S3 CLI - Introduction

Here is a video about simple AWS CLI commands to work with S3. With this video, you may also refresh your basic information about the AWS CLI that we experienced in the IAM section. Then we will examine the basic S3 CLI commands by applying them in the next lessons.

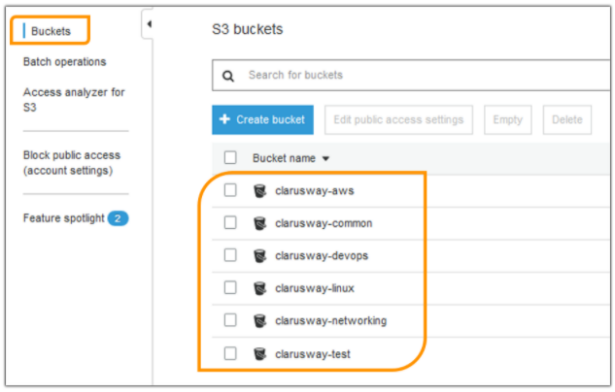
https://youtu.be/qotVe8C62NQ?list=PLQP5dDPLts67DnDIb2IvXd6qPFbLabDCO

List Buckets & Objects

To view, all the user-owned buckets or objects in those buckets, **ls** command can be executed. Now we will practice how to list buckets and objects step by step.

Listing Buckets:

- First, let's look at the S3 Management Console.



Now let's look at how we can see these buckets using CLI.

- As we learned before, we will begin with `aws` command and then specify the AWS service as `s3` .
- Finally, use `ls` command to list all the buckets in S3.

```
$ aws s3 ls
```

- You can also use this comment for the same desired output.

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

- Executing one of these commands will return an output like below.

```
$ aws s3 ls
2020-02-15 22:45:13 clarusway-aws
2020-02-15 22:46:49 clarusway-common
2020-02-15 22:45:34 clarusway-devops
2020-02-15 22:46:20 clarusway-linux
2020-02-15 22:45:54 clarusway-networking
2020-02-15 22:47:16 clarusway-test
```

- As you can see, the output lists all of the six buckets that we have seen on S3 Management console.
- The timestamp in the output is the date the bucket was created.
- The timezone has also been adjusted to show the timezone of your computer.

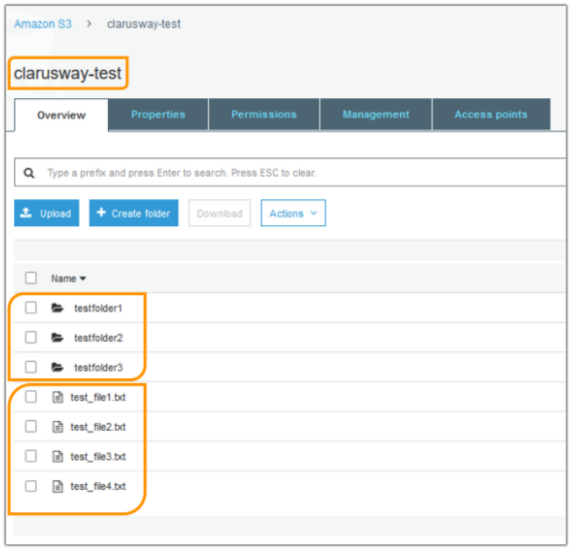
Listing Bucket Objects:

The following command can be used to list contents inside the bucket.

```
$ aws s3 ls s3://bucket-name

```

- Again, let's examine the bucket that we want to list its content through the Management Console.



- As you can see in the image above, there are 3 folders and 4 objects in the bucket named **clarusway-test**.
- Now, let's list them by using CLI commands.

```
$ aws s3 ls s3://clarusway-test

```

- Executing this command will return an output like below.

```
$ aws s3 ls s3://clarusway-test
                           PRE testfolder1/
                           PRE testfolder2/
                           PRE testfolder3/
2020-02-16 01:46:32          0 test_file1.txt
2020-02-16 01:46:32          0 test_file2.txt
2020-02-16 01:46:32          0 test_file3.txt
2020-02-16 01:46:32          0 test_file4.txt

```

In the output, there are three folders indicated by **PRE** inside the **clarusway-test** bucket: **testfolder1, testfolder2, testfolder3**.

- **PRE** stands for the Prefix of an S3 object.
- The output doesn't display the content of these folders.

