CPD Lab 4: Serverless Computing and Web Services

Introduction

In this lab you will complete two Qwiklabs. The first introduces you to AWS Lambda showing triggering a Lambda function on adding an item to S3. The second Qwiklab introduces you to API Gateway to configure API endpoints.

You will also implement an SQS queue to trigger a Lambda function whenever a message is added to the queue.

Tasks:

- Task 1: Qwiklab: Introduction to AWS Lambda
- Task 2: Qwiklab: Introduction to Amazon API Gateway
- Task 3: AWS Lambda with SQS
- Task 4: Releasing the allocated resources

Task 1: Qwiklab: Introduction to AWS Lambda

AWS Lambda makes it easier to build event-driven serverless applications quickly that can scale automatically.

This is a free lab of 45 minutes duration. You can search for the lab on Qwiklabs.com or open through https://www.qwiklabs.com/focuses/10259?parent=catalog

In this lab you will:

- Create an AWS Lambda function
- Configure an Amazon S3 bucket as a Lambda Event Source
- Trigger a Lambda function by uploading an object to Amazon S3
- Monitor AWS Lambda S3 functions through Amazon CloudWatch Log

Task 2: Qwiklab: Introduction to Amazon API Gateway

API gateway is a fully managed endpoint for the applications to access data, business logic, or functionality from your back-end services.

This is a free lab of 55 minutes duration. You can search for the lab on Qwiklabs.com or open through https://www.qwiklabs.com/focuses/10176?parent=catalog

In this lab you will:

- Create an AWS Lambda function
- Configure an Amazon API Gateway endpoints
- Debug API Gateway and Lambda with Amazon CloudWatch

Task 3: Using AWS Lambda with SQS Queue

SQS is a managed service that can be used for decoupling of processes or store messages. In this task you will configure SQS so that a Lambda function is triggered once a message arrives in SQS queue.

Step 1: Create a Lambda Function

- Select Lambda service from AWS management console
- Click 'Create Function'
- Select 'Author from scratch' and enter 'Function name' as MyLambda
- For 'Runtime' select Python 3.7
- Under Permissions, expand 'Choose or create an execution role'
- Select 'Create a new role from AWS policy templates'
- Provide 'Role name' as 'MyLambdaRole' and click 'Create function'
- This takes some time to get created
- From 'Configuration' tab under MyLambda go to 'Execution role' and click 'View the MyLambdaRole on the IAM console'
- The IAM page opens with the role MyLambdaRole already selected
- Click 'Attach policies'
- Enter SQS in the search box and select 'AmazonSQSFullAccess'
- Expand 'AmazonSQSFullAccess' to view the policy
- Click 'Attach Policy'

Step 2: Create an SQS Queue

- Select SQS service on AWS management console and click 'Create New Queue'
- Set 'Queue Name' as MySQS
- Let the 'Region' as selected by default
- For type of queue, select 'Standard Queue'
- Before proceeding to next step review the features displayed for the 'Standard' and 'FIFO' queue
- Select 'Configure Queue'
- Read through the 'Queue Attributes' and put a check mark against 'Server-Side Encryption (SSE) Settings'. Leave 'AWS KMS Customer Master Key (CMK) and 'Data Key Reuse Period' at default values
- Click 'Create Queue'
- Under 'Queue Actions' select 'Configure Trigger for Lambda function'
- Select the function 'MyLambda' and click 'Save'
- This will ensure that the Lambda function MyLambda will be triggered whenever a message arrives in the SQS

Step 3: Triggering Lambda from SQS Queue

- From the AWS management console, select MySQS
- From 'Queue Actions' select 'Send a Message'
- Under 'Message Body' enter 'first sqs message' and click 'Send Message'
- On the AWS management console, from under 'Services' select Lambda and click the entry for your Lambda function 'MyLambda'
- Open the 'Monitoring' tab and then click 'View Logs in CloudWatch'
- Click the entry under 'Log Streams'
- This confirms that the Lambda function is being triggered

Step 4: Printing event details in the lambda function

- Click the 'Configuration' tab under 'MyLambda'
- In the 'Function Code' in 'lambda_function.py'
    - Add **print(event)** under **# TODO implement**
- Click 'Save' button at the top
- Navigate to MySQS and send another message
- Navigate to 'MyLambda' and open 'View Logs in CloudWatch' as in Step 3 above
- Expand the new message in 'Log Stream' and observe that your message is present along with related details
- The message event is described in https://docs.aws.amazon.com/lambda/latest/dg//with-sqs.html

Task 4: Releasing the allocated resources

Make sure that you have released all the resources created in Task 3 above.

- Deleted Lambda function created in Task 3, Step 1
- Deleted the role under IAM created in Task 3, Step 1
- Deleted SQS queue created in Task 3, Step 2

Links

The links below are for your reference only in case further information is required to help complete Task 3:

1. Using AWS Lambda with Amazon SQS
   https://docs.aws.amazon.com/lambda/latest/dg/with-sqs.html

2. Setting Lambda Execution Role
   https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html

3. Example of SQS access policies
   https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-basic-examples-of-sqs-policies.html

4. Amazon SQS Message Attributes
   https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-message-attributes.html

5. Details about AWS Lambda Function Handler in Python
   https://docs.aws.amazon.com/lambda/latest/dg/python-handler.html

6. AWS Lambda Function Logging in Python
   https://docs.amazonaws.cn/en_us/lambda/latest/dg/python-logging.html

7. AWS Lambda Context Object in Python
   https://docs.aws.amazon.com/lambda/latest/dg/python-context.html