

Module 13

Implementing a Serverless Architecture with AWS Lambda

COS20019 – CLOUD ARCHITECTURE COMPUTING

Lab 09

Tran Thanh Minh

103809048

Task 1: Create a Lambda function

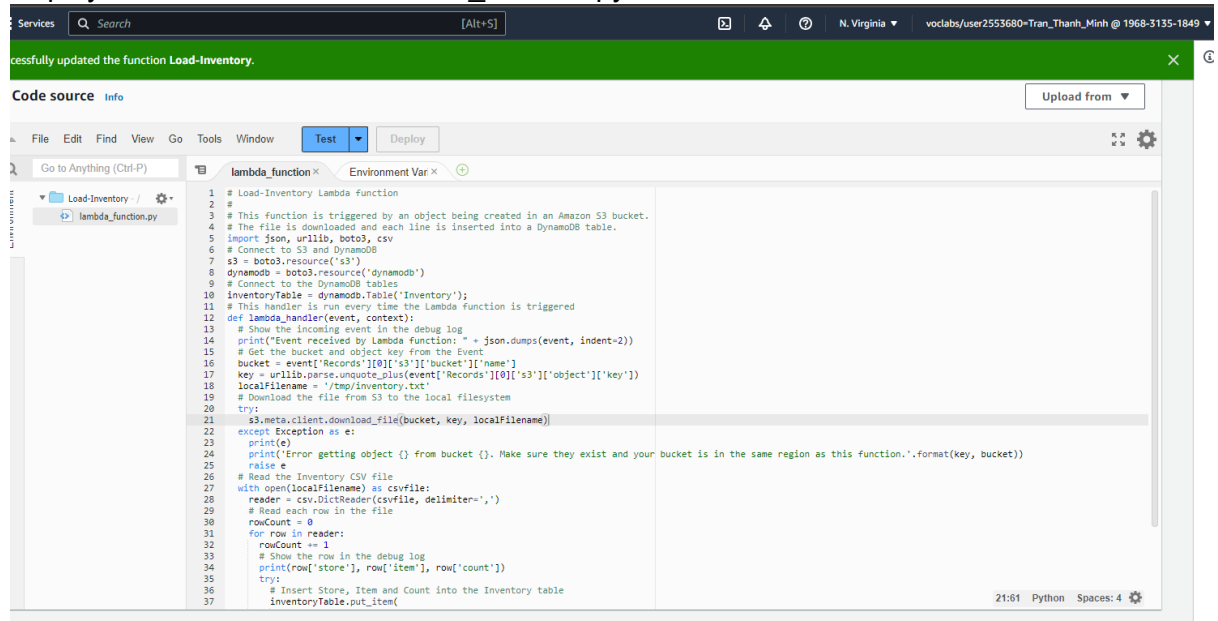
1. Configure Lambda

The screenshot shows the AWS Lambda console configuration page for a new function. The top navigation bar includes the 'Services' menu, a search bar, and the user's profile information (N. Virginia, voclabs/user2553680-Tran_Thanh_Minh @ 1968-3135-1849). Below the navigation bar, there are three tabs: 'Start with a simple Hello World example.', 'Build a Lambda application from sample code and configuration presets for common use cases.', and 'Select a container image to deploy for your function.' The 'Basic information' section is currently active. It contains the following fields and options:

- Function name:** A text input field containing 'Load-Inventory'. Below the field, a note states: 'Enter a name that describes the purpose of your function. Use only letters, numbers, hyphens, or underscores with no spaces.'
- Runtime:** A dropdown menu set to 'Python 3.7'. A note states: 'Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.'
- Architecture:** Two radio button options: 'x86_64' (selected) and 'arm64'. A note states: 'Choose the instruction set architecture you want for your function code.'
- Permissions:** A section with a note: 'By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.'
- Change default execution role:** A section with three radio button options: 'Create a new role with basic Lambda permissions', 'Use an existing role' (selected), and 'Create a new role from AWS policy templates'. A note states: 'Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.'
- Existing role:** A dropdown menu set to 'Lambda-Load-Inventory-Role'. A note states: 'Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.'

