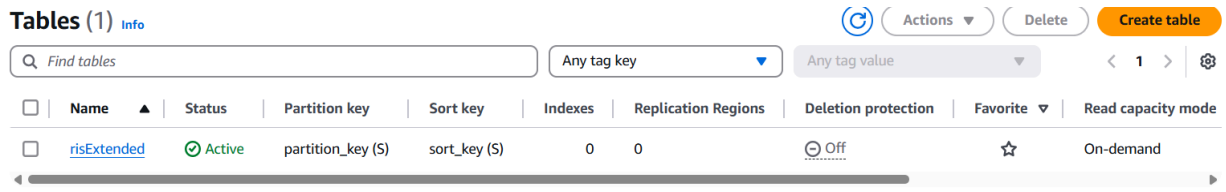
Mini 2 Lab 4 Report

Link to github: [Github]

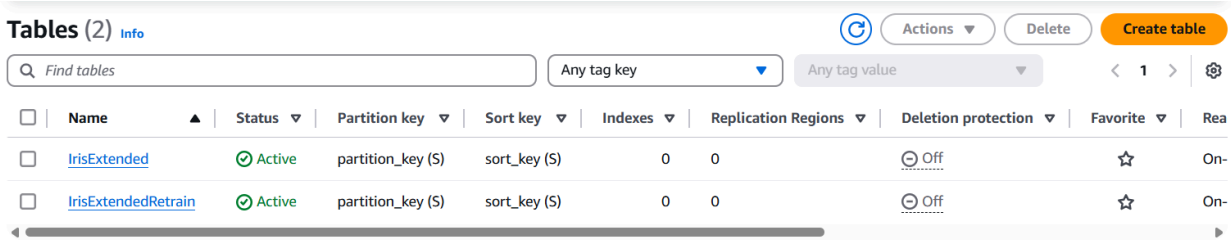
Preparation and Setup



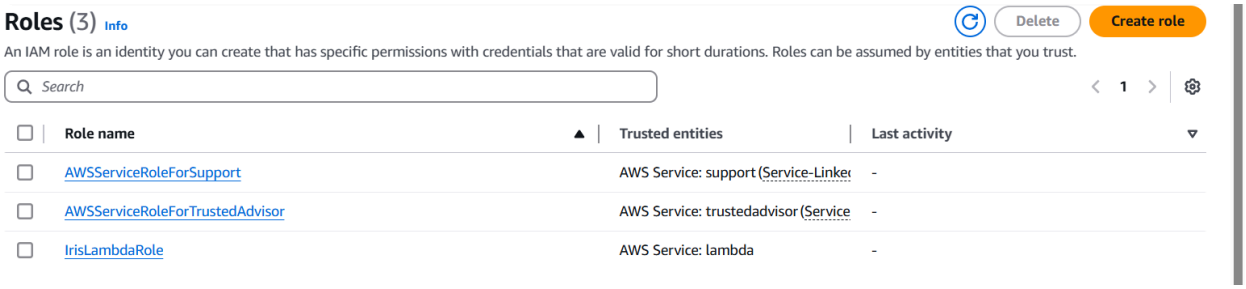
Boto3

```
(eleanorvenv) PS C:\Users\STUDENT\Documents\CMU\Spring 2025\AI Systems Design\Assignments\mini 2\Lab4\Lab4> python b11.py
{'AccountMaxReadCapacityUnits': 80000, 'AccountMaxWriteCapacityUnits': 80000, 'TableMaxReadCapacityUnits': 40000, 'TableMaxWriteCapacityUnits': 40000, 'ResponseMetadata':
{'RequestId': 'EBPUAGG0JUNFN5O5C58EUI2S97VV4KQNSO5AEMVJF66Q9ASUAAJG', 'HTTPStatusCode': 200, 'HTTPHeaders': {'server': 'Server', 'date': 'Thu, 01 May 2025 09:17:41 GMT', '
content-type': 'application/x-amz-json-1.0', 'content-length': '143', 'connection': 'keep-alive', 'x-amzn-requestid': 'EBPUAGG0JUNFN5O5C58EUI2S97VV4KQNSO5AEMVJF66Q9ASUAAJG
', 'x-amz-crc32': '3062975651'}, 'RetryAttempts': 0}}
```

