

## **Guided Lab: Connecting AWS Lambda to Amazon RDS**

### **Description**

In modern cloud applications, serverless computing with AWS Lambda is often combined with managed databases like Amazon RDS to build scalable, cost-efficient solutions. Connecting Lambda functions to RDS allows you to execute database queries, manage data, and build data-driven applications without managing servers. This lab demonstrates how to securely connect a Lambda function to an RDS instance using environment variables to manage database connection settings.

### **Why do this?**

Connecting Lambda to RDS allows you to build dynamic, data-driven applications where Lambda handles the backend logic, and the data is stored and managed in RDS. This setup is ideal for use cases like API backends, serverless data processing, and real-time applications where you need scalable and secure access to a relational database.

### **Prerequisites**

This lab assumes you have a basic understanding of AWS Lambda, Amazon RDS Database, basic networking in AWS, and Python programming.

If you find any gaps in your knowledge, consider taking the following lab:

- Creating an AWS Lambda function
- Using Environment Variables in AWS Lambda
- Creating an Amazon RDS database
- Security Group VS Network Access Control List

### **Objectives**

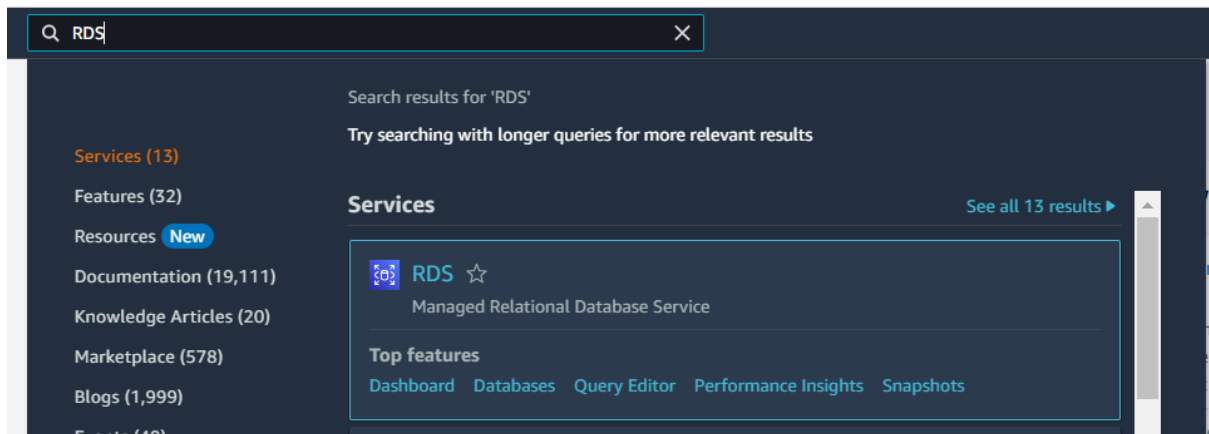
In this lab, you will:

- Create an RDS instance
- Configure a Lambda function to connect to the RDS instance.
- Securely store database credentials using environment variables.
- Write and execute a simple SQL query from the Lambda function.

### **Lab Steps**

#### **Create an Amazon RDS Instance**

1. Navigate to the RDS Console.



## 2. Create a Database with the following configuration:

- **Choose a database creation method:** Standard create
- **Engine options:**
  - **Engine type:** MySQL
  - **Engine Version:** Leave it as default

## Choose a database creation method [Info](#)

### ☒ Standard create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

### ☐ Easy create

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

## Engine options

### Engine type [Info](#)

#### ☐ Aurora (MySQL Compatible)



#### ☐ Aurora (PostgreSQL Compatible)



#### ☒ MySQL



#### ☐ MariaDB



#### ☐ PostgreSQL



#### ☐ Oracle

ORACLE®

#### ☐ Microsoft SQL Server



#### ☐ IBM Db2

IBM Db2

### Edition

#### ☒ MySQL Community

### Engine version [Info](#)

View the engine versions that support the following database features.

#### ▼ Hide filters

#### ☒ Show versions that support the Multi-AZ DB cluster [Info](#)

Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

#### ☒ Show versions that support the Amazon RDS Optimized Writes [Info](#)

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

### Engine Version

MySQL 8.0.35



#### ☐ Enable RDS Extended Support [Info](#)

Amazon RDS Extended Support is a paid offering [\[?\]](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#) [\[?\]](#).

- **Templates:** Free tier
  - **Settings:**
    - **DB instance identifier:** Kabayan-DB
    - **Master username:** admin (you can change this as you desire)
    - **Credentials management:** Self managed
    - For the simplicity of this lab we will tick the checkbox of **Auto generate password**.

*But take note that in PRODUCTION, you need to create a strong password for the security of your Database*

### 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 60 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.

☒ **Auto generate password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**i** You can view your credentials after you create your database. Click the 'View credential details' in the database creation banner to view the password.

- **Instance configuration:**
  - **DB instance size:**, db.t3.micro
- **Storage:**
  - **Storage type:** General Purpose SSD (gp2)
  - **Allocated storage:** 20

## 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)

db.t3.micro  
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

## 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

The minimum value is 20 GiB and the maximum value is 6,144 GiB

**i** After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. [Learn more](#)

### ► Storage autoscaling

- **Connectivity:**
  - **Compute resource:** Don't connect to an EC2 compute resource
  - **Virtual private cloud (VPC):** leave it as the default
  - **DB subnet group:** Create new DB Subnet Group
  - **Public access:** Yes

Ensure the RDS instance is publicly accessible if you plan to connect from Lambda within the same VPC

- - **VPC security group (firewall):** Select Create new
    - **New VPC security group name:** DB-SG

- **Availability Zone:** Select your preferred one (e.g., us-east-1a)

### Connectivity [Info](#)

**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.

vpc-0b91cff63a365f28b  
3 Subnets, 3 Availability Zones

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.

Create new DB Subnet Group

**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

**New VPC security group name**

DB-SG

**Availability Zone [Info](#)**

us-east-1a

- Leave the rest as default and click **Create Database**.

