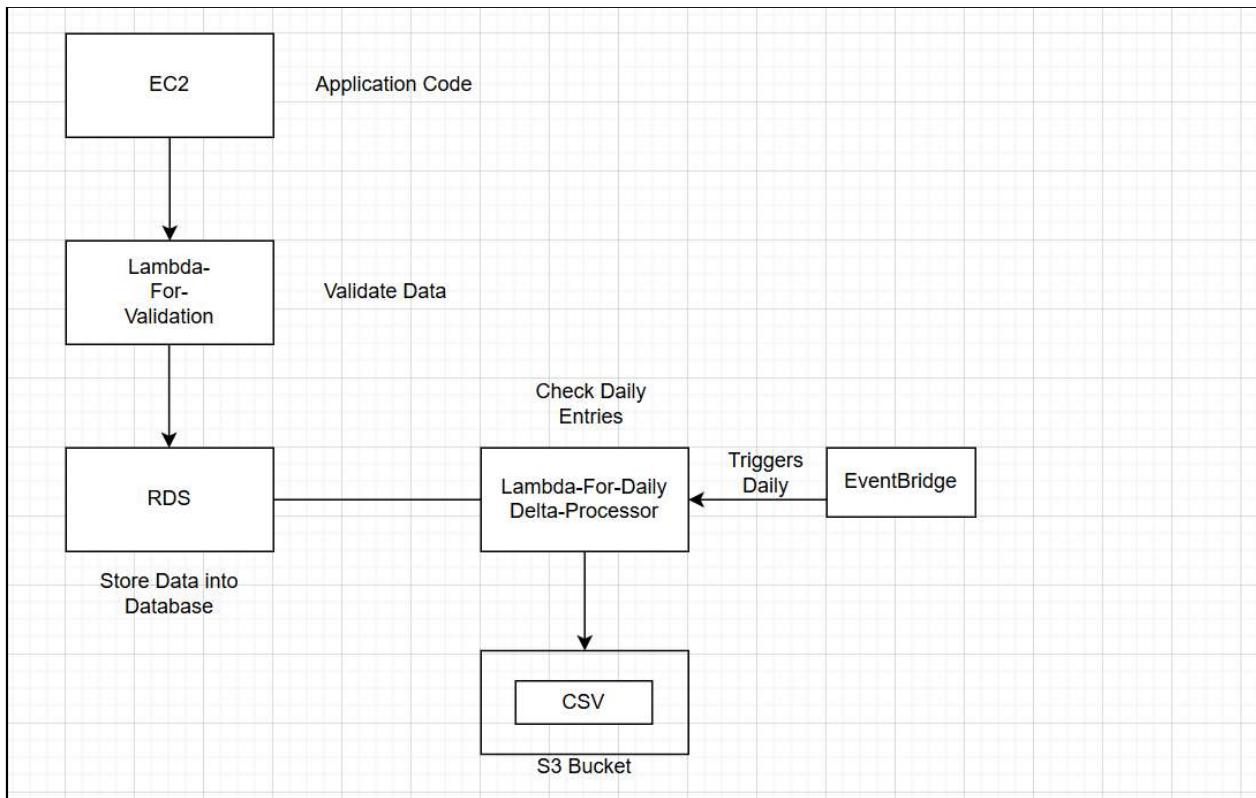


Task 21

Automated Order Processing System



Step 1 : Create VPC

- 2 Public subnet → EC2
- 2 Private subnet → RDS

vpc-0f75964754a5bd1d1

vpc-0f75964754a5bd1d1 / Application-VPC

Actions ▾

Details		Info	
VPC ID	vpc-0f75964754a5bd1d1	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-0c6cbc1a0749873e5	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
Encryption control ID	-	Encryption control mode	-
		Block Public Access	Off
		DHCP option set	dopt-05e9ba97e526c0e9d
		IPv4 CIDR	10.0.0.0/18
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-0af47a3c5856f7513
		IPv6 pool	-
		Owner ID	215317654435



Step 2 : Create Subnet Group

- Go to RDS click on Subnet Group
- Click create subnet .
- Name : order-processing-app-SG
- VPC : same as created above.
- Availability Zone : select in which Private instance are present
- Subnets : Private-Subnet
- Create.

The screenshot shows the AWS RDS Subnet groups page. The left sidebar has a 'Subnet groups' section highlighted with a red box. The main area displays a table titled 'Subnet groups (0)' with columns for Name, Description, Status, and VPC. A message at the bottom states 'No db subnet groups' and 'You don't have any db subnet groups.' A 'Create DB subnet group' button is located at the bottom right of the table area.

Subnets selected (2)			
Availability zone	Subnet name	Subnet ID	CIDR block
ap-south-1a	Pvt-sub-1	subnet-045ac77b53b432406	10.0.32.0/20
ap-south-1b	Pvt-sub-2	subnet-01c66f20c675d4e5a	10.0.48.0/20

Cancel Create

The screenshot shows the 'Create DB subnet group' wizard. The first step, 'Subnet group details', is displayed. It includes fields for 'Name' (order-processing-app-SBG), 'Description' (order processing application DB Group), and 'VPC' (Application-VPC (vpc-0f75964754a5bd1d1)). The 'VPC' dropdown shows '4 Subnets, 2 Availability Zones'.

Add subnets

Availability Zones

Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone

ap-south-1a X ap-south-1b X

Subnets

Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

Pvt-sub-1
Subnet ID: subnet-045ac77b53b432406 CIDR: 10.0.32.0/20 X

Pvt-sub-2
Subnet ID: subnet-01c66f20c675d4e5a CIDR: 10.0.48.0/20 X

 For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Step 3 : Create Security Group

Step 3.1 : Security group for RDS

- Inbound Rule :
 - MySql : 3306 (Allow)

Create security group [Info](#)
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)
RDS-SG
Name cannot be edited after creation.

Description [Info](#)
Allow Mysql

VPC Info
vpc-0f75964754a5bd1d1 (Application-VPC)

Inbound rules [Info](#)

Type	Protocol	Port range	Source	Description - optional
MySQL/Aurora	TCP	3306	Anywhere	0.0.0.0/0

[Add rule](#)

Step 3.2 : Security group for EC2

- Inbound Rule :
 - MySql : 3306 (Allow)
 - SSH : 22
 - HTTP : 80
 - Custom TCP : 50

sg-0b08347591cec71b0 - application-server-SG

sg-0b08347591cec71b0 - application-server-SG [Actions](#)

Details

Security group name application-server-SG	Security group ID sg-0b08347591cec71b0	Description Allows SSH, HTTP and MySql	VPC ID vpc-0f75964754a5bd1d1
Owner 215317654435	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) [Outbound rules](#) [Sharing](#) [VPC associations](#) [Related resources - new](#) [Tags](#)

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-08f245ca6a3f6d0a0	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-0a3f94b73289857d2	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0
-	sgr-052856c5de11785c0	IPv4	Custom TCP	TCP	5000	0.0.0.0/0
-	sgr-06a6c66fd0a61ba8	IPv4	HTTP	TCP	80	0.0.0.0/0

Step 3.2 : Security group for Lambda

- Inbound Rule :
 - MySql : 3306 (Allow)

The screenshot shows the AWS Security Groups console for a security group named "sg-0df9093de06ae12ef - lambda-SG".

