

Karthik Pradeep Hegadi

2KE20CS032

Assignment 48

Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

AWS

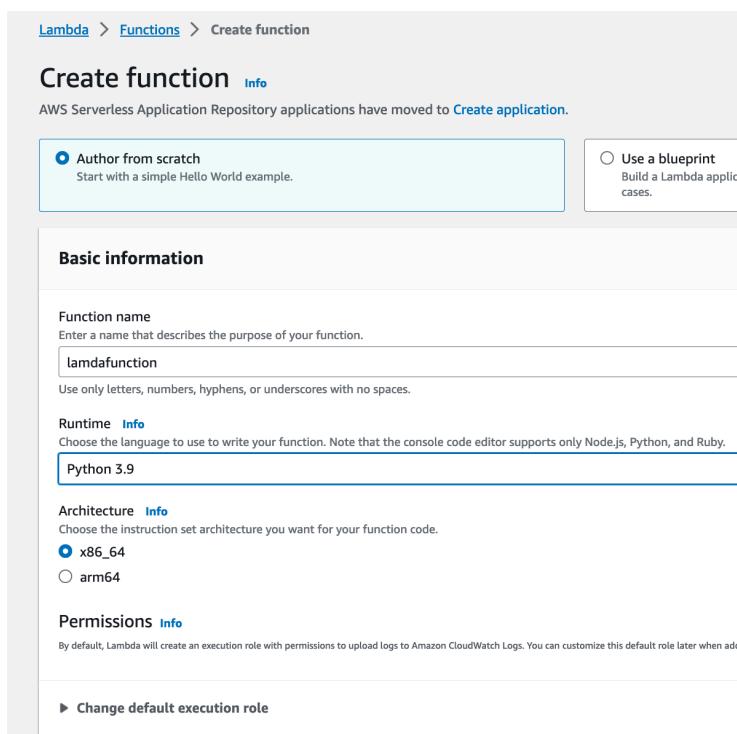
Assignment: 1 : Serverless computing using AWS LAMBDA

Overview:

- 1.Deploy code in AWS lambda
- 2.Create and invoke API for lambda function

Code deployment using AWS Lambda

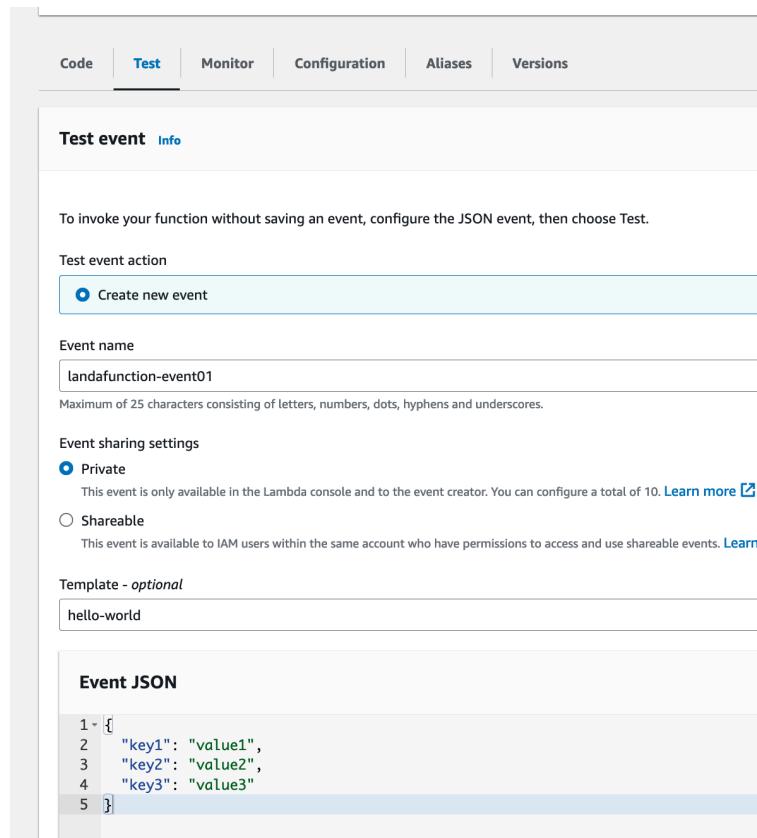
1. Navigate to Lambda page and click on create function
2. Provide the function name
- 3.Select the Runtime as python 3.9
- 4.Click on create function and it should be created successfully



