Making Objects Publicly Available from IBM Cloud Object Storage

James Belton
IBM Cloud Technical Subject Matter Expert
September 2018

Introduction

IBM Cloud Object Storage offers a way to store and serve unstructured data in the cloud. Developers can access objects via APIs, as well as store new objects through applications.

Users also need to make objects such as video clips, presentations or documents available publicly outside of applications – for example as an embedded URL within an email. This can be more difficult from IBM Cloud Object Storage, since it does not support automatic static website hosting (though it is possible to manually configure a web server and use it to serve publicly accessible content hosted in Object Storage. For more information, see this tutorial) but it is possible on an object-by-object basis.

I've created this 'how-to' guide to complement the official IBM Cloud documentation pages and provide a step-by-step approach to providing quick public access to particular objects stored in IBM Cloud Object Storage, via a URL.

This will be done in two ways: the first using the IBM Cloud Object Storage API via Identity and Access Management; the second using a pre-signed URL and HMAC Authentication. In both cases, commands will be issued from the command line.

Note – this will make files available on the public internet, to anyone or process that has its URL. Ensure that any files that you expose do not contain sensitive information that should not be shared with the world.

Requirements

You will need an IBM Cloud account and have an instance of Cloud Object Storage provisioned, with one or more buckets. You will also need at least one object in your bucket, to make publicly available. For more information on creating instances of Cloud Object Storage and buckets, refer to the user documentation.

The first method will involve issuing cURL commands. Therefore, ensure that cURL is installed on your computer and working. OSes such as MacOS and Linux variants normally have this installed. Windows users may need to install it.

If you decide to follow the second method of access, namely using a pre-signed URL and HMAC Authentication, you will also need the AWS CLI installed. This is the official command line interface for AWS and is compatible with the IBM COS S3 API. It is written in Python, so this means a working Python environment is also required. Installing Python is outside of the scope of this document.

Which Method Should I Use?

Each method is simple to implement. The main difference between the two is that with the first method - using IAM - the grant is not temporary; it will last until it is explicitly revoked. This may be useful where the object needs to be exposed for an indefinite period of time. The second method, that uses the AWS CLI, sets a time limit for exposure, defaulting to one hour but it can be explicitly set higher or lower. This way, you don't need to remember to revoke access if access is to be time-limited.

Method One – Using the IBM Cloud Object Storage API via Identity and Access Management

In this method, we shall be using commands issued though cURL to provide public access to an object. This will provide a URL, allowing access to the object via a browser. There are three steps to this:

1. Obtain an API Key (if you don't already have one)
2. Obtain a bearer token
3. Create the public access

As a fourth step, I'll also show the command to remove public access to the object.

Obtain an API Key

API Keys provide a means to identify yourself to IBM Cloud without typing in your username and password. They are particularly useful when scripting where a user identity is required, and they can be revoked and recreated without the need to change the actual user password.
The cURL command that we need to run to obtain a barer token uses an API Key to identify you as a valid user and that you have the necessary access rights to perform the action. Obtaining an API Key is straightforward and can be done in a couple of ways, either through the IBM Cloud UI or via the 'bluemix' CLI. The official documentation provides excellent guidance for each method, though the basic steps using the UI are as follows:

1. Go to **Manage** > **Security** > **IBM Cloud API keys**.
2. Click **Create API key**.
3. Enter a name and description for your API key.
4. Click **Create API key**.
5. Then, click **Show** to display the API key to copy and save it for later, or click **Download API key**. This will download a file called apiKey.json.

Note: For security reasons, the API key is only available to be copied or downloaded at the time of creation. If the API key is lost, you must create a new API key.

You need to keep your API Key as safely as you keep your username and password and if you feel that someone else has access to your API key, you should delete and create a new one.

If you wish, you can set an environment variable hold the value of your API Key. For example:

```
export MYAPIKEY={apiKeyValue}
```

This can also make entering some of the commands in the following sections easier.

Obtain a Bearer Token

The next step is to obtain a bearer token, using your API Key. This token, to give it it's full name is an IAM oauth token and it is used to authenticate the requests made via cURL. Remember, you need your API Key to create the bearer token.

