Using AWS S3 via the CLI on FreeBSD

Introduction

This document details the basic steps required to save and retrieve files to and from AWS S3, including Glacier and Glacier Deep Archive.

The AWS CLI allows actions on S3 to be scripted, which makes operations on multiple files a lot easier.

Glacier and Glacier Deep Archive are the cheaper forms of long-term storage offered on S3, but they are a little fiddly to use, as restores need to be requested and then acted upon later. Deep Archive is the cheapest, with the longest potential retrieval times.

Packages required

The following packages are required, so install them, or build from ports (if not already installed):

- devel/py-awscli (interface to the AWS command line interface).
- net/py-s3cmd (useful package with additional AWS CLI commands). More details are available at http://s3tools.org.

In addition, some dependencies may be installed.

If a system other than FreeBSD is being used, the packages will have to be located and installed for the system in use. These are generally easily found via Google.

Operations

Note that operations on files in Glacier and Glacier Deep Archive require prior restore of those files, and an additional option to override the warning concerning archived files; see later.

Conventions

The following conventions are used to save space and multiple explanations below:

- *bucket-name*; the name of a bucket on S3. Bucket names must be unique **across the entire S3 namespace**, and should be DNS compliant; thus, it is suggested that a domain name or similar is incorporated into bucket names. In the examples below, the domain name `example.com` is used. Bucket names can contain lower case letters, digits, hyphens and periods. They can start and end only with a letter or a digit, and they cannot contain a period next to a hyphen or another period. They are referred to by a URI starting with `s3://`
- *s3-name*; the name of a file on S3, with a possible prefix (pseudo pathname)
- *local-name*; the name of a file or directory on the local computer, either as source or destination
- *s3-URI*; a file on S3, whether as source or destination. Specified as `s3://`*`bucket-name`*`/`*`s3-name`*

Common options

The `--no-follow-symlinks` option ensures that duplicates are not accidentally uploaded. This is useful with several commands.

The `--quiet` option suppresses informational messages, where appropriate.

The `--recursive` option causes the command to operate on files with sub-prefixes (or subdirectories) as appropriate.

Creating a bucket

Files on S3 are stored in *buckets*; each user is normally allowed 100 buckets. Within buckets, files can have prefixes to their names; these act similarly to directory paths, and they are useful for organisational purposes.

To create a bucket from the command line:

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

Options are unlikely to be required here. For example:

```
$ aws s3 mb s3://mybucket.example.com
```

Listing the contents of a bucket

Use the command:

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

To list the contents of a prefix (‘subdirectory’), append the prefix name to the end of the bucket specification. Include a trailing slash (‘/’) to ensure that the actual contents (not just the prefix) are listed.

The `--recursive` option will list the contents of sub-prefixes as well.

Copying a file to a bucket

Single files can be uploaded using the web interface, or by using the following CLI command.

```
$ aws s3 cp local-name s3://bucket-name options
```

Note that *filename* cannot be a wildcard. Use the `–quiet` option to suppress advisory messages.

Also note that the storage class should usually be specified, as it defaults to `s3`, which is the most expensive:

```
$ aws s3 cp local-name s3://bucket-name --storage-class class
```

Common values for *class* include:

- `STANDARD` (normal S3)
- `GLACIER` (S3 Glacier)
- `DEEP_ARCHIVE` (S3 Deep Archive)

Note that for GLACIER and DEEP_ARCHIVE, retrieval will not be immediate and will involve an additional step.

Copying multiple files to a bucket

One way to copy multiple files is to use the `cp` command with the `–recursive` option.

However, to keep local and S3 files synchronised, use the `sync` CLI command. This example assumes that the current directory, and its subdirectories, are all to be copied. If the files already exist on S3, they will not be uploaded again unless they have changed.

```
        $ aws s3 sync . s3://bucket-name –storage-class class --no-follow-symlinks
```

Note the use of `--no-follow-symlinks`, which is nearly always advisable here.

This command can be used regularly to keep the files on S3 up to date.

Moving files

Files can be moved directly between the local system (local-path below) and S3 (s3-URI below), or vice versa; they can also be moved within S3. Use these commands:

```
        $ aws s3 mv local-name s3-URI
        $ aws s3 mv s3-URI local-name
        $ aws s3 mv s3-URI s3-URI
```

Deleting files

Files can be deleted using this command:

```
        $ aws s3 rm s3-URI
```

Deleting a bucket

An empty bucket can be deleted with:

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

A non-empty bucket can be deleted using the `--force` option, unless there are versioned objects within it; that situation is outwith the scope of this document.

Accessing files in Glacier and Glacier Deep Archive

Files in Glacier and Glacier Deep Archive need to be restored to normal S3 temporarily before they can be accessed. The only difference between the two is the potential time taken for the restoration.

Initial restoration

Different priorities can be given for restoration; higher priorities incur higher costs. The possible priorities are as follows, indicated by the option `--restore-priority`:

- `--restore-priority=bulk`; restoration within 5-12 hours (Glacier) and 48 hours (Deep Archive)
- `--restore-priority=standard` (default); restoration within 3-5 hours (Glacier) and 12 hours (Deep Archive)
- `--restore-priority=expedited`; restoration within about 5 minutes (not available for Deep Archive)

Files are temporarily restored to S3, and they incur charges while they are there. The number of days that they are retained is specified using the option `--restore-days=`*`number`*, where *number* indicates the retention period in days; the default is 1 day.

The actual retrieval command is:

```
        $ s3cmd restore s3-URI [--restore-priority=priority] [--restore-days=days]
```

The `--recursive` option is available for restoration if required.

Restoration is only initiated by this command.

Monitoring restoration status

This is most easily done (for now) via the web interface. A single file can be checked with the command:

```
        $ s3api head-object --bucket bucket-name --key s3-name
and inspecting the output.
```

Downloading retrieved files

Once restored, files are accessed in the same way as if they were normal S3 files (using `cp`, `mv`, `sync`, etc.) However, because they are nominally archived files, the additional option `--force-glacier-transfer` must be given.

Bob Eager

September 2023