Mounting an IBM Cloud Object Storage Bucket to a Linux-based Server using s3fs

James Belton
IBM Cloud Acceleration Architect
July 2019, Updated April 2021

Introduction

If you're a user of IBM Cloud Object Storage, you'll know that there are two main ways that IBM provides to interact with your provisioned service – programmatically through the REST API and visually via the IBM Cloud console.

The IBM Cloud console is the easier of the two for most users, but it has certain limitations, such as a fundamental need for access to the console and a file size of 200MB.

But what how about being able to access an Object Storage bucket directly from a Linux-based server, as though it was an ordinary, 'connected' drive?

Well fortunately, the Cloud Object Storage API also supports the most common set of s3 API operations, which means that many s3 compatible tools can be used to connect to it and this includes s3fs[1], which allows Linux-based machines (and that includes macOS machines) to mount Object Storage buckets like drives.

This guide provides step by step instructions on how to install, configure and use s3fs with Linux to mount IBM Cloud Object Storage buckets to the operating system. I've primarily used centOS, but where appropriate, I've included other Linux flavours, including macOS too.

Having a server directly access Object Storage can be really useful. Object Storage is both low-cost (pay for what you use) and effectively infinite in size. It's great for storing all sorts of files, including things like backups. Using s3fs, you can access files in your object storage with familiar commands and use it in a way that you would any other disk storage, from multiple servers, so it makes a great low-cost 'file share' too. Note though that Object Storage is not as fast as regular storage, so you'll probably notice longer read and write times and of course, you will need to remain connected to the internet.

Requirements

It is assumed that you have an IBM Cloud account and have an instance of Cloud Object Storage provisioned, with one or more buckets. For more information on creating instances of Cloud Object Storage and buckets, refer to the user documentation.
For a deeper overview of IBM Cloud Object Storage, read this guide that I have written.
It is also assumed that you have root level access (or can run commands via 'sudo') on the machine where you are mounting the Object Storage drive.

Want to know more about IBM Cloud? Try my IBM Cloud Foundation Skills Series on YouTube.

[1] See https://github.com/s3fs-fuse/s3fs-fuse for full details

Part One – Create Cloud Object Storage Credentials

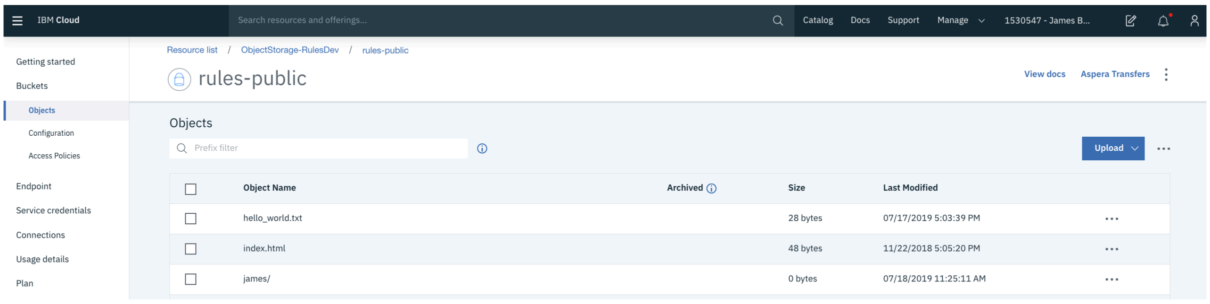
In order to connect to Cloud Object Storage using s3fs from your machine, we need a few details, namely:

- Some credentials
- The bucket’s S3 Enpoint Address

Because this will use S3 API compatible calls, HMAC credentials are required. These are referred to as an Access Key ID and a Secret. Don’t worry, the credentials and address are easily created and / or found.

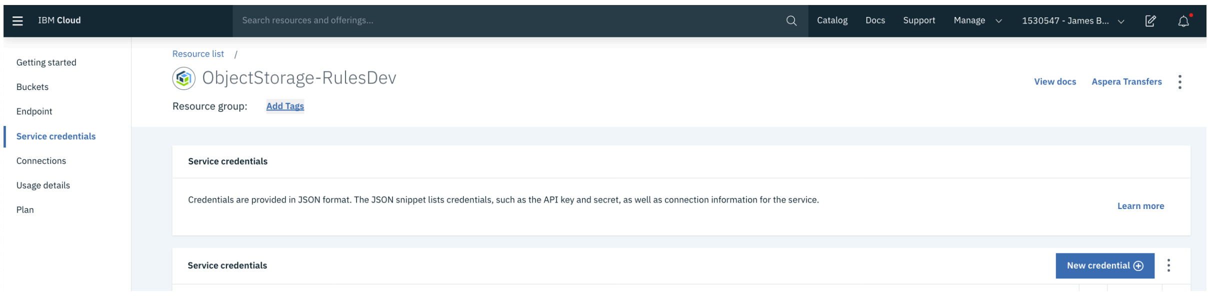
Step One - Create credentials to obtain the Access Key ID and Secret

