

# **CMPE 272: Enterprise SW Plat**

## **Model Quality Monitoring Assignment**

### **Group 10**

Siddhant Satish Parmar

Roger Kuo

Anirudh Satyam

Kumar Raja Pavuluri

- 1. Explain how AWS Identity and Access Management (IAM) controls permissions using Roles, Policies, and Groups. What is the difference between a Role and a Policy?**

AWS access permissions are applied via Policies. Policies are objects that define permissions for whatever they are associated with. Policies can be applied to identities, such as Roles and Groups. The permissions in the Policies determine whether a request from an identity is allowed or denied. To grant access to services and resources, specific policies can be attached to identities and resources.

A Policy, as previously mentioned, is just a set of permissions. A Role is an identity that can be applied to multiple, unspecified users and Groups.

- 2. Explain the function of the two DynamoDB tables we created above?**

The first table “LabScores” is used to store input features and predicted classes. It is set up to notify the system, via a Lambda function, of incoming records and updates in the table. The second table “LabRetraining” is used for receiving and storing retraining records from the Lambda function after it passes all the tests written in the Lambda function. The two tables are set up for models to log results of credit scoring.

- 3. Outline the logic in the Lambda function. When is it called? What test does it apply against new items? Where does it write its results?**

The Lambda function has a function “createItems” which takes parameters “params” which is a JS object in which item details from the LabScores table is stored. Then we implement a for loop in which all the event records are traversed, and if the eventName is “INSERT” (i.e. if any item is inserted in the LabScores table then we will fetch that entry and store the item details in the “params” object.

Once the item details from the LabScores are fetched, we perform some tests on the new items, tests like:

- Checking if the “Class” is 0 then the “gap” will be equal to the “Probability” else the “gap” is equal to 1 - “Probability”.
- After setting the “gap”, we check if the “gap” > 0.25 then the “createItems” function will be called and we write the new items in the LabRetraining table.

Since we created a Trigger inside the LabScores table, the “LabLambdaFn” will be triggered whenever there is some change in the LabScores table, but it will write results in the LabRetraining table only when the event is “INSERT”.

#### **4. What is the role of CloudWatch in our AWS back-end?**

Cloudwatch helps monitor and manage our applications and resources. It helps with storing metrics in a repository, and statistics of our data can be retrieved from the metrics. It can be used to graph our statistical data. Cloudwatch also provides insights on optimizing performance and managing resource utilization. It provides a unified dashboard to help look over our operations. In our AWS backend it shows us eventID, "eventName (which in our case would be insert since the item in Labretraining is created when eventName == INSERT), eventVersion, eventSource, awsRegion, dynamodb (which shows us ApproximateCreationDateTime, Keys, NewImage, SequenceNumber, SizeBytes, StreamViewType(which would be NEW\_IMAGE in our case)

#### **5. Screenshots for A) Steps to create tables and Lambda function. B) List of items in Scoring table and Retraining table.**

**A) Steps to create tables and Lambda function:-**



- Setup a second DynamoDB table to receive retraining records.

The screenshot shows the 'Create table' wizard in the AWS Management Console. The 'Table details' section is active, showing the table name 'LabRetraining'. The 'Partition key' is 'partition\_key' and the 'Sort key' is 'sort\_key', both with a 'String' data type. The 'Settings' section shows 'Default settings' selected. A green banner at the top of the console indicates 'The LabRetraining table was created successfully.'

**Table details** [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**

This will be used to identify your table.

LabRetraining

Between 3 and 255 characters, containing only letters, numbers, underscores (`_`), hyphens (`-`), and periods (`.`).

**Partition key**

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

partition\_key String

1 to 255 characters and case sensitive.

**Sort key - optional**

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

sort\_key String

1 to 255 characters and case sensitive.

**Settings**

☒ **Default settings**

The fastest way to create your table. You can modify these settings now or after your table has been created.

☐ **Customize settings**

Use these advanced features to make DynamoDB work better for your needs.

**Default settings**

**DynamoDB**

Dashboard

**Tables**

Update settings

Explore items

PartiQL editor [New](#)

Backups

Exports to S3

Reserved capacity

**DAX**

Clusters

Subnet groups

Parameter groups

Events

Tell us what you think

Return to the previous console experience

Density settings

**Tables (2)** [Info](#)

Find tables by table name

Any table tag

	Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode	Size	Table class
<input type="checkbox"/>	LabRetraining	Active	partition_key (S)	sort_key (S)	0	Provisioned with auto scaling (S)	Provisioned with auto scaling (S)	0 bytes	DynamoDB Standard
<input type="checkbox"/>	LabScores	Active	partition_key (S)	sort_key (S)	0	Provisioned with auto scaling (S)	Provisioned with auto scaling (S)	0 bytes	DynamoDB Standard

- Creating the required AWS role and policies.

