Object Storage Migration

March 2022

SAMSUNG SDS

Contact

Geunhyoung Lim
Technology Innovation Team, Samsung SDS
geungol.lim@samsung.com

He is an SCP storage architect and is particularly interested in backup/HPC.

Revision History

Version	Revision Date	Revised Details
1.0	March, 2022	First version published

Contents

1. Overview

This document is intended to provide technical guide when users plan to migrate Object Storage Data to Samsung Cloud Platform(hereafter SCP) Object Storage. First, we introduce the Data Migration Tools, and then guide you through the scenario-based migration examples to commands in detail.

2. Migration Tools

Rclone can be considered as a Cloud Object Storage Migration Tool.
Rclone is an open source migration tool supporting more that 40 cloud storages including AWS, Azure, GCP, and it is a tool that supports various functions such as Migration as well as Mount. Both Linux and Windows are provided with the same command line, and detailed information can be found on the Rclone web page (rclone.org).

Data Type	Protocol	Tool	notes
File	S3 Azure GCP	Rclone	- OpenSource Migration Tool - Supports File to Object, Object to Object migration - Provides various functions other than migration

3. Object Migration

3.1 Rclone Installation

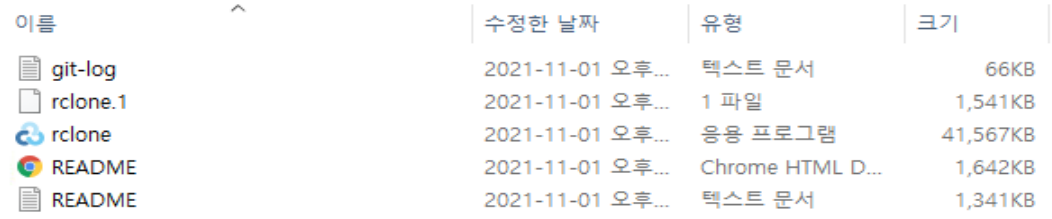
In a Linux environment

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

```
[root@linux-test2 ~]# rclone version
rclone v1.57.0
- os/version: centos 7.6.1810 (64 bit)
- os/kernel: 3.10.0-957.el7.x86_64 (x86_64)
- os/type: linux
- os/arch: amd64
- go/version: go1.17.2
- go/linking: static
- go/tags: none
```

In a Windows environment

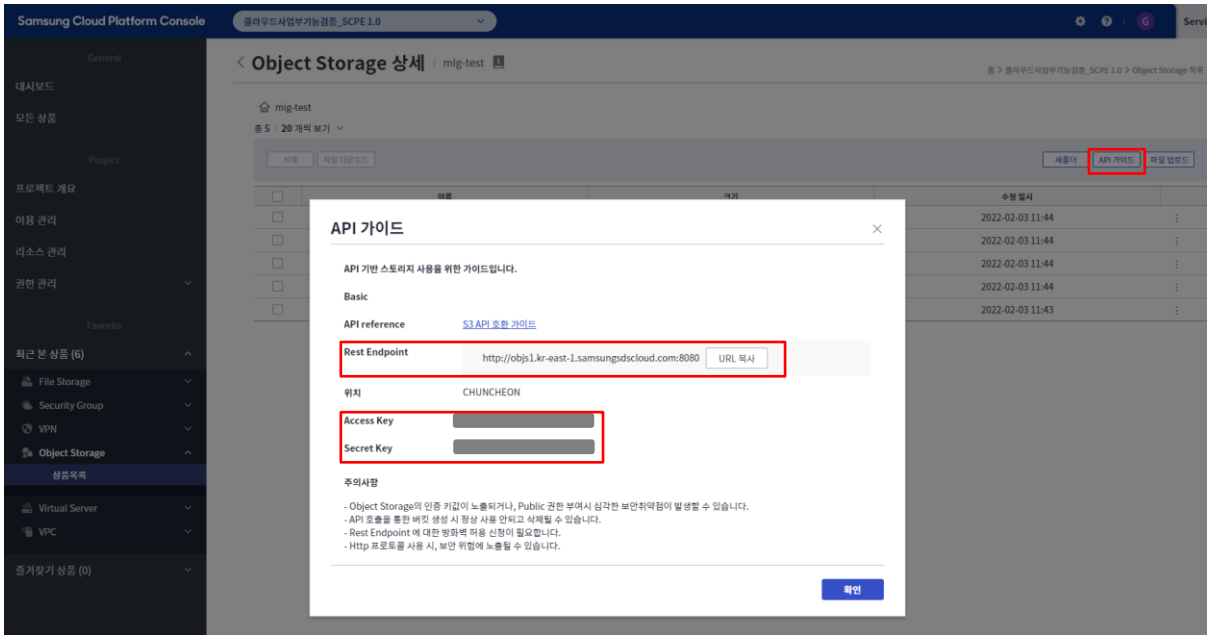
Download the Windows installation file from the Rclone webpage (rclone.org/download/) and unzip it. After decompression, go to the installation path in the CMD window and execute the rclone command



```
c:₩rclone>rclone version
rclone v1.57.0
- os/version: Microsoft Windows Server 2016 Standard 1607 (64 bit)
- os/kernel: 10.0.14393.4825 (x86_64)
- os/type: windows
- os/arch: amd64
- go/version: go1.17.2
- go/linking: dynamic
- go/tags: cmount
```