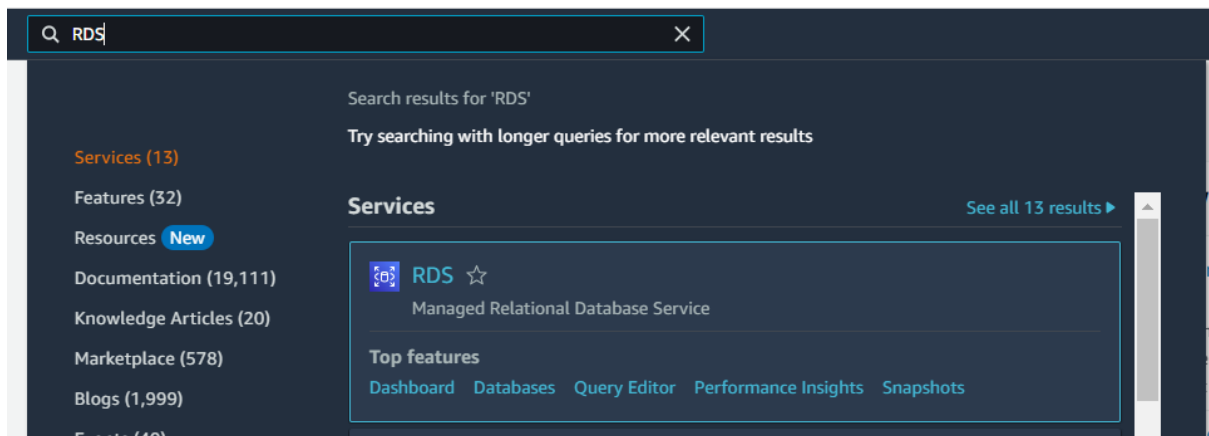
*This will take a few minutes to finish.*

**3. Note down the database endpoint, username, and password** for later use.

## Lab Steps

### Create an Amazon RDS Instance

1. Navigate to the RDS Console.



2. Create a Database with the following configuration:

- **Choose a database creation method:** Standard create
- **Engine options:**
  - **Engine type:** MySQL
  - **Engine Version:** Leave it as default

## Choose a database creation method [Info](#)

### ☒ Standard create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

### ☐ Easy create

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

## Engine options

### Engine type [Info](#)

#### ☐ Aurora (MySQL Compatible)



#### ☐ Aurora (PostgreSQL Compatible)



#### ☒ MySQL



#### ☐ MariaDB



#### ☐ PostgreSQL



#### ☐ Oracle

ORACLE®

#### ☐ Microsoft SQL Server



#### ☐ IBM Db2

IBM Db2

### Edition

#### ☒ MySQL Community

### Engine version [Info](#)

View the engine versions that support the following database features.

#### ▼ Hide filters

#### ☒ Show versions that support the Multi-AZ DB cluster [Info](#)

Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

#### ☒ Show versions that support the Amazon RDS Optimized Writes [Info](#)

Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

### Engine Version

MySQL 8.0.35



#### ☐ Enable RDS Extended Support [Info](#)

Amazon RDS Extended Support is a paid offering [\[?\]](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#) [\[?\]](#).



- **Templates:** Free tier
  - **Settings:**
    - **DB instance identifier:** Kabayan-DB
    - **Master username:** admin (you can change this as you desire)
    - **Credentials management:** Self managed
    - For the simplicity of this lab we will tick the checkbox of **Auto generate password**.

*But take note that in PRODUCTION, you need to create a strong password for the security of your Database*

## 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 60 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.

☒ **Auto generate password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**i** You can view your credentials after you create your database. Click the 'View credential details' in the database creation banner to view the password.

- **Instance configuration:**
  - **DB instance size:**, db.t3.micro
- **Storage:**
  - **Storage type:** General Purpose SSD (gp2)
  - **Allocated storage:** 20

## 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)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

## 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

The minimum value is 20 GiB and the maximum value is 6,144 GiB

**i** After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. [Learn more](#)

### ► Storage autoscaling

- **Connectivity:**
  - **Compute resource:** Don't connect to an EC2 compute resource
  - **Virtual private cloud (VPC):** leave it as the default
  - **DB subnet group:** Create new DB Subnet Group
  - **Public access:** Yes

Ensure the RDS instance is publicly accessible if you plan to connect from Lambda within the same VPC

- - **VPC security group (firewall):** Select Create new
    - **New VPC security group name:** DB-SG

- **Availability Zone:** Select your preferred one (e.g., us-east-1a)

### Connectivity [Info](#)

**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.

vpc-0b91cff63a365f28b  
3 Subnets, 3 Availability Zones

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.

Create new DB Subnet Group

**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

**New VPC security group name**

DB-SG

**Availability Zone [Info](#)**

us-east-1a

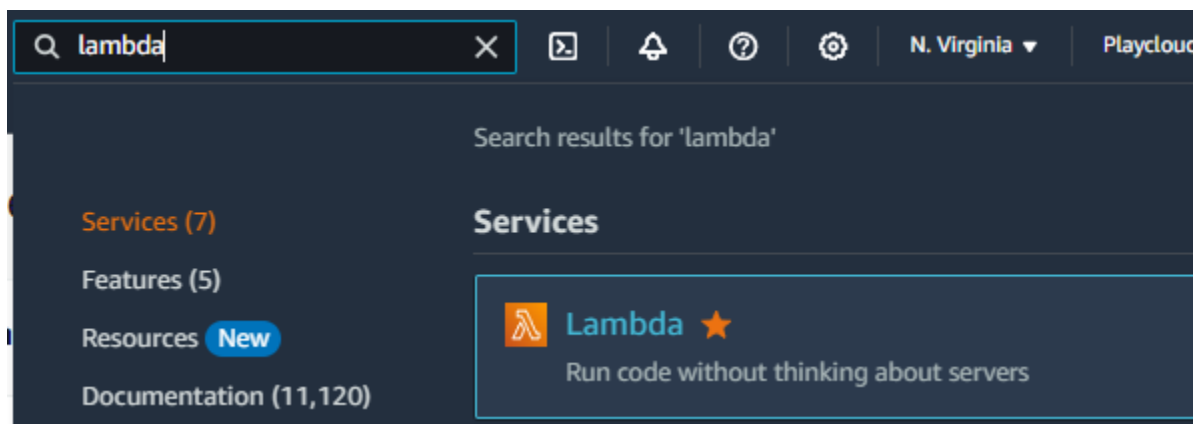
- Leave the rest as default and click **Create Database**.