Details:

- Security group name: sg-0df9093de06ae12ef - lambda-SG
- Owner: 215317654435
- Security group ID: sg-0df9093de06ae12ef
- Description: demo
- VPC ID: vpc-0f75964754a5bd1d1
- Inbound rules count: 1 Permission entry
- Outbound rules count: 1 Permission entry

Inbound rules (1):

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-06bd28b15663ba5c2	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0

Step 4 : Create Database

- AWS Console → RDS → Create DB
- Engine: MySQL
- Template: Free tier
- DB name: ordersdb
- Username: admin
- Password: Strong Password
- Public access: NO
- VPC: same as EC2
- Subnet group: Created above

The screenshot shows the 'Create database' wizard in the AWS RDS console. The first section, 'Choose a database creation method', offers two options: 'Full configuration' (selected) and 'Easy create'. The second section, 'Engine options', displays various database engines with their icons: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), PostgreSQL, MariaDB, Oracle (with ORACLE logo), Microsoft SQL Server (with Microsoft SQL Server logo), and IBM Db2 (with IBM Db2 logo). The third section, 'Edition', shows 'MySQL Community' selected. The fourth section, 'Engine version', lists 'MySQL 8.4.7' and includes an optional checkbox for 'Enable RDS Extended Support'. The bottom of the page contains standard AWS navigation and search tools.

Templates

Choose a sample template to meet your use case.

Production
Use defaults for high availability and fast, consistent performance.

Dev/Test
This instance is intended for development use outside of a production environment.

Sandbox
To develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

Availability and durability

Deployment options [Info](#)

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

Multi-AZ DB cluster deployment (3 instances)
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:

- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

Multi-AZ DB instance deployment (2 instances)
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:

- 99.95% uptime
- Redundancy across Availability Zones

Single-AZ DB instance deployment (1 instance)
Creates a single DB instance without standby instances. This setup provides:

- 99.95% uptime
- No data redundancy

The diagram illustrates three deployment options for Amazon RDS instances:

- Multi-AZ DB cluster deployment (3 instances):** Shows a Primary instance in AZ 1 with a Write/read endpoint. Two Readable standby + SSD instances are in AZ 2 and AZ 3, each with its own Write/read endpoint.
- Multi-AZ DB instance deployment (2 instances):** Shows a Primary instance in AZ 1 with a Write/read endpoint and a Standby instance in AZ 2 with a Standby endpoint.
- Single-AZ DB instance deployment (1 instance):** Shows a Primary instance in AZ 1 with a Write/read endpoint.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed
Create your own password or have RDS create a password that you manage.

Master password [Info](#)

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ ^ @

Confirm master password [Info](#)

▼ Additional credentials settings

Database authentication options [Info](#)

Password authentication
Authenticates using database passwords.

Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

Hide filters

Show instance classes that support Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Include previous generation classes

Standard classes (includes m classes)
 Memory optimized classes (includes r and x classes)
 Burstable classes (includes t classes)

Instance type

2 vCPUs 1 GiB RAM EBS Bandwidth: Up to 2,085 Mbps Network: Up to 5 Gbps

Storage

Storage type [Info](#)
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage [Info](#)
20 GiB
Allocated storage value must be 20 GiB to 6,144 GiB

Additional storage configuration

Connectivity

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) [Info](#)
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Application-VPC (vpc-0f75964754a5bd1d1)
4 Subnets, 2 Availability Zones

After a database is created, you can't change its VPC. Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group [Info](#)
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

order-processing-app-sbg
2 Subnets, 2 Availability Zones

Public access [Info](#)

Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) [Info](#)
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups
Choose one or more options

RDS-SG X

Availability Zone [Info](#)
ap-south-1a

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy [Info](#)
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

Certificate authority - optional [Info](#)
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiration: May 20, 2061

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

Tags - optional
A tag consists of a case-sensitive key-value pair.
No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Monitoring [Info](#)
Choose monitoring tools for this database. Database Insights provides a combined view of Performance Insights and Enhanced Monitoring for your fleet of databases. Database Insights pricing is separate from RDS monthly estimates. See [Amazon CloudWatch pricing](#).

Database Insights - Advanced
• Retains 15 months of performance history
• Fleet-level monitoring
• Integration with CloudWatch Application Signals

Database Insights - Standard

▼ Additional monitoring settings

Enhanced Monitoring, CloudWatch Logs and DevOps Guru

Enhanced Monitoring

Enable Enhanced monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

OS metrics granularity

60 seconds

Monitoring role for OS metrics

default ▾   

The monitoring role is an IAM role that allows RDS to send Enhanced Monitoring metrics to Amazon CloudWatch Logs. Choose an existing monitoring role, or choose default to have RDS automatically create the IAM role rds-monitoring-role for you.

Log exports

Select the log types to publish to Amazon CloudWatch Logs.

- Audit log
- Error log
- General log
- iam-db-auth-error log
- Slow query log

IAM role

The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

Creating database ordersdb
Your database might take a few minutes to launch. You can use settings from ordersdb to simplify configuration of suggested database add-ons while we finish creating your DB for you.

 X

Databases (1)

DB identifier	Status	Role	Engine	Upgrade rollout order	Region ...	Size	Recommendations	CPU
ordersdb	Creating	Instance	MySQL Co...	SECOND	ap-south-1a	db.t4g.micro	-	-

Step 5 : Create EC2

Step 5.1 : Create IAM Role

- Name : EC2-Secrets-Manager-Access-Role
- Permissions :
 - SecretManagerReadWrite
 - AWSLambda_FullAccess
 - AmazonS3FullAccess
- Attach to EC2.

The screenshot shows the 'EC2-Secrets-Manager-Access-Role' configuration page. The 'Summary' section displays basic information: Creation date (February 11, 2026, 10:55 (UTC+05:30)), Last activity (35 minutes ago), ARN (arn:aws:iam::215317654435:role/EC2-Secrets-Manager-Access-Role), and Maximum session duration (1 hour). The 'Permissions' tab is selected, showing two attached policies: 'AWSLambda_FullAccess' and 'SecretsManagerReadWrite'. The 'Add permissions' button is visible at the top right of the policy list.

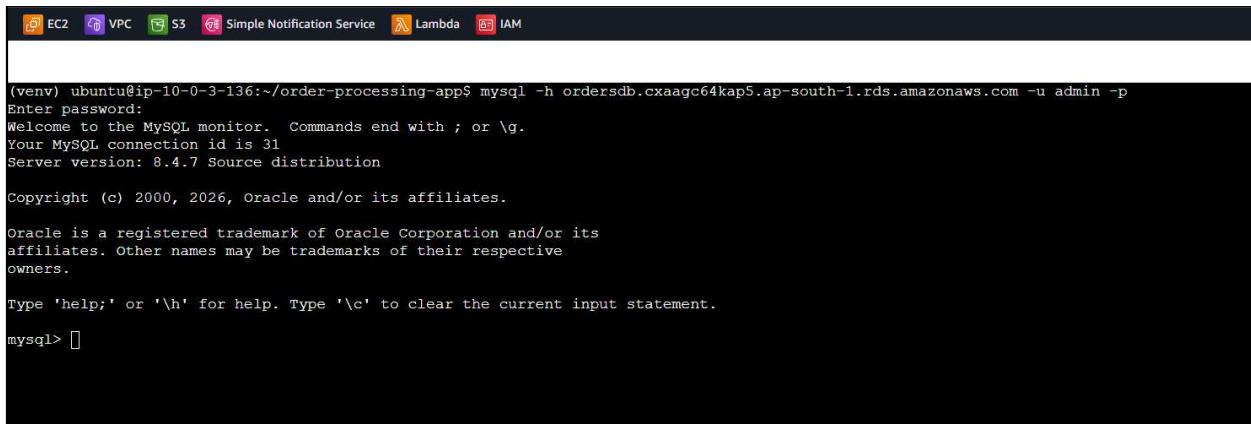
Step 5.2 : Create EC2

- Name : order-app-server
- Choose OS (AMI) : Ubuntu Server 22.04 LTS (easy for beginners)
- Instance type: t3.micro
- Security group: Security group created above for EC2
- Launch

