CPD Lab 5: Data in the Cloud

Introduction

In this lab session you will complete two Qwiklabs. The first introduces you to AWS S3 bucket policies and versioning. In another task, you will use S3 to host a static website.

The second Qwiklab introduces you to DynamoDB and shows how to create, query and delete data in DynamoDB table. In another task, you will create a DynamoDB table and write to the table from a Lambda function.

Tasks:

- Task 1: Qwiklab: Introduction to Amazon Simple Storage Service (S3)
- Task 2: Hosting a website on S3
- Task 3: Qwiklab: Introduction to Amazon DynamoDB
- Task 4: Using Lambda function to write to DynamoDB
- Task 5: Releasing the allocated resources

Task 1: Qwiklab: Introduction to Amazon Simple Storage Service (S3)

This is a free lab of 60 minutes duration. You can search for the lab on Qwiklabs.com or open through https://www.qwiklabs.com/focuses/10406?parent=catalog

In this lab you will:

- Create a bucket in Amazon S3
- Add an object to a bucket
- Manage access permissions on an object or a bucket
- Create a bucket policy
- Use bucket versioning

Task 2: Hosting a website on Simple Storage Service (S3)

S3 can be used to host a static website.

Step 1: Enabling Website Hosting

- Select S3 under ‘Services’ on AWS management console
- Click ‘Create Bucket’ and set name as ‘mybucket’ (note that the name has to be globally unique)
- Click ‘Create’
- Select ‘mybucket’ and under ‘Properties’ tab click ‘Static website hosting’
- Select ‘Use this bucket to host a website’

- For ‘Index document’ specify the name as ‘index.html’. *This index document will be returned when requests are made to the root domain*
- Click ‘Save’
- Select ‘Upload’ under ‘Overview’ tab to upload the index.html document provided in “CPD Lab5 Resources.zip” to ‘mybucket’ and reproduced below:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
 <title>My Website Home Page</title>
</head>
<body>
 <h1>Welcome to my website</h1>
 <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

Step 2: Setting Permissions Required for Website Access

By default, S3 doesn’t allow public access to the bucket.

- Select ‘mybucket’ and choose the ‘Permissions’ tab
- Click ‘Block public access’ button and click ‘Edit’ in the box below (Block all public access)
- Uncheck ‘Block all public access’ and choose ‘Save’
- Enter *confirm* to confirm the settings
- To check the settings, select ‘mybucket’ and under ‘Access’ column the entry should be ‘Objects can be public’
- Although the objects in ‘mybucket’ are public now but we also need to provide a bucket policy for it to be accessible
- Select ‘mybucket’ and Choose ‘Permissions’ tab and choose ‘Bucket Policy’
- Copy the following bucket policy to the ‘Bucket policy editor’

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PublicReadGetObject",
         "Effect": "Allow",
         "Principal": "*",
         "Action": [
            "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::example-bucket/*"
         ]
      }
   ]
}
```

- Replace the bucket name in above from ‘example-bucket’ to ‘mybucket’
- Click ‘Save’

Step 3: Test the Website Endpoint

- From under S3, select ‘mybucket’ and Click on ‘Properties’ tab and then select ‘Static website hosting’
- Copy the Endpoint and open through a browser to observe index.html

Task 3: Qwiklab: Introduction to Amazon DynamoDB

This is a free lab of 40 minutes duration. You can search for the lab on Qwiklabs.com or open through https://www.qwiklabs.com/focuses/10407?parent=catalog

In this lab you will:

- Create an Amazon DynamoDB
- Enter data into an Amazon DynamoDB table
- Query an Amazon DynamoDB table
- Delete an Amazon DynamoDB table

Task 4: Using Lambda function to write to DynamoDB table

In this task you will create a table using AWS SDK for Python (Boto3) and add an item to it. You will then use a Lambda function to write another item to the table.

Step 1: Create a DynamoDB Table

You will now use SDK for Python (Boto3) to create a table.

- Use the file Lab5DynamoDBTable.py from “CPD Lab5 Resouces.zip” to create a Music table

  If there are issues with using the SDK then use AWS management console to create the table

- For more information on ‘ProvisionedThroughput’ in Lab5DynamoDBTable.py see https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadWriteCapacityMode.html#HowItWorks.ProvisionedThroughput.Manual
- On AWS management console, navigate to DynamoDB under ‘Services’ and click ‘Tables’ to confirm that table ‘Music’ has been created

Step 2: 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 complete
- 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 DynamoDB in the search box and select ‘AmazonDynamoDBFullAccess’
- Click ‘Attach Policy’

Step 3: Writing to DynamoDB table from the Lambda function

You will now use MyLambda function to add an item to DynamoDB table. There is no event source for this Lambda function and you will trigger it using the ‘Test’ feature.

- Select Lambda service from AWS management console
- Click on ‘MyLambda’
- Under ‘Function Code’ replace the existing code with the following:

```
import json
import boto3

def lambda_handler(event, context):
    # Get the service resource.
    dynamodb = boto3.resource('dynamodb')
    # Instantiate a table resource object
    table = dynamodb.Table('Music')
    table.put_item(
    Item={
        'Artist': 'John Lennon',
        'Song': 'Imagine',
        'Album': 'Imagine',
        'Year': 1971,
        'Genre': 'Soft rock',
        }
    )
```

- Click ‘Save’ at right top
- To configure a ‘Test’ event, click ‘Test’ button on top-right

- Select ‘Create new test event’ and select ‘Hello World’ from the drop-down for ‘Event template’
- Enter ‘MyTestEvent’ for the ‘Event Name’
- As you don’t need to pass any value from this event, in the box containing

  ```
  {
    "key1": "value1",
    "key2": "value2",
    "key3": "value3"
  }
  ```

  Change it to

  ```
  {
  }
  ```
- Click ‘Create’
- Click ‘Test’ to run ‘MyLambda’ function under ‘Configuration’ tab
- The ‘Execution results’ will be visible under ‘MyLambda’
- Click ‘Monitoring’ tab and then click ‘View Logs in CloudWatch’
- Select the entry under ‘Log Streams’
- These logs are very helpful for development and troubleshooting
- Navigate to ‘Music’ table and you should see the added item
- Delete the table
- Delete the Lambda function and Lambda Role

Task 5: Releasing the allocated resources

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

- Deleted S3 bucket created in Task 2
- Deleted Lambda function and Lambda Role created in Task 4
- Deleted the DynamoDB table created in Task 4

Links

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

Task 2

1. Hosting a Static Website on Amazon S3
   https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html

   https://docs.aws.amazon.com/AmazonS3/latest/user-guide/static-website-hosting.html#test-your-website-endpoint

Task 4

2. Partitions and Data Distribution
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.Partitions.html

3. Core Components of Amazon DynamoDB
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html

4. Choosing the Right DynamoDB Partition Key
https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/