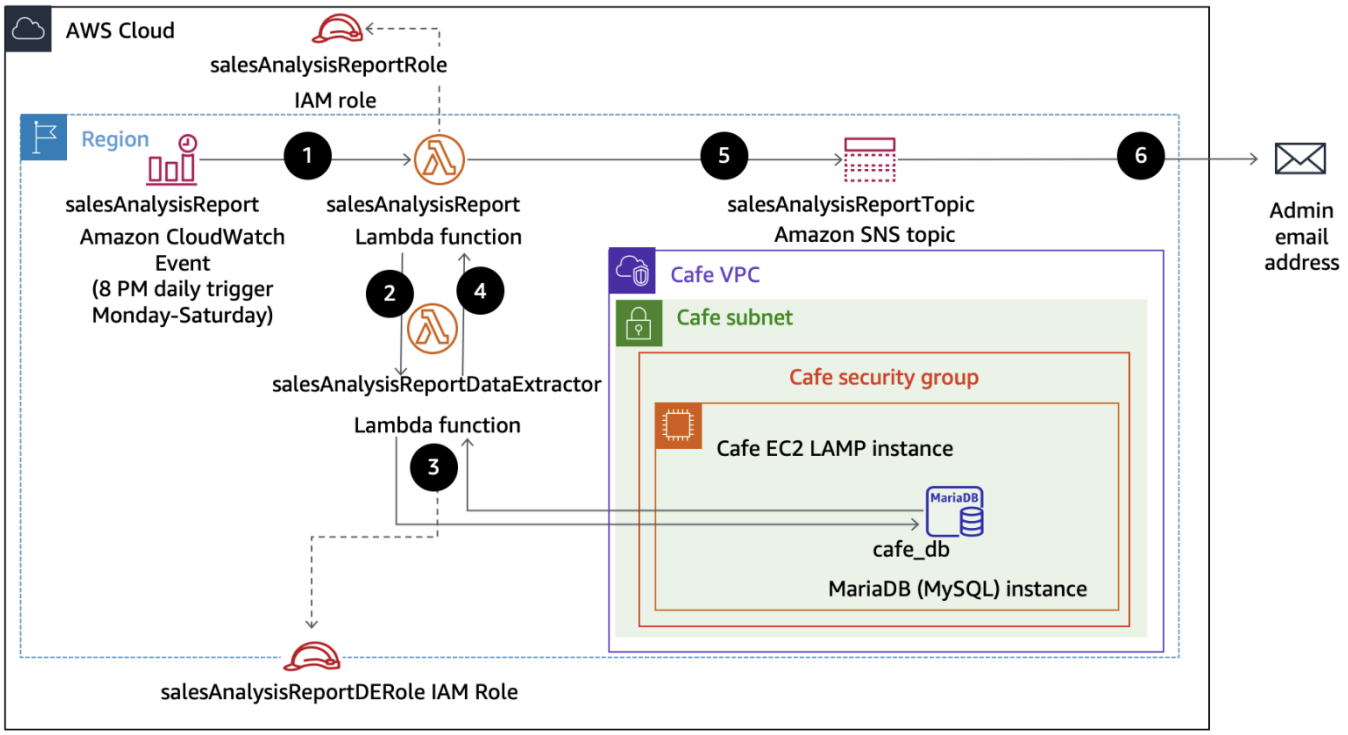
Working with AWS Lambda

In this lab, you deploy and configure an AWS Lambda based serverless computing solution. The Lambda function generates a sales analysis report by pulling data from a database and emailing the results daily. The database connection information is stored in Parameter Store, a capability of AWS Systems Manager. The database itself runs on an Amazon Elastic Compute Cloud (Amazon EC2) Linux, Apache, MySQL, and PHP (LAMP) instance.

The following diagram shows the architecture of the sales analysis report solution and illustrates the order in which actions occur.



The diagram includes the following function steps:

Step	Details
1	An Amazon CloudWatch Events event calls the salesAnalysisReport Lambda function at 8 PM every day Monday through Saturday.
2	The salesAnalysisReport Lambda function invokes another Lambda function, salesAnalysisReportDataExtractor, to retrieve the report data.
3	The salesAnalysisReportDataExtractor function runs an analytical query against the café database (cafe_db).
4	The query result is returned to the salesAnalysisReport function.
5	The salesAnalysisReport function formats the report into a message and publishes it to the salesAnalysisReportTopic Amazon Simple Notification Service (Amazon SNS) topic.
6	The salesAnalysisReportTopic SNS topic sends the message by email to the administrator.

In this lab, the Python code for each Lambda function is provided to you so that you can focus on the SysOps tasks of deploying, configuring, and testing the serverless solution components.

After completing this lab, you will be able to do the following:

- Recognize necessary AWS Identity and Access Management (IAM) policy permissions to facilitate a Lambda function to other Amazon Web Services (AWS) resources.
- Create a Lambda layer to satisfy an external library dependency.
- Create Lambda functions that extract data from database, and send reports to user.
- Deploy and test a Lambda function that is initiated based on a schedule and that invokes another function.
- Use CloudWatch logs to troubleshoot any issues running a Lambda function.