3.2 Configuring Object Storage

Below example shows how to configure SCP Object Storage.
Bucket's Access Key, Secret Key, and Rest Endpoint information can be checked in SCP Console -> Object Storage -> Bucket -> API Guide



1) Select “rclone config” and then select n) New remote

```
e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q>
```

2) Type a relevent name

```
name> SCP_OBJ
```

3) Select “Amazon S3 Compliant Storage”

```
 1 / 1Fichier
   \ "fichier"
 2 / Alias for an existing remote
   \ "alias"
 3 / Amazon Drive
   \ "amazon cloud drive"
 4 / Amazon S3 Compliant Storage Providers including AWS, Alibaba, Ceph, Digital Ocean, Dreamhost, IBM COS, Minio, SeaweedFS, and Tencent COS
   \ "s3"
```

4) Select “14 / Any other S3 compatible provider”

```
11 / StackPath Object Storage
   \ "StackPath"
12 / Tencent Cloud Object Storage (COS)
   \ "TencentCOS"
13 / Wasabi Object Storage
   \ "Wasabi"
14 / Any other S3 compatible provider
   \ "Other"
provider>
```

5) Press Enter to choose “false” as a default for credentials.

```
Option env_auth.
Get AWS credentials from runtime (environment variables or EC2/ECS meta data if no env vars).
Only applies if access_key_id and secret_access_key is blank.
Enter a boolean value (true or false). Press Enter for the default ("false").
Choose a number from below, or type in your own value.
 1 / Enter AWS credentials in the next step.
   \ "false"
 2 / Get AWS credentials from the environment (env vars or IAM).
   \ "true"
env_auth>
```

6) Type Access Key and Secret Key

```
Option access_key_id.
AWS Access Key ID.
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
access_key_id>
Option secret_access_key.
AWS Secret Access Key (password).
Leave blank for anonymous access or runtime credentials.
Enter a string value. Press Enter for the default ("").
secret_access_key>
```

7) Since there is no region information, enter it blank and press Enter.

```
Choose a number from below, or type in your own value.
   / Use this if unsure.
 1 | Will use v4 signatures and an empty region.
   \ ""
   / Use this only if v4 signatures don't work.
 2 | E.g. pre Jewel/v10 CEPH.
   \ "other-v2-signature"
region>
```

8) Type EndPoint information

```
Option endpoint.
Endpoint for S3 API.
Required when using an S3 clone.
Enter a string value. Press Enter for the default ("").
endpoint> http://objs1.kr-east-1.samsungsdscloud.com:8080
```

9) Leave the Location constraint information blank and Enter

```
Option location_constraint.
Location constraint - must be set to match the Region.
Leave blank if not sure. Used when creating buckets only.
Enter a string value. Press Enter for the default ("").
location_constraint>
```

