AWS re:Start

LAB

Working with AWS Lambda

WEEK 9

aws re/start

Overview

AWS Lambda is a serverless computing service that runs your code in response to events without provisioning servers. It automatically scales applications and charges only for the compute time used, making it cost-effective for tasks like data processing and real-time file handling.

You can write Lambda functions in languages such as Python, Node.js, or Java, and trigger them with AWS services like S3, DynamoDB, and API Gateway. This integration allows you to build scalable, efficient applications with minimal management, handling everything from backend processing to real-time data analysis.

Topics covered

- 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.

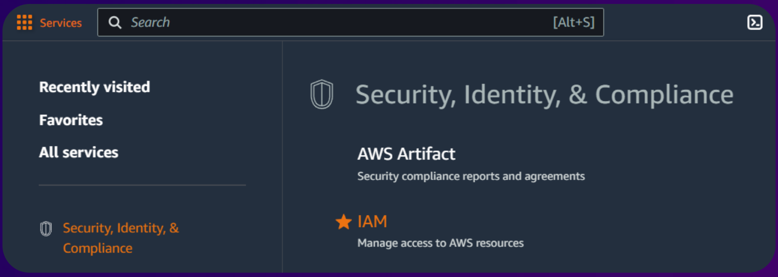
aws re/start

Task 1

Observing the IAM role settings

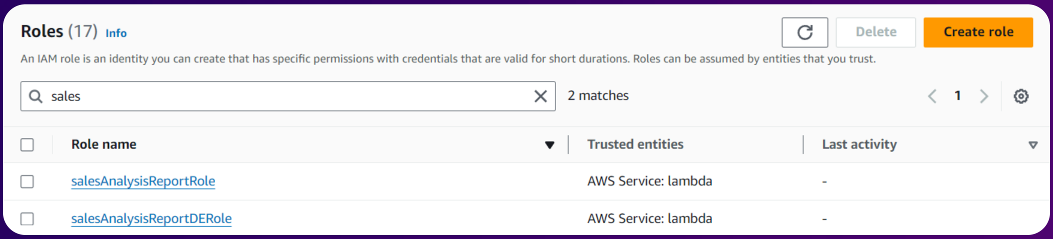
Step 1: Access the AWS Management Console

Open the AWS Management Console, and select IAM.


Services
Search
[Alt+S]
Recently visited
Favorites
All services
Security, Identity, & Compliance
Security, Identity, & Compliance
AWS Artifact
Security compliance reports and agreements
IAM
Manage access to AWS resources

Step 2: Review Roles

Navigate to the **Roles** section, and review the following roles.



An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
salesAnalysisReportRole	AWS Service: lambda	-
salesAnalysisReportDERole	AWS Service: lambda	-

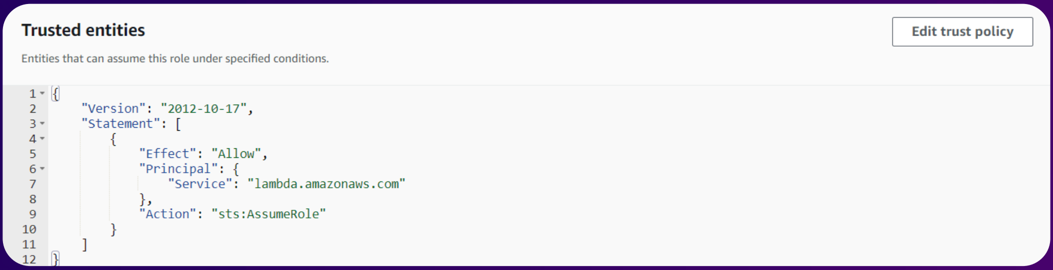
aws re/start

Task 1

Observing the IAM role settings

Step 3: Review trust relationships

Choose the **salesAnalysisReportRole** role, and choose the **Trust relationships** tab, and notice that lambda.amazonaws.com is listed as a trusted entity, which means that the Lambda service can use this role.



Trusted entities

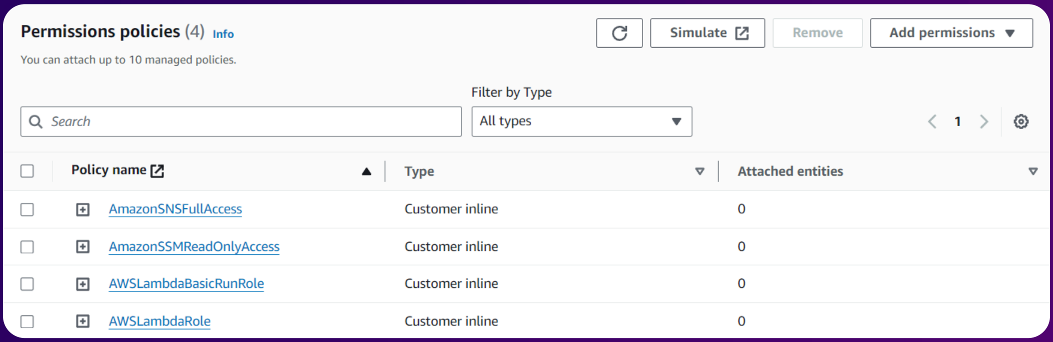
Entities that can assume this role under specified conditions.

Edit trust policy

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

Step 4: The salesAnalysisReport role

Choose the **Permissions** tab, and review the four permissions policies assigned to the **salesAnalysisReport** role.



Permissions policies (4) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonSNSFullAccess	Customer inline	0
AmazonSSMReadOnlyAccess	Customer inline	0
AWSLambdaBasicRunRole	Customer inline	0
AWSLambdaRole	Customer inline	0

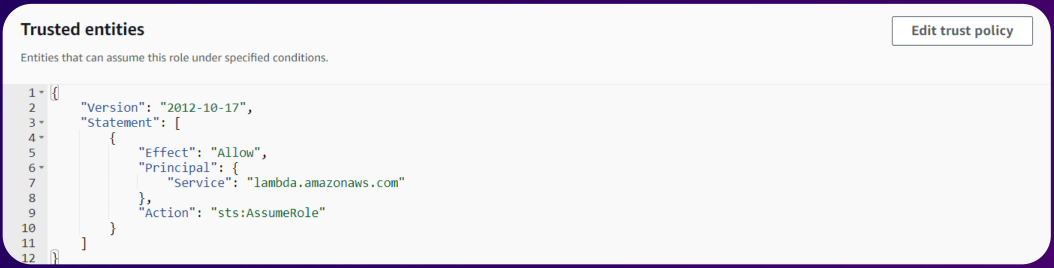
aws re/start

Task 1

Observing the IAM role settings

Step 5: Review trust relationships

Choose the **salesAnalysisReportDERole** role, and choose the **Trust relationships** tab, and notice that lambda.amazonaws.com is listed as a trusted entity.



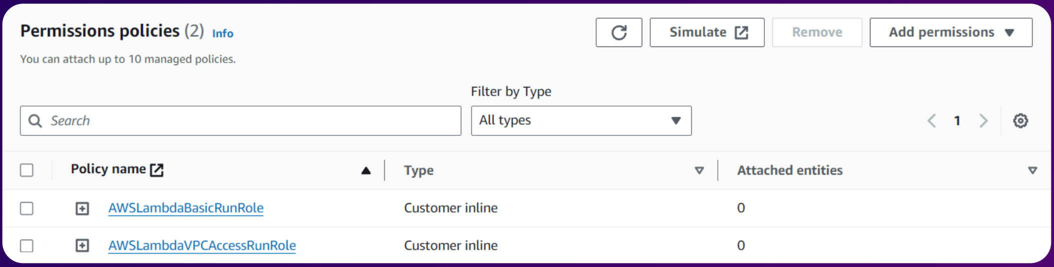
Trusted entities

Entities that can assume this role under specified conditions.

