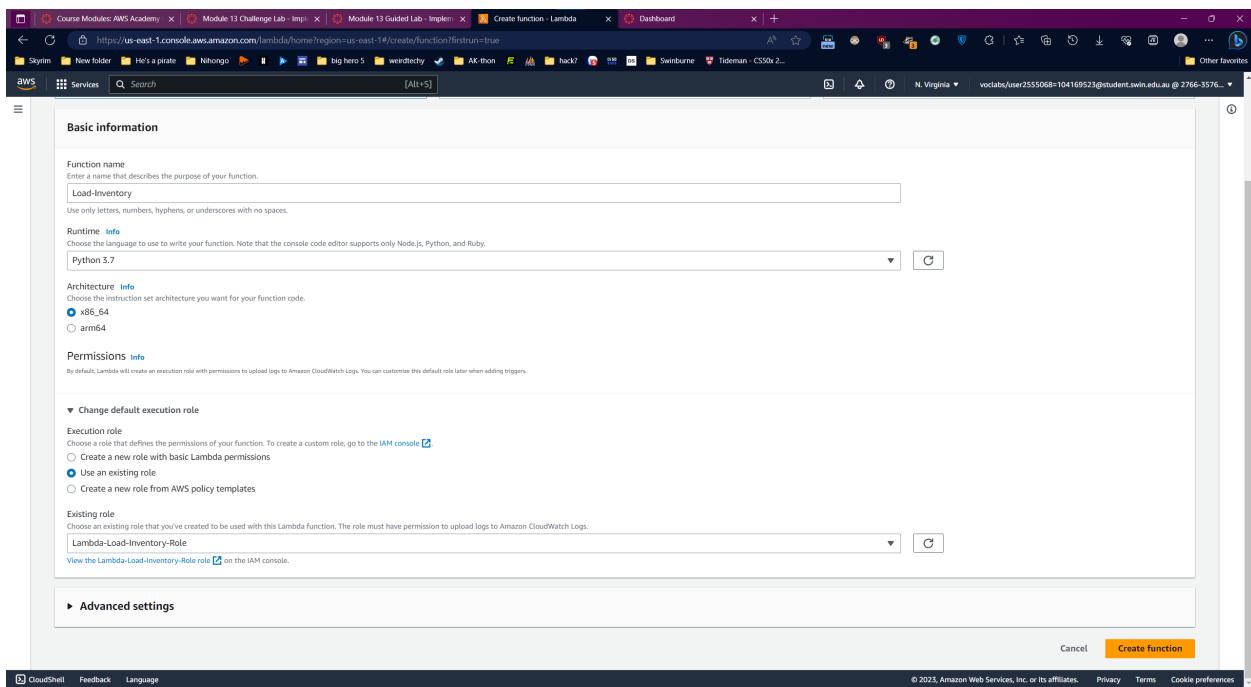


# Module 13 – Guided Lab 2: Implementing a Serverless Architecture on AWS

## Task 1: Creating a Lambda function to load data

### Create function



## Replace code

The screenshot shows the AWS Lambda console interface. The top navigation bar has tabs for Course Modules, AWS Academy, Module 13 Challenge Lab - Impl., Module 13 Guided Lab - Impl., Load Inventory - Lambda, Dashboard, and Other favorites. Below the navigation is a toolbar with AWS, Services, Search, and a dropdown for N. Virginia. The main area shows a success message: "Successfully updated the function Load-Inventory." Below this is a navigation bar with Code, Test, Monitor, Configuration, Aliases, and Versions. The Code source tab is active, displaying the code editor for 'lambda\_function.py'. The code implements a Lambda function that handles S3 events, reads CSV files, and inserts data into a DynamoDB table. The 'Test' tab is also visible in the navigation bar.

## Task 2: Configuring an Amazon S3 event

## Create bucket

The screenshot shows the AWS S3 Management Console with a green header bar indicating the bucket 'inventory-69' was successfully created. The main content area displays an 'Account snapshot' with a storage lens link and a 'Buckets (1) info' section. A table lists the single bucket 'inventory-69' with details: Name (inventory-69), AWS Region (US East (N. Virginia) us-east-1), Access (Bucket and objects not public), and Creation date (July 17, 2023, 22:22:54 (UTC+07:00)).

Name	AWS Region	Access	Creation date
inventory-69	US East (N. Virginia) us-east-1	Bucket and objects not public	July 17, 2023, 22:22:54 (UTC+07:00)

## Property > Event Notifications

The screenshot shows the AWS S3 console interface for the 'inventory-69' bucket. The top navigation bar includes tabs for Course Modules, AWS Academy, Module 13 Challenge Lab - Impl, Module 13 Guided Lab - Impl, Load-Inventory - Lambda, Dashboard, and other browser tabs like 'Load-Inventory - Lambda' and 'inventory-69 - S3 bucket'. The main content area is titled 'Event Notifications (0)' and contains sections for Server access logging, AWS CloudTrail data events, Amazon EventBridge, and Transfer acceleration.

