



178- [JAWS] - Lab - Trabajar con AWS Lambda

Datos Generales:

Nombre: Tomás Alfredo Villaseca Constantinescu

País: Chile

Fecha: 06/11/2023

Contacto: tomas.villaseca.c@gmail.com

Después de completar este laboratorio, podrá hacer lo siguiente:

- Reconocer los permisos de política de AWS IAM necesarios para facilitar una función de Lambda a otros recursos de Amazon Web Services (AWS).
- Crear una capa de Lambda para satisfacer una dependencia de biblioteca externa.
- Crear funciones de Lambda que extraigan datos de la base de datos y envíen informes al usuario.
- Implementar y probar una función de Lambda que se inicie en función de una programación y que invoque a otra función.
- Utilizar los registros de CloudWatch para solucionar cualquier problema de ejecución de una función de Lambda.

Resumen Laboratorio:

En este laboratorio, implementará y configurará una solución informática sin servidor basada en AWS Lambda.

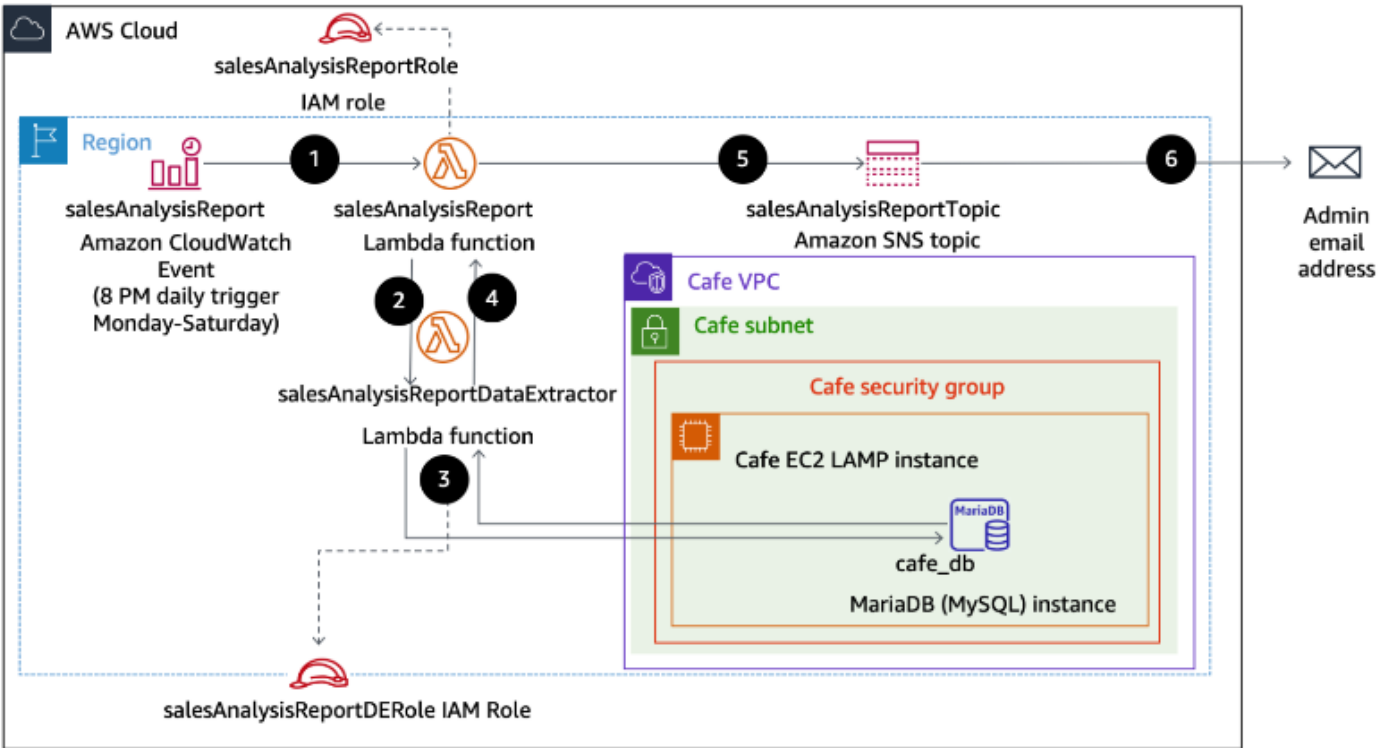
La función de Lambda genera un informe de análisis de ventas extrayendo datos de una base de datos y enviando los resultados por correo electrónico diariamente.

