Data Replication and Migration from Ceph RGW to Cloud

Soumya Koduri
Principal Software Engineer
Red Hat
Agenda

- Brief Introduction to Ceph
- Ceph Object Gateway (RGW)
- Hybrid Cloud & its Data movement
- Ceph RGW & Public Cloud integration
  - Cloud Sync (Data Replication)
  - Cloud Transition (Data Migration)
- Future work
Ceph
Ceph - A platform for petabyte-scale storage

Ceph is an open, massively scalable storage solution for modern workloads like cloud infrastructure, data analytics, media repositories, and backup and restore.
Ceph Architecture

- **LIBRADOS**: A library allowing apps to directly access Rados, with support for C, C++, Java, Python, Ruby, and PHP.
- **RADOSGW**: A bucket-based REST gateway, compatible with S3 and Swift.
- **RBD**: A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver.
- **CEPH FS**: A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE.

**RADOS**: A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes.
Ceph RGW
Ceph Object Gateway (RadosGW)

- Object storage interface
- Provides RESTful HTTP API to store objects and metadata
- Built on top of librados
- S3 and Swift compatible API
- Unified namespace
Hybrid Cloud

Data movement challenges
Hybrid Cloud

- unifies public cloud, private cloud and on-premises infrastructure
- creates single, unified and flexible distributed computing environment
- provide a way to leverage the advantages of the cloud while maintaining on-premises infrastructure for the most critical processes or most sensitive data.
- Efficient compared to public cloud or private cloud alone but biggest challenge is
  - the movement of data within the hybrid cloud.
Data movement challenges in hybrid cloud

Using DIY data transfer solution (eg., Rsync, Rclone, s3cmd, AWS CLI)

- lot of manual work
- lacks monitoring and error detection abilities
- doesn’t support incremental synchronization
- cost and time impacts
Data movement challenges in hybrid cloud

Using Multi-Cloud Gateway

- Clients have to connect to MCG (even On Premises)
  - Eg., Noobaa, Zenko
- Data Redundancy & Management Overhead
- Low I/O throughput
- Cannot leverage the full capabilities of On-Prem RGW server
- Not easy to detect and debug errors
Integrating Ceph RGW with Public Cloud

- Using Cloud Sync & Cloud Transition
  - For Data replication & Migration
- A full-fledged service
- Secure, reliable, efficient and fast data transfers
- Can transfer to multiple endpoints
- Incremental transfer
- Parallel processing
- Tools to monitor status and detect errors
Cloud Sync

- A SYNC MODULE
  - Built atop of multisite framework
  - allows for forwarding data and metadata to a different external tier
  - Asynchronous
- Syncs data to remote cloud service using REST APIs
- Unidirectional
- Compatible with AWS (S3)
- Zone-level & Bucket-level sync possible
- Sync Info Provider (WIP)
How to Configure

Similar to Multi-Site configuration but target zone tier type needs to be defined as cloud.

```bash
# radosgw-admin zone create --rgw-zonegroup={zone-group-name} --rgw-zone={zone-name} --endpoints={http://fqdn},{http://fqdn} --tier-type=cloud
```

The cloud tier configuration can be then done using the following command

```bash
# radosgw-admin zone modify --rgw-zonegroup={zone-group-name} --rgw-zone={zone-name} --tier-config={key}={val},{key}={val}
```

For more information, refer to

https://docs.ceph.com/en/latest/radosgw/cloud-sync-module/

For bucket-level granularity sync, refer to

Trivial Configuration

```json
{
    "connection": {
        "access_key": <access>,
        "secret": <secret>,
        "endpoint": <endpoint>,
        "host_style": <path | virtual>,
    },
    "acls": [ {
        "type": <id | email | uri>,
        "source_id": <source_id>,
        "dest_id": <dest_id> }
    ] ...
},
"target_path": <target_path>,

- target path configurable
- possible to map permissions of specific source users to specific destination users.
```
Cloud Transition
Cloud Transition

- Using S3 Lifecycle configuration and Storage-classes
  - set of rules that define actions that S3 server applies to a group of objects
    - Transition actions
    - Expiration actions
- Associates remote cloud endpoints to Lifecycle storage classes
- Enables data transition to cloud using lifecycle policies
- Unidirectional
  - Once transitioned, source object will be deleted.
  - Metadata can be retained.
- Compatible with AWS (S3)
- Available with Quincy Release (Ceph Upstream)
How to configure

Similar to adding regular storage class but with tier-type defined as cloud-s3

```
# radosgw-admin zonegroup placement add --rgw-zonegroup={zone-group-name} --placement-id={placement-id} --storage-class={storage-class-name} --tier-type=cloud-s3
```

The cloud tier configuration can be then done using the following command

```
# radosgw-admin zonegroup placement modify --rgw-zonegroup={zone-group-name} --placement-id={placement-id} --storage-class={storage-class-name} --tier-config={key}={val},{key}={val}
```

Note: Unlike regular storage classes, do not need a data pool.


### Configuring lifecycle policy:

The cloud storage class once configured can then be used like any other storage class in the bucket lifecycle rules. For example,

```
<Transition>
  <StorageClass>CLOUDTIER</StorageClass>
  ....
</Transition>
```

Note: cloud storage class should be specified last among all the storage classes the object transitions to
Trivial Configuration

```json
{
  "access_key": <access>,
  "secret": <secret>,
  "endpoint": <endpoint>,
  "host_style": <path | virtual>,
  "acls": [ 
    { "type": <id | email | uri>,
      "source_id": <source_id>,
      "dest_id": <dest_id> }, ...
  ],
  "target_path": <target_path>,
  "target_storage_class":<target-storage-class>" ,
  "retain_head_object":<true|false>
}
```

- target path & storage class configurable
- possible to map permissions of specific source users to specific destination users.
Current Capabilities & Limitations

- Uni-directional
- Only S3 compatible
- Once transitioned, the object cannot be read or transitioned back
- original object modification time and ETag get stored as metadata attributes on the destination objects
Future Work
Future work

- Support replication & transition to other cloud providers (like Azure).
- Sync Info Provider (Work In Progress)
  - [https://github.com/ceph/ceph/pull/38619](https://github.com/ceph/ceph/pull/38619)
- Federation between RGW and Cloud services.
  - Map Users, roles, bucket policies etc
- Send presigned redirect or read-through the objects transitioned to cloud
- Support s3:RestoreObject operation on cloud transitioned objects.
Resources

https://docs.ceph.com/en/latest/

https://docs.ceph.com/en/latest/radosgw/cloud-sync-module/

https://docs.ceph.com/en/latest/radosgw/cloud-transition/

https://github.com/ceph/ceph/pull/38619

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle-mgmt.html

https://www.ibm.com/cloud/learn/hybrid-cloud
Thank You

Email: skoduri@redhat.com
OFTC: #ceph, #ceph-devel