- Server access logging:** A section for log requests to your bucket, currently set to 'Disabled'.
- AWS CloudTrail data events:** A section for configuring CloudTrail data events to log Amazon S3 object-level API operations. It displays a message: "You don't have permission to get AWS CloudTrail data events details. You or your AWS administrator must update your IAM permissions to allow cloudtrail:DescribeTrails. After you obtain the necessary permission, choose Refresh. Learn more about Identity and access management in Amazon S3." A 'Refresh' button is present.
- Amazon EventBridge:** A section for using Amazon EventBridge to build event-driven applications at scale using S3 event notifications. It shows 'Send notifications to Amazon EventBridge for all events in this bucket' is set to 'Off'.
- Transfer acceleration:** A section for CloudFront distribution settings.

At the bottom, the navigation bar includes CloudShell, Feedback, Language, and links to © 2023, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

**Create event notification** (info)

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

**General configuration**

Event name: Load-Inventory  
Event name can contain up to 255 characters.

Prefix - optional: /images/  
Limit the notifications to objects with key starting with specified characters.

Suffix - optional: jpg  
Limit the notifications to objects with key ending with specified characters.

**Event types**  
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

**Object creation**

All object create events  
s3:ObjectCreated:<sup>\*</sup>

Put  
s3:ObjectCreated:Put

Post  
s3:ObjectCreated:Post

Copy  
s3:ObjectCreated:Copy

Multipart upload completed  
s3:ObjectCreated:CompleteMultipartUpload

**Object removal**

Lifecycle transition events

- Lifecycle transition events
  - s3:LifeCycleTransition
  - All lifecycle expiration events
    - s3:LifeCycleExpiration\*
  - Delete marker added by Lifecycle for a versioned object
    - s3:LifeCycleExpirationDeleteMarkerCreated

Intelligent-Tiering

- Intelligent-Tiering archive events
  - s3:IntelligentTiering

**Destination**

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

Choose a destination to publish the event. [Learn more](#)

Lambda function Run a Lambda function script based on S3 events.

SNS topic Fanout messages to systems for parallel processing or directly to people.

SQS queue Send notifications to an SQS queue to be read by a server.

Specify Lambda function

Choose from your Lambda functions

Enter Lambda function ARN

Lambda function

Load-Inventory

**Save changes**

# Task 3: Testing the loading process

## Add file to S3

Course Modules: AWS Academy > Module 13 Challenge Lab - Implement > Module 13 Guided Lab - Implement > Load-Inventory - Lambda > S3 Management Console > Dashboard

Amazon S3 > Buckets > inventory-69 > Upload

**Upload** [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files or Add folder

**Files and folders (0)**

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
No files or folders			

You have not chosen any files or folders to upload.

**Destination**

Destination

s3://inventory-69

▶ Destination details

Bucket settings that impact new objects stored in the specified destination.

**Permissions**

Grant public access and access to other AWS accounts.

**Properties**

Specify storage class, encryption settings, tags, and more.

File name: tabc.voca

**Open**

## Get Dashboard Link

The screenshot shows the AWS Academy dashboard with the following details:

- Assignment:** Module 13 Guided Lab - Implementing a Serverless Architecture with AWS Lambda
- Due:** No Due Date
- Points:** 100
- Status:** Submitting an external tool

A modal window titled "Credentials" is open, displaying the following information:

- Cloud Access:**
  - AWS CLI: Show
  - Cloud Labs: Remaining session time: 02:50:14 (171 minutes)  
Session started at: 2023-07-17T06:18:31-0700  
Session to end at: 2023-07-17T11:18:31-0700
  - Accumulated lab time: 00:09:00 (9 minutes)
  - No running instance
- SSH key:** Show, Download PEM, Download PPK
- AWS SSO:** Download URL
- SecretKey:** jK80/Ss1F2m+f++oH7UP0cJ9EK5/LuBoSHmHboto
- Dashboard:** <https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/117-TF-200-ACACAD-20-EN/mod13-guided/web/inventory.htm?region=us-east-1&poolId=us-east-1:6aa0550e-6a8c-454d-abb3-ecbc2562fc42>
- IdentityPoolId:** us-east-1:6aa0550e-6a8c-454d-abb3-ecbc2562fc42
- AccessKey:** AKIAUUA200XETOJRSFWVM

## Result

The screenshot shows the "Inventory Dashboard" with the following interface:

**Choose a store to view current inventory levels.**

**Store:** All Stores

Store	Item	Count
All Stores	Echo Show	72
All Stores	Echo Plus	59
All Stores	Echo Look	81
All Stores	Echo Dot	84
All Stores	Echo (2nd Gen)	96
All Stores	Amazon Tap	40

**Inventory**

Item	Count
Echo Show	72
Echo Plus	59
Echo Look	81
Echo Dot	84
Echo (2nd Gen)	96
Amazon Tap	40

This page uses an Amazon Cognito identity to retrieve data directly from Amazon DynamoDB.