Requesting the token is straight forward. Simply run the following cURL command:

```
curl -X "POST" "https://iam.bluemix.net/oidc/token" \
     -H 'Accept: application/json' \
     -H 'Content-Type: application/x-www-form-urlencoded' \
     --data-urlencode "apikey={api-key}" \
     --data-urlencode "response_type=cloud_iam" \
     --data-urlencode "grant_type=urn:ibm:params:oauth:grant-type:apikey"
```

You will need to substitute {api-key} in the fourth line with your actual API Key. If you set an environment variable to hold the value of your API Key, you can substitute this in, rather than the long value of the actual key.

For example:

```
curl -X "POST" "https://iam.bluemix.net/oidc/token" \
     -H 'Accept: application/json' \
     -H 'Content-Type: application/x-www-form-urlencoded' \
     --data-urlencode "apikey=$MYAPIKEY" \
     --data-urlencode "response_type=cloud_iam" \
     --data-urlencode "grant_type=urn:ibm:params:oauth:grant-type:apikey"
```

When run, this will output several lines of json encoded information. The bit that is required (the access token) is the set of charaters that are in speech marks (") between the words "access token" and "refresh token". Highlight and copy them, remembering to omit the speech marks. There's a lot of it and the string is likely to go over several lines on the screen.

Again, to make it easier to work with, it might be a good idea to create an environment variable to hold the value of the access token (the following examples assume this, though you can also substitute in the full string value):

```
export ACCESS_TOKEN={the copied string}
```

Another thing to note is that the token will expire after 60 minutes. This is a security feature to stop anyone with access to the token having long-term access. If the token expires, another will need to be generated.

Now we have the bearer token, it's time to create the public access to the object.

Enable Public Access to an Object

Public access can either be granted as an object is created (i.e. a new object) or it can also be granted to an object that already exists.

Creating a New Object and Granting Public Access

This is done using the following cURL command:

```
curl -X "PUT" "https://{endpoint}/{bucket-name}/{object-name}" \
     -H "x-amz-acl: public-read" \
     -H "Authorization: Bearer {token}" \
     -H "Content-Type: text/plain; charset=utf-8" \
     -d "{object-contents}"
```

You need to supply the endpoint, bucket-name, object-name, token and object-contents values.

For the endpoint, via the IBM Cloud dashboard, access your Cloud Object Storage instance and click on the bucket-name. The click Configuration in the left-hand menu, scroll down to Endpoints and the value needed it show under 'Public'. For example, this may be '`s3.eu-gb.objectstorage.softlayer.net`'.

The bucket-name is the name of the bucket that to object is to be uploaded to (make sure that the bucket is accessible from the endpoint).

The object-name is the name that you wish to give to the object, while the object-contents is the contents of the object.

An example of a working command is:

```
curl -X "PUT" "https://s3.eu-gb.objectstorage.softlayer.net/rules-public/HelloWorld.txt" \
     -H "x-amz-acl: public-read" \
     -H "Authorization: Bearer $ACCESS_TOKEN" \
     -H "Content-Type: text/plain; charset=utf-8" \
     -d "Hello World, this is a test."
```

So, running the above cURL command will create an object called HelloWorld.txt, in the rules-public bucket. The object can then be accessed by anyone via the URL https://s3.eu-gb.objectstorage.softlayer.net/rules-public/HelloWorld.txt

Note the `x-amz-acl: public-read` part of the command. This is setting an Access Control List (acl) setting, which is set to 'public-read' by this command.

Granting Public Access to an Object that Already Exists

This is done using the following cURL command:

```
curl -X "PUT" "https://{endpoint}/{bucket-name}/{object-name}?acl" \
     -H "x-amz-acl: public-read" \
     -H "Authorization: Bearer {token}" \
     -H "Content-Type: text/plain; charset=utf-8"
```

You need to supply the endpoint, bucket-name, object-name, and token values.

For the endpoint, via the IBM Cloud dashboard, access your Cloud Object Storage instance and click on the bucket-name. The click Configuration in the left-hand menu, scroll down to

Endpoints and the value needed it show under ‘Public’. For example, this may be ‘s3.eu-gb.objectstorage.softlayer.net’.

The bucket-name is the name of the bucket that the object resides in.

