

Implementing a Serverless Architecture with AWS Managed Services

Traditionally, applications run on servers. These can be physical ('bare metal') servers or virtual environments running on top of physical servers, but they still need servers to be purchased and provisioned, and for capacity to be managed. On the other hand, AWS Lambda can run serverless code without having to pre-allocate servers. Simply provide the code and define a trigger and the function can run whenever required – once per week or hundreds of times per second, and you only pay for what you use.

This lab demonstrates how to trigger an AWS Lambda function when a file is uploaded to Amazon S3. The file will be loaded into an Amazon DynamoDB table, and the data will be available for viewing on a Dashboard page that pulls the data directly from DynamoDB. The solution is **completely serverless, automatically scalable and incurs very little cost**.

The system **does not use Amazon EC2**. The system will automatically scale when it is used and incurs practically no cost when it is *not* being used (just a few cents for data storage).

In this lab you will:

- Use AWS managed services to implement a serverless architecture
- Trigger AWS Lambda functions from Amazon S3 and Amazon DynamoDB
- Configure Amazon SNS to send notifications

Scenario

Accessing the AWS Management Console

Task 1: Create a Lambda Function to Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 6: Configure an Amazon S3 Event

Task 7: Test the Loading Process

Task 8: Configure Notifications

Task 9: Test the Loading Process

Task 10: Configure Notifications

Task 11: Create a Lambda Function to Send Notifications

Task 12: Configure an Amazon S3 Event

Task 13: Test the Loading Process

Task 14: Configure Notifications

Task 15: Test the Loading Process

Task 16: Configure Notifications

Task 17: Test the Loading Process

Task 18: Configure Notifications

Task 19: Create a Lambda Function to Send Notifications

Task 20: Configure an Amazon S3 Event

Task 21: Test the Loading Process

Task 22: Configure Notifications

Task 23: Create a Lambda Function to Send Notifications

Task 24: Configure Notifications

Task 25: Create a Lambda Function to Send Notifications

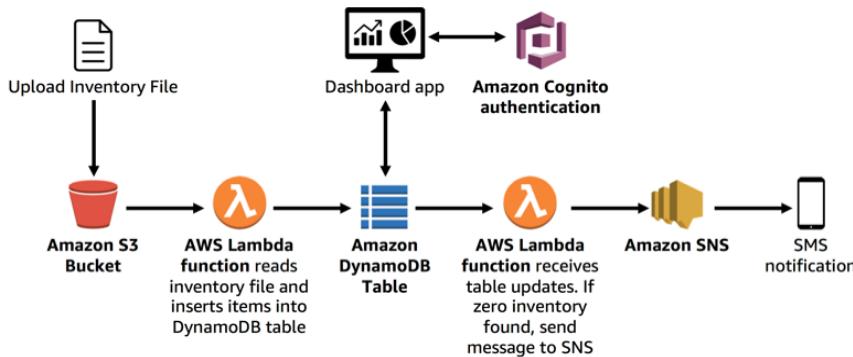
Task 26: Configure an Amazon S3 Event

Task 27: Test the Loading Process

Task 28: Configure Notifications

Task 29: Create a Lambda Function to Send Notifications

You are creating an inventory tracking system. Stores from around the world will upload an inventory file to Amazon S3 and your team wants to be able to view the inventory levels and send a notification when inventory levels are low.



The scenario workflow is:

- You will **upload** an *inventory file* to an Amazon S3 bucket
- This will **trigger an AWS Lambda function** that will read the file and insert items into an **Amazon DynamoDB table**
- A serverless, web-based Dashboard application will use Amazon Cognito to authenticate to AWS and then gain access to the DynamoDB table to display inventory levels
- Another AWS Lambda function will receive updates from the DynamoDB table and will send a message to an **Amazon Simple Notification Service (SNS) topic** when an inventory item is out-of-stock
- Amazon SNS will then **send an SMS or email notification to you** to request additional inventory

Duration

This lab takes approximately **20 minutes** to complete.