Step 1: Observing the IAM role settings

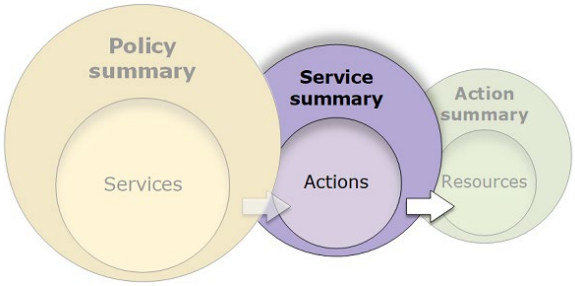
สร้างฟังก์ชัน Lambda สองฟังก์ชัน ฟังก์ชันแต่ละฟังก์ชันต้องการสิทธิ์การเข้าถึงทรัพยากร AWS ที่ฟังก์ชันนั้นโต้ตอบด้วย

วิเคราะห์ IAM roles และสิทธิ์ที่ roles เหล่านั้นจะมอบให้กับฟังก์ชัน Lambda salesAnalysisReport และ salesAnalysisReportDataExtractor ที่จะสร้างขึ้นในภายหลัง

Step 1.1: Observing the salesAnalysisReport IAM role settings

1) Services > Security, Identity, & Compliance > IAM > Roles > salesAnalysisReportRole
2) เลือก Trust Relationships และจะเห็นว่า lambda.amazonaws.com อยู่ในรายการของ trusted entity ซึ่งหมายความว่า Lambda service สามารถใช้ IAM role นี้ได้

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```



การกำหนด role ให้ Lambda service สามารถใช้ได้ คลิกที่ตัว code > Edit statement > AssumeRole

3) เลือก Permissions เพื่อดูรายการสิทธิ์ที่กำหนดไว้สำหรับ role นี้ โดยจะมี role ที่กำหนดไว้ 4 รายการ สามารถขยายแต่ละ role เพื่อดูรายละเอียดของสิทธิ์ที่แต่ละ role มอบให้ โดยคลิกที่ไอคอน +

- AmazonSNSFullAccess ให้สิทธิ์เต็มรูปแบบในการเข้าถึงทรัพยากร SNS
- AmazonSSMReadOnlyAccess ให้สิทธิ์เฉพาะการอ่านเท่านั้นในการเข้าถึงทรัพยากร Systems Manager
- AWSLambdaBasicRunRole ให้สิทธิ์ในการเขียนลงใน CloudWatch Logs (ซึ่งเป็นสิ่งที่จำเป็นสำหรับทุก Lambda function)
- AWSLambdaRole ให้ Lambda function สามารถเรียกใช้ Lambda function อื่นๆได้
- code การให้สิทธิ์ในแต่ละ role อ่านเพิ่มเติมได้ใน Additional file

ฟังก์ชัน Lambda: salesAnalysisReport ที่จะถูกสร้างขึ้นในภายหลัง จะใช้ role: salesAnalysisReportRole

Step 1.2: Observing the salesAnalysisReportDE IAM role settings

4) Services > Security, Identity, & Compliance > IAM > Roles > salesAnalysisReportDERole
5) เลือก Trust Relationships และจะเห็นว่า lambda.amazonaws.com อยู่ในรายการของ trusted entity ซึ่งหมายความว่า Lambda service สามารถใช้ IAM role นี้ได้

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

การกำหนด Role ให้ Lambda service สามารถใช้ได้ คลิกที่ตัว code > Edit trust policy > AssumeRole

6) เลือก Permissions และสังเกตสิทธิ์ที่ได้รับใน role นี้:

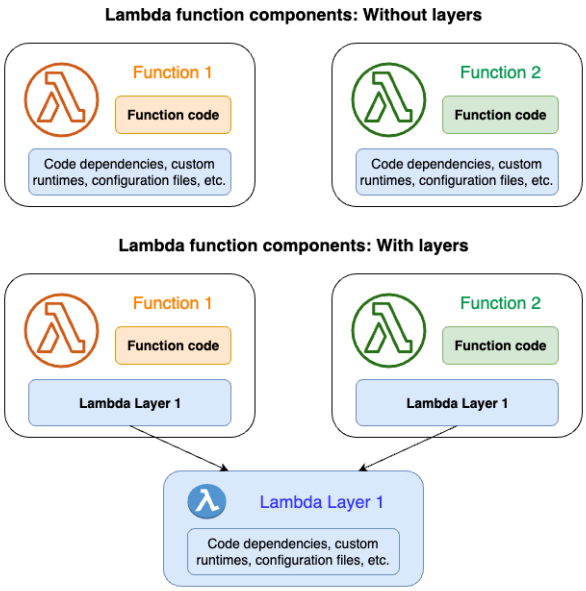
- AWSLambdaBasicRunRole ให้สิทธิ์ในการเขียนลงใน CloudWatch Logs
- AWSLambdaVPCAccessRunRole ให้สิทธิ์ในการจัดการ Elastic Network Interfaces (ENIs) เพื่อเชื่อมต่อฟังก์ชันเข้ากับ Virtual Private Cloud (VPC)
- code การให้สิทธิ์ในแต่ละ role อ่านเพิ่มเติมได้ใน Additional file

ฟังก์ชัน Lambda: salesAnalysisReportDataExtractor ที่จะถูกสร้างขึ้นในภายหลัง จะใช้ role: salesAnalysisReportDERole