There are also 4 files at the **/** level: **test_file1.txt, test_file2.txt, test_file3.txt, test_file4.txt.**

- The timestamp is the date the object was created.
- The second column displays the size of those S3 objects. They are all zero here because these .txt files are empty.

Recursive & Summarize Options

Listing Bucket Objects Recursively:

- As mentioned before, `aws s3 ls s3://bucket-name` command does not display the content of the folders inside the bucket.
- The `--recursive` option can be used to see all the contents of a bucket recursively.

```
$ aws s3 ls s3://clarusway-test --recursive
2020-02-16 01:46:32          0 test_file1.txt
2020-02-16 01:46:32          0 test_file2.txt
2020-02-16 01:46:32          0 test_file3.txt
2020-02-16 01:46:32          0 test_file4.txt
2020-02-15 23:31:37          0 testfolder1/
2020-02-16 02:42:16         55 testfolder1/test.html
2020-02-15 23:31:49          0 testfolder2/
2020-02-16 02:42:35         12 testfolder2/content.txt
2020-02-15 23:32:01          0 testfolder3/
2020-02-16 02:42:50       3667 testfolder3/script.py

```

- As you can see, `--recursive` option displays all the files in the s3 bucket including sub-folders.

Summarize:

- The `--summarize` option indicates the total number of objects in the S3 bucket and the total size of all those objects.
- The output displays summarize information in the last two lines.

```
$ aws s3 ls s3://clarusway-test --recursive --summarize
2020-02-16 01:46:32          0 test_file1.txt
2020-02-16 01:46:32          0 test_file2.txt
2020-02-16 01:46:32          0 test_file3.txt
2020-02-16 01:46:32          0 test_file4.txt
2020-02-15 23:31:37          0 testfolder1/
2020-02-16 02:42:16         55 testfolder1/test.html
2020-02-15 23:31:49          0 testfolder2/
2020-02-16 02:42:35         12 testfolder2/content.txt
2020-02-15 23:32:01          0 testfolder3/
2020-02-16 02:42:50       3667 testfolder3/script.py

Total Objects: 10
   Total Size: 3734

```

- The clarusway-test bucket has 10 objects totally.
- The total size of those 10 objects is 3734 bytes.

Create S3 Bucket

- The `mb` that stands for **make bucket** is used to create a new bucket.
- The whole command for creating a new bucket is like below:

```
$ aws s3 mb s3://bucket-name
```

- First, let's list the current buckets in S3.

```
$ aws s3 ls
2020-02-15 22:45:13 clarusway-aws
2020-02-15 22:46:49 clarusway-common
2020-02-15 22:45:34 clarusway-devops
2020-02-15 22:46:20 clarusway-linux
2020-02-15 22:45:54 clarusway-networking
2020-02-15 22:47:16 clarusway-test

```

- Then, let's create a new bucket named **test** via using CLI command like below.

```
$ aws s3 mb s3://test
 make_bucket failed: s3://test An error occurred (BucketAlreadyExists) when calling
      the CreateBucket operation: The requested bucket name is not available.
The bucket namespace is shared by all users of the system. Please select a different
    name and try again.

```

- Because bucket names must be unique in AWS, an error message is returned for bucket name.
- So, let's try **cli-test-clarusway** as a different and unique bucket name.

```
$ aws s3 mb s3://cli-test-clarusway
make_bucket: cli-test-clarusway
```

- As you can see, we have created a new bucket easily via using CLI commands.
- But, we did not specify a region for the bucket. So, **in which region is the new bucket created?**
- If a region name is not specified in the CLI command while creating a new bucket, AWS creates this bucket in the region where you had set in your CLI configuration.
- To create a bucket in a different region than the one from your CLI configuration, the `--region` option comes in the picture.
- Let's create a new bucket with this option.

```
aws s3 mb s3://cli-test-clarusway2 --region us-east-2
 make_bucket: cli-test-clarusway2

```

- Now, let's list the buckets in S3 to check newly created two buckets.

```
$ aws s3 ls
2020-02-15 22:45:13 clarusway-aws
2020-02-15 22:46:49 clarusway-common
2020-02-15 22:45:34 clarusway-devops
2020-02-15 22:46:20 clarusway-linux
2020-02-15 22:45:54 clarusway-networking
2020-02-15 22:47:16 clarusway-test
2020-02-17 00:38:49 cli-test-clarusway
2020-02-17 00:53:45 cli-test-clarusway2