*This will take a few minutes to finish.*

**3. Note down the database endpoint, username, and password** for later use.

## Create the Lambda Function

1. Navigate to the AWS Lambda Console



2. Create a new Lambda function using the following configurations:

- Choose **Author from scratch**.
- Function name: myLambdaFunction
- Select Python 3.12 as the runtime.
- **Execution role:**
  - Select Use an Existing Role: PlayCloud-Sandbox

## Basic information

### Function name

Enter a name that describes the purpose of your function.

myLambdaFunction

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

### 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 3.12

### Architecture [Info](#)

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

☒ x86\_64

☐ arm64

### 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.

PlayCloud-Sandbox

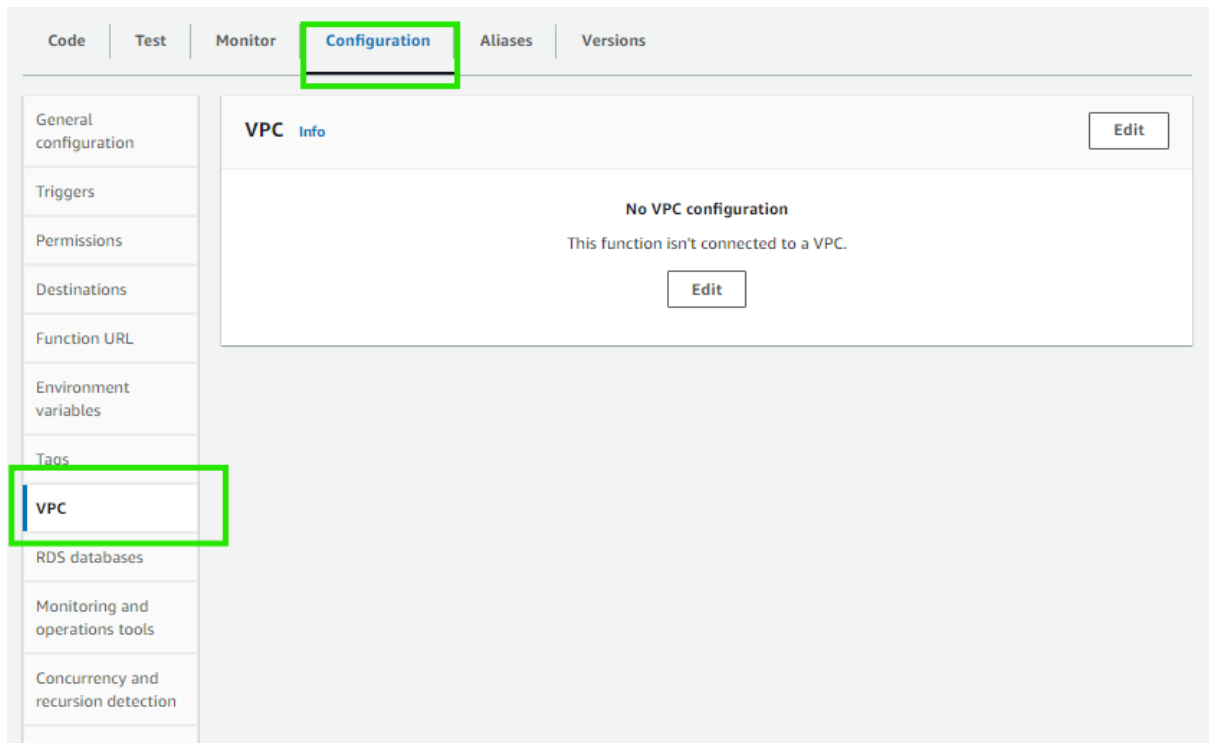
[View the PlayCloud-Sandbox role](#) on the IAM console.

- Click **Create function**

## Configure VPC and Security Groups

1. Ensure that your Lambda function and RDS instance are in the same VPC.
2. Modify the Lambda function's VPC settings to include the necessary subnets and security groups to communicate with the RDS instance.

- Navigate to the **Configuration tab** and click on **VPC**.



- Click on Edit.
- Add the following setting:
  - **VPC:** Choose the default
  - **Subnets:** Choose the same subnets where you set the RDS Instance.
  - **Security groups:** Choose the DB-SG, the same SG as the RDS Instance.

### VPC Info

Choose a VPC for your function to access.

vpc-0b91cff63a365f28b (192.168.5.0/26) ▼



☐ 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 ▼



subnet-05dd984809a960943 (192.168.5.0/28) us-east-1a ✕  
aws:cloudformation:logical-id: PublicSubnet1  
aws:cloudformation:stack-id: arn:aws:cloudformation:us-east-1:533267157116:stack/StackSet-baseline-62f8b10a-2421-4f01-b82a-f160a36ee4ee/f465d6c0-b48c-11ee-aadf-126d9b281ee7  
aws:cloudformation:stack-name: StackSet-baseline-62f8b10a-2421-4f01-b82a-f160a36ee4ee  
Baseline: True

subnet-0eb3774bde866be97 (192.168.5.16/28) us-east-1b ✕  
aws:cloudformation:logical-id: PublicSubnet2  
aws:cloudformation:stack-id: arn:aws:cloudformation:us-east-1:533267157116:stack/StackSet-baseline-62f8b10a-2421-4f01-b82a-f160a36ee4ee/f465d6c0-b48c-11ee-aadf-126d9b281ee7  
aws:cloudformation:stack-name: StackSet-baseline-62f8b10a-2421-4f01-b82a-f160a36ee4ee  
Baseline: True

subnet-0429348038cb4d820 (192.168.5.32/28) us-east-1c ✕  
aws:cloudformation:logical-id: PublicSubnet3  
aws:cloudformation:stack-id: arn:aws:cloudformation:us-east-1:533267157116:stack/StackSet-baseline-62f8b10a-2421-4f01-b82a-f160a36ee4ee/f465d6c0-b48c-11ee-aadf-126d9b281ee7  
aws:cloudformation:stack-name: StackSet-baseline-62f8b10a-2421-4f01-b82a-f160a36ee4ee  
Baseline: True

### Security groups

Choose the VPC security groups for Lambda to use to set up your VPC configuration. The table below shows the inbound and outbound rules for the security groups that you choose.

Choose security groups ▼



sg-0b5962429357be062 (DB-SG) ✕  
Created by RDS management console

Inbound rules

Outbound rules

- - **Click Save** and wait for the function to update. Ensure the RDS security group allows inbound connections on port **3306** for MySQL.







