Kunjal Desai

Kunjal.x.desai@kp.org

Abstract

The document provides introduction to rclone tool and detailed instructions on how to configure remotes and execute commands to migrate IBM COS to Azure Blob storage. The document is intended for technical audiences involved with Bluemix To Azure Migration initiative.

rclone – how to

Using rclone tool to migrate IBM Cloud Object Storage to Azure Blob Storage

Contents

[Background 2](#_Toc77890258)

[Download the tool 2](#_Toc77890259)

[Configuring the tool 2](#_Toc77890260)

[Configuring connection to IBM Cloud Object storage 2](#_Toc77890261)

[Configuring connection to Azure Blob storage 4](#_Toc77890262)

[Migrating IBM COS files to Azure Blob 5](#_Toc77890263)

[Config File 7](#_Toc77890264)

[Verification of Migration 8](#_Toc77890265)

[Check command 8](#_Toc77890266)

[Flags 8](#_Toc77890267)

[Other flags 8](#_Toc77890268)

[Powershell Script 9](#_Toc77890269)

[Verification Report Format 9](#_Toc77890270)

# Background

Rclone is a command line program to manage files on cloud storage. It supports more than 40 cloud storage products including IBM Cloud Object Storage and Azure Blob Storage. The focus of this document will be on configuration, connection and execution of commands that will copy buckets and files from IBM COS to Azure blob storage.

# Download the tool

The tool is available to download from <https://rclone.org/download/>

Since this tool’s website is blocked from KP, a zipped copy of the tool is attached below



# Configuring the tool

The tool needs to be configured prior to using it. Configuration of the tool involves setting up one or more **remote**. remote is a named connection to a specific cloud storage product/provider. You can think of them as named connections which have information required to establish connection. Typically two remotes (or two named connections) will be required, 1 each for

1. source IBM COS
2. destination Azure blob storage

During setting up a new remote, rclone opens an interactive session with the user where it will ask the user for input or chose from a list of values. The interactive session is conducted purely from the command prompt.

## Configuring connection to IBM Cloud Object storage

Prior to commencing configuration for IBM cloud object storage, keep the credentials for storage service handy. It will be obtained from Bluemix console and typically will look similar to sample given below

|  |
| --- |
| {  "accessKeyID": "<accessKeyId>",  "secretAccessKey": "<secretAccessKey>",  "region": "<region>",  "endpoint-url": "<endpointUrl>" } |

For the purpose of examples in this document, it is assumed that the rclone utility is stored at

C:\cos2blob\. Navigate to c:\cos2blob via command prompt and execute command **rclone config**

Text

Description automatically generated

The table below indicates the values to be provided for configuring the remote

|  |  |  |
| --- | --- | --- |
| **Input** | **Description** | **Value** |
| name> | Name of remote. For example bmx\_training | bmx\_training |
| storage> | Chose from a list of options provided. | 4 or s3 |
| provider> | Chose from a list of S3 compatible providers | 6 or IBMCOS |
| env\_auth> | True or false | Hit Enter |
| access\_key\_id> | accessKeyId from IBM COS Credentials | <accessKeyId> |
| secret\_access\_key> | secretAccessKey from IBM COS credentials | <secretAccessKey> |
| region> | Region from IBM COS credentials | <Region> |
| endpoint> | Endpoint from IBM COS credentials | <endpointUrl> |
| location\_constraint> | Since KP Bluemix is on private cloud, make no selection | Hit enter. |
| acl> | Default | Hit enter |
| Edit Advanced Config> | To set more granular options. Continue with defaults (n) | N |
| Confirmation prompt to save the configured remote | You can edit the values of the remote, if required. | Y |

## Configuring connection to Azure Blob storage

Prior to commencing configuration for Azure Blob storage, keep the Account name and Public Access Key for Azure blob storage handy. It can be obtained from Azure portal

For the purpose of examples in this document, it is assumed that the rclone utility is stored at

C:\cos2blob\. Navigate to c:\cos2blob via command prompt and execute command **rclone config**

Text

Description automatically generated

The table below indicates the values to be provided for configuring the remote for Azure Blob Storage

|  |  |  |
| --- | --- | --- |
| **Input** | **Description** | **Value** |
| name> | Name of remote. For example Az\_trial | az\_trial |
| storage> | Chose from a list of options provided. | 25 or azureblob |
| account> | Provide the account name of Azure Blob storage | <accountName> |
| service\_principal\_file> | NA | Hit enter |
| key> | Public primary key | <primaryKey> |
| Sas\_url> | NA | Hit enter |
| use\_msi> | NA | Hit enter |
| use\_emulator> | NA | Hit Enter |
| Edit\_Advanced\_config | To set more granular options. Continue with defaults (n) | N |
| y/e/d | Confirm the values provided. | Y |

# Migrating IBM COS files to Azure Blob

To migrate IMB COS buckets and their files to Azure Blob, the COPY command of rclone is utilized.

The copy operation occurs per bucket. So if IBM COS has 4 buckets then the copy command should be executed 4 times.