Instance summary for i-0f791ebc13df7b83e (order-processing-server) Info		
Updated 5 minutes ago		
Instance ID i-0f791ebc13df7b83e	Public IPv4 address 3.110.102.197 open address	Private IPv4 addresses 10.0.3.136
IPv6 address -	Instance state Running	Public DNS -
Hostname type IP name: ip-10-0-3-136.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-3-136.ap-south-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t3.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 3.110.102.197 [Public IP]	VPC ID vpc-0f75964754a5bd1d1 (Application-VPC)	Auto Scaling Group name -
IAM role EC2-Secrets-Manager-Access-Role	Subnet ID subnet-0560b9c0a28a47f6a (Pub-sub-1)	Managed false
IMDSv2 Required	Instance ARN arn:aws:ec2:ap-south-1:215317654435:instance/i-0f791ebc13df7b83e	

Step 5.3 : Connect Database

- Connect RDS database through EC2.
 - Mysql -h <RDS-endpoint> -u <username> -p <password>



The screenshot shows a terminal window with the AWS CloudWatch interface. The tabs at the top include EC2, VPC, S3, Simple Notification Service, Lambda, and IAM. The main area displays a MySQL session:

```
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ mysql -h ordersdb.cxaagc64kap5.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 31
Server version: 8.4.7 Source distribution

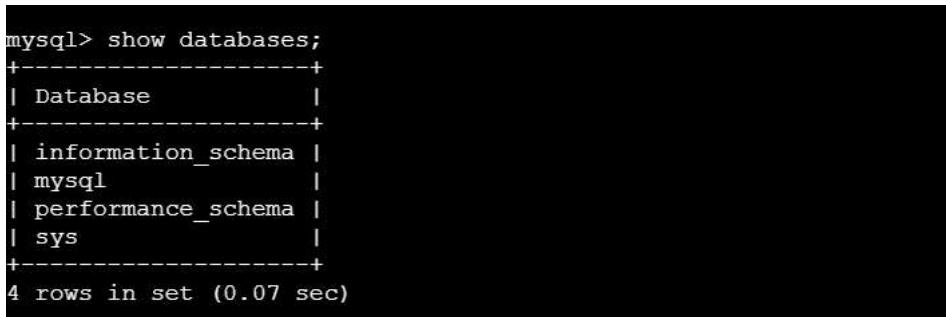
Copyright (c) 2000, 2026, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

- Verify databases
- Show databases;



```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.07 sec)
```

- Create new database for application.
- CREATE DATABASE <database_name>;

```
mysql> CREATE DATABASE orders;
Query OK, 1 row affected (0.09 sec)
```

```
mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| orders          |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.06 sec)

mysql> █
```

- Create table
 - Use <database_name>
- Write table creation query;

```
mysql> USE orders;
Database changed
mysql>
mysql> CREATE TABLE orders (
    ->   order_id INT AUTO_INCREMENT PRIMARY KEY,
    ->   customer_name VARCHAR(100),
    ->   amount DECIMAL(10,2),
    ->   status VARCHAR(20),
    ->   created_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_orders |
+-----+
| orders           |
+-----+
1 row in set (0.03 sec)
```

```
mysql> desc orders;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra       |
+-----+-----+-----+-----+-----+
| order_id | int    | NO   | PRI | NULL    | auto_increment |
| customer_name | varchar(100) | YES  |     | NULL    |             |
| amount | decimal(10,2) | YES  |     | NULL    |             |
| status | varchar(20) | YES  |     | NULL    |             |
| created_time | timestamp | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+
5 rows in set (0.07 sec)
```

Step 5.4 : Install dependencies on Server.

- Install python and check version
 - sudo apt install python3 -y
 - Python --version

```
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ python --version
Python 3.12.3
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ █
```

- Create virtual environment.
 - Python3 -m venv venv
- Activate virtual environment
 - Source venv/bin/activate

```
ubuntu@ip-10-0-3-136:~/order-processing-app$ sudo chown -R ubuntu:ubuntu ~/order-processing-app
ubuntu@ip-10-0-3-136:~/order-processing-app$ python3 -m venv venv
ubuntu@ip-10-0-3-136:~/order-processing-app$ source venv/bin/activate
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ ls
venv
```

- Install flask
 - Pip install flask
 - pip3 install flask boto3
 - Pip install flask pymysql boto3

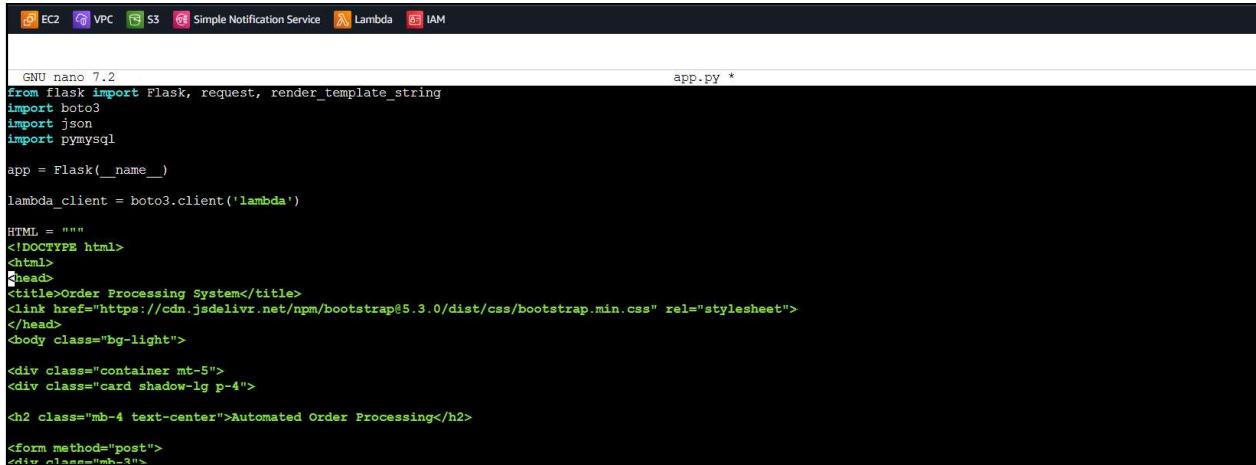
```
ubuntu@ip-10-0-3-136:~/order-processing-app$ source venv/bin/activate
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ ls
venv
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ pip install flask
Collecting flask
  Using cached flask-3.1.2-py3-none-any.whl.metadata (3.2 kB)
Collecting blinker>=1.9.0 (from flask)
  Using cached blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from flask)
  Using cached click-8.3.1-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2.0 (from flask)
  Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from flask)
  Using cached jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from flask)
  Using cached markupsafe-3.0.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (2.7 kB)
Collecting werkzeug>=3.1.0 (from flask)
  Using cached werkzeug-3.1.5-py3-none-any.whl.metadata (4.0 kB)
Using cached flask-3.1.2-py3-none-any.whl (103 kB)
```