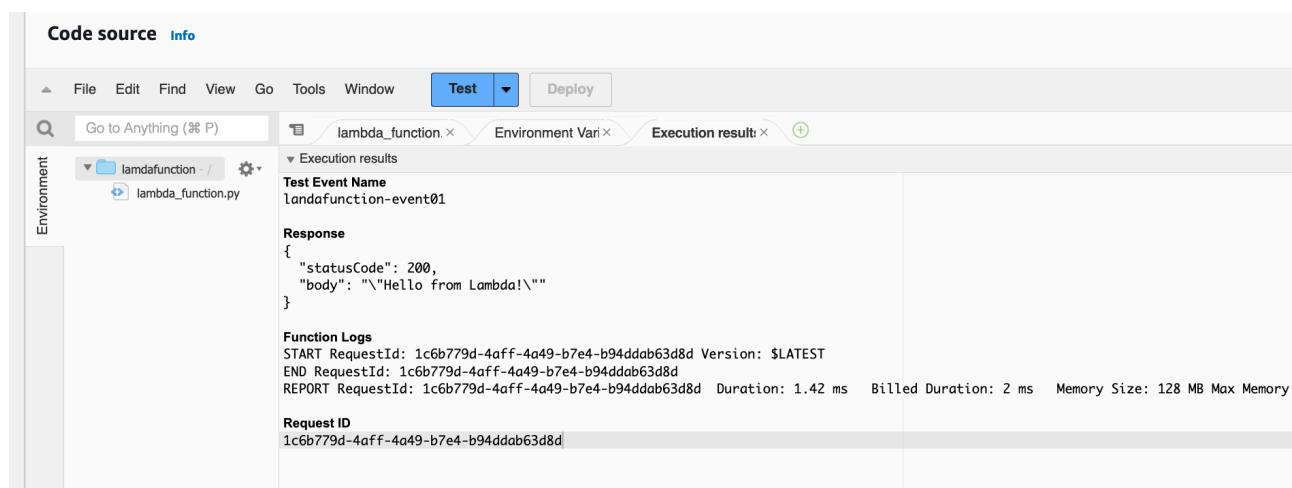
5. Scroll down and you can see a sample python code that you can use it for the deployment

6. Go to test tab to create a new event, provide the event name and click on save event and the event will get saved

7. Now navigate back to the code tab and click on test



8. You will get the execution results as below



Configure REST API to call the lambda function

1. Navigate to API gateway dashboard
2. Scroll down and select Build option on REST api in the API type
3. Click on New api, provide the api name, description and click on create API you will get a response as below



API Gateway > APIs > Create API > Create REST API

Create REST API

API details

New API
Create a new REST API.

Clone existing API
Create a copy of an API in this AWS account.

Import API
Import an API from an OpenAPI definition.

Example API
Learn about API Gateway with an example API.

API name
my-reset-api-01

Description - optional
this api is created for assignment of gradious 5

API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence.
Private APIs are only accessible from VPCs.