La información de conexión de la base de datos se almacena en Parameter Store, una función de AWS Systems Manager. La base de datos se ejecuta en una instancia de Linux, Apache, MySQL y PHP (LAMP) de Amazon EC2.

En este laboratorio, se le proporciona el código Python de cada función Lambda para que pueda centrarse en las tareas de SysOps de desplegar, configurar y probar los componentes de la solución sin servidor.

Diagrama de la arquitectura a implementar que incluye los pasos de funcionamiento:

Pasos	Detalle
1	Un evento de Amazon CloudWatch Events llama a la función Lambda salesAnalysisReport a las 20:00 horas todos los días de lunes a sábado.
2	La función Lambda salesAnalysisReport invoca otra función Lambda, salesAnalysisReportDataExtractor, para recuperar los datos del informe.
3	La función salesAnalysisReportDataExtractor ejecuta una consulta analítica en la base de datos de la cafetería (cafe_db).
4	El resultado de la consulta se devuelve a la función salesAnalysisReport.
5	La función salesAnalysisReport formatea el informe en un mensaje y lo publica en el topic salesAnalysisReportTopic de Amazon SNS.
6	El topic SNS salesAnalysisReportTopic envía el mensaje por correo electrónico al administrador.



Tarea 1: Observar la configuración del rol de IAM

En este laboratorio, creará dos funciones de Lambda. Cada función requiere permisos para acceder a los recursos de AWS con los que interactúan.

En esta tarea, analizará las funciones de IAM y los permisos que conceden a las funciones de Lambda que creará más adelante.

Paso 1: IAM → Roles → salesAnalysisReportRole

<input checked="" type="checkbox"/>	Role name	Trusted entities
<input checked="" type="checkbox"/>	salesAnalysisReportRole	AWS Service: lambda

Paso 2: salesAnalysisReportRole → Trust relationships

Permissions

Trust relationships

Tags (1)

Access A

Trusted entities

Entities that can assume this role under specified conditions.

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "lambda.amazonaws.com"
8       },
9       "Action": "sts:AssumeRole"
10    }
11  ]
12 }
```

Paso 3: salesAnalysisReportRole → Permissions

- AmazonSNSFullAccess → Proporciona acceso completo a recursos de SNS
- AmazonSSMReadOnlyAccess → Proporciona acceso de lectura a recursos Systems Manager
- AWSLambdaBasicRunRole → Proporciona permisos de escritura para CloudWatch Logs.
- AWSLambdaRole → Entrega la habilidad a una función lambda de invocar otra función lambda.

✓	Policy name ↗
✓	+ AmazonSNSFullAccess
✓	+ AmazonSSMReadOnlyAccess
✓	+ AWSLambdaBasicRunRole
✓	+ AWSLambdaRole

Paso 4: IAM → Roles → salesAnalysisReportDERole

✓	Role name	Trusted entities
✓	salesAnalysisReportDERole	AWS Service: lambda

Paso 5: salesAnalysisReportDERole → Trust relationships

Permissions
Trust relationships
Tags (1)
Access

Trusted entities

Entities that can assume this role under specified conditions.

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "lambda.amazonaws.com"
8       },
9       "Action": "sts:AssumeRole"
10    }
11  ]
12 }
```

Paso 6: salesAnalysisReportDERole → Permissions

- AWSLambdaBasicRunRole → Proporciona permisos de escritura para CloudWatch Logs.
- AWSLambdaVPCLambdaAccessRunRole → Proporciona permisos para administrar NICs para conectar una función lambda con una VPC.

✓	Policy name ↗
✓	+ AWSLambdaBasicRunRole
✓	+ AWSLambdaVPCLambdaAccessRunRole

Tarea 2: Crear una capa de Lambda y una función de Lambda del extractor de datos

En esta tarea, primero se creará una capa Lambda y, a continuación, se creará una función Lambda que utiliza la capa.