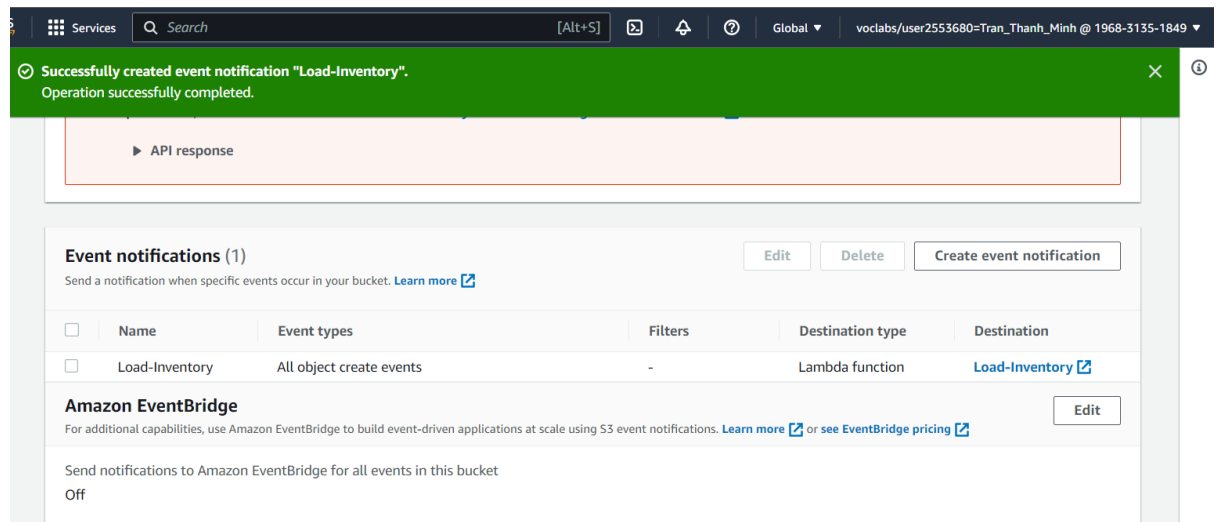
At the bottom of the 'Existing role' section, there is a link: 'View the Lambda-Load-Inventory-Role role on the IAM console.'