Regional

Cancel Create API

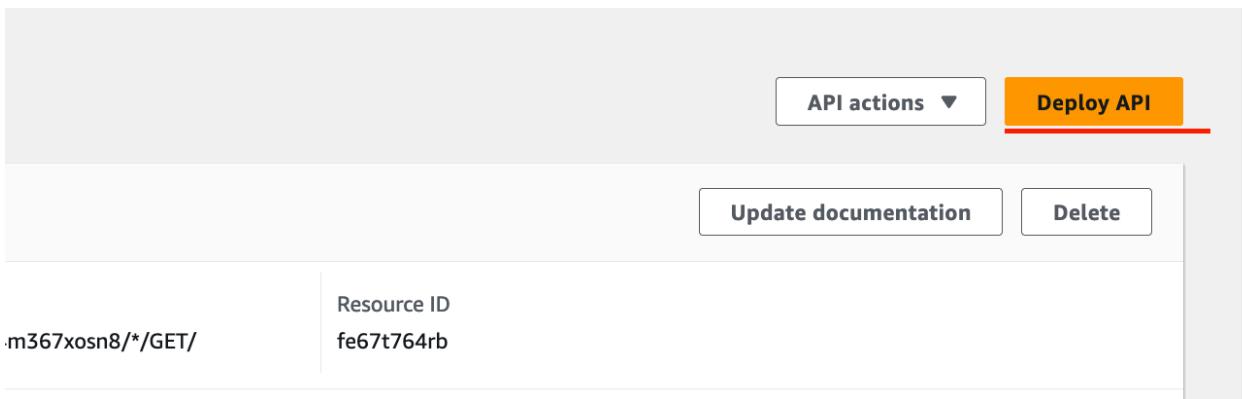
The screenshot shows the AWS API Gateway interface. At the top right, there are 'Delete' and 'Create method' buttons. Below them, there are dropdown menus for 'Integration type' (set to '/pe'), 'Authorization', and 'API key'. A main section titled 'No methods' displays the message 'No methods defined.'.

This screenshot shows the configuration for Lambda proxy integration. It includes a dropdown for 'Lambda function' set to 'ap-south-1', a search bar containing 'arn:aws:lambda:ap-south-1:405819896469:function:lan', and a note about granting API Gateway permission to invoke the Lambda function.

4. Navigate to Actions and click on 'Create Method' and select GET from the resource dropdown and click on tick mark

This screenshot shows the 'Create Method' dialog. Under 'Method type', 'GET' is selected. Under 'Integration type', 'Lambda function' is selected, with a note about integrating with a Lambda function. Other options shown are 'HTTP' and 'Mock'.

5. In the LAMBDA function text box type the lambda function that you have created before and leave remaining all settings as default
6. Click on Save and then click 'OK' in the prompt
7. Resource will be created and you will get a response as below
8. Click on Actions and click on Deploy API



9. Select the Deployment stage as New stage, provide the stage name, stage description, and deployment description and click on deploy

The screenshot shows the 'Stages' page under 'my-reset-api-01 (4m367xosn8)'. It lists a single stage named 'stage-01'. The 'Stage details' section shows the stage name as 'stage-01', the API cache status as 'Inactive', and the invoke URL as 'https://4m367xosn8.execute-api.ap-south-1.amazonaws.com/stage-01'. There is also an 'Active deployment' section.

10. You will get a response page as below

11. Now click on the Invoke ur and you will get the function invoked using REST API

Stage name	Rate Info
stage-01	-
API cache	Burst Info
⊖ Inactive	-
Invoke URL	
https://4m367xosn8.execute-api.ap-south-1.amazonaws.com/stage-01	
Active deployment m689m7 on December 31, 2023, 10:49 (UTC+05:30)	
Logs and tracing Info	
CloudWatch logs	Detailed metrics
⊖ Inactive	⊖ Inactive

← → ⌂ 🛡️ 🔒 https://4m367xosn8.execute-api.ap-south-1.amazonaws.com/stage-01

[JSON](#) [Raw Data](#) [Headers](#)

Save Copy Collapse All Expand All Filter JSON

```
statusCode: 200
body:      """Hello from Lambda!"""
```

Karthik Pradeep Hegadi

2KE20CS032

Assignment 49

Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

AWS

Assignment: 2 : Creating Lambda Function to access RDS Database

Creating RDS Database

1 Navigate to RDS service and Create a Database

2. Choose Standard create, Engine type MySQL, Leave Engine version as default, choose Templates as Free tier
3. In Settings Give a name for your Database or leave it as default, leave Master username as default, Set Master password for your Database

The screenshot shows the AWS RDS 'Choose a database creation method' step. It offers two options: 'Standard create' (selected) and 'Easy create'. The 'Standard create' option allows setting all configuration options, including availability, security, backups, and maintenance. The 'Easy create' option uses recommended best-practice configurations, with some options changeable after creation. Below this, the 'Engine options' section shows various engine types: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), MariaDB, PostgreSQL, and Oracle. Each engine has a corresponding icon: two stacked cylinders with asterisks for Aurora, a fish for MySQL, a seal for MariaDB, a spiral for PostgreSQL, and a classical Oracle logo.

