Software Engineer for Cloud Project 1

Instructions
Every screenshot requested in this workbook is compulsory and carries 5 marks. Updated Python Script carries 20 marks. Lambda Function Code In Python carries 25 marks.
Your AWS account ID must be clearly visible in every screenshot using the AWS console; missing id or using someone else's id is not permitted. Such cases will be considered as plagiarism and severe penalty will be imposed.
All screenshots must be in the order mentioned under "Expected Screenshots" for every step
DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances.
The file should be renamed in the format BATCH_FIRSTNAME_LASTNAME_PROJECT1. For example: ACSEOCT20_VIJAY_DWIVEDI_PROJECT1.docx

Resource Clean Up
Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
After completing the lab, make sure to delete each resource created in reverse chronological order.

Submission Files -

You need to submit the following -

1. Project workbook with added screenshots.
2. Updated Python script (StockPriceIngestion.py)
3. Lambda Function Code In Python

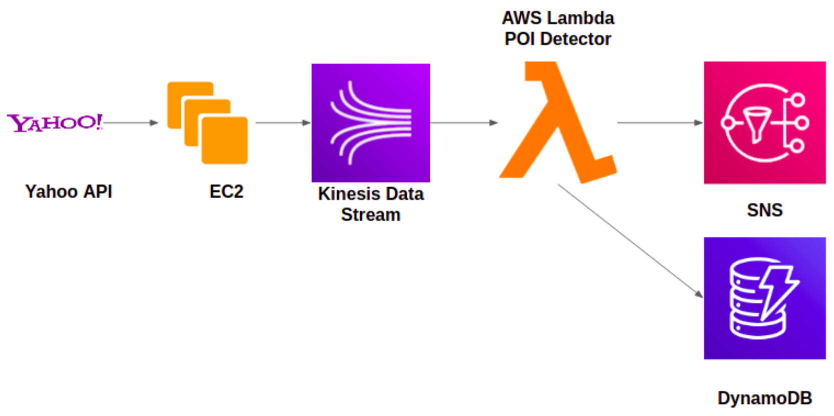
Problem Statement :

We want to build a system that streams stock pricing information for various stocks and then notifies the stakeholders about the fetched information of the stocks.

We'll use the Yahoo Finance APIs to query the running price of stocks and general information like 52-week high/low values.

Yahoo Finance API - It provides functions to download historical market data from Yahoo! finance. While this functionality in production would generally run on a paid api that provides real-time stock price data, we'll mimic it by using historical data for an older time period and streaming it over kinesis.
Link - https://pypi.org/project/yfinance/
Please go through this link to understand how to access different stocks data and information.
You need to pull the data for the following 10 stocks -
MSFT, MVIS, GOOG, SPOT, INO, OCGN, ABML, RLLCF, JNJ, PSFE
You can check the details of these stocks here - https://finance.yahoo.com/lookup/

Architecture diagram



Architecture Implementation	
1	Create EC2 Instance
2	Run Python Script to Pull data from Yahoo Stock API
3	Use boto3 and kinesis client to Push data from EC2 to Kinesis Stream
4	Configure SNS to publish the notification through mail, Configure DynamoDB to store the alert data.
5	Write a Lambda function to detect the POIs and to Push the Notification to SNS
6	Use the same Lambda Function to push data to DynamoDB

Step 1: Create EC2 Instance

Step number	a
Step name	Create EC2 Instance
Instructions	1) Navigate to EC2 Services from AWS management console page 2) Choose \`instances\` available in the left pane. 3) Click on \`launch instances\` 4) Choose the free tier for \`Amazon Linux\` 5) Choose instance type as t2.micro 6) Configure instance details keep all the option as be default 7) Keep default storage option and provide optional tag details 8) Create a new security group for the instance.(Configure SSH option) Make it more secure by choosing source as MyIP. 9) Click on the launch button, It will prompt you to create a new key pair. Provide the name and download the key for login purposes.
Expected screenshots	1) Created EC2 instance running on the instances page.

<Insert Screenshot a(1) here>

Step number	b
Step name	Use SSH to login into EC2 instance
Instructions	1) Depending on the operating system you are using do the SSH login on the EC2 instance 2) For windows system use puttygen to convert the pem file into ppk and then do the login using putty 3) For linux based system perform the normal SSH guideline available on AWS instance→ connect page
Expected screenshots	1) Logged in screenshot of the newly created instance

<Insert Screenshot b(1) here>

Step 2 : Run Python Script to Pull data from Public API

Step number	a	
Step name		
Instructions		1) Your goal is to get stock information for the ten stocks specified in the doc. Please take the postMarketPrice. 2) Further, you should call the static info api for the stocks to get their current 52WeekHigh and 52WeekLow values. 3) Update the given python code (StockPriceIngestion.py) accordingly. 4) Print the data on the console and run the script.
Expected screenshots		1) EC2 terminal after successful execution of the edited script. It doesn't have to show all of the data.

<Insert Screenshot a(1) here >

Step 3 : Use boto3 and kinesis client to Push data from EC2 to Kinesis Stream

Step number	a
Step name	Data in kinesis stream

Instructions	1) Go to the Kinesis stream services. 2) Configure the data stream to accept the data pushed by the script. 3) Create and attach appropriate policy and the IAM role to push the data to Kinesis stream from EC2. (helpful link - https://docs.amazonaws.cn/en_us/streams/latest/dev/tutorial-stock-data-kplkcl-iam.html) 4) Edit the name of the created input kinesis stream inside the python script as required. 5) You should craft individual data records with information about the stockid, postMarketPrice, 52WeekHigh and 52WeekLow values and push them individually on the Kinesis stream.
Expected screenshots	1) Screenshot of created kinesis data stream 2) Screenshot of the minimum 10 data points pushed on kinesis data stream (Screenshot of the EC2 terminal)

<Insert Screenshot a(1) here>
<Insert Screenshot a(2) here>

Step 4 : Write a Lambda function to get stock information and to push the Notification to SNS and add to DynamoDB

Step number	a
Step name	Write the lambda function
Instructions	1) Choose lambda services and create a lambda handler. Set it up to act as a consumer to your Kinesis data stream (https://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html) 2) For each stock, get the information and push it to SNS and dynamodb.
Expected screenshots	1) Created Lambda handler

<Insert Screenshot a(1) here>

Step number	b
Step name	SNS topic and subscription creation
Instructions	1) Goto AWS services look for SNS service 2) Create a standard topic 3) Create a subscription and subscribe to the topic using your email
Expected screenshots	1) SNS ARN for the created topic 2) Confirmation email for the created subscription 3) Point of interest data received through, once lambda handler is deployed

<Insert Screenshot for b(1) here >

<Insert Screenshot for b(2) here>

<Insert Screenshot for b(3) here>

Step number	c
Step name	Dynamodb table creation
Instructions	1) Goto AWS services and search for dynamodb 2) Create a table by providing the partition key and sort key
Expected screenshots	1) Empty Table 2) Table with stored data

<Insert Screenshot for c(1) here >

<Insert Screenshot for c(2) here>