Red Hat OpenShift Data Science 1

Integrating data from Amazon S3

Use data stored in an Amazon Web Services (AWS) Simple Storage Service (S3) bucket

Last Updated: 2021-12-14

Red Hat OpenShift Data Science 1 Integrating data from Amazon S3

Use data stored in an Amazon Web Services (AWS) Simple Storage Service (S3) bucket

Legal Notice

Abstract

Learn how to use data stored in an Amazon Web Services (AWS) Simple Storage Service (S3) bucket.

Table of Contents

PREFACE

This documentation is provided for the Field Trial release of Red Hat OpenShift Data Science.

See the following documents for service and life cycle information related to this Field Trial release:

- OpenShift Data Science Service Definition
- OpenShift Data Science Life Cycle

When working in a Jupyter Notebook, you may want to work with data stored in an Amazon Web Services (AWS) Simple Storage Service (S3) bucket. This section covers commands and procedures for working with data stored in Amazon S3.

CHAPTER 1. PREREQUISITES

- A JupyterHub server running on Red Hat OpenShift Data Science.
- Access to a Amazon Web Services S3 bucket.
- Locate the **S3 Access Key ID** and **S3 Secret Access Key** for your Amazon S3 account.
- A Jupyter Notebook

CHAPTER 2. CREATING AN AMAZON S3 CLIENT USING NOTEBOOK CELLS

To interact with data in Amazon S3 buckets, you must create a local client to handle requests to that service.

Prerequisites

- Access to a JupyterHub notebook server running on Red Hat OpenShift Data Science.
- Define values for the **AWS_ACCESS_KEY_ID** and **AWS_SECRET_ACCESS_KEY** environment variables when you launch your notebook server, using the values from your Amazon Web Services account under **My Security Credentials**.

Procedure

1. In a new notebook cell, import the required libraries by adding the following:

```
import os
import boto3
from boto3 import session
```

2. In another new notebook cell, define the following to create your session and client.
   a. Define your credentials.

```
key_id = os.environ.get(AWS_ACCESS_KEY_ID)
secret_key = os.environ.get(AWS_SECRET_ACCESS_KEY)
```

   b. Define the client session.

```
session = boto3.session.Session(aws_access_key_id=key_id,
aws_secret_access_key=secret_key)
```

   c. Define the client connection.

```
s3_client = boto3.client(s3, aws_access_key_id=key_id,
aws_secret_access_key=secret_key)
```

Verification

- Create a new cell and run an Amazon S3 command such as the following:

```
s3_client.list_buckets()
```

A successful response includes a **HTTPStatusCode** of **200** and a list of **Buckets** similar to the following:

```
'Buckets': [{'Name': 'my-app-asdf3-image-registry-us-east-1-wbmlcvbasdfasdgvtsmkpt',
  'CreationDate': datetime.datetime(2021, 4, 21, 6, 8, 52, tzinfo=tzlocal())},
 {'Name': 'cf-templates-18rxasdfggawsvb-us-east-1',
  'CreationDate': datetime.datetime(2021, 2, 15, 18, 35, 34, tzinfo=tzlocal())}
```

CHAPTER 3. LISTING AVAILABLE AMAZON S3 BUCKETS USING NOTEBOOK CELLS

You can check which buckets you have access to by listing the buckets available to your account.

Prerequisites

- Configure an Amazon S3 client in a previous cell in the notebook. See Creating an Amazon S3 client using notebook cells for more information.

Procedure

1. Create a new notebook cell and use the **s3_client** to list available buckets.

```
s3_client.list_buckets()
```

2. You can make this list of buckets easier to read by only printing the name, instead of the full response, for example:

```
for bucket in s3_client.list_buckets()[Buckets]:
    print(bucket[Name])
```

This returns output similar to the following:

```
my-app-asdf3-image-registry-us-east-1-wbmlcvbasdgasdgtkpt
cf-templates-18rxuasgasgvb-us-east-1
```

Additional resources

- Creating an Amazon S3 client using notebook cells
- Amazon Web Services list buckets command reference

CHAPTER 4. LISTING FILES IN AVAILABLE AMAZON S3 BUCKETS USING NOTEBOOK CELLS

You can check the files available in buckets you have access to by listing the objects in the bucket. Because buckets use object storage instead of a typical file system, object naming works differently from normal file naming. Objects in a bucket are always known by a key, which consists of the full path in the bucket plus the name of the file itself.