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

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.

Free tier

Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

[Info](#)

Availability and durability

Deployment options [Info](#)

The deployment options below are limited to those supported by the engine you selected above.

- Single DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a single DB instance with no standby DB instances.

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.

database-1

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.

admin

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

Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.
[Learn more](#)

Auto generate a password

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

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm master password [Info](#)

4. Instance Configuration and Storage you can leave it as default

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](#)

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

ⓘ 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

5. In Connectivity Choose first option(Don't connect to an EC2 Compute Resource) and in VPC select your VPC, Public access No, VPC security groups, you can create a new security group

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.

my-vpc-01 (vpc-0e190ca43b317839f)

5 Subnets, 2 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

rds-security-group

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-2019 (default)

Expiry: Aug 22, 2024



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

► Additional configuration

6In Database Authentication choose first method

7Disable Monitoring and click on Create database, Your database will be created

The screenshot shows the 'Instance configuration' section of the AWS RDS console. Under 'DB instance class', the 'Burstable classes (includes t classes)' option is selected. A summary box indicates the instance has 2 vCPUs, 1 GiB RAM, and a Network speed of 2,085 Mbps.

The screenshot shows the 'Databases' page in the AWS RDS console. It lists one database named 'database-1' with the status 'Creating'. The page includes a 'Create database' button and various filtering and sorting options.

8.Go to the database and open your security group, In inbound rules, add a rule in securitygroup, choose the type as MYSQL/Aurora and allow your VPC range, In Outbound rule allow all traffic

The screenshot shows the 'Inbound rules' configuration page for a security group. It displays two rules: one allowing traffic from '157.50.67.84/32' and another allowing traffic from '10.0.0.0/8'. Both rules are of type 'MySQL/Aurora' on port 3306. The 'Add rule' button is visible at the bottom left.

Connect to RDS instance

1 Launch an ec2 instance and install mysql alone there follow mattermost software Installation Document

```
Verifying : mysql-community-common-8.0.35-1.el9.x86_64          3/6
Verifying : mysql-community-icu-data-files-8.0.35-1.el9.x86_64      4/6
Verifying : mysql-community-libs-8.0.35-1.el9.x86_64          5/6
Verifying : mysql-community-server-8.0.35-1.el9.x86_64          6/6

Installed:
  mysql-community-client-8.0.35-1.el9.x86_64    mysql-community-client-plugins-8.0.35-1.el9.x86_64
  mysql-community-common-8.0.35-1.el9.x86_64    mysql-community-icu-data-files-8.0.35-1.el9.x86_64
  mysql-community-libs-8.0.35-1.el9.x86_64     mysql-community-server-8.0.35-1.el9.x86_64

Complete!
```

```
ec2-user@ip-10-0-1-197.ap-south-1.compute.internal ~ (0.969s)
sudo systemctl status mysqld
● mysqld.service - MySQL Server
  Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
  Active: active (running) since Sun 2023-12-31 07:16:14 UTC; 4s ago
    Docs: man:mysqld(8)
          http://dev.mysql.com/doc/refman/en/using-systemd.html
  Main PID: 4915 (mysqld)
    Status: "Server is operational"
      Tasks: 38 (limit: 1114)
     Memory: 426.9M
        CPU: 5.869s
       CGroup: /system.slice/mysqld.service
```

2. Connect to mysql using the command as follows,
mysql -h your-rds-dnsname -u admin -p

```
mysql -h database-1.c5usk8ag8mpj.ap-south-1.rds.amazonaws.com -p 3306 -u admin -p
Enter password:
ERROR 1049 (42000): Unknown database '3306'

ec2-user@ip-10-0-1-197.ap-south-1.compute.internal ~
mysql -h database-1.c5usk8ag8mpj.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 47
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2023, 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>
```