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

Step 6: The salesAnalysisDEReport role

Choose the **Permissions** tab, and review the two permissions policies assigned to the **salesAnalysisDEReport** role.



Permissions policies (2) Info

You can attach up to 10 managed policies.

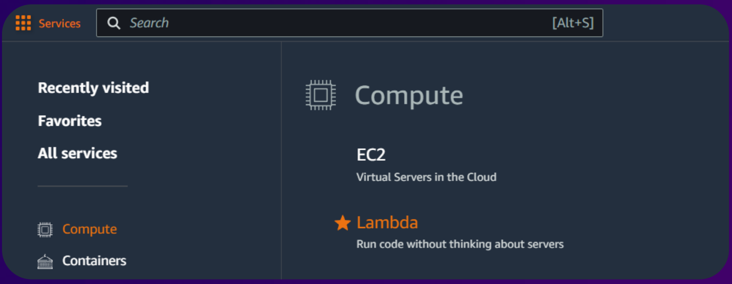
Policy name	Type	Attached entities
AWSLambdaBasicRunRole	Customer inline	0
AWSLambdaVPCAccessRunRole	Customer inline	0

Task 2

Creating a Lambda layer and a data extractor Lambda function

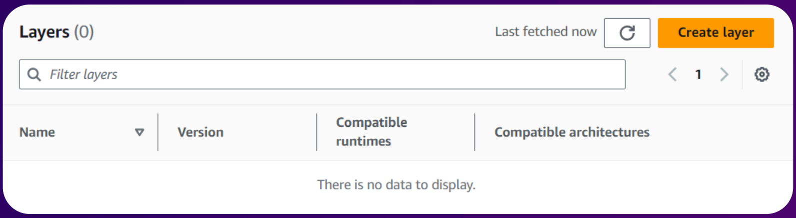
Step 1: Access the Lambda service

In the AWS Management Console, select Lambda.



Step 2: Create layer

Navigate to the **Layers** section, and select Create layer.



Name	Version	Compatible runtimes	Compatible architectures

There is no data to display.

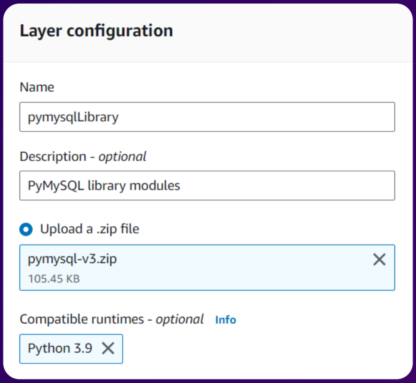
aws re/start

Task 2

Creating a Lambda layer and a data extractor Lambda function

Step 3: Layer configuration

In the **Layer configuration** section, configure the following settings.

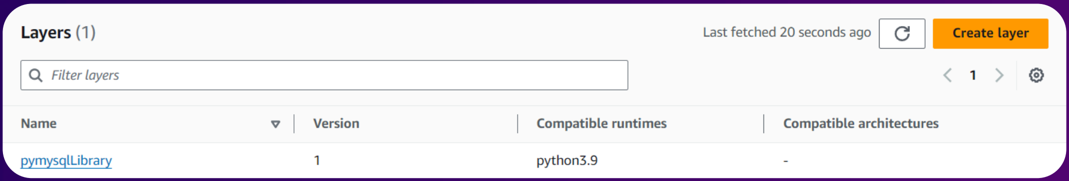


Layer configuration

Name	pymysqlLibrary
Description - *optional*	PyMySQL library modules
● Upload a .zip file	pymysql-v3.zip 105.45 KB
Compatible runtimes - *optional* Info	Python 3.9

Step 4: Review Layer creation

Review the newly created **pymysqlLibrary** layer.



Layers (1)

Last fetched 20 seconds ago

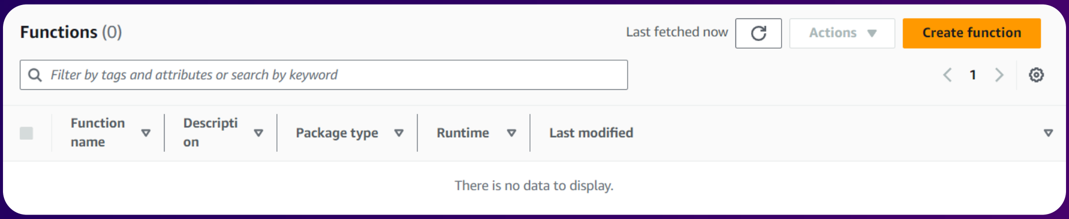
Name	Version	Compatible runtimes	Compatible architectures
pymysqlLibrary	1	python3.9	-

Task 2

Creating a Lambda layer and a data extractor Lambda function

Step 5: Create function

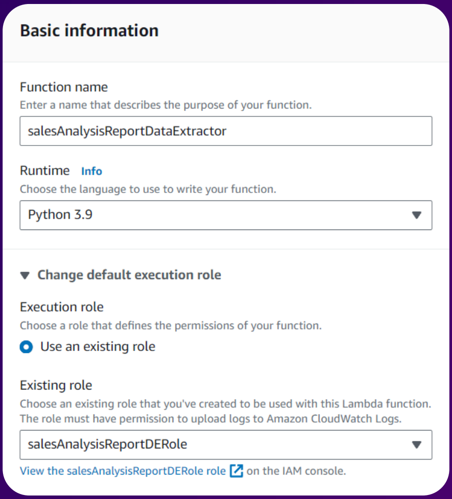
Navigate to the **Functions** section, and select Create function.



There is no data to display.

Step 6: Basic information

In the **Create function** page, select Author from scratch, and configure the following settings in the **Basic information** section.



Basic information

Function name
Enter a name that describes the purpose of your function.
salesAnalysisReportDataExtractor

Runtime Info
Choose the language to use to write your function.
Python 3.9

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function.
● Use an existing role

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
salesAnalysisReportDERole
View the salesAnalysisReportDERole role on the IAM console.

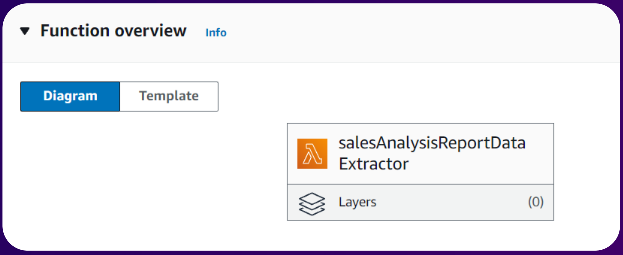
aws re/start

Task 2

Creating a Lambda layer and a data extractor Lambda function

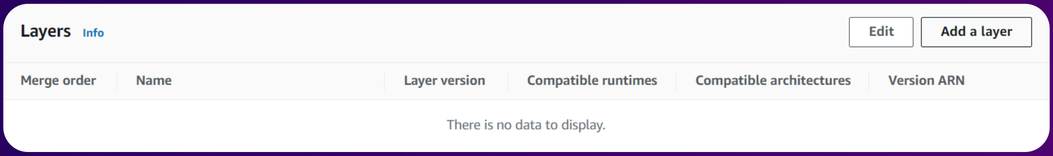
Step 7: Layers

In the **Function overview** panel, choose Layers.



Step 8: Add a layer

In the **Layers** panel, choose Add a layer.



Layers Info

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN

There is no data to display.