```
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ pip install flask pymysql boto3
Requirement already satisfied: flask in ./venv/lib/python3.12/site-packages (3.1.2)
Collecting pymysql
  Using cached pymysql-1.1.2-py3-none-any.whl.metadata (4.3 kB)
Collecting boto3
  Downloading boto3-1.42.47-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: blinker>=1.9.0 in ./venv/lib/python3.12/site-packages (from flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in ./venv/lib/python3.12/site-packages (from flask) (8.3.1)
Requirement already satisfied: itsdangerous>=2.2.0 in ./venv/lib/python3.12/site-packages (from flask) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in ./venv/lib/python3.12/site-packages (from flask) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in ./venv/lib/python3.12/site-packages (from flask) (3.0.3)
Requirement already satisfied: werkzeug>=3.1.0 in ./venv/lib/python3.12/site-packages (from flask) (3.1.5)
Collecting botocore<1.43.0,>=1.42.47 (from boto3)
  Downloading botocore-1.42.47-py3-none-any.whl.metadata (5.9 kB)
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
  Using cached jmespath-1.1.0-py3-none-any.whl.metadata (7.6 kB)
Collecting s3transfer<0.17.0,>=0.16.0 (from boto3)
  Using cached s3transfer-0.16.0-py3-none-any.whl.metadata (1.7 kB)
```

Step 5.5 : Create Application and run

- Create application file
- Sudo nano app.py
- Write the code

```
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ nano app.py
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$
```



The screenshot shows the AWS Lambda function editor interface. At the top, there are navigation tabs: EC2, VPC, S3, Simple Notification Service, Lambda, and IAM. Below the tabs, the file name 'app.py' is displayed with a star icon indicating it's a new file. The code area contains the following Python code:

```
GNU nano 7.2
app.py *
from flask import Flask, request, render_template_string
import boto3
import json
import pymysql
app = Flask(__name__)
lambda_client = boto3.client('lambda')
HTML = """
<!DOCTYPE html>
<html>
<head>
<title>Order Processing System</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="bg-light">
<div class="container mt-5">
<div class="card shadow-lg p-4">
<h2 class="mb-4 text-center">Automated Order Processing</h2>
<form method="post">
<div class="mb-3">
```

Step 6 : Configure Secrets in secret manager.

- AWS Console → Secrets Manager
- Create secret:
 - Type: Credentials for RDS
 - username: admin
 - password: <Password>
 - host: RDS endpoint
 - db: ordersdb
- Secret name:
 - order-db-secret

Choose secret type

Secret type Info

Credentials for Amazon RDS database Credentials for Amazon DocumentDB database Credentials for Amazon Redshift data warehouse
 Credentials for other database Managed external secret Secrets vended by your third-party software vendor. Other type of secret API key, OAuth token, other.

Credentials

Username

Password Show password

Encryption key

Encryption key Info
You can encrypt using the KMS key that Secrets Manager creates or a customer-managed KMS key that you create.

Database

Database Info

DB instance	DB engine	Status	Creation date (UTC)
ordersdb	mysql	available	12 February 2026 at 05:43:52

Configure secret

Secret name and description Info

Secret name

A descriptive name that helps you find your secret later.

Secret name must only contain alphanumeric characters and the characters /_+=.:@~

Description - optional

Maximum 250 characters.

AWS		Search	[Alt+S]	Sejal Pawar (2153-1765-4435) ▾		Sejal Pawar
EC2	VPC	S3	Simple Notification Service	Lambda	IAM	
AWS Secrets Manager		Secrets				
Secret name	Description	Last retrieved (UTC)	Created on (UTC)			
order-db-secret	Mysql Access	-	12 February 2026 at 06:00:04			

Step 7 : Configure Lambda

Step 7.1 : Create IAM Role for Lambda

- Name : order-app-lambda-Role
- Permissions :
 - SecretManagerReadWrite
 - AWSLambdaBasicExecutionRole
 - AWSLambdaVPCAccessExecutionRole
- Attach to EC2.

The screenshot shows the 'order-app-lambda-role' configuration page in the AWS IAM console. The role allows Lambda functions to call AWS services on behalf of the user. It was created on February 11, 2026, at 12:12 (UTC+05:30) and last updated 18 hours ago. The ARN is arn:aws:iam::215317654435:role/order-app-lambda-role, and the maximum session duration is 1 hour. The 'Permissions' tab is selected, showing three attached managed policies: AWSLambdaBasicExecutionRole, AWSLambdaVPCAccessExecutionRole, and SecretsManagerReadWrite. There are tabs for 'Trust relationships', 'Tags', 'Last Accessed', and 'Revoke sessions'. Buttons for 'Edit', 'Delete', 'Simulate', 'Remove', and 'Add permissions' are visible.

Step 7.2 : Create lambda Function.

- Create Function
- Author from scratch
- Function name : order-validator
- Runtime : Python 3.14
- Attach role created above.

aws Search [Alt+S] Asia Pacific (Mumbai) Sejal Pawar (21)

EC2 VPC S3 Simple Notification Service Lambda IAM

Lambda Functions Create function

Create function Info

Choose one of the following options to create your function.

Author from scratch Start with a simple Hello World example.

Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.

Container image Select a container image to deploy for your function.

Basic information

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

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

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

Durable execution - new Info
Enable durable execution to simplify building resilient multi-step applications that checkpoint progress and resume after interruptions. Supports Python and Node.js runtimes. [View pricing](#).
 Enable

Architecture Info

Choose the instruction set architecture you want for your function code.

arm64

x86_64

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ 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.

[View the lambda-order-role](#) on the IAM console.

Step 7.3 : Create .zip file of dependencies on local system.

- Install python and check version.
 - Python –version

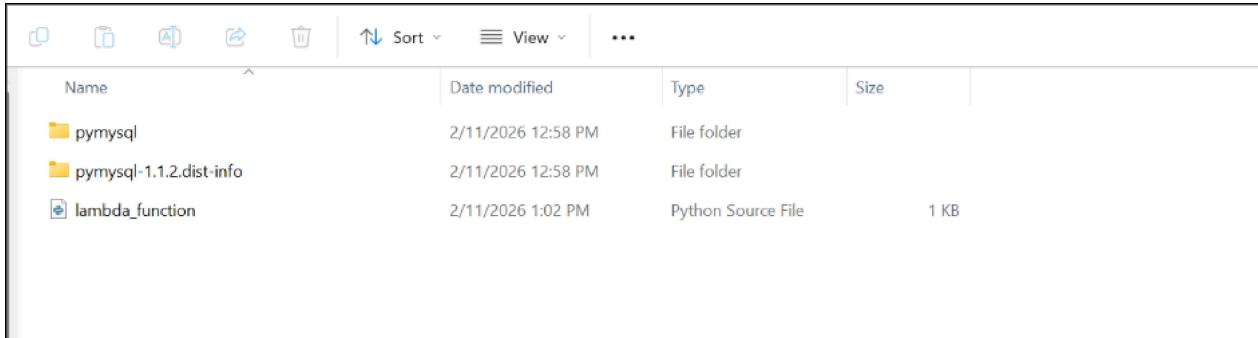
```
PS C:\Users\SejalPawar\Desktop\lambda-package> python --version  
Python 3.14.3
```

- Install pymysql
 - Python -m pip install pymysql -t

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Python 3.14.3
PS C:\Users\SejalPawar\Desktop\lambda-package> python -m pip install pymysql -t .
Collecting pymysql
  Downloading pymysql-1.1.2-py3-none-any.whl.metadata (4.3 kB)
Downloaded pymysql-1.1.2-py3-none-any.whl (45 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.2
```


- Zip the folder



Step 7.4 : Create layer

- Click on layers
- Create layer
- Name : pymysql-layer
- Upload .zip file.
- Architecture : same as Lambda function
- Runtime : same as lambda function

Lambda > Functions > order-validator

order-validator

Function overview Info

Description

Last modified 6 minutes ago

Function ARN arn:aws:lambda:ap-south-1:215317654435:function:order-validator

Function URL Info

Additional resources

- Capacity providers New
- Code signing configurations
- Event source mappings
- Layers** (0) (highlighted)
- Replicas

Related AWS resources

Step Functions state machines

EC2 VPC S3 Simple Notification Service Lambda IAM

Lambda > Layers > Create layer

Create layer

Layer configuration

Name pymysql-layer

Description - optional

Upload a .zip file
 Upload a file from Amazon S3

[Choose file](#)

lambda.zip
130.40 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.