2. Deploy new code source to lambda_function.py



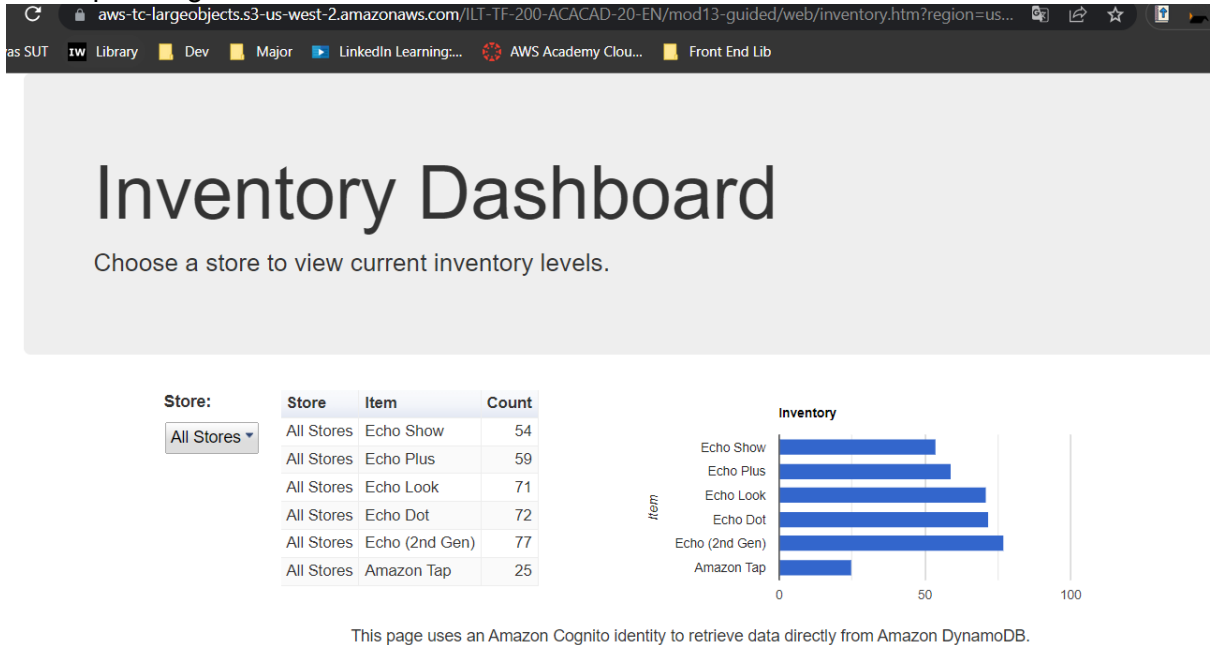
Task 2: Configure an Amazon S3 event

1. Create the event notifications for the S3 Event



Task 3: Testing the loading process

1. After uploading the CSV files



2. Items are displayed

The screenshot shows the AWS DynamoDB console interface. The left sidebar shows the 'Tables (1)' section with a filter for 'Any tag key' and a search bar. The main area displays the 'Inventory' table. A green status bar indicates 'Completed. Read capacity units consumed: 0.5'. Below this, the 'Items returned (30)' section shows a table of items. The table has columns for 'Store', 'Item', and 'Count'. The items are listed in a table with checkboxes for each row.

Store	Item	Count
Calcutta	Amazon Tap	15
Calcutta	Echo (2nd Gen)	0
Calcutta	Echo Dot	7
Calcutta	Echo Look	3
Calcutta	Echo Plus	16
Calcutta	Echo Show	14
Shanghai	Amazon Tap	0
Shanghai	Echo (2nd Gen)	13

Task 4: Configuring notifications

1. Create simple notification service

The screenshot shows the Amazon SNS console interface. At the top, a green banner indicates that the 'Topic NoStock' was created successfully. Below this, the breadcrumb navigation shows 'Amazon SNS > Topics > NoStock'. The main heading is 'NoStock', with 'Edit', 'Delete', and 'Publish message' buttons to its right. A 'Details' section contains the following information:

Details	
Name	NoStock
Display name	-
ARN	arn:aws:sns:us-east-1:196831351849:NoStock
Topic owner	196831351849
Type	Standard

2. Configure subscription for protocol email

The screenshot shows the 'Create subscription' page in the Amazon SNS console. At the top, a warning banner mentions 'Important changes for sending text messages (SMS) to US destinations'. The breadcrumb navigation is 'Amazon SNS > Subscriptions > Create subscription'. The main heading is 'Create subscription'. The 'Details' section contains the following fields:

- Topic ARN:** A dropdown menu showing 'arn:aws:sns:us-east-1:196831351849:NoStock'.
- Protocol:** A dropdown menu showing 'Email'.
- Endpoint:** A text input field containing '103809048@student.swin.edu.au'.

Below the input fields, a blue box contains the text: 'After your subscription is created, you must confirm it. [Info](#)'. At the bottom, there are two optional policy sections:

- Subscription filter policy - optional** [Info](#): This policy filters the messages that a subscriber receives.
- Redrive policy (dead-letter queue) - optional** [Info](#): Send undeliverable messages to a dead-letter queue.

Task 5: Creating a Lambda function to send notification

1. Create new lambda function

Services Search [Alt+S] N. Virginia voclabs/user2553680=Tran_Thanh_Minh @ 1968-3135-1849

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ Author from scratch
Start with a simple Hello World example.

