Migrating from AWS

Migrating your data is a complex and daunting task, but don't let it stop you from making the right decision. Using the right tools will make moving from Amazon™ to IBM® Cloud Object Storage more secure, globally accessible, and completed with confidence.

**Before you begin**

Determine your goals and process for your migration before starting your migration. You may also consider training and partnerships to be beneficial. Your planning and assessment stage will consider many possibilities, including security and technical capabilities.

Documentation for any project will help keep you keep track of your resources as well as your goals. After assessing your existing projects, you may benefit by updating them to use IBM Cloud Object Storage libraries like those for ([Java](https://cloud.ibm.com/docs/cloud-object-storage/libraries?topic=cloud-object-storage-java), [Python](https://cloud.ibm.com/docs/cloud-object-storage/libraries?topic=cloud-object-storage-python), [Node.js](https://cloud.ibm.com/docs/cloud-object-storage/libraries?topic=cloud-object-storage-node)). If you're interested in programmer interfaces, the [REST API](https://cloud.ibm.com/docs/cloud-object-storage/api-reference?topic=cloud-object-storage-compatibility-api) will provide an in-depth look at operations and configurations.

**Provision and configure IBM Cloud Object Storage**

1. If you haven't already, create an instance of IBM Cloud Object Storage from the [Console](https://cloud.ibm.com/catalog/services/cloud-object-storage).
2. Create any buckets that you anticipate will be needed to store your transferred data. If you haven't already, 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. While Object Storage is compatible with the S3 API, it may be necessary to create new [Service credentials](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-service-credentials), or bring your own keys for your projects. In this guide, we will use [HMAC credentials](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-uhc-hmac-credentials-main) similar to the format of AWS credentials.
4. Managing [encryption](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-encryption) provides insights into security. Refer to product documentation on [IBM® Key Protect for IBM Cloud™](https://cloud.ibm.com/docs/key-protect?topic=key-protect-about) and [Hyper Protect Crypto Services](https://cloud.ibm.com/docs/hs-crypto?topic=hs-crypto-overview) for more information.

**Determine your solution**

It is true that a massively complex [migration](https://www.ibm.com/cloud/mass-data-migration) requires a complete service to plan and implement migrating your data to IBM Cloud Object Storage. But whatever the size of your data, your goals and timetable take precedence. Once you have provisioned and set your target, it is time to choose a process to achieve your goals on your time.

There are many ways to achieve the goal of migrating your AWS data. Integrated solutions provide comprehensive guides to migration, as shown in the [IBM Cloud Pak for Integration](https://www.ibm.com/cloud/cloud-pak-for-integration/high-speed-data-transfer). In addition to full-featured migration services, you may also want to investigate third party migration tools as part of your investigation. But don't forget that there are many CLI and GUI tools readily available for use as part of your migration.

* [COS CLI](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-cli-plugin-ic-cos-cli) can be used for many operations. For example, you may wish to use the CLI to configure your IBM Cloud Object Storage instances, and to create and configure buckets.
* [AWS CLI](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-aws-cli) can be used to list your current bucket's contents to prepare for migrating from AWS, among other operations:

aws s3 ls --recursive s3://<BUCKET\_NAME> --summarize > bucket-contents-source.txt

* [rclone](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-rclone) has many uses, and we'll look at it specifically, next.

Migrate your data

Based on the process and tools you've chosen, you will want to choose a strategy for migrating your data. We can take a look at a simplified process using the command line and the Go-based rclone executable as an example.

1. Install rclone from [either a package manager or precompiled binary](https://rclone.org/install/). There are more configuration options available with explanations at the IBM Cloud Object Storage [documentation](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-rclone).
2. curl https://rclone.org/install.sh | sudo bash

**Configure rclone with your AWS credentials**

You may start by creating 'profiles' for your source and destination of the migration in rclone. A profile contains the configuration and credentials needed for working with your date. You will be migrating from AWS, so you will need those credentials to continue. Also, you will create a profile for your destination credentials specifically for IBM Cloud Object Storage.

1. There are many options to configuring rclone and following the rclone config wizard is one way you can create profiles. You can create an rclone config file in ~/.rclone.conf by using the command as shown. Please use the root path of your home directory if the path shown isn't available.
2. touch ~/.config/rclone/rclone.conf
3. Create the AWS configuration settings by copying the following and pasting into rclone.conf using an appropriate editor.
4. [AWS]
5. type = s3
6. provider = AWS
7. env\_auth = false
8. access\_key\_id =
9. secret\_access\_key =
10. region =
11. Paste your AWS access\_key\_id and secret\_access\_key as obtained per instructions [here](https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html) into the appropriate fields of your configuration as shown.

**Configure rclone with your COS credentials**

To complement the credentials of the source, we look at configuring the destination profile next.

1. Create the COS configuration settings by copying the following and pasting into rclone.conf using an appropriate editor.
2. [COS]
3. type = s3
4. provider = IBMCOS
5. env\_auth = false
6. region =
7. access\_key\_id =
8. secret\_access\_key =
9. endpoint =
10. Paste your [HMAC](https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-uhc-hmac-credentials-main) access\_key\_id and secret\_access\_key into the appropriate fields of your configuration as shown in the first step. As noted in the beginning of the guide, you will want to enter the appropriate values for your instance regarding your [region and endpoint](https://cloud.ibm.com/docs/cloud-object-storage/basics?topic=cloud-object-storage-endpoints).

**Verify your configurations**

1. List the buckets from your source to verify rclone is properly configured for retrieval.
2. rclone lsd AWS:
3. List the COS bucket for your destination you created to verify rclone is properly configured for storage.
4. rclone lsd COS:

Use rclone to migrate from AWS

1. Do a dry run (no data copied) of rclone to sync the objects in your source bucket (for example, content-to-be-migrated) to the target COS bucket (for example, new-bucket).
2. rclone --dry-run copy AWS:content-to-be-migrated COS:new-bucket
3. Check that the files you want to migrate appear after running the command. If everything looks as you expect, remove the --dry-run flag and add a -v flag to show a verbose output while the data is being copied. Using the optional --checksum flag avoids updating any files that have the same MD5 hash and object size in both locations.
4. rclone -v copy --checksum AWS:content-to-be-migrated COS:new-bucket

As you perform the migration of your data using the process you've outlined, you will want to validate and verify the results.