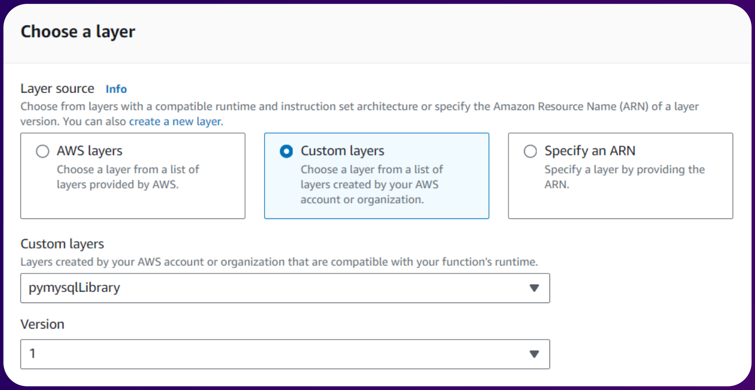
aws re/start

Task 2

Creating a Lambda layer and a data extractor Lambda function

Step 9: Choose a layer

In the **Choose a layer** section, configure the following settings.



Choose a layer

Layer source Info
Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also create a new layer.

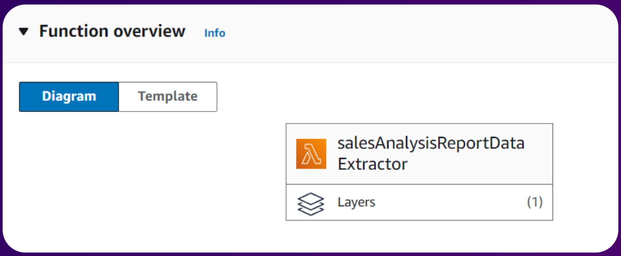
AWS layers — Choose a layer from a list of layers provided by AWS.
[✓ Custom layers] — Choose a layer from a list of layers created by your AWS account or organization.
Specify an ARN — Specify a layer by providing the ARN.

Custom layers
Layers created by your AWS account or organization that are compatible with your function's runtime.
pymysqlLibrary

Version
1

Step 10: Review Function overview

The **Function overview** panel shows a count of (1) in the Layers node for the function.



Function overview Info

Diagram | Template

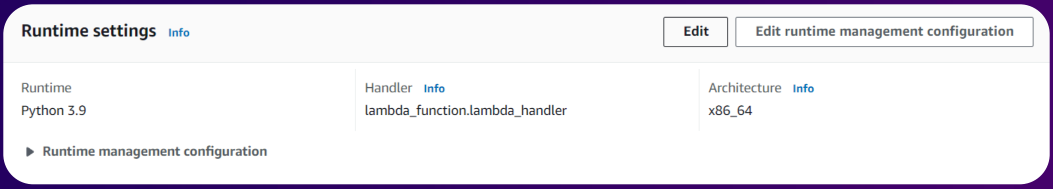
salesAnalysisReportDataExtractor
Layers (1)

Task 2

Creating a Lambda layer and a data extractor Lambda function

Step 11: Edit Runtime settings

In the **Runtime settings** panel, choose Edit.



Runtime settings Info

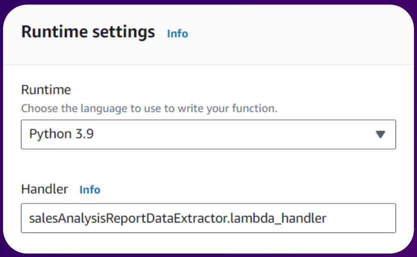
Edit
Edit runtime management configuration

Runtime	Handler Info	Architecture Info
Python 3.9	lambda_function.lambda_handler	x86_64

▶ Runtime management configuration

Step 12: Runtime settings

In the **Runtime settings** section, configure the following settings.



Runtime settings Info

Runtime
Choose the language to use to write your function.
Python 3.9

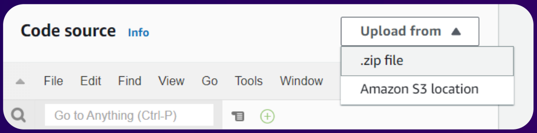
Handler Info
salesAnalysisReportDataExtractor.lambda_handler

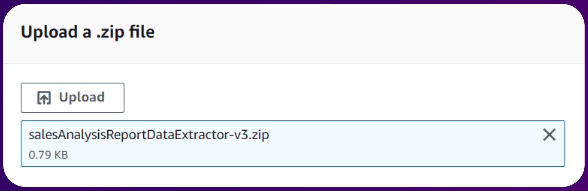
Task 2

Creating a Lambda layer and a data extractor Lambda function

Step 13: Upload Code source

In the **Code source** panel, choose Upload from, and select the following .zip file. The Lambda function code is imported and displayed in the **Code source** panel.

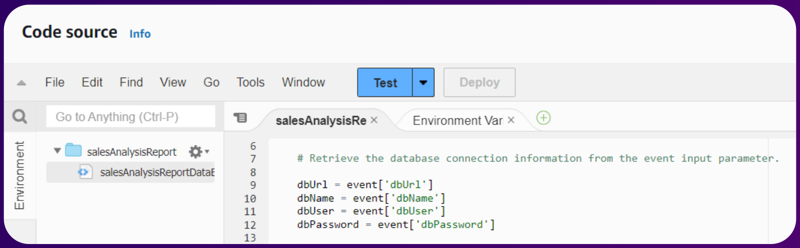




salesAnalysisReportDataExtractor-v3.zip
0.79 KB

Step 14: Review Code source

Review the Python code that implements the function. Notice that the function expects to receive the database connection information (dbURL, dbName, dbUser, and dbPassword) in the event input parameter.



```

# Retrieve the database connection information from the event input parameter.

dbUrl = event['dbUrl']
dbName = event['dbName']
dbUser = event['dbUser']
dbPassword = event['dbPassword']

```

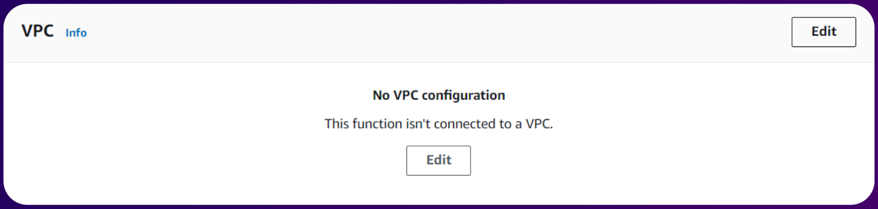
aws re/start

Task 2

Creating a Lambda layer and a data extractor Lambda function

Step 15: Edit VPC configuration

Choose the **Configuration** tab, choose **VPC**, and choose Edit.

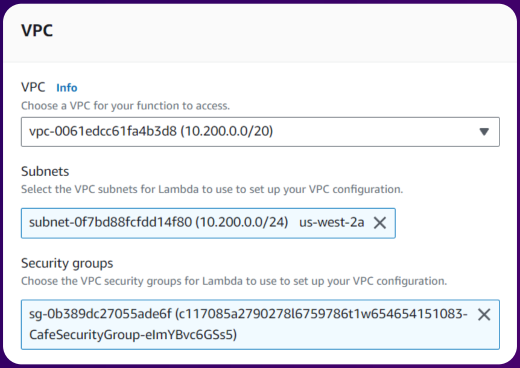


No VPC configuration

This function isn't connected to a VPC.

Step 16: VPC

In the **VPC** section, configure the following settings.



VPC

Choose a VPC for your function to access.

vpc-0061edcc61fa4b3d8 (10.200.0.0/20)

Subnets

Select the VPC subnets for Lambda to use to set up your VPC configuration.

subnet-0f7bd88fcfdd14f80 (10.200.0.0/24) us-west-2a

Security groups

Choose the VPC security groups for Lambda to use to set up your VPC configuration.