☐ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

☐ Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

2. Deploy the code to code source in Lambda

Services Search [Alt+S] N. Virginia voclabs/user2553680=Tran_Thanh_Minh @ 1968-3135-1849

Successfully updated the function **Check-Stock**.

Code Test Monitor Configuration Aliases Versions

Code source

Upload from ▼

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

▼ Check-Stock - /
↳ lambda_function.py

```
1 # Stock Check Lambda function
2 #
3 # This function is triggered when values are inserted into the Inventory DynamoDB table.
4 # Inventory counts are checked and if an item is out of stock, a notification is sent to an SNS Topic.
5 import json, boto3
6 # This handler is run every time the Lambda function is triggered
7 def lambda_handler(event, context):
8     # Show the incoming event in the debug log
9     print("Event received by Lambda function: " + json.dumps(event, indent=2))
10    # For each Inventory item added, check if the count is zero
11    for record in event['Records']:
12        newImage = record['dynamodb'].get('NewImage', None)
13        if newImage:
14            count = int(record['dynamodb']['NewImage']['Count']['N'])
15            if count == 0:
16                store = record['dynamodb']['NewImage']['Store']['S']
17                item = record['dynamodb']['NewImage']['Item']['S']
18                # Construct message to be sent
19                message = store + ' is out of stock of ' + item
20                print(message)
21                # Connect to SNS
22                sns = boto3.client('sns')
23                alertTopic = 'NoStock'
24                snsTopicArn = [t['TopicArn'] for t in sns.list_topics()['Topics']]
25                # Send message to SNS
26                sns.publish(
27                    TopicArn=snsTopicArn,
28                    Message=message,
29                    Subject='Inventory Alert!',
30                    MessageStructure='raw'
31                )
32            # Finished!
33            return 'Successfully processed {} records.'.format(len(event['Records']))
```

3. Add the trigger for the lambda function

Add trigger

Trigger configuration [info](#)

DynamoDB
aws database event-source-mapping nosql polling

DynamoDB table
Choose or enter the ARN of a DynamoDB table.

Q X C

Use: "arn:aws:dynamodb:us-east-1:196831351849:table/Inventory"

Inventory
testing (recommended).

Batch size
The number of records in each batch to send to the function.

Starting position
The position in the stream to start reading from. For more information, see [ShardIteratorType](#) in the Amazon Kinesis API Reference.

Batch window - optional
The maximum amount of time to gather records before invoking the function, in seconds.

► **Additional settings**

In order to read from the DynamoDB trigger, your execution role must have proper permissions.

After submitting my grade is:

Module 13 Guided Lab - Implementing a Serverless Architecture with AWS Lambda

Due No Due Date Points 100 Submitting an external tool

Submit Details AWS Start Lab End Lab 00:00:00 Instructions **Grades** Actions

Files ☐ README ☒ Terminal ☒ Source ☐

EN_US

Submitting your work

52. At the top of these instructions, choose **Submit** to record your progress and when prompted, choose **Yes**.

53. If the results don't display after a couple of minutes, return to the top of these instructions and choose **Grades**.

Tip: You can submit your work multiple times. After you change your work, choose **Submit** again. Your last submission is what will be recorded for this lab.

54. To find detailed feedback on your work, choose **Details** followed by ► **View Submission Report**.

Lab complete 🎓

🎉 Congratulations! You have completed the lab.

55. Choose **End Lab** at the top of this page, and then select **Yes** to confirm that you want to end the lab.

A panel indicates that *DELETE has been initiated...* You may close this message box now.

56. Select the X in the top right corner to close the panel.

Task	Score
Total score	40/40
[Task 1A] Load-Inventory Function Created	5/5
[Task 1B] Load-Inventory Execution Role	5/5
[Task 2A] Bucket Created	5/5
[Task 2B] Bucket Event	5/5
[Task 3] Database Table Data	5/5
[Task 4] SNS Topic Created	5/5
[Task 5A] Check-Stock Function Created	5/5
[Task 5B] Check-Stock has Trigger	5/5

Submit failed.
The lab must be started to Submit.