🔧 AWS Lambda + Aurora Serverless (MySQL) Integration Project

🚩 Problem Scenario

Modern cloud-native applications often require **seamless**, **scalable**, and **secure** database connectivity from AWS Lambda — without the overhead of managing persistent infrastructure. Traditional Lambda-to-RDS connectivity introduces challenges like:

- Complex **VPC networking** setup
- **Dependency packaging** (e.g., native MySQL libraries)
- Secure **secret management** for credentials

🎯 Goal:

To build a **serverless**, **cost-efficient**, and **secure** system that connects AWS Lambda to **Amazon Aurora Serverless v2** (MySQL-compatible), executes queries, and validates the entire stack using **Python** and **PyMySQL**.

☑️ Solution Overview

We solved the problem using:

- ✅ **Amazon Aurora Serverless v2** with MySQL compatibility
- ✅ **AWS Lambda** written in Python using the **PyMySQL** library
- ✅ **Amazon RDS Data API** for serverless query execution
- ✅ **IAM Role + AWS Secrets Manager** for secure, credential-free connectivity
- ✅ **Amazon RDS Query Editor v2** to create and validate DB schema

🛠️ Step-by-Step Implementation

1️⃣ 🌐 Create Aurora Serverless Cluster

- **Engine**: Aurora MySQL (v3.08.2 – MySQL 8.0.39 compatible)
- **Cluster Name**: `my-aurora-cluster`
- **Capacity Type**: Aurora Serverless v2 (Auto-scaling ACUs)
- **Connectivity**: VPC + Security Group (port 3306 open)

2 ☑ Enable RDS Data API

- Navigate to RDS dashboard → Select cluster → Enable **Data API**
- Allows executing SQL queries via **HTTPS** without a persistent connection

3 🔑 Configure IAM & Secrets

- Create a secret in **Secrets Manager** with DB credentials
- Create an **IAM Role** for Lambda with these policies:
  - `AmazonRDSDataFullAccess`
  - `SecretsManagerReadWrite`
- Attach the role to the Lambda function

4 Create Lambda Function with PyMySQL Layer

- Language: Python
- Packaged `PyMySQL` into a Lambda Layer:

```
pip install pymysql -t python
zip -r pymysql-layer.zip python
```

- Uploaded the `.zip` as a Lambda layer and attached it to the function

5 ☑ Verify Lambda Layer Works

Tested with:

```
import pymysql

def lambda_handler(event, context):
    return {
        'statusCode': 200,
        'body': "☑ PyMySQL layer is working correctly!"
    }
```

✅ Layer and basic Lambda functionality were confirmed

6 ❌ Initial Connection Fails

Initial attempts failed with:

```
Can't connect to MySQL server... [Errno 16] Device or resource busy
```

✅ Root Cause:

- Data API not enabled
- Database (`demo_db`) didn't exist

7️⃣ 📋 Create Database and Table

Using **RDS Query Editor v2**, after enabling Data API:

```
CREATE DATABASE demo_db;
USE demo_db;

CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100)
);

INSERT INTO users (name, email) VALUES ('Siva Sandeep', 'siva@example.com');
```

✅ Data inserted and verified using the editor

8️⃣ 🔍 Query DB Using Lambda + RDS Data API

Connected Lambda to the Aurora DB via `boto3` and Data API:

```
import boto3
import os

client = boto3.client('rds-data')

def lambda_handler(event, context):
    response = client.execute_statement(
        resourceArn=os.environ['DB_ARN'],
        secretArn=os.environ['SECRET_ARN'],
        database='demo_db',
        sql="SELECT * FROM users"
    )
    return {
        'statusCode': 200,
        'body': response['records']
    }
```

✅ Lambda returned the DB records successfully 🎉

🧠 Key AWS Services Used

- **Amazon Aurora Serverless v2** – MySQL-compatible, scalable DB
- **AWS Lambda** – Serverless compute to run backend logic
- **Amazon RDS Data API** – Query RDS without open connections
- **Secrets Manager** – Credential storage and rotation

- **IAM Roles** – Secure access control
- **RDS Query Editor v2** – SQL CLI in AWS Console

Future Improvements

Feature	Description
🔗 API Gateway	Trigger Lambda over HTTP REST calls
📊 CloudWatch	Monitor Lambda execution logs and metrics
🌐 Frontend UI	Display DB records using HTML/React/Vue.js
🔐 VPC Lambda	Explore direct DB connection from VPC (non-Data API)

Workspace Output (Paste Your Results Below)

📝 Paste screenshots, logs, DB outputs, Lambda test results, etc. here:

```
[Screenshot of Lambda test result]
[Secrets Manager screenshot]
[RDS Query Editor with SELECT output]
[IAM Role policy attachment screenshot]
[CloudWatch logs snippet]
```