- Descargar archivos pymysql-v3.zip y salesAnalysisReportDataExtractor-v3.zip requeridos para realizar el laboratorio.
- salesAnalysisReportDataExtractor-v3.zip → Implementación de Python de una función Lambda que hace uso de la biblioteca PyMySQL para acceder a la base de datos MySQL café.
- Pymysql-v3.zip → biblioteca PyMySQL

Tarea 2.1 – Crear una capa Lambda

Se creará una capa Lambda llamada pymysqlLibrary y se cargará en ella la biblioteca PyMySQL para que pueda ser utilizada por cualquier función que la requiera.

Las capas Lambda proporcionan un mecanismo flexible para reutilizar código entre funciones de forma que no sea necesario incluir el código en el paquete de despliegue de cada función.

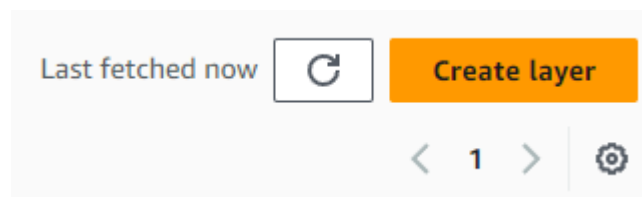
Paso 1: Lambda → Layers → Create Layer

▼ Additional resources

[Code signing configurations](#)

[Layers](#)

[Replicas](#)



Paso 2: Create Layer → Configuration

- Name = pymysqlLibrary
- Description = PyMySQL library modules
- Upload a .zip file → pymysql-v3.zip
- Compatible runtimes → Python 3.9

Name

pymysqlLibrary

Description - *optional*

PyMySQL library modules

- ☒ Upload a .zip file
☐ Upload a file from Amazon S3

 Upload

pymysql-v3.zip
105.45 KB



For files larger than 10 MB, consider uploading using Amazon S3.

Compatible architectures - *optional* [Info](#)

Choose the compatible instruction set architectures for your layer.


- ☐ x86_64
☐ arm64

Compatible runtimes - *optional* [Info](#)


Choose up to 15 runtimes.

Runtimes



Python 3.9 

Paso 3: Create Layer → Create

 Successfully created layer pymysqlLibrary version 1.

Name	Version	Compatible runtimes
pymysqlLibrary	1	python3.9

Tarea 2.2 – Crear una función Lambda extractora de datos

Paso 1: Lambda → Functions → Create Function

AWS Lambda

Dashboard

Applications

Functions

Actions ▼

Create function

Paso 2: Create Function → Settings

- Author from scratch

☒ Author from scratch
Start with a simple Hello World example.

- Function name = salesAnalysisReportDataExtractor

Function name

Enter a name that describes the purpose of your function.

salesAnalysisReportDataExtractor

Use only letters, numbers, hyphens, or underscores with no spaces.

- Runtime → Python 3.9

Runtime [Info](#)

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.9

- Change default execution role → Execution role → Use an existing role → salesAnalysisReportDERole

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- ☐ Create a new role with basic Lambda permissions
- ☒ Use an existing role
- ☐ Create a new role from AWS policy templates

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.

Paso 3: Create Function → Create


<input checked="" type="checkbox"/>	Function name
<input checked="" type="checkbox"/>	salesAnalysisReportDataExtractor


Tarea 2.3 – Agregar la capa Lambda a la función Lambda

Paso 1: salesAnalysisReportDataExtractor → Function Overview → Layers → Add Layer

▼ Function overview

Info

 salesAnalysisReportDataExtract
or

 Layers (0)

Layers

Info

Edit

Add a layer

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					

Paso 2: Add Layer → Settings

- Choose layer → Custom layers
- Custom layers → pymysqlLibrary
- Version → 1

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

▼

Paso 3: Add Layer → Add

Layers Info			
Merge order	Name	Layer version	Compatible runtimes
1	pymysqlLibrary	1	python3.9

Tarea 2.4 – Importar el código para la función Lambda extractora de datos

Paso 1: salesAnalysisReportDataExtractor → Function Overview → Runtime Settings → Edit

