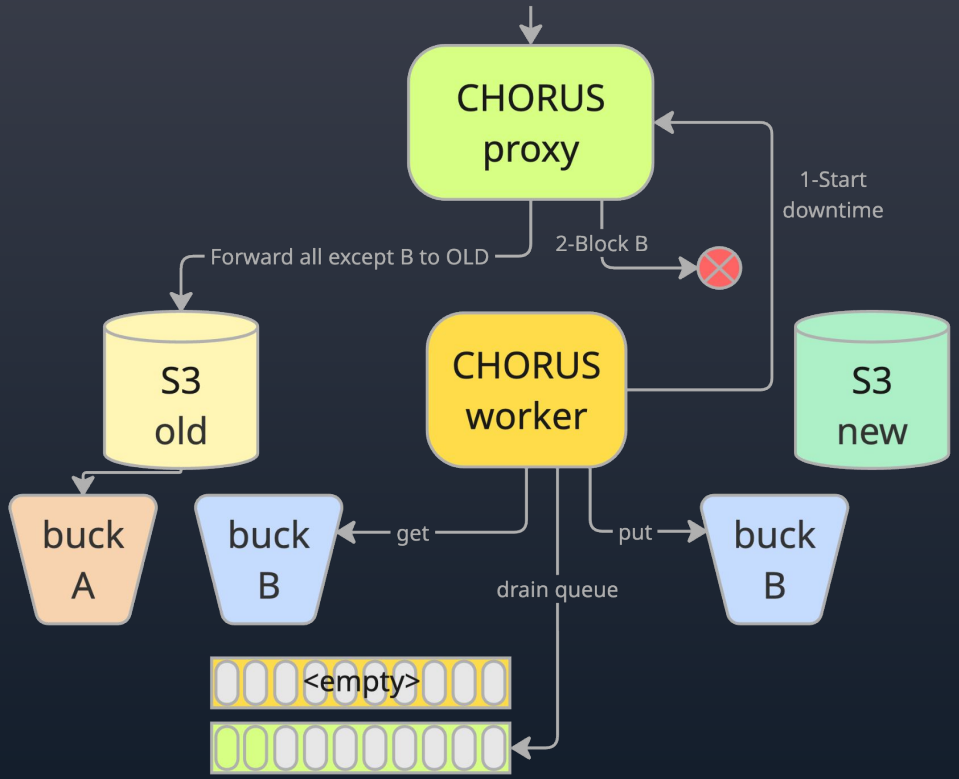
Plan Downtime Window

Per migration (Per bucket)

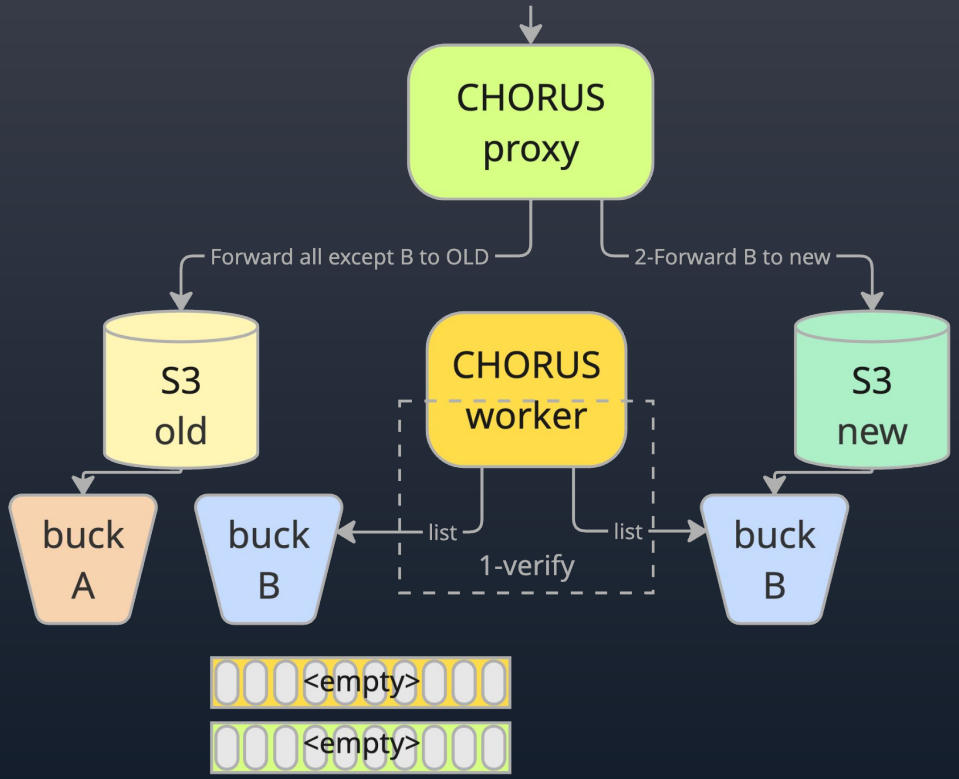
- START: Date (12.11.2025 4:00) | CRON ([0 4 * * *] - try every night at 4:00)
  - start only if change events < N
  - max downtime duration - rollback to old and retry next time

