Check the integrity of a dataset stored on Amazon S3 V.2

Sonia García-Ruiz[1]

[1]University College London, University of London

Ryten Laboratory

Sonia García-Ruiz

VERSION 2

MAY 05, 2023

DISCLAIMER

All the steps indicated within this protocol are orientated to be used in an Ubuntu machine.
This protocol has been successfully tested on the following two Ubuntu versions:

- Ubuntu 16.04.6 LTS (Xenial Xerus)
- Ubuntu server 22.04.1 LTS (Jammy Jellyfish)

OPEN ACCESS

DOI: dx.doi.org/10.17504/protocols.io.n92ld9qy9g5b/v2

External link: https://github.com/SoniaRuiz/aws-s3-integrity-check

Protocol Citation: Sonia García-Ruiz 2023. Check the integrity of a dataset stored on Amazon S3. **protocols.io** https://dx.doi.org/10.17504/protocols.io.n92ld9qy9g5b/v2Version created by Sonia García-Ruiz

Protocol status: Working
We use this protocol and it's working

Created: Apr 13, 2023

Last Modified: May 05, 2023

PROTOCOL integer ID: 80463

Keywords: aws-s3, s3-bucket, aws-cli, md5, cloud-computing, etag, md5sum, integrity monitoring, cloud-storage

ABSTRACT

Background

Amazon Simple Storage Service (Amazon S3) has become a widely used and reliable platform for storing large biomedical datasets. However, unintended changes to the original data can occur during the data writing and transmission, ultimately altering the original contents of the transferred object and generating unexpected results when later accessed. Despite the interest in verifying end-to-end data integrity, there are no existing open-source and easy-to-use tools to accomplish this mission;

Results

To bridge this gap, here we present aws-s3-integrity-check, a user-friendly, lightweight and reliable bash tool to verify the integrity of a dataset stored within an Amazon S3 bucket. By using this tool, we completed the integrity verification of >392K genomic and transcriptomic records ranging between 5 Bytes and 10 Gigabytes (GB) in size and occupying a total of 4.5 Terabytes (TB) of Amazon S3 cloud storage space in less than 13.5 hours. The aws-s3-integrity-check tool also provides file-by-file on-screen and log-file-based information about the status of each individual integrity check;

Conclusions

To the best of our knowledge, the aws-s3-integrity-check bash tool is the only open-source tool that allows verifying the integrity of a dataset uploaded to the Amazon S3 Storage system in a quick, reliable and efficient manner. The aws-s3-integrity-check tool can be used to test any file type and file size and it is freely available for use and download at https://github.com/SoniaRuiz/aws-s3-integrity-check and https://hub.docker.com/r/soniaruiz/aws-s3-integrity-check.

MATERIALS

The dataset used in one of our tests was generated by Feleke, Reynolds et al. [DOI: 10.1007/s00401-021-02343-x] and is available under request from EGA with accession number EGAD00001009264.

The dataset used in one of our tests was a subset of the original dataset generated by Guelfi et al. [DOI: 10.1038/s41467-020-14483-x], which is available under request from EGA with accession number EGAS00001003065.

Log files produced during the testing phase of the aws-s3-integrity-check tool are available at https://github.com/SoniaRuiz/aws-s3-integrity-check/tree/master/logs.

The Dockerised version of this tool is available on Docker Hub: https://hub.docker.com/r/soniaruiz/aws-s3-integrity-check

BEFORE START INSTRUCTIONS

1. This protocol will guide you through the installation of the following software dependencies:

- jq (version jq-1.5-1-a5b5cbe, https://stedolan.github.io/jq/) . *jq* is a command-line JSON processor.
- xxd (version 1.10 27oct98 by Juergen Weigert, https://manpages.ubuntu.com/manpages/bionic/en/man1/xxd.1.html). *xxd* is a command-line binary/hexadecimal conversion tool.
- AWS Command Line Interface (CLI), (version 2, https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)
- s3md5 (https://github.com/antespi/s3md5)

2. This protocol will also ask you to authenticate on AWS using either the *'aws configure'* or *'aws configure sso'* command in your Linux command-line terminal.

IMPORTANT: for the correct operation of this protocol, **JSON** must be chosen as the preferred output format during the *'aws configure'* and *aws configure sso'* commands execution.

For example, in case you choose to authenticate on AWS using the *"aws configure"* command:

```
$ aws configure
AWS Access Key ID [None]: your_AWS_access_key_ID
AWS Secret Access Key [None]: your_AWS_secret_access_key
Default region name [None]: your_AWS_region_name
Default output format [None]: json
```

3. This protocol requires a dataset stored on an Amazon S3 bucket and an additional copy of that dataset stored locally. This requirement is due to the fact that the 'aws-s3-integrity-check' tool checks the integrity of a dataset on Amazon S3 by comparing the contents of those files with the contents of a local version of those files, so it can verify that the integrity of the files has not been compromised during the data transmission. Please, skip this requirement in case your files are already stored on an Amazon S3 bucket and there is an existing copy of them stored locally.

