Moving data between buckets

At some point it will become necessary to move or backup your data to a different IBM® Cloud Object Storage region. One approach to moving or replicating data across object storage regions is to use a 'sync' or 'clone' tool, such as [the open-source rclone command-line utility](https://rclone.org/docs/).

This utility syncs a file tree between two locations, including cloud object storage. When rclone writes data to COS, it uses the COS/S3 API to segment large objects and uploads the parts in parallel according to sizes and thresholds set as configuration parameters.

This guide provides instructions for copying data from one IBM Cloud Object Storage bucket to another Object Storage bucket within the same region or to a second Object Storage bucket in a different Object Storage region. These steps need to be repeated for all the data that you want to copy from each bucket. After the data is migrated you can verify the integrity of the transfer by using rclone check, which will produce a list of any objects that don't match either file size or checksum. Additionally, you can keep buckets in sync by regularly running rclone sync from your available sources to your chosen destinations.

**Create a destination IBM Cloud Object Storage bucket**

You have the option of using your existing instance of IBM Cloud Object Storage or creating a new instance. If you want to reuse your existing instance, skip to step #2.

1. Create an instance of IBM Cloud Object Storage from the [catalog](https://cloud.ibm.com/catalog/services/cloud-object-storage" \t "_blank).
2. Create any buckets that you need to store your transferred data. Read through the [getting started guide](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-getting-started-cloud-object-storage) to familiarize yourself with key concepts such as [endpoints](https://cloud.ibm.com/docs/cloud-object-storage/basics?topic=cloud-object-storage-endpoints) and [storage classes](https://cloud.ibm.com/docs/cloud-object-storage/basics?topic=cloud-object-storage-classes).
3. **The rclone utility will not copy any bucket configurations or object metadata**. Therefore, if you are using any of the Object Storage features such as expiration, archive, key protect, etc. be sure to configure them appropriately before migrating your data. To view which features are supported at your COS destination, please refer to the [feature matrix](https://cloud.ibm.com/docs/services/cloud-object-storage?topic=cloud-object-storage-service-availability).

Feature configuration and access policies documentation can be viewed at the IBM Cloud portal pages listed below:

* + [IBM Cloud Identity and Access Management - IAM](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-iam)
  + [Activity Tracker](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-at)
  + [SysDig Monitoring](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-mm-cos-integration)
  + [Object Expiry](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-expiry)
  + [Cloud Object Storage Firewall](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-setting-a-firewall)
  + [Content Delivery Network - CDN](https://cloud.ibm.com/docs/cis?topic=cis-resolve-override-cos)
  + [Archive](https://cloud.ibm.com/docs/cloud-object-storage-infrastructure?topic=cloud-object-storage-infrastructure-archiving-data)
  + [KeyProtect](https://cloud.ibm.com/docs/key-protect?topic=key-protect-integrate-cos)
  + [IBM Cloud Functions](https://cloud.ibm.com/docs/openwhisk?topic=openwhisk-pkg_obstorage)

**Set up a compute resource to run the migration tool**

1. Choose a Linux™/macOS™/BSD™ machine or an IBM Cloud Infrastructure Bare Metal or Virtual Server with the best proximity to your data. Selecting a data center in the same region as the destination bucket is generally the best choice (e.g. if moving data from mel01 to au-syd, use a VM or Bare Metal in au-syd). The recommended Server configuration is: 32 GB RAM, 2-4 core processor, and private network speed of 1000 Mbps.
2. If you are running the migration on an IBM Cloud Infrastructure Bare Metal or Virtual Server use the **private** COS endpoints to avoid network egress charges.
3. Otherwise, use the **public** or **direct** COS endpoints.
4. Install rclone from [either a package manager or precompiled binary](https://rclone.org/install/).

curl https://rclone.org/install.sh | sudo bash

**Configure rclone for COS source data**

Create 'profiles' for your source and destination of the migration in rclone.

If needed, obtain COS credentials

1. Select your COS instance in the IBM Cloud console.
2. Click **Service Credentials** in the navigation pane.
3. Click **New credential** to generate credential information.
4. Select **Advanced** options.
5. Turn HMAC credentials to **On**.
6. Click **Add**.
7. View the credential that you created, and copy the JSON contents.

Get COS endpoint

1. Click **Buckets** in the navigation pane.
2. Click the migration destination bucket.
3. Click **Configuration** in the navigation pane.
4. Scroll down to the **Endpoints** section and choose the endpoint based on where you are running the migration tool.
5. Create the COS destination by copying the following and pasting into rclone.conf.

[COS\_SOURCE]

type = s3

provider = IBMCOS

env\_auth = false

access\_key\_id =

secret\_access\_key =

endpoint =

Use [COS\_DESTINATION] as the name of the profile you need to create to configure the destination. Repeat the steps above,

Using your credentials and desired endpoint, complete the following fields:

access\_key\_id = <access\_key\_id>

secret\_access\_key = <secret\_access\_key>

endpoint = <bucket endpoint>

**Configure rclone for COS destination data**

Repeat the previous steps for the destination buckets.

**Verify that the source and destination are properly configured**

* List the buckets associated with the source to verify rclone is properly configured.

rclone lsd COS\_SOURCE:

* List the buckets associated with the destination to verify rclone is properly configured.

rclone lsd COS\_DESTINATION:

Note: If you are using the same COS instance for the source and destination, the bucket listings will match.

**Run rclone**

1. Test your configuration with a dry run (where no data is copied) of rclone to test the copy of the objects in your source bucket (for example, source-test) to target bucket (for example, destination-test).
2. rclone --dry-run copy COS\_SOURCE:source-test COS\_DESTINATION:destination-test
3. Check that the files you want to migrate appear in the command output. If everything looks good, remove the --dry-run flag and, optionally add -v and/or -P flag to copy the data and track progress. Using the optional --checksum flag avoids updating any files that have the same MD5 hash and object size in both locations.
4. rclone -v -P copy --checksum COS\_SOURCE:source-test COS\_DESTINATION:destination-test

Try to max out the CPU, memory, and network on the machine running rclone to get the fastest transfer time.

There are other parameters to consider when tuning rclone. Different combinations of these values will impact CPU, memory, and transfer times for the objects in your bucket.

| **Flag** | **Type** | **Description** |
| --- | --- | --- |
| --checkers | int | Number of checkers to run in parallel (default 8). This is the number of checksums compare threads running. We recommend increasing this to 64 or more. |
| --transfers | int | This is the number of objects to transfer in parallel (default 4). We recommend increasing this to 64 or 128 or higher when transferring a large number of small files. |
| --multi-thread-streams | int | Download large files (> 250M) in multiple parts in parallel. This will improve the download time of large files (default 4). |
| --s3-upload-concurrency | int | The number of parts of large files (> 200M) to upload in parallel. This will improve the upload time of large files (default 4). |
| Table 1. Rclone options | | |

Migrating data using rclone copy only copies but does not delete the source data.

The copy process should be repeated for all other source buckets that require migration/copy/backup