Accessing the AWS Management Console

- Click **Start Lab** to launch the lab.

Task 2: Configure an Amazon S3 Event

.....management

Task 3: Test the I ..

Task 4: Create a Lambda Function to Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

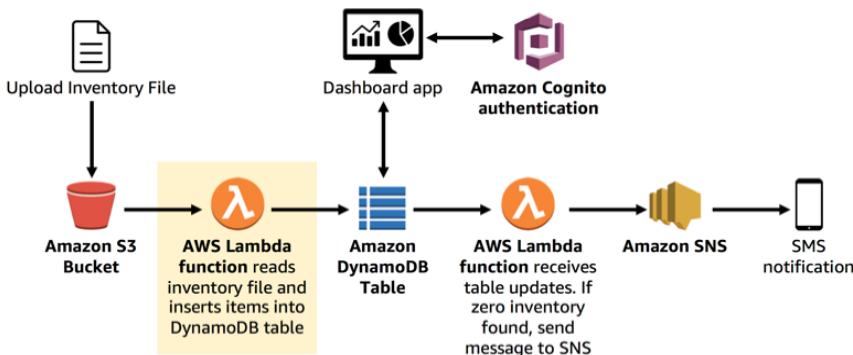
Task 5: Create a Lambda Function to Event

- Click **Open Console**
- Sign in to the AWS Management Console using the credentials shown to the left of these instructions.

⚠ Please do not change the Region during this lab.

Task 1: Create a Lambda Function to Load Data

In this task, you will create an **AWS Lambda function** that will process an inventory file. The Lambda function will read the file and insert information into an Amazon DynamoDB table.



1. In the **AWS Management Console**, on the **Services** menu, click **Lambda**.
2. Click **Create a function**

💡 Blueprints are code templates for writing Lambda functions.

Blueprints are provided for standard Lambda triggers such as creating Alexa skills and processing Amazon Kinesis Firehose streams. This lab provides you with a pre-written Lambda function, so you will **Author from scratch**.

3. Configure the following:

Task 3: Test the Loading Process

Task 4: Configure Notifications
Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 2: Configure an Amazon S3 Event
Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 2: Configure an Amazon S3 Event

- **Name:** Load-Inventory
 - **Runtime:** Python 3.7
 - **Role:** Choose an existing role
 - **Existing role:** Lambda-Load-Inventory-Role

This role gives execution permissions to the Lambda function so it can access Amazon S3 and Amazon DynamoDB.

4. Click **Create function**
 5. Scroll down to the **Function code** section, then delete all of the code that appears in the code editor.
 6. Copy and paste the following code into the **Function code** editor:

```
# Load-Inventory Lambda function
#
# This function is triggered by an object being created in an
Amazon S3 bucket.
# The file is downloaded and each line is inserted into a
# Connect to S3 and DynamoDB
s3 = boto3.resource('s3')
dynamodb = boto3.resource('dynamodb')

# Connect to the DynamoDB tables
inventoryTable = dynamodb.Table('Inventory');

# This handler is executed 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))

    # Get the bucket and object key from the Event
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']
['object']['key'])
    localFilename = '/tmp/inventory.txt'

    # Download the file from S3 to the local filesystem
    try:
        s3.meta.client.download_file(bucket, key, localFilename)
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure
they exist and your bucket is in the same region as this
function.'.format(key, bucket))
        raise e

    # Read the Inventory CSV file
```

```

with open(localFilename) as csvfile:
    reader = csv.DictReader(csvfile, delimiter=',',)

    # Read each row in the file
    rowCount = 0
    for row in reader:
        rowCount += 1

        # Show the row in the debug log
        print(row['store'], row['item'], row['count'])

    try:
        ...
    except Exception as e:
        print(e)
        print("Unable to insert data into DynamoDB"
              "table".format(e))

    # Finished!
    return "%d counts inserted" % rowCount

```

Examine the code. It is performing the following steps:

- Download the file from Amazon S3 that triggered the event
- Loop through each line in the file
- Insert the data into the DynamoDB *Inventory* table

7. Click **Save** at the top of the page.

Next, you will configure Amazon S3 to trigger the Lambda function when a file is uploaded.