Summary

You implemented a **fully serverless**, **secure**, and **scalable** data access pattern using:

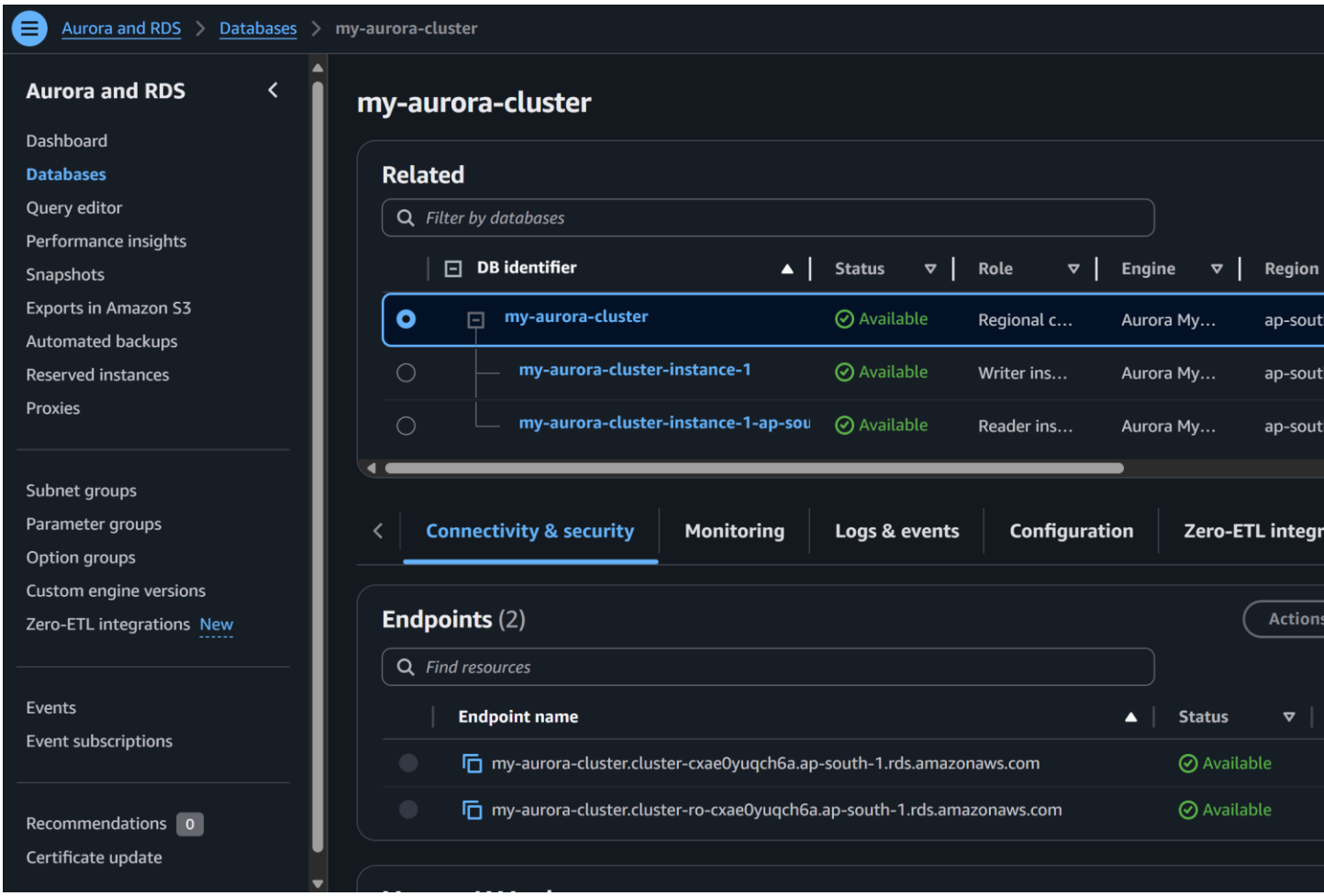
- Aurora Serverless v2 (MySQL)
- AWS Lambda with Python
- RDS Data API for seamless query execution