10) Select either private or public-read for ACL

```
   / Owner gets FULL_CONTROL.
 1 | No one else has access rights (default).
   \ "private"
   / Owner gets FULL_CONTROL.
 2 | The AllUsers group gets READ access.
   \ "public-read"
```

11) Select No for advanced config

```
Edit advanced config?
y) Yes
n) No (default)
y/n>
```

12) After checking the information you have set, if there is no problem, select y to complete the registration

```
--------------------
[SCP_OBJ]
type = s3
provider = Other
access_key_id = [redacted]
secret_access_key = [redacted]
endpoint = http://objs1.kr-east-1.samsungsdscloud.com:8080
acl = private
--------------------
y) Yes this is OK (default)
e) Edit this remote
d) Delete this remote
y/e/d>
```

13) After registration is complete, you can check the registered information through the "rclone config show" command. Normal access can be checked with "rclone ls RemoteName:BucketName" command

```
[root@linux-test2 source]# rclone ls SCP_OBJ:mig-test
1073741824 testfile1
1073741824 testfile2
524288000 testfile3
524288000 testfile4
```

3.3 Migration process

1) Configure source and target storage information.
   : run "rclone config" to set information for each item

```
#rclone config
```

2) Check config registration
   : If the object of the storage bucket registered with "rclone ls" is searched, it is registered normally.

```
#rclone ls [config name]:[bucket name]
```

3) Perform Dry run (pre-migration verification)
   : "--dry-run" option is also available as pre-validation

```
#rclone sync source:path dest:path --dry-run --progress
```

4) Perform Migration

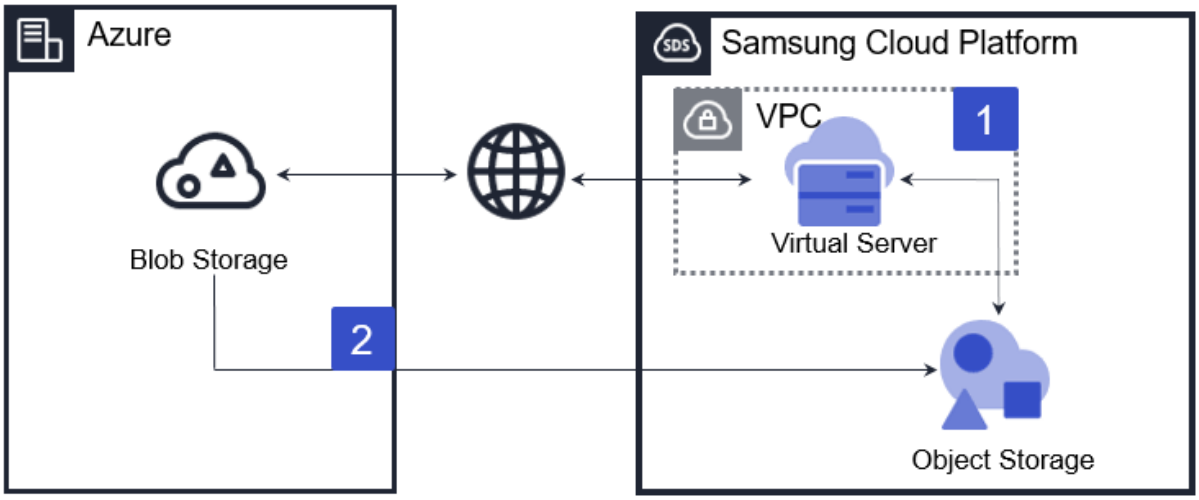
```
#rclone sync source:path dest:path --progress [flags]
```

5) Verify after completion of Migration

```
#rclone check source:path dest:path [flags]
```

Examples

Ex1. Azure Blob Storage to SCP Object Storage Migration



1) From SCP VM, configure Azure Blob Storage and SCP Object Storage using rclone config.

```
[root@linux-test2 ~]# rclone config show
[azure]
type = azureblob
account = limst01
key = [redacted]

[SCP_OBJ]
type = s3
provider = Other
access_key_id = [redacted]
secret_access_key = [redacted]
endpoint = http://objs1.kr-east-1.samsungsdscloud.com:8080
acl = private
```