Runtime settings [Info](#) Edit

Runtime Python 3.9	Handler Info lambda_function.lambda_handler	Architecture Info x86_64
-----------------------	--	---

► Runtime management configuration

Paso 2: Runtime Settings → Edit

- Handler = salesAnalysisReportDataExtractor.lambda_handler

Handler [Info](#)

salesAnalysisReportDataExtractor.lambda_handler

Paso 3: salesAnalysisReportDataExtractor → Function Overview → Code Source

- Upload from → .zip file → salesAnalysisReportDataExtractor-v3.zip

Code | Test | Monitor | Configuration | Aliases | Versions

Code source [Info](#) Upload from ▼

Upload from ▲

.zip file

Amazon S3 location

Upload a .zip file

ⓘ When you upload a new .zip file package, it overwrites the existing code.

Upload

salesAnalysisReportDataExtractor-v3.zip
0.79 KB

×

For files larger than 10 MB, consider uploading using Amazon S3.

El código de la función Lambda es importado y se muestra en el panel Code Source.

Paso 4: salesAnalysisReportDataExtractor → Function Overview → Code Source → Enviroment

- Revisar el código de Python que implementa la función.

```
1 import boto3
2 import pymysql
3 import sys
4
5 def lambda_handler(event, context):
6
7     # Retrieve the database connection information from the event input parameter.
8
9     dbUrl = event['dbUrl']
10    dbName = event['dbName']
11    dbUser = event['dbUser']
12    dbPassword = event['dbPassword']
13
14    # Establish a connection to the Cafe database, and set the cursor to return results as a Python dictionary.
15
16    try:
17        conn = pymysql.connect(host=dbUrl, user=dbUser, passwd=dbPassword, db=dbName, cursorclass=pymysql.cursors.DictCursor)
18
19    except pymysql.Error as e:
20        print('ERROR: Failed to connect to the Cafe database.')
21        print('Error Details: %d %s' % (e.args[0], e.args[1]))
22        sys.exit()
23
24    # Execute the query to generate the daily sales analysis result set.
25
26    with conn.cursor() as cur:
27        cur.execute("SELECT c.product_group_number, c.product_group_name, a.product_id, b.product_name, CAST(sum(a.quantity) AS
28        result = cur.fetchall()
29
30    # Close the connection.
31
32    conn.close()
33
34    # Return the result set.
35
36    return {'statusCode': 200, 'body': result}
```

Tarea 2.5 – Configurar los ajustes de red para la función Lambda

Como muestra el diagrama de arquitectura al principio de este laboratorio, esta función Lambda requiere acceso de red a la base de datos de la cafetería, que se ejecuta en una instancia LAMP de EC2. Debe especificar la VPC de la instancia, la subred y la información del grupo de seguridad en la configuración de la función.

Paso 1: salesAnalysisReportDataExtractor → Configuration → VPC → Edit

Code	Test	Monitor	Configuration	Aliases	Versions
------	------	---------	---------------	---------	----------

Tags

VPC

Monitoring and operations tools

No VPC configuration

This function isn't connected to a VPC.

Edit

Paso 2: VPC → Edit

- VPC → Café VPC

```
vpc-001b834121f730d39 (10.200.0.0/20)
aws:cloudformation:logical-id: VPC
aws:cloudformation:stack-id: arn:aws:cloudformation:us-west-2:924376459018:stack/c94303a2111800l5186593t1w924376459018/62a0bef0-7c4a-11ee-ae1b-0a29fbb185cb
aws:cloudformation:stack-name: c94303a2111800l5186593t1w924376459018
cloudlab: c94303a2111800l5186593t1w924376459018   Name: Cafe VPC
VPC: Cafe
```



- Subnets → Café Public Subnet 1

```
subnet-0592f76c7431f2c6c (10.200.0.0/24) us-west-2a
aws:cloudformation:logical-id: PublicSubnet1
aws:cloudformation:stack-id: arn:aws:cloudformation:us-west-2:924376459018:stack/c94303a2111800l5186593t1w924376459018/62a0bef0-7c4a-11ee-ae1b-0a29fbb185cb
aws:cloudformation:stack-name: c94303a2111800l5186593t1w924376459018
cloudlab: c94303a2111800l5186593t1w924376459018   Name: Cafe Public Subnet 1
```