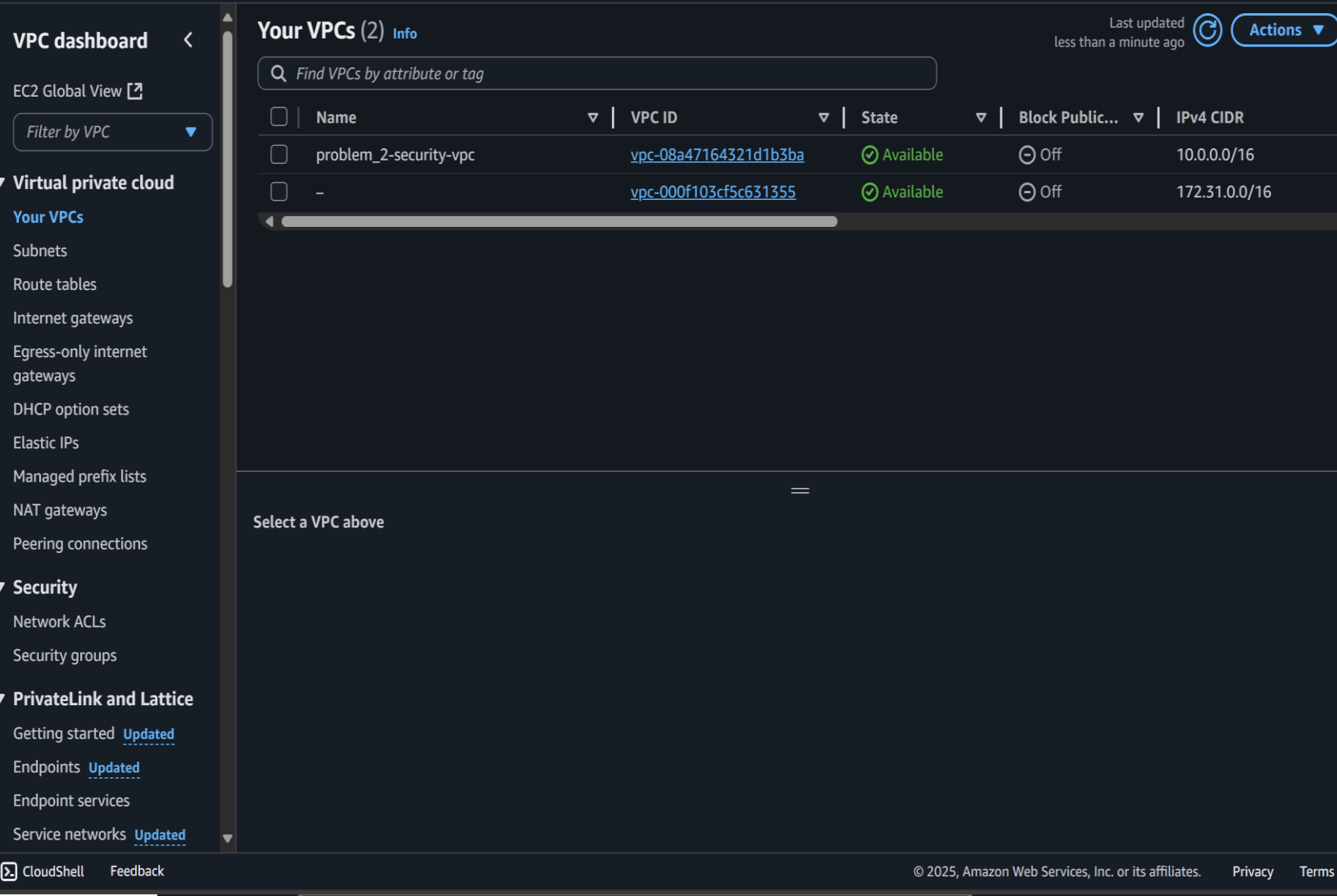
This architecture is perfect for **modern cloud-native applications** and can be used as a reusable template for future microservices or prototypes.

