

MIGRATION TOOL

for Rubrik + Cloudian buckets

v1.3.5

Disclaimer / Warning

Use this tool with precautions (review the config file created manually for a double-check) for your environment : it is NOT an official tool supported by Cloudian. Cloudian can NOT be involved for any bugs or misconfiguration due to this tool. So you are using it at your own risks and be aware of the restrictions.

Table of Contents

1 Flowchart of the migration.....3

2 Installation.....6

3 Deploy OVA.....7

4 Configuration.....9

5 Look at the tool.....11

6 Full example.....12

7 Logs.....16

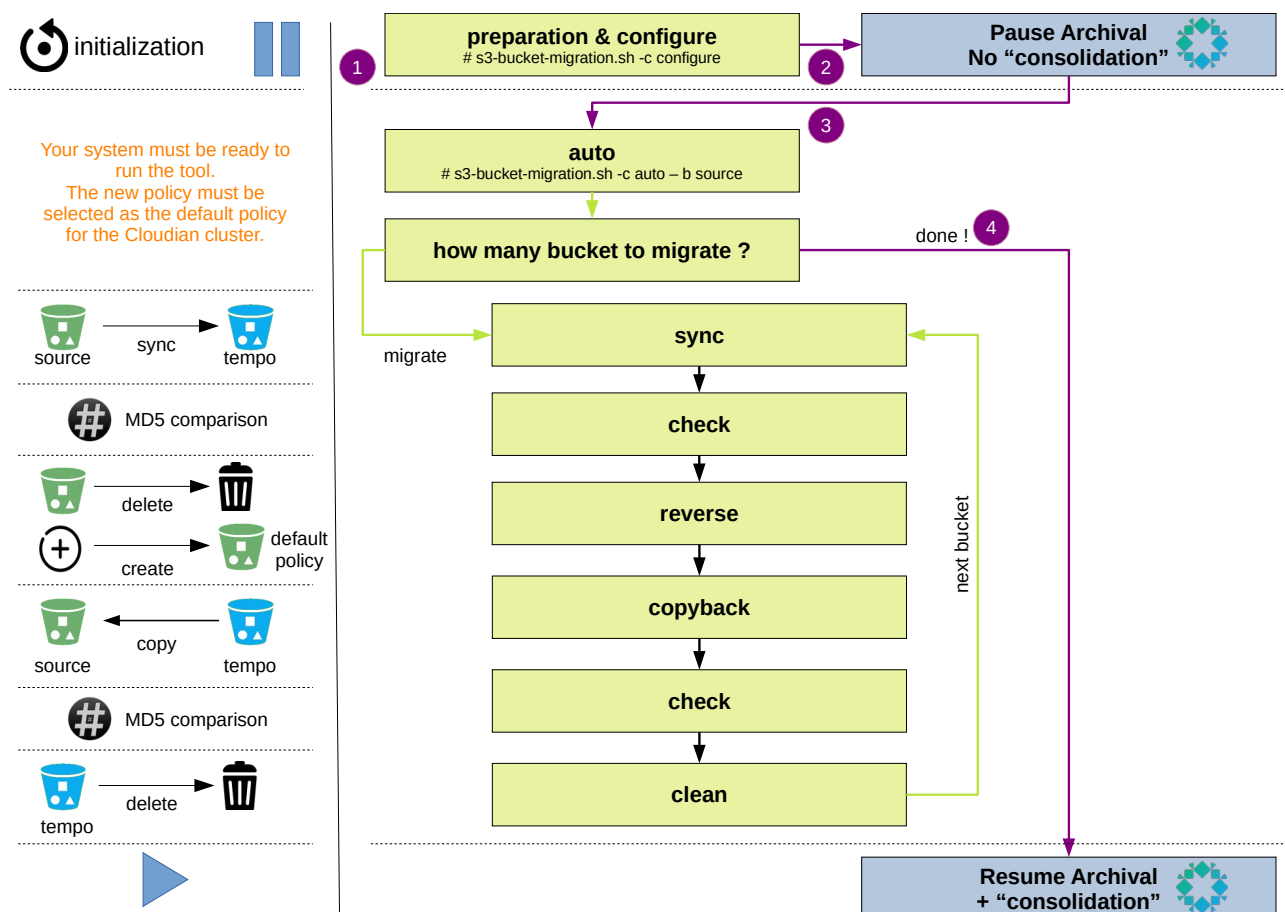
1 Flowchart of the migration

Due to the s3 compatibility, we can't rename a name of a bucket easily. This tool is provided as it is and is able to migrate a list of buckets with the same Archival Location.

Recommendations

- I strongly recommend to put the Archival Location in “pause” mode.
- Based on my experience and the mechanisms of Rubrik, I strongly recommend to disable any archive consolidation if it is enabled on the buckets which have to be migrated (please contact your Rubrik support representative for any question about consolidation if needed). In fact, the archiving consolidation continues to operate even if the Archive Location is “paused” (Rubrik : works as designed) → you might have some changes during the migration time-frame.
- look at the Rubrik job listing and be sure that there is no job in progress on the targeted Archive Location → “Paused” doesn't mean that all the current jobs are finished or canceled.

The script “s3-bucket-migration.sh” follows the flowchart below :



In this flowchart, the source bucket is always migrated to another bucket with a suffix “-tempo” (meaning temporary bucket). For example, the migration of the bucket “vm-bronze-rubrik-0” will use a second bucket “vm-bronze-rubrik-0-tempo” for the migration.

Based on the statement above, you must check and double-check the free capacity of your Cloudian cluster and the capacity you will need to migrate a bucket or a couple of buckets.

Notice at a moment, we have to delete the source bucket (Example : “vm-bronze-rubrik-0”) in order to re-create it with the full advantages of the new storage policy.

As a migration is a sensitive operations, there are several tests and checks implemented before removing the source bucket (see : MD5 hash).

The tool will continue automatically if there is no error detected during the migration else it’s a full stop for the rest of the migration. In this case, you have to review the logs, to resolve the error and to use the tool with the step-by-step process (or to re-run a full migration if possible).