The object name is the name of an existing object in the target bucket that you want to make publicly accessible.

An example of a working command is:

```
curl -X "PUT" "https://s3.eu-gb.objectstorage.softlayer.net/rules-
public/HelloWorld.txt?acl" \
     -H "x-amz-acl: public-read" \
     -H "Authorization: Bearer $ACCESS_TOKEN " \
     -H "Content-Type: text/plain; charset=utf-8" \
```

So, the above command will make the contents of the object HelloWorld.txt publicly available via the URL https://s3.eu-gb.objectstorage.softlayer.net/rules-public/HelloWorld.txt

Again, note that the Access Control List for the object is being set to ‘public-read’.

You can make any type of object available this way – try it with a .mov object, for example.

Making an Object Private Again

You may only wish to make an object public for limited period of time. To remove public access to an object, you must run the following cURL command:

```
curl -X "PUT" "https://{endpoint}/{bucket-name}/{object-name}?acl" \
     -H "Authorization: Bearer {token}" \
     -H "x-amz-acl:" \
     -H "Content-Type: text/plain; charset=utf-8"
```

You need to supply the endpoint, bucket-name, object-name, and token values.

For the endpoint, via the IBM Cloud dashboard, access your Cloud Object Storage instance and click on the bucket-name. The click Configuration in the left-hand menu, scroll down to Endpoints and the value needed it show under ‘Public’. For example, this may be ‘s3.eu-gb.objectstorage.softlayer.net’.

The bucket-name is the name of the bucket that to object resides in. The object name is the name of the object that you want to remove public access from.

An example of a working command is:

```
curl -X "PUT" "https://s3.eu-gb.objectstorage.softlayer.net/rules-
public/HelloWorld.txt?acl" \
     -H "x-amz-acl:" \
     -H "Authorization: Bearer $ACCESS_TOKEN " \
     -H "Content-Type: text/plain; charset=utf-8"
```

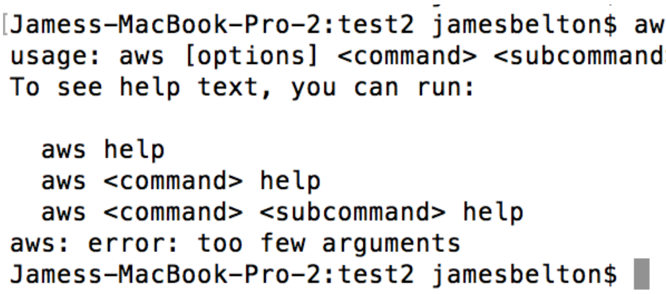
So, the above command will remove public access to the file HelloWorld.txt in the rules-public bucket (i.e. it will no longer be accessible via the URL `https://s3.eu-gb.objectstorage.softlayer.net/rules-public/HelloWorld.txt`.

What's actually happening is that we're removing the acl from the file.

Note that using this method, objects will remain public indefinitely, until either they are removed from the Object Store bucket or the ACL is reset as per the above example.

Method Two – using a pre-signed URL and HMAC Authentication.

The second method that I'll walk through is using a pre-signed URL to provide temporary public access to an object. This method uses the AWS CLI, which you must first download and install on your local device. Once installed, you will be able to open a command window and type 'aws' to check it is working:



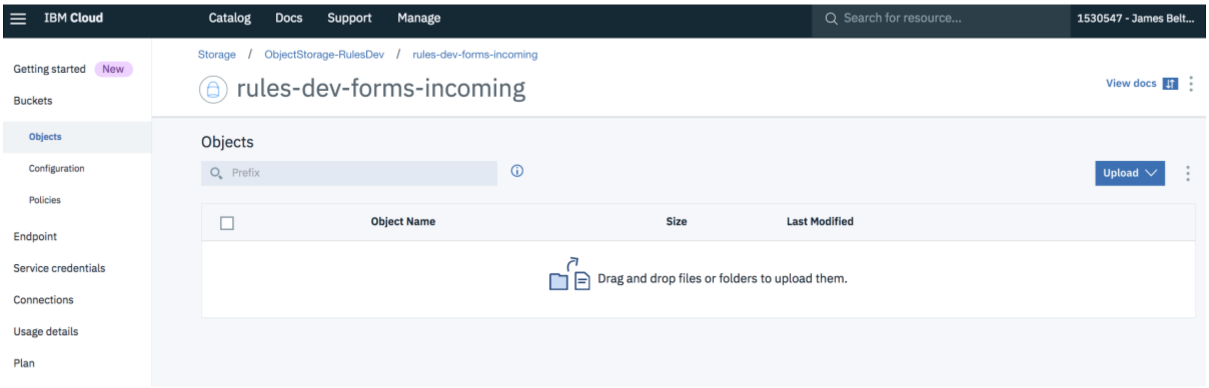
```
Jamess-MacBook-Pro-2:test2 jamesbelton$ aws
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: too few arguments
Jamess-MacBook-Pro-2:test2 jamesbelton$ 
```