Step 2: Creating a Lambda layer and a data extractor Lambda function

ขั้นตอนนี้จะแบ่งเป็น 2 ส่วน ดังนี้

1. สร้าง Lambda layer: เลเยอร์ คือ แพ็คเกจของโค้ดที่สามารถใช้ซ้ำกับ Lambda function หลายฟังก์ชันได้
2. สร้าง Lambda function ที่ใช้เลเยอร์ดังกล่าว: Lambda function คือ ส่วนหนึ่งของโค้ดที่รันบนคลาวด์เพื่อตอบสนองต่ออีเวนต์



Lambda layer เป็น Subset ของ Lambda function

เมื่อเพิ่มเลเยอร์ให้กับ Lambda function, Lambda จะแยกเนื้อหาของเลเยอร์เข้าไปในไดเรกทอรี /opt ใน execution environment ของฟังก์ชัน, natively supported Lambda runtimes (สามารถใช้งานเลเยอร์ใน Lambda ได้โดยตรงโดยไม่จำเป็นต้องทำการกำหนดค่าหรือติดตั้งเพิ่มเติม) จะรวม path เข้าไปยัง directory ใน /opt ที่เฉพาะเจาะจง ซึ่งทำให้ฟังก์ชันนั้นๆสามารถเข้าถึงเนื้อหาของเลเยอร์ได้

ข้อจำกัด

สามารถเพิ่มเลเยอร์ได้สูงสุด 5 เลเยอร์ต่อ 1 ฟังก์ชัน

สามารถใช้เลเยอร์กับ Lambda function ที่ deploy เป็นไฟล์ .zip ถาวรเท่านั้น.

7) ดาวน์โหลดไฟล์ที่ต้องใช้ในแล็บนี้

pymysql-v3.zip salesAnalysisReportDataExtractor-v3.zip

หมายเหตุ:

- ไฟล์ salesAnalysisReportDataExtractor-v3.zip เป็นการประยุกต์ใช้งาน Lambda function ด้วยภาษา Python ซึ่งใช้ PyMySQL open-source client library เพื่อเข้าถึงฐานข้อมูล MySQL ของร้าน café, ไลบรารีนี้ได้ถูกแพ็คเกจเป็นไฟล์ pymysql-v3.zip ซึ่งจะถูกอัปโหลดไปยัง Lambda layer ต่อไป

Step 2.1: Creating a Lambda Layer

สร้าง Lambda Layer ที่ชื่อว่า pymysqlLibrary และอัปโหลด client library ลงไป เพื่อให้ฟังก์ชันใดๆที่ต้องการสามารถใช้งานได้ นอกจากนี้ Lambda Layer ยังเป็นกลไกที่ยืดหยุ่นในการนำโค้ดกลับมาใช้งานระหว่างฟังก์ชัน เพื่อให้โค้ดไม่ถูกรวมเข้าไปใน deployment package ของแต่ละฟังก์ชัน

8) Services > Compute > Lambda > Layers > Create Layer
9) กำหนดการตั้งค่าเลเยอร์ ดังนี้

- Name: pymysqlLibrary
- Description: PyMySQL library modules
- เลือก Upload a .zip file เพื่ออัปโหลดไฟล์ pymysql-v3.zip
- สำหรับ Compatible runtimes (runtime environments) เลือก Python 3.9

10) Create

เคล็ดลับ: Lambda layers feature กำหนดให้ไฟล์ .zip ที่บรรจุโค้ดหรือไลบรารีต้องตรงตามโครงสร้างโฟลเดอร์เฉพาะ ไฟล์ pymysqlLibary.zip ที่ใช้ในแล็บนี้ ถูกแพ็คโดยใช้โครงสร้างโฟลเดอร์ดังนี้

```
python/
├── PyMySQL-1.0.2.dist-info
└── pymysql
```

Step 2.2: Creating a data extractor Lambda function

11) Lambda > Functions > Create function และกำหนดค่าดังนี้

- เลือก Author from scratch
- Function name: salesAnalysisReportDataExtractor
- Runtime: Python 3.9
- Change default execution role กำหนดการตั้งค่า ดังนี้
    - Execution role: Use an existing role.
    - Existing role: salesAnalysisReportDERole

12) Create function

Step 2.3: Adding the Lambda layer to the function

13) Function overview > Layers > Add a layer

14) กำหนดค่าดังนี้

- Choose a layer: Custom layers
- Custom layers: pymysqlLibrary
- Version: 1

15) Add

Function overview panel จะแสดงจำนวน (1) ใน Layer node สำหรับฟังก์ชัน

Step 2.4: Importing the code for the data extractor Lambda function

16) Lambda > Functions > salesAnalysisReportDataExtractor

17) Runtime settings > Edit

18) Handler: salesAnalysisReportDataExtractor.lambda_handler

19) Save

20) Code source > Upload from > .zip file > upload > salesAnalysisReportDataExtractor-v3.zip

21) Save

Lambda function code จะถูกนำเข้าและแสดงใน Code source panel

22) Review Python code ที่ใช้สร้างฟังก์ชันนี้

อ่าน Comments ภายในโค้ดเพิ่มเติมได้ใน Additional file เพื่อเข้าใจขั้นตอนต่างๆในการทำงานของโค้ด โปรดสังเกตว่าฟังก์ชันมีการรับข้อมูลการเชื่อมต่อฐานข้อมูล (dbURL, dbName, dbUser, และ dbPassword) ผ่านพารามิเตอร์ event input