[Search Compatible architectures](#)

arm64 [X](#)

Compatible runtimes - optional | [Info](#)
 Choose up to 15 runtimes.

[Search Compatible runtimes](#)

Python 3.14 [X](#)

[Create](#)

Step 7.5 : Attach layer to function

- Click on layers.
- Add layers
- Custom layers
- Select layer created above.
- Select Version.
- Create

order-validator

[Function overview](#) [Info](#)

[Diagram](#) [Template](#)

order-validator

[Layers](#) (0)

[+ Add trigger](#)

[+ Add destination](#)

[Export to Infrastructure Composer](#) [Download](#)

Description

Last modified 8 minutes ago

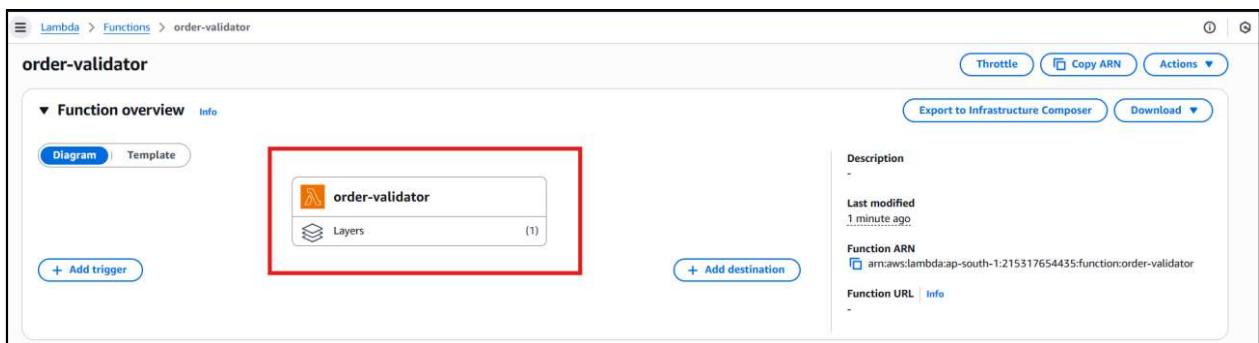
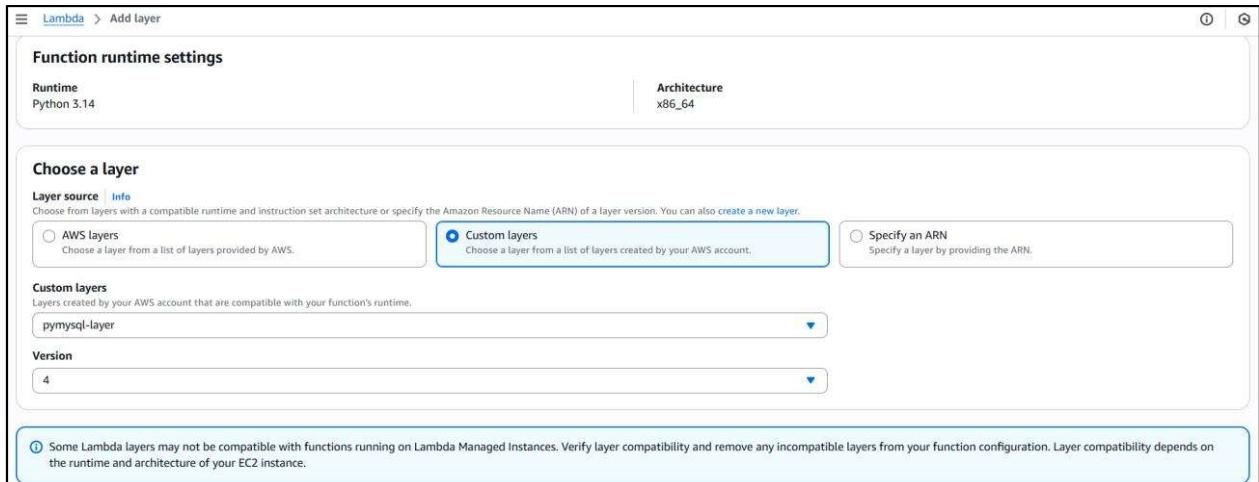
Function ARN [arn:aws:lambda:ap-south-1:215317654435:function:order-validator](#)

Function URL [Info](#)

[Layers](#) [Info](#)

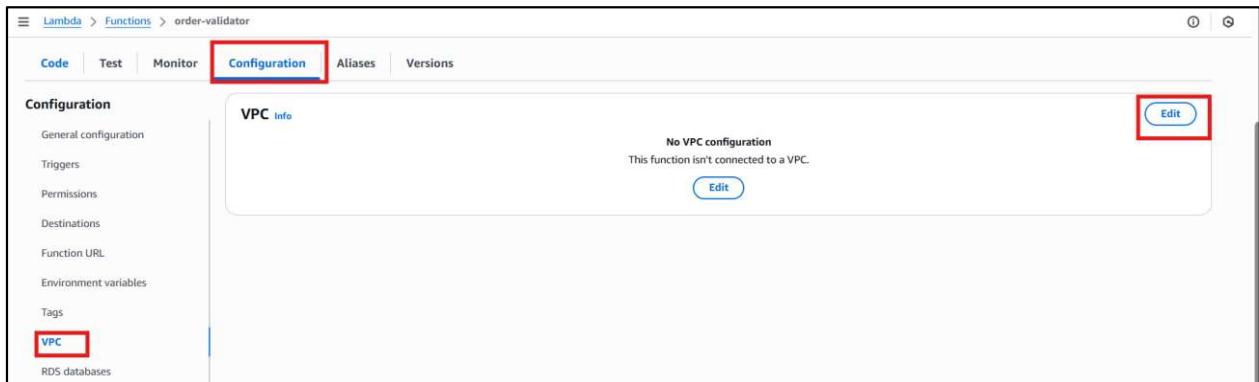
Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					

[Edit](#) [Add a layer](#)



Step 7.6 : Configure networking part for Lambda.

- Click on configuration
- Select VPC > Edit
- Select same VPC as RDS and EC2
- Select Private subnets
- Attach security group
 - Inbound allow : 3306 (MySQL)



VPC

When you connect a function to a VPC in your account, it does not have access to the internet unless your VPC provides access. To give your function access to the internet, route outbound traffic to a NAT gateway in a public subnet.

[Learn more ↗](#)

VPC | Info
Select a VPC for your resource to access.
vpc-0f7964754a5bd1d1 (10.0.0.0/18) [Choose](#)

Allow IPv6 traffic for dual-stack subnets
You can allow outbound IPv6 traffic to subnets that have both IPv4 and IPv6 CIDR blocks.

