Step1: create Pvt RDS

Step 2: Go to secret manager and create a secret. Here we have to give our RDS infra like username and password and selected the DB.

Step-by-Step: Configure AWS Secrets Manager for Amazon RDS (MySQL)

This guide walks through securely storing and automatically rotating RDS credentials using AWS Secrets Manager.

✅ Step 1: Open Secrets Manager

1. Go to the **AWS Console**.
2. Search and open **Secrets Manager**.
3. Click on **Store a new secret**.

✅ Step 2: Choose Secret Type

- **Secret type**: Credentials for Amazon RDS database

Enter your credentials:

- **Username**: `admin`
- **Password**: `******`

✅ Step 3: Choose Database

- **Select your RDS database**: `mydbinstance`
- Click **Next**

✓ Step 4: Configure Secret

- **Secret name**: `dbsecret`
- Optionally, provide a description.
- Click **Next**

✓ Step 5: Configure Automatic Rotation

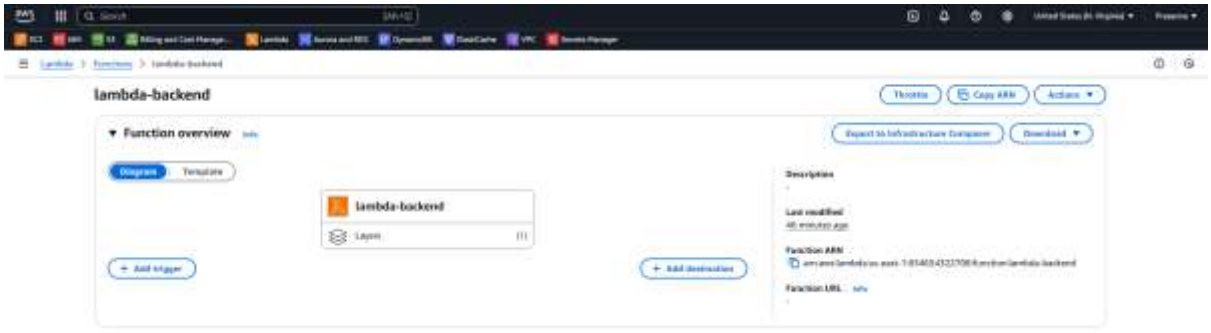
1. Enable **Automatic rotation**.
2. **Rotation schedule**:
    - **Time unit**: `Weeks`
    - **Weeks**: `1`
    - **Day of the week**: `Mondays`
3. **Rotation function**:
    - Use the existing Lambda template: `secretsmanager: mysql`

Click **Next** and then **Store**.

🎉 Done!

You've now securely stored your RDS credentials and enabled automatic rotation using AWS Secrets Manager.

Step 3: create Lambda a function with python as runtime.



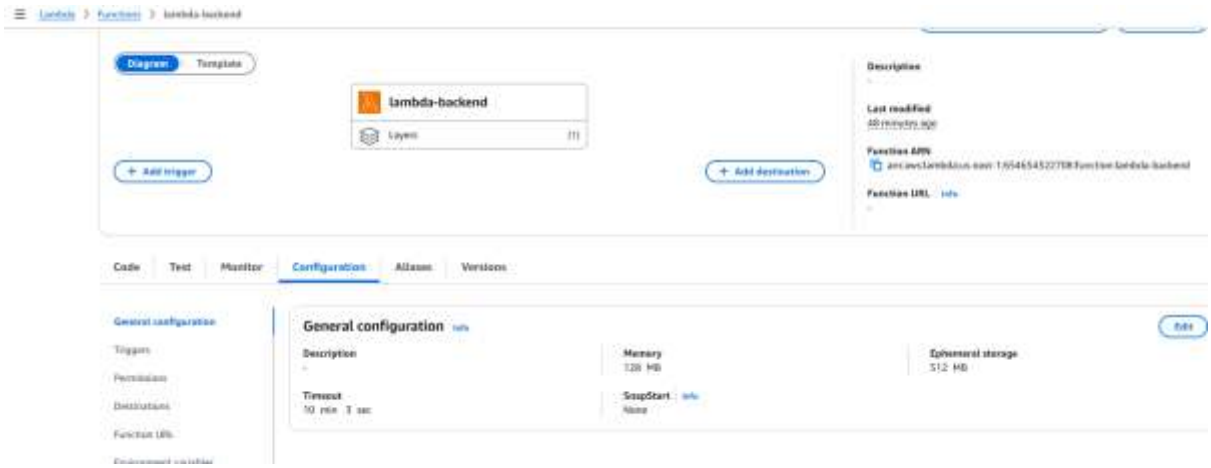
Step 4: Then we have to paste the code with our secret name.


```
import json
import boto3
import pymysql

# Define the database and table name
new_db_name = "test"
table_name = "mytable"

# Function to retrieve secret from AWS Secrets Manager
def get_secret(secret_name):
    client = boto3.client('secretsmanager')
    response = client.get_secret_value(SecretId=secret_name)
    return json.loads(response['SecretString'])

# Function to connect to RDS using secret credentials
def connect_to_rds(secret):
    connection = pymysql.connect(
        host=secret['host'],       # Ensure your secret includes 'host'
        user=secret['username'],
        password=secret['password'],
        cursorclass=pymysql.cursors.DictCursor
    )
    return connection
```