- Security Groups → CafeSecurityGroup

```
sg-0b507acc147f33c9e (c94303a2111800l5186593t1w924376459018-CafeSecurityGroup-1P3R2Z4ZROMME)
Security group for the Cafe instance
aws:cloudformation:logical-id: CafeSecurityGroup
aws:cloudformation:stack-id: arn:aws:cloudformation:us-west-2:924376459018:stack/c94303a2111800l5186593t1w924376459018/62a0bef0-7c4a-11ee-ae1b-0a29fbb185cb
aws:cloudformation:stack-name: c94303a2111800l5186593t1w924376459018
cloudlab: c94303a2111800l5186593t1w924376459018   Name: CafeSecurityGroup
```

Tarea 3: Probar la función de Lambda del extractor de datos

Tarea 3.1 – Realizar un Test de la función Lambda

Probará la función salesAnalysisReportDataExtractor. Para invocarla, debe proporcionar los valores de los parámetros de conexión a la base de datos de la cafetería. Recuerde que éstos se almacenan en Parameter Store.

Paso 1: Systems Manager → Application Management → Parameter Store

▼ Application Management

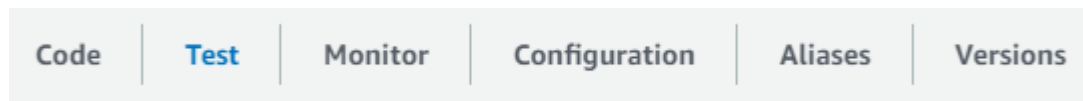
Application Manager

AppConfig

Parameter Store

- Copiar y pegar el valor de los siguientes parámetros:
- /café/dbUrl → ec2-52-36-89-70.us-west-2.compute.amazonaws.com
- /café/dbName → cafe_db
- /café/dbUser → root
- /café/dbPassword → Re:Start!9

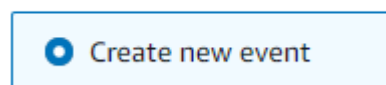
Paso 2: salesAnalysisReportDataExtractor → Test



Paso 3: Test → Test Event

- Test Event Action → Create new event

Test event action



- Event name → SARDTestEvent

Event name

SARDTestEvent

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

- Template → hello-world

Template - optional

hello-world

- Event JSON → Reemplazar el JSON object con el object entregado por el laboratorio (ingresar los valores copiados anteriormente en Parameter Store).

Event JSON

```
1 {
2   "dbUrl": "ec2-52-36-89-70.us-west-2.compute.amazonaws.com",
3   "dbName": "cafe_db",
4   "dbUser": "root",
5   "dbPassword": "Re:Start!9"
6 }
```

Paso 4: Save → Test

- Resultado de la prueba → "Execution result: failed"



Executing function: failed ([logs](#))



[Details](#)

Tarea 3.2 – Troubleshooting de la función Lambda

Paso 1: salesAnalysisReportDataExtractor → Execution results → Details

```
{
  "errorMessage": "2023-11-06T02:42:09.800Z 30f9f4df-3111-4101-a58b-3123a31ad702 Task timed out after 3.01 seconds"
}
```

Mensaje indica que la función ha expirado después de 3 segundos.

Log Output:

- START → Indica que la función comenzó su ejecución.
- END → Indica que la función finalizó su ejecución.
- REPORT → Proporciona un resumen de las estadísticas de rendimiento y la utilización de recursos.

Log output

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

```
START RequestId: 30f9f4df-3111-4101-a58b-3123a31ad702 Version: $LATEST
2023-11-06T02:42:09.800Z 30f9f4df-3111-4101-a58b-3123a31ad702 Task timed out after 3.01 seconds
```

```
END RequestId: 30f9f4df-3111-4101-a58b-3123a31ad702
```

```
REPORT RequestId: 30f9f4df-3111-4101-a58b-3123a31ad702   Duration: 3006.07 ms    Billed Duration: 3000 ms    Memory Size: 128 MB    Max Memory Used: 57 MB    Init Duration: 298.16 ms
```

Tarea 3.3 – Analizar y corregir la función Lambda

En esta tarea, analizará y corregirá el problema observado al probar la función Lambda.