First, don’t panic and don’t cry :-) logically you should still have two buckets :

- the source : “vm-bronze-rubrik-0”
- the tempo destination : “vm-bronze-rubrik-0-tempo”

The logs should tell you the error like a MD5 mismatch which means an error during the copy. In the “running” folder, you should have more log files with all the details. For your information, those log files are deleted automatically after each successful migration.

Good to know

There are a lot of parameters which might impact the migration and those are tuned manually or automatically by a software release (up to CDM 5.0).

The most important parameter is : **partSizeForCloudUploadInMB**

Default minimum upload part size when performing native multipart upload. We start with this value and if it would result in greater than `maxNumPartsForCloudUpload` parts, double the part size. Apply this logic to ensure the number of parts with the chosen part size are under.

Default: 16 MB → the tool handles this MPU size and a default “**maxNumPartsForS3Upload**” of 10000

maxNumPartsForS3Upload

Max number of parts for a single multipart upload to S3 (previously named `maxNumPartsForCloudUpload` prior to 4.3)

Default: 10000

maxPartSizeForS3UploadInMB

Max part size when uploading to S3 using multipart upload (previously named `maxPartSizeForCloudUploadInMB` prior to 4.3)

Default: 5 GB

We can perform a full migration even if those parameters have changed but the MD5 hash calculations would be different and you need to disable it for the moment. Several attempts were successful without MD5 hash checking (Rubrik MPU size tuned).

The next release of CDM (5.1) will change the partSize to 64 MB as a default parameter. So, the backup chain will include objects based on 16 MB chunks and 64 MB chunks.

There will be a new version of the tool to be compatible with the CDM 5.1 and new backup chains.

Performance & timing

As a reference and to evaluate the time you need to migrate a Rubrik Archival Location, we can achieve on a small Cloudian cluster :

- 5 nodes
- from RF3 to EC3+2 (so read & write on the same cluster)
- migration of 5 buckets with automatic mode
- 1562 objects
- ~ 3270 GB
- ~ 4,5 hours for the full migration : copy + copy-back (two copies + MD5 hash process which is time consuming and depends on the number of objects)
- ~ 400 MB/sec for the writes
- Cloudian HyperStore 7.1.7
- Rubrik CDM 5.0

Of course, those performances depend on the host which performs the migration (CPU, memory, etc).

2 Installation

You have two options :

1. Download the Debian OVA with all tools included for the migration and ready to go (see next chapter).
2. Follow the instructions below and install the tools on a machine of your choice.

You have to download and install several files :

- the tool/script : s3-bucket-migration.sh
- the python “packages” : s3cmd & s4cmd

Proceed to the package installations – depending on your Operating System – or by using the python tool “pip3” (or pip) :

```
root@snoopy:~# pip3 install s3cmd
Collecting s3cmd
  Downloading s3cmd-2.1.0-py2.py3-none-any.whl (145 kB)
    |████████████████████████████████████████| 145 kB 8.9 MB/s
Requirement already satisfied: python-dateutil in /usr/lib/python3/dist-packages (from s3cmd) (2.7.3)
Requirement already satisfied: python-magic in /usr/lib/python3/dist-packages (from s3cmd) (0.4.16)
Installing collected packages: s3cmd
Successfully installed s3cmd-2.1.0
root@snoopy:~#
root@snoopy:~#
root@snoopy:~# pip3 install s4cmd
Collecting s4cmd
  Downloading s4cmd-2.1.0.tar.gz (27 kB)
Requirement already satisfied: boto3>=1.3.1 in /usr/lib/python3/dist-packages (from s4cmd) (1.9.253)
Requirement already satisfied: pytz>=2016.4 in /usr/lib/python3/dist-packages (from s4cmd) (2019.3)
Building wheels for collected packages: s4cmd
  Building wheel for s4cmd (setup.py) ... done
  Created wheel for s4cmd: filename=s4cmd-2.1.0-py3-none-any.whl size=46575
sha256=3de1b3b51df8c055aba5a7465a134187e0419f9cec52be96340090abf6ad7686
  Stored in directory: /root/.cache/pip/wheels/a6/5b/b1/ec0b40be095faaf0147e66cb459af8c5adc61ad8d7d4cbe935
Successfully built s4cmd
Installing collected packages: s4cmd
Successfully installed s4cmd-2.1.0
root@snoopy:~#
```

For this documentation, I will use my Debian packages :

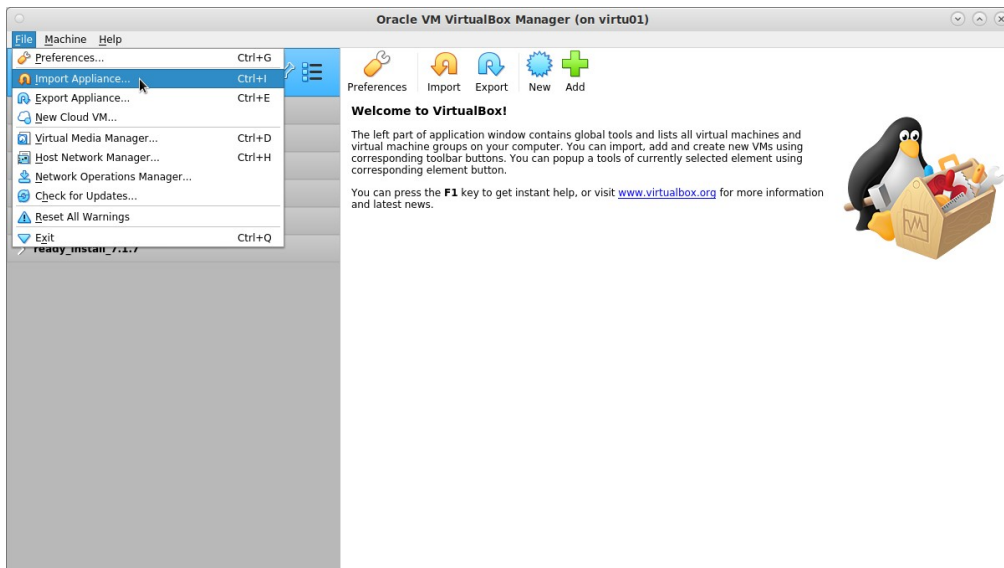
```
plong@snoopy:~$ s3cmd --version
s3cmd version 2.1.0
plong@snoopy:~$ s4cmd --version
s4cmd version 2.1.0
```