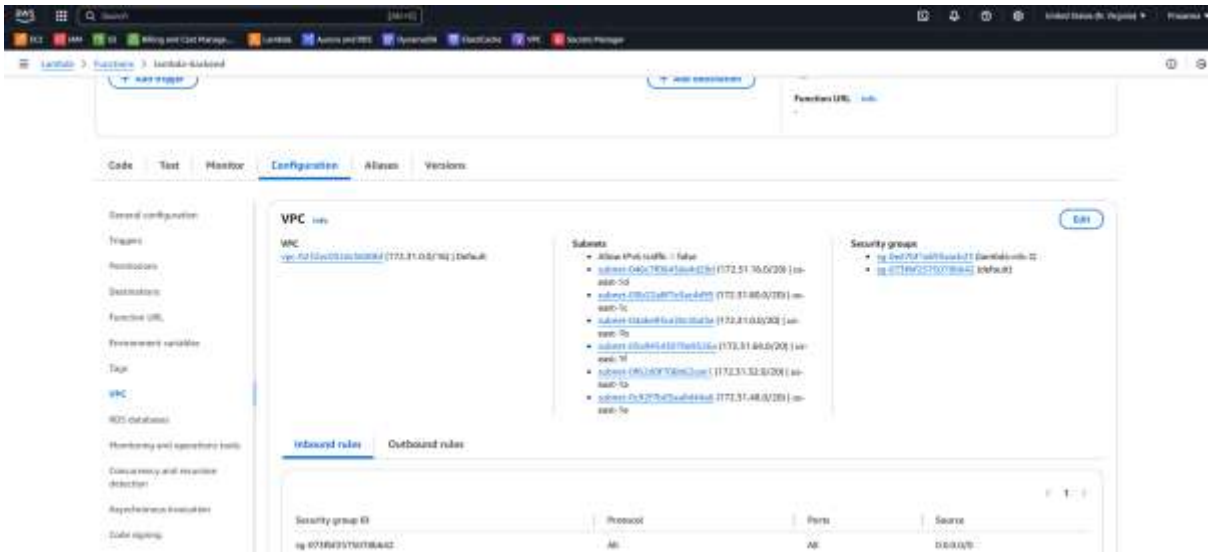
```python
# Lambda function handler
def lambda_handler(event, context):
    secret_name = "dbsecret"  # Replace with your actual secret name
    connection = None

    try:
        # Get DB credentials from Secrets Manager
        secret = get_secret(secret_name)

        # Connect to RDS
        connection = connect_to_rds(secret)

        with connection.cursor() as cursor:
            # Create database if not exists
            cursor.execute(f"CREATE DATABASE IF NOT EXISTS {new_db_name};")
            cursor.execute(f"USE {new_db_name};")

            # Create table if not exists
            create_table_sql = f"""
            CREATE TABLE IF NOT EXISTS {table_name} (
                id INT AUTO_INCREMENT PRIMARY KEY,
                name VARCHAR(255) NOT NULL,
                created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
            );
            """
            cursor.execute(create_table_sql)

        return {
            'statusCode': 200,
            'body': f"Database '{new_db_name}' and table '{table_name}'
created successfully."
        }

    except Exception as e:
        print("Error:", str(e))
        return {
            'statusCode': 500,
            'body': str(e)
        }

    finally:
        if connection:
            connection.close()
```

Step 5: create a new custom layer with the python zip file and add to Lambda function.