Check config registration

```
[root@linux-test2 ~]# rclone ls azure:azure-test
1073741824 testfile1
1073741824 testfile2
```

2) Perform Dry run

```
[root@linux-test2 ~]# rclone sync azure:azure-test SCP_OBJ:mig-test --dry-run --progress
2022-01-21 16:02:20 NOTICE: testfile2: Skipped copy as --dry-run is set (size 1Gi)
2022-01-21 16:02:20 NOTICE: testfile1: Skipped copy as --dry-run is set (size 1Gi)
Transferred:            2 GiB / 2 GiB, 100%, 0 B/s, ETA -
Transferred:            2 / 2, 100%
Elapsed time:         0.3s
2022/01/21 16:02:20 NOTICE:
Transferred:            2 GiB / 2 GiB, 100%, 0 B/s, ETA -
Transferred:            2 / 2, 100%
Elapsed time:         0.3s
```

3) Perform migration

```
[root@linux-test2 ~]# rclone sync azure:azure-test SCP_OBJ:mig-test --progress
Transferred:      677.992 MiB / 2 GiB, 33%, 57.638 MiB/s, ETA 23s
Transferred:            0 / 2, 0%
Elapsed time:        12.0s
Transferring:
 *                                  testfile1: 40% /1Gi, 35.545Mi/s, 17s
 *                                  testfile2: 25% /1Gi, 23.091Mi/s, 32s
```

4) Verify after completion of Migration

```
[root@linux-test2 ~]# rclone check azure:azure-test SCP_OBJ:mig-test --progress
2022-01-21 16:20:34 NOTICE: S3 bucket mig-test: 0 differences found
2022-01-21 16:20:34 NOTICE: S3 bucket mig-test: 2 matching files
Transferred:              0 B / 0 B, -, 0 B/s, ETA -
Checks:                 2 / 2, 100%
Elapsed time:         0.4s
[root@linux-test2 ~]#
```

Ex2. File to SCP Object Storage Migration

1) Perform Dry run

```
[root@linux-test2 ~]# rclone sync /source SCP_OBJ:mig-test --dry-run --progress
2022-02-03 10:36:54 NOTICE: testfile1: Skipped copy as --dry-run is set (size 1Gi)
2022-02-03 10:36:54 NOTICE: testfile4: Skipped copy as --dry-run is set (size 100Mi)
2022-02-03 10:36:54 NOTICE: testfile3: Skipped copy as --dry-run is set (size 100Mi)
2022-02-03 10:36:54 NOTICE: testfile2: Skipped copy as --dry-run is set (size 1Gi)
Transferred:       2.195 GiB / 2.195 GiB, 100%, 0 B/s, ETA -
Transferred:           4 / 4, 100%
Elapsed time:        0.0s
2022/02/03 10:36:54 NOTICE:
Transferred:       2.195 GiB / 2.195 GiB, 100%, 0 B/s, ETA -
Transferred:           4 / 4, 100%
Elapsed time:        0.0s
```

2) Perform migration

```
[root@linux-test2 ~]# rclone sync /source SCP_OBJ:mig-test --progress
Transferred:      187.336 MiB / 2.195 GiB, 8%, 48.059 MiB/s, ETA 42s
Transferred:            0 / 4, 0%
Elapsed time:         4.0s
Transferring:
 *                                  testfile1:  4% /1Gi, 8.364Mi/s, 1m57s
 *                                  testfile2:  3% /1Gi, 9.999Mi/s, 1m38s
 *                                  testfile3: 70% /100Mi, 20.852Mi/s, 1s
 *                                  testfile4: 32% /100Mi, 8.777Mi/s, 7s
```

3) Verify after completion of Migration

```
[root@linux-test2 ~]# rclone check /source SCP_OBJ:mig-test --progress
2022-02-03 10:43:49 NOTICE: S3 bucket mig-test: 0 differences found
2022-02-03 10:43:49 NOTICE: S3 bucket mig-test: 4 matching files
Transferred:              0 B / 0 B, -, 0 B/s, ETA -
Checks:                 4 / 4, 100%
Elapsed time:         2.8s
```