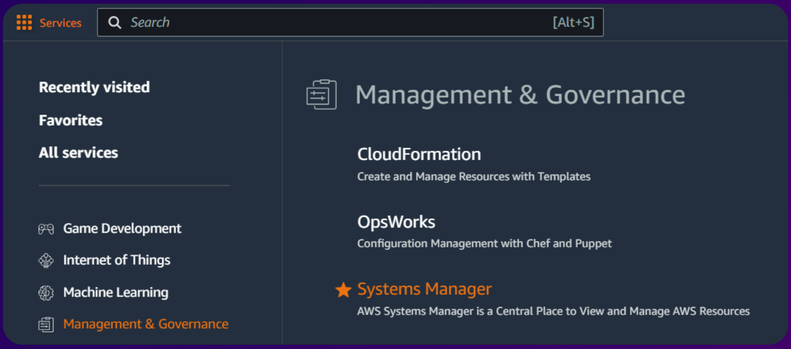
sg-0b389dc27055ade6f (c117085a2790278l6759786t1w654654151083-CafeSecurityGroup-elmYBvc6GSs5)

Task 3

Testing the data extractor Lambda function

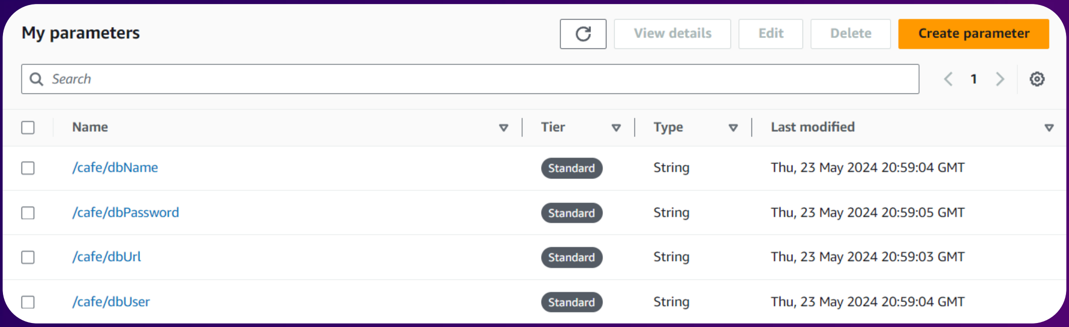
Step 1: Access the Systems Manager service

In the AWS Management Console, select Systems Manager.



Step 2: Review My parameters

Navigate to the **Parameter Store** section, and review the following parameters and their values.



Name	Tier	Type	Last modified
/cafe/dbName	Standard	String	Thu, 23 May 2024 20:59:04 GMT
/cafe/dbPassword	Standard	String	Thu, 23 May 2024 20:59:05 GMT
/cafe/dbUrl	Standard	String	Thu, 23 May 2024 20:59:03 GMT
/cafe/dbUser	Standard	String	Thu, 23 May 2024 20:59:04 GMT

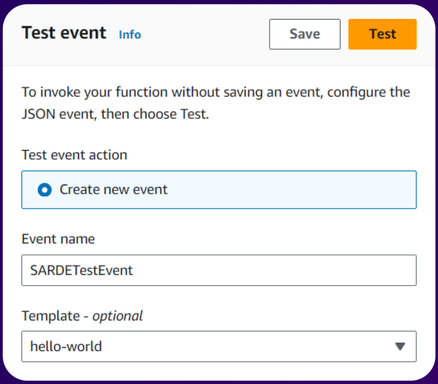
aws re/start

Task 3

Testing the data extractor Lambda function

Step 3: Test event

On the **salesAnalysisReportDataExtractor** function page, choose the **Test** tab. Configure the **Test event** panel as follows. Then, choose Save and Test.


Test event Info

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

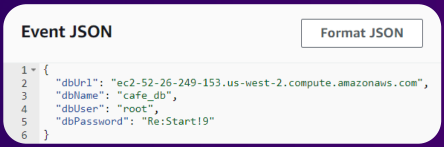
● Create new event

Event name

SARDETestEvent

Template - *optional*

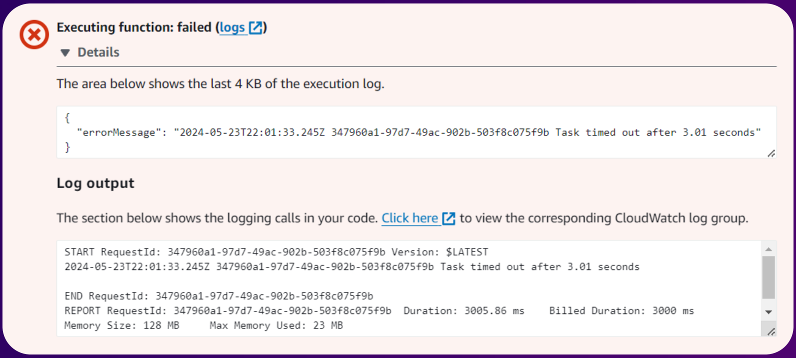
hello-world


Event JSON

```
{
  "dbUrl": "ec2-52-26-249-153.us-west-2.compute.amazonaws.com",
  "dbName": "cafe_db",
  "dbUser": "root",
  "dbPassword": "Re:Start!9"
}
```

Step 4: Troubleshoot the failed execution

In the Executing function: failed pane, review the Details section, the returned errorMessage object, and the Log output.


Executing function: failed (logs)

▼ Details

The area below shows the last 4 KB of the execution log.

```
{
  "errorMessage": "2024-05-23T22:01:33.245Z 347960a1-97d7-49ac-902b-503f8c075f9b Task timed out after 3.01 seconds"
}
```

Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: 347960a1-97d7-49ac-902b-503f8c075f9b Version: $LATEST
2024-05-23T22:01:33.245Z 347960a1-97d7-49ac-902b-503f8c075f9b Task timed out after 3.01 seconds