You will need to create some new credentials to enable s3fs to connect. To do this, ensure that you are logged into your IBM Cloud account and from the Dashboard, you have accessed your Cloud Object Storage instance. Go to the bucket that you want to connect to – in this example, we’re using a bucket called ‘rules-public’. You should see a screen similar to the following:



Object Name	Archived	Size	Last Modified
hello_world.txt		28 bytes	07/17/2019 5:03:39 PM
index.html		48 bytes	11/22/2018 5:05:20 PM
james/		0 bytes	07/18/2019 11:25:11 AM

Click Service Credentials and then click the New Credentials button:



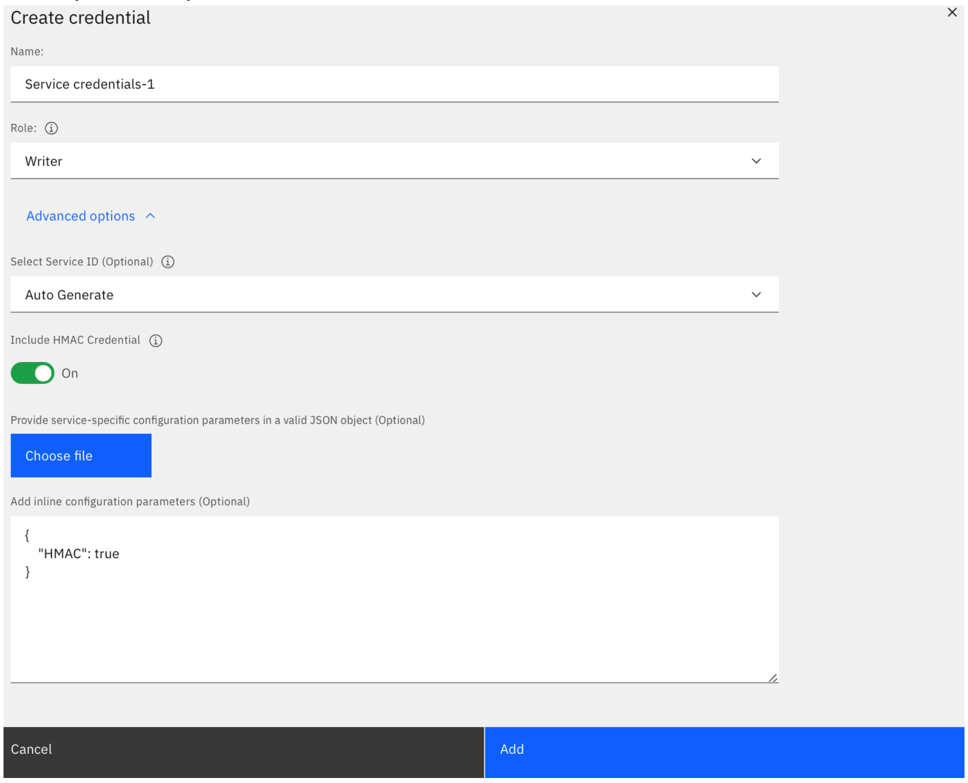
Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

In the panel that appears, provide a name for your new credential. One is provided by default, you can use this or specify your own. Set the role field to Writer.

Click the Advanced Options link.

In the Select Service ID drop down, if you already have a Service ID for your Object Storage, then select this. If not, then select Create New Service ID and type a name for the service in the next field, or just leave set at Auto Generate.

IMPORTANT: you must click / move the 'Include HMAC Credential' slider so that it shows as 'On'. and this will then show `{"HMAC":true}` in the Add Inline Configuration Parameters box. Note that if you don't do this, you will not get the credentials that you need. You should end up with a panel similar to that in the next screen shot.



Create credential

Name: Service credentials-1

Role: Writer

Advanced options

Select Service ID (Optional): Auto Generate

Include HMAC Credential: On

Provide service-specific configuration parameters in a valid JSON object (Optional)

Add inline configuration parameters (Optional)

```
{
  "HMAC": true
}
```

Click Add.

Once the credentials have created (a matter of a couple of seconds), click View Credentials. This will show a JSON formatted list of credential information. Near the top, you will see values similar to the following:

```
"cos_hmac_keys": {
    "access_key_id": "f2XXXc7a4dbaXX9c92XX0a37XXXd09dX",
    "secret_access_key": "c5XXXX8cb6dXXX83d50f7XXXa5fXXXX8fcdXXXXX580XXXX2"
  },
```

The values displayed for `access_key_id` and `secret_access_key` are needed.

