COURSEWARE

S3 AWS-CLI Commands

Contents

Overview

We have mainly been looking at working with S3 using the Management Console, however, we can also use the AWS CLI to complete the same tasks, as well as others, in a more programmatic way.
Using the CLI allows us to script and automate what you would normally have to do manually in the management console.

This module will take you through some of the basic commands when working with S3 using the AWS CLI.

Help

The first command you should always be familiar with is the `help` command and how to access it:

```
aws s3 help
```

This will provide you with a list of the commands that you can perform with the CLI. To get more information about these commands you can call the help command on that particular command as below:

```
aws s3 cp help
```

The example above will show more information about the aws `cp` command.

Listing your buckets

We may need to look at a list of the buckets that we have in a particular account.
To do this we can make use of the `ls` command:

```
aws s3 ls
```

This will show a list of all the s3 buckets that you have within your account.

If you read the help for the `ls` command you will see that there are many options that you can pass to the command to get more or different information.

We will likely also want to get some information about the objects within our buckets.
We can use the command below to list the objects that exist within a particular bucket:

```
# aws s3 ls s3://[BUCKET_NAME]
aws s3 ls s3://my-bucket
```

Technically we do not need to use the `s3://[BUCKET_NAME]` schema, as AWS S3 `ls` does not interact with our local file system. However it is good to help avoid ambiguity with what a command is doing.

If we had a bucket with a large number of objects within it, we may want to list the contents of the bucket, along with a summary of the bucket (number of objects and size) in an easy to read way.

```
aws s3 ls s3://[BUCKET_NAME] --summarize --human-readable
aws s3 ls s3://my-bucket --summarize --human-readable
```

Creating a bucket

To create an S3 bucket with the CLI is quite straightforward; we use the `mb` (make bucket) command.
We still need to ensure that we are using a globally unique name for our bucket that meets all the required rules.

```
# aws s3 mb s3://[BUCKET_NAME]
aws s3 mb s3://my-bucket
```

Deleting a bucket

To delete a bucket using the CLI we should first ensure that the bucket is empty, you could do this using the list command discussed above.

You can then use the `rb` (remove bucket) command:

```
# aws s3 rb s3://[BUCKET_NAME]
 aws s3 rb s3://my-bucket
```

If however you have a need to, you can delete a bucket without first emptying it by supplying the `--force` option:

```
# aws s3 rb s3://[BUCKET_NAME] --force
aws s3 rb s3://my-bucket --force
```

Working with Objects

There are a number of commands that can be used to interact with objects stored in your buckets.
These commands fall in line with some commands you may be familiar with from working with Unix based systems, `cp`, `mv` and `rm` in particular.

Copying Files

This command would copy a file from a location on the local machine into the named bucket using the Standard-IA Storage class:

```
# aws s3 cp [FILEPATH] s3://[BUCKET_NAME]
aws s3 cp ./my-file.txt s3://my-bucket
```

The entire contents of a folder can also be copied at once, including its subdirectories by using the recursive option:

```
# aws s3 cp [FILEPATH] s3://[BUCKET_NAME]
aws s3 cp . s3://my-bucket --recursive
```

In many cases a folder that you are copying the entirety of may have files or folders that you do not want to upload.
We can choose which files to *exclude* using the `--exclude` option, passing one file at a time:

```
# aws s3 cp [FILEPATH] s3://[BUCKET_NAME] --recursive
aws s3 cp . s3://my-bucket --recursive --exclude ".git"
```

When excluding files and folders you may also use wildcards; `*.md` for example to ignore all Markdown files.

Moving Files

We could use the `mv` command in a similar way, however remember that the Unix `mv` command will delete the file from the first location and place it at the second.

```
# aws s3 mv s3://[BUCKET_NAME]/[OBJECT_KEY] [LOCAL_FILE_PATH]
aws s3 mv s3://my-bucket/test.txt test.txt
```

We can also use the `cp` and `mv` commands to transfer a file from S3 into our local file system.

Removing Files

```
# aws s3 rm s3://[BUCKET_NAME]/[PATH]/[SUB_DIRECTORY]/[FILE_TO_DELETE]
aws s3 rm s3://my-bucket/static/images/mini-dachshund.png
```

The command above will delete a specific file from a bucket, you can add the `--recursive` option and specify a folder within an S3 bucket to delete a particular folder and all of its contents.

Tutorial

Create a Bucket

First we will create a new bucket for the purposes of this tutorial.
Also for this task create 2 simple text files that will be uploaded to the bucket, `1.txt` and `2.txt` will be used for these examples.

Note that you will not be able to use `my-bucket` as a name, you will need to create a globally unique name.
A suggestion for this could be your forename and a time stamp; for example `[FORENAME]-[TIMESTAMP]` could used on a terminal using `bob-$(date +%s)` resulting in an ouput like this for someone called bob: `bob-1579174995`

```
aws s3 mb s3://[BUCKET_NAME]
```

Upload to the Bucket

We now want to upload our 2 sample files to the bucket:

```
aws s3 cp <filepath> s3://[BUCKET_NAME]
```

Delete the Local Files

Our next task is to delete the files from our local machine, in a future step we want to download the files from the bucket.

```
rm  [FILES]
```

Move a File from the Bucket

We now need to move one of the files from our bucket back into our local file system, to do this we will make use of the `mv` command:

```
aws s3 mv s3://[BUCKET_NAME]/[FILE_TO_COPY] test.txt
```

Remove a File from the Bucket

The next step will be to remove the other file from the bucket, to do this we will make use of the `rm` command:

```
aws s3 rm s3://[BUCKET_NAME]/[PATH]/[SUB_DIRECTORY]/[FILE_TO_DELETE]
```

Remove the Bucket

Our final step now is to completely remove the bucket, to do this will be making use of the `rb` command:

```
aws s3 rb s3://[BUCKET_NAME]
```

Exercises

1. Automate the deployments of a static website hosted in an S3 bucket.
    1. Create a bucket that can serve a static website (this doesn't necessarily need to be done through the CLI)
    2. Using an automation tool of your choice and the AWS S3 CLI, deploy new changes from a static website hosted in a Git repository to the S3 bucket.