- Función intenta conectarse a la base de datos MySQL, espera una cierta cantidad de tiempo para establecer una conexión exitosa. Una vez transcurrido este tiempo, si la conexión no es exitosa, la función es finalizada.
- Base de datos MySQL utiliza el protocolo MySQL (puerto 3306 para el acceso del cliente).
- Revisar las reglas de entrada del grupo de seguridad asociado.

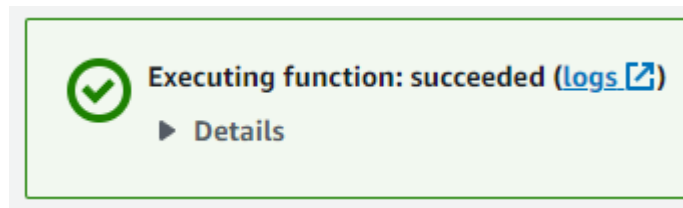
<input checked="" type="checkbox"/> CafeSecurityGroup sg-0b507acc147f33c9e			
Security group ID	Protocol	Ports	Source
sg-0b507acc147f33c9e	Custom TCP	80	0.0.0.0/0
sg-0b507acc147f33c9e	Custom TCP	22	0.0.0.0/0

Edit Inbound Rules

Security group ID	Protocol	Ports	Source
sg-0b507acc147f33c9e	Custom TCP	80	0.0.0.0/0
sg-0b507acc147f33c9e	Custom TCP	22	0.0.0.0/0
sg-0b507acc147f33c9e	Custom TCP	3306	0.0.0.0/0

Paso 1: salesAnalysisReportDataExtractor → Test → Test Again

- Resultado de la prueba → “Execution result: succeeded (logs)”



Paso 2: salesAnalysisReportDataExtractor → Execution results → Details

- La función entrega el siguiente JSON object:
- Body está vacío porque no hay datos de pedidos.








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

Tarea 3.4 – Hacer un pedido y Testear nuevamente

Accederá al sitio web de la cafetería y realizará algunos pedidos para rellenar los datos de la base de datos.

Paso 1: Abrir una nueva pestaña de navegador web e ingresar URL de página web de la cafetería.



- <http://publicIP/cafe> → Reemplazar publicIP con el IPv4 de CaféInstance EC2

	Name 	Instance ID	Instance state 
	CafeInstance	i-09002364bacdfa751	 Running  

Public IPv4 address

 52.36.89.70 | [open address](#) 

52.36.89.70/cafe/

 RRSS  Anime  Política  Pagos  Trabajo  Cyber Security

Café

[Home](#) [About Us](#) [Contact Us](#) [Menu](#) [Order History](#)

Paso 2: Página web → Menu

- Realizar pedidos para llenar la base de datos.

Order Number: 1 Date: 2023-11-05 Time: 21:49:09 Total Amount: \$47.50

Item	Price	Quantity	Amount
Croissant	\$1.50	10	\$15.00
Donut	\$1.00	5	\$5.00
Chocolate Chip Cookie	\$2.50	5	\$12.50
Hot Chocolate	\$3.00	5	\$15.00

Paso 3: salesAnalysisReportDataExtractor → Test → Test Again

- La función entrega el siguiente JSON object:

```
{
  "statusCode": 200,
  "body": [
    {
      "product_group_number": 1,
      "product_group_name": "Pastries",
      "product_id": 1,
      "product_name": "Croissant",
      "quantity": 10
    },
  ],
}
```


Tarea 4: Configurar las notificaciones

En esta tarea, se creará un SNS Topic y luego se suscribirá una dirección de correo electrónico al Topic.

Paso 1: SNS → Topic → Create Topic

- Type → Standard

Type [Info](#)

Topic type cannot be modified after topic is created

☐ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

☒ 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
- DisplayName = SARTopic

Name

salesAnalysisReportTopic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional* [Info](#)

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.

SARTopic

Maximum 100 characters.