Step 2.5: Configuring network settings for the function

ขั้นตอนสุดท้ายก่อนที่จะทดสอบฟังก์ชัน คือ การกำหนดค่าการตั้งค่าเครือข่าย ตาม architecture ที่แสดงไว้ ฟังก์ชันนี้ต้องการการเข้าถึงเครือข่ายไปยังฐานข้อมูลร้าน cafe ซึ่งทำงานบน EC2 LAMP instance ดังนั้น จึงจำเป็นต้องระบุข้อมูล VPC, subnet, และ security group ของ instance ในการกำหนดค่าของฟังก์ชัน

23) Configuration > VPC

24) Edit และกำหนดค่าดังนี้

- VPC: Cafe VPC (Name)
- Subnets: Cafe Public Subnet 1 (Name)

เคล็ดลับ: สามารถละเว้นคำเตือน (ถ้ามี) ที่แนะนำให้เลือกอย่างน้อยสอง subnet เพื่อเรียกใช้ High Availability mode เนื่องจากไม่สามารถใช้กับฟังก์ชันนี้ได้

- Security groups: CafeSecurityGroup (Name)

25) Save

Step 3: Testing the data extractor Lambda function

Step 3.1: Launching a test of the Lambda function

ตอนนี้เราพร้อมที่จะทดสอบฟังก์ชัน salesAnalysisReportDataExtractor แล้ว ในการเรียกใช้งานจำเป็นต้องระบุค่าสำหรับพารามิเตอร์การเชื่อมต่อฐานข้อมูลของร้าน cafe และจำว่าข้อมูลเหล่านี้ถูกเก็บไว้ใน Parameter Store

26) Services > Management & Governance > Systems Manager

27) Parameter Store > คัดลอกและวาง Value ในแต่ละรายการต่อไปนี้ ลงใน text editor document

- /cafe/dbUrl
- /cafe/dbName
- /cafe/dbUser
- /cafe/dbPassword

28) Lambda Management Console > salesAnalysisReportDataExtractor > Test

29) Test event และกำหนดค่าดังนี้

- Test event action: Create new event
- Event name: SARDETestEvent
- Template: choose hello-world
- Event JSON แทน JSON object ด้วย JSON object นี้:

```
{
  "dbUrl": "<value of /cafe/dbUrl parameter>",
  "dbName": "<value of /cafe/dbName parameter>",
  "dbUser": "<value of /cafe/dbUser parameter>",
  "dbPassword": "<value of /cafe/dbPassword parameter>"
}
```

แทนค่าของแต่ละพารามิเตอร์ด้วย Value ที่คุณวางไว้ใน text editor document ในขั้นตอนก่อนหน้านี้ และครอบค่าเหล่านี้ด้วยเครื่องหมาย " "

30) Save

31) Test

หลังจากนั้นสักพัก หน้าเว็บจะแสดงข้อความ “Execution result: failed”

Step 3.2: Troubleshooting the data extractor Lambda function

32) Execution result > Details เพื่อดูรายละเอียด และสังเกตว่า object error ที่คืนค่ากลับมา มีข้อความที่คล้ายกับข้อความต่อไปนี้ หลังจากที่ฟังก์ชันทำงานแล้ว

```
{
  "errorMessage": "2019-02-14T04:14:15.282Z ff0c3e8f-1985-44a3-8022-519f883c8412 Task
timed out after 3.00 seconds"
}
```

ข้อความนี้ระบุว่าฟังก์ชันหมดเวลาหลังจากผ่านไป 3 วินาที

ส่วนของผลลัพธ์ของ Log (Log output) จะมีบรรทัดที่ขึ้นต้นด้วยคำสำคัญดังต่อไปนี้

- START หมายถึง ฟังก์ชันเริ่มทำงาน
- END หมายถึง ฟังก์ชันทำงานเสร็จสิ้น
- REPORT สรุปข้อมูลประสิทธิภาพและสถิติการใช้ทรัพยากรที่เกี่ยวข้องกับเมื่อฟังก์ชันทำงาน

สาเหตุของข้อผิดพลาดนี้คืออะไร?

Step 3.3: Analyzing and correcting the Lambda function

33) วิเคราะห์และแก้ไขปัญหาที่พบจากการทดสอบฟังก์ชัน Lambda

ต่อไปนี้จะเป็นแนะนำเพื่อช่วยค้นหาวิธีแก้ไข

- ฟังก์ชันนี้เริ่มต้นด้วยการเชื่อมต่อกับฐานข้อมูล MySQL ที่รันอยู่ใน EC2 instance ที่แยกออกมา ฟังก์ชันจะรอเป็นระยะเวลาหนึ่งเพื่อรอการเชื่อมต่อให้สำเร็จ หากหลังจากเวลานี้ผ่านไปแล้วและการเชื่อมต่อไม่สำเร็จ ฟังก์ชันจะหมดเวลา
- โดยค่าเริ่มต้น ฐานข้อมูล MySQL ใช้ MySQL protocol และเปิดการฟัง (listen) บนพอร์ตหมายเลข 3306 เพื่อให้ผู้ใช้เข้าถึงได้
- เลือกแท็บ Configuration อีกครั้ง > VPC ตรวจสอบ Inbound rules สำหรับ Security group ที่ใช้โดย EC2 instance ที่รันฐานข้อมูล > พอร์ตของฐานข้อมูล (3306) ถูกระบุไว้หรือไม่ > หากไม่ก็สามารถเลือกที่ Security group link เพื่อแก้ไขและเพิ่ม Inbound rule ได้

