

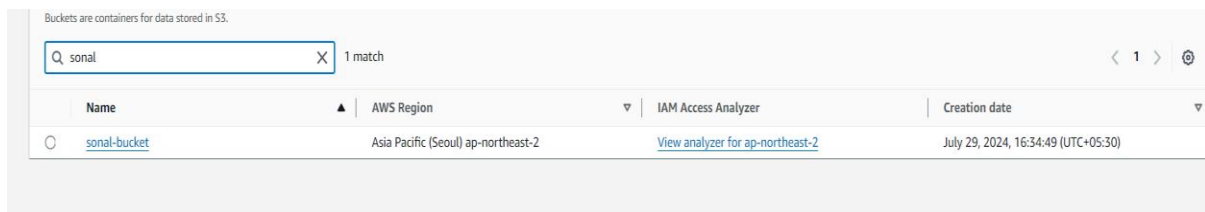
Assignment 2: Automated S3 Bucket Cleanup Using AWS Lambda and Boto3

Objective: To gain experience with AWS Lambda and Boto3 by creating a Lambda function that will automatically clean up old files in an S3 bucket.

Step 1. S3 Setup

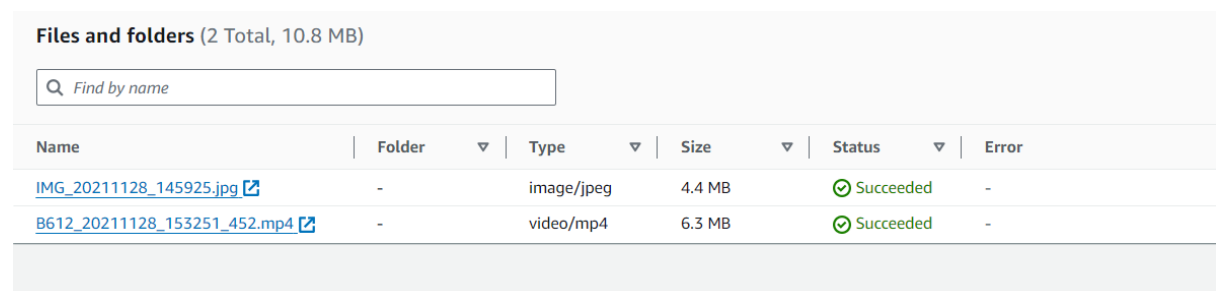
1. Create a New S3 Bucket:

- Go to the s3 Dashboard
- Click "Create bucket".



2. Upload Files:

- Navigate to your bucket.
- Click "Upload" and add multiple files to the bucket. Ensure some files have a modified date older than 30 days.



Step 2. Create IAM Role for Lambda

- Name the role 'sonal-LambdaS3CleanupRole'
- Attach the AmazonS3FullAccess policy.
- Create the role

Step 3. Create the Lambda Function


- Give your function a name 'sonal-s3AutoCleanup'
- Choose Python 3.11 as the runtime.
- Under "Permissions", choose "Use an existing role".
- Select the role you created earlier 'sonal-LambdaS3CleanupRole'.
- Click "Create function".


[Lambda](#) > [Functions](#) > **sonal-s3AutoCleanup**

sonal-s3AutoCleanup

▼ **Function overview** [Info](#)

Diagram Template

 **sonal-s3AutoCleanup**

 Layers (0)

+ Add trigger

+ Add destination

Step 4. Write the Boto3 Python script to:

- Initialize a boto3 S3 client.
- List objects in the specified bucket.
- Delete objects older than 30 days.
- Print the names of deleted objects for logging purposes.

```

1  import boto3
2  from datetime import datetime, timezone, timedelta
3
4  def lambda_handler(event, context):
5      s3 = boto3.client('s3')
6
7      bucket_name = 'sonal-bucket'
8      days_old = 30 # Temporary adjustment for testing
9      delete_time = datetime.now(timezone.utc) - timedelta(days=days_old)
10
11     # List objects in the specified bucket
12     response = s3.list_objects_v2(Bucket=bucket_name)
13
14     print(response)
15
16     if 'Contents' not in response:
17         print("Bucket is empty.")
18         return {
19             'statusCode': 200,
20             'body': "Bucket is empty."
21         }
22
23     delete_keys = []
24     for obj in response['Contents']:
25         print(f"Object: {obj['Key']}, LastModified: {obj['LastModified']}")
26         if obj['LastModified'] < delete_time:
27             delete_keys.append({'key': obj['Key']})
28             print(f"Object {obj['Key']} is older than 30 days and will be deleted")
29
30     try:
31         if delete_keys:
32             print(f"Deleting objects: {delete_keys}")
33             s3.delete_objects(Bucket=bucket_name, Delete={'Objects': delete_keys})
34             deleted_objects = [key['Key'] for key in delete_keys]
35             print(f"Deleted objects: {deleted_objects}")
36         else:
37             print("No objects older than 30 days found.")
38
39     except ClientError as e:
40         print(f"An error occurred: {e}")

```

Python script

```
s3.py  X  s3local.py
s3.py > lambda_handler
1  import boto3
2  from datetime import datetime, timezone, timedelta
3
4  def lambda_handler():
5      s3 = boto3.client('s3')
6      bucket_name = 'sonal-bucket'
7      days_old = 30 # Temporary adjustment for testing
8      delete_time = datetime.now(timezone.utc) - timedelta(days=days_old)
9
10     # List objects in the specified bucket
11     response = s3.list_objects_v2(Bucket=bucket_name)
12
13     print(response)
14
15     if 'Contents' not in response:
16         print("Bucket is empty.")
17         return {
18             'statusCode': 200,
19             'body': "Bucket is empty."
20         }
21
22     delete_keys = []
23     for obj in response['Contents']:
24         print(f"Object: {obj['Key']}, LastModified: {obj['LastModified']}")
25         if obj['LastModified'] < delete_time:
26             delete_keys.append({'Key': obj['Key']})
27             print(f"Object {obj['Key']} is older than 30 days and will be deleted")
28
29     if delete_keys:
30         print(f"Deleting objects: {delete_keys}")
31         s3.delete_objects(Bucket=bucket_name, Delete={'Objects': delete_keys})
32         deleted_objects = [key['Key'] for key in delete_keys]
33         print(f"Deleted objects: {deleted_objects}")
34     else:
35         print("No objects older than 30 days found.")
36
37     return {
38         'statusCode': 200,
39         'body': f"Deleted objects: {delete_keys}"
40     }
```

Step 4. Manual Invocation:

- After saving your function, manually trigger it.
- Go to the S3 dashboard and confirm that only files newer than 30 days remain.