Prerequisites

- Configure an Amazon S3 client in a previous cell in the notebook. See Creating an Amazon S3 client using notebook cells for more information.

Procedure

1. Create a new notebook cell and list the objects in the bucket. For example:

```
bucket_name = std-user-bucket1
s3_client.list_objects_v2(Bucket=bucket_name)
```

   This returns a number of objects in the following format:

```
{Key:
docker/registry/v2/blobs/sha256/00/0080913dd3f10aadb34asfgsgsdgasdga072049c93606b98b
ec84adb259b424f/data,
LastModified: datetime.datetime(2021, 4, 22, 1, 26, 1, tzinfo=tzlocal()),
ETag: "6e02fad2deassadfsf900a4bd7344ffe",
Size: 4052,
StorageClass: STANDARD}
```

2. You can make this list easier to read by only printing the key instead of the full response, for example:

```
bucket_name = std-user-bucket1
for key in s3_client.list_objects_v2(Bucket=bucket_name)[Contents]:
    print(key[Key])
```

   This returns output similar to the following:

```
docker/registry/v2/blobs/sha256/00/0080913dd3f10aadb34asfgsgsdgasdga072049c93606b98b
ec84adb259b424f/data
```

3. You can also filter your query to list for a specific "path" or file name, for example:

```
bucket_name = std-user-bucket1
for key in s3_client.list_objects_v2(Bucket=bucket_name,Prefix=start_of_file_path)
[Contents]:
    print(key[Key])
```

Additional resources

- Creating an Amazon S3 client using notebook cells

- Amazon Web Services list objects command reference

CHAPTER 5. DOWNLOADING FILES FROM AVAILABLE AMAZON S3 BUCKETS USING NOTEBOOK CELLS

You can download a file to your notebook server using the **download_file** method.

Prerequisites

- Configure an Amazon S3 client in a previous cell in the notebook. See Creating an Amazon S3 client using notebook cells for more information.

Procedure

1. Define the following in a notebook cell:
   a. The bucket that the file is in.

```
bucket_name = name_of_the_bucket
```

   b. The name of the file to download.

```
file_name = name_of_the_file_to_download # Full path from the bucket
```

   c. The name that you want the file to have after it is downloaded. This can be a full path, a relative path, or just a new file name.

```
new_file_name = name_of_the_file_once_downloaded
```

2. Download the file, specifying the previous variables as arguments.

```
s3_client.download_file(bucket_name, file_name, new_file_name)
```

NOTE

If you want to retrieve a file as an object that you can then stream as a standard file using the read() method, refer to the Amazon Sev Services get object command reference

Additional resources

- Creating an Amazon S3 client using notebook cells
- Amazon Web Services download file command reference

CHAPTER 6. UPLOADING FILES TO AVAILABLE AMAZON S3 BUCKETS USING NOTEBOOK CELLS

You can upload files from your notebook server to an Amazon S3 bucket using the **upload_file** method.

Prerequisites

- Configure an Amazon S3 client in a previous cell in the notebook. See Creating an Amazon S3 client using notebook cellsCreating an Amazon S3 client using notebook cells for more information.

Procedure

1. Define the following in a notebook cell:
   a. The name of the file to upload. This must include the full local path to the file.

```
file_name = name_of_the_file_to_upload
```

   b. The name of the bucket to upload the file to.

```
bucket_name = name_of_the_bucket
```

   c. The full key to use to save the file to the bucket.

```
key = full_path_and_file_name
```

2. Upload the file, specifying the previous variables as arguments.

```
s3_client.upload_file(file_name, bucket_name, key)
```

Additional resources

- Creating an Amazon S3 client using notebook cells
- Amazon Web Services upload file command reference

CHAPTER 7. ADDITIONAL RESOURCES

- Red Hat OpenShift Data Science documentation