34) เมื่อทำการแก้ไขเรียบร้อยแล้ว กลับไปที่ salesAnalysisReportDataExtractor > Test

หากขึ้น Execution result: succeeded (logs) แสดงว่าฟังก์ชันทำงานเสร็จสมบูรณ์แล้ว

35) เลือก Details ฟังก์ชันส่งคืน JSON object ดังนี้

```
{
  "statusCode": 200,
  "body": []
}
```

ฟิลด์ body ว่างเปล่า (ข้อมูลรายงานที่ฟังก์ชันดึงมา) เนื่องจากไม่มีข้อมูลการสั่งซื้อในฐานข้อมูล

Step 3.4: Placing an order and testing again

เข้าถึงเว็บไซต์ของร้าน cafe และสั่งซื้อสินค้าบางรายการเพื่อเติม (ส่ง) ข้อมูลเข้าไปในฐานข้อมูล

36) เพื่อเปิดเว็บไซต์ร้าน cafe ในแท็บเบราว์เซอร์ใหม่ ให้ค้นหาหมายเลข Public IP ของ EC2 instance ของร้าน cafe URL สำหรับเว็บไซต์มีรูปแบบ http://publicIP/cafe โดยที่ publicIP คือ หมายเลข Public IP ของ EC2 instance ของร้าน cafe มีวิธีการหาหมายเลข Public IP มีดังนี้

- Services > Compute > EC2 > Instances > CafeInstance
- คัดลอกที่อยู่ Public IPv4 ไปวางใน text editor
- เปิดแท็บเบราว์เซอร์ใหม่ พิมพ์ http://publicIP/cafe โดยแทนที่ publicIP ด้วยที่อยู่ Public IPv4 ที่คัดลอกไว้ก่อนหน้านี้
- กด Enter เพื่อโหลดเว็บไซต์ร้าน cafe

37) ที่เว็บไซต์ร้าน cafe เลือกเมนู และสั่งซื้อสินค้าเพื่อสร้างข้อมูลในฐานข้อมูล
ตอนนี้มีข้อมูลการสั่งซื้อในฐานข้อมูลแล้ว และสามารถทดสอบฟังก์ชันอีกครั้งได้

38) salesAnalysisReportDataExtractor > Test
JSON object จะส่งคืนค่ากลับมา และตอนนี้จะมีข้อมูลปริมาณสินค้าอยู่ในฟิลด์ body ดังนี้


```
{
  "statusCode": 200,
  "body": [
    {
      "product_group_number": 1,
      "product_group_name": "Pastries",
      "product_id": 1,
      "product_name": "Croissant",
      "quantity": 1
    },
    {
      "product_group_number": 2,
      "product_group_name": "Drinks",
      "product_id": 8,
      "product_name": "Hot Chocolate",
      "quantity": 2
     }
    ]
}
```


เท่านี้ก็จะสามารถสร้าง salesAnalysisReportDataExtractor Lambda function ได้แล้ว

Step 4: Configuring notifications

สร้าง SNS (Simple Notification Service) topic และ subscribe อีเมลกับ topic นั้น

Step 4.1: Creating an SNS topic

สร้าง SNS topic สำหรับเผยแพร่รายงานการวิเคราะห์ยอดขาย และ subscribe อีเมลกับ topic นั้น และ topic ดังกล่าวมีหน้าที่ส่งข้อความทั้งหมดที่ได้รับไปยังสมาชิกทุกคน เราจะใช้ Amazon SNS console เพื่อดำเนินงานนี้

39) Services > Application Integration > Simple Notification Service

40) Topics > Create topic

41) กำหนดค่าดังนี้

- Type: Standard
- Name: salesAnalysisReportTopic
- Display name: SARTopic

42) Create topic

43) คัดลอกและวางค่า ARN ลงใน text editor document

จะต้องระบุค่า ARN นี้ เมื่อทำการกำหนดค่า Lambda function ในลำดับต่อไป

Step 4.2: Subscribing to the SNS topic

44) Create subscription และกำหนดค่าดังนี้

Protocol: Email

Endpoint: email address

45) Create subscription > subscription จะถูกสร้างขึ้นและมีสถานะเป็น Pending confirmation

46) ตรวจสอบกล่องจดหมายของที่อยู่อีเมลที่ subscribe ไว้

จะเห็นอีเมลจาก SARTopic มีหัวข้อ "AWS Notification - Subscription Confirmation."

47) เปิดอีเมลและเลือก Confirm subscription

แท็บเบราว์เซอร์ใหม่จะเปิดขึ้นมาและแสดงหน้าเว็บพร้อมข้อความ "Subscription confirmed!"