Successfully updated the function myLambdaFunction.

Function overview [Info](#) [Export to Application Composer](#) [Download](#)

**Diagram** **Template**


 myLambdaFunction

 Layers (0)

[+ Add trigger](#) [+ Add destination](#)

Description  
-

Last modified  
45 minutes ago

Function ARN  
 arn:aws:lambda:us-east-1:533267157116:function:myLambdaFunction

Function URL [Info](#)  
-

**Code** **Test** **Monitor** **Configuration** **Aliases** **Versions**

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

**VPC** [Info](#) [Edit](#)

VPC  
vpc-0b91cff63a365f28b (192.168.5.0/26)

Subnets

- Allow IPv6 traffic = false
- subnet-05dd984809a960943 (192.168.5.0/28) | us-east-1a
- subnet-0eb3774bde866be97 (192.168.5.16/28) | us-east-1b
- subnet-0429348038cb4d820 (192.168.5.32/28) | us-east-1c

Security groups

- sg-0b5962429357be062 (DB-SG)

[Inbound rules](#) [Outbound rules](#)

Security group ID	Protocol	Ports	Source
sg-0b5962429357be062	Custom TCP	3306	119.111.224.153/32

- You can also double-check this by clicking the security group.

**Code** **Test** **Monitor** **Configuration** **Aliases** **Versions**

General configuration  
Triggers  
Permissions  
Destinations  
Function URL  
Environment variables  
Tags  
**VPC**

**VPC** [Info](#) [Edit](#)

VPC  
vpc-0b91cff63a365f28b (192.168.5.0/26)

Subnets

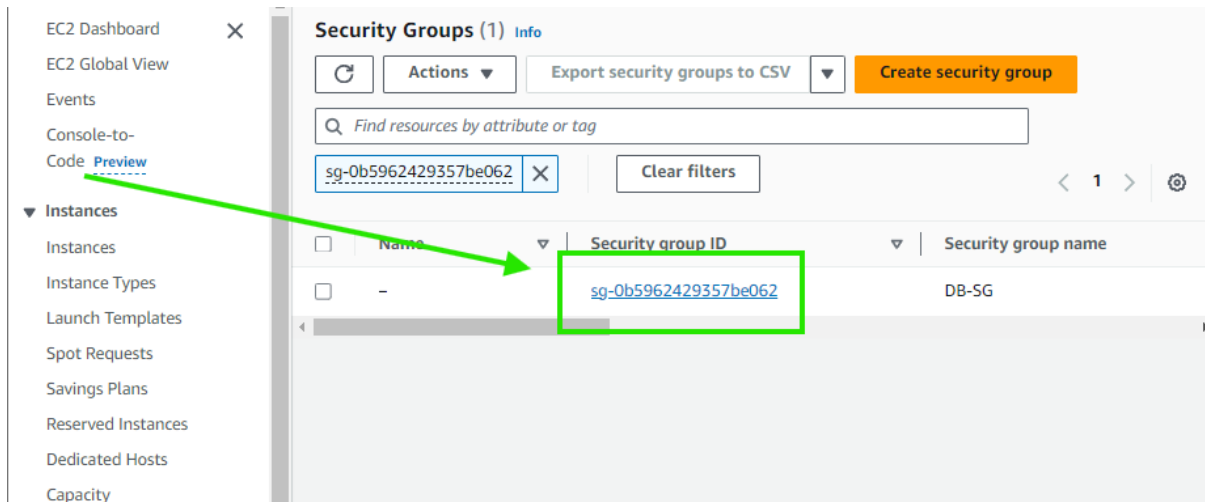
- Allow IPv6 traffic = false
- subnet-05dd984809a960943 (192.168.5.0/28) | us-east-1a
- subnet-0eb3774bde866be97 (192.168.5.16/28) | us-east-1b
- subnet-0429348038cb4d820 (192.168.5.32/28) | us-east-1c

Security groups

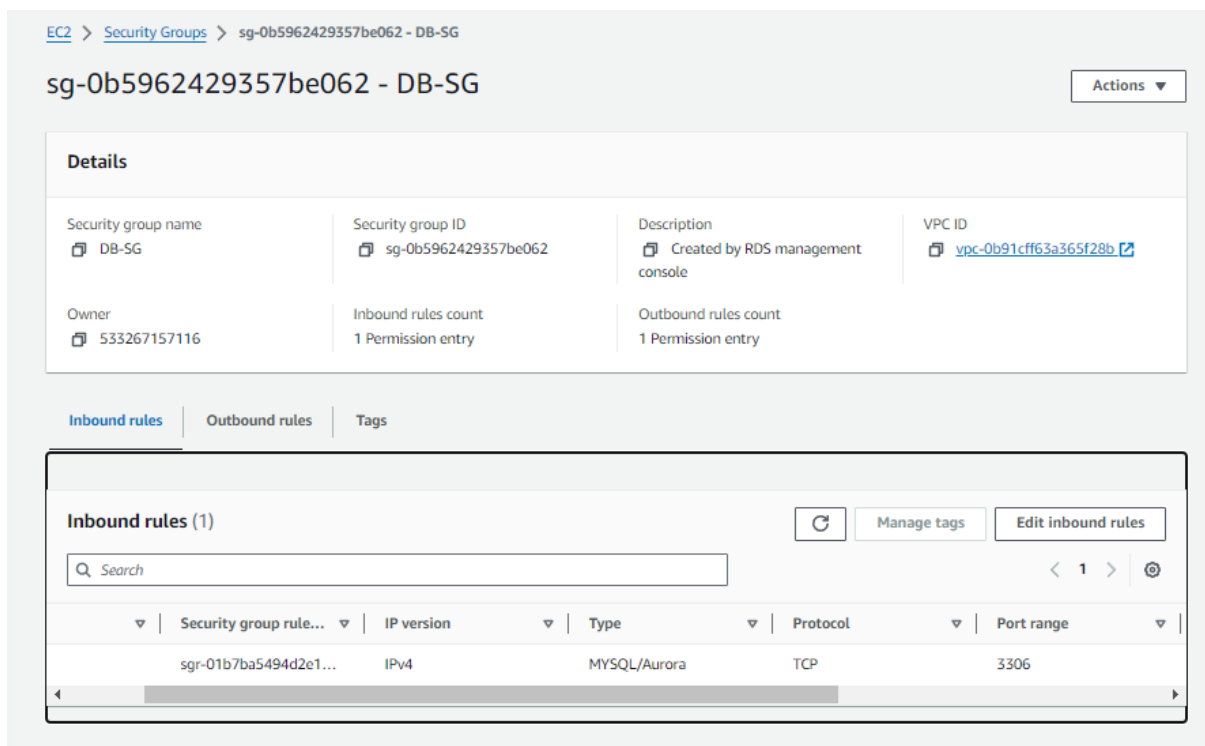
- sg-0b5962429357be062 (DB-SG)

[Inbound rules](#) [Outbound rules](#)

- You will then be redirected to the security group tab. Click on the Security group ID.