Task 2: Configure an Amazon S3 Event

Stores from around the world will provide inventory files to load into the inventory tracking system. Rather than uploading files via FTP, the stores can upload directly to Amazon S3. This can be done via a web page, a script or as part of a program. Once a file is received, the AWS Lambda function will be triggered and it will load the inventory into a DynamoDB table.

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Send Notifications

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 3. TEST THE LOADING PROCESS

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 3. TEST THE LOADING PROCESS

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 3. TEST THE LOADING PROCESS

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

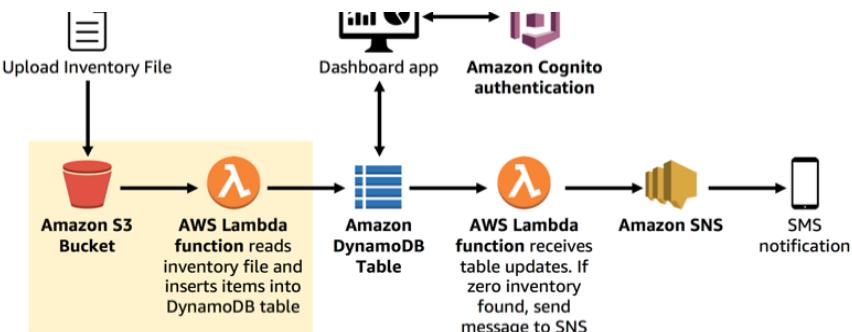
Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process





In this task you will create an Amazon S3 bucket and configure it to trigger the Lambda function.

8. On the **Services** menu, click **S3**.

9. Click **+ Create bucket**

Each bucket must have a unique name, so you will add a random number to the bucket name. For example: *inventory-123*

10. For **Bucket name** enter: **inventory-123** (Replacing 123 with a random number)

11. Click **Create**

💡 If you receive an error stating **The requested bucket name is not available**, then click the first **Edit** link, change the bucket name and try again until it is accepted.

You will now configure the bucket to automatically trigger the Lambda function whenever a file is uploaded.

12. Click the name of your *inventory-bucket*.

13. Click the **Properties** tab.

14. Scroll down to **Advanced settings**, then click **Events**.

You will configure an event to trigger when an object is created in the S3 bucket.

15. Click **+ Add notification** then configure:

- **Name:** **Load-Inventory**
- **Events:** All object create events
- **Send to:** *Lambda Function*

Task 4: Configure Notifications

EVENT

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications
Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Event

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

- **Lambda: Load-Inventory**

- Click **Save**

This will tell Amazon S3 to trigger the *Load-Inventory* Lambda function you created earlier whenever an object is created in the bucket.

Your bucket is now ready to receive inventory files!

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

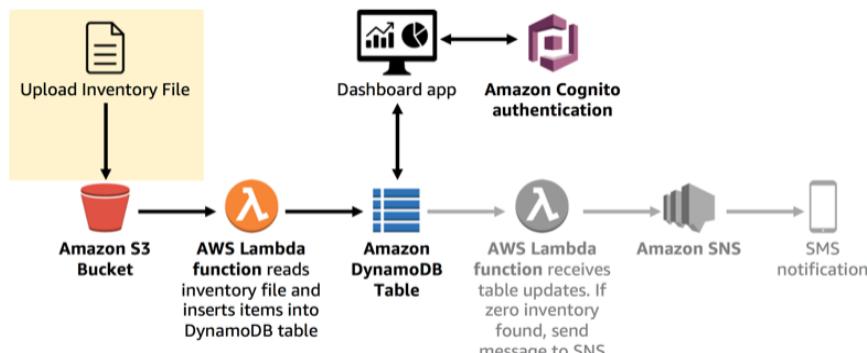
Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

You are now ready to test the loading process. You will upload an inventory file, then check that it loaded successfully.