Step 5: Creating the salesAnalysisReport Lambda function

สร้างและกำหนดค่า Lambda function ที่ชื่อว่า salesAnalysisReport ซึ่งเป็นฟังก์ชันหลักในการทำงานของรายงานวิเคราะห์การขาย ฟังก์ชันดังกล่าวทำงานดังนี้

- ดึงข้อมูลการเชื่อมต่อฐานข้อมูลจาก Parameter Store
- เรียกใช้ salesAnalysisReportDataExtractor Lambda function เพื่อดึงข้อมูลรายงานจากฐานข้อมูล
- จัดรูปแบบและเผยแพร่ข้อความที่มีข้อมูลรายงานไปยัง SNS topic

Step 5.1: Connecting to the CLI Host instance

ใช้ EC2 Instance Connect เพื่อเข้าสู่ระบบ CLI Host instance ที่รันอยู่ในบัญชี AWS ซึ่งมี AWS Command Line Interface (AWS CLI) ติดตั้งอยู่แล้วและมีโค้ด Python ที่จำเป็นสำหรับการสร้างฟังก์ชัน Lambda ต่อไป จากนั้นจะต้องรัน

คำสั่ง AWS CLI เพื่อสร้างฟังก์ชัน Lambda ใหม่ ท้ายที่สุดเราจะสามารถทำการทดสอบแบบ unit test ฟังก์ชันใหม่ได้โดยใช้ Lambda management console

48) EC2 Management Console > Instances

49) CLI Host: check box

50) Connect

51) EC2 Instance Connect > Connect

Step 5.2: Configuring the AWS CLI

Amazon Linux instances มี AWS CLI ติดตั้งพร้อมใช้งานไว้แล้ว; อย่างไรก็ตาม ยังคงต้องระบุ credentials เพื่อเชื่อมต่อ AWS CLI client เข้ากับบัญชี AWS

52) ใน EC2 Instance Connect terminal window ให้รันคำสั่งต่อไปนี้เพื่ออัปเดตซอฟต์แวร์ AWS CLI พร้อมด้วย credentials

    aws configure

53) ที่ prompts กำหนดค่าดังนี้

- AWS Access Key ID: ตามที่แล็บให้มา
- AWS Secret Access Key: ตามที่แล็บให้มา
- Default region name: ตามที่แล็บให้มา
- Default output format: json

Step 5.3: Creating the salesAnalysisReport Lambda function using the AWS CLI

54) เพื่อตรวจสอบว่าไฟล์ salesAnalysisReport-v2.zip ที่มีโค้ดสำหรับฟังก์ชัน Lambda salesAnalysisReport อยู่บน CLI Host เรียบร้อยแล้ว ให้รันคำสั่งต่อไปนี้ในเทอร์มินัล

```
cd activity-files
ls
```

```
[ec2-user@ip-10-200-0-137 ~]$ aws configure
AWS Access Key ID [None]: AKIA47CRVEEHAP35RBPK
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]: json
[ec2-user@ip-10-200-0-137 ~]$ cd activity-files
[ec2-user@ip-10-200-0-137 activity-files]$ ls
salesAnalysisReport-v2.zip
[ec2-user@ip-10-200-0-137 activity-files]$
```

หมายเหตุ:

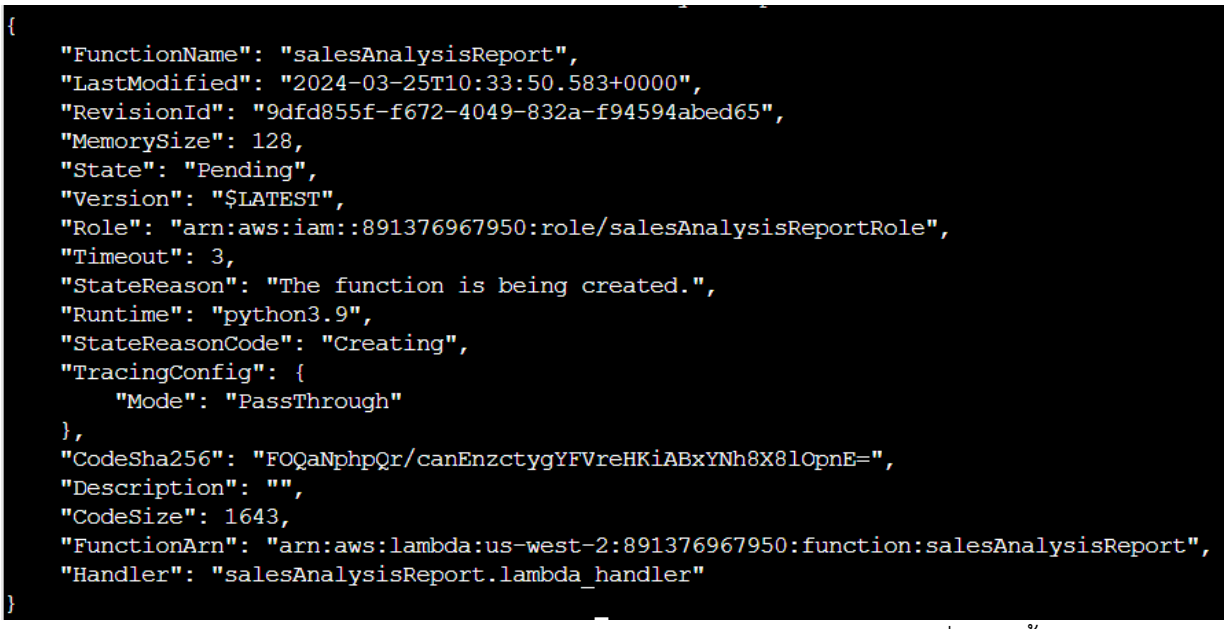
- ก่อนที่จะสร้างฟังก์ชัน ต้องทำการดึงข้อมูล ARN ของ IAM role ชื่อว่า salesAnalysisReportRole ออกมาก่อน สามารถทำได้ตามขั้นตอนต่อไปนี้