The command-line s3cmd and s4cmd should be available in : **/usr/bin** or **/usr/local/bin**

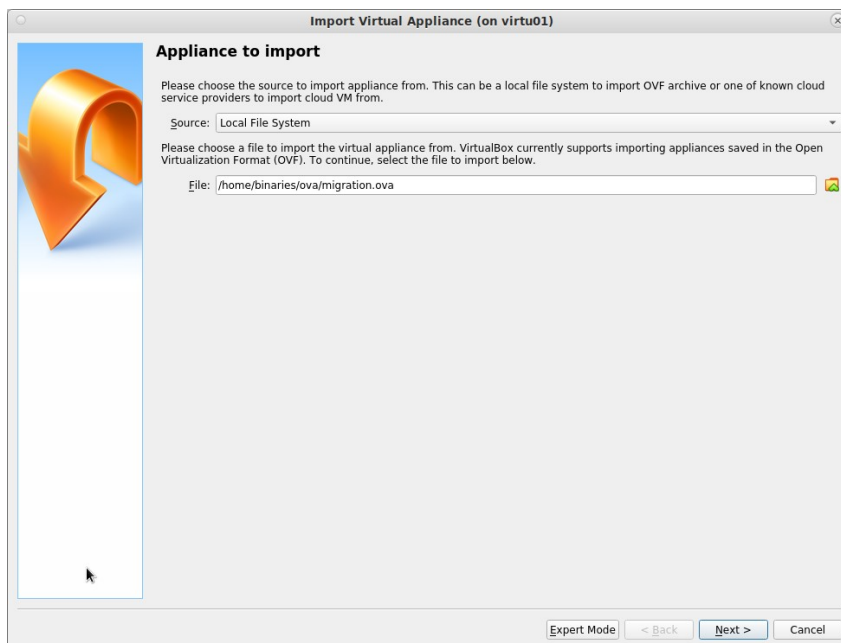
```
plong@snoopy:~$ which s3cmd
/usr/bin/s3cmd
plong@snoopy:~$ which s4cmd
/usr/bin/s4cmd
```

3 Deploy OVA

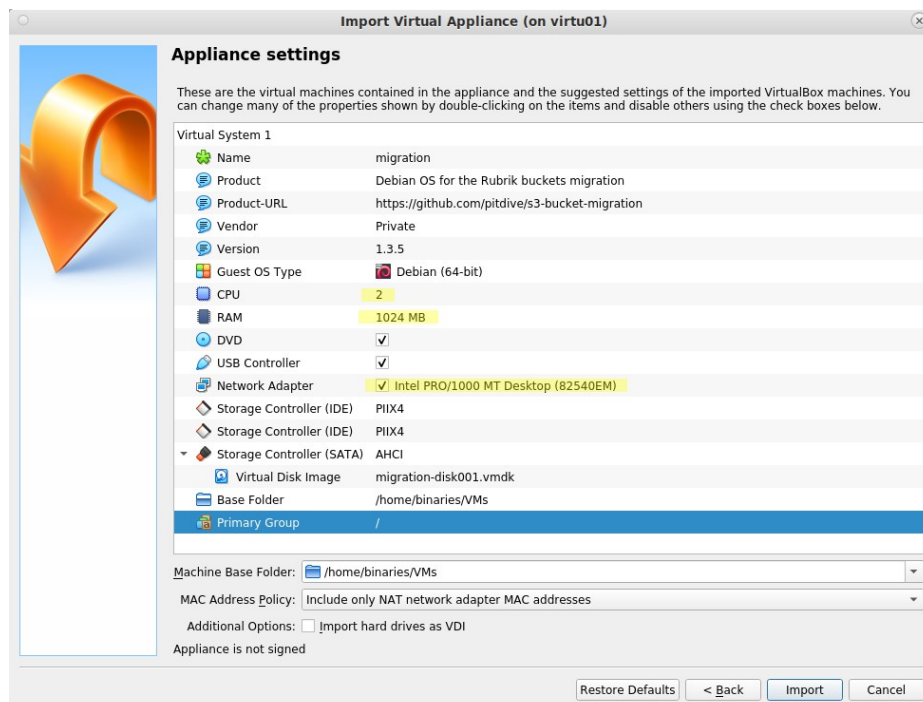
If you decided to download the OVA, here are some screenshots and steps to help you to deploy the OVA under VirtualBox (<https://www.virtualbox.org/>). First, run virtualbox and import the OVA.



Select the OVA file :



Customize the Virtual Machine and adjust some parameters :



- the CPU
- the RAM
(increase it to 10 – 16 GB if possible)
- the network card
(use a 10Gbit/s network card)

Click on the button “Import” and wait a couple of minutes.

Then, start the virtual machine :

user : migration
password : migration

root access
password : password

**** US keyboard ****

Ready to continue to the next chapter.

4 Configuration

Firstly, you may want to update the tool version.

```
migration@migration:~$ ./s3-bucket-migration.sh -c update

Disclaimer / Warning

Use this tool with precautions (review the config file created manually for a double-check) for your environment :
it is NOT an official tool supported by Cloudian.

Cloudian can NOT be involved for any bugs or misconfiguration due to this tool. So you are using it at your own
risks and be aware of the restrictions.

Trying to retrieve the latest version of the script with the default network access ...

2020-05-14 11:16:16 URL:https://raw.githubusercontent.com/pitdive/s3-bucket-migration/s4cmd/s3-bucket-migration.sh
[18878/18878] -> "s3-bucket-migration.sh" [1]

Keeping the oldest version with the name : ./s3-bucket-migration.sh.oldversion

Script upgraded to VERSION="v1.3.5"
```

Then, run the tool to configure and initialize the s3 access :