16. Right-click this link to download a Zip file: [inventory-files.zip](#)

17. Unzip the file.

The Zip file contains multiple inventory CSV files that you can use to test the system. Here is the contents of the Berlin file:

```
store,item,count
Berlin,Echo Dot,12
Berlin,Echo (2nd Gen),19
Berlin,Echo Show,18
Berlin,Echo Plus,0
Berlin,Echo Look,10
Berlin,Amazon Tap,15
```

18. Return to your S3 bucket in the console by clicking the **Overview** tab.

19. Click **Upload** and upload one of the CSV files to the bucket. (You can choose any of the inventory files.)

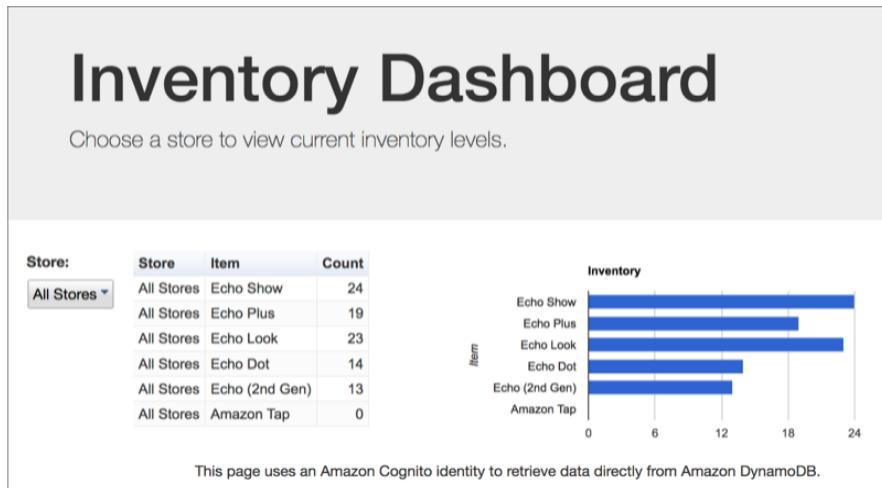
Amazon S3 will automatically trigger the Lambda function, which will load the data into a DynamoDB table.

A serverless dashboard application has been provided for you to view the results.

20. Copy the **Dashboard** URL shown to the left of the instructions you are currently reading.

21. Open a new web browser tab, paste the URL you copied and hit Enter.

The dashboard application will be displayed, showing the inventory data you loaded into the bucket. The data is being retrieved from DynamoDB, which proves that the Lambda function was successfully triggered.



⚠ If no information is displayed, ask your instructor to help diagnose the problem.

This dashboard application is served as a static web page from Amazon S3. The dashboard authenticates via Amazon Cognito as an *anonymous user*, which provides sufficient permission for the dashboard to retrieve data from DynamoDB.

You can also view the data within the DynamoDB table.

22. On the **Services** menu, click **DynamoDB**.

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 6: Test the System

Task 1: Create a Lambda Function to Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

23. In the left navigation pane, click **Tables**.

24. Click the **Inventory** table (click on the name itself).

25. Click the **Items** tab.

The data from the inventory file will be displayed, showing the Store, Item and inventory Count.