3.For Creating Database, tables, Inserting Content into the tables, follow the attachedimages

```
mysql> create database simpleDB ;
Query OK, 1 row affected (0.01 sec)

mysql> use simpleDB;
Database changed
mysql> CREATE TABLE CUSTOMERS (
    ->     ID INT NOT NULL,
    ->     NAME VARCHAR(20) NOT NULL,
    ->     AGE INT NOT NULL,
    ->     ADDRESS CHAR(25),
    ->     SALARY DECIMAL(18, 2),
    ->     PRIMARY KEY (ID)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> 
```

```
    ->     NAME VARCHAR(20) NOT NULL,
    ->     AGE INT NOT NULL,
    ->     ADDRESS CHAR(25),
    ->     SALARY DECIMAL(18, 2),
    ->     PRIMARY KEY (ID)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_simpleDB |
+-----+
| CUSTOMERS          |
+-----+
1 row in set (0.01 sec)

mysql> 
```

```
| Tables_in_simpleDB |
+-----+
| CUSTOMERS      |
+-----+
1 row in set (0.01 sec)

mysql> INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY) VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY) VALUES (2, 'Khilan', 25, 'Delhi', 1500.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY) VALUES (3, 'Kaushik', 23, 'Kota', 2000.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY) VALUES (4, 'Chaitali', 25, 'Mumbai', 6500.00);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY) VALUES (5, 'Hardik', 27, 'Bhopal', 8500.00);
Query OK, 1 row affected (0.00 sec)

mysql> █
```

```
mysql> commit;
Query OK, 0 rows affected (0.01 sec)

mysql> select * form tables;
ERROR 1064 (42000): You have an error in your SQL syntax; che
e 1
mysql> select * form customers;
ERROR 1064 (42000): You have an error in your SQL syntax; che
line 1
mysql> select * form CUSTOMERS;
ERROR 1064 (42000): You have an error in your SQL syntax; che
line 1
mysql> select * from CUSTOMERS;
+----+-----+----+-----+----+
| ID | NAME      | AGE | ADDRESS    | SALARY |
+----+-----+----+-----+----+
| 1  | Ramesh    | 32 | Ahmedabad  | 2000.00 |
| 2  | Khilan    | 25 | Delhi       | 1500.00 |
| 3  | Kaushik   | 23 | Kota        | 2000.00 |
| 4  | Chaitali  | 25 | Mumbai      | 6500.00 |
| 5  | Hardik    | 27 | Bhopal     | 8500.00 |
+----+-----+----+-----+----+
5 rows in set (0.01 sec)

mysql> █
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> commit;
Query OK, 0 rows affected (0.01 sec)

mysql> █
```

Creating a Lambda Function

1. Navigate to Lambda Function and create a function and copy the function in the attached document and make all the necessary changes - **Lambda Function for RDS DB.pdf**

2. You need to have aws-sdk libraries, if you want to execute lambda functions, we can install the aws-sdk using nom, For that go to your centos machine in which you have

nodejs and pm, create one folder in your CentOS and navigate into the folder and execute nom install aws-sdk and nom install mysql , You will get node modules and fewfiles, Create a Zip file for that folders and files. Using filezilla, send that zip file to your windows.

3. Add layers to your lambda functions. Navigate to layers and click on create layer

4. Give a layer name and upload that zip file which you have created before

5. Choose the run time as Nodejs16

6. In the lambda Configuration choose your VPC, Subnet, Security groups Which will make your lambda available at your VPC level,

7. In the lambda Configuration, In Permissions, make sure lambda has correct roles to access your Database, If not create a role and attach it to the lambda

Reference video:

references

AWS - Lambda - Layer Using Node.js

https://www.youtube.com/watch?v=fIJ_cfQ53vQ&t=121s

Connect to MySQL on AWS RDS using NodeJS

<https://www.youtube.com/watch?v=6Nt-JI3CzxE&t=184s>

Lambda with AWS RDS Tutorial: Connecting to MySQL on Lambda using mysql-connector-python