Services

Search for services, features, blogs, docs, and more

[Option+5]

Global

Siddhant Satish Parmar

IAM

Roles

Create role

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Select trusted entity

Trusted entity type

☒ AWS service

Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ Web identity

Allow users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ SAML 2.0 federation

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

☐ EC2

Allows EC2 instances to call AWS services on your behalf.

☒ Lambda

Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

Cancel

Next

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Services

Search for services, features, blogs, docs, and more

[Option+5]

Global

Siddhant Satish Parmar

IAM

Roles

Create role

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Add permissions

Permissions policies (Selected 2/751)

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter

4 matches

dynamodb

Clear filters

	Policy name	Type	Description
<input checked="" type="checkbox"/>	AmazonDynamoDBFullAccess	AWS m...	Provides full access to Amazon DynamoDB via the AWS Management Console.
<input checked="" type="checkbox"/>	AWSLambdaDynamoDBExecutionRole	AWS m...	Provides list and read access to DynamoDB streams and write permissions to CloudWatch logs.
<input type="checkbox"/>	AmazonDynamoDBReadOnlyAccess	AWS m...	Provides read only access to Amazon DynamoDB via the AWS Management Console.
<input type="checkbox"/>	AWSLambdaInvocation-DynamoDB	AWS m...	Provides read access to DynamoDB Streams.

Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Services

Search for services, features, blogs, docs, and more

[Option+S]

Global

Siddhant Satish Parmar

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

## Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

LabLambdaRole

Maximum 128 characters. Use alphanumeric and "+=, @, \_" characters.

Description

Add a short explanation for this policy.

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and "+=, @, \_" characters.