END RequestId: 347960a1-97d7-49ac-902b-503f8c075f9b
REPORT RequestId: 347960a1-97d7-49ac-902b-503f8c075f9b  Duration: 3005.86 ms    Billed Duration: 3000 ms
Memory Size: 128 MB     Max Memory Used: 23 MB
```

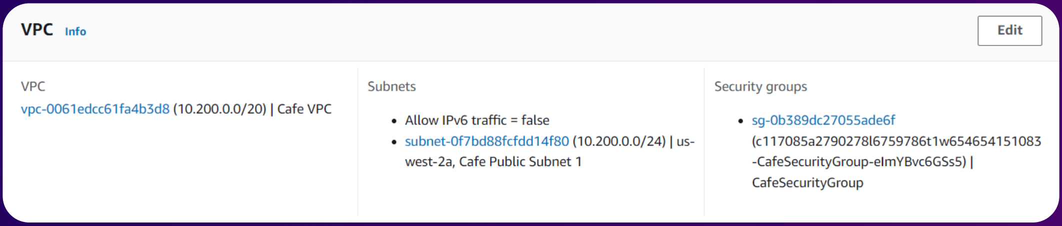
aws re/start

Task 3

Testing the data extractor Lambda function

Step 5: Analyze the issue

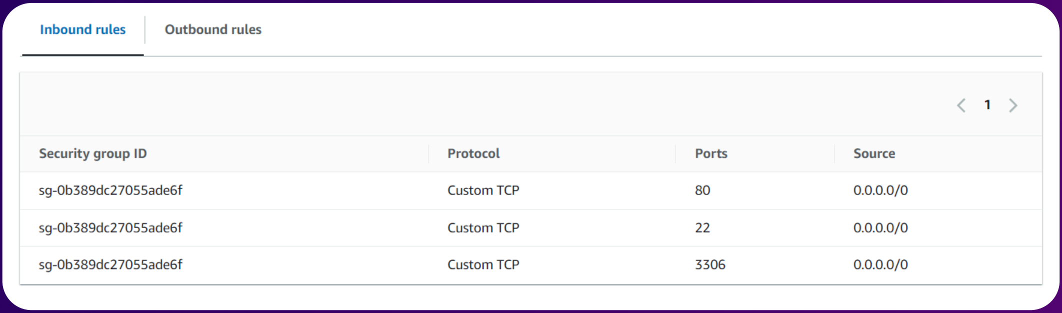
One of the first things that this function does is connect to the MySQL database running in a separate EC2 instance. It waits a certain amount of time to establish a successful connection. After this time passes, if the connection is unsuccessful, the function times out. By default, a MySQL database uses the MySQL protocol and listens on port number 3306 for client access. Choose the **Configuration** tab, and choose **VPC**.



VPC	Subnets	Security groups
vpc-0061edcc61fa4b3d8 (10.200.0.0/20) | Cafe VPC	• Allow IPv6 traffic = false • subnet-0f7bd88fcfdd14f80 (10.200.0.0/24) | us-west-2a, Cafe Public Subnet 1	• sg-0b389dc27055ade6f (c117085a2790278l6759786t1w654654151083-CafeSecurityGroup-eImYBvc6GSs5) | CafeSecurityGroup

Step 6: Correct the issue

Add an inbound rule to the associated security group to permit MySQL traffic on port number 3306.



Security group ID	Protocol	Ports	Source
sg-0b389dc27055ade6f	Custom TCP	80	0.0.0.0/0
sg-0b389dc27055ade6f	Custom TCP	22	0.0.0.0/0
sg-0b389dc27055ade6f	Custom TCP	3306	0.0.0.0/0

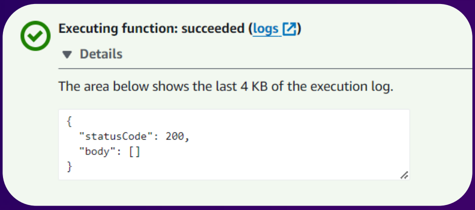
aws re/start

Task 3

Testing the data extractor Lambda function

Step 7: Test the function

Return to the **salesAnalysisReportDataExtractor** function page. Choose the **Test** tab, and choose Test again. Notice the message Executing function: succeeded. The function ran successfully. Notice that the body field of the returned JSON object, which contains the report data that the function extracted, is empty because there is no order data in the database.



Executing function: succeeded (logs)

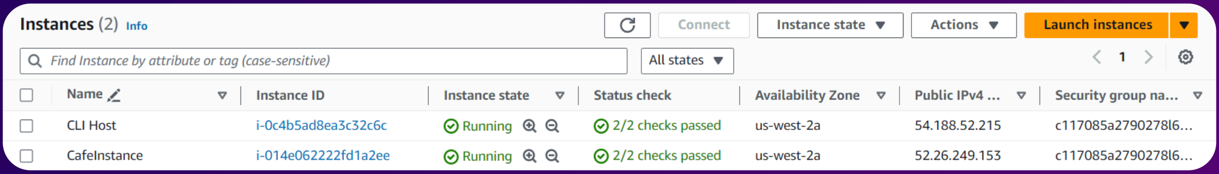
▼ Details

The area below shows the last 4 KB of the execution log.

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

Step 8: Retrieve the CafeInstance IP

Navigate to the **Instances** Section, and make note of the **CafeInstance** Public IPv4 address.



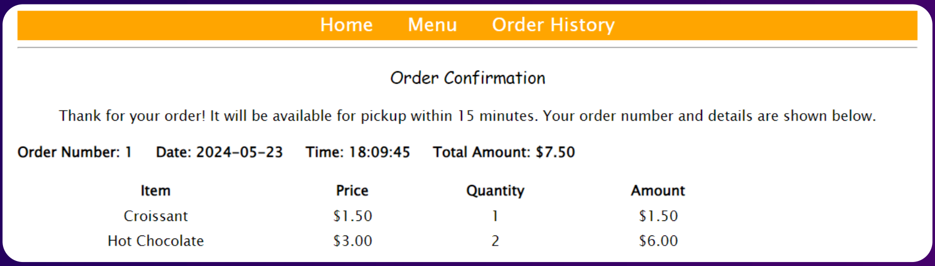
Name	Instance ID	Instance state	Status check	Availability Zone	Public IPv4 ...	Security group na...
CLI Host	i-0c4b5ad8ea3c32c6c	Running	2/2 checks passed	us-west-2a	54.188.52.215	c117085a2790278l6...
CafeInstance	i-014e062222fd1a2ee	Running	2/2 checks passed	us-west-2a	52.26.249.153	c117085a2790278l6...

Task 3

Testing the data extractor Lambda function

Step 9: Place an order

Access the café website using the Public IPv4 address of the **CafeInstance**. On the café website, choose **Menu**, and place some orders to populate data in the database. Now that there is order data in the database, you test the function again.



Order Confirmation

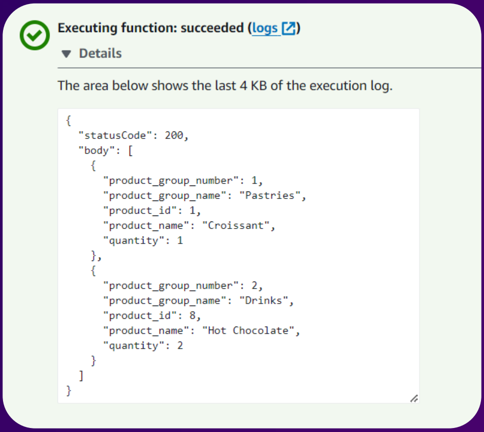
Thank for your order! It will be available for pickup within 15 minutes. Your order number and details are shown below.

Order Number: 1 **Date: 2024-05-23** **Time: 18:09:45** **Total Amount: $7.50**

Item	Price	Quantity	Amount
Croissant	$1.50	1	$1.50
Hot Chocolate	$3.00	2	$6.00

Step 10: Test the function again

Return to the **salesAnalysisReportDataExtractor** function page. Choose the **Test** tab, and choose Test. The returned JSON object now contains product quantity information in the body field.



Executing function: succeeded (logs)

The area below shows the last 4 KB of the execution log.

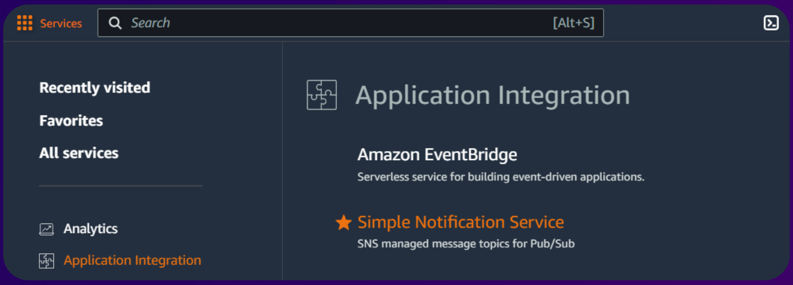
```
{
  "statusCode": 200,
  "body": [
    {
      "product_group_number": 1,
      "product_group_name": "Pastries",
      "product_id": 1,
      "product_name": "Croissant",
      "quantity": 1
    },
    {
      "product_group_number": 2,
      "product_group_name": "Drinks",
      "product_id": 8,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

Task 4

Configuring notifications

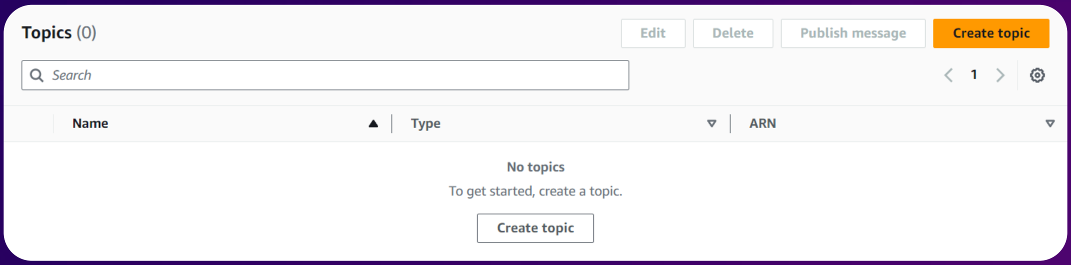
Step 1: Access the Simple Notification Service

In the AWS Management Console, select Simple Notification Service.