55) Roles > salesAnalysisReportRole > summary > ARN

56) คัดลอกและวาง ARN ลงใน text editor document

57) ใช้คำสั่ง Lambda create-function เพื่อสร้างฟังก์ชัน Lambda และกำหนดค่าให้ใช้ IAM role ชื่อว่า salesAnalysisReportRole ในการกำหนดสิทธิ์การเข้าถึงและการดำเนินการต่างๆใน AWS
เพื่อดำเนินการขั้นตอนนี้ ให้วางคำสั่งต่อไปนี้ที่ terminal window command prompt แทนที่ <salesAnalysisReportRoleARN> ด้วยค่า ARN ของ salesAnalysisReportRole ที่ได้คัดลอกไว้ในขั้นตอนก่อนหน้า และแทนที่ <region> ด้วยโค้ด us-west-2 ซึ่งเป็นภูมิภาคที่ใช้ในการสร้างฟังก์ชัน Lambda ก่อนหน้านี้ (ในการค้นหาโค้ดนี้ ไปที่เมนูบนสุดทางด้านขวาของ Lambda management console เลือก Region dropdown menu)

```
aws lambda create-function \
--function-name salesAnalysisReport \
--runtime python3.9 \
--zip-file fileb://salesAnalysisReport-v2.zip \
--handler salesAnalysisReport.lambda_handler \
--region <region> \
--role <salesAnalysisReportRoleARN>
```



salesAnalysisReport Lambda function จะถูกสร้างหลังจากรันคำสั่งเสร็จสิ้น

เมื่อคำสั่งดำเนินการเสร็จสิ้น จะส่งคืน JSON object ที่อธิบายแอตทริบิวต์ของฟังก์ชัน เท่านี้ก็จะสามารถกำหนดค่าและทำการทดสอบแบบ unit test ได้แล้ว

Step 5.4: Configuring the salesAnalysisReport Lambda function

58) Lambda management console > Functions > salesAnalysisReport > Details

59) ตรวจสอบ Function overview, Code source สำหรับการสร้างฟังก์ชัน
แนะนำให้อ่านผ่านโค้ดฟังก์ชัน และใช้คำอธิบายภายในโค้ดเพื่อช่วยให้เข้าใจใน logic
โปรดสังเกตบรรทัดที่ 26 ว่าฟังก์ชันดึงค่า ARN ของ topic ที่ต้องการเผยแพร่ จาก Environment variable ที่ชื่อว่า topicARN ดังนั้น จึงต้องกำหนดตัวแปรนั้นใน Environment variables panel.

```
TOPIC_ARN = os.environ['topicARN'] ## Line 26 ##
```

60) Configuration > Environment variables > Edit

61) Add environment variable และกำหนดค่า ดังต่อไปนี้

Key: topicARN

Value: วางค่า ARN ของ SNS topic ที่ชื่อว่า salesAnalysisReportTopic ที่คัดลอกไว้ก่อนหน้านี้

62) Save

Step 5.5: Testing the salesAnalysisReport Lambda function

ตอนนี้เราพร้อมที่จะทดสอบฟังก์ชันแล้ว

63) Test และกำหนดค่าดังนี้

Test event action: Create new event

Even name: SARTestEvent

Template: hello-world

ฟังก์ชันไม่ต้องการพารามิเตอร์ใดๆ (ให้ Event JSON เป็นค่า default)

64) Save

65) Test

เคล็ดลับ: หากได้รับข้อผิดพลาดเกี่ยวกับ timeout error ให้เลือกปุ่ม test อีกครั้ง บางครั้งเมื่อรันฟังก์ชันครั้งแรก มันจะใช้เวลานานกว่าปกติในการเตรียมความพร้อม และค่า default timeout ของ Lambda (3 วินาที) จะเกินไปกว่านี้เล็กน้อย โดยทั่วไปจะสามารถรันฟังก์ชันอีกครั้งได้ แล้วข้อผิดพลาดจะหายไป หรือในทางเลือกอื่นคือเพิ่มค่า timeout โดยทำตามขั้นตอนเหล่านี้:

Configuration > General configuration > Edit > ปรับเปลี่ยน Timeout ตามความเหมาะสม > Save

66) เลือก Details ฟังก์ชันส่งคืน JSON object ดังนี้

```
{
    "statusCode": 200,
    "body": "\"Sale Analysis Report sent.\""
}
```

67) ตรวจสอบกล่องจดหมาย

หากไม่มีอะไรผิดพลาด ควรจะได้รับอีเมลจากการ AWS Notifications ที่มีหัวข้อว่า " Daily Sales Analysis Report.” อีเมลนั้นควรจะมีรายงานที่คล้ายกับภาพต่อไปนี้ ขึ้นอยู่กับคำสั่งซื้อที่ถูกสั่งบนเว็บไซต์ร้าน cafe

```
                    Sales Analysis Report
                       Date: 2019-02-15

Product Group: Pastries

              Item Name                          Quantity
              *********                          ********
              Croissant                            1
                Donut                              2
        Chocolate Chip Cookie                      7
                Muffin                             4
    Strawberry Blueberry Tart                      9
         Strawberry Tart                           6

Product Group: Drinks

              Item Name                          Quantity
              *********                          ********
                Coffee                             7
           Hot Chocolate                           10
                Latte                              9
```

68) สามารถทดสอบฟังก์ชันการทำงานของรายงานได้โดยการสั่งซื้อเพิ่มเติมผ่านเว็บไซต์ของร้าน cafe และจะสังเกตเห็นการเปลี่ยนแปลงข้อมูลในรายงานที่คุณได้รับ