```

- Buckets named **cli-test-clarusway** and **cli-test-clarusway2** are succesfully

Delete S3 Bucket

- The `rb` that stands for **remove bucket** is used to delete an S3 bucket.
- The whole command for deleting a bucket is like below:

```
$ aws s3 rb s3://bucket-name
```

- First, let's list the current buckets in S3.

```
$ aws s3 ls
2020-02-15 22:45:13 clarusway-aws
2020-02-15 22:46:49 clarusway-common
2020-02-15 22:45:34 clarusway-devops
2020-02-15 22:46:20 clarusway-linux
2020-02-15 22:45:54 clarusway-networking
2020-02-15 22:47:16 clarusway-test
2020-02-17 00:38:49 cli-test-clarusway
2020-02-17 00:53:45 cli-test-clarusway2

```

- Then, let's delete **cli-test-calrusway2** bucket via using CLI command like below.

```
$ aws s3 rb s3://cli-test-clarusway2
remove_bucket: cli-test-clarusway2

```

- Now, let's list the buckets in S3 again to check the deletion process.

```
$ aws s3 ls
2020-02-15 22:45:13 clarusway-aws
2020-02-15 22:46:49 clarusway-common
2020-02-15 22:45:34 clarusway-devops
2020-02-15 22:46:20 clarusway-linux
2020-02-15 22:45:54 clarusway-networking
2020-02-15 22:47:16 clarusway-test
2020-02-17 00:38:49 cli-test-clarusway

```

- The **cli-test-clarusway2** bucket is deleted successfully.
- Now, let's delete **clarusway-test** bucket which is not empty.
- First, list the objects in this bucket.

```
$ aws s3 ls s3://clarusway-test
                           PRE testfolder1/
                           PRE testfolder2/
                           PRE testfolder3/
2020-02-16 01:46:32          0 test_file1.txt
2020-02-16 01:46:32          0 test_file2.txt
2020-02-16 01:46:32          0 test_file3.txt
2020-02-16 01:46:32          0 test_file4.txt

```

- There are three folders and four objects in the bucket.
- Then, type the below command to delete this bucket.

```
$ aws s3 rb s3://clarusway-test
remove_bucket failed: s3://clarusway-test An error occurred (BucketNotEmpty) when
    calling the DeleteBucket operation: The bucket you tried to delete is not empty

```

- Because the bucket is not empty and contains some object, the above error message, is returned.
- To delete a bucket along with all its objects, we must use the `--force` option.

```
$ aws s3 rb s3://bucket-name --force
```

- Let's try again to delete the same bucket with `--force` option this time.

```
$ aws s3 rb s3://clarusway-test --force
delete: s3://clarusway-test/test_file1.txt
delete: s3://clarusway-test/test_file2.txt
delete: s3://clarusway-test/testfolder1/test.html
delete: s3://clarusway-test/testfolder1/
delete: s3://clarusway-test/test_file4.txt
delete: s3://clarusway-test/testfolder3/
delete: s3://clarusway-test/testfolder3/script.py
delete: s3://clarusway-test/test_file3.txt
delete: s3://clarusway-test/testfolder2/
delete: s3://clarusway-test/testfolder2/content.txt
remove_bucket: clarusway-test

```

- By using the `--force` option this time, clarusway-test bucket has been successfully deleted along with all its objects.

Delete Object(s) in S3 Bucket

- To delete a specific file from an S3 bucket, `rm` option can be used as shown below.
- The following command will delete the specified file from the given S3 bucket.

```
$ aws s3 rm s3://bucket-name/file-name
```

- To delete all the objects from an S3 bucket, `--recursive` option can be used after the specified bucket name as shown below.

```
$ aws s3 rm s3://bucket-name --recursive
```

Upload File to S3

- The `cp` that stands for **copy** is used to copy a file via CLI.
- In this example, we will copy a file from the local computer to the S3 bucket.
- The whole command for copying a local file to an S3 bucket is like below:

```
$ aws s3 cp file-name s3://bucket-name/file-name /path/to/directory/new-file-name
```