Step 2: Create topic

Navigate to the **Topics** section, and select Create topic.



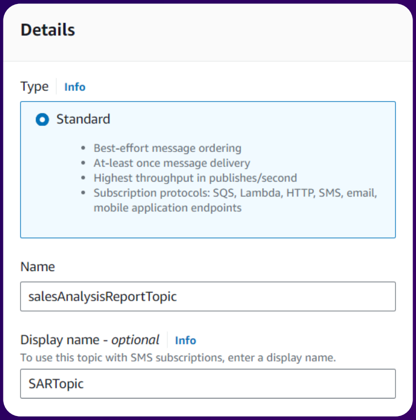
aws re/start

Task 4

Configuring notifications

Step 3: Topic Details

In the **Details** section, configure the following settings.



Details

Type Info

Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

salesAnalysisReportTopic

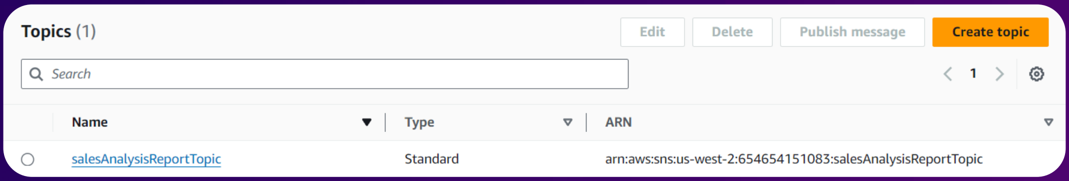
Display name - *optional* Info

To use this topic with SMS subscriptions, enter a display name.

SARTopic

Step 4: Review Topic Creation

Review the newly created **salesAnalysisReportTopic**, and make note of the ARN value.



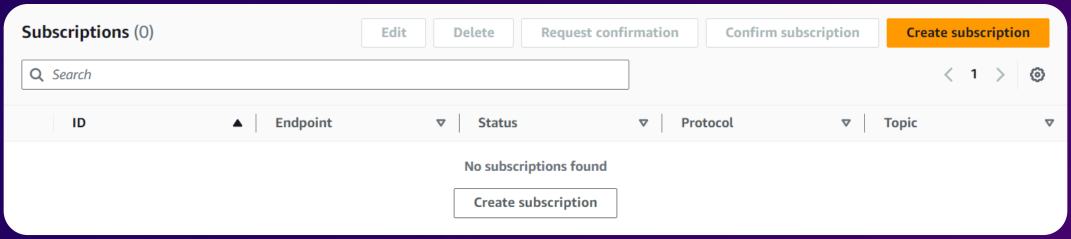
Name	Type	ARN
salesAnalysisReportTopic	Standard	arn:aws:sns:us-west-2:654654151083:salesAnalysisReportTopic

Task 4

Configuring notifications

Step 5: Create subscription

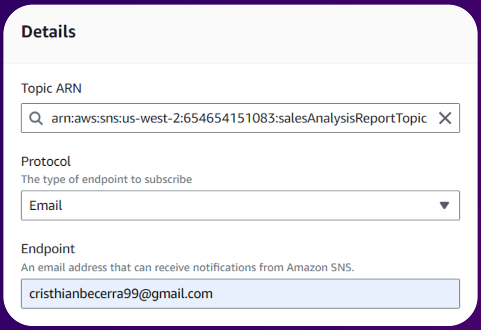
Navigate to the **Subscriptions** section, and select Create subscription.



No subscriptions found

Step 6: Subscription Details

In the **Details** section, configure the following settings.



Details

Topic ARN
arn:aws:sns:us-west-2:654654151083:salesAnalysisReportTopic

Protocol
The type of endpoint to subscribe
Email

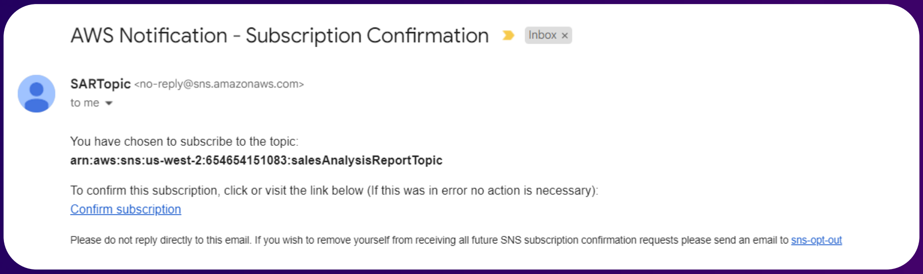
Endpoint
An email address that can receive notifications from Amazon SNS.
cristhianbecerra99@gmail.com

Task 4

Configuring notifications

Step 7: Check your inbox

Check the inbox for the email address that you provided. You should see an email from SARTopic with the subject "AWS Notification - Subscription Confirmation."



AWS Notification - Subscription Confirmation

SARTopic <no-reply@sns.amazonaws.com>
to me

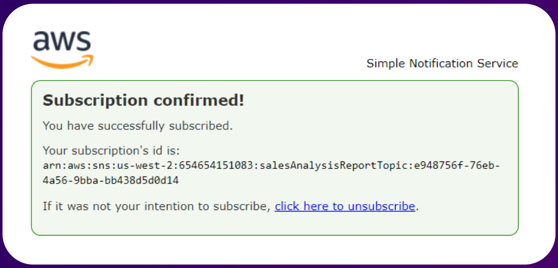
You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:654654151083:salesAnalysisReportTopic

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out

Step 8: Confirm subscription

Choose Confirm subscription. A new browser tab opens and displays a page with the message "Subscription confirmed!".



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
```
arn:aws:sns:us-west-2:654654151083:salesAnalysisReportTopic:e948756f-76eb-
4a56-9bba-bb438d5d0d14
```

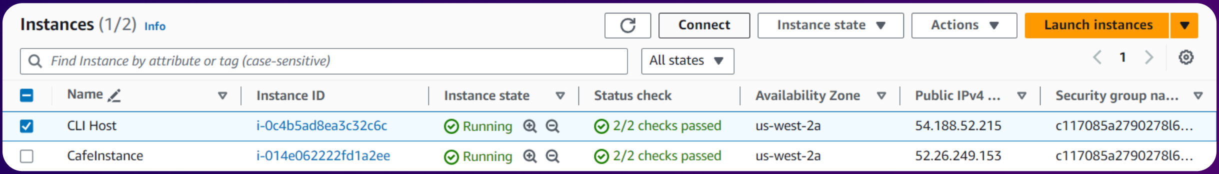
If it was not your intention to subscribe, click here to unsubscribe.

Task 5

Creating the salesAnalysisReport Lambda function

Step 1: Connect to the CLI Host instance

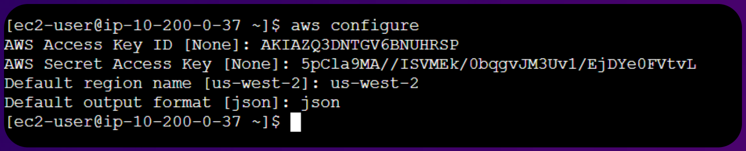
On the EC2 Management Console, navigate to the **Instances** section. Select the **CLI Host** instance, and connect to the instance using **EC2 Instance Connect**.



	Name	Instance ID	Instance state	Status check	Availability Zone	Public IPv4 ...	Security group na...
☑	CLI Host	i-0c4b5ad8ea3c32c6c	Running	2/2 checks passed	us-west-2a	54.188.52.215	c117085a2790278l6...
☐	CafeInstance	i-014e062222fd1a2ee	Running	2/2 checks passed	us-west-2a	52.26.249.153	c117085a2790278l6...

Step 2: Configure the AWS CLI

In the EC2 Instance Connect terminal window, run the aws configure command to update the AWS CLI software with the credentials. At the prompts, enter the following information.