→ the answers provided in the example below must be adjusted with your environment.

```
plong@snoopy:~$ ./s3-bucket-migration.sh -c configure

Disclaimer / Warning

Use this tool with precautions (review the config file created manually for a double-check) for your environment :
it is NOT an official tool supported by Cloudian.

Cloudian can NOT be involved for any bugs or misconfiguration due to this tool. So you are using it at your own
risks and be aware of the restrictions.

Enter new values or accept defaults in brackets with Enter.
Refer to user manual for detailed description of all options.

Access key and Secret key are your identifiers for Amazon S3. Leave them empty for using the env variables.
Access Key: 590bf0fbeat8ca20fbc5
Secret Key: 25D88mu1gIDP7fzvdbgpmbscoCQp/bU2ciB/ZDUd
Default Region [US]: cloudlab

Use "s3.amazonaws.com" for S3 Endpoint and not modify it to the target Amazon S3.
S3 Endpoint [s3.amazonaws.com]: s3-cloudlab.demo.lab

Use "%(bucket)s.s3.amazonaws.com" to the target Amazon S3. "%(bucket)s" and "%(location)s" vars can be used
if the target S3 system supports dns based buckets.
DNS-style bucket+hostname:port template for accessing a bucket [% (bucket)s.s3.amazonaws.com]: %(bucket)s.s3-
cloudlab.demo.lab

Encryption password is used to protect your files from reading
by unauthorized persons while in transfer to S3
Encryption password:
Path to GPG program [/usr/bin/gpg]:

When using secure HTTPS protocol all communication with Amazon S3
servers is protected from 3rd party eavesdropping. This method is
slower than plain HTTP, and can only be proxied with Python 2.7 or newer
Use HTTPS protocol [Yes]: no

On some networks all internet access must go through a HTTP proxy.
Try setting it here if you can't connect to S3 directly
HTTP Proxy server name:

New settings:
Access Key: 590bf0fbeat8ca20fbc5
Secret Key: 25D88mu1gIDP7fzvdbgpmbscoCQp/bU2ciB/ZDUd
Default Region: cloudlab
```

```

S3 Endpoint: s3-cloudlab.demo.lab
DNS-style bucket+hostname:port template for accessing a bucket: %(bucket)s.s3-cloudlab.demo.lab
Encryption password:
Path to GPG program: /usr/bin/gpg
Use HTTPS protocol: False
HTTP Proxy server name:
HTTP Proxy server port: 0

Test access with supplied credentials? [Y/n] y
Please wait, attempting to list all buckets...
Success. Your access key and secret key worked fine :-)

Now verifying that encryption works...
Not configured. Never mind.

Save settings? [y/N] y
Configuration saved to 's3cfg.conf'
Please provide the @IP or name of the Clodian puppet master : cloudlab01
Adding config into : s3cfg.conf
PING cloudlab01 (192.168.110.201) 56(84) bytes of data.
64 bytes from cloudlab01 (192.168.110.201): icmp_seq=1 ttl=64 time=0.536 ms

--- cloudlab01 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.536/0.536/0.536/0.000 ms
The authenticity of host 'cloudlab01 (192.168.110.201)' can't be established.
ECDSA key fingerprint is SHA256:GndvX4wgQsAUXvAJJsfs93PrewPsGJcbKgRNphDaBd4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new
keys
root@cloudlab01's password: *****

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@cloudlab01'"
and check to make sure that only the key(s) you wanted were added.

Please provide the Sysadmin password for Admin API connection : public
Configuration done.
You can now continue for the automatic migration for multi-buckets with the command :
./s3-bucket-migration.sh -c auto -b <bucketname>

```

The configuration is done and now the tool is able to perform the migration without requesting any password or access key. Next step, migration of a set of buckets.

5 Look at the tool

If you want to “test” the tool with a set of Rubrik buckets and see the impact on the buckets, you can use the option “**howmany**”, there is no migration requested in this option and we will only show you “How Many” buckets would be migrated with the “auto” migration.

```
migration@migration:~$ ./s3-bucket-migration.sh -c howmany -b vm-bronze-rubrik-0

Disclaimer / Warning
Use this tool with precautions (review the config file created manually for a double-check) for your environment :
it is NOT an official tool supported by Cloudian.

Cloudian can NOT be involved for any bugs or misconfiguration due to this tool. So you are using it at your own
risks and be aware of the restrictions.

- We plan to migrate this : -
-->  vm-bronze-rubrik-0
-->  vm-bronze-rubrik-1
-->  vm-bronze-rubrik-2
- End of listing -
```

In the case of a migration with the option “auto”, there is always a request for an approval before starting. This request has also a timeout which cancel the request if it’s over (60 seconds).

```
migration@migration:~$ ./s3-bucket-migration.sh -c auto -b vm-bronze-rubrik-0

Disclaimer / Warning
Use this tool with precautions (review the config file created manually for a double-check) for your environment :
it is NOT an official tool supported by Cloudian.

Cloudian can NOT be involved for any bugs or misconfiguration due to this tool. So you are using it at your own
risks and be aware of the restrictions.

- We will plan to migrate this : -
-->  vm-bronze-rubrik-0
-->  vm-bronze-rubrik-1
-->  vm-bronze-rubrik-2
- End of listing -

- Checking MPU size for all the buckets ... -
vm-bronze-rubrik-0 --> Biggest object SIZE is : 4401  which is smaller than the max size MPU x PART: 167772160000
vm-bronze-rubrik-1 --> Biggest object SIZE is : 4401  which is smaller than the max size MPU x PART: 167772160000
vm-bronze-rubrik-2 --> Biggest object SIZE is : 10965988  which is smaller than the max size MPU x PART:
167772160000
...looks good
Do you want to continue (type : yes to continue, or anything else to abort) ? Well, aborting the script.
```

