AWS VPC and RDS Integration with Lambda Function

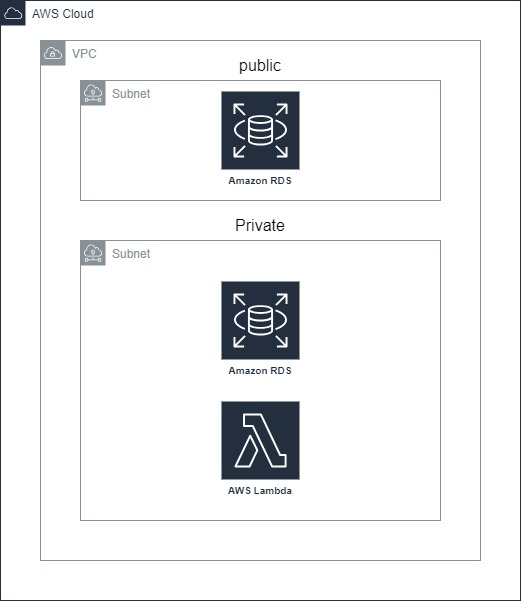
# Objective:

To create a secure and accessible MySQL database setup using AWS VPC, RDS, and Lambda, demonstrating the differences between private and public subnet configurations.

**Technologies used:**

* Amazon VPC
* Amazon RDS (MySQL)
* AWS Lambda
* Python (pymysql)

# Architecture:



# Project Steps:

**Create a Virtual Private Cloud (VPC):**

* Set up a new VPC to logically isolate AWS resources.

**Create Subnets:**

* Public Subnet: For resources that need to be accessible from the internet.
* Private Subnet: For resources that should not be accessible from the internet.

**Create RDS Subnet Groups:**

* Create two RDS subnet groups, one for public access and one for private access using the VPC subnets.

**Create MySQL Database Instances:**

**Private Database Instance:**

* Deployed within the private subnet group.
* Configured with no public access.

**Public Database Instance:**

* Deployed within the public subnet group.
* Configured with public access allowed.

**Create a Lambda Function:**

* Use Python 3.9 for the Lambda function.
* Create a Python layer using Cloud Shell to manage dependencies.

**Lambda Function Code:**

import sys

import logging

import pymysql

rds\_host = "host\_url"

name = "admin"

password = "password"

db\_name = "testdb"

port = 3306

conn = pymysql.connect(host=rds\_host, user=name,password=password,db=db\_name,port=port)

def lambda\_handler(event, context):

"""

This function inserts content into MySQL RDS instance

"""

item\_count = 0

with conn.cursor() as cur:

cur.execute("create table Employee\_test (EmpID int NOT NULL, Name varchar(255) NOT NULL, PRIMARY KEY (EmpID))")

cur.execute('insert into Employee\_test (EmpID, Name) values(1, "Joe")')

cur.execute('insert into Employee\_test (EmpID, Name) values(2, "Bob")')

cur.execute('insert into Employee\_test (EmpID, Name) values(3, "Mary")')

conn.commit()

cur.execute("select \* from Employee\_test")

for row in cur:

item\_count += 1

return "Added %d items to RDS MySQL table" %(item\_count)

```

**Deploy and Test the Lambda Function:**

* Initial Deployment: The Lambda function could not connect to the database.
* Resolved Deployment: Deployed the Lambda function within the same VPC using private subnets, which allowed successful connectivity and execution.

**Accessing the Database:**

Private Database: Not accessible directly from tools like DBeaver due to security constraints.

Public Database: Created another database with a public subnet and a security group allowing inbound traffic, making it accessible from anywhere.

**Challenges and Solutions:**

**Challenge:** Initial inability to connect the Lambda function to the RDS instance.

**Solution:** Ensured the Lambda function was deployed within the same VPC and private subnets, allowing internal access to the RDS instance.

# Outcome:

Successfully set up a secure and scalable database environment using AWS VPC, RDS, and Lambda. Demonstrated the importance of subnet configurations for security and accessibility.

