Transfert S3 DCSR to other support

Data in the S3 DCSR should be transfert to another file system as soon as possible. There is no backup for S3 data. This documentation describes the transfert using Curnagl cluster and the rclone command.

Introduction

What is S3?

Amazon S3 (Simple Storage Service) is a scalable object storage service used for storing and retrieving any amount of data at any time. It organizes data into containers called “buckets.” Each bucket can store an unlimited number of objects, which are the fundamental entities stored in S3.

Understanding S3 Bucket Structure:

- **Buckets**: These are the top-level containers in S3. Each bucket has a unique name and is used to store objects.
- **Objects**: These are the files stored in a bucket. Each object is identified by a unique key (or ID) within the bucket.
- **Object Keys**: While S3 does not have a traditional file system hierarchy, it uses a flat namespace. The / character in object keys is used to simulate a directory structure, making it easier to organize and manage objects. However, these are not actual directories but part of the object’s key. S3 Endpoint Access

Accessing S3 is similar to accessing any other web service over HTTP, which most users are already familiar with. The endpoint URL follows the same structure as a typical web address, making it straightforward to understand and use.

An S3 endpoint address typically looks like this: https://dnsname.com/bucket-name/object-key

- Endpoint: https://dnsname.com
- Bucket Name: bucket-name
- Object Key: object-key

For example, if you have a bucket named my-bucket and an object with the key folder1/file.txt, the S3 URL would be: https://dnsname.com/my-bucket/folder1/file.txt

IAM Key Pairs

To access and manage your S3 resources securely, you will use IAM (Identity and Access Management) key pairs instead of a traditional login and password. An IAM key pair consists of an Access Key ID and a Secret Access Key. These keys are used to authenticate your requests to AWS services. • Access Key ID: This is similar to a username. • Secret Access Key: This is similar to a password and should be kept secure.

Unlike a traditional login and password, different IAM key pairs can be attached to different sets of permissions defined in their policy files. These policies control what actions the keys are allowed to perform, enhancing security by ensuring that each key pair has only the necessary permissions for its intended tasks.

Requirements

- Have an account in the cluster
- Enough space in NAS or work to transfert the data

Rclone configuration

Use a text editor to create a configuration file in your home directory. Be sure to replace the S3 server name and the cryptographic key values with the ones sent in the email S3 form DCSR.

```
mkdir -p ~/.config/rclone
nano ~/.config/rclone/rclone.conf
```

The configuration file should look like this :

```
[s3-dci-ro]
type = s3
provider = Other
access_key_id = T******************M
secret_access_key = S************************************i
region =
endpoint = https://scl-s3.unil.ch
```

For many different S3 tools, the pair of authentication/cryptographic keys have different names. For Rclone, they are named `access_key_id` and `secret_access_key`. Corresponding respectively to **Access key** and **Private key** in the mail sent by DCSR.

Next, secure your key file:

```
chmod 600 ~/.config/rclone/rclone.conf
```

Now, **s3-dci-ro** is a S3 configured connection alias that you can use in Rclone without repeating the connection information in the CLI.

s3-dci-ro: In this connection alias, the cryptographic keys are assigned to a user attached to a read-only policy on the S3 cluster. This prevents you from modifying or accidentally deleting your source data when using this connection alias.

Use Rclone in CLI on the Curnagl front node

List the content of your bucket named "bucket1" (This command only show the directories.).

```
rclone lsd s3-dci-ro:bucket1
```

You can also navigate sub-directories with the `rclone lsd` command:

```
rclone lsd s3-dci-ro:bucket1/dir1
```

You can use `rclone lsf` command to list the file and the folders.

Within an S3 cluster, all entities are represented as URLs that point to specific objects. These objects are stored uniformly, without any inherent hierarchical structure. The concept of "folders" does not truly exist. However, by convention, the "/" character in the Object IDs (the URLs) is interpreted as a folder delimiter by the S3 client application. Consequently, the "ls" command essentially performs a filtering and sorting operation on information stored at the same level. This approach does not scale well, hence, it is not advisable to execute an "ls" command on a significantly large number of files or objects.

The differents ways to do a listing of files and folders on S3 with Rclone are described on the following pages:

- rclone ls
- rclone lsl

- rclone lsd
- rclone lsf
- rclone lsjson

The command `rclone copy -v` can be utilized to copy all files from a source folder to a destination folder. It's important to note that `rclone` does not duplicate the initial folder, but only its file contents into the destination folder. Furthermore, `rclone` does not recopy a file if it already exists in the destination, allowing for the resumption of an interrupted copy operation.