Step 1: Select trusted entities

Edit

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10        "Service": [
11          "lambda.amazonaws.com"
12        ]
13      }
14    ]
15  }
16 }
```

Edit

Step 2: Add permissions

Edit

Feedback

Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Services

Search for services, features, blogs, docs, and more

[Option+S]

Global

Siddhant Satish Parmar

Step 1

Step 2

Add permissions

Step 3

Name, review, and create

Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
<a href="#">AmazonDynamoDBFullAccess</a>	AWS managed	Permissions policy
<a href="#">AWSLambdaDynamoDBExecutionRole</a>	AWS managed	Permissions policy

Tags

Add tags (Optional)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags

Cancel

Previous

Create role

Feedback

Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- **Configure and deploy your Lambda function to monitor DynamoDB and forward retraining records.**

The screenshot shows the 'Enable DynamoDB stream' page in the AWS Management Console. The breadcrumb trail is 'DynamoDB > Tables > LabScores > Enable DynamoDB stream'. The page title is 'Enable DynamoDB stream'. Below the title is a section 'DynamoDB stream details' with a description: 'Capture item-level changes in your table, and push the changes to a DynamoDB stream. You then can access the change information through the DynamoDB Streams API.' Under 'View type', there are four radio button options: 'Key attributes only' (unselected), 'New image' (selected), 'Old image' (unselected), and 'New and old images' (unselected). At the bottom right of the form are 'Cancel' and 'Enable stream' buttons.

The screenshot shows the 'Create function' page in the AWS Management Console. The breadcrumb trail is 'AWS > Services > Lambda > Create function'. The page title is 'Create function'. Below the title is a section 'Choose one of the following options to create your function.' with four tabs: 'Author from scratch' (selected), 'Use a blueprint', 'Container image', and 'Browse serverless app repository'. Below the tabs is a section 'Basic information' with the following fields: 'Function name' (text input with value 'LabLambdaFn'), 'Runtime' (dropdown menu with value 'Node.js 14.x'), 'Architecture' (radio button options with 'x86\_64' selected), 'Permissions' (radio button options with 'Use an existing role' selected), and 'Execution role' (radio button options with 'Use an existing role' selected). At the bottom right of the form are 'Cancel' and 'Create function' buttons.

Services

Search for services, features, blogs, docs, and more

[Option+S]

Oregon

Siddhant Satish Parmar

Successfully created the function LabLambdaFn. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Code source

info

File Edit Find View Go Tools Window

Test Deploy

Changes not deployed

Go to Anything (⌘ P)

LabLambdaFn

index.js

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

var AWS = require('aws-sdk');

AWS.config.update({region: 'us-west-2'});

const docClient = new AWS.DynamoDB.DocumentClient();

async function createItem(params){

try {

await docClient.put(params).promise();

} catch (err) {

return err;

}

}

exports.handler = (event, context, callback) => {

event.Records.forEach((record) => {

var record\_str = JSON.stringify(record, null, 2);

console.log('Stream record: ', record\_str);

if (record.eventName === 'INSERT') {

var params = {

TableName: 'LabRetraining',

Item: {

"partition\_key": record.dynamodb.NewImage.partition\_key.S,

"sort\_key": record.dynamodb.NewImage.sort\_key.S,

"Class": record.dynamodb.NewImage.Class.S,

"Probability": record.dynamodb.NewImage.Probability.S,

"Label": record.dynamodb.NewImage.Label.S,

"Features": record.dynamodb.NewImage.Features.S

}

}

};

var int\_class = parseInt(record.dynamodb.NewImage.Class.S);

var float\_probability = parseFloat(record.dynamodb.NewImage.Probability.S);

51:1 JavaScript Spaces: 4

Code properties

Package size

304.0 byte

SHA256 hash

UTJfXfTosQY08f6CxoZ0bCL76H0dA48LnlMm4gpgDw=

Last modified

April 30, 2022, 07:16 PM PDT

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Services

Search for services, features, blogs, docs, and more

[Option+S]

Oregon

Siddhant Satish Parmar

Successfully updated the function LabLambdaFn.

Code source

info

File Edit Find View Go Tools Window

Test Deploy

Changes not deployed

Go to Anything (⌘ P)

LabLambdaFn

index.js

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

var AWS = require('aws-sdk');

AWS.config.update({region: 'us-west-2'});

const docClient = new AWS.DynamoDB.DocumentClient();

async function createItem(params){

try {

await docClient.put(params).promise();