- You should see the current Inbound rule, Port range 3306 for MySQL/Aurora, with your IP as the source. If you cannot see this, add the inbound rules for this setting. You can also set the source as **anywhere IPv4** for this lab.



## Upload the Lambda Function Code

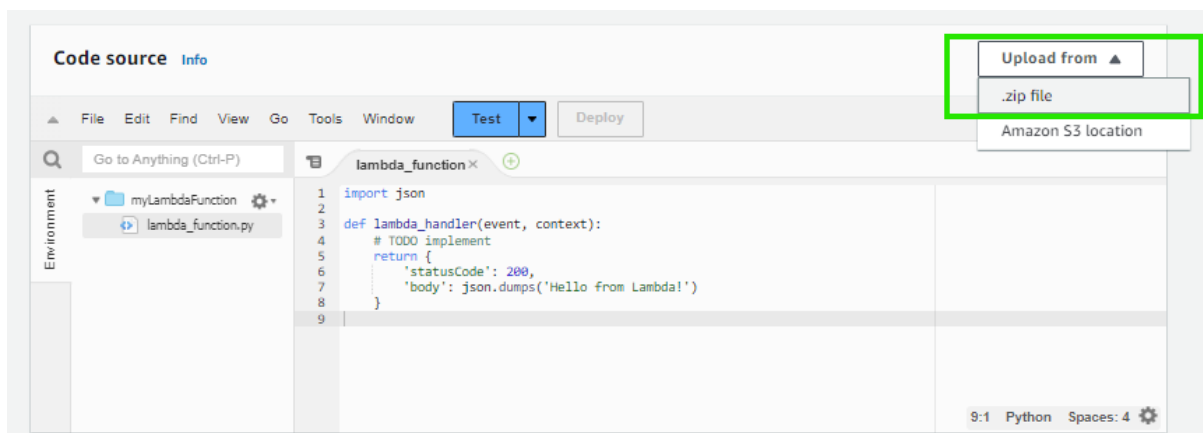
- Navigate back to the Lambda Function Console to the **Code** tab.

**Note:** We are using the **old console editor** for this lab. You're welcome to use either the old or new editor, whichever you prefer; the steps remain the same, though the interface may have a slightly different appearance in the new editor.

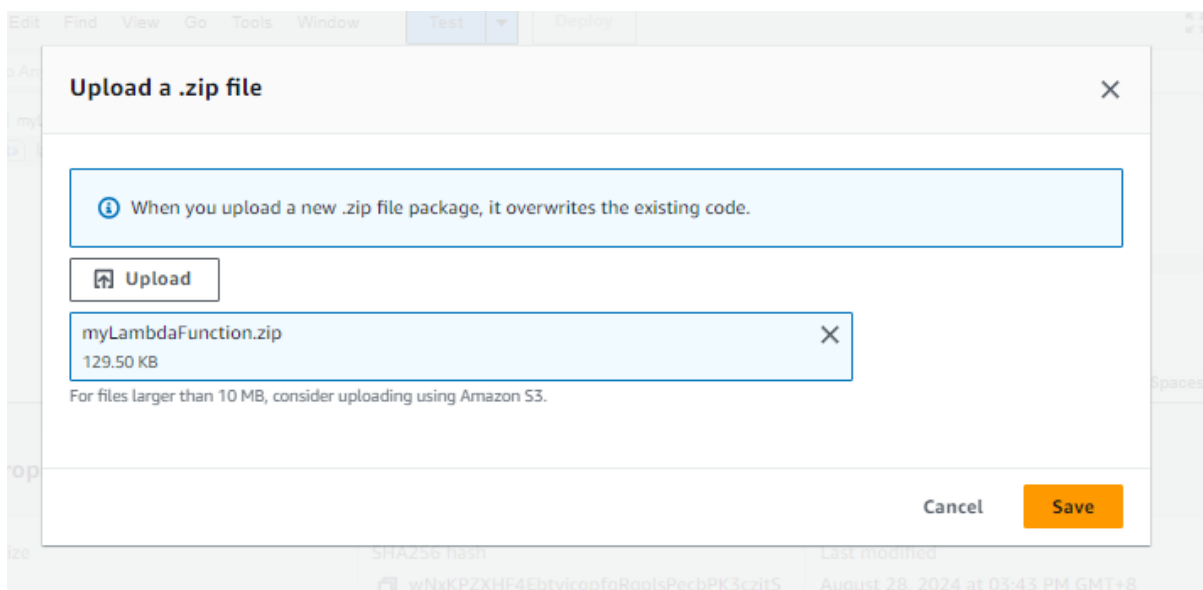
- Download the following zip file:

<https://media.tutorialsdojo.com/public/myLambdaFunction.zip>

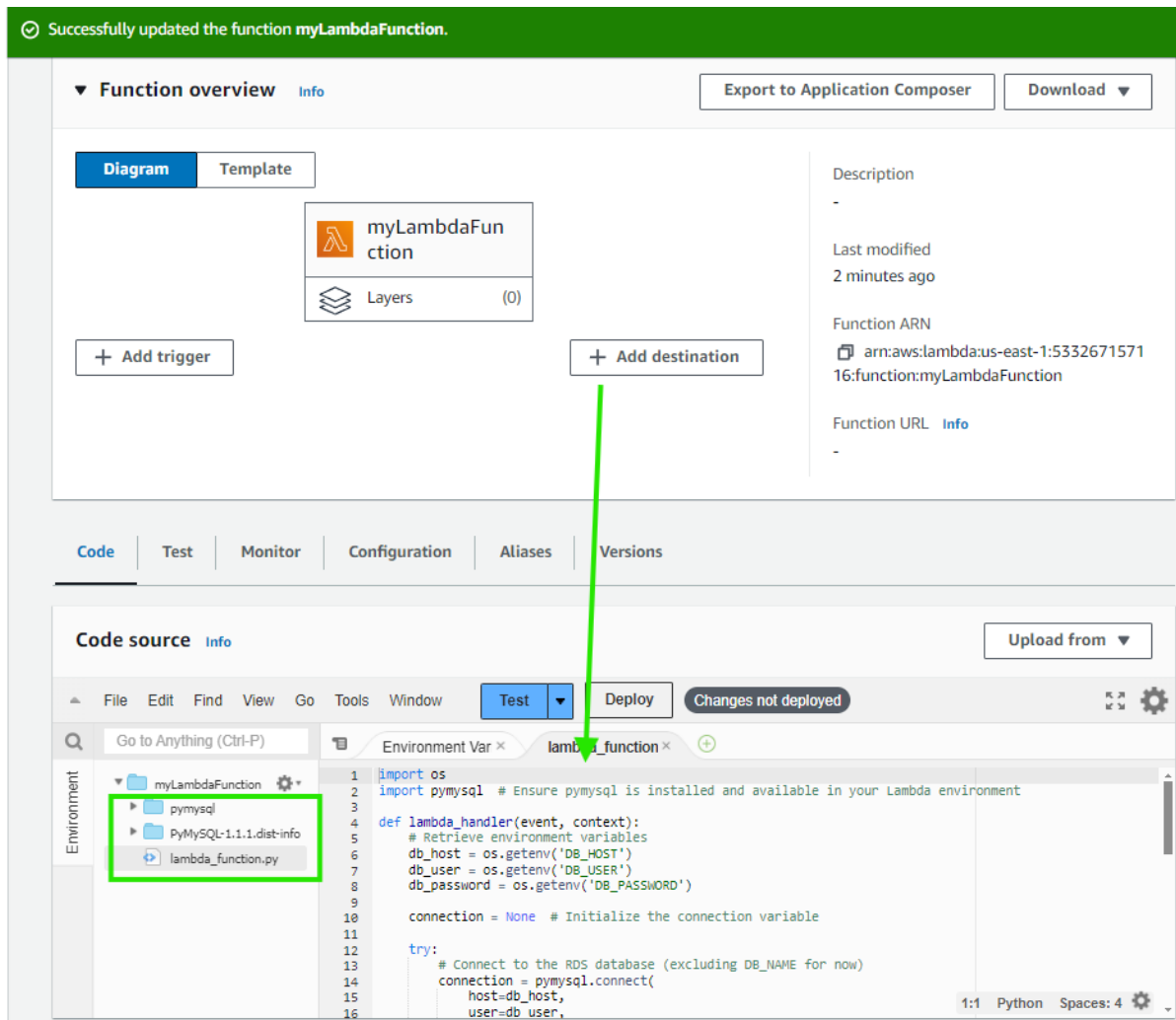
3. Click the upload from the button and select the .zip file



4. Click upload, select where you downloaded the zip file earlier from your local machine, and then click on save.



5. Your code should now be updated.



6. Take your time to review what is added to your code and lambda environment.

This AWS Lambda function, utilizing the pymysql module, connects to a MySQL database using credentials stored in environment variables. It performs a simple query, `SHOW DATABASES;`, to retrieve a list of all databases on the MySQL server and checks if a specified database exists. Based on the result, it returns an HTTP 200 status code if the database is found, a 404 status if not, or a 500 status in case of a connection error. The connection is safely closed after the operation.

#### 7. Add Environment Variables to the Lambda Function

- In the Lambda console, navigate to the "Configuration" tab and add the following environment variables:
  - `DB_HOST`: The endpoint of your RDS instance.
  - `DB_USER`: The database username.
  - `DB_PASSWORD`: The database password.
  - `DB_NAME` : **MySQL**

- Click **Save**

The screenshot shows the AWS Lambda console with the **Configuration** tab selected. On the left sidebar, the **Environment variables** option is highlighted. The main content area displays **Environment variables (4)** with a note: "The environment variables below are encrypted at rest with the default Lambda service key." Below this is a search bar and a table of variables.

Key	Value
DB_HOST	kabayan-db.ctkuoim22etu.us-east-1.rds.amazonaws.com
DB_NAME	mysql
DB_PASSWORD	gW5SJUofpoKBSgdzYUQ4
DB_USER	admin

8. Lastly, click on the General configuration.

The screenshot shows the AWS Lambda console with the **Configuration** tab selected. On the left sidebar, the **General configuration** option is highlighted with a green box. The main content area displays **General configuration** with an **Edit** button. The configuration details are as follows:

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout 0 min 3 sec	SnapStart <a href="#">Info</a> None	

- change the **Timeout** to 1 min and save it

This screenshot shows the **General configuration** section after the timeout has been updated. The **Timeout** is now set to **1 min 0 sec**.

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout 1 min 0 sec	SnapStart <a href="#">Info</a> None	

## Set up Lambda connection

1. Navigate back to the RDS Instance you created earlier.

Amazon 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

RDS > Databases > kabayan-db

kabayan-db

↻

Modify

Actions ▼

Summary

DB identifier kabayan-db	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU <div>2.77%</div>	Class db.t3.micro	Current activity <div>0</div> Connections	Region & AZ us-east-1a	