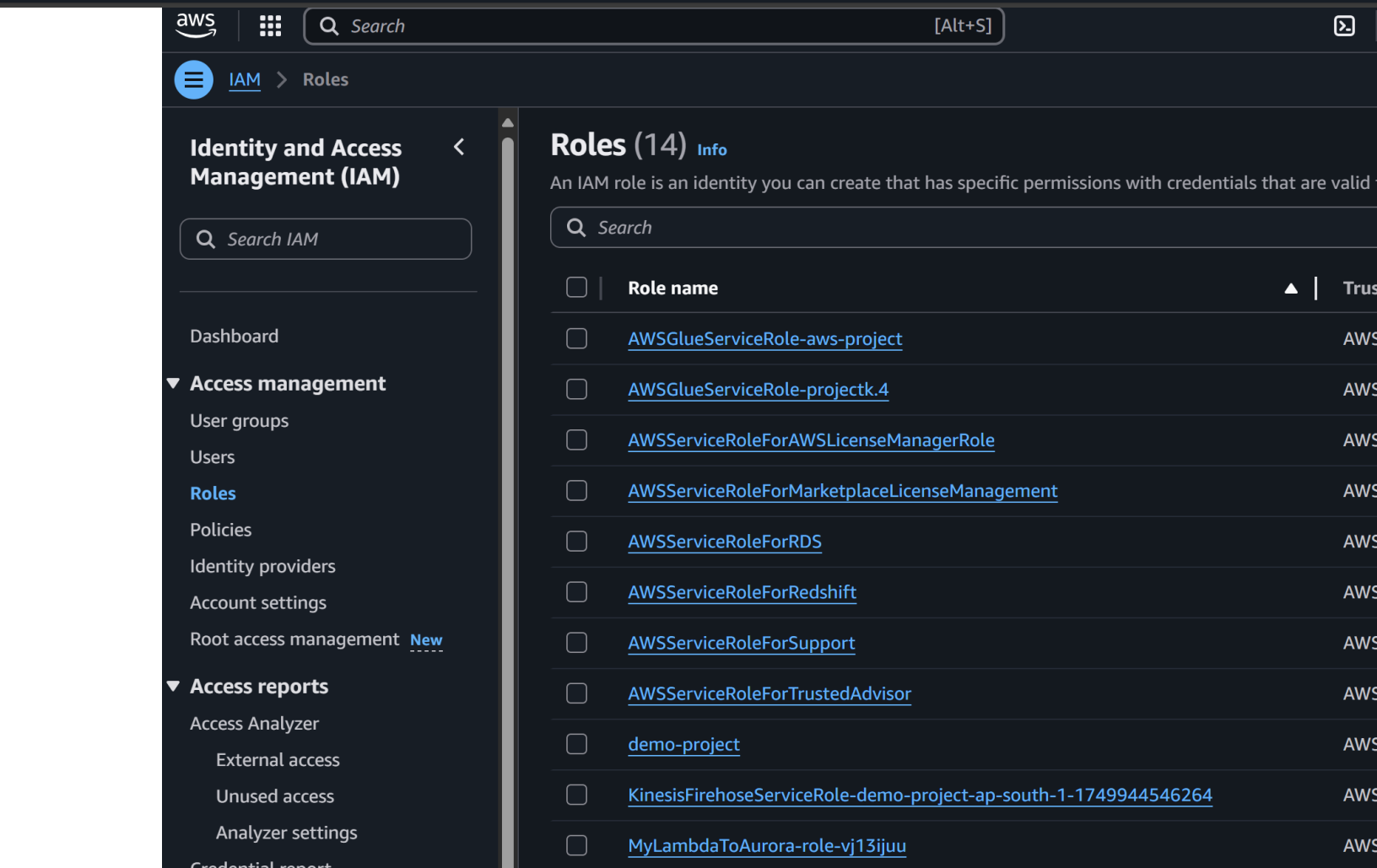
Tags

#AWS #LambdaAuroraIntegration #ServerlessMySQL #PythonPyMySQL #DataAPI #FintechServerless

OUTPUT:


Aurora and RDS > Databases > my-aurora-cluster
Aurora and RDS
Dashboard
Databases
Query editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update
my-aurora-cluster
Related
Filter by databases
DB identifier
Status
Role
Engine
Region
my-aurora-cluster
Available
Regional c...
Aurora My...
ap-sout
my-aurora-cluster-instance-1
Available
Writer ins...
Aurora My...
ap-sout
my-aurora-cluster-instance-1-ap-sou
Available
Reader ins...
Aurora My...
ap-sout
Connectivity & security
Monitoring
Logs & events
Configuration
Zero-ETL integr
Endpoints (2)
Actions
Find resources
Endpoint name
Status
my-aurora-cluster.cluster-cxae0yuqch6a.ap-south-1.rds.amazonaws.com
Available
my-aurora-cluster.cluster-ro-cxae0yuqch6a.ap-south-1.rds.amazonaws.com
Available


VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Security
Network ACLs
Security groups
PrivateLink and Lattice
Getting started Updated
Endpoints Updated
Endpoint services
Service networks Updated
Your VPCs (2) Info
Last updated less than a minute ago
Actions
Find VPCs by attribute or tag
Name
VPC ID
State
Block Public...
IPv4 CIDR
problem_2-security-vpc
vpc-08a47164321d1b3ba
Available
Off
10.0.0.0/16
–
vpc-000f103cf5c631355
Available
Off
172.31.0.0/16
Select a VPC above
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms


aws
Search
[Alt+S]
IAM
Roles
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Roles (14) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid
Search
Role name
AWSGlueServiceRole-aws-project
AWSGlueServiceRole-projectk.4
AWSServiceRoleForAWSLicenseManagerRole
AWSServiceRoleForMarketplaceLicenseManagement
AWSServiceRoleForRDS
AWSServiceRoleForRedshift
AWSServiceRoleForSupport
AWSServiceRoleForTrustedAdvisor
demo-project
KinesisFirehoseServiceRole-demo-project-ap-south-1-1749944546264
MyLambdaToAurora-role-vj13ijuu