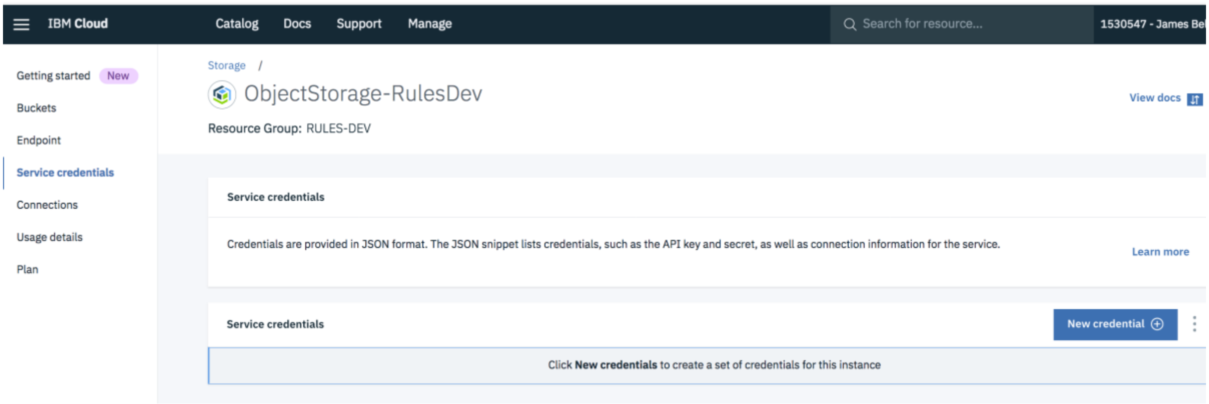
Obtain Your HMAC Credentials

Rather than IAM, accessing Cloud Object Storage via the AWS CLI uses HMAC authentication. This is a pair of ID/Key values, namely an Access Key ID and a Secret Access Key. You need to generate these via the IBM Cloud UI as follows:

Ensure that you are logged into your IBM Cloud account and from the Dashboard, and you have accessed your Cloud Object Storage instance. You should see a screen similar to the following:



Click Service Credentials and then click the New Credentials button:



ObjectStorage-RulesDev

Resource Group: RULES-DEV

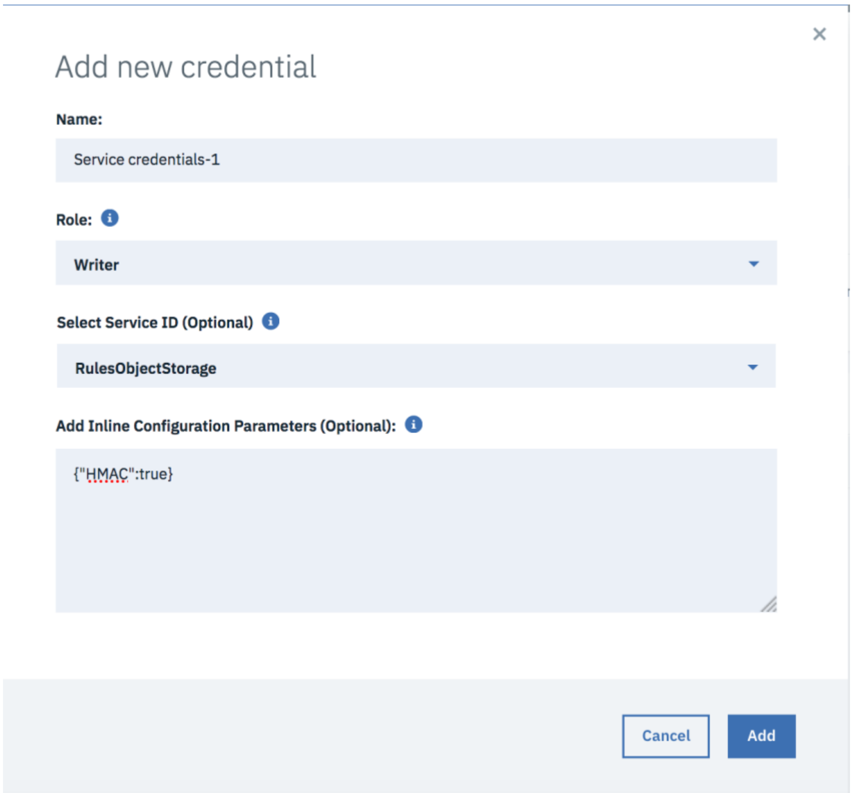
Service credentials

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.

Service credentials

Click **New credentials** to create a set of credentials for this instance

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. 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. In the Add Inline Configuration Parameters box, type: `{“HMAC”:true}` – note that if you leave this out, you will not get the credentials that you need. You should end up with a panel similar to that in the next screen shot.



Add new credential

Name: Service credentials-1

Role: Writer

Select Service ID (Optional): RulesObjectStorage

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 those that are needed.

You also need to make a note of the endpoint value. Via the IBM Cloud dashboard, access your Cloud Object Storage instance and click on the bucket-name where the object exists that you want to make public. The click Configuration in the left-hand menu, scroll down to Endpoints and the value needed it show under 'Public'. For example, this may be '`s3.eu-gb.objectstorage.softlayer.net`'.

Configure the AWS CLI

Next, configure the AWS CLI, to ensure that it connects to the IBM Cloud Object Storage service. To do this, at a command prompt type:

```
aws configure
```

This will start the configuration process. You will be asked for:

- `AWS Access Key ID` – enter the access_key_id value obtained in the previous step
- `AWS Secret Access Key` – enter the secret_access_key value obtained in the previous step
- `Default Region Name` – enter 'us-standard'
- `Default Output Format` – enter 'json'

You should then be put back to a prompt.

Run the Command

The command that needs to be run has the following form:

```
aws --endpoint-url https://{endpoint} \
s3 presign s3://{bucket-name}/{object_name} \
 --expires-in {seconds}
```

The `–expires-in` operator is optional. If you omit it, then the default value of 3600 (one hour) will be used but if you set a value, then access will be removed when the time limit is exceeded.

A working version of the command is as follows:

```
aws --endpoint-url https://s3.eu-gb.objectstorage.softlayer.net s3 \
presign s3://rules-public/test2/IMG_1987.MOV \
--expires-in 600
```

Running this command will then generate a URL which is returned to the screen. The URL contains the location of the object as well as the required authorization parameters. For example:

```
https://s3.eu-gb.objectstorage.softlayer.net/rules-
public/test2/IMG_1987.MOV?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Expires=600&X-Amz-
Credential=f2cXXc7a4dXXX99c92dXXX37764dXXXf%2F2XXX0925%2Fus-
standard%2Fs3%2Faws4_request&X-Amz-SignedHeaders=host&X-Amz-
Date=20180925T110301Z&X-Amz-
Signature=XXX4c72f7547XXXa65XXc8d3XXXX104f8XXXf30XXX436cXXXX8f4a981838XXXX
```

This can be copied and pasted into a browser. In the case of the example above, a movie object called IMG_1987.MOV will be played.

Note that the –expires-in parameter was set to 600 – this means that in 600 seconds (10 minutes) the public access will be rescinded, and the object will no longer be available.

About the Author

James Belton is an IBM Cloud Technical Subject Matter Expert who works with clients and other IBMers, helping them use and consume IBM Cloud services. He has written a number of similar guides, which are published on his Github page and has also written articles that have been published on IBM developerWorks. His personal blog site is The IBM Cloud Digest. James also tweets via @jamesbelton and can be found on LinkedIn.