

BDM 3203 - Hadoop Ecosystems for Big Data 01

Assignment 2

Submitted by: Group E

Student IDs:

Aadarsha Chapagain(C0825975)

Roshan Acharya (C0831342)

**Anjana Kuriakose
(C0829580)**

Onyinye Mbanefo (C0831578)

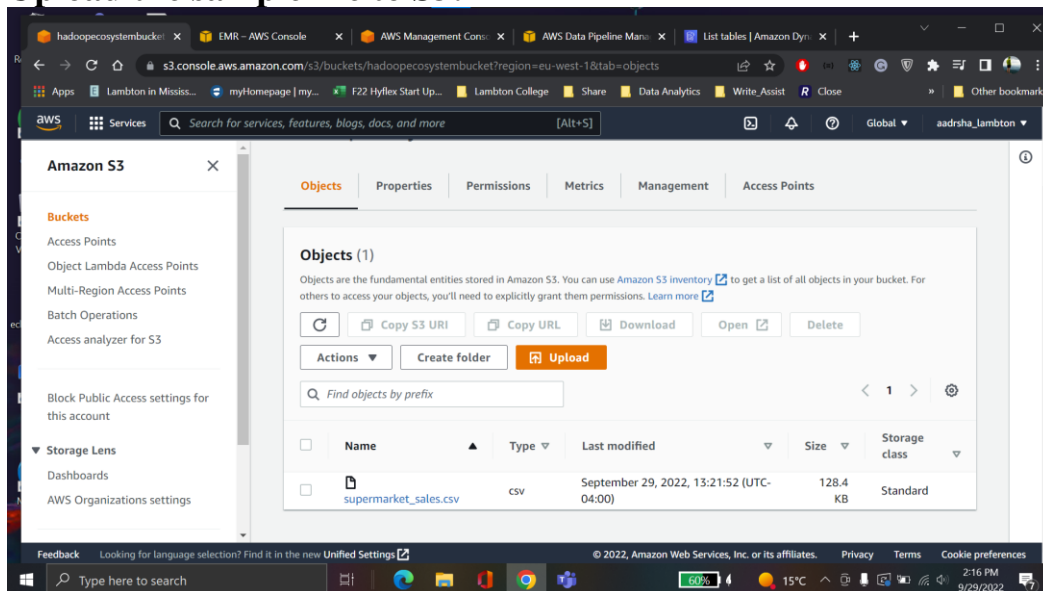
Submitted to: Prof. Teresa Zhu

Create a Dynamo DB in AWS and choose any dataset (MUST BE APPROVED BY YOUR INSTRUCTOR) then run a few queries (7-10 queries based on the business questions you create)

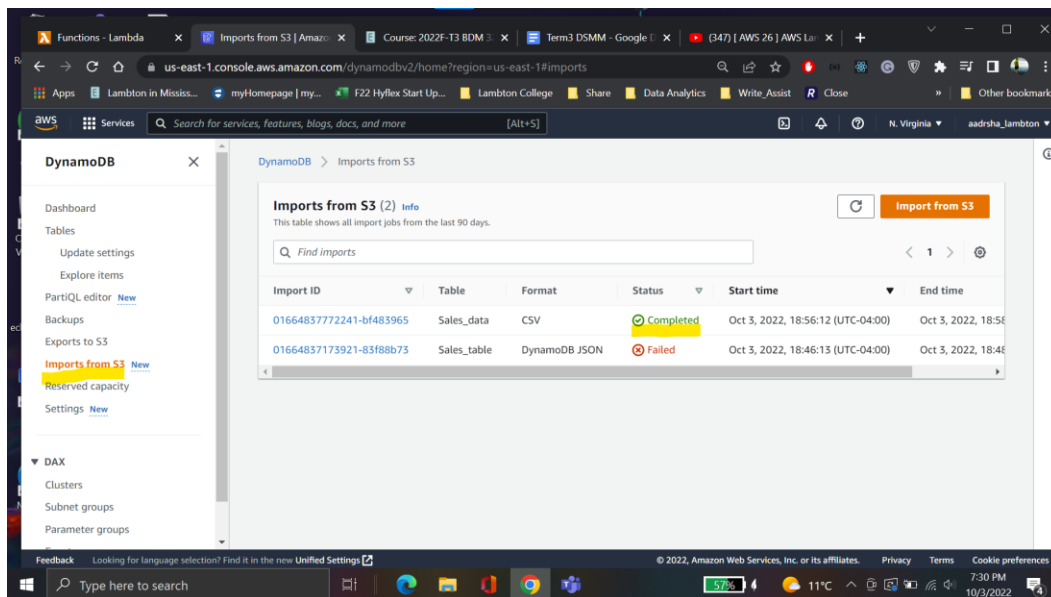
Implement and take screenshots with time stamps and upload your report.