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications
Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications
Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 5: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 6: Test the System

Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 2: Configure an Amazon S3 Event

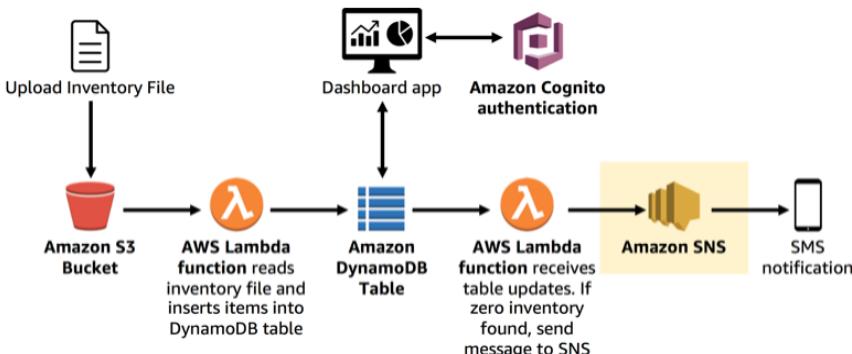
Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Task 4: Configure Notifications

You wish to notify inventory management staff when a store runs out of stock of an item. For this serverless notification functionality, you will use **Amazon Simple Notification Service (SNS)**.



Amazon SNS is a flexible, fully managed publish/subscribe messaging and mobile notifications service for the delivery of messages to subscribing endpoints and clients. With SNS you can fan-out messages to a large number of subscribers, including distributed systems and services, and mobile devices.

26. On the **Services** menu, click **Simple Notification Service**.

27. Click **Get started** if it appears, then click **Create topic** and configure:

- **Topic name:** NoStock
- **Display name:** NoStock
- **Click** **Next Step**

- Click **Create topic**

To receive notifications you must **subscribe** to the Topic. You can choose to receive notifications via several methods, such as SMS and email. Select the Topic created, click **Actions**, and then **subscribe** to the Topic.

28. Click **Create subscription** and configure:

- **Protocol:** *SMS*
- **Endpoint:** Enter your phone number in the format: 1-123-555-6789
- Click **Create subscription**

 If you do not wish to receive an SMS, you can instead subscribe via **Email**, providing an email address that you can access from the classroom. After creating an Email subscription, a confirmation email will be sent to you. Open the email and click the *Confirm subscription* link.

Any message sent to the SNS Topic will be forwarded to you via SMS or Email.

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

~~Send Notifications~~

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

~~Task 3: Test the Loading Process~~

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

~~Send Notifications~~

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

~~Send Notifications~~

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

~~Send Notifications~~

Event

Task 3: Test the Loading Process

~~Task 4: Configure Notifications~~

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Event

Task 3: Test the Loading Process

~~Task 4: Configure Notifications~~

Task 5: Create a Lambda Function to

Send Notifications

Task 6: Test the System

Load Data

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

~~Event~~

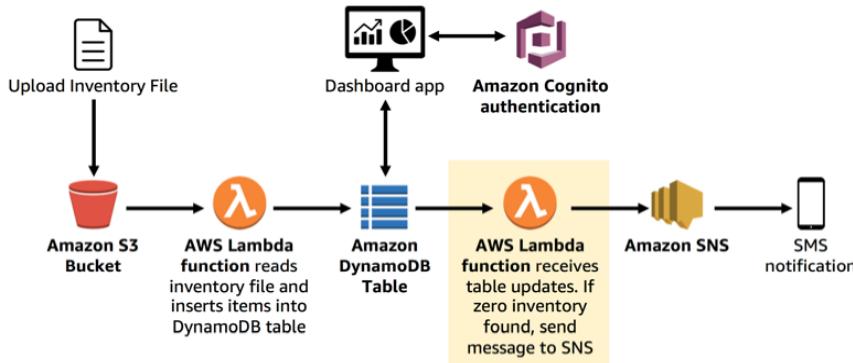
Task 5: Create a Lambda Function to Send Notifications

While you could modify the existing *Load-Inventory* Lambda function to check inventory levels while the file is being loaded, this is not a good architectural practice. Rather than overloading the *Load-Inventory* function with business logic, you will create another Lambda function that is triggered whenever data is loaded into the DynamoDB table. This will be triggered by a *DynamoDB Stream*.