Lets assume the following bucket structure in IBM COS

|  |  |
| --- | --- |
| Bucket Name | File Name |
| Bucket01 | File\_b1\_f1.txt |
| File\_b1\_f2.txt |
| File\_b1\_f3.txt |
| Bucket02 | File\_b2\_f1.txt |
| File\_b2\_f2.txt |
| File\_b2\_f3.txt |
| Bucket03 | File\_b3\_f1.txt |

If the above IBM COS is configured via rclone with name BMX, to be migrated to Azure Blob Storage config name of Az1 then to migrate all IBM COS buckets to Az1 the commands would be

|  |
| --- |
| rclone copy BMX:Bucket01 Az1:Bucket01  rclone copy BMX:Bucket02 Az1:Bucket02  rclone copy BMX:Bucket03 Az1:Bucket03 |

If Az1 does not have any of Bucket01, Bucket02 or Bucket03 then rclone will create it.

# Config File

All the definition of remotes which are created through the interactive sessions (steps above) are stored in plain text in a config file.

The path of the config file can be found by executing the following on command line

|  |
| --- |
| rclone config file |

By default it will be at C:\Users\<NUID>\.config\rclone\rclone.conf

To add a configuration quickly, without having to go through the interactive process, simply add the defined section below to the config file.

Note – None of the values needs to be put in single nor double quotes.

|  |  |
| --- | --- |
| IBM COS template | Description |
| [<remote Name>] | Name of the config/remote within square brackets |
| type = s3 | As is |
| provider = IBMCOS | As is |
| access\_key\_id = <accessKey> | <accessKey> from credentials |
| secret\_access\_key = <secretAccessKey> | <secretAccessKey> from credentials |
| region = <Region> | <Region> from credentials |
| endpoint = <endpoint> | <endpoint> from credentials |
|  | Blank line |

|  |  |
| --- | --- |
| Azure Blob template | Description |
| [<remote name>] | Name of the config/remote within square brackets |
| type = azureblob | As is |
| account = <accountName> | Account Name |
| key = <PrimaryKey> | Primary Key |
|  | Blank Line |

# Verification of Migration

If the rclone copy command will executed with –checksum flag then while copying rclone will compare checksum values and size of the file to determine equality. Equal files are skipped during copy process.

All the examples given assume that we have two remotes called

1. bmx\_atm – remote that connects object service of ATM
2. az\_atm – remote that connects to Azure Blob Storage of ATM

Example

|  |
| --- |
| rclone copy bmx\_atm:devDocUploads az\_atm:devDocUploads –checksum |

Each object storage provider can use a different hash for checksum. Luckily S3 and Azure both use MD5 hence checksum validation is possible.

# Check command

The check command will check the files in source and destination and show a report whose format is controlled by flags.

Example – comparison based on checksum and sizes;

|  |
| --- |
| rclone check bmx\_atm:devDocUploads az\_atm:devDocUploads |

# Flags

Optionally flags can be appended to a basic check command to control how check behaves.

Example – comparison based on sizes only; ignoring checksum

|  |
| --- |
| rclone check bmx\_atm:devDocUploads az\_atm:devDocUploads –size-only |

Example – checks for existence of source files in destination; not the other way around.

|  |
| --- |
| rclone check bmx\_atm:devDocUploads az\_atm:devDocUploads –one-way |

## Other flags

If you want the report on the screen then give a – (dash) instead of filename.

|  |  |
| --- | --- |
| --combined <filename> | Make a combined report of changes |
| --differ <filename> | Reports all non-matching files |
| --error <filename> | Report all files with errors (hashing difference) |
| --match <filename> | Report all matching files |
| --missing-on-dst <filename> | Report all files missing from the destination |
| --missing-on-src <filename> | Report all files missing from the source |

# Powershell Script

Rclone does not support root level copy. In other words the copy command always expects a bucket name as a source and/or destination i.e it copies buckets and files

If the source object storage service has X buckets then the copy command must be executed X number of times. This can be inefficient and time consuming. To overcome this limitation a powershell script is attached below.

The ideal sequence of events while using the powershell script is as follows

1. Setup the remotes from command line or editing the config file.
2. Edit the powershell script. Provide values for the first 4 variables

|  |  |
| --- | --- |
| Variable | Description |
| $rmtSource | Name of source remote |
| $rmtDest | Name of destination remote |
| $rcPath | Full path to rclone.exe but without the extension |
| $rcReportPath | Full path to a folder under which migration reports will be saved. The script will create the path if it doesn’t exists. |

## Verification Report Format

Reports will be generated per bucket. Each line in the report represents the status for each file. The status is represented by a symbol followed by the filename.

The symbols have the following meaning.

|  |  |
| --- | --- |
| Symbol | Meaning |
| = | File is present in source and destination and is identical |
| + | Present in source but absent in destination |
| - | Absent in source but present in destination |
| \* | Present in source and destination but their checksums or size are different. |
| ! | There was an error reading or hashing the file in source or destination |

Changes to names of Container in Azure

IBM COS allows it’s buckets names to have capital letter, number and underscores. However Azure Containers allows only lower case letters, numbers and hyphen (No capitals or underscores). To circumvent these limitations of Azure containers names, the powershell script will change the name of IBM COS bucket’s name when creating Azure Container by using the following rules

1. Convert to lowercase
2. Replace underscore with hyphen

The powershell script is given below