To upload your local files to your S3 bucket, you can use the AWS CLI *'sync'* command as follows:

```
$ aws s3 sync /path-to-your-local-folder/your_data
s3://bucket-name
```

To download a dataset from Amazon S3, you can use the AWS CLI *'sync'* command as follows:

```
$ aws s3 sync s3://bucket-name /path-to-your-local-folder
```

Install software dependencies

10m

1 Check if the command-line JSON processor jq is installed on your Ubuntu computer:

```
$ jq --version
```

In case it is not installed, please run the following command to install it

```
$ sudo apt-get install jq
```

2 Check if the command-line binary/hexadecimal conversion tool xxd is installed on your Ubuntu

computer:

```
$ xxd --version
```

In case it is not installed, please run the following command to install it

```
$ sudo apt-get install xxd
```

3 Check if the AWS Command Line Interface (CLI) is installed on your Ubuntu computer:

```
$ aws --version
```

In case it is not installed, please run the following command to install it

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

AWS CLI is an open-source tool to interact with AWS services using commands in a command-line shell. AWS CLI also provides direct access to all Amazon S3 API operations.

AWS Authentication

5m

4 Authenticate on Amazon Web Services (AWS) using AWS CLI. Depending on the authentication mode chosen, the AWS command to use may differ.

If you log in on AWS using an IAM role (KEY + SECRET), more info:

```
$ aws configure
```

If you log in on AWS using the AWS Single Sign-On (SSO) service, more info:

```
$ aws configure sso
```

****IMPORTANT:** for the correct function of this tool, you should choose **json** as the preferred output format during the AWS authentication process.

Clone GitHub repositories and tool configuration

10m

5 Clone the s3md5 repo:

```
$ git clone https://github.com/antespi/s3md5.git
```

6 Grant execution access to the s3md5 script file:

```
$ chmod 755 ./s3md5/s3md5
```

7 Clone the aws-s3-integrity-check repo:

```
$ git clone https://github.com/SoniaRuiz/aws-s3-integrity-check.git
```

8 Move the *'s3md5'* folder within the *'aws-s3-integrity-check'* folder:

```
$ mv ./s3md5 ./aws-s3-integrity-check
```

9 The *'aws-s3-integrity-check'* folder should now look similar to the following structure:

```
$ ls -l ./aws-s3-integrity-check
total 48
-rwxr-xr-x 1 your_user your_group 939 May  5 11:08 Dockerfile
-rw-r--r-- 1 your_user your_group 11357 May  5 11:08 LICENSE
-rw-r--r-- 1 your_user your_group 5422 May  5 11:08 README.md
-rw-r--r-- 1 your_user your_group 205 May  5 11:08 aws-s3-integrity-
check.Rproj
-rw-r--r-- 1 your_user your_group 9147 May  5 11:08
aws_check_integrity.sh
drwxr-xr-x 2 your_user your_group 4096 May  5 11:08 logs
drwxr-xr-x 3 your_user  your_group 4096 May  5 11:08 s3md5
```

Bash Tool Execution

10 Run the *'aws-check-integrity.sh'* bash script following the instructions below:

```
$  bash aws_check_integrity.sh [-l|--local <path_local_folder>] [-
b|--bucket <bucket_name>] [-p|--profile <aws_profile>]
```

Usage :

- **[-l|--local <path_local_folder>].** This argument is required. Path to a local folder containing the original version of the files uploaded to Amazon S3. Example: -l /data/nucCyt/raw_data/
- **[-b|--bucket <bucket_name>].** This argument is required. The name of the Amazon S3 bucket containing the files uploaded from the local folder [-l|--local <path_local_folder>]. Example: -b nuccyt
- **[-p|--profile <aws_profile>].** This argument is optional. AWS profile in case the user has authenticated on AWS using the command **aws configure sso**. Example: -p my_aws_profile
- **[-h|--help].** This argument is optional. Shows further help options.

Example #1 (if the user has authenticated on Amazon s3 using SSO):

```
$ bash aws_check_integrity.sh --local /data/nucCyt/ --bucket nuccyt
--profile my_aws_profile
```

Example #2 (if the user has authenticated on Amazon s3 using SSO):

```
$ bash aws_check_integrity.sh -l /data/nucCyt/ -b nuccyt -p
my_aws_profile
```

Example #3 (if the user has authenticated on Amazon s3 using KEY + SECRET):

```
$ bash aws_check_integrity.sh --local /data/nucCyt/ -b nuccyt
```