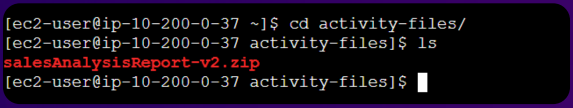
```
[ec2-user@ip-10-200-0-37 ~]$ aws configure
AWS Access Key ID [None]: AKIAZQ3DNTGV6BNUHRSP
AWS Secret Access Key [None]: 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
Default region name [us-west-2]: us-west-2
Default output format [json]: json
[ec2-user@ip-10-200-0-37 ~]$ █
```

Task 5

Creating the salesAnalysisReport Lambda function

Step 3: Verify code files

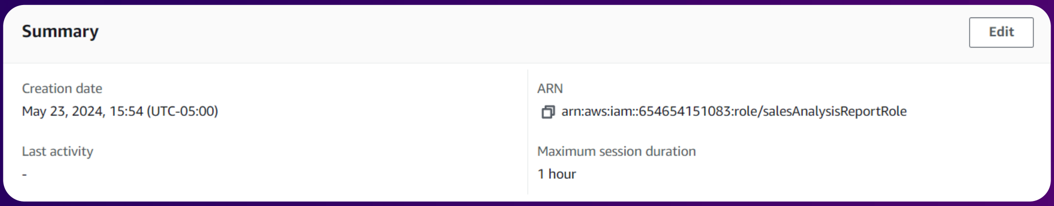
To verify that the salesAnalysisReport-v2.zip file containing the code for the **salesAnalysisReport** Lambda function is already on the **CLI Host**, run the following commands in the terminal.



```
[ec2-user@ip-10-200-0-37 ~]$ cd activity-files/
[ec2-user@ip-10-200-0-37 activity-files]$ ls
salesAnalysisReport-v2.zip
[ec2-user@ip-10-200-0-37 activity-files]$
```

Step 4: Retrieve the ARN of an IAM role

Open the IAM management console, make note of the ARN value for the **salesAnalysisReportRole** role.



Summary

Creation date
May 23, 2024, 15:54 (UTC-05:00)

ARN
arn:aws:iam::654654151083:role/salesAnalysisReportRole

Last activity
-

Maximum session duration
1 hour

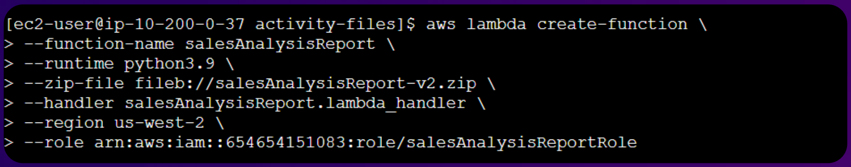
aws re/start

Task 5

Creating the salesAnalysisReport Lambda function

Step 5: Create the Lambda function

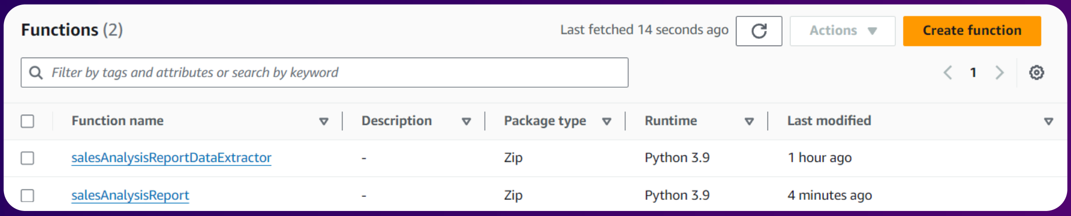
Use the aws lambda create-function command to create the Lambda function. Once the command completes, it returns a JSON object describing the attributes of the function.



```
[ec2-user@ip-10-200-0-37 activity-files]$ aws lambda create-function \
> --function-name salesAnalysisReport \
> --runtime python3.9 \
> --zip-file fileb://salesAnalysisReport-v2.zip \
> --handler salesAnalysisReport.lambda_handler \
> --region us-west-2 \
> --role arn:aws:iam::654654151083:role/salesAnalysisReportRole
```

Step 6: Review Functions

Open the Lambda management console, navigate to the **Functions** section, and select the **salesAnalysisReport** function.


Functions (2)
Last fetched 14 seconds ago
Actions
Create function
Filter by tags and attributes or search by keyword
1

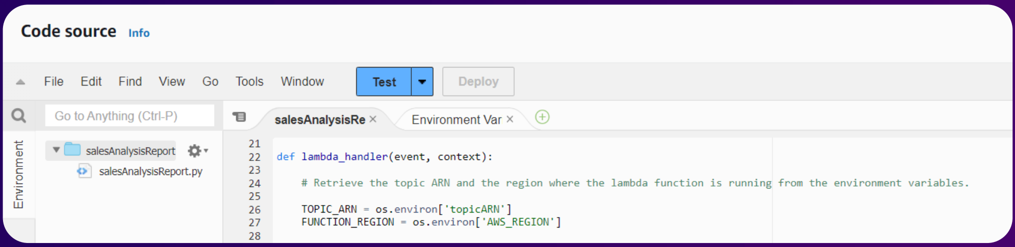
	Function name	Description	Package type	Runtime	Last modified
☐	salesAnalysisReportDataExtractor	-	Zip	Python 3.9	1 hour ago
☐	salesAnalysisReport	-	Zip	Python 3.9	4 minutes ago

Task 5

Creating the salesAnalysisReport Lambda function

Step 7: Review Code source

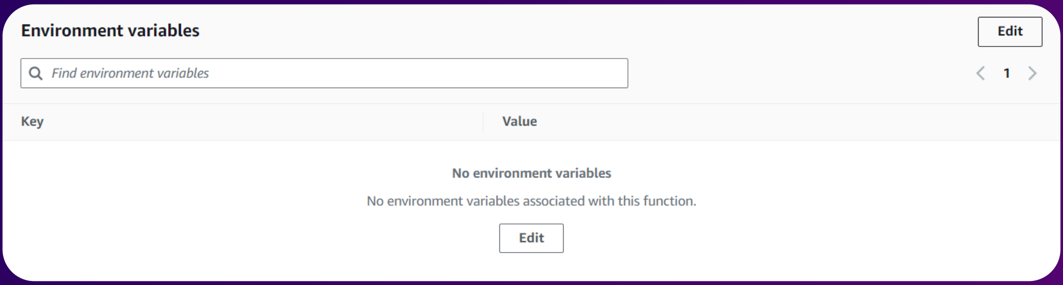
Review the details in the **Code source** panel for the created function. Notice that the function retrieves the ARN of the topic to publish to, from an environment variable named topicARN. Therefore, you need to define that variable in the **Environment variables** panel.



```

def lambda_handler(event, context):

    # Retrieve the topic ARN and the region where the lambda function is running from the environment variables.

    TOPIC_ARN = os.environ['topicARN']
    FUNCTION_REGION = os.environ['AWS_REGION']

```

Step 8: Edit Enviroment variables

Choose the **Configuration** tab, and choose **Environment variables**. Choose Edit.



Environment variables

Edit

Find environment variables

Key	Value

No environment variables

No environment variables associated with this function.

Edit

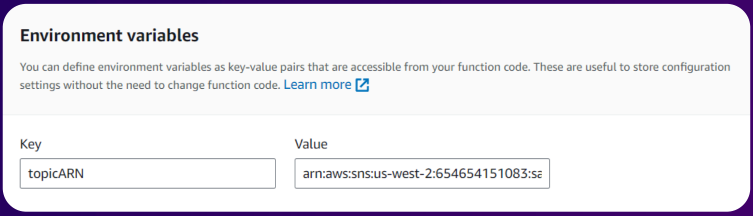
aws re/start

Task 5

Creating the salesAnalysisReport Lambda function

Step 9: Environment variables

In the **Environment variables** section, configure the following options.