6 Full example

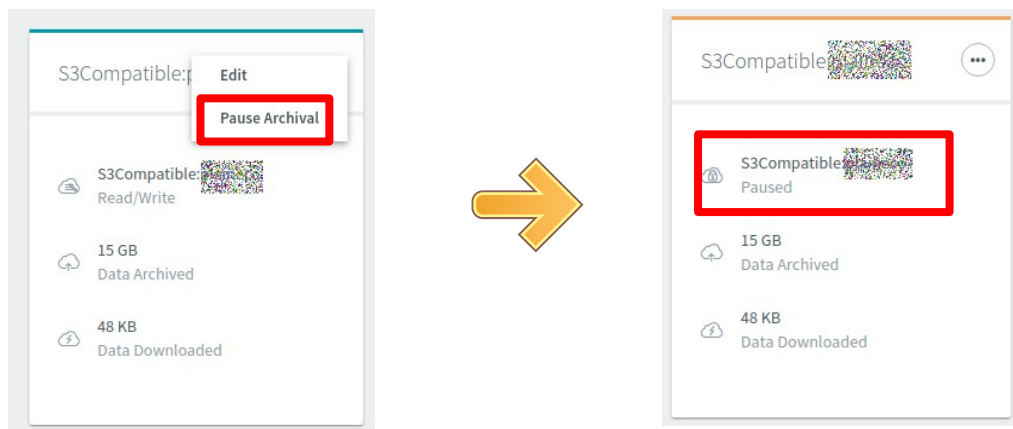
BUCKETS		OBJECTS	
NAME	REGION		
vm-bronze-rubrik-0	cloudlab		
vm-bronze-rubrik-1	cloudlab		
vm-bronze-rubrik-2	cloudlab		
vm-gold-rubrik-0	cloudlab		
vm-gold-rubrik-1	cloudlab		
vm-gold-rubrik-2	cloudlab		
vm-silver-rubrik-0	cloudlab		
vm-silver-rubrik-1	cloudlab		
vm-silver-rubrik-2	cloudlab		

Here is the listing of the Rubrik buckets to migrate.

For the first migration, we are going to migrate the “vm-bronze” SLA composed of 3 buckets :

- vm-bronze-rubrik-0
- vm-bronze-rubrik-1
- vm-bronze-rubrik-2

Then, you have to go to the Rubrik GUI and force the “Pause Archival” mode on the ArchivalLocation before running the tool (avoid upload of new objects during the full sync.) :



*** remember : disabling the archiving consolidation is recommended for the migration ***

*** You can re-enable it after the migration ***

Next, check on the Cloudian cluster that the default storage policy is the new policy chosen for the new buckets which will be created during the migration.

```

plong@snoopy:~$ ./s3-bucket-migration.sh -c auto -b vm-bronze-rubrik-0

Disclaimer / Warning
Use this tool with precautions (review the config file created manually for a double-check) for your environment :
it is NOT an official tool supported by Cloudian.
Cloudian can NOT be involved for any bugs or misconfiguration due to this tool. So you are using it at your own
risks and be aware of the restrictions.

- We plan to migrate this : -

-->  vm-bronze-rubrik-0
-->  vm-bronze-rubrik-1
-->  vm-bronze-rubrik-2

- End of listing -

- Checking MPU size for all the buckets ... -

vm-bronze-rubrik-0 --> Biggest object SIZE is : 15 which is smaller than the max size MPU x PART: 167772160000
vm-bronze-rubrik-1 --> Biggest object SIZE is : 10965988 which is smaller than the max size MPU x PART:
167772160000
vm-bronze-rubrik-2 --> Biggest object SIZE is : 15 which is smaller than the max size MPU x PART: 167772160000
...looks good

Do you want to continue (type : yes to continue, or anything else to abort) ? yes
OK, let's Go !
--- Preparing to migrate the bucket : vm-bronze-rubrik-0 ---
Checking the source bucket : vm-bronze-rubrik-0
s3://vm-bronze-rubrik-0/ (bucket):
=> Bucket vm-bronze-rubrik-0 is available and readable. Continuing ...

Creation of a temporary bucket to migrate firstly the data ... : vm-bronze-rubrik-0-tempo
Bucket vm-bronze-rubrik-0-tempo is now available. Continuing ...
--- Synchronization from the bucket vm-bronze-rubrik-0 to the bucket vm-bronze-rubrik-0-tempo : ---
Checking for big files and MPU size compatibility ...
#
Size and number of objects for the buckets :
      15      1 objects s3://vm-bronze-rubrik-0/
      15      1 objects s3://vm-bronze-rubrik-0-tempo/

Sync is finished.
Next step --> reverse command. ./s3-bucket-migration.sh -c reverse -b vm-bronze-rubrik-0

--- Check the content of the buckets ---
MD5 calculations for vm-bronze-rubrik-0 and vm-bronze-rubrik-0-tempo
This might take long time and might use a lot of ressources ... depending on the structure of the buckets
Done. All checksums are in the log file : vm-bronze-rubrik-0.log

Hashs are good, let's continue ...

--- Reverse the operations from the bucket : vm-bronze-rubrik-0-tempo ---
Purging the bucket : vm-bronze-rubrik-0
Need to connect to the cluster by using SSH and root access...
Bucket: vm-bronze-rubrik-0 purged.
Waiting a couple of seconds to apply changes ...
Creating a new bucket : vm-bronze-rubrik-0
Delete is finished, if there is no error, please Go Forward ...
Next step --> Copyback command. ./s3-bucket-migration.sh -c copyback -b vm-bronze-rubrik-0

--- Copy-back from the bucket : vm-bronze-rubrik-0-tempo to the bucket : vm-bronze-rubrik-0 ---
Copy-back in progress ...
#
Size and number of objects for the buckets : vm-bronze-rubrik-0
      15      1 objects s3://vm-bronze-rubrik-0/
      15      1 objects s3://vm-bronze-rubrik-0-tempo/

Done.
Next step --> Clean-up command. ./s3-bucket-migration.sh -c clean -b vm-bronze-rubrik-0

--- Check the content of the buckets ---
MD5 calculations for vm-bronze-rubrik-0 and vm-bronze-rubrik-0-tempo
This might take long time and might use a lot of ressources ... depending on the structure of the buckets
Done. All checksums are in the log file : vm-bronze-rubrik-0.log

Hashs are good, let's continue ...

--- Clean-up and deletion of non-necessary objects + bucket... ---
Purging the bucket : vm-bronze-rubrik-0-tempo
Need to connect to the cluster by using SSH and root access...
Bucket: vm-bronze-rubrik-0-tempo purged.
Waiting a couple of seconds to apply changes ...
Cleaning unnecessary log files ...
Job done

--- Preparing to migrate the bucket : vm-bronze-rubrik-1 ---
Checking the source bucket : vm-bronze-rubrik-1
s3://vm-bronze-rubrik-1/ (bucket):
=> Bucket vm-bronze-rubrik-1 is available and readable. Continuing ...

Creation of a temporary bucket to migrate firstly the data ... : vm-bronze-rubrik-1-tempo
Bucket vm-bronze-rubrik-1-tempo is now available. Continuing ...
--- Synchronization from the bucket vm-bronze-rubrik-1 to the bucket vm-bronze-rubrik-1-tempo : ---
Checking for big files and MPU size compatibility ...
#####
Size and number of objects for the buckets :
493518782      14 objects s3://vm-bronze-rubrik-1/
493518782      14 objects s3://vm-bronze-rubrik-1-tempo/

Sync is finished.
Next step --> reverse command. ./s3-bucket-migration.sh -c reverse -b vm-bronze-rubrik-1