Subnets
Select the VPC subnets for Lambda to use to set up your VPC configuration.
Choose subnets [Choose](#)
subnet-045ac77b53b432406 (10.0.32.0/20) ap-south-1a [Choose](#)
Name: Pvt-sub-1
subnet-01c66f20c675d4e5a (10.0.48.0/20) ap-south-1b [Choose](#) [Choose](#)
Name: Pvt-sub-2

Security groups
Select security groups for your VPC configuration. The following table shows the inbound and outbound rules for your selected security groups.
Choose security groups [Choose](#)
sg-0ee2d4c15611a7334 (order-db-SG) [Choose](#)
Allow Mysql 3306

- Deploy and test the function.

[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

Executing function: succeeded (logs ↗)

[Details](#)

```
{
  "statusCode": 200,
  "body": "{\"Hello from Lambda!\"}"
}
```

Summary

Code SHA-256	Execution time
R6fl2f5y9xHPTo4NojRtY2hz04DIAxWgwXm4xSMnwRQ=	10 seconds ago
Function version	Request ID
\$LATEST	da3e54ed-8717-4799-aa91-204cc761af9b
Duration	Billed duration
2.06 ms	128 ms
Resources configured	Max memory used
128 MB	40 MB
Init duration	
125.69 ms	

i-0f791ebc13df7b83e

Instance summary for i-0f791ebc13df7b83e (order-processing-server) [Info](#)

Updated 24 minutes ago

Instance ID	Public IPv4 address
i-0f791ebc13df7b83e	3.110.102.197 open address ↗
IPv6 address	Instance state
-	Running
Hostname type	Private IP DNS name (IPv4 only)
IP name: ip-10-0-3-136.ap-south-1.compute.internal	ip-10-0-3-136.ap-south-1.compute.internal
Answer private resource DNS name	Instance type
-	t3.micro
Auto-assigned IP address	VPC ID
3.110.102.197 [Public IP]	vpc-0f7964754a5bd1d1 (Application-VPC) Choose
IAM role	Subnet ID
EC2-Secrets-Manager-Access-Role ↗	subnet-0560b9c0a28a47f6a (Pub-sub-1) Choose
IMDSv2	Instance ARN
Required	arn:aws:ec2:ap-south-1:215317654435:instance/i-0f791ebc13df7b83e

[Actions ▲](#)

- [Connect](#)
- [Instance state ▾](#)
- [Actions ▲](#)
- [Instance diagnostics](#)
- [Instance settings](#)
- [Networking](#)
- [Security](#) **(Red Box)**
- [Image and templates](#)
- [Storage](#)
- [Monitor and troubleshoot](#)

AWS Compute Optimizer finding
[Opt-in to AWS Compute Optimizer for recommendations. | Learn more ↗](#)

Elastic IP addresses
-

Auto Scaling Group name
-

Managed
false

EC2 > Instances > i-0f791ebc15df7b83e > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

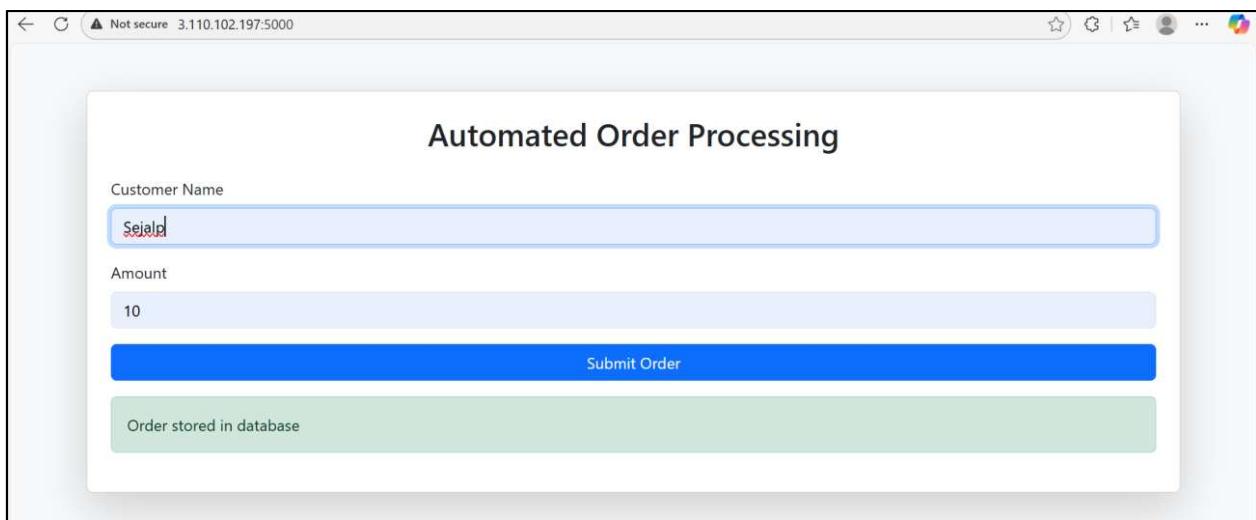
Instance ID
 i-0f791ebc15df7b83e (order-processing-server)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

Step 8 : Step Run application

- Run the application on browser.
 - <public_ip:5000>

```
ubuntu@ip-10-0-3-136:~/order-processing-app$ source venv/bin/activate
(venv) ubuntu@ip-10-0-3-136:~/order-processing-app$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.3.136:5000
Press CTRL+C to quit
110.227.211.247 - - [12/Feb/2026 11:02:37] "GET / HTTP/1.1" 200 -
Lambda Response: {'statusCode': 200, 'body': '{"status": "FAILED", "message": "Database error: \'dbname\'"}'}
110.227.211.247 - - [12/Feb/2026 11:02:43] "POST / HTTP/1.1" 200 -
Lambda Response: {'statusCode': 200, 'body': '{"status": "SUCCESS", "message": "Order stored in database"}'}
111.125.244.2 - - [12/Feb/2026 11:07:15] "POST / HTTP/1.1" 200 -
```