- OR Without Downtime

S3 migration with chorus

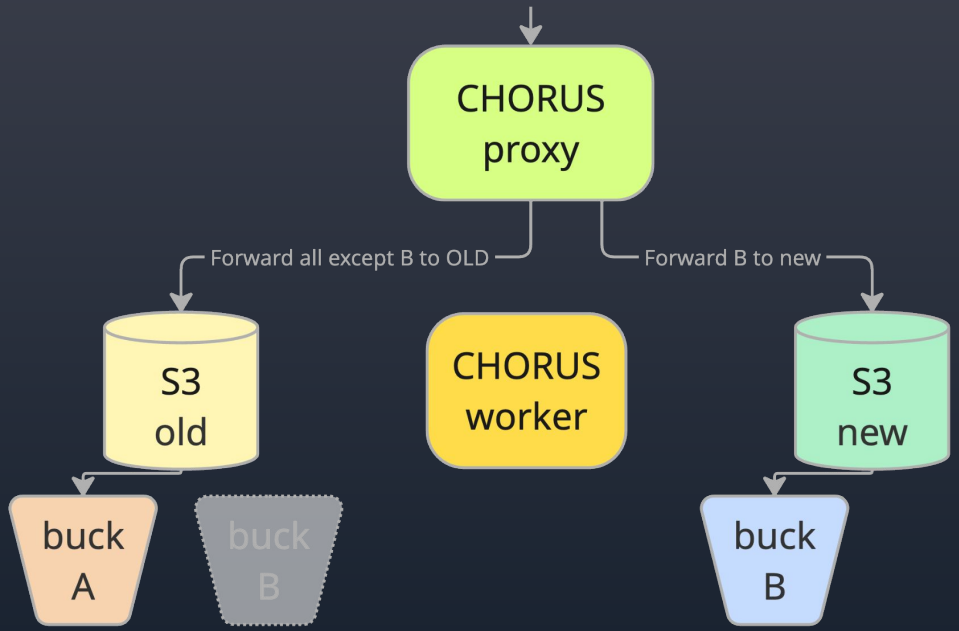


S3 migration with chorus



Optionally verify before unblock

S3 migration with chorus



Next:

- Start next bucket
- Sync bucket B back to old

Any to Any today

- S3 ⇔ S3: Ceph | Minio | AWS | Any S3-compatible storage ✅
  - Payload, metadata, tags, ACL ✅
  - Versioned buckets 🟡 - versionID not retained

CL'SO

Any to Any today

- S3 ⇔ S3: Ceph | Minio | AWS | Any S3-compatible storage ✅
  - Payload, metadata, tags, ACL ✅
  - Versioned buckets 🟡 - versionID not retained
  - Versioned buckets ✅ - versionID retained for Ceph & Minio - next release 0.6.1
- Swift ⇔ Swift: Openstack | RGW-swift ✅ - next release 0.6.1