Creating a second table

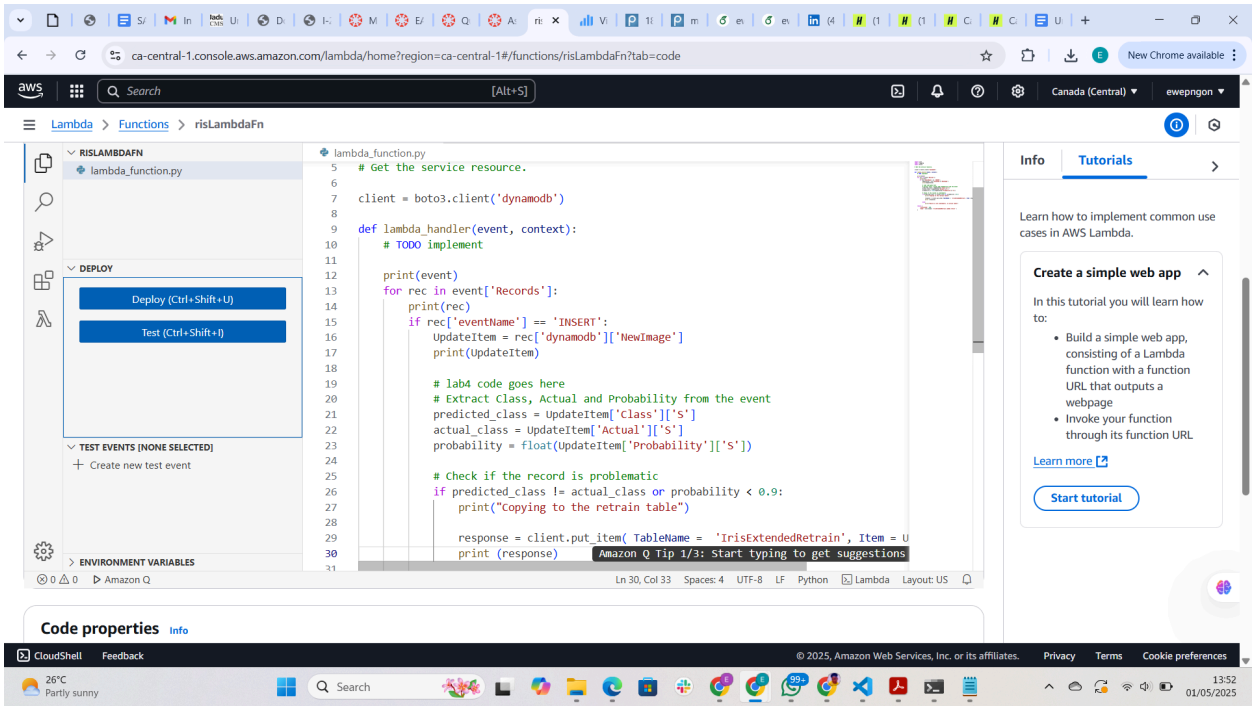


Creating Roles

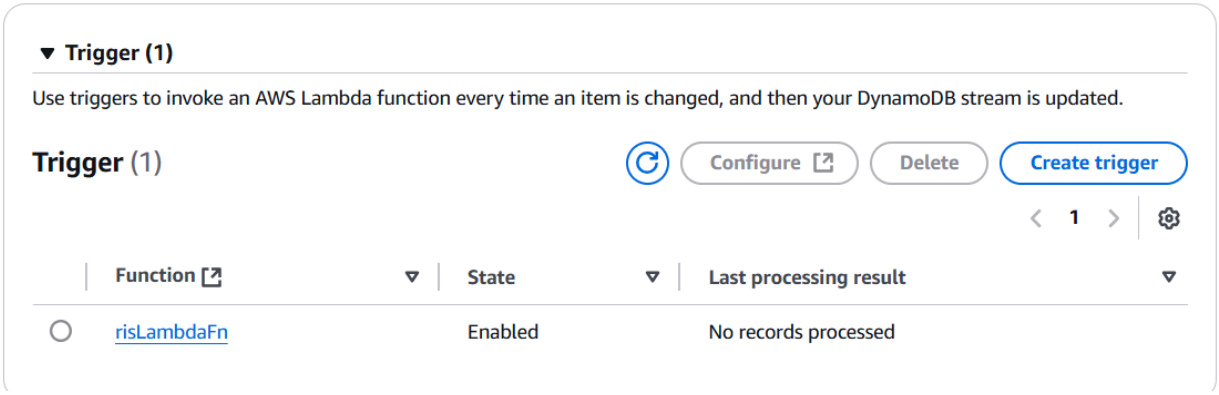


Create Trigger

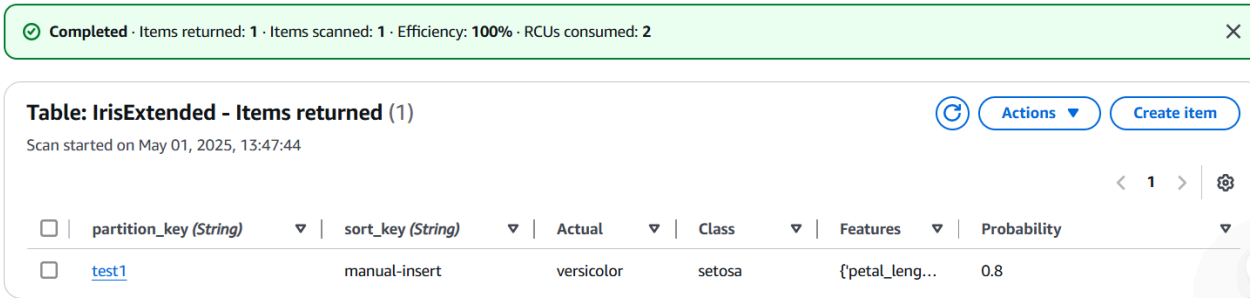
Trigger function