Step 2 - Obtain the Object Storage Endpoint Address Value

The next step is to determine the Endpoint Address value that we'll need for s3fs to make a connection to. This is simple to find, just:

a. Click Buckets on the left-hand menu
b. Click on the bucket you want to connect to
c. Click Configuration on the left-hand menu
d. Scroll down to the Endpoints section. You'll notice that three types of Endpoint are shown – Private, Public and Direct. The one you'll use depends on the machine you are mounting the bucket to:

   a. If you are mounting the bucket to a machine **outside** IBM Cloud, use the Public Endpoint
   b. If you are mounting the bucket to a Classic virtual machine or Bare Metal Server that resides inside IBM Cloud, then use the Private Endpoint. You then don't pay for any traffic to and from your bucket and performance is better
   c. If you are mounting the bucket to a VSI in VPC, then use the Direct Endpoint. Again, this gives you better performance and no traffic charges

e. Where there are multiple endpoints – buckets that are cross-region or cross-geo will have multiple endpoints shown - select the one geographically closest as this will typically offer the lowest network latency.

Part Two – Set Up Linux and Connect

In this part, we'll get onto the Linux host, install s3fs and use it to connect to our bucket. Here, we're primarily going to use centOS (they'll work for Red Hat too) but I'll mention commands for other Linux distros too.

First, log onto the centOS host. If you do not log into the host as the 'root' user, then place '`sudo`' before each of the commands to run them as the root user. If you have neither root access, nor access to sudo, then you'll need to talk to your system administrator.

Step 1 – Install s3fs

Once logged in, the first thing you'll need to do is to install EPEL and then s3fs-fuse. To do this simply type, at the command prompt:

```
yum install epel-release
yum install s3fs-fuse
```

Each command should take no more than a couple of minutes to run, though other dependencies may also be needed. Hit 'y' if you are asked to confirm installation.

For Debian and Ubuntu, use the command:

```
apt-get install s3fs
```

NOTE: On Ubuntu instances from IBM Cloud, you may get a 'not found' type error when you try to install s3fs. If so, you need to add to apt-get repositories that are available by default. To do this, run the command:

```
add-apt-repository main
```

For macOS use:

```
brew cask install osxfuse
brew install s3fs
```

Step 2 – Create the password file

The next step is to create a password file which contains the credentials needed to connect to the Object Storage bucket. There are two locations where this information can be stored, either in the user `$HOME` directory as `.passwd-s3fs` to give user only access or in `/etc/password-s3fs`, which gives system-wide access. Here, we're going to use `$HOME/.passwd-s3fs`.

The contents of this file is basically the `access_key_id` and the `secret_access_key_id` that you gathered in Part 1, Step 1 above, in the form:

```
ACCESS_KEY_ID:SECRET_ACCESS_KEY_ID
```

So, taking the example above, the contents of the .passwd-s3fs file will be:

```
f2XXXc7a4dbaXX9c92XX0a37XXXd09dX:c5XXXX8cb6dXXX83d50f7XXXa5fXXXX8fcdXXXXX58
0XXXX2
```

Of course, yours will be different as your key pair will be different.

There are various ways to create this file, but perhaps the easiest way is to type:
```
echo ACCESS_KEY_ID:SECRET_ACCESS_KEY_ID > $HOME/.passwd-s3fs
```

substituting ACCESS_KEY_ID and SECRET_ACCESS_KEY_ID with your values, not forgetting the ':' to separate them.

Next, the file should be protected so that only the user can access the file, so type:

```
chmod 600 $HOME/.passwd-s3fs
```

You are now ready to mount the Object Storage bucket.

Step 3 – Mount the Bucket to the File System

First, you need to create a mountpoint, which is effectively a directory name in the Linux filesystem to which you will connect the Object Storage. In this example, we'll mount it to `/mycosbucket` Simply type:

```
mkdir /mycosbucket
```

Now we can use s3fs to mount the object storage bucket to that mountpoint. The form of the command is:

```
S3fs BUCKET_NAME /mycosbucket -o passwd_file=$HOME.passwd-s3fs
-o url=https://OBJECT-STORAGE-ENDPOINT-ADDRESS
```

The value BUCKET_NAME is the name of the bucket created in IBM Cloud Object Storage. Using the example from Part 1, Step 1 above, the bucket name we'll use here is 'rules-public'.

The value for OBJECT-STORAGE-ENDPOINT-ADDRESS is taken from the instructions in Part1, Step 2 above. This gives us the following command (note yours may be different as you'll probably have a different BUCKET_NAME and possibly a different OBJECT-STORAGE-ENDPOINT-ADDRESS value)

```
s3fs rules-public /mycosbucket -o passwd_file=$HOME/.passwd-s3fs -o
url=https://s3.eu-gb.cloud-object-storage.appdomain.cloud
```

Having run this command, and assuming you have no errors, you should be able to switch to the `/mycosbucket` directory (`cd /mycosbucket`) and list the content (`ls -l`) and generally use the directory as you would any other drive on the machine.

When you disconnect the server (by unmounting the drive or turning off the server) the data persists. You can also mount the bucket to multiple machines, so it makes a great, low cost file share as well.