There are several benefits to this architectural approach:

- Each Lambda function performs a single, specific function. This makes the code simpler and more maintainable.
- Additional business logic can be added by creating additional Lambda functions. Each function operates independently, so existing functionality is not impacted.

In this task, you will create another Lambda function that looks at inventory as it is loaded into the DynamoDB table. If it notices that an item is Out of Stock, it will send a notification via the Amazon SNS topic you created earlier.



29. On the **Services** menu, click **Lambda**.

30. Click **Create function** and configure:

- **Name:** Check-Stock
- **Runtime:** Python 3.7
- **Role:** Choose an existing role
- **Existing role:** Lambda-Check-Stock-Role
- Click **Create function**

This role has been configured with permissions to send a notification to Amazon SNS.

31. Scroll down to the **Function code** section, then delete all of the code that appears in the code editor.

32. Copy the following code, and paste it into the **Function code** editor:

```
# Stock Check Lambda function
#
# This function is triggered when values are inserted into
# the Inventory DynamoDB table.
(event, indent=2)

# For each inventory item added, check if the count is zero
for record in event['Records']:
    count = int(record['dynamodb']['NewImage']['Count'][0])

    if count == 0:
        store = record['dynamodb']['NewImage']['Store'][0]
        item = record['dynamodb']['NewImage']['Item'][0]
```

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications
Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 6: Test the System

Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

```

# Construct message to be sent
message = store + ' is out of stock of ' + item
print(message)

# Connect to SNS
sns = boto3.client('sns')
alertTopic = 'NoStock'
snsTopicArn = [t['TopicArn'] for t in sns.list_topics()]
return 'Successfully processed {} records.'.format(len(event['Records']))

```

Examine the code. It is performing the following steps:

- Loop through the incoming records
- If the inventory count is zero, send a message to the *NoStock* SNS topic

You will now configure the function to be triggered whenever data is added to the *Inventory*table in DynamoDB.

33. Scroll up to the **Add triggers** section.

34. Within the *Add triggers* section, click **DynamoDB**.

35. Scroll down to the **Configure triggers** section and configure:

- **DynamoDB Table:** *Inventory*
- Click **Add**

36. Click **Save** at the top of the page.

You are now ready to test the system!

Task 6: Test the System

You will now upload an inventory to Amazon S3, which will trigger the original *Load-Inventory*function. This function will load data into DynamoDB, which will then trigger the new *Check-Stock* Lambda function. If the Lambda function detects an item with zero inventory, it will send a message to Amazon SNS, which will notify you via SMS or

Task 5: Create a Lambda Function to Send Notifications

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 5: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

LuaU Data

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 6: Test the System

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 2: Configure an Amazon S3

Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 3: Test the Loading Process

Email.

37. On the **Services** menu, click **S3**.

38. Click on the name of your *inventory*-bucket.

39. **Upload** a different inventory file.

40. Return to the **Inventory System Dashboard** and refresh  the page.

You should now be able to use the **Store** pull-down menu to view inventory from both stores.

Also, you should receive a **notification via SMS/Email** telling you that the store is out of stock of an item (every inventory file has one item out-of-stock).

 If you did not receive a notification please wait a few minutes and try uploading a different inventory file. The DynamoDB trigger may sometimes take a few minutes to enable.

41. Try uploading multiple inventory files at the same time. What do you think will happen?

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 6: Test the System

Task 1: Create a Lambda Function to Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications
Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications



Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to

Accessing the AWS Management Console

Task 1: Create a Lambda Function to Load Data

Task 2: Configure an Amazon S3 Event

Task 3: Test the Loading Process

Task 4: Configure Notifications

Task 5: Create a Lambda Function to Send Notifications

Task 6: Test the System

Lab Complete

Lab Complete

Congratulations! You have completed the lab.

Click **End Lab** at the top of this page to clean up your lab environment.