## Go to DynamoDB > Tables > Inventory > Explore table items

The screenshot shows the AWS DynamoDB service dashboard. On the left, a sidebar menu for 'DynamoDB' is open, showing options like 'Tables', 'Update settings', 'Explore items', and 'Exports to S3'. The main content area is titled 'Tables (1) Info' and displays a table with one row for the 'Inventory' table. The table columns include Name, Status, Partition key, Sort key, Indexes, Deletion protection, Read capacity mode, Write capacity mode, Total size, and Table class. The 'Inventory' table has an 'Active' status, a 'Store (\$)' partition key, and an 'Item (\$)' sort key. It uses 'Provisioned (5)' for both read and write capacity modes, with a total size of 0 bytes and a Standard table class.

The screenshot shows the 'Inventory' table details page. At the top, there's a navigation bar with tabs for 'Overview', 'Indexes', 'Monitor', 'Global tables', 'Backups', 'Exports and streams', and 'Additional settings'. A prominent message at the top says 'Protect your DynamoDB table from accidental writes and deletes' with a note about Point-in-time recovery (PITR). Below this, the 'General information' section provides details such as the partition key ('Store String'), sort key ('Item (String)'), capacity mode ('Provisioned'), and table status ('Active'). The 'Additional info' section includes table class ('DynamoDB Standard'), indexes ('0 global, 0 local'), dynamoDB stream ('On'), and time-to-live (TTL). The 'Amazon Resource Name (ARN)' is also listed. At the bottom, an 'Items summary' section shows '0 items' and a 'Get live item count' button.

The screenshot shows the AWS DynamoDB console with the 'Inventory' table selected. The table has three items:

Store	Item	Count
Calcutta	Amazon Tap	15
Calcutta	Echo (2nd Gen)	0
Calcutta	Echo Dot	7

## Task 4: Configuring notifications

Go to topics > create topic

The screenshot shows the AWS SNS console with the 'Topics' page. There are no topics listed yet.

Dashboard Course Modules: AWS Academy Module 13 Challenge Lab - Implement Module 13 Guided Lab - Implement Create topic | Amazon SNS Inventory System

Amazon SNS > Topics > Create topic

### Create topic

**Details**

Type [Info](#) Topic type cannot be modified after topic is created.

Standard

FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 500 publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, emails, mobile application endpoints

Name NoStock Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (\_).

Display name - optional [Info](#) To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

My Topic Maximum 100 characters.

▶ Encryption - optional Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

▶ Access policy - optional [Info](#) This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

▶ Data protection policy - optional [Info](#) This policy defines which sensitive data to monitor and to prevent from being exchanged via your topic.

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Create Subscription

Dashboard Course Modules: AWS Academy Module 13 Challenge Lab - Implement Module 13 Guided Lab - Implement Details | Topics | Amazon SNS Inventory System

Amazon SNS > Topics > NoStock

Topic NoStock created successfully. You can create subscriptions and send messages to them from this topic.

Publish message

NoStock

Edit Delete Publish message

**Details**

Name	NoStock	Display name	
ARN	arn:aws:sns:us-east-1:276635760934:NoStock	Topic owner	276635760934
Type	Standard		

[Subscriptions](#) [Access policy](#) [Data protection policy](#) [Delivery policy \(HTTP/S\)](#) [Delivery status logging](#) [Encryption](#) [Tags](#) [Integrations](#)

**Subscriptions (0)**

ID	Endpoint	Status	Protocol
No subscriptions found. You don't have any subscriptions to this topic.			
<a href="#">Create subscription</a>			

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'Create subscription' page in the AWS SNS console. The 'Topic ARN' field contains 'arn:aws:sns:us-east-1:276635760934:NoStock'. The 'Protocol' dropdown is set to 'Email'. The 'Endpoint' field contains '104169523@student.swin.edu.au'. A note at the bottom states: 'After your subscription is created, you must confirm it.' Below the form, there are sections for 'Subscription filter policy - optional' and 'Redrive policy (dead-letter queue) - optional'. At the bottom right are 'Cancel' and 'Create subscription' buttons.