Paso 2: SNS → Topic → Create Topic → Create

- ARN → arn:aws:sns:us-west-2:924376459018:salesAnalysisReportTopic

Paso 3: salesAnalysisReportTopic → Create Subscription

- Protocol → Email
- Endpoint = tomas.villaseca.c@gmail.com

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

tomas.villaseca.c@gmail.com

Paso 4: Revisar correo electrónico → Confirmar suscripción

SARTopic

para mí ▼

You have chosen to subscribe to the topic:

`arn:aws:sns:us-west-2:924376459018: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](#)



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

`arn:aws:sns:us-west-2:924376459018:salesAnalysisReportTopic:11801529-738d-44b1-9a80-0b4da45e6da3`

If it was not your intention to subscribe, [click here to unsubscribe](#).

Tarea 5: Crear una función de Lambda del informe de análisis de ventas

En esta tarea, creará y configurará la función Lambda salesAnalysisReport. Esta función es el motor principal del flujo del informe de análisis de ventas.

- Recupera la información de conexión a la base de datos desde Parameter Store.
- Invoca la función Lambda salesAnalysisReportDataExtractor, que recupera los datos del informe de la base de datos.
- Formatea y publica un mensaje que contiene los datos del informe en el SNS topic.

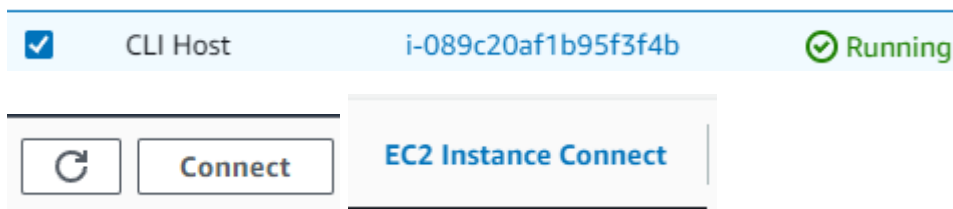
En esta tarea, se utilizará EC2 Instance Connect para iniciar sesión en la Host Instance que se ejecuta en la cuenta de AWS que ya tiene instalada la CLI de AWS y el código Python necesario para crear la siguiente función de Lambda.

Ejecutará un comando de la CLI de AWS para crear la función de Lambda.

Por último, probará la nueva función utilizando la consola de administración de Lambda.

Tarea 5.1 – Conectarse a la CLI Host Instance

Paso 1: EC2 → Instances → CLI Host → Connect



```
#_
~\  #####
~~ \_#####\
~~  \###|
~~   \#/
~~    v~' '->
   ~~~
   ~~-./
   _/m/'

Amazon Linux 2

AL2 End of Life is 2025-06-30.

A newer version of Amazon Linux is available!

Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-200-0-98 ~]$
```

Tarea 5.2 – Configurar la CLI de AWS

Paso 1: Ejecutar el comando **aws configure** para ingresar las credenciales de la CLI (puede encontrar las credenciales en “Details” del laboratorio).

```
[ec2-user@ip-10-200-0-98 ~]$ aws configure
AWS Access Key ID [None]: AKIA5OOJBM4FO6U76JNR
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]: json
[ec2-user@ip-10-200-0-98 ~]$
```

Tarea 5.3 – Crear la función Lambda salesAnalysisReport usando la CLI

Paso 1: Verificar que el archivo salesAnalysisReport-v2.zip se encuentra en la instancia ingresando el siguiente comando:

```
cd activity-files
ls
```

```
[ec2-user@ip-10-200-0-98 ~]$ cd activity-files
[ec2-user@ip-10-200-0-98 activity-files]$ ls
salesAnalysisReport-v2.zip
[ec2-user@ip-10-200-0-98 activity-files]$
```

Paso 2: IAM → Roles → salesAnalysisReportRole → Summary

- ARN → arn:aws:iam::924376459018:role/salesAnalysisReportRole

Paso 3: Crear función Lambda salesAnalysisReport ingresando el siguiente comando:

- Reemplazar <región> con la región utilizada en el laboratorio.
- Reemplazar <salesAnalysisReportRoleARN> con el ARN copiado anteriormente.

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