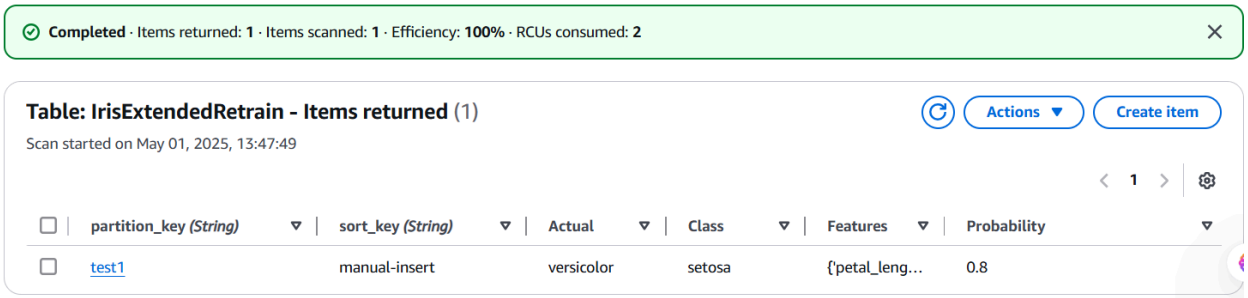
Trigger created



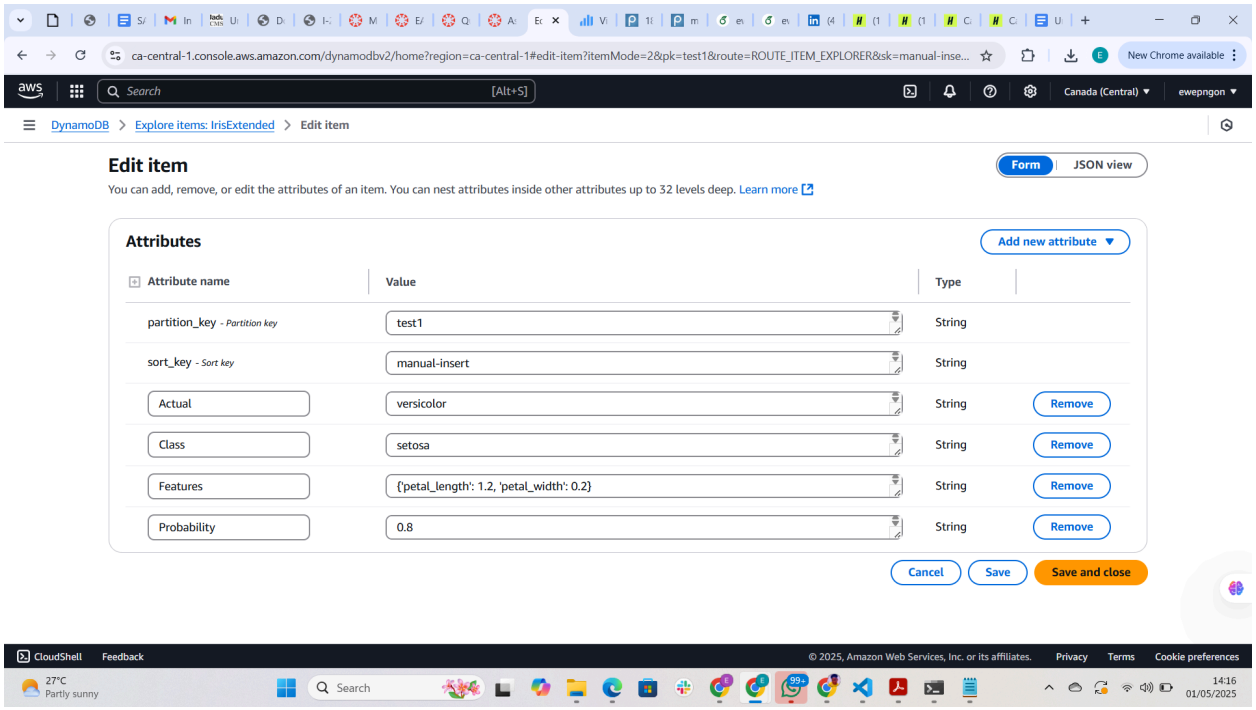
Complete Back-end Deployment Test



Iris Extended Retrained



Testing the Lambda fubction

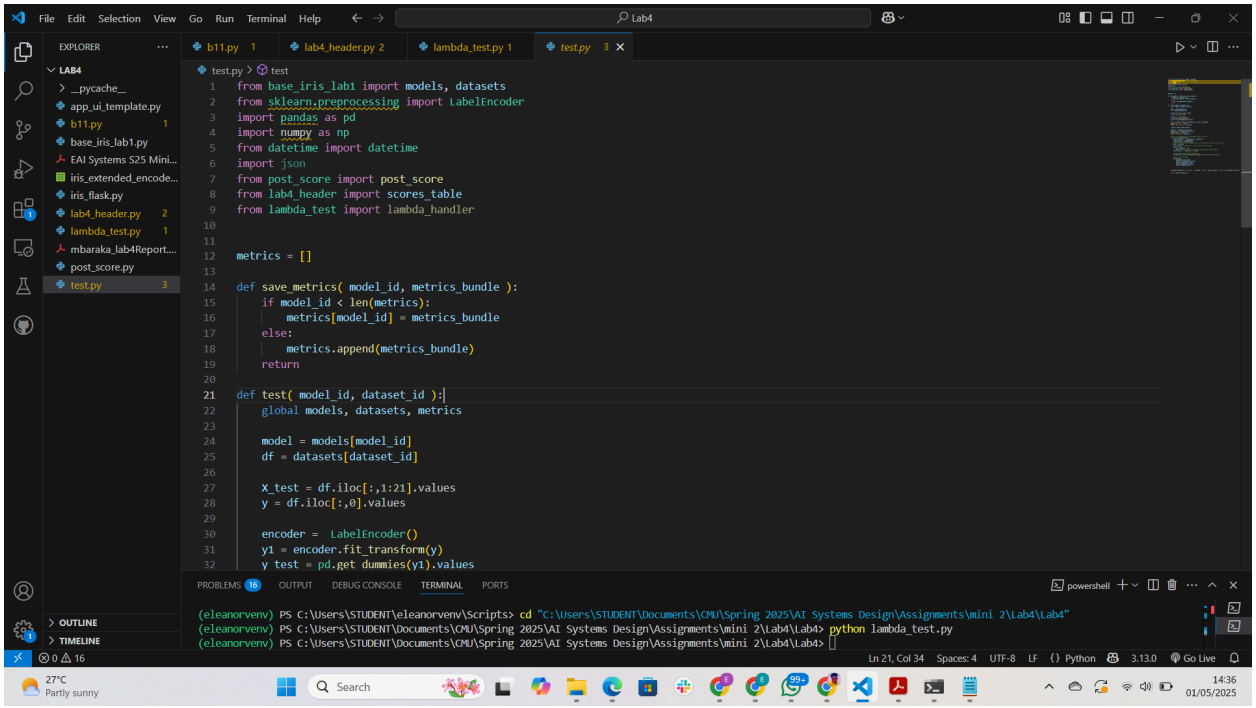


Updated Lambda.py code

```python
import json
import logging
import boto3

# Get the service resource.

client = boto3.client('dynamodb')

def lambda_handler(event, context):
    # TODO implement
```

