**Lab Exercise- Create a Lambda UDF for Calling Athena Data Queries**

Creating a Lambda function that acts as a User Defined Function (UDF) for calling Athena queries is a powerful way to extend Athena's capabilities. Below is a step-by-step lab exercise to help you understand how to create and use such a function.

**Step 1: Set Up Athena**

* Create a Sample Database and Table in Athena:
* Log in to the AWS Management Console.
* Open the Amazon Athena service.

Run the following SQL commands in the Athena query editor:

CREATE DATABASE IF NOT EXISTS sample\_db;

CREATE EXTERNAL TABLE IF NOT EXISTS sample\_db.customers (

customer\_id INT,

customer\_name STRING,

customer\_address STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 's3://your-bucket-name/sample-data/';

Replace s3://your-bucket-name/sample-data/ with your actual S3 path.

**Step 2: Create an IAM Role for Lambda**

* Create a Role with Permissions:
* Navigate to the IAM console.

**Create a new role with the following policies:**

* **AmazonAthenaFullAccess**: To allow Athena queries.
* **AmazonS3FullAccess**: To allow access to data stored in S3.
* **CloudWatchLogsFullAccess**: To allow logging from Lambda.

Attach this role to your Lambda function in the next step.

**Step 3: Create the Lambda Function**

**Open the AWS Lambda Console:**

* Navigate to the AWS Lambda service.
* Click "Create function."
* Choose "Author from scratch."
* Function Name: AthenaQueryFunction
* Runtime: Python 3.x (e.g., Python 3.8)
* Role: Use the role created in Step 2.

**Write the Lambda Function:**

Replace the default code with the following Python code:

import boto3

import time

def lambda\_handler(event, context):

# Initialize the Athena and S3 clients

athena = boto3.client('athena')

s3\_output = 's3://your-bucket-name/athena-results/' # Replace with your bucket

# Define the query

query = "SELECT \* FROM sample\_db.customers LIMIT 10;"

# Start the Athena query

response = athena.start\_query\_execution(

QueryString=query,

QueryExecutionContext={

'Database': 'sample\_db'

},

ResultConfiguration={

'OutputLocation': s3\_output

}

)

# Get the query execution ID

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

# Wait for the query to complete

while True:

query\_status = athena.get\_query\_execution(QueryExecutionId=query\_execution\_id)

status = query\_status['QueryExecution']['Status']['State']

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

break

time.sleep(1)

# Fetch the query results

if status == 'SUCCEEDED':

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

return {

'statusCode': 200,

'body': result['ResultSet']['Rows']

}

else:

return {

'statusCode': 500,

'body': 'Query failed: ' + status

}

Replace s3://your-bucket-name/athena-results/ with your actual S3 path for Athena

query results.

**Deploy the Lambda Function:**

* Click "Deploy" to save and deploy the Lambda function.

**Step 4: Test the Lambda Function**

**Create a Test Event:**

* In the Lambda console, create a new test event.
* Use the default event template (you don’t need to pass any specific input for this simple example).

**Run the Test:**

* Click "Test" to run the Lambda function.
* The function should execute the Athena query and return the results.