When launching a copy operation in the background with an ampersand & or a screen/tmux, it is recommended to use a log file with the verbosity set to `-v`. This log file will collect information about the copied files, errors, and provide a status update every minute on the amount of data copied so far.

Here is an example of a command to copy a subset of your data from your DCI S3 bucket to an LTS sub-folder on the Isilon NAS of the DCSR. Please substitute the paths to be relevant for your use case.

```
rclone copy -v --log-file=$log_file.log $connection_alias:$bucket/$path $NAS_PATH
```

You need to adapt the following parameters:

- `$log file`: path to the rclone log file
- `$connection_alias`: connection alias (e.g., s3-dci-ro)
- `$bucket`: S3 bucket name sent by email
- `$path`: directory path you whant to access inside the bucket
- `$NAS_PATH`: This is your destination folder path on the DCSR NAS

It should give you something like:

```
rclone copy -v --log-file=./rclone_to_LTS.log s3-dci-ro:bucket/dir1/dir2
/nas/FAC/Faculty/Unit/PI/project/LTS/project_toto
```

If the copy operation is expected to take an extended period of time, and you need to disconnect your terminal sessions, you can execute the Rclone commands within a tmux session. Tmux is available on the Curnagle cluster. More information here on its usage.

To monitor the copy process and identify potential errors, you can view the progress of the copy operation by opening the Rclone log file using the Linux "tail" command:

```
tail -f rclone_to_LTS.log
```

Every minute, a consolidated status of the transfer will be displayed in the logs. You can exit the tail command by pressing `CTRL+C`.

Upon completion of the transfer, a summary of the copy process, including any errors, will be available at the end of the log file. It is recommended to verify that there are no errors for each copy session.

Job script template to perform copy

To transfer data from the S3 storage cluster to the `/scratch/...` or `/work/...` directory on **Curnagl**, you will need to modify the `rclone_ro_s3_copy.sh` SLURM submission file shown here.

```
#!/bin/bash -l

#SBATCH --mail-user $user.name@unil.ch
#SBATCH --job-name rclone_copy
#SBATCH --time 1-00:00:00
#SBATCH --mail-type ALL
#SBATCH --output %x-%j.out
#SBATCH --cpus-per-task 4
#SBATCH --mem 1G
#SBATCH --export NONE



## Name of your S3 Bucket (sent by email from DCSR)

S3_BUCKET_NAME=""

# Path to the source folder within the S3 bucket to be replicated by rclone

# (only de content of this folder will be coipied in the destination, not the folder itself !)

IN_BUCKET_SOURCE_PATH=""

# Path to the destination folder in which the data wil be copied

DESTINATION_PATH=""
```

```
# Do not change the code after this line

mkdir -p $DESTINATION_FOLDER_PATH

rclone copy -v --log-file=$SLURM_JOB_NAME.log --max-backlog=1000 s3-dci-
ro:$S3_BUCKET_NAME/$IN_BUCKET_SOURCE_PATH $DESTINATION_PATH
```

You should edit the previous file with your real email account and you should put a value for the `S3_BUCKET_NAME`, `IN_BUCKET_SOURCE_PATH` and `DESTINATION_PATH` variables.

Submit it from the front node of the Curnagl cluster with the `sbatch` command:

```
sbatch rclone_ro_s3_copy.sh
```

Please refrain from running more than one copy job at a time, either to the NAS or the HPC storage, as the IOPS on the storage systems on both the source and destination are limited resources.

Transfert file from cluster to S3

If you want to put thing on the S3, you should use the following command:

```
rclone copy -v /nas/FAC/Faculty/Unit/PI/project/LTS/project_toto s3-dci-ro:bucket/dir1/dir2
--s3-no-check-bucket
```

> it is important to not forget the option `--s3-no-check-bucket`

Performance expected

These are some measures taken on `18/09/2025`.

From S3 to NAS

Directory with several fies:

```
Transferred:       18.210 GiB / 18.210 GiB, 100%, 11.818 KiB/s, ETA 0s
Transferred:            9 / 9, 100%
```

```
Elapsed time:      3m47.2s
```

From /work to S3

One big file:

```
Transferred:        2.830 GiB / 2.830 GiB, 100%, 161.094 MiB/s, ETA 0s
Transferred:            1 / 1, 100%
Elapsed time:        18.9s
```

Directory with several files, 20GB:

```
Transferred:       18.358 GiB / 18.358 GiB, 100%, 197.257 MiB/s, ETA 0s
Transferred:         2093 / 2093, 100%
Elapsed time:      1m33.8s
```

Créé 2024-12-17 10:10:06 UTC par Cristian Ruiz
Mis à jour 2025-11-18 14:37:34 UTC par Cristian Ruiz