**Scheduling DynamoDB Backups**

Introduction

Let's assume you want to take a backup of one of your DynamoDB tables to an S3 bucket every day. A simple way to achieve this is to use an Amazon CloudWatch Events rule to trigger an AWS Lambda function daily. In this hands-on AWS lab, you will write a Lambda function in Python using the Boto3 library.

To accomplish this, we will do the following:

* Configure an IAM role
* Set up a CloudWatch rule
* Create a Lambda function

Log in to the AWS Management Console with the credentials provided on the lab instructions page. Make sure you are using the *us-east-1 (N. Virginia)* region.

Create a Lambda Function

1. In the AWS Management Console, navigate to the Lambda service.
2. Click **Create function**.
3. Select **Author from scratch**, and configure the following settings:
   * **Name:** CreateDynamoDBBackup
   * **Runtime:** Python 3.7
   * **Role:** Create a custom role
4. In the IAM Management Console, under *Role Summary*, click **View policy document**.
5. Click **Edit**.
6. In a new browser tab, open [this file](https://raw.githubusercontent.com/linuxacademy/content-lambda-boto3/master/Scheduling-DynamoDB-Backups/lambda_execution_role.json), and copy the contents to your clipboard.
7. Paste the text into the IAM policy document editor, replacing the text that was there before.
8. Click **Allow**.
9. Switch back to your Lambda Management Console browser tab, and click **Create function**.
10. Scroll down the page to the code editor.
11. Select the default text, and delete it.
12. In a new browser tab, open [this file](https://raw.githubusercontent.com/linuxacademy/content-lambda-boto3/master/Scheduling-DynamoDB-Backups/lambda_function.py), and copy the contents to your clipboard.
13. Paste the Python source code into the Lambda code editor.
14. Click **Save**.

Configure a Test Event

1. On the *CreateDynamoDBBackup* page, click **Test**.
2. In the *Configure test event* menu, configure the following settings:
   * **Event name:** Test
   * **Contents:** {"TableName": "Person"}
3. Click **Create**.
4. Click **Test**. You should see a successful execution result.
5. In a new browser tab, open the DynamoDB Management Console.
6. Click **Backups** in the left sidebar.
7. The page should show the backup that we took using the Lambda function.

Create a CloudWatch Rule to Trigger the Lambda Function

1. In a new brower tab, open the CloudWatch Management Console.
2. Click **Rules** in the left sidebar.
3. Click **Create rule**.
4. Under *Event Source*, click **Schedule**, and configure the following settings:
   * **Fixed rate of:** 1 Minutes
5. Under *Targets*, click **Add target**.
6. Under *Lambda function*, configure the following settings:
   * **Function:** CreateDynamoDBBackup
   * **Configure input:** Constant (JSON text)
7. In the text box under *Constant (JSON text)*, type {"TableName": "Person"}.
8. Click **Configure details**.
9. For *Name*, type "BackupDynamoDBPerson".
10. Click **Create rule**.
11. In the left sidebar, click **Logs**.
12. Click the CreateDynamoDBBackup log group to open it.
13. Click the name of the log stream to open it.
14. Wait for the CloudWatch rule to trigger the next backup job you have scheduled.
15. Verify that the scheduled backup job ran.
16. Switch to your DynamoDB browser tab, and verify that the backup file exists in the list of DynamoDB backups.
17. Verify that old backups were deleted for subsequent runs. (The Lambda function will retain the three most recent backups.)