- Verify in Mysql server
- Order is **PROCESSED** and stored in RDS

```
mysql> select * from orders;
+-----+-----+-----+-----+
| order_id | customer_name | amount | status | created_time |
+-----+-----+-----+-----+
| 1 | SP | 7777.00 | PROCESSED | 2026-02-12 11:07:15 |
| 2 | Sejalp | 56666.00 | PROCESSED | 2026-02-12 11:07:25 |
| 3 | Mruda | 987.00 | PROCESSED | 2026-02-12 11:07:33 |
| 4 | Sejalp | 10.00 | PROCESSED | 2026-02-12 11:12:47 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- When Value is less than or equal to Zero It shows **invalid amount**

Automated Order Processing

Customer Name

Amount

Invalid amount

Step 9 : Create bucket

- Create bucket
- Select general purpose
- Enable public access

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Asia Pacific (Mumbai) ap-south-1

Bucket type [Info](#)

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)
order-daily-reports-215317654435

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn more](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Object Ownership

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings is independent of one another.

Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permission applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

[Amazon S3](#) > [Buckets](#) > order-daily-reports-215317654435 [Info](#)

[Objects](#) [Metadata](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

[Upload](#)

Step 10 : Configure lambda function for Report generation

Step 10.1 : Create Role

- Name : lambda-daily-delta-task-role
- Permissions:
 - AmazonS3FullAccess
 - SecretsManagerReadWrite

The screenshot shows the 'lambda-daily-delta-task-role' role details. It includes a summary section with creation date (February 12, 2026), ARN (arn:aws:iam::215317654435:role/lambda-daily-delta-task-role), and maximum session duration (1 hour). Below this is a 'Permissions' tab showing two attached policies: 'AmazonS3FullAccess' and 'SecretsManagerReadWrite'. The 'Add permissions' button is visible.

Step 10.2 : Create lambda function .

- Create Function
- Author from scratch
- Function name : daily-delta-processor
- Runtime : Python 3.12
- Attach role created above.

The screenshot shows the 'Create function' wizard. It starts with options to 'Author from scratch', 'Use a blueprint', or 'Container image'. The 'Author from scratch' option is selected. In the 'Basic information' step, the function name is set to 'daily-delta-processor', runtime is chosen as 'Python 3.12', and the code editor shows a simple 'Hello World' example.

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
 arm64
 x86_64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ 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.
 lambda-daily-delta-task-role [View the lambda-daily-delta-task-role role](#) [on the IAM console](#).

Step 10.6 : Attach layer to function

- Click on layers.
- Add layers
- Custom layers
- Select layer created above.
- Select Version.
- Create

order-validator

Function overview [Info](#)

[Diagram](#) [Template](#)

order-validator

Layers (0) [+ Add destination](#)

[+ Add trigger](#)

[Throttle](#) [Copy ARN](#) [Actions](#)

[Export to Infrastructure Composer](#) [Download](#)

Description
-

Last modified
8 minutes ago

Function ARN
 arn:aws:lambda:ap-south-1:215317654435:function:order-validator

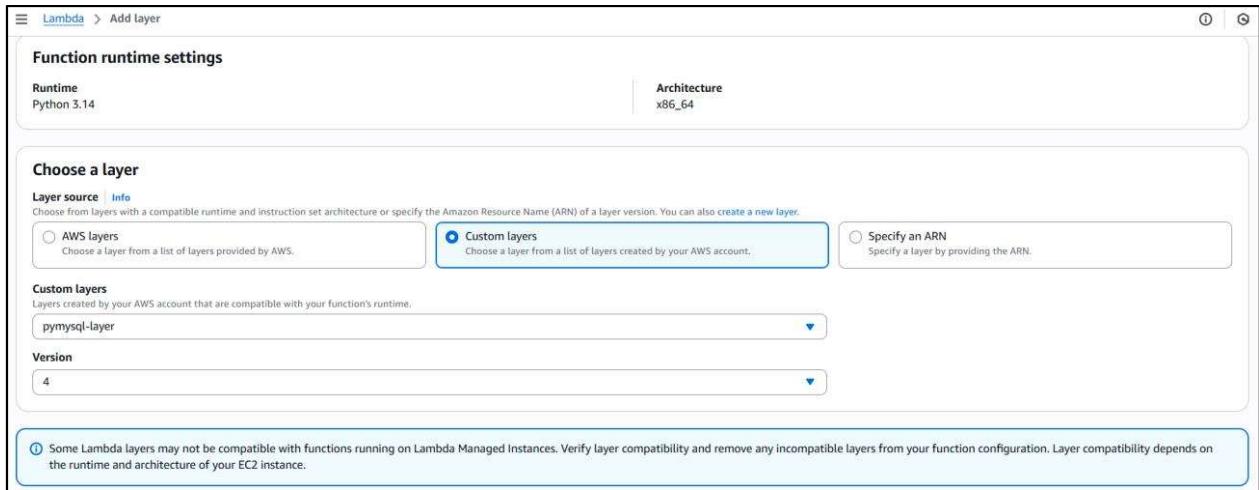
Function URL [Info](#)
-

Layers [Info](#)

Merge order | Name | Layer version | Compatible runtimes | Compatible architectures | Version ARN

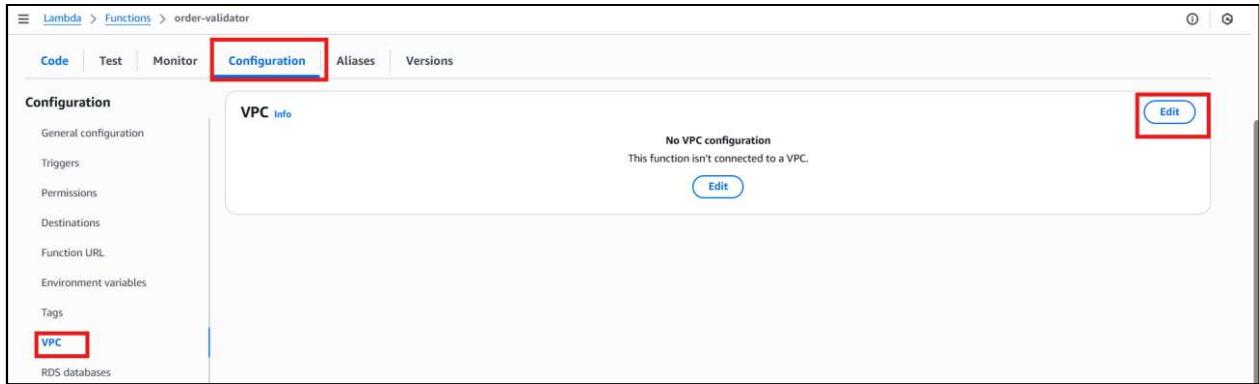
There is no data to display.

[Edit](#) [Add a layer](#)



Step 10.4 : Configure networking part for Lambda.

- Click on configuration
- Select VPC > Edit
- Select same VPC as RDS and EC2
- Select Private subnets
- Attach security group
 - Inbound allow : 3306 (MySQL)



VPC

When you connect a function to a VPC in your account, it does not have access to the internet unless your VPC provides access. To give your function access to the internet, route outbound traffic to a NAT gateway in a public subnet.

VPC | Info
Select a VPC for your resource to access.
vpc-0f75964754a5bd1d1 (10.0.0.0/18)

Allow IPv6 traffic for dual-stack subnets
You can allow outbound IPv6 traffic to subnets that have both IPv4 and IPv6 CIDR blocks.

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

subnet-01c66f20c675d4e5a (10.0.48.0/20) ap-south-1b subnet-045ac77b53b432406 (10.0.32.0/20) ap-south-1a
Name: Pvt-sub-2 Name: Pvt-sub-1

Security groups
Select security groups for your VPC configuration. The following table shows the inbound and outbound rules for your selected security groups.

sg-0ee2d4c15611a7334 (order-db-SG)
Allow Mysql 3306

[View security group rules](#)

Code | Test | Monitor | Configuration | Aliases | Versions

Configuration

- General configuration
- Triggers
- Permissions
- Destinations
- Function URL
- Environment variables
- Tags
- VPC**
- RDS databases
- Monitoring and operations tools
- Concurrency and recursion detection

VPC Info
VPC vpc-0f75964754a5bd1d1 (10.0.0.0/18) | Application-VPC

Subnets			
<ul style="list-style-type: none"> Allow IPv6 traffic = false subnet-045ac77b53b432406 (10.0.32.0/20) ap-south-1a, Pvt-sub-1 subnet-01c66f20c675d4e5a (10.0.48.0/20) ap-south-1b, Pvt-sub-2 			
Security group rules (2)			
Last fetched 2 minutes ago <input type="button" value="Edit"/>			
Inbound rules	Protocol	Ports	Source
sg-0ee2d4c15611a7334	Custom TCP	80	0.0.0.0/0
sg-0ee2d4c15611a7334	Custom TCP	3306	0.0.0.0/0

- Test lambda function.

Code | Test | Monitor | Configuration | Aliases | Versions

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

Details