} catch (err) {

return err;

}

}

exports.handler = (event, context, callback) => {

event.Records.forEach((record) => {

var record\_str = JSON.stringify(record, null, 2);

console.log('Stream record: ', record\_str);

if (record.eventName === 'INSERT') {

var params = {

TableName: 'LabRetraining',

Item: {

"partition\_key": record.dynamodb.NewImage.partition\_key.S,

"sort\_key": record.dynamodb.NewImage.sort\_key.S,

"Class": record.dynamodb.NewImage.Class.S,

"Probability": record.dynamodb.NewImage.Probability.S,

"Label": record.dynamodb.NewImage.Label.S,

"Features": record.dynamodb.NewImage.Features.S

}

}

};

var int\_class = parseInt(record.dynamodb.NewImage.Class.S);

var float\_probability = parseFloat(record.dynamodb.NewImage.Probability.S);

51:1 JavaScript Spaces: 4

Code properties

Package size

718.0 byte

SHA256 hash

+Oh7hNkNQuTaUAQx1zQ+o1JgTzV9XHgF8ZtQCv6JKU=

Last modified

April 30, 2022, 07:16 PM PDT

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Services

Search for services, features, blogs, docs, and more

[Option+S]

Oregon

Siddhant Satish Parmar

DynamoDB

Items: LabScores

Item editor

Create item

FormJSON

Attributes

Add new attribute

Attribute name	Value	Type
partition_key - Partition key	CMPE_272	String
sort_key - Sort key	Group_10	String

CancelCreate item

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

Services

Search for services, features, blogs, docs, and more

[Option+S]

Oregon

Siddhant Satish Parmar

DynamoDB

Tables

Update settings

Explore items

PartiQL editor

Backups

Exports to S3

Reserved capacity

DAX

Clusters

Subnet groups

Parameter groups

Events

Tell us what you think

Return to the previous console experience

Density settings

DynamoDB

Items

LabScores

Tables (2)

Any table tag

Find tables by table name

< 1 >

LabRetraining

LabScores

LabScores

Autopreview

Actions

Create item

Update table settings

Scan/Query items

ScanQuery

Table or index

LabScores

Filters

RunReset

CompletedRead capacity units consumed: 0.5

Items returned (1)

partition\_keysort\_key

CMPE\_272Group\_10

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

Services

Search for services, features, blogs, docs, and more

[Option+S]

Oregon

Siddhant Satish Parmar

DynamoDB

Tables

Update settings

Explore items

PartiQL editor

Backups

Exports to S3

Reserved capacity

DAX

Clusters

Subnet groups

Parameter groups

Events

Tell us what you think

Return to the previous console experience

Density settings

Tables (2)

Any table tag

Find tables by table name

< 1 >

LabRetraining

LabScores

LabRetraining

Autopreview

Actions

Create item

Update table settings

Scan/Query items

ScanQuery

Table or index

LabRetraining

Filters

RunReset

CompletedRead capacity units consumed: 0.5

Items returned (1)

partition...sort\_key

CMPE\_272Group\_10

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Services

Search for services, features, blogs, docs, and more

[Option+S]

Oregon

Siddhant Satish Parmar

Lambda

Functions

LabLambdaFn

Throttle

Copy ARN

Actions

Function overview

Info

Code

Test

Monitor

Configuration

Aliases

Versions

Metrics

Logs

Traces

View logs in CloudWatch

View X-Ray traces in ServiceLens

View Lambda Insights

View profiles in CodeGuru

CloudWatch Logs Insights

Info

Lambda logs all requests handled by your function and automatically stores logs generated by your code through Amazon CloudWatch Logs. To validate your code, instrument it with custom logging statements. The following tables list the most recent and most expensive function invocations across all function activity. To view logs for a specific function version or alias, visit the Monitor section at that level.

1h3h12h1d3d1wCustom

Refresh

Dropdown

Add to dashboard

Recent invocations

Table with 8 columns: #, Timestamp, RequestID, LogStream, DurationInMS, BilledDurationIn..., MemorySetInMB, MemoryUsedInMB

Services

[Option +]
 Siddhant Satish Parmar ▾

### CloudWatch ×

- Favorites ▶
- Dashboards
- Alarms ⚠️ 4 🟢 12 🔴 0
- Logs
  - Log groups
  - Logs Insights
- Metrics
- X-Ray traces
- Events
- Application monitoring
- Insights
- Settings
- Getting Started

▶ 2022-05-01T20:39:23.557-07:00	2022-05-02T03:39:23.557Z 7957aeaa-aee7-4ef7-8a94-efc5a1b24111 INFO INSERT
▶ 2022-05-01T20:39:23.658-07:00	END RequestId: 7957aeaa-aee7-4ef7-8a94-efc5a1b24111
▶ 2022-05-01T20:39:23.658-07:00	REPORT RequestId: 7957aeaa-aee7-4ef7-8a94-efc5a1b24111 Duration: 101.41 ms Billed Duration: 102 ms Memory Size: 128 MB Max Memory Used: 7...
▶ 2022-05-01T20:40:21.393-07:00	START RequestId: b2ce2f3a-32b3-4237-85f3-135b52dc7475 Version: SLATEST