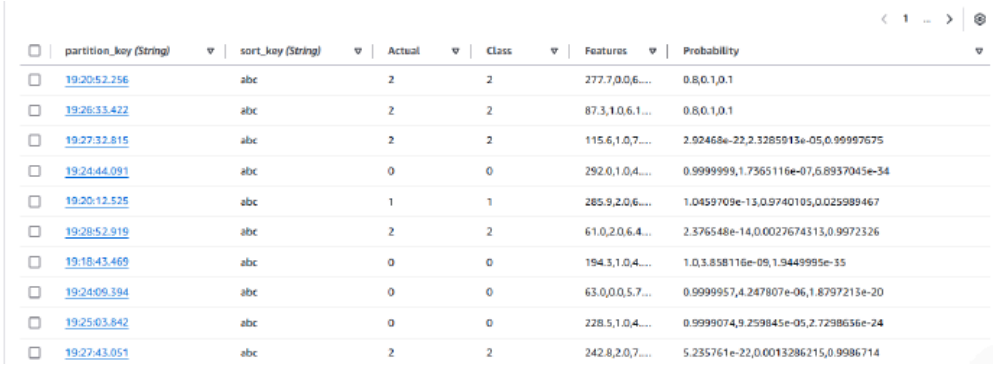
```
    print(event)
    for rec in event['Records']:
        print(rec)
        if rec['eventName'] == 'INSERT':
            UpdateItem = rec['dynamodb']['NewImage']
            print(UpdateItem)

            # lab4 code goes here
            # Extract Class, Actual and Probability from the event
            predicted_class = UpdateItem['Class']['S']
            actual_class = UpdateItem['Actual']['S']
            probability = float(UpdateItem['Probability']['S'])

            # Check if the record is problematic
            if predicted_class != actual_class or probability < 0.9:
                print("Copying to the retrain table")

                response = client.put_item( TableName =  'IrisExtendedRetrain',
Item = UpdateItem )
                print (response)

            else:
                print("Record is not problematic, no action taken")

    return {
        'statusCode': 200,
        'body': json.dumps( 'IrisExtendedRetrain Lambda return' )
    }
```

Updated Test.py function(Full code uploaded to github)

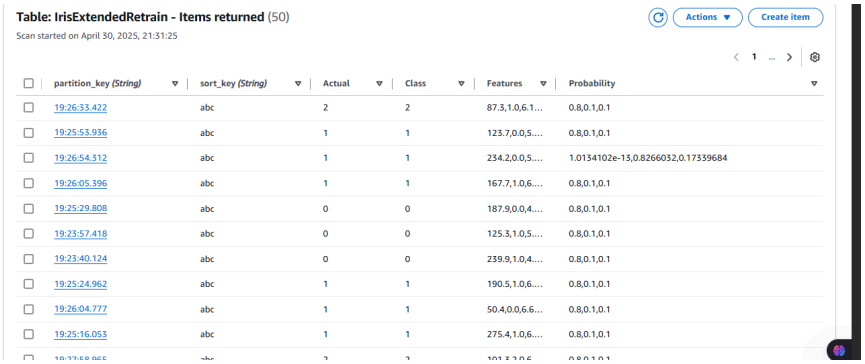


App Running

Communication to cloud

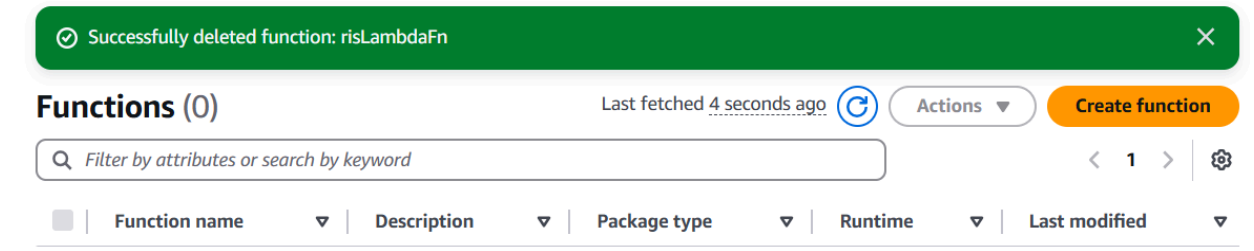


partition_key (String)	sort_key (String)	Actual	Class	Features	Probability
19:20:52.256	abc	2	2	277.7,0.0,6....	0.8,0.1,0.1
19:26:33.422	abc	2	2	87.3,1.0,6.1...	0.8,0.1,0.1
19:27:32.815	abc	2	2	115.6,1.0,7....	2.92468e-22,2.3285913e-05,0.99997675
19:24:44.091	abc	0	0	292.0,1.0,4....	0.9999999,1.7365116e-07,6.8937045e-34
19:20:12.525	abc	1	1	285.9,2.0,6....	1.0459709e-13,0.9740105,0.025989467
19:28:52.919	abc	2	2	61.0,2.0,6.4...	2.376548e-14,0.0027674313,0.9972326
19:18:43.469	abc	0	0	194.3,1.0,4....	1.0,3.858116e-09,1.9449995e-35
19:24:09.394	abc	0	0	63.0,0.0,5.7...	0.9999957,4.247807e-06,1.8797213e-20
19:25:03.842	abc	0	0	228.5,1.0,4....	0.9999074,9.259845e-05,2.7298636e-24
19:27:43.051	abc	2	2	242.8,2.0,7....	5.235761e-22,0.0013286215,0.9986714



Cleanup

Lambda function



Tables