```
{
  "status": "SUCCESS",
  "file": "daily_summary_2026-02-12.csv",
  "today": "2026-02-12",
  "today_amount": 71118,
  "order_increment": 5,
  "amount_increment": 71118
}
```

Summary

Code SHA-256	Execution time
dkB6lcSpVhTrOoqoGDzm0zaB32IVLS/IpJzHxEJykNB=	13 minutes ago
Function version	Request ID
\$LATEST	5446c991-3e21-4f23-ba04-fb89a8f3faf3
Duration	Billed duration
3530.92 ms	3864 ms
Resources configured	Max memory used
128 MB	91 MB
Init duration	
332.77 ms	

Step 11 : Verify file is upload in S3

- Open S3 Bucket Created above.
- Click on bucket name.
- Files are generated.

Name	Type	Last modified	Size	Storage class
daily_report_2026-02-12.csv	csv	February 12, 2026, 17:07:56 (UTC+05:30)	281.0 B	Standard
daily_summary_2026-02-12.csv	csv	February 12, 2026, 17:12:09 (UTC+05:30)	140.0 B	Standard

Step 12 : Create Event bridge Rule.

- AWS console > EventBridge
- Select EventBridge scheduled rule
- Click on create schedule rule.
- Name : daily-delta-trigger
- Define Schedule : Schedule pattern
 - Set Schedule for 24 hours
- Target : AWS services
- Select target : lambda Function
- Target this account
- Select Function : daily-delta-processor (Created above)

Amazon EventBridge
A serverless service for building event-driven applications

Get started

- EventBridge Rule with event pattern
- EventBridge Scheduled rule
- A rule that will invoke a target at a scheduled time.
- EventBridge Pipe
- A pipe connects an event source to a target with optional filtering and enrichment.
- EventBridge Schedule
- A schedule invokes a target one-time or at regular intervals defined by a cron or rate expression.
- EventBridge Schema registry
- Schema registries collect and organize schemas.

Create scheduled rule

Step 1

- Define rule detail
- Step 2
Define schedule
- Step 3
Select target(s)
- Step 4 - optional
Configure tags
- Step 5
Review and create

Define rule detail Info

Scheduled rule detail

Name

Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-_,_.

Description - optional

Event bus Info

Custom or partner event bus is not supported when Schedule is selected.

Enable the rule on the selected event bus

EventBridge Scheduler - A new AWS scheduling capability!
A new EventBridge scheduling functionality that provides one-time and recurring scheduling functionality independent of Event buses and rules. You can create a schedule to invoke targets such as a Lambda function. [Learn More](#)

[Continue in EventBridge Scheduler](#)

Step 1

- Define rule detail
- Define schedule
- Step 3
Select target(s)
- Step 4 - optional
Configure tags
- Step 5
Review and create

Define schedule Info

Schedule pattern

Schedule pattern

Choose the schedule type that best meets your needs.

A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

A schedule that runs at a regular rate, such as every 10 minutes.

Rate expression Info

Enter a value and the unit of time to run the schedule.

rate ()

Value
Unit, e.g. mins, hours...

[Cancel](#) [Previous](#) [Next](#)

Target 1

Target types
Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

EventBridge event bus
 EventBridge API destination
 AWS service

Select a target | [Info](#)
Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)

Lambda function ▾

Target location

Target in this account Target in another AWS account

Function

daily-delta-processor ▾ [C](#)

▶ Configure version/alias

Permissions

Use execution role (recommended)

Execution role
EventBridge needs permission to send events to the target specified above. By continuing, you are allowing us to do so. [EventBridge and AWS Identity and Access Management](#) ▾

Create a new role for this specific resource Use existing role

Rule daily-delta-trigger was created successfully X

daily-delta-trigger

[Edit](#) [Disable](#) [Delete](#) [CloudFormation Template](#) ▾

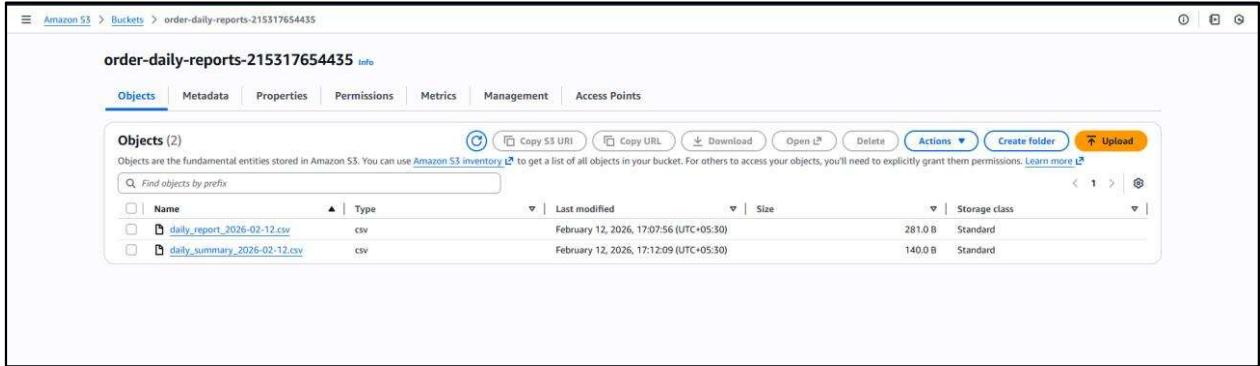
Rule details	Info
Rule name daily-delta-trigger	Status <input checked="" type="radio"/> Enabled
Description -	Rule ARN arn:aws:events:ap-south-1:215317654435:rule/daily-delta-trigger
	Event bus name default
	Event bus ARN arn:aws:events:ap-south-1:215317654435:event-bus/default
Type Scheduled Standard	

[Event schedule](#) | [Targets](#) | [Monitoring](#) | [Tags](#)

Event schedule [Info](#) [Edit](#)

Fixed rate of
24 hour

- After 24 hrs new report is generated.



- Report generated in S3.

Troubleshooting

During implementation of the Lambda + RDS integration, I faced several issues. Below are the problems and how I resolved them.

1. Lambda Error: No module named 'pymysql'

Issue

- My Lambda function failed with:
- Runtime.ImportModuleError: No module named 'pymysql'

Reason

- The required Python dependency was not included in the Lambda deployment package.

Resolution

- I installed pymysql locally and packaged it with the function:
 - pip install pymysql -t .
 - zip -r function.zip .

Then I uploaded the zip file to Lambda.

Alternatively, I created a Lambda Layer and attached it to the function.

2. Secret Key Error: KeyError 'dbname'

Issue

- Lambda Show:
 - KeyError: 'dbname'

Reason

- The database name key was missing in AWS Secrets Manager.

Resolution

- I updated the secret JSON to include:
 - "dbname": "orders"

3. Lambda Timeout

Issue

- Task timed out after 3.00 seconds

Reason

- Lambda could not connect to RDS in time.

Resolution

- I increased the timeout:
 - Lambda → Configuration → General configuration
 - Timeout: 15 seconds

4. Lambda Cannot Connect to RDS

Issue

- Function kept timing out.

Checks I performed

- Lambda and RDS are in the same VPC
- Security group allows port 3306
- RDS inbound rule allows Lambda security group
- Correct rule:
 - Type: MySQL
 - Port: 3306
 - Source: Lambda security group

After fixing security groups, the connection worked.

5. IAM Permission Error

Issue

- The execution role does not have permission to call CreateNetworkInterface

Reason

- Lambda role lacked VPC permissions.

Resolution

- I attached this policy to the Lambda role:
 - AWSLambdaVPCAccessExecutionRole