A cheat sheet for the AWS S3 CLI

What is AWS S3?

A flexible, efficient, online cloud services solution is called Amazon Simple Storage Service (Amazon S3). The service is established to record and restore information and applications digitally, to be used with Amazon Web Services (AWS).

Amazon S3 has been designed with limited core functionality for web-scale computation for programmers.

What is AWS CLI?

The abbreviation for CLI is Client Console Session. From a CLI, you can control and keep track of all of your AWS services by utilizing the AWS Command Line Interface (AWS CLI), which is a single tool that controls everything.

In addition to the AWS Management Console and API, a third method for managing the bulk of AWS services is also known as the Command Line Interface, and it can be helpful to the clients.

AWS made it feasible for users of macOS, Linux, and Windows may now manage the core AWS services via a local terminal connection and the command-line interface.

As a result, with only one installation and minimum preparation, you may start utilizing all of the features provided by the AWS Management Console via the terminal program.

Top AWS S3 CLI Commands

Here we’re discussing some of the AWS S3 CLI Commands.

1. Create a New S3 Bucket

Make use of the mb option. Make Bucket is abbreviated as mb.

A new S3 bucket will be created by the following:

```
$ aws s3 mb s3://acit-linux-training2023
```

user’s configuration file, which is seen below:

```
$ cat ~/.aws/config
```

2. Delete an S3 Bucket

Make use of the rb option. Remove bucket is referred to as rb.

The provided bucket is deleted by the next command:

```
$ aws s3 rb s3://tgsbucket
```

Use the -force option as illustrated below to remove a bucket and all of its items:

```
$ aws s3 rb s3://tgsbucket --force
```

3. List all Objects in a Bucket Recursively

Use the following command to show all objects, along with the data of nested folders, in recursive mode:

```
$ aws s3 ls s3://tgsbucket --recursive
```

4. List All S3 Buckets

Use the following ls command to see every bucket that the administrator owns:

```
$ aws s3 ls
```

The bucket was formed on the date indicated by the timestamp in the output above. The time zone will change to match the time zone on your computer.

The command below is identical to the one above:

```
aws s3 ls s3://
```

In the output from the previous statement:

- Recursive choice ensures that, it shows every file in the s3 bucket, along with any subfolders.
- The file size is shown in a format that is readable by humans. The size's possible values may be found in the second column as follows: Bytes/MiB/KiB/GiB/TiB/PiB/EiB.
- Compare alternatives to ensure that the last two lines of the output above are shown. This shows the total number of objects in the S3 bucket as well as their combined size.

6. Copy Local File to S3 Bucket

In the example below, we are moving the getdata.php file from a local laptop to an S3 bucket:

```
$ aws s3 cp getdata.php s3://tgsbucket
```

Follow these steps to move the getdata.php file to an S3 bucket with a different label:

```
$ aws s3 cp getdata.php s3://tgsbucket/getdata-new.php
```

Additionally, as seen below, you may enter the complete path for the local file:

```
$ aws s3 cp /home/project/getdata.php s3://tgsbucket
```

7. Download a File from S3 Bucket

Follow the instructions below to retrieve a specific file from an S3 bucket. The next command transfers getdata.php from the specified S3 bucket to the current directory:

```
$ aws s3 cp s3://tgsbucket/getdata.php .
```

As seen below, you can download the file to your local computer with various names:

```
$ aws s3 cp s3://tgsbucket/getdata.php getdata-local.php
```

As seen below, you can download the file to your local computer with a different name:

```
$ aws s3 cp s3://tgsbucket/getdata.php /home/project/
```

Download the file from the S3 bucket to the designated local computer folder as seen below. This will download the getdata.php file to the local machine’s /home/project folder:

```
$ aws s3 cp s3://tgsbucket/getdata.php /home/project/
```

Move a File from Local to S3 Bucket

As you may anticipate, when you transfer a file from a local system to an S3 bucket, the file is really transported from the local machine to the S3 bucket:

```
$ ls -l source.json

$ aws s3 mv source.json s3://tgsbucket
```

Move All Files from a Local Folder to S3 Bucket

The following files are in the subdirectory in this example:

```
$ ls -1 data
```

The following transfers all the files in the local machine’s data directory to tgsbucket:

```
$ aws s3 mv data s3://tgsbucket/data --recursive
```

AWS S3api commands

Creating a Bucket:

```
aws s3api create-bucket --bucket my-bucket-name --region us-east-1
```

This command creates a new S3 bucket named "my-bucket-name" in the "us-east-1" region.

Uploading an Object to a Bucket:

```
aws s3api put-object --bucket my-bucket-name --key my-object-key --body my-file.txt
```

This command uploads a local file named "my-file.txt" to the S3 bucket "my-bucket-name" with the object key "my-object-key".

Downloading an Object from a Bucket:

```
aws s3api get-object --bucket my-bucket-name --key my-object-key my-download.txt
```

This command downloads an object with the key "my-object-key" from the S3 bucket "my-bucket-name" and saves it as "my-download.txt" locally.

Copying an Object within a Bucket:

```
aws s3api copy-object --bucket my-bucket-name --copy-source my-bucket-name/my-source-object-key --key my-destination-object-key
```

This command copies an object with the source key "my-source-object-key" within the S3 bucket "my-bucket-name" to a new object with the key "my-destination-object-key".

Deleting an Object from a Bucket:

```
aws s3api delete-object --bucket my-bucket-name --key my-object-key
```

This command deletes an object with the key "my-object-key" from the S3 bucket "my-bucket-name".

Listing Objects in a Bucket:

```
aws s3api list-objects --bucket my-bucket-name
```

This command lists all objects in the S3 bucket "my-bucket-name".

Setting Object ACL (Access Control List):

```
aws s3api put-object-acl --bucket my-bucket-name --key my-object-key --acl public-read
```

This command sets the ACL for an object with the key "my-object-key" in the S3 bucket "my-bucket-name" to "public-read", making it publicly readable.