▶ 2022-05-01T20:40:21.395-07:00	2022-05-02T03:40:21.395Z b2ce2f3a-32b3-4237-85f3-135b52dc7475 INFO Stream record: { "eventID": "89428b568b5437fa7952f74aa809e55f", "event...
▶ 2022-05-01T20:40:21.395-07:00	2022-05-02T03:40:21.395Z b2ce2f3a-32b3-4237-85f3-135b52dc7475 INFO REMOVE
▶ 2022-05-01T20:40:21.395-07:00	END RequestId: b2ce2f3a-32b3-4237-85f3-135b52dc7475
▶ 2022-05-01T20:40:21.395-07:00	REPORT RequestId: b2ce2f3a-32b3-4237-85f3-135b52dc7475 Duration: 1.15 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 78 MB
▶ 2022-05-01T20:41:17.166-07:00	START RequestId: d06bd3ca-4cc1-43a3-88f6-207961d8d5ac Version: SLATEST
▼ 2022-05-01T20:41:17.168-07:00	2022-05-02T03:41:17.168Z d06bd3ca-4cc1-43a3-88f6-207961d8d5ac INFO Stream record: { "eventID": "3d39a173f4da9cb3f2abaf4d775413a9", "event...
	2022-05-02T03:41:17.168Z d06bd3ca-4cc1-43a3-88f6-207961d8d5ac INFO Stream record: <pre>{   "eventID": "3d39a173f4da9cb3f2abaf4d775413a9",   "eventName": "INSERT",   "eventVersion": "1.1",   "eventSource": "aws:dynamodb",   "awsRegion": "us-west-2",   "dynamodb": {     "ApproximateCreationDateTime": 1651462877,     "Keys": {       "partition_key": {         "S": "CMPE_272"       },       "sort_key": {         "S": "Group_10"       }     },     "NewImage": {       "partition_key": {         "S": "CMPE_272"       },       "sort_key": {         "S": "Group_10"       }     },     "SequenceNumber": "501800000000016310877580",     "SizeBytes": 74,     "StreamViewType": "NEW_IMAGE"   },   "eventSourceARN": "arn:aws:dynamodb:us-west-2:941854147474:table/LabScores/stream/2022-05-02T02:55:09.968" }</pre>
▶ 2022-05-01T20:41:17.168-07:00	2022-05-02T03:41:17.168Z d06bd3ca-4cc1-43a3-88f6-207961d8d5ac INFO INSERT
▶ 2022-05-01T20:41:17.259-07:00	END RequestId: d06bd3ca-4cc1-43a3-88f6-207961d8d5ac
▶ 2022-05-01T20:41:17.259-07:00	REPORT RequestId: d06bd3ca-4cc1-43a3-88f6-207961d8d5ac Duration: 91.33 ms Billed Duration: 92 ms Memory Size: 128 MB Max Memory Used: 78 MB

No newer events at this moment. Auto retry paused. [Resume](#)

**B) List of items in Scoring table and Retraining table:-**

- **List of items in LabScores table**

aws

Services

Search for services, features, blogs, docs, and more

Open in new tab

Alerts

Help

Oregon

Siddhant Satish Parmar

DynamoDB

X

Dashboard

Tables

Update settings

Explore Items

PartiQL editor

Backups

Exports to S3

Reserved capacity

▼ DAX

Clusters

Subnet groups

Parameter groups

Events

Tell us what you think

Return to the previous console experience

Density settings

DynamoDB

Items

LabScores

Tables (2)

X

Any table tag

▼

Find tables by table name

Q

<

1

>

⊗

☐ LabRetraining

☒ LabScores

LabScores

Autopreview

⌂

Actions

Create item

Update table settings

▼ Scan/Query items

Scan

Query

Table or index

LabScores

▼

▶ Filters

Run

Reset

✔ Completed

Read capacity units consumed: 0.5

Items returned (4)

<

1

>

⊗

⌵

<input type="checkbox"/>	partition...	sort_key	Class	Features	Label	Probability
<input type="checkbox"/>	key7	key8	1	Anirudh, 23, Male	False	0.8
<input type="checkbox"/>	key5	key6	0	Roger, 22, Male	True	0.7
<input type="checkbox"/>	key3	key4	1	Kumar, 24, Male	False	0.4
<input type="checkbox"/>	key1	key2	0	Siddhant, 23, Male	True	0.5

Feedback

Looking for language selection? Find it in the new Unified Settings

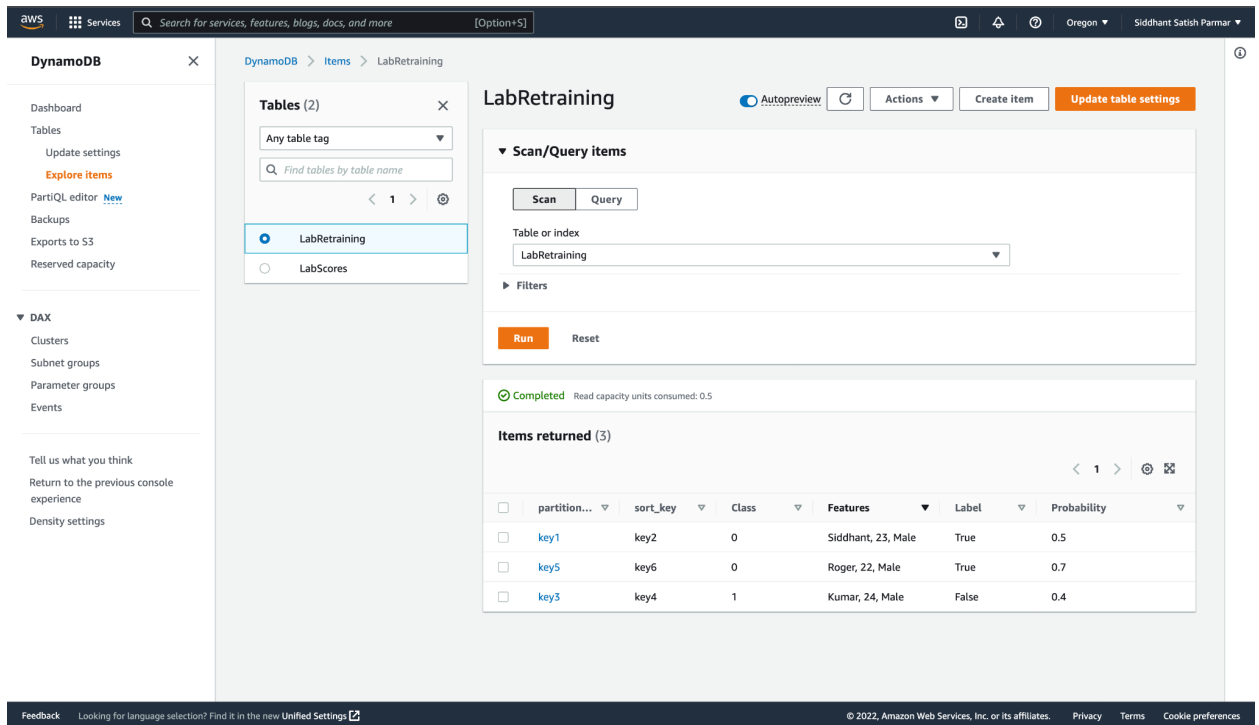
© 2022, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

- List of items in LabRetraining table



The screenshot shows the AWS Management Console interface for the LabRetraining table in DynamoDB. The left sidebar contains navigation options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Reserved capacity, DAX, Clusters, Subnet groups, Parameter groups, Events, and Density settings. The main content area displays the LabRetraining table with a 'Scan/Query items' section. The 'Scan/Query items' section shows a completed scan with 3 items returned. The items are listed in a table with columns: partition..., sort\_key, Class, Features, Label, and Probability.

partition...	sort_key	Class	Features	Label	Probability
key1	key2	0	Siddhant, 23, Male	True	0.5
key5	key6	0	Roger, 22, Male	True	0.7
key3	key4	1	Kumar, 24, Male	False	0.4

## 6. If you made any extensions to the logic of the Lambda function for selecting retraining samples, explain your extensions and show the code.

The changes we made to the Lambda function was that we extracted the age of the person from the “Features” and stored it in a variable named “age”. Now while writing records to the LabRetraining table we will check if the person is above the age of 21, then only the records will be written to the LabRetraining table.

**Code for the LabLambdaFn is shown below:**

