**Standard Operating Procedure (SOP) - Storing and Deleting Excel Files in Amazon S3 using Python**

**Objective**: This procedure outlines how to use Python to upload an Excel file to an Amazon S3 bucket. We will utilize the Boto3 library to interact with S3.

**Prerequisites**:

1. An AWS account with permissions to access and write to the desired S3 bucket.
2. Python installed on your computer.
3. Required Python libraries: **boto3**.

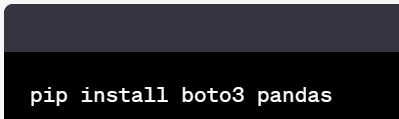
**Steps**:

**1. Set Up AWS Credentials**:

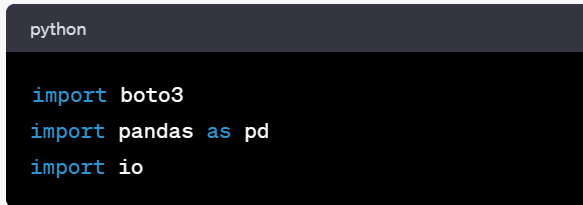
* Ensure you have your AWS access key and secret access key ready. Keep them safe and don't share them publicly.

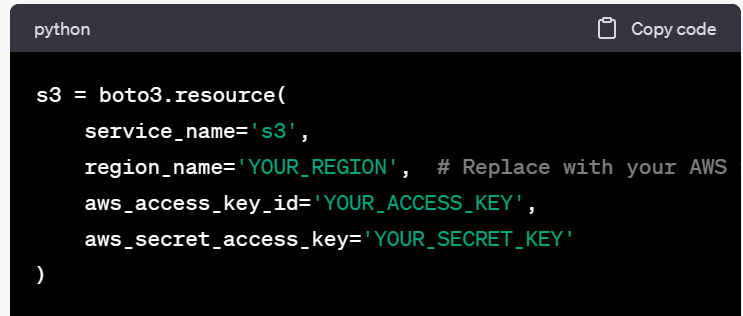
**2. Install Required Python Libraries**:

* Open a terminal or command prompt.
* Install **boto3** and **pandas** by running:



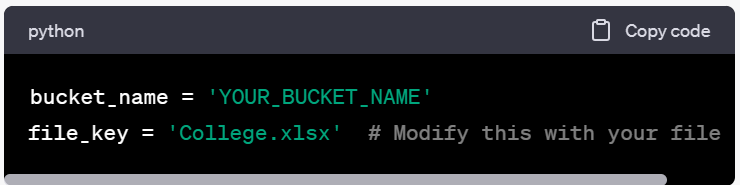
**3. Python Code**:

* Import the necessary libraries: 
* Create an S3 resource object using your AWS credentials:

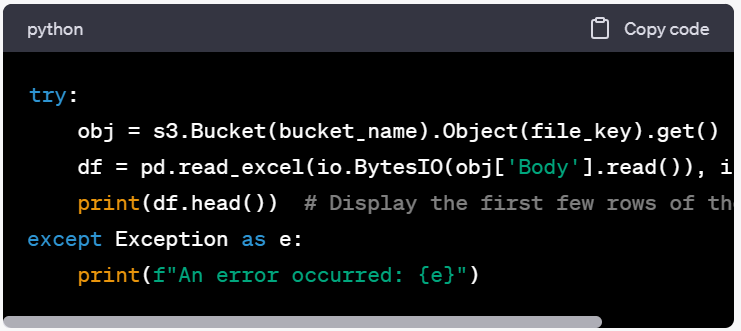


**4. Read Excel File from S3**:

* Specify the S3 bucket name and the path to your Excel file:

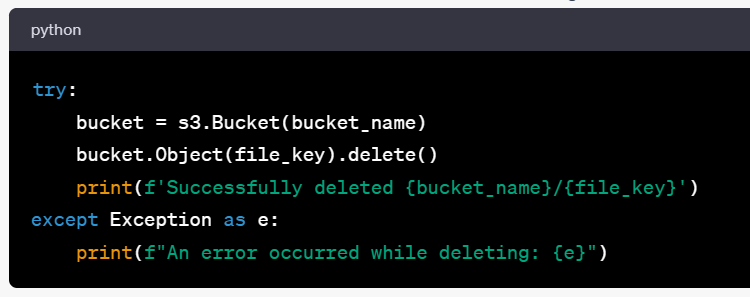


* Retrieve the Excel file from S3 and read it into a Pandas DataFrame:



**5. Delete a File from S3**:

To delete the Excel file from the S3 bucket, use the following code:



**6. Execute the Code**:

* Run your Python script. It will upload the Excel file to the specified S3 bucket and delete it from the bucket when required.

**Best Practices**:

* Keep your AWS access and secret access keys secure and avoid hard-coding them in your script.
* Ensure appropriate permissions are configured for the S3 bucket.
* Implement error handling to handle potential issues during both the upload and delete processes.

**Note**: Replace **'YOUR\_REGION'**, **'YOUR\_ACCESS\_KEY'**, **'YOUR\_SECRET\_KEY'**, **'YOUR\_BUCKET\_NAME'**, and **'College.xlsx'** with your specific AWS settings and file details.