<https://www.youtube.com/watch?v=D2OrhX4XkXQ>

Private RDS Instance & AWS Lambda

<https://www.youtube.com/watch?v=UgWjbSixRg4&t=474s>

Installing npm modules in aws

<https://www.youtube.com/watch?v=RnFowJ130pc&t=247s>

For both the method I have installed requested modules in host machine and transferred those files using FileZilla

Using python 3.10

Lambda > Functions > nodejs--python

nodejs--python

▼ Function overview [Info](#)

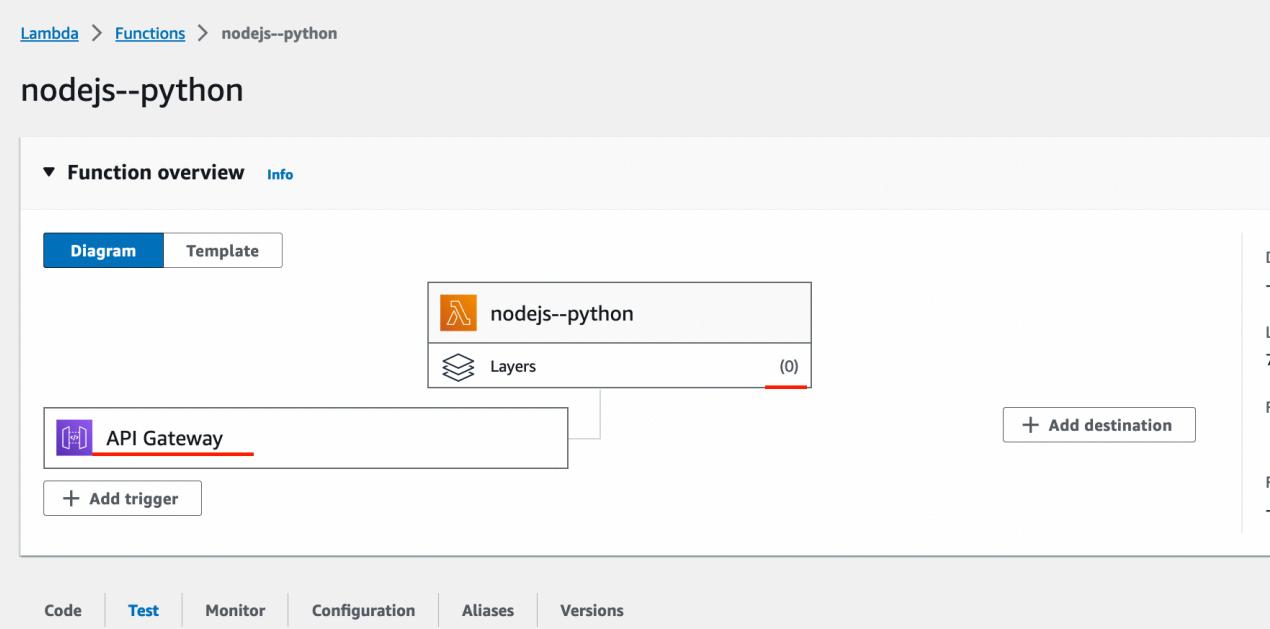
[Diagram](#) [Template](#)

 nodejs--python
 Layers (0)

 API Gateway [+ Add destination](#)

[+ Add trigger](#)

Code [Test](#) Monitor Configuration Aliases Versions



nodejs--python

 nodejs--python
 Layers (0)

 API Gateway [+ Add destination](#)

[+ Add trigger](#)