```

index.js
1 var AWS = require('aws-sdk');
2 AWS.config.update({region: 'us-west-2'});
3 const docClient = new AWS.DynamoDB.DocumentClient();
4
5 async function createItem(params){
6   try {
7     await docClient.put(params).promise();
8   } catch (err) {
9     return err;
10  }
11 }
12
13 exports.handler = (event, context, callback) => {
14
15   event.Records.forEach( (record) => {
16     var record_str = JSON.stringify(record, null, 2);
17     console.log('Stream record: ', record_str);
18
19     if (record.eventName === 'INSERT') {
20
21       var params = {
22         TableName: 'LabRetraining',
23         Item: {
24           "partition_key": record.dynamodb.NewImage.partition_key.S,
25           "sort_key": record.dynamodb.NewImage.sort_key.S,
26           "Class": record.dynamodb.NewImage.Class.S,
27           "Probability": record.dynamodb.NewImage.Probability.S,
28           "Label": record.dynamodb.NewImage.Label.S,
29           "Features": record.dynamodb.NewImage.Features.S
30         }
31       };
32       var int_class = parseInt(record.dynamodb.NewImage.Class.S);
33       var float_probability = parseFloat(record.dynamodb.NewImage.Probability.S);
34       var features = record.dynamodb.NewImage.Features.S;
35       var feats = features.split(", ");
36       var age = parseInt(feats[1]);
37       var gap = 0;
38
39       if ( int_class == 0 )
40       {
41         gap = float_probability;
42       }
43       else
44       {
45         gap = 1 - float_probability;
46       }
47       if ( gap > 0.25 && age > 21)
48       {
49         createItem(params)
50       }
51     }
52   }),
53
54   callback(null, `Successfully processed ${event.Records.length} records.`);
55 }
56

```