

## Assignment 15: Implement a Log Cleaner for S3

**Objective:** Create a Lambda function that automatically deletes logs in a specified S3 bucket that are older than 90 days.

### 1. Create a new Lambda function.

Go to Lambda.

Click Create function.

Function name: nikhilS3LogCleanup90Days

Runtime: Python 3.13

Permissions: Choose Use an existing role (prashantb12-role-9p53470y).

The screenshot shows the AWS Lambda console interface for creating a new function. The breadcrumb navigation at the top indicates the path: Lambda > Functions > nikhilS3LogCleanup90Days > Edit basic settings. The main configuration area includes several sections:   
1. **Ephemeral storage**: A text input field is set to '512' MB. Below it, a note states: 'Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.'   
2. **SnapStart**: A dropdown menu is set to 'None'. A note explains that SnapStart reduces startup time by caching the function code.   
3. **Timeout**: Two input fields are set to '1' min and '0' sec.   
4. **Execution role**: Two radio buttons are present. The first, 'Use an existing role', is selected. The second is 'Create a new role from AWS policy templates'.   
5. **Existing role**: A dropdown menu shows the selected role as 'service-role/prashantb12-role-9p53470y'. A link below the dropdown says 'View the prashantb12-role-9p53470y role on the IAM console.'   
The top of the console shows the AWS logo, a search bar, and account information: 'Canada (Central)' and 'Account ID: 9750-5002-4946'.

## Add Python Code

```
Assignment15.py 1 X
D:\> DevOpsAndCloud > HeroViered > ServerlessArchitectureAndCloudAutomation > Assignment-15 > Assignment15.py > lambda_handler

1 import boto3
2 import datetime
3
4 def lambda_handler(event, context):
5     BUCKET_NAME = "nikhilmathur-s3"
6     DAYS_THRESHOLD = 20 #20 days old files for this test
7
8     s3 = boto3.client("s3")
9
10    # Calculate cutoff date (UTC)
11    cutoff_date = datetime.datetime.now(datetime.timezone.utc) - datetime.timedelta(days=DAYS_THRESHOLD)
12
13    deleted_files = []
14    continuation_token = None
15
16    # Use paginator to handle >1000 objects
17    while True:
18        if continuation_token:
19            response = s3.list_objects_v2(Bucket=BUCKET_NAME, ContinuationToken=continuation_token)
20        else:
21            response = s3.list_objects_v2(Bucket=BUCKET_NAME)
22
23        if "Contents" not in response:
24            print(f"No objects found in bucket {BUCKET_NAME}")
25            break
26
27        for obj in response["Contents"]:
28            key = obj["Key"]
29            last_modified = obj["LastModified"]
30
31            if last_modified < cutoff_date:
32                print(f"Deleting {key} (LastModified: {last_modified})")
33                s3.delete_object(Bucket=BUCKET_NAME, Key=key)
34                deleted_files.append(key)
35
36        # Check if more pages of results exist
37        if response.get("IsTruncated"):
38            continuation_token = response["NextContinuationToken"]
39        else:
40            break
41
42    print(f"✅ Deleted {len(deleted_files)} objects from {BUCKET_NAME}")
43    return {"deleted_count": len(deleted_files), "deleted_files": deleted_files}
44
```

## Click on test for Output

The screenshot shows the AWS Lambda console interface. At the top, a green banner indicates "Successfully updated the function nikhilS3LogCleanup90Days." Below this, the "TEST" tab is selected, showing a list of test events. The "Test" button is highlighted. The "Execution Results" tab is active, displaying the output of the function. The output is a JSON object with "deleted\_count" set to 2218 and "deleted\_files" containing a list of file keys. The console also shows the function's code in the background.

```
7 DAYS_THRESHOLD = 20 #20 days old files
8
9 s3 = boto3.client("s3")

Response:
{"deleted_count": 2218,
 "deleted_files": [
  "2025-07-24-02-37-25-159CA9EF7A0D31C7",
  "2025-07-24-02-38-39-10B236A8420ACDE2",
  "2025-07-24-02-47-29-61E6D041D380BB4",
  "2025-07-24-03-19-06-4B5D56C5C14362A",
  "2025-07-24-03-30-45-8E8B6EABF886E99",
  "2025-07-24-03-39-11-CCDD80F5D67B09A3",
  "2025-07-24-03-40-48-41DF353DEE2B19CA",
  "2025-07-24-03-44-42-847CA55EF4A2F766",
  "2025-07-24-03-50-25-984BD69E10E1F3D8",
  "2025-07-24-03-51-14-B2F2EC15148B1E0B",
  ...
  ]
}
```