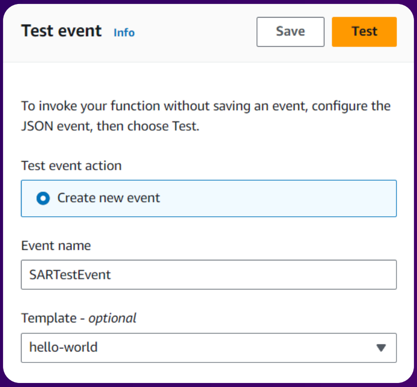
Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more

Key	Value
topicARN	arn:aws:sns:us-west-2:654654151083:sa

Step 10: Test event

Choose the **Test** tab, and configure the test event as follows. Then, choose Save and Test.



Test event Info

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action: (●) Create new event

Event name: SARTestEvent

Template - *optional*: hello-world

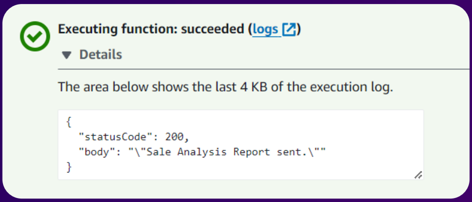
aws re/start

Task 5

Creating the salesAnalysisReport Lambda function

Step 11: Test the function

The message Executing function: succeeded appears. If you get a timeout error, choose the Test button again. Sometimes, when you first run a function, it takes a little longer to initialize, and the Lambda default timeout value (3 seconds) is exceeded. Usually, you can run the function again, and the error will go away. Alternatively, you can increase the timeout value.


Executing function: succeeded (logs)

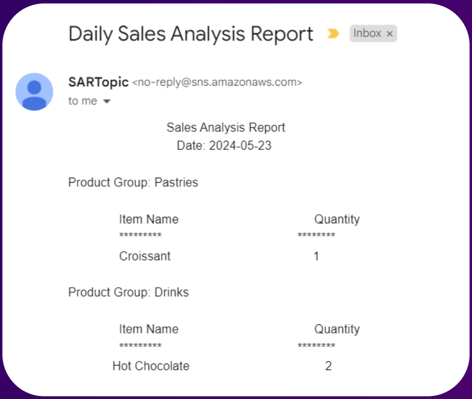
Details

The area below shows the last 4 KB of the execution log.

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

Step 12: Check your email inbox

If there were no errors, you should receive an email from AWS Notifications with the subject "Daily Sales Analysis Report."


Daily Sales Analysis Report Inbox ×

SARTopic <no-reply@sns.amazonaws.com>
to me

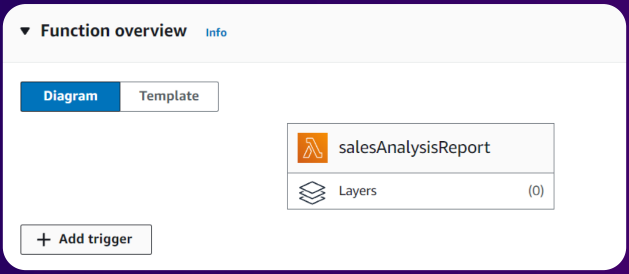
```
                Sales Analysis Report
                  Date: 2024-05-23

Product Group: Pastries

        Item Name                 Quantity
        *********                ********
        Croissant                   1

Product Group: Drinks

        Item Name                 Quantity
        *********                ********
       Hot Chocolate                2
```

Task 5

Creating the salesAnalysisReport Lambda function

Step 13: Add trigger

In the **Function overview** panel, choose Add trigger.



▼ **Function overview** Info

Diagram | Template

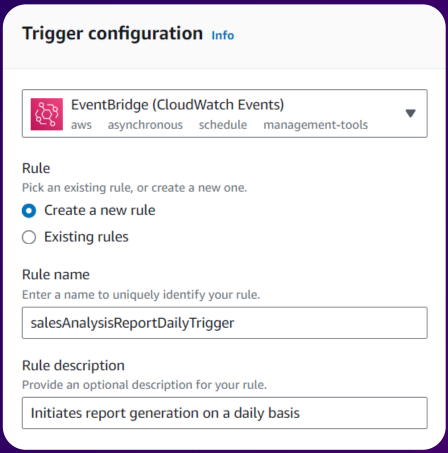
salesAnalysisReport

Layers (0)

+ **Add trigger**

Step 14: Trigger configuration

In the **Trigger configuration** section, configure the following settings.



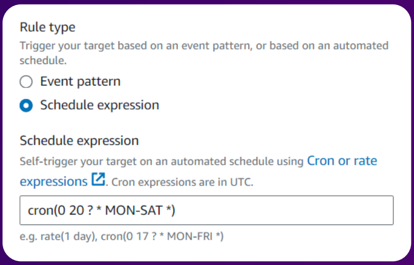
Trigger configuration Info

EventBridge (CloudWatch Events)
aws asynchronous schedule management-tools

Rule
Pick an existing rule, or create a new one.
● Create a new rule
○ Existing rules

Rule name
Enter a name to uniquely identify your rule.
salesAnalysisReportDailyTrigger

Rule description
Provide an optional description for your rule.
Initiates report generation on a daily basis



Rule type
Trigger your target based on an event pattern, or based on an automated schedule.
○ Event pattern
● Schedule expression

Schedule expression
Self-trigger your target on an automated schedule using Cron or rate expressions. Cron expressions are in UTC.
cron(0 20 ? * MON-SAT *)
e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Task 5

Creating the salesAnalysisReport Lambda function

Step 15: Review Function overview

The new trigger is created and displayed in the **Function overview** panel.

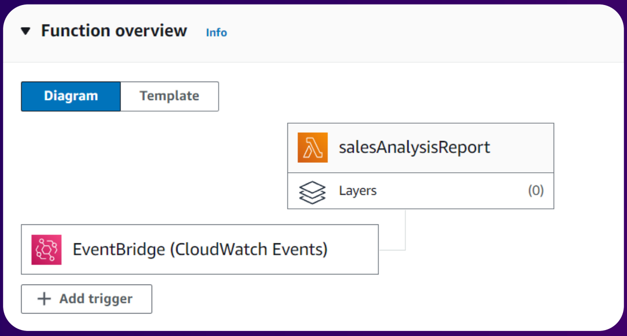

▼ Function overview Info

Diagram | Template

salesAnalysisReport

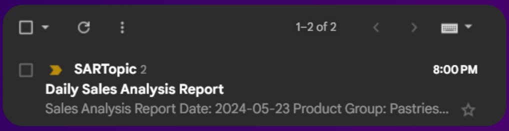
Layers (0)

EventBridge (CloudWatch Events)

+ Add trigger

Step 16: Check your email inbox

If there were no errors, you should see a new email from AWS Notifications with a subject of "Daily Sales Analysis Report." The CloudWatch Events event invoked this message at the time that you specified in the Cron expression.


1–2 of 2

SARTopic 2 8:00 PM
Daily Sales Analysis Report
Sales Analysis Report Date: 2024-05-23 Product Group: Pastries...

aws re/start

Conclusions

AWS Lambda
AWS Lambda allows you to run code without provisioning or managing servers, enabling scalable and cost-effective computing with automatic scaling and high availability.

Lambda Function
A Lambda function is a self-contained piece of code written in a supported language, executed in response to specific triggers, events, or conditions.

Runtime
The runtime provides the execution environment for Lambda functions, including the necessary libraries, dependencies, and runtime languages like Node.js, Python, or Java.

Layers
Layers in AWS Lambda enable you to manage and share common code, libraries, and dependencies across multiple functions, promoting code reuse and efficiency.

Triggers
Triggers are event sources that invoke Lambda functions, such as changes in data state in DynamoDB, updates in an S3 bucket, or messages in an SQS queue, enabling event-driven architecture.

aws re/start

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru

aws re/start