ตอนนี้เราได้ทดสอบฟังก์ชัน salesAnalysisReport ของ Lambda ด้วยการทดสอบแบบ unit test เรียบร้อยแล้ว

Step 5.6: Adding a trigger to the salesAnalysisReport Lambda function

เพื่อให้การทำงานของฟังก์ชัน salesAnalysisReport สมบูรณ์ จำเป็นต้องกำหนดค่าให้มีการรันรายงานทุกวันตั้งแต่วันจันทร์ถึงวันเสาร์ เวลา 20:00 น. โดยใช้อีเวนต์ของ CloudWatch Events (EventBridge) เป็นตัวเรียกใช้งานฟังก์ชัน

69) Function overview > Add trigger

70) กำหนดค่าดังนี้

Trigger configuration: EventBridge (CloudWatch Events)

Rule: Create a new rule

Rule name: salesAnalysisReportDailyTrigger

Rule description: Initiates report generation on a daily basis

Rule Type: Schedule expression

Schedule expression: สามารถระบุเวลาที่ต้องการให้ระบบทำงานตามกำหนดการโดยใช้ Cron expression รูปแบบทั่วไปของ Cron expression ประกอบด้วย 6 ฟิลด์ที่คั่นด้วยช่องว่าง ดังนี้

- cron(Minutes Hours Day-of-month Month Day-of-week Year) นอกจากนี้ เวลาทั้งหมดในนิพจน์ Cron จะอ้างอิงตามเวลา UTC (เวลาสากลเชิงพิกัด)
- ถ้าอยู่ที่กรุงเทพฯ (UTC time zone +7) ต้องปรับค่าเวลาตาม UTC โดยหัก 7 ชั่วโมงจากเวลาที่ต้องการ หากเวลานี้คือ 10:00 น. ให้ใส่คำสั่งนี้: `cron(03 ? * MON-SAT *)`

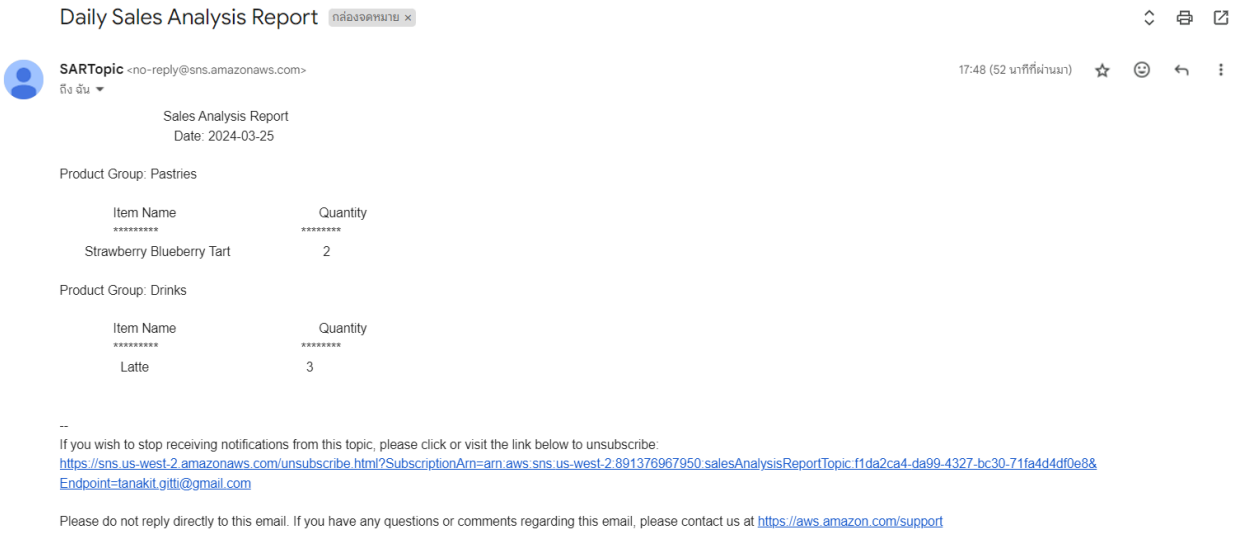
71) Add

ทริกเกอร์ใหม่จะถูกสร้างขึ้นและแสดงใน Function overview panel และ Triggers pane

72) รอ 5 นาที จากนั้นให้ตรวจสอบกล่องจดหมายอีเมล

หากไม่มีอะไรผิดพลาด ควรจะได้รับอีเมลใหม่จาก AWS Notifications ที่มีหัวข้อว่า "Daily Sales Analysis Report"

อีเวนต์ของ CloudWatch Events (EventBridge) ได้เรียกใช้ข้อความนี้ตามเวลาที่ถูกระบุไว้ในนิพจน์ Cron



Daily Sales Analysis Report กล่องจดหมาย ×

SARTopic <no-reply@sns.amazonaws.com>
ถึง ฉัน

17:48 (52 นาทีที่ผ่านมา)

Sales Analysis Report
Date: 2024-03-25

Product Group: Pastries

Item Name	Quantity
Strawberry Blueberry Tart	2

Product Group: Drinks

Item Name	Quantity
Latte	3

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:891376967950:salesAnalysisReportTopic:f1da2ca4-da99-4327-bc30-71fa4d4df0e8&Endpoint=tanakit.gitti@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support