< Connectivity & security

Monitoring

Logs & events

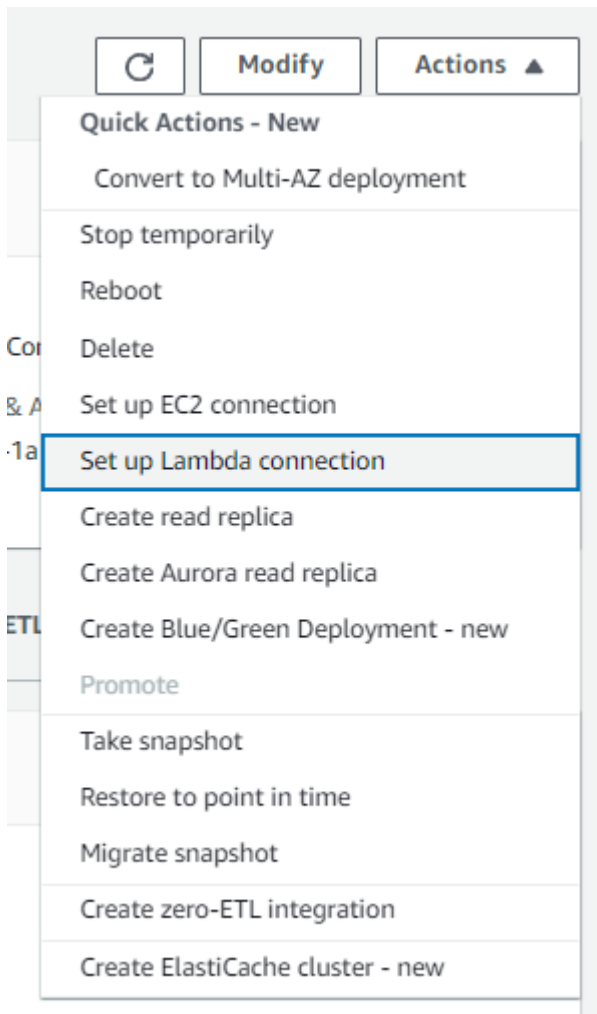
Configuration

Zero-ETL integrations >

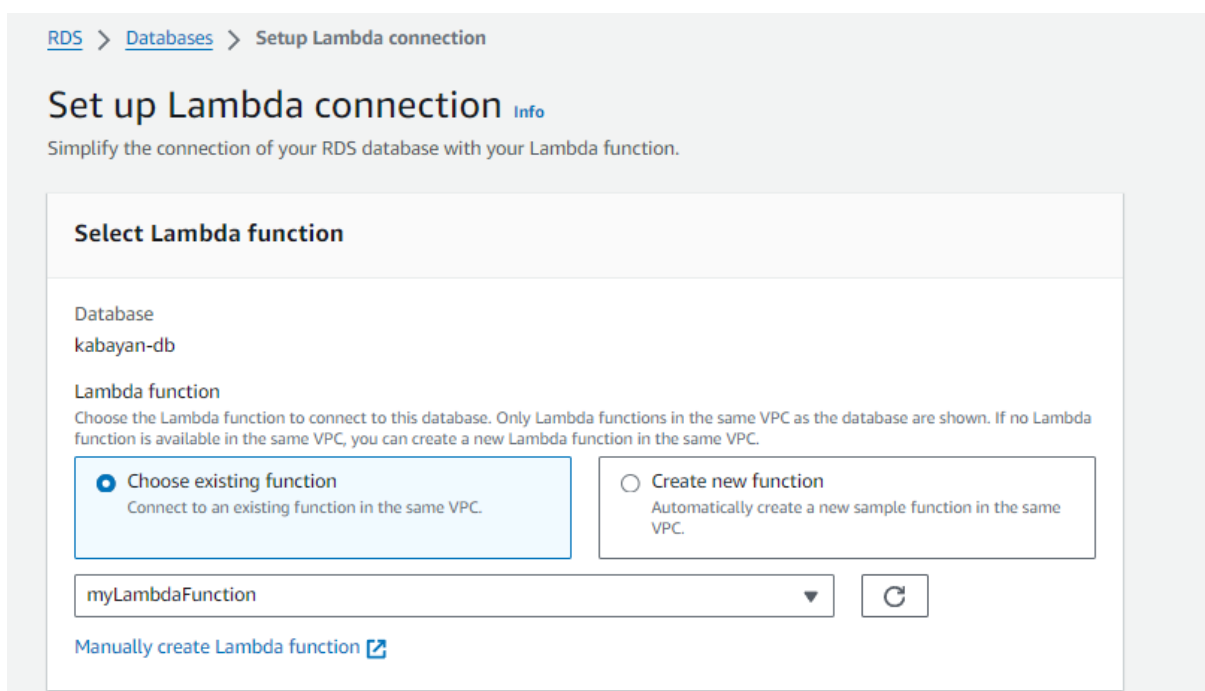
Connectivity & security

Endpoint & port	Networking	Security
Endpoint kabayan-db.ctkuoi m22etu.us-east-1.rds.amazonaws.com	Availability Zone us-east-1a	VPC security groups DB-SG (sg-0b5962429357be062)
Port 3306	VPC vpc-0b91cff63a365f28b	Active
	Subnet group default-vpc-0b91cff63a365f28b	Publicly accessible Yes
		Certificate authority

2. Click on the **Actions** dropdown and select **Set up Lambda connection**.



3. Add the Lambda Function we created earlier (i.e myLambdaFunction)



4. For this lab, **uncheck** the **Connect using RDS proxy**

**RDS Proxy** [Info](#)

You can use RDS Proxy to manage the connection between your database and your Lambda function. An RDS proxy simplifies connection management and makes applications more resilient to database failures.

RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

☐ **Connect using RDS Proxy**  
RDS automatically creates an IAM role for the proxy. The proxy also requires a Secrets Manager secret. If you don't have a Secrets Manager secret, RDS can create one for you.

[Manually create new RDS Proxy](#)

5. Click **Set up** and wait for the connection setup to succeed.

✔ **Connection setup successfully for RDS database kabayan-db and Lambda function myLambdaFunction**

[View details](#)

✕

[RDS](#) > Databases

Databases (1)

