**Lab Exercise- Create a Lambda Function to Run Athena Data Queries with Parameters**

This lab exercise guides you through creating an AWS Lambda function that runs parameterized queries on Amazon Athena. This is particularly useful for dynamic query generation based on input parameters, allowing you to customize data retrieval and processing.

**Objective:**

* Create a Lambda function that runs an Athena query with parameters.
* Use input parameters to dynamically construct and execute Athena queries.
* Retrieve and store the query results in an S3 bucket.

**Step 1: Set Up Your Environment**

**Create an S3 Bucket:**

* You'll need an S3 bucket to store the query results.
* Create a bucket using the S3 console or the AWS CLI:

aws s3 mb s3://your-bucket-name

**Create an IAM Role for Lambda:**

The Lambda function needs permissions to execute Athena queries and read/write to S3.

Create a new IAM role with the following policies:

* **AmazonAthenaFullAccess**
* **AmazonS3FullAccess**
* **AWSLambdaBasicExecutionRole**

Note the ARN of this role for later use when creating the Lambda function.

**Step 2: Create the Lambda Function**

Go to AWS Lambda Console:

* Create a new function and select "Author from scratch."
* Choose the Python 3.x runtime and attach the IAM role you created earlier.

**Write the Lambda Function Code:**

Use the following Python code to create a Lambda function that runs an Athena query with parameters:

import boto3

import time

def lambda\_handler(event, context):

# Athena client

client = boto3.client('athena')

# Extract parameters from the event

table\_name = event.get('table\_name', 'your\_default\_table')

condition = event.get('condition', '1=1') # Default to no filtering

# Set up the query

query = f"SELECT \* FROM {table\_name} WHERE {condition} LIMIT 10;"

database = "your\_database"

output = "s3://your-bucket-name/athena-results/"

# Start the query execution

response = client.start\_query\_execution(

QueryString=query,

QueryExecutionContext={'Database': database},

ResultConfiguration={'OutputLocation': output}

)

# Get the query execution ID

query\_execution\_id = response['QueryExecutionId']

# Wait for the query to finish

while True:

status = client.get\_query\_execution(QueryExecutionId=query\_execution\_id)

state = status['QueryExecution']['Status']['State']

if state in ['SUCCEEDED', 'FAILED', 'CANCELLED']:

break

time.sleep(2)

# Check the status

if state == 'SUCCEEDED':

# Fetch results if needed

result = client.get\_query\_results(QueryExecutionId=query\_execution\_id)

return {

'statusCode': 200,

'body': f"Query succeeded and results are stored in {output}"

}

else:

return {

'statusCode': 400,

'body': f"Query {state}"

}

Replace "your\_database" and "your-bucket-name" with your actual Athena database and S3 bucket names.

**Deploy the Lambda Function:**

Deploy the function and test it with a sample event.

**Step 3: Test the Lambda Function**

Invoke the Lambda Function with Parameters:

Manually invoke the function from the Lambda console with a test event like this:

{

"table\_name": "your\_table\_name",

"condition": "customer\_id > 1001"

}

This example will query records from the your\_table\_name table where customer\_id is greater than 1001.

**Verify the Results:**

After the Lambda function runs, check the specified S3 bucket (s3://your-bucket-name/athena-results/) for the query output.