Code [Test](#) Monitor Configuration Aliases Versions

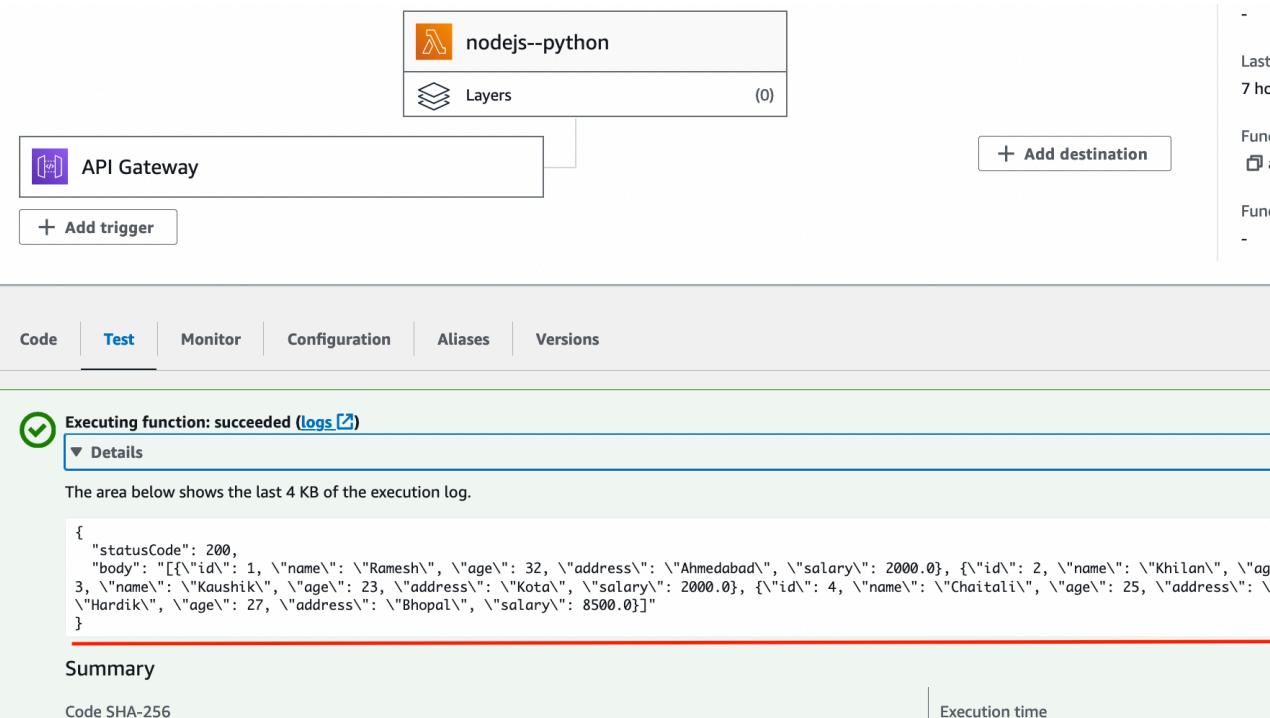
 Executing function: succeeded ([logs](#)) [Details](#)

The area below shows the last 4 KB of the execution log.

```
{ "statusCode": 200, "body": "[{"id": 1, "name": "Ramesh", "age": 32, "address": "Ahmedabad", "salary": 2000.0}, {"id": 2, "name": "Khilan", "age": 3, "name": "Kaushik", "age": 23, "address": "Kota", "salary": 2000.0}, {"id": 4, "name": "Chaitali", "age": 25, "address": "Hardik", "age": 27, "address": "Bhopal", "salary": 8500.0}]"
```

Summary

Code SHA-256: X91LDatjzPUTQAxWoAaPA2oq0eo7K1wP1e3EbJlmhZc= Execution time: 11 seconds ago (January 1, 2024 at 09:18 AM CDT)



Using nodejs 16x and nodejs 18x

lambdafunctionn

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

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

 lambdafunctionn
 Layers (0)

 API Gateway [+ Add destination](#)

[+ Add trigger](#)

even for node js is same but here i took s3 bucket to store the node_modules

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

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

▼ Details

The area below shows the last 4 KB of the execution log.

```
{ "statusCode": 200, "body": "[{"ID": 1, "NAME": "Ramesh", "AGE": 32, "ADDRESS": "Ahmedabad", "SALARY": 2000.0}, {"ID": 2, "NAME": "Khilan", "AGE": 25, "NAME": "Kaushik", "AGE": 23, "ADDRESS": "Kota", "SALARY": 2000.0}, {"ID": 4, "NAME": "Chaitali", "AGE": 25, "ADDRESS": "Mumbai", "NAME": "Hardik", "AGE": 27, "ADDRESS": "Bhopal", "SALARY": 8500.0}]"
```

Summary

Code SHA-256

Execution time