```

```

--- Check the content of the buckets ---
MD5 calculations for vm-bronze-rubrik-1 and vm-bronze-rubrik-1-tempo
This might take long time and might use a lot of ressources ... depending on the structure of the buckets
Done. All checksums are in the log file : vm-bronze-rubrik-0.log

Hashs are good, let's continue ...

--- Reverse the operations from the bucket : vm-bronze-rubrik-1-tempo ---
Purging the bucket : vm-bronze-rubrik-1
Need to connect to the cluster by using SSH and root access...
Bucket: vm-bronze-rubrik-1 purged.
Waiting a couple of seconds to apply changes ...
Creating a new bucket : vm-bronze-rubrik-1
Delete is finished, if there is no error, please Go Forward ...
Next step --> Copyback command. ./s3-bucket-migration.sh -c copyback -b vm-bronze-rubrik-1

--- Copy-back from the bucket : vm-bronze-rubrik-1-tempo to the bucket : vm-bronze-rubrik-1 ---
Copy-back in progress ...
#####
Size and number of objects for the buckets : vm-bronze-rubrik-1
493518782      14 objects s3://vm-bronze-rubrik-1/
493518782      14 objects s3://vm-bronze-rubrik-1-tempo/

Done.
Next step --> Clean-up command. ./s3-bucket-migration.sh -c clean -b vm-bronze-rubrik-1

--- Check the content of the buckets ---
MD5 calculations for vm-bronze-rubrik-1 and vm-bronze-rubrik-1-tempo
This might take long time and might use a lot of ressources ... depending on the structure of the buckets
Done. All checksums are in the log file : vm-bronze-rubrik-0.log

Hashs are good, let's continue ...

--- Clean-up and deletion of non-necessary objects + bucket... ---
Purging the bucket : vm-bronze-rubrik-1-tempo
Need to connect to the cluster by using SSH and root access...
Bucket: vm-bronze-rubrik-1-tempo purged.
Waiting a couple of seconds to apply changes ...
Cleaning unnecessary log files ...
Job done

--- Preparing to migrate the bucket : vm-bronze-rubrik-2 ---
Checking the source bucket : vm-bronze-rubrik-2
s3://vm-bronze-rubrik-2/ (bucket):
=> Bucket vm-bronze-rubrik-2 is available and readable. Continuing ...

Creation of a temporary bucket to migrate firstly the data ... : vm-bronze-rubrik-2-tempo
Bucket vm-bronze-rubrik-2-tempo is now available. Continuing ...
--- Synchronization from the bucket vm-bronze-rubrik-2 to the bucket vm-bronze-rubrik-2-tempo : ---
Checking for big files and MPU size compatibility ...
#
Size and number of objects for the buckets :
      15      1 objects s3://vm-bronze-rubrik-2/
      15      1 objects s3://vm-bronze-rubrik-2-tempo/

Sync is finished.
Next step --> reverse command. ./s3-bucket-migration.sh -c reverse -b vm-bronze-rubrik-2

--- Check the content of the buckets ---
MD5 calculations for vm-bronze-rubrik-2 and vm-bronze-rubrik-2-tempo
This might take long time and might use a lot of ressources ... depending on the structure of the buckets
Done. All checksums are in the log file : vm-bronze-rubrik-0.log

Hashs are good, let's continue ...

--- Reverse the operations from the bucket : vm-bronze-rubrik-2-tempo ---
Purging the bucket : vm-bronze-rubrik-2
Need to connect to the cluster by using SSH and root access...
Bucket: vm-bronze-rubrik-2 purged.
Waiting a couple of seconds to apply changes ...
Creating a new bucket : vm-bronze-rubrik-2
Delete is finished, if there is no error, please Go Forward ...
Next step --> Copyback command. ./s3-bucket-migration.sh -c copyback -b vm-bronze-rubrik-2

--- Copy-back from the bucket : vm-bronze-rubrik-2-tempo to the bucket : vm-bronze-rubrik-2 ---
Copy-back in progress ...
#
Size and number of objects for the buckets : vm-bronze-rubrik-2
      15      1 objects s3://vm-bronze-rubrik-2/
      15      1 objects s3://vm-bronze-rubrik-2-tempo/

Done.
Next step --> Clean-up command. ./s3-bucket-migration.sh -c clean -b vm-bronze-rubrik-2

--- Check the content of the buckets ---
MD5 calculations for vm-bronze-rubrik-2 and vm-bronze-rubrik-2-tempo
This might take long time and might use a lot of ressources ... depending on the structure of the buckets
Done. All checksums are in the log file : vm-bronze-rubrik-0.log

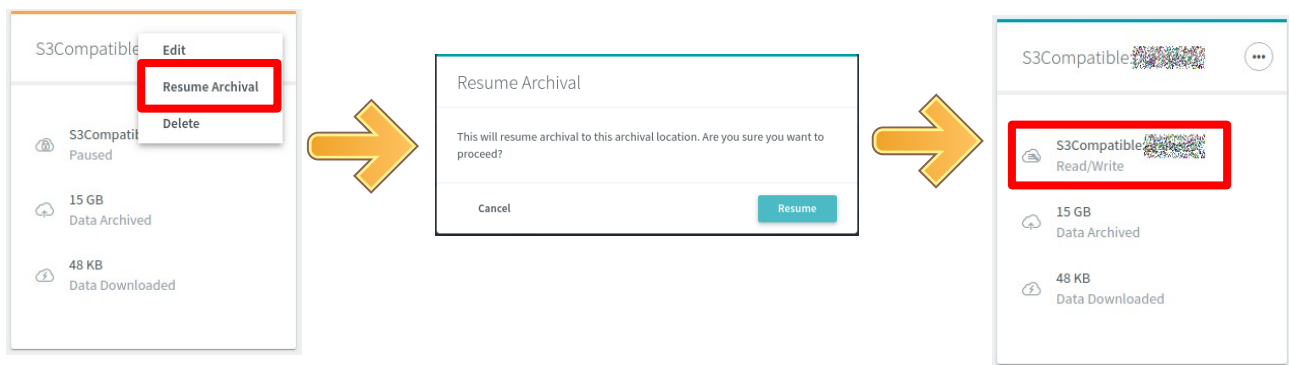
Hashs are good, let's continue ...

--- Clean-up and deletion of non-necessary objects + bucket... ---
Purging the bucket : vm-bronze-rubrik-2-tempo
Need to connect to the cluster by using SSH and root access...
Bucket: vm-bronze-rubrik-2-tempo purged.
Waiting a couple of seconds to apply changes ...
Cleaning unnecessary log files ...
Job done

Nothing else to do.
Please resume the Archival Location matching the current migration