Example #4:

```
$ bash aws_check_integrity.sh --help
```

General info

11 The *'aws-s3-integrity-check'* tool will:

1. Process and evaluate the arguments [-l|--local], [-b|--bucket] and/or [-p|--profile] received.
2. If the arguments received are correct, this script will test whether the user had read access over the files contained within the Amazon S3 bucket indicated using the parameter: [-b|--bucket <bucket_name>].
3. If the user has correctly authenticated on AWS and has read access over the files stored on the indicated Amazon S3 bucket, this script will interact with Amazon S3 and retrieve the ETag number assigned to the totality of the files contained within the bucket. This query will return a JSON metadata object.
4. Next, in case the local path indicated through the parameter [-l|--local <path_local_folder>] exists, is a directory, and the user has read access to it, this tool will loop through its files. Per each file, it will check whether its name exists among the entries retrieved on the JSON metadata object, indicated within the *"Key"* field. If that is the case, the local file exists on the indicated remote Amazon S3 bucket.
5. If the file exists on the S3 bucket, this tool will evaluate the size of the local file prior to calculating its checksum value. If the data content of the file is smaller than 8 Megabytes (MB), it calculates its Content-MD5 value by using the function md5sum. If the file is larger than 8 MB, the *aws-s3-integrity-check* tool will automatically call the function *s3md5* (version 1.2, https://github.com/antespi/s3md5) using the command *"s3md5 8 local_file_path"*.
6. The *aws-s3-integrity-check'* tool will then filter the metadata JSON object by the fields "ETag" and "Key" using the function *select* (jq JSON processor, version jq-1.5-1-a5b5cbe, https://stedolan.github.io/jq/) and will compare the local and remote checksum values for the local file. In case both numbers are identical, the integrity of the local file will be proven.
7. This process will automatically be repeated for each of the files contained within the local folder [-l|--local <local_folder_path>].

Finally, the *aws-s3-integrity-check* tool will inform the user about the outcome of each aforementioned step by using on-screen messages and log files in a .txt format. The name of the log files will follow the following pattern: *bucketname_S3_integrity_log.timestamp.txt*. The log file

will be stored within a folder named 'logs'. In case this folder doesn't exist, it will be automatically created by the script in the same path in which the aws-s3-integrity-check.sh has been executed.

Docker

12 To run the Dockerised version of this tool, please follow the substeps indicated below.

12.1 Check if the Docker engine is installed on your Ubuntu machine:

```
$ docker --version
```

In case it is not installed, please install it on your Ubuntu machine. This step requires multiple commands, so please check the latest guidance here.

12.2 Download the Docker image:

```
$ docker pull soniaruiz/aws-s3-integrity-check
```

12.3 The Docker image will require the user to indicate the following volumes as parameters:

- **[-v <local_folder>:<local_folder>].** Required argument. Please, replace the strings [<local_folder>:<local_folder>] with the **absolute** path to a local folder containing the files to be tested. This argument is used to mount the local folder as a local volume to the Docker image to allow the Docker image to have read access to the files to test. **Important**: reference the local folder only using the absolute path to the folder. Example: *-v /data/nucCyt:/data/nucCyt*
- **[-v "$PWD/logs/:/usr/src/logs"].** Required argument. This argument should not be changed and sent to the Docker image as it is shown. It represents the path to the local log folder. This argument is used to mount the local *logs* folder as a local volume to the Docker image. It allows the Docker image to record the outputs produced during the tool execution within the local “logs/” folder.
- **[-v "$HOME/.aws:/root/.aws:ro"]**. Required argument. This argument should not be changed and, therefore, it should be sent to the Docker image as it is shown. It represents the path to the folder containing the information about the user authentication on Amazon S3. This parameter is used to mount the local AWS credential directory as a read-only volume to the Docker image. It allows the Docker image to have access to the authentication information of the user on AWS.

****IMPORTANT:** volumes (-v) need to be specified before the name of the Docker image.

12.4 The Docker image will require the user to indicate the following arguments:

- **[-l|--local <local_folder>].**Required argument. Path to a local folder containing the original version of the files uploaded to Amazon S3. Example: -l /data/nucCyt/raw_data/
- **[-b|--bucket <bucket_name>].** Required argument. Name of the Amazon S3 bucket containing the files uploaded from the local folder '-l <local_folder>'. Example: -b nuccyt
- **[-p|--profile <aws_profile>].** Optional argument. AWS profile in case the user has logged in using the command*aws configure sso*. Example: -p my_aws_profile
- **[-h|--help].** Optional argument. Shows further help options.

12.5 **Usage.** Considering the local folder *"/data/nucCyt/"*containing a dataset uploaded to the Amazon S3 bucket *"nuccyt"*:

Example #1 (if the user has authenticated on Amazon s3 using SSO):

```
$ docker run -v /data/nucCyt:/data/nucCyt -v
"$PWD/logs:/usr/src/logs" -v "$HOME/.aws:/root/.aws:ro"
soniaruiz/aws-s3-integrity-check:latest -l /data/nucCyt/ -b
nuccyt -p my_aws_profile
```

Example #2 (if the user has authenticated on Amazon s3 using an IAM role (KEY + SECRET)):

```
$ docker run -v /data/nucCyt:/data/nucCyt -v
"$PWD/logs:/usr/src/logs" -v "$HOME/.aws:/root/.aws:ro"
soniaruiz/aws-s3-integrity-check:latest -l /data/nucCyt/ -b
nuccyt
```