Screenshots

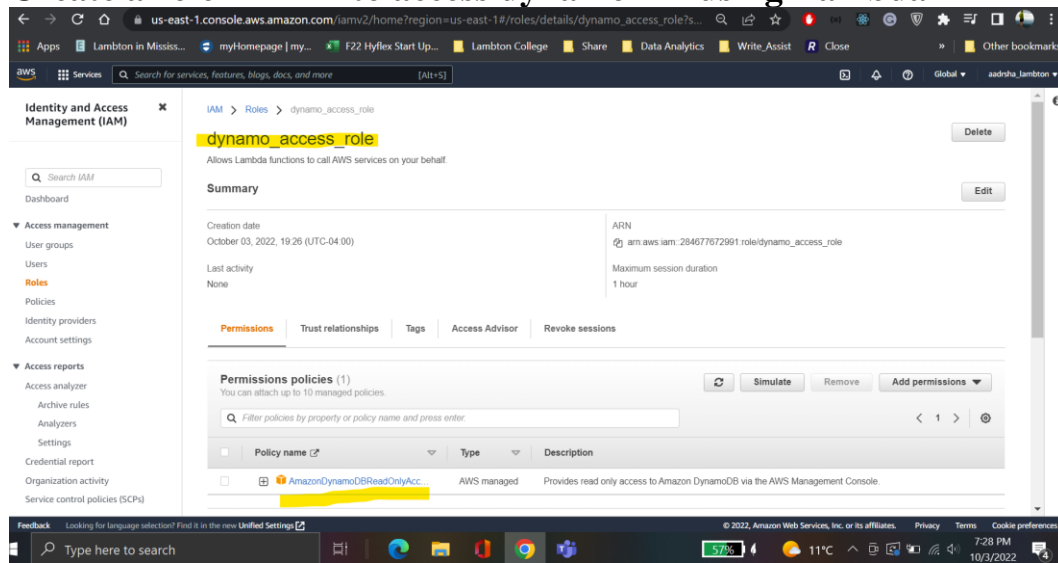
Upload the sample file to S3:



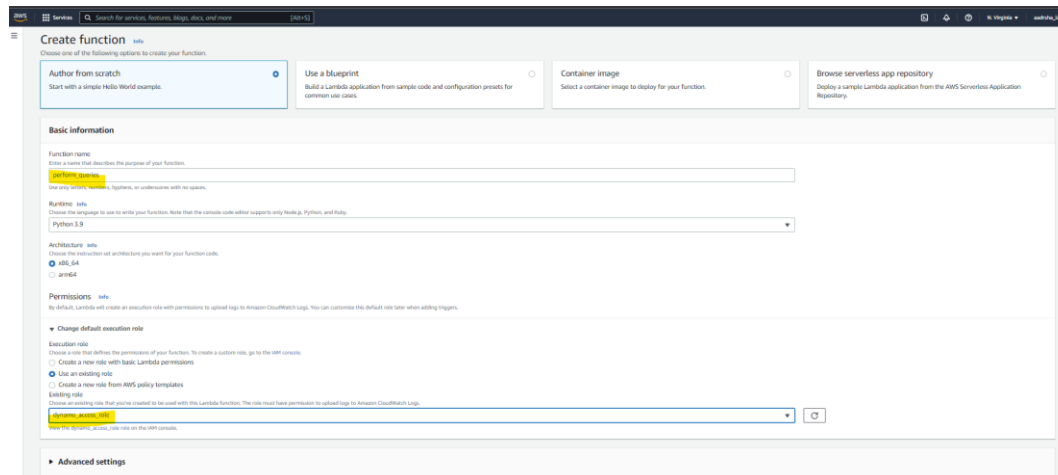
Use Import from S3 feature of dynamodb to import data from S3 to Dynamodb



Create a role in IAM to access dynamo DB using Lambda



Create Lambda Functions



Use boto Library to perform queries in dynamo DB

1. Table.scan()

import boto3

```
def lambda_handler(event, context):
    client = boto3.resource("dynamodb")
    table = client.Table("Sales_table")
    projections = table.scan
    print("Projections:",projections)
```

```
# TODO implement
return None
```

