**Task: Trigger Lambda on S3 Upload to Send Email via SNS**

**Name:** Abhishek Hengade

**Batch:** B-56

This guide provides a comprehensive walkthrough of setting up a serverless workflow on AWS. The goal is to automatically send an email notification whenever a new file is uploaded to an S3 bucket. This is achieved by using an S3 event to trigger a Lambda function, which then publishes a message to an SNS topic.

**## 1. Create an SNS Topic and Subscription**

The first step is to create a **Simple Notification Service (SNS)** topic, which will act as a communication channel, and then subscribe an email address to it.

* **Create the SNS Topic**
  + Navigate to the SNS console and click **Create topic**.
  + Select the **Standard** type for flexibility.
  + Provide a name for your topic, such as s3-task.
* **Create an Email Subscription**
  + Once the topic is created, create a subscription to it.
  + Set the **Protocol** to **Email** and enter the desired email address in the **Endpoint** field.
  + After creating the subscription, its status will be "Pending confirmation" until you click the verification link sent to your email inbox.
* **Confirm the Subscription**
  + Check your email for a message from AWS and click the confirmation link. The subscription status in the SNS console will then change to **Confirmed**.

**## 2. Create an IAM Role and Policy for Lambda**

Next, you need to create an IAM role that grants your Lambda function the necessary permissions to execute and interact with other AWS services.

* **Create the IAM Role**
  + In the IAM console, create a new role.
  + For the **Trusted entity type**, select **AWS service**, and for the **Use case**, choose **Lambda**. This allows the Lambda service to assume this role.
* **Attach a Basic Execution Policy**
  + In the permissions step, search for and attach the AWS managed policy named AWSLambdaBasicExecutionRole. This policy grants the function permission to write logs to Amazon CloudWatch, which is essential for debugging.
* **Create and Attach a Custom Policy for SNS**
  + To allow the Lambda function to send messages, you need to create a custom policy.
  + This policy will grant the sns:Publish action on the specific SNS topic you created earlier. It is crucial to use the full **ARN (Amazon Resource Name)** of the SNS topic in the Resource field of the policy.
  + Attach this custom policy to your role.

**## 3. Create and Configure the Lambda Function**

The Lambda function contains the code that will be executed when a file is uploaded to S3.

* **Create the Function**: In the Lambda console, create a new function from scratch, selecting a runtime like Python.
* **Assign the IAM Role**: In the function's permissions settings, assign the IAM role you created in the previous step.
* **Add the Code**: The function's code should be written to parse the S3 event, extract details like the bucket name and filename, and then use the AWS SDK to publish a message to your SNS topic.
* **Set Environment Variables**: It's a best practice to store the SNS topic's ARN in an environment variable within the Lambda function's configuration.

**## 4. Configure the S3 Trigger**

The final step is to configure the S3 bucket to invoke your Lambda function automatically.

* **Add a Trigger**: In the Lambda function's configuration, add a new trigger and select S3 as the source.
* **Specify Bucket and Event Type**: Choose your S3 bucket (e.g., yoga-bucket) and set the **Event type** to **All object create events**.
* **Save**: Once saved, the connection is active.

With this setup complete, every new file uploaded to your S3 bucket will trigger the Lambda function, which in turn will publish a message to your SNS topic, sending a notification to all subscribed email addresses.

