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

Student name: Duy Hung D Student ID: 104182779

Tutorial session: Monday 1:00PM

1. INTRODUCTION

To implement a serverless architecture on AWS, one must make use of key services such as AWS Lambda for compute, Amazon S3 for storage, and DynamoDB for databases. These components can be connected through AWS API Gateway to create applications that are scalable and event-driven. It is important to configure event triggers for Lambda functions and set up IAM roles for secure permissions. Deployment can be facilitated using tools like AWS SAM or CloudFormation. Monitoring and logging can be ensured by utilizing CloudWatch. Additionally, security measures such as API Gateway authentication and encryption should be considered, and testing and scaling strategies should be implemented. Managing costs can be achieved by using the AWS Pricing Calculator. Lastly, it is recommended to set up a CI/CD pipeline for continuous deployment, while adhering to AWS best practices throughout the process.

1. Main

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

In the AWS Management Console, choose Lambda from the Services menu. And then create a new function.

A screenshot of a computer

Description automatically generated

*Figure 1 Lambda console*

Enter the function name is Load-Inventory and Python 3.8 runtime version.

A screenshot of a computer

Description automatically generated

*Figure 2 Basic information in function*

Expand the execution role and choose use an existing it. The role that I used is Lambda-Load-Inventory-Role. The Lambda function is granted permissions through this role in order to access Amazon S3 and DynamoDB.

A screenshot of a computer

Description automatically generated

*Figure 3 Change default execution role*

In the Code source section, choose lambda\_function.py in the Environment pane. Change the default code to the program that can 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. After that, choose Deploy to save the changes

A screenshot of a computer

Description automatically generated

*Figure 4 Code source*

**Task 2: Configuring an Amazon S3 event**

On the Services menu, choose S3 and create a new bucket. Each bucket should possess a distinct name, necessitating the addition of a random number to the bucket name. As an illustration, inventory-123 can be utilized.

A screenshot of a computer

Description automatically generated

*Figure 5 S3 console*

For Bucket name enter inventory-<number>. This number is randomly generated and I set it as 231.

A screenshot of a computer

Description automatically generated

*Figure 6 General configuration in bucket*

Choose Create bucket. You will now configure the bucket to automatically trigger the Lambda function when a file is uploaded. And select the bucket name you have created before, see the properties tab and create event notification. You will set up an event to be triggered when a new object is added to the S3 bucket.

A screenshot of a computer

Description automatically generated

*Figure 7 Event notifications*

Set “Load-Inventory” as the name and select All object create events.

A screenshot of a computer

Description automatically generated

*Figure 8 General configuration and event types of event notification*

Set Lambda function in Destination, and the name is Load-inventory. When a bucket contains an object, this configuration instructs Amazon S3 to activate the Load-Inventory Lambda function that was previously established.

A screenshot of a computer

Description automatically generated

*Figure 9 Destination in event notification*

**Task 3: Testing the loading process**

Download some python file in aws. These files represent the inventory files that are available for you to utilize in order to evaluate the system. These files are in the format of comma-separated values (CSV) files. Return to S3 console and choose Objects tab and upload CSV files.

A screenshot of a computer

Description automatically generated

*Figure 10 Objects in inventory-231*

Select one CSV file and upload it. Amazon S3 will immediately invoke the Lambda function, which will load the data into a DynamoDB table. You can view the results using the serverless Dashboard application.

A screenshot of a computer

Description automatically generated

*Figure 11 Upload CSV file to bucket*

Return to aws academy browser, choose the Details button and click on Show button in AWS, copy the Dashboard URL.

A screenshot of a computer

Description automatically generated

*Figure 12 URL in credentials*

Open a new web browser and paste the URL. The dashboard application will open and display the inventory data you entered into the bucket. The data was obtained from DynamoDB, indicating that the upload successfully activated the Lambda function.

A screenshot of a computer

Description automatically generated

*Figure 13 Inventory Dashboard*

Return to Management console, choose DynamoDB on the Services menu. And then choose Tables and see Inventory tab. The data from the inventory file will be displayed. It illustrates the store, item and inventory count

A screenshot of a computer

Description automatically generated

*Figure 14 Inventory in DynamoDB*

**Task 4: Configuring notifications**

Continue with Services menu, choose Simple Notification Services. Amazon SNS is a customizable, fully managed publish/subscribe messaging and mobile notification service. It sends messages to subscribing endpoints and clients. Amazon SNS allows you to send messages to a wide number of subscribers, including dispersed systems, services, and mobile devices. Set the topic name as NoStock and choose Next step.

A computer screen with a black and white screen

Description automatically generated

*Figure 15 Simple Notification Service*

Keep the Standard topic and and NoStock as the name, and then create topic

A screenshot of a computer

Description automatically generated

*Figure 16 Details in SNS topic*

A screenshot of a computer

Description automatically generated

*Figure 17 Create SNS topics successfully*

After that, we need to create subscription and configure the personal email. After that a confirmation email message will be sent to your email, check and confirm it.

A screenshot of a computer

Description automatically generated

*Figure 18 Create subscription*

**Task 5: Creating a Lambda function to send notifications**

Choose against Lambda on the Services menu, and then create a new function.

A screenshot of a computer

Description automatically generated

*Figure 19 Lambda functions*

Configure the settings: Check-Stock as a function name, use runtime python 3.8 version and use an existing role as Lambda-Check-Stock-Role.

A screenshot of a computer

Description automatically generated

*Figure 20 Basic information in function*

After that, create function and in the Code source section, edit lambda\_function.py. Delete all default code and copy the code that included loop through the incoming records and send a message to the NoStock SNS. Choose Deploy to save the code changes

A screenshot of a computer

Description automatically generated

*Figure 21 Code source*

At the top of the page, add trigger in Designer section. Connect with DynamoDB and set he Inventory DynamoDB Table, and then choose Add

A screenshot of a computer

Description automatically generated

*Figure 22 Trigger configuration*

A screenshot of a computer

Description automatically generated

*Figure 23 Added trigger to Check-Stock successfully*

**Task 6: Testing the System**

Choose against S3 from the Services menu, and select inventory-231 to upload other inventory file.

A screenshot of a computer

Description automatically generated

*Figure 24 Buckets inventory in S3*

Here I try to add inventory-calcutta.txt file, and then return to Inventory System Dashboard and refresh.

A screenshot of a computer

Description automatically generated

*Figure 25 Upload file in inventory-231*

You should now be able to view both stores' inventory using the Store menu. Additionally, you should be notified via SMS or email if the store has an out-of-stock item.

A screenshot of a computer

Description automatically generated

*Figure 26 Inventory Dashboard updated*

This is the dashboard when I try to upload multiple inventory files. All item will be counted in several files

A screenshot of a computer

Description automatically generated

*Figure 27 Inventory Dashboard updated*