☒ Group resources

↻

Modify

Actions ▾

Restore from S3

Create database

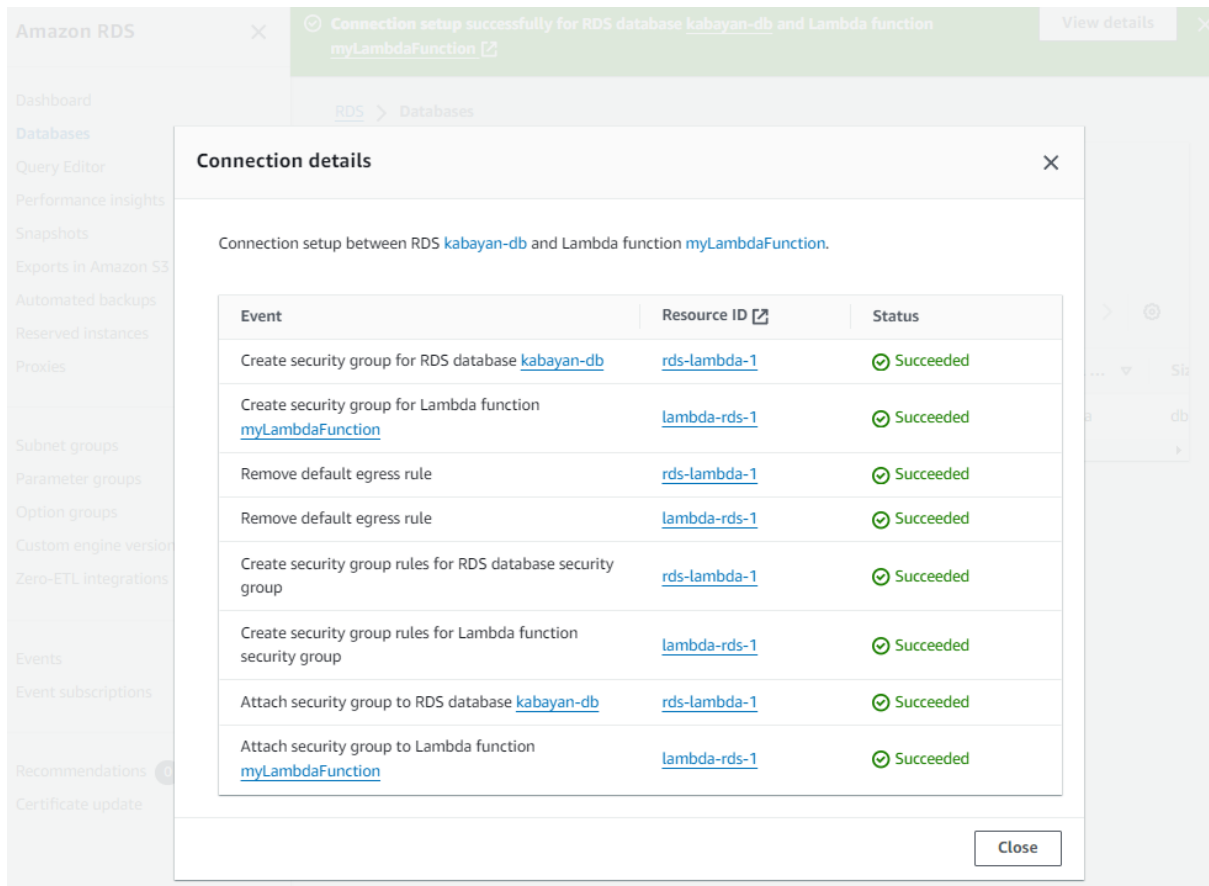
🔍 Filter by databases

< 1 > ⚙️

	DB identifier ▲	Status ▾	Role ▾	Engine ▾	Region & ... ▾	Size
<input type="radio"/>	<a href="#">kabayan-db</a>	✔ Available	Instance	MySQL Community	us-east-1a	db

- You can also view the details.



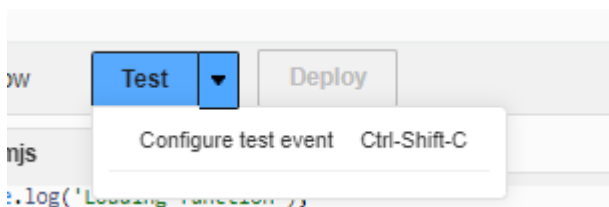


6. Navigate back to the Lambda Function Console and **wait for the Lambda Function to finish updating.**

*Take Note that the Lambda Function will also be updated*

### Test the Lambda Function

1. Navigate to the Code tab and click the arrow dropdown of the BLUE **Test** button



2. Click on **Configure test event**, and follow the configuration below:

- Event name: Test
- Template- optional: hello-world
  - Leave the rest as default

Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event

Edit saved event

Event name

Test

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

Event sharing settings

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

Format JSON

1 {

2 "key1": "value1",

3 "key2": "value2",

4 "key3": "value3"

5 }

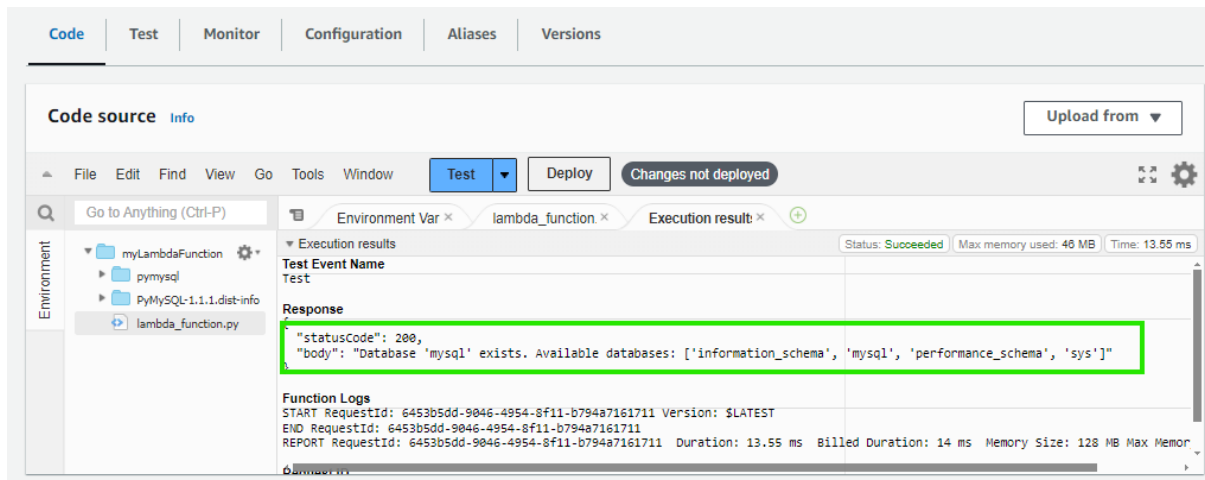
Cancel

Invoke

Save

- Click on Save

3. Now, click on **Test** and check the output to verify the successful execution of the SQL query.



That's it! Congratulations! You have gained hands-on experience in securely connecting an AWS Lambda function to an Amazon RDS database. You've learned how to manage database credentials using environment variables, ensuring that sensitive data like passwords are not hardcoded into your application code and enhancing security and maintainability.

This setup is crucial for building scalable, serverless applications that interact with relational databases, enabling use cases such as real-time data processing, API backends, and other data-driven applications. By understanding how to configure Lambda and RDS in a secure and efficient manner, you are now equipped to create dynamic, data-driven solutions in a serverless environment.