Execution results

Status: Succeeded Max memory used: 66 MB Time: 1151.86 ms

Test Event Name
test_event

Response
null

Function Logs
START RequestId: 9a7574c1-bc4b-4ec5-8a23-4a850766fa08 Version: \$LATEST
Projections: <bound method ResourceFactory._create_action.<locals>.do_action of dynamodb.Table(name='Sales_table')>
END RequestId: 9a7574c1-bc4b-4ec5-8a23-4a850766fa08
REPORT RequestId: 9a7574c1-bc4b-4ec5-8a23-4a850766fa08 Duration: 1151.86 ms Billed Duration: 1152 ms Memory Size: 128 MB

Request ID
9a7574c1-bc4b-4ec5-8a23-4a850766fa08

2.

Get_item

import boto3

```
def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('Sales_data')
    response = table.get_item(
        Key={
            'Invoice ID': '560-30-5617'
        }
    )
    print("response:",response)
    return None
```

Successfully updated the function perform_queries.

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment

perform_queries
lambda_function.py

Execution results

Status: Succeeded Max memory used: 68 MB Time: 1368.06 ms

Test Event Name
test_event

Response
null

Function Logs
START RequestId: fab4478e-b6c9-4985-a450-3bd820c8ebec Version: \$LATEST
response: {'Item': {'Date': '3/24/2019', 'City': 'Mandalay', 'gross income': '6.1925', 'Invoice ID': '560-30-5617', 'cogs': '123.'}}
END RequestId: fab4478e-b6c9-4985-a450-3bd820c8ebec
REPORT RequestId: fab4478e-b6c9-4985-a450-3bd820c8ebec Duration: 1368.06 ms Billed Duration: 1369 ms Memory Size: 128 MB

Request ID
fab4478e-b6c9-4985-a450-3bd820c8ebec

3.

```
def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('Sales_data')
    response = table.get_item(
        Key={
            'Branch': 'B'
        }
    )
    print("response:",response)
    return None
```

Execution results

Status: Succeeded Max memory used: 68 MB Time: 1385.51 ms

Test Event Name
test_event

Response
null

Function Logs
START RequestId: baf9a5f7-fb73-4a69-a873-9f482a26a622 Version: \$LATEST
response: {'Item': {'Date': '3/2/2019', 'City': 'Mandalay', 'gross income': '48.69', 'Invoice ID': '303-96-2227', 'cogs': '973.8'}}
END RequestId: baf9a5f7-fb73-4a69-a873-9f482a26a622
REPORT RequestId: baf9a5f7-fb73-4a69-a873-9f482a26a622 Duration: 1385.51 ms Billed Duration: 1386 ms Memory Size: 128 MB

Request ID
baf9a5f7-fb73-4a69-a873-9f482a26a622

4. Projection Expressions

```
import boto3
def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('Sales_data')
    resp = table.scan(ProjectionExpression="City, cogs")
    print("response:",resp)
    return None
```

Execution results

Status: Succeeded Max memory used: 68 MB Time: 1455.53 ms

Test Event Name
test_event

Response
null

Function Logs
START RequestId: 192005e5-daf7-4118-a356-9d08e19edaa3 Version: \$LATEST
response: {'Items': [{'cogs': '618.38', 'City': 'Yangon'}, {'cogs': '973.8', 'City': 'Mandalay'}, {'cogs': '40.35', 'City': 'Naypyit'}]}
END RequestId: 192005e5-daf7-4118-a356-9d08e19edaa3
REPORT RequestId: 192005e5-daf7-4118-a356-9d08e19edaa3 Duration: 1455.53 ms Billed Duration: 1456 ms Memory Size: 128 MB Max

Request ID
192005e5-daf7-4118-a356-9d08e19edaa3

5. KeyCondition Expression

```
import json
import boto3
from boto3.dynamodb.conditions import Key

def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('Sales_data')

    resp = table.query(
        KeyConditionExpression=
        Key('Branch').eq('B') & Key('Customer Type').eq('Normal')
    )

    print(resp['Items'][0])

    return None
```

6.

```
import json
import boto3

def lambda_handler(event, context):
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('Sales_data')

    response = table.delete_item(
        Key={
            'Invoice ID': '849-09-3807',
        },
    )
    return None
```

Summary

We have uploaded the data to AWS s3 imported it to Dynamodb and perform several queries using lambda function.

Reference

https://www.youtube.com/watch?v=8zhv6GDSDE8&ab_channel=JustmeandOpensource
<https://highlandsolutions.com/blog/hands-on-examples-for-working-with-dynamodb-boto3-and-python>