Lab 13: Event-Driven AWS Lambda with S3 Trigger and SNS Notification

---

Objective:

Deploy a Lambda function that is automatically triggered when a file is uploaded to an S3 bucket. The Lambda function will publish a notification to an SNS topic informing about the event.

Estimated Duration: 60 minutes

Prerequisites:

- AWS Free Tier account
- IAM user with S3, Lambda, and SNS access
- Confirmed email address for SNS subscription

Part A: Create an S3 Bucket

1. Navigate to **S3** > **Create bucket**
2. Bucket name: `lambda-s3-trigger-demo` (must be globally unique)
3. Region: e.g., `ap-south-1`
4. Disable "Block all public access"
5. Leave all other settings as default and click **Create bucket**

Part B: Create an SNS Topic and Subscription

1. Go to **SNS** > **Topics** > **Create topic**
2. Type: **Standard**
3. Name: `lambda-s3-notify`
4. Click **Create topic**

Create Email Subscription:

- Go to the topic and click **Create subscription**
- Protocol: `Email`
- Endpoint: your email (e.g., `student@example.com`)
- Confirm the email by clicking the link received in your inbox

Part C: Create a Lambda Function

1. Go to **Lambda** > **Create function**
2. Name: `S3TriggerProcessor`
3. Runtime: **Python 3.10**
4. Execution Role:

- Select "Create a new role with basic Lambda permissions"
- Attach additional policies after creation:
    - `AmazonS3ReadOnlyAccess`
    - `AmazonSNSFullAccess`

Part D: Add S3 Trigger to Lambda

1. In Lambda, go to **Configuration** > **Triggers** > **Add Trigger**
2. Source: `S3`
3. Bucket: `lambda-s3-trigger-demo`
4. Event type: `PUT`
5. Click **Add**

Part E: Lambda Function Code

1. In Lambda > Code > Replace default code with:

```
import json
import boto3

def lambda_handler(event, context):
    sns_client = boto3.client('sns')
    sns_topic_arn = 'add your arn'

    records = event.get('Records', [])
    if not records:
        return {"statusCode": 400, "body": "No S3 event records."}

    s3_info = records[0]['s3']
    bucket = s3_info['bucket']['name']
    file_key = s3_info['object']['key']

    message = f"""
    Lambda executed successfully!
    File: {file_key}
    □ Bucket: {bucket}
    """

    sns_client.publish(
        TopicArn=sns_topic_arn,
        Subject="S3 File Upload Notification",
        Message=message
    )

    return {
        'statusCode': 200,
        'body': json.dumps('Notification sent successfully.')
    }
```

2. Click **Deploy** to save changes

Part F: Test the Integration

1. Go to **S3** > **lambda-s3-trigger-demo** > **Upload**
2. Upload any file (e.g., `report.txt`)
3. Wait a few seconds
4. Check your email inbox for an SNS notification like:

```
Lambda executed successfully!
File: report.txt
Bucket: lambda-s3-trigger-demo
```

Part G: Cleanup (Optional)

- Delete Lambda function
- Delete SNS topic and email subscription
- Delete S3 bucket

Student Assignment

- Modify the Lambda code to include file size and upload timestamp
- Add HTML formatting to the email body
- Extend the logic to only notify for `.pdf` or `.jpg` uploads