## Confirmation in email:

The screenshot shows an Outlook inbox with several emails from 'AWS Notifications <no-reply@sns.amazonaws.com>'. One email is highlighted, titled 'AWS Notification - Subscription Confirmation'. It contains the following text:  
You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:276635760934:NoStock  
To confirm this subscription, click or visit the link below (if this was in error no action is necessary):  
[Confirm subscription](https://sns.us-east-1.amazonaws.com/confirm_sub?SubscriptionArn=arn:aws:sns:us-east-1:276635760934:NoStock&Signature=qQAO1vdW%2Bk8Zj2RnPdPjPzv%3D&actSwt=true)  
Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](mailto:sns-opt-out).

# Task 5: Creating a Lambda function to send notifications

## Create new Lambda function

The screenshot shows the 'Create function - Lambda' wizard. In the 'Basic information' section, the function name is 'Check-Stock' and the runtime is 'Python 3.7'. Other settings include 'x86\_64' architecture and an existing role named 'Lambda-Check-Stock-Rule'. The 'Advanced settings' section is collapsed.

## Change code

The screenshot shows the 'Check-Stock - Lambda' function configuration. The 'Code source' tab is selected, displaying the Python code for the 'lambda\_function' handler:

```
# Stock Check Lambda Function
# This function is triggered when values are inserted into the Inventory DynamoDB table.
# An inventory item is checked and if an item is out of stock, a notification is sent to an SNS Topic.
import json, boto3
# This handler is run every time the Lambda function is triggered
def lambda_handler(event, context):
    # Show the incoming event in the debug log
    print("Event received by Lambda Function: ", json.dumps(event, indent=2))
    # If the count is zero, then check if the count is zero
    for record in event['Records']:
        dynamodb = boto3.resource('dynamodb')
        # Get the record
        item = dynamodb.Table('NewImage').get_item(
            Key={
                'Name': record['NewImage']['Name']
            }
        )
        # Get the count
        count = int(record['NewImage']['Count'])
        if count == 0:
            # Store the record
            store = record['NewImage']
            # Construct message to be sent
            message = store['Item'][0]['S']
            # Connect to SNS
            snsTopicArn = "arn:aws:sns:us-east-1:123456789012:AlertTopic"
            snsTopicArn = [t['TopicArn'] for t in sns.list_topics()[1]['Topics']][0]
            # Send message to SNS
            sns.Topic.publish(
                TopicArn=snsTopicArn,
                Message=message,
                Subject='Inventory Alert!',
                MessageStructure='raw'
            )
    # Finished!
return "Successfully processed " + str(len(event['Records']))
```

## Add trigger

The screenshot shows the 'Add trigger' configuration page for a DynamoDB trigger. The 'Trigger configuration' section is open, showing the following details:

- DynamoDB**: Selected as the trigger type.
- DynamoDB table**: 'Inventory' is selected from the dropdown.
- Batch size**: Set to 100.
- Starting position**: Set to 'Latest'.
- Batch window - optional**: Set to 1 second.
- Additional settings**: A note states that the execution role must have proper permissions to read from the DynamoDB trigger.

At the bottom right of the configuration panel are 'Cancel' and 'Add' buttons. The 'Add' button is highlighted in orange.

# Task 6: Testing the System

Upload another file

The screenshot shows the 'Upload' page in the AWS S3 Management Console. The 'Files and folders' section displays one item:

Name	Type	Size
labs.vocareum.com_we_b_255068_1795497.0,_ASNLB_public_scripts,_inventory-berlin.csv	text/csv	140.0 B

The 'Destination' section shows the destination as 's3://inventory-69'. The 'Permissions' and 'Properties' sections are collapsed. At the bottom right are 'Cancel' and 'Upload' buttons. The 'Upload' button is highlighted in orange.

Screenshot of a web browser showing an Inventory Dashboard. The dashboard displays current inventory levels for various items across different stores.

The page title is "Inventory System".

The main heading is "Inventory Dashboard". Below it, a sub-instruction says "Choose a store to view current inventory levels."

A dropdown menu labeled "Store:" shows "All Stores" selected. A table lists items and their counts:

Store	Item	Count
All Stores	Echo Show	72
All Stores	Echo Plus	59
All Stores	Echo Look	81
All Stores	Echo Dot	84
All Stores	Echo (2nd Gen)	96
All Stores	Amazon Tap	39
Atom	Echo Show	~72
Atom	Echo Plus	~59
Atom	Echo Look	~81
Atom	Echo Dot	~84
Atom	Echo (2nd Gen)	~96
Atom	Amazon Tap	~39

An inventory bar chart is displayed on the right side:

The chart has a scale from 0 to 100. The bars represent the count for each item across all stores.

A note at the bottom states: "This page uses an Amazon Cognito identity to retrieve data directly from Amazon DynamoDB."