```
[ec2-user@ip-10-200-0-98 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::924376459018:role/salesAnalysisReportRole
{
  "FunctionName": "salesAnalysisReport",
  "LastModified": "2023-11-06T03:00:46.848+0000",
  "RevisionId": "a54645af-ac1c-4660-8664-b56ae640f486",
  "MemorySize": 128,
  "State": "Pending",
  "Version": "$LATEST",
  "Role": "arn:aws:iam::924376459018:role/salesAnalysisReportRole",
  "Timeout": 3,
  "StateReason": "The function is being created.",
  "Runtime": "python3.9",
  "StateReasonCode": "Creating",
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "CodeSha256": "FOQaNphpQr/canEnzctygYFVreHKiABxYNh8X8lOpnE=",
  "Description": "",
  "CodeSize": 1643,
  "FunctionArn": "arn:aws:lambda:us-west-2:924376459018:function:salesAnalysisReport",
  "Handler": "salesAnalysisReport.lambda_handler"
}
[ec2-user@ip-10-200-0-98 activity-files]$
```

Una vez que el comando se completa, se devuelve un JSON object que describe los atributos de la función.

Tarea 5.4 – Configurar la función Lambda salesAnalysisReport

Paso 1: Lambda → Functions → salesAnalysisReport

	Function name
<input checked="" type="checkbox"/>	salesAnalysisReport

Paso 2: salesAnalysisReport → Configuration → Environment Variables → Edit

No environment variables

No environment variables associated with this function.

Edit

- Add environment variable
- Key = topicARN
- Value = Pegar el ARN de salesAnalysisReportTopic copiado anteriormente.

Key	Value
topicARN	arn:aws:sns:us-west-2:924376459018:salesAnalysisReportTopic
Add environment variable	

Key	Value
topicARN	arn:aws:sns:us-west-2:924376459018:salesAnalysisReportTopic

Tarea 5.5 – Testear la función Lambda salesAnalysisReport

Paso 1: salesAnalysisReport → Test

- Test event action → Create new event
- Event name = SARTestEvent

Test event action

☒ Create new event

Event name

SARTestEvent

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

- Template → hello-world

Template - *optional*

hello-world

Paso 2: salesAnalysisReport → Test

- Resultado del Test → “Execution result: succeeded (logs)”



Executing function: succeeded ([logs](#))

► Details

Paso 3: salesAnalysisReport → Test → Execution Function → Details

- La función Lambda entrega el siguiente JSON object:

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

Paso 4: Revisar correo electrónico → Notificación SNS

SARTopic <no-reply@sns.amazonaws.com>
para tomas.villaseca.c ▼

Sales Analysis Report
Date: 2023-11-06

Product Group: Pastries

Item Name *****	Quantity *****
Croissant	10
Donut	5
Chocolate Chip Cookie	5

Product Group: Drinks


Item Name *****	Quantity *****
Hot Chocolate	5


Tarea 5.6 – Añadir un Trigger a la función Lambda salesAnalysisReport

Para completar la implementación de la función salesAnalysisReport, configurará el informe para que se inicie de lunes a sábado a las 20:00 cada día. Para ello, utilizará un evento de CloudWatch Events como mecanismo de activación.

Paso 1: salesAnalysisReport → Function Overview → Add trigger

▼ Function overview [Info](#)

 **salesAnalysisReport**

 **Layers** (0)

+ Add trigger

Paso 2: Add trigger → Settings

- 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 → Ingresar una cron expression que será ejecutada en los 5 minutos posteriores.

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(20 00 ? * MON-SAT *)

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Paso 3: Add trigger → Add

Triggers (1/1) [Info](#)

Find triggers



Trigger




EventBridge (CloudWatch Events): [salesAnalysisReportDailyTrigger](#)

arn:aws:events:us-west-2:924376459018:rule/salesAnalysisReportDailyTrigger


Rule state: **ENABLED**

► Details

Paso 4: Revisar correo electrónico → Notificación SNS




SARTopic <no-reply@sns.amazonaws.com>
para tomas.villaseca.c ▾

 inglés ▾

>

español ▾

[Traducir mensaje](#)



Sales Analysis Report
Date: 2023-11-06

Product Group: Pastries

Item Name *****	Quantity *****
Croissant	10
Donut	5
Chocolate Chip Cookie	5

Product Group: Drinks

Item Name *****	Quantity *****
Hot Chocolate	5

Laboratorio Completado