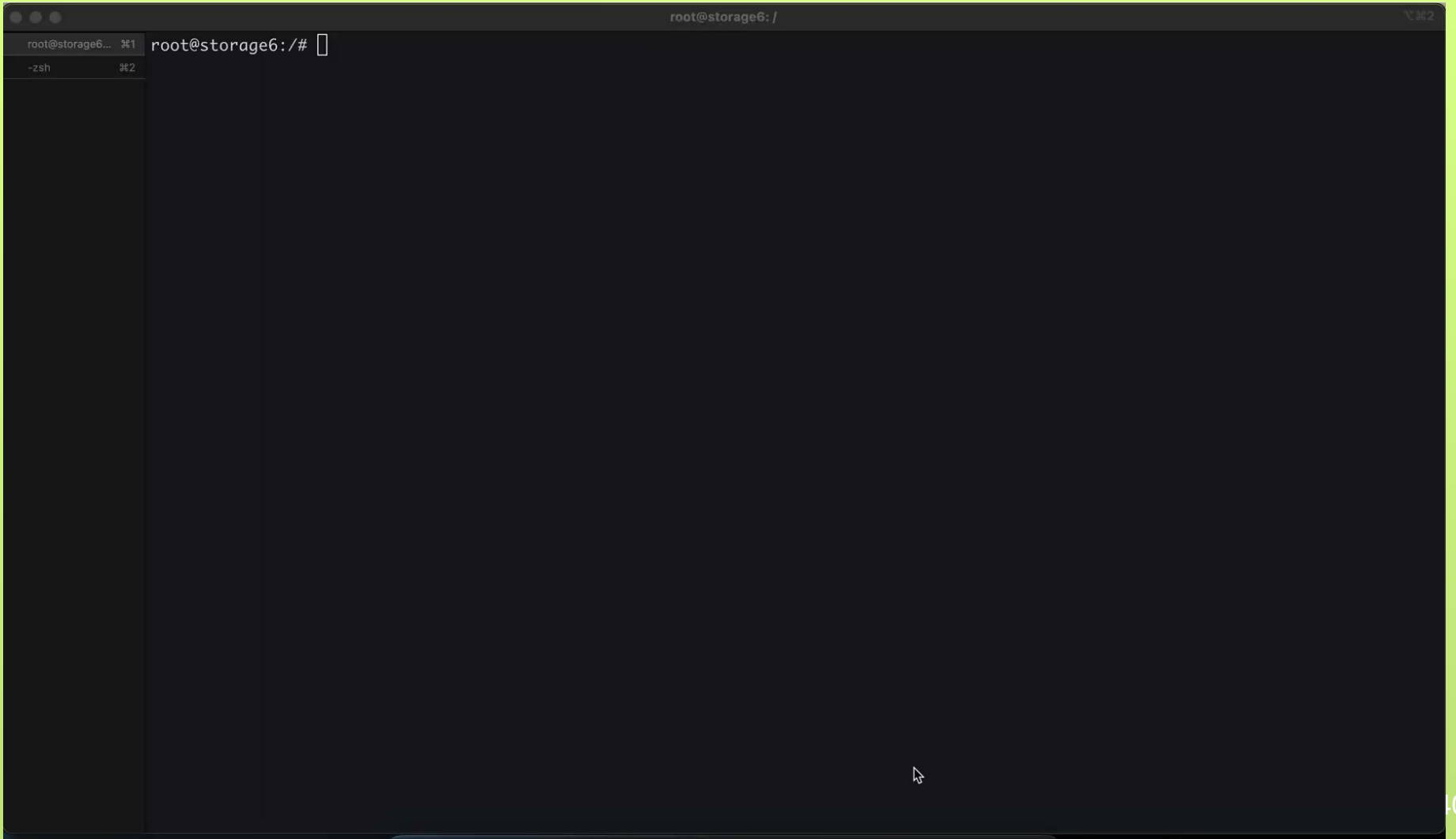
Any to Any today

- S3 ⇔ S3: Ceph | Minio | AWS | Any S3-compatible storage ✅
  - Payload, metadata, tags, ACL ✅
  - Versioned buckets 🟡 - versionID not retained
  - Versioned buckets ✅ - versionID retained for Ceph & Minio - next release 0.6.1
- Swift ⇔ Swift: Openstack | RGW-swift ✅ - next release 0.6.1

Goals:

- 100% S3/Swift API support 🪣
- Any [Azure, GCP, ...] => Ceph: ☁️=>🐙

DEMO


root@storage6: /
root@storage6... ⌘1
-zsh ⌘2
root@storage6:/#