- First, let's list the content of the folder on the local computer.
- The name of the folder is **clarusway-local**.

```
elon@PC:~$ cd clarusway-local/
elon@PC:~/clarusway-local$ ls
content.txt  script.py  test.html  test_file1.txt  test_file2.txt  test_file3.txt
    test_file4.txt  test_file5.txt

```

- Now, let's copy the script.py file from **clarusway-local folder** on the local computer to the **cli-test-clarusway bucket** on S3.

```
$ aws s3 cp script.py s3://cli-test-clarusway
upload: .\script.py to s3://cli-test-clarusway/script.py

```

- To check the copy process, let's list the content of the **cli-test-clarusway bucket**.

```
$ aws s3 ls s3://cli-test-clarusway
2020-02-17 02:12:48       3667 script.py

```

- As you can see, the **script.py file** is successfully copied to the **cli-test-clarusway bucket**.
- If needed, the file can also be renamed while copying as below.

```
$ aws s3 cp file-name s3://bucket-name/new-file-name
```

Upload Folder to S3

- In this example, we will move one step forward and copy a folder from the local computer with all contents to the S3 bucket.
- To copy a folder with all the contents, the below comment is used with the `--recursive` option.

```
$ aws s3 cp folder-name s3://bucket-name/folder-name --recursive
```

- Now, let's copy the **clarusway-local** folder to **cli-test-clarusway** bucket.

```
$aws s3 cp clarusway-local s3://cli-test-clarusway/clarusway-s3 --recursive

upload: clarusway-local\test_file1.txt to s3://cli-test-clarusway/clarusway-s3
    /test_file1.txt
upload: clarusway-local\test_file2.txt to s3://cli-test-clarusway/clarusway-s3
    /test_file2.txt
upload: clarusway-local\test_file4.txt to s3://cli-test-clarusway/clarusway-s3
    /test_file4.txt
upload: clarusway-local\test_file3.txt to s3://cli-test-clarusway/clarusway-s3
    /test_file3.txt
upload: clarusway-local\content.txt to s3://cli-test-clarusway/clarusway-s3/content
    .txt
upload: clarusway-local\script.py to s3://cli-test-clarusway/clarusway-s3/script.py
upload: clarusway-local\test_file5.txt to s3://cli-test-clarusway/clarusway-s3
    /test_file5.txt
upload: clarusway-local\test.html to s3://cli-test-clarusway/clarusway-s3/test.html

```

- Any folder name can be given to the folder in the command.
- By typing the **clarusway-s3** as a folder name after the bucket name, the files in **clarusway-local** are copied to S3 bucket under the **clarusway-s3** folder name.

```
$aws s3 ls s3://cli-test-clarusway
                           PRE clarusway-s3/
```

```
aws s3 ls s3://cli-test-clarusway --recursive
2020-02-17 02:38:09         12 clarusway-s3/content.txt
2020-02-17 02:38:09       3667 clarusway-s3/script.py
2020-02-17 02:38:09         55 clarusway-s3/test.html
2020-02-17 02:38:09          0 clarusway-s3/test_file1.txt
2020-02-17 02:38:09          0 clarusway-s3/test_file2.txt
2020-02-17 02:38:09          0 clarusway-s3/test_file3.txt
2020-02-17 02:38:09          0 clarusway-s3/test_file4.txt
2020-02-17 02:38:09          3 clarusway-s3/test_file5.txt

```

Download File/Folder from S3

Because of the `cp` is used to copy a file via CLI, the same command we used in the uploading lesson will be used to download files or folders from an S3 bucket to a local computer, but with a small difference. The source and destination addresses have to be changed.

Downloading file from S3 bucket:

```
$ aws s3 cp s3://bucket-name/file-name new-file-name
```

- Or, if you want to save the file with the **original name** in the bucket, a **dot** symbol can be used instead of a new-file-name as below.

```
$ aws s3 cp s3://bucket-name/file-name .
```

- Both commands copy the given file in the S3 bucket to the current directory on the local computer.

Downloading folder from S3 bucket:

- The following command will download all the files of the given folder in the given bucket to the **current directory on the local computer**.

```
aws s3 cp s3://bucket-name/folder-name . --recursive
```

- A directory path and a new folder name can also be defined as below.

```
aws s3 cp s3://bucket-name/folder-name \path\to\directory\new-folder-name
    --recursive

```

Move File/Folder

When you move file/folder from local machine to S3 bucket or vise versa, as you would expect, the file will be physically moved from source to destination. Therefore, it will no longer be available in the source.