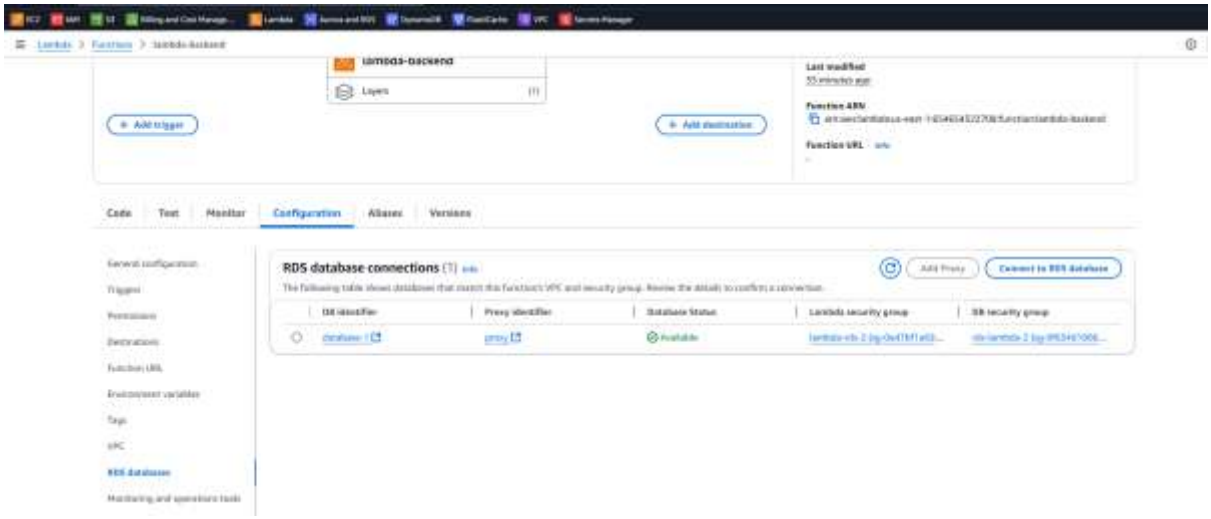
Step: 6: Then go to the configuration, increase the time to 10 mins



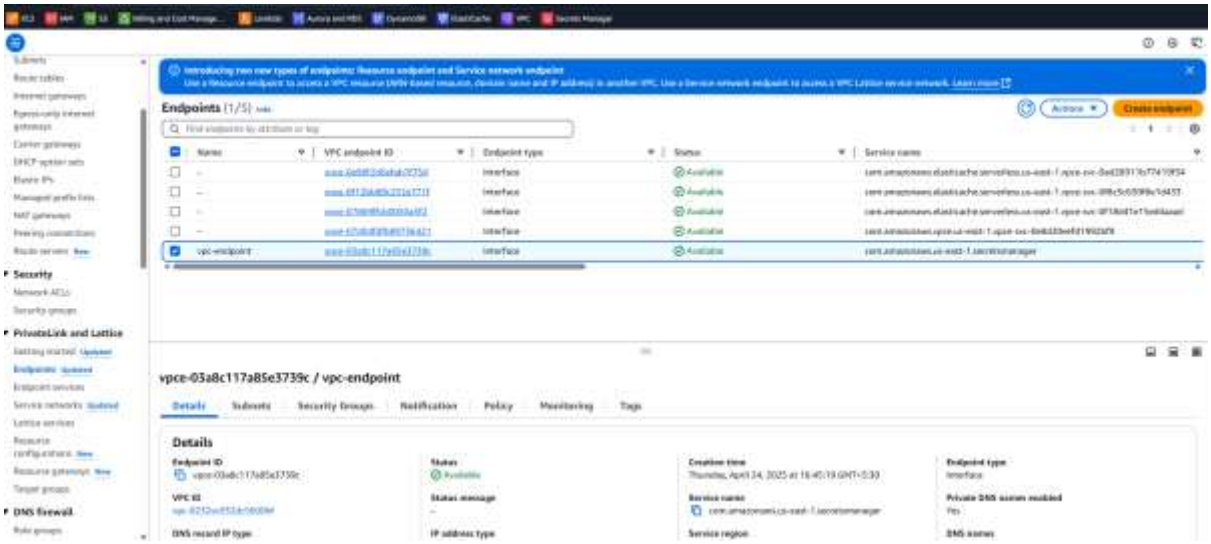
Step:7: go to the VPC option and give our VPC details.



Step 8: Go to RDS Databases under configuration, add our RDS here.



Step 7: Go to VPC endpoints in VPC and there we have to select secret option and give our VPC and private subnets which we have given for RDS.



Step 8: Output : test the code