Red Hat OpenShift Data Science 1

集成来自 Amazon S3 的数据

使用存储在 Amazon Web Services(AWS)Simple Storage Service(S3)存储桶中的数据

Last Updated: 2023-10-18

Red Hat OpenShift Data Science 1 集成来自 Amazon S3 的数据

使用存储在 Amazon Web Services(AWS)Simple Storage Service(S3)存储桶中的数据

法律通告

摘要

了解如何使用存储在 Amazon Web Services(AWS)Simple Storage Service(S3)存储桶中的数据。

目录

前言

在 Jupyter Notebook 中操作时，您可能希望使用存储在 Amazon Web Services(AWS)Simple Storage Service(S3)存储桶中的数据。本节介绍使用存储在 Amazon S3 中的数据的命令和步骤。

第 1 章 先决条件

- 在 Red Hat OpenShift Data Science 上运行的 Jupyter 服务器。
- 访问 Amazon Web Services S3 存储桶。
- 为 Amazon S3 帐户找到 **AWS Access Key ID** 和 **AWS Secret Access Key**。
- 一个 Jupyter 笔记本。

第 2 章 使用笔记本单元创建 AMAZON S3 客户端

要与 Amazon S3 存储桶中的数据交互，您必须创建一个本地客户端来处理对该服务的请求。

先决条件

- 访问在 Red Hat OpenShift Data Science 上运行的 Jupyter 笔记本服务器。
- 在启动笔记本服务器时使用您的 Amazon Web Services 帐户中的值，使用 **My Security Credentials** 下的 Amazon Web Services 帐户中的值，定义 **AWS_ACCESS_KEY_ID** 和 **AWS_SECRET_ACCESS_KEY** 环境变量的值。

流程

1. 在新的笔记本单元中，添加以下内容来导入所需的库：

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

2. 在另一个新的笔记本中，定义以下内容以创建会话和客户端。
   a. 定义您的凭证。

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

   b. 定义客户端会话。

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

   c. 定义客户端连接。

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

验证

- 创建新的单元，并运行 Amazon S3 命令，如下所示：

```
s3_client.list_buckets()
```

成功的响应包括 **HTTPStatusCode** 为 **200**，以及类似以下的 **Buckets** 列表：

```
'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())}
```

第 3 章 使用笔记本单元列出可用的 AMAZON S3 存储桶

您可以通过列出可用于帐户的存储桶来检查您有权访问的存储桶。

先决条件

- 在笔记本中的单元格中配置 Amazon S3 客户端。如需更多信息，请参阅使用笔记本单元创建 Amazon S3 客户端。

流程

1. 创建新的笔记本单元，并使用 **s3_client** 列出可用存储桶。

```
s3_client.list_buckets()
```

2. 您可通过只打印名称而不是完整响应来更轻松地阅读此存储桶列表，例如：

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

这会返回类似如下的输出：

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

其他资源

- 使用笔记本单元创建 Amazon S3 客户端
- Amazon Web Services list bucket 命令参考

第 4 章 使用笔记本单元列出可用 AMAZON S3 存储桶中的文件

您可以通过列出存储桶中的对象来检查您可以访问的 bucket 中可用的文件。由于 bucket 使用对象存储而非典型的文件系统，因此对象命名的工作方式与普通文件命名不同。bucket 中的对象始终由一个键来代表，它由存储桶中的完整路径以及文件本身的名称组成。

先决条件

- 在笔记本中的单元格中配置 Amazon S3 客户端。如需更多信息，请参阅使用笔记本单元创建 Amazon S3 客户端。

流程

1. 创建新的笔记本单元，并列出存储桶中的对象。例如：

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

这会以以下格式返回几个对象：

```
{'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. 您可以只打印键而不是完整的响应，从而使此列表更易于阅读，例如：

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

这会返回类似如下的输出：

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

3. 您还可以过滤查询来列出特定"path"或文件名，例如：

```
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'])
```

在上例中，将 ***<start_of_file_path>*** 替换为您自己的值。

其他资源

- 使用笔记本单元创建 Amazon S3 客户端
- Amazon Web Services list 对象命令参考

第 5 章 使用笔记本单元从可用的 AMAZON S3 存储桶下载文件

您可以使用 **download_file** 方法将文件下载到笔记本服务器。

先决条件

- 在笔记本中的单元格中配置 Amazon S3 客户端。如需更多信息，请参阅使用笔记本单元创建 Amazon S3 客户端。

流程

1. 在笔记本单元中定义以下详情：
   a. 该文件所在存储桶。将 ***<name_of_the_bucket>*** 替换为您自己的值。

   ```
   bucket_name = '<name_of_the_bucket>'
   ```

   b. 要下载的文件的名称。将 ***<name_of_the_file_to_download>*** 替换为您自己的值。

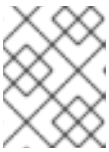
   ```
   file_name = '<name_of_the_file_to_download>' # Full path from the bucket
   ```

   c. 下载文件后需要具有的名称。这可以是完整路径、相对路径或只包括新文件名。将 ***<name_of_the_file_when_downloaded>*** 替换为您自己的值。

   ```
   new_file_name = '<name_of_the_file_when_downloaded>'
   ```

2. 下载文件，并将前面的变量指定为参数。

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



注意

如果要检索文件作为对象，您可以使用 read() 方法作为标准文件流，请参阅 Amazon Web Services get object command reference 。

其他资源

- 使用笔记本单元创建 Amazon S3 客户端
- Amazon Web Services 下载文件命令参考

第 6 章 使用笔记本单元将文件复制到可用的 AMAZON S3 存储桶

您可以使用 **upload_file** 方法将文件从笔记本服务器上传到 Amazon S3 存储桶。

先决条件

- 在笔记本中的单元格中配置 Amazon S3 客户端。如需更多信息，请参阅使用笔记本单元创建 Amazon S3 客户端。

流程

1. 在笔记本单元中定义以下详情：
   a. 要上传的文件的名称。这必须包含文件的完整路径。将 ***<name_of_the_file_to_upload>*** 替换为您自己的值。

   ```
   file_name = '<name_of_the_file_to_upload>'
   ```

   b. 要上传文件的存储桶的名称。将 ***<name_of_the_bucket>*** 替换为您自己的值。

   ```
   bucket_name = '<name_of_the_bucket>'
   ```

   c. 用于将文件保存到存储桶的完整键。将 ***<full_path_and_file_name>*** 替换为您自己的值。

   ```
   key = '<full_path_and_file_name>'
   ```

2. 上传文件，并将前面的变量指定为参数。

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

其他资源

- 使用笔记本单元创建 Amazon S3 客户端
- Amazon Web Services upload file 命令参考

第 7 章 其他资源

- Red Hat OpenShift Data Science 文档