**Project Title**: Event-Driven Stock Tracker with AWS Lambda, DynamoDB, and SNS

### ****Setup and Requirements****

1. **AWS Account**: Ensure you have access to the AWS Management Console.
2. **AWS CLI Installed**: Configure it with your credentials.
3. **IAM Roles**:
   * A role with access to Lambda, DynamoDB, and SNS.
   * Attach policies: AWSLambdaBasicExecutionRole, AmazonDynamoDBFullAccess, AmazonSNSFullAccess.

### ****Step 1: Create a DynamoDB Table****

1. Go to the DynamoDB service in the AWS Console.
2. Click **"Create Table"**:
   * **Table name**: StockTracker
   * **Primary key**: StockID (String)
3. Leave the rest of the settings as default and create the table.

### ****Step 2: Create an SNS Topic****

1. Go to the SNS service in the AWS Console.
2. Click **"Create Topic"**:
   * **Type**: Standard
   * **Name**: StockAlerts
3. Click **"Create Subscription"**:
   * **Protocol**: Email
   * **Endpoint**: Provide your email address.
4. Confirm the subscription through the email you'll receive.

### ****Step 3: Write and Deploy the Lambda Function****

1. Go to the Lambda service in the AWS Console.
2. Create a new Lambda function:
   * **Function name**: StockHandler
   * **Runtime**: Python 3.9 (or any supported runtime)
   * **Permissions**: Attach the IAM role created earlier.
3. Use the following Lambda function code:

import boto3

import json

import os

# Initialize DynamoDB and SNS clients

dynamodb = boto3.client('dynamodb')

sns = boto3.client('sns')

# Environment variables for DynamoDB table and SNS topic

TABLE\_NAME = os.environ['TABLE\_NAME']

SNS\_TOPIC\_ARN = os.environ['SNS\_TOPIC\_ARN']

def lambda\_handler(event, context):

# Parse incoming event data

stock\_id = event['StockID']

stock\_value = event['Value']

# Store stock data in DynamoDB

dynamodb.put\_item(

TableName=TABLE\_NAME,

Item={

'StockID': {'S': stock\_id},

'Value': {'N': str(stock\_value)}

}

)

# Trigger an alert if the stock value crosses a threshold

threshold = 100 # Example threshold

if stock\_value > threshold:

sns.publish(

TopicArn=SNS\_TOPIC\_ARN,

Message=f"Alert! Stock {stock\_id} has exceeded the threshold with a value of {stock\_value}.",

Subject="Stock Value Alert"

)

return {

'statusCode': 200,

'body': json.dumps('Stock processed successfully!')

}

 Set environment variables for the Lambda function:

* **TABLE\_NAME**: StockTracker
* **SNS\_TOPIC\_ARN**: ARN of the StockAlerts topic created earlier.

 Deploy the Lambda function.

### ****Step 4: Test the Lambda Function****

1. Go to the Lambda function and click **"Test"**.
2. Create a new test event with the following payload:

{

"StockID": "AAPL",

"Value": 120

}

1. Execute the test:
   * If the value exceeds the threshold (100 in this example), you'll receive an email notification through SNS.
   * The stock data will also be stored in DynamoDB.

### ****Verification****

* **DynamoDB**: Check the StockTracker table for entries.
* **SNS**: Confirm that the email notification is received when conditions are met.