```

Last step, you have to resume the archival on the Archival Location (Rubrik GUI) :



7 Logs

During the migration, the tool will log all the steps, all the copies and all the md5 hashes. For the migration of the buckets “vm-bronze-rubrik-0”, there will be a file based with the bucketname with the extension “.log” like :

vm-bronze-rubrik-0.log

Example

Example of log file with a full migration of Rubrik buckets on a customer environment :

log file : migration2-rubrik-0.log

```
Migration of migration2-rubrik-0 is starting ... --->>>
*** Preparation before SYNC ***
vendredi 10 avril 2020, 11:23:50 (UTC+0200)
Bucket 's3://migration2-rubrik-0-tempo/' created

*** Sync in progress ***
s3://migration2-rubrik-0/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/407c8d61-6609-435b-8bd9-1d9d6419129e/4291cbbd-66ec-4d53-b9f5-2e496c68dcc6_3b1a7dca-6a2f-4133-b59e-6862a05b8b67/data.patch__metadata =>
s3://migration2-rubrik-0-tempo/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/407c8d61-6609-435b-8bd9-1d9d6419129e/4291cbbd-66ec-4d53-b9f5-2e496c68dcc6_3b1a7dca-6a2f-4133-b59e-6862a05b8b67/data.patch__metadata
... cutting lines ...
s3://migration2-rubrik-0/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/9e4bb77f-9d38-4fc0-8a75-acf091d2fc42/EMPTY_3c6a2639-814f-4c85-af38-08798396ee00/data.patch => s3://migration2-rubrik-0-tempo/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/9e4bb77f-9d38-4fc0-8a75-acf091d2fc42/EMPTY_3c6a2639-814f-4c85-af38-08798396ee00/data.patch

*** Get the content of the bucket migration2-rubrik-0-tempo ***
2020-04-10 09:23 748305104 s3://migration2-rubrik-0-tempo/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/11e211de-ef96-4f9f-ae55-093d41dbc141/bf934493-7f40-48e5-966d-0fd9ea736e7c_5c38413f-1310-4d09-b525-2ffadb3de7ce/data.patch
... cutting lines ...
2020-04-10 09:23 64 s3://migration2-rubrik-0-tempo/snappables/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/edaa2698-5734-47d0-9447-189f9e986416/index/filesystem.metadata__metadata

*** Calculating objects and size ***
Size and number of objects for the buckets migration2-rubrik-0
30243013152 110 objects s3://migration2-rubrik-0/

Size and number of objects for the bucket migration2-rubrik-0-tempo
30243013152 110 objects s3://migration2-rubrik-0-tempo/

*** MD5 calculations ***
vendredi 10 avril 2020, 11:25:12 (UTC+0200)
----- BUCKET migration2-rubrik-0 -----
2020-04-09 22:09 748305104 c9aa9f0550864f4d11a53e6b258ef7c3-45
s3://migration2-rubrik-0/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/11e211de-ef96-4f9f-ae55-093d41dbc141/bf934493-7f40-48e5-966d-0fd9ea736e7c_5c38413f-1310-4d09-b525-2ffadb3de7ce/data.patch
... cutting lines ...
2020-04-06 11:43 64 edfb5ad7554a06be1778ff88a18600d8
s3://migration2-rubrik-0/snappables/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/edaa2698-5734-47d0-9447-189f9e986416/index/filesystem.metadata__metadata
----- BUCKET migration2-rubrik-0-tempo -----
2020-04-10 09:23 748305104 c9aa9f0550864f4d11a53e6b258ef7c3-45
s3://migration2-rubrik-0-tempo/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/11e211de-ef96-4f9f-ae55-093d41dbc141/bf934493-7f40-48e5-966d-0fd9ea736e7c_5c38413f-1310-4d09-b525-2ffadb3de7ce/data.patch
... cutting lines ...
2020-04-10 09:23 64 edfb5ad7554a06be1778ff88a18600d8
s3://migration2-rubrik-0-tempo/snappables/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/edaa2698-5734-47d0-9447-189f9e986416/index/filesystem.metadata__metadata

*** There is no error on #objects, bucket size and MD5 hashes ***

*** Reverse in progress for the bucket migration2-rubrik-0-tempo ***
vendredi 10 avril 2020, 11:25:12 (UTC+0200)
Purging the bucket : migration2-rubrik-0
Bucket 's3://migration2-rubrik-0/' removed
Bucket 's3://migration2-rubrik-0/' created

*** Copy-Back in progress from the bucket migration2-rubrik-0-tempo to the bucket migration2-rubrik-0 ***
...
*** Calculating objects and size ***
Size and number of objects for the bucket migration2-rubrik-0
30243013152 110 objects s3://migration2-rubrik-0/

Size and number of objects for the bucket migration2-rubrik-0-tempo
30243013152 110 objects s3://migration2-rubrik-0-tempo/
```



```

*** MD5 calculations ***
vendredi 10 avril 2020, 11:26:40 (UTC+0200)
----- BUCKET migration2-rubrik-0 -----
2020-04-10 09:25 748305104 c9aa9f0550864f4d11a53e6b258ef7c3-45
s3://migration2-rubrik-0/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/11e211de-ef96-4f9f-ae5-093d41dbc141/bf934493-7f40-48e5-966d-0fd9ea736e7c_5c38413f-1310-4d09-b525-2ffadb3de7ce/data.patch