Command syntax is just as we learned before, but only change the source and destination addresses and use `mv` instead of `cp`.

Moving a file from local to S3:

```
$ aws s3 mv file-name s3://bucket-name
```

Moving a file from S3 to local:

```
$ aws s3 mv s3://bucket-name/file-name /path/to/directory/new-file-name
```

Moving a folder with all the contents from local to S3 :

```
$ aws s3 mv folder-name s3://bucket-name/new/folder-name --recursive
```

Moving a folder with all the contents from S3 to local :

```
$ aws s3 mv s3://bucket-name/folder-name /path/to/directory/new-folder-name
```

Move/Copy Between Buckets

When you move or copy a file/folder from an S3 bucket to another S3 bucket, you can again use the same syntax that we learned before for moving/copying files or folders from local to S3 or vise versa. But this time, the source and destination addresses would be both a bucket.

Copying a file between S3 buckets:

```
$ aws s3 cp s3://source-bucket-name/file-name s3://destination-bucket-name
```

Copying a folder between S3 buckets:

```
$ aws s3 cp s3://source-bucket-name/folder-name s3://destination-bucket-name
```

Moving a file between S3 buckets:

```
$ aws s3 mv s3://source-bucket-name/file-name s3://destination-bucket-name
```

Moving a folder between S3 buckets:

```
$ aws s3 mv s3://source-bucket-name/folder-name s3://destination-bucket-name
```

Copying all objects between S3 buckets:

```
$ aws s3 cp s3://source-bucket-name s3://destination-bucket-name
```

Moving all objects between S3 buckets:

```
$ aws s3 mv s3://source-bucket-name s3://destination-bucket-name
```

S3 Synchronization - Sync

When you use the `sync` command it only copies the new or updated files from the source directory to the destination, recursively.

Syncing local current directory to a bucket:

- The following **`sync`** command syncs objects under a specified bucket to files in a local directory by uploading the local files to s3.
- A local file will require uploading if the size of the local file is different than the size of the s3 object, the last modified time of the local file is newer than the last modified time of the s3 object or the local file does not exist under the specified bucket.

```
$ aws s3 sync . s3://bucket-name
```

Syncing a bucket to local current directory:

- The following **`sync`** command syncs objects under a specified bucket to files in a local directory by downloading s3 objects.
- An s3 object will require downloading if the size of the s3 object differs from the size of the local file, the last modified time of the s3 object is newer than the last modified time of the local file, or the s3 object does not exist in the local directory.
- Take note that when objects are downloaded from s3, the last modified time of the local file is changed to the last modified time of the s3 object.

```
$ aws s3 sync s3://bucket-name .
```

Syncing buckets:

- The following **`sync`** command syncs objects under a specified bucket to objects under another specified bucket by copying s3 objects.
- An s3 object will require copying if the sizes of the two s3 objects differ, the last modified time of the source is newer than the last modified time of the destination, or the s3 object does not exist under the specified bucket destination.

```
$ aws s3 sync s3://source-bucket-name s3://destination-bucket-name
```

Syncing buckets in different regions:

- The following **`sync`** command syncs files between two buckets in different regions:

```
$ aws s3 sync s3://source-bucket-name s3://destination-bucket-name --source-region
    us-east-1 --region us-east-2
```

Excluding files when syncing:

- To exclude any file from syncing, `--exclude` option can be used as shown below:

```
$ aws s3 sync . s3://bucket-name --exclude "file-name"
```

- The above command syncs the current local file and the specified bucket but excludes the specified file from syncing.
- The wildcard characters can also be used to specify the file that will be excluded as shown below.

```
$ aws s3 sync . s3://bucket-name --exclude "*.jpg"
```

- The above command syncs the current local file and the specified bucket but excludes all the files with **.jpeg** extension.

Deleting files when syncing:

- When `--delete` option is used with sync command, any files existing under the source but not existing in the destination will be deleted.
- The following `sync` command syncs objects under a specified prefix and bucket to files in a local directory by uploading the local files to s3.
- Because the `--delete` option is used, any files existing under the specified bucket but not existing in the local directory will be deleted.

```
$ aws s3 sync . s3://bucket-name --delete
```

https://youtu.be/brBs3yCy2vU