... cutting lines ...
2020-04-10 09:25 64 edfb5ad7554a06be1778ff88a18600d8
s3://migration2-rubrik-0/snappables/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/edaa2698-5734-47d0-9447-189f9e986416/index/filesystem.metadata__metadata
----- BUCKET migration2-rubrik-0-tempo -----
2020-04-10 09:23 748305104 c9aa9f0550864f4d11a53e6b258ef7c3-45
s3://migration2-rubrik-0-tempo/blobstore/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051-2000_0_536870912000/11e211de-ef96-4f9f-ae5-093d41dbc141/bf934493-7f40-48e5-966d-0fd9ea736e7c_5c38413f-1310-4d09-b525-2ffadb3de7ce/data.patch
... cutting lines ...
2020-04-10 09:23 64 edfb5ad7554a06be1778ff88a18600d8
s3://migration2-rubrik-0-tempo/snappables/430/295/4c3a1b73-227a-4482-b758-39ff4c9ba081-vm-15051/edaa2698-5734-47d0-9447-189f9e986416/index/filesystem.metadata__metadata

*** There is no error on #objects, bucket size and MD5 hashes ***

*** Cleanup in progress from the bucket migration2-rubrik-0-tempo ***
vendredi 10 avril 2020, 11:26:41 (UTC+0200)
Purging the bucket : migration2-rubrik-0-tempo
Bucket 's3://migration2-rubrik-0-tempo/' removed
<<<--- *** Job done ***

*** Preparation before SYNC ***
vendredi 10 avril 2020, 11:26:47 (UTC+0200)
Bucket 's3://migration2-rubrik-1-tempo/' created

*** Sync in progress ***
s3://migration2-rubrik-1/rubrik_encryption_key_check.txt =>
s3://migration2-rubrik-1-tempo/rubrik_encryption_key_check.txt
s3://migration2-rubrik-1/rubrik_cluster_lock.txt => s3://migration2-rubrik-1-tempo/rubrik_cluster_lock.txt

*** Get the content of the bucket migration2-rubrik-1-tempo ***
2020-04-10 09:26 160 s3://migration2-rubrik-1-tempo/rubrik_cluster_lock.txt
2020-04-10 09:26 32 s3://migration2-rubrik-1-tempo/rubrik_encryption_key_check.txt

*** Calculating objects and size ***
Size and number of objects for the buckets migration2-rubrik-1
192 2 objects s3://migration2-rubrik-1/

Size and number of objects for the bucket migration2-rubrik-1-tempo
192 2 objects s3://migration2-rubrik-1-tempo/

*** MD5 calculations ***
vendredi 10 avril 2020, 11:26:50 (UTC+0200)
----- BUCKET migration2-rubrik-1 -----
2020-04-06 06:39 160 5a53822a26f1f0efbebf6192d18e2679 s3://migration2-rubrik-1/rubrik_cluster_lock.txt
2020-04-06 06:39 32 cb35f88c7b4f32a6da001c22d18388cb
s3://migration2-rubrik-1/rubrik_encryption_key_check.txt
----- BUCKET migration2-rubrik-1-tempo -----
2020-04-10 09:26 160 5a53822a26f1f0efbebf6192d18e2679
s3://migration2-rubrik-1-tempo/rubrik_cluster_lock.txt
2020-04-10 09:26 32 cb35f88c7b4f32a6da001c22d18388cb
s3://migration2-rubrik-1-tempo/rubrik_encryption_key_check.txt

*** There is no error on #objects, bucket size and MD5 hashes ***

*** Reverse in progress for the bucket migration2-rubrik-1-tempo ***
vendredi 10 avril 2020, 11:26:51 (UTC+0200)
Purging the bucket : migration2-rubrik-1
Bucket 's3://migration2-rubrik-1/' removed
Bucket 's3://migration2-rubrik-1/' created

*** Copy-Back in progress from the bucket migration2-rubrik-1-tempo to the bucket migration2-rubrik-1 ***
vendredi 10 avril 2020, 11:26:56 (UTC+0200)
s3://migration2-rubrik-1-tempo/rubrik_encryption_key_check.txt =>
s3://migration2-rubrik-1/rubrik_encryption_key_check.txt
s3://migration2-rubrik-1-tempo/rubrik_cluster_lock.txt => s3://migration2-rubrik-1/rubrik_cluster_lock.txt

*** Calculating objects and size ***
Size and number of objects for the bucket migration2-rubrik-1
192 2 objects s3://migration2-rubrik-1/

Size and number of objects for the bucket migration2-rubrik-1-tempo
192 2 objects s3://migration2-rubrik-1-tempo/

*** MD5 calculations ***
vendredi 10 avril 2020, 11:26:59 (UTC+0200)
----- BUCKET migration2-rubrik-1 -----
2020-04-10 09:26 160 5a53822a26f1f0efbebf6192d18e2679 s3://migration2-rubrik-1/rubrik_cluster_lock.txt
2020-04-10 09:26 32 cb35f88c7b4f32a6da001c22d18388cb
s3://migration2-rubrik-1/rubrik_encryption_key_check.txt
----- BUCKET migration2-rubrik-1-tempo -----
2020-04-10 09:26 160 5a53822a26f1f0efbebf6192d18e2679
s3://migration2-rubrik-1-tempo/rubrik_cluster_lock.txt
2020-04-10 09:26 32 cb35f88c7b4f32a6da001c22d18388cb
s3://migration2-rubrik-1-tempo/rubrik_encryption_key_check.txt

*** There is no error on #objects, bucket size and MD5 hashes ***

*** Cleanup in progress from the bucket migration2-rubrik-1-tempo ***
vendredi 10 avril 2020, 11:26:59 (UTC+0200)
Purging the bucket : migration2-rubrik-1-tempo
Bucket 's3://migration2-rubrik-1-tempo/' removed
<<<--- *** Job done ***

```