



AWS : Serverless BUILD SERVERLESS APPLICATION ON AWS CLOUD

Presented by :
Amrit Choudhary

RDS

POSTGRESQL



PostgreSQL has become the preferred open source relational database for many enterprise developers and start-ups, powering leading business and mobile applications. Amazon RDS makes it easy to set up, operate, and scale PostgreSQL deployments in the cloud. With Amazon RDS, you can deploy scalable PostgreSQL deployments in minutes with cost-efficient and resizable hardware capacity.

Repository to deploy:

<https://github.com/trainmefordevsecops/serverless-project/tree/master/infrastructure/lambda-project/rds>

commands:

```
git clone git@github.com:trainmefordevsecops/serverless-project.git
cd serverless-project/infrastructure/lambda-project/rds
```

```
terraform13 init
```

```
terraform13 plan
```

```
terraform13 apply
```

**** need to update vpc and subnet id's ****

```
(base) amrits-MacBook-Pro:rds ihealth$ ls -ltr
total 40
-rw-r--r--@ 1 ihealth  staff   190 Aug  8  2021 backend.tf
-rw-r--r--@ 1 ihealth  staff    90 Feb  7  2022 provider.tf
-rw-r--r--@ 1 ihealth  staff 2235 Sep 25 16:26 main.tf
-rw-r--r-- 1 ihealth  staff  167 Sep 25 16:50 README.txt
-rw-r--r--@ 1 ihealth  staff 2295 Sep 26 22:27 vars.tf
(base) amrits-MacBook-Pro:rds ihealth$ pwd
/Users/ihealth/Desktop/assignment/new/serverless-project/infrastructure/lambda-project/rds
(base) amrits-MacBook-Pro:rds ihealth$
```

Lambda :

STEPS)

1) Create s3 bucket = mydb-query

This is the placeholder for csv file which is the trigger for lambda to execute Postgresql query

2) Lambda function

Executes the python code to query postgresql, sends logs to cloudwatch

This lambda can be used to process any batch operation in serverless mode.

Amazon S3 > Buckets > mydb-query

mydb-query [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)



Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	test.csv	csv	April 23, 2022, 18:07:28 (UTC+08:00)	17.0 B	Standard


query

Throttle Copy ARN Actions

Function overview [Info](#)

 query

 Layers (0)

 S3

[Add trigger](#)

[Add destination](#)

Description

-

Last modified
1 minute ago

Function ARN
[arn:aws:lambda:ap-south-1:875132019252:function:query](#)

Function URL [Info](#)

-

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

Execution result: succeeded ([logs](#))

Details

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

☐

Summary

Code SHA-256
19c7u1RS0S3DBba3ADUUpzYz25GxLz2bJQSiQjTO2rY-

Init duration
743.65 ms

Billed duration
26 ms

Max memory used
68 MB

Log output

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

START RequestId: 2242875b-a087-4a79-b93e-f077b987625a Version: \$LATEST

monish

delete from app_user;

END RequestId: 2242875b-a087-4a79-b93e-f077b987625a

REPORT RequestId: 2242875b-a087-4a79-b93e-f077b987625a Duration: 25.58 ms Billed Duration: 26 ms Memory Size: 1024 MB Max Memory Used: 68 MB Init Duration: 743.65 ms

Request ID
2242875b-a087-4a79-b93e-f077b987625a

Duration
25.58 ms

Resources configured
1024 MB


Lambda :



STEPS)

3) cloudwatch : Store logs

CloudWatch > Log groups > /aws/lambda/query > 2022/04/23/[\$LATEST]1d31cb530a8e481bbb8448b6a84ed927

Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

☐ View as text  **Actions** ▼

Clear 1m 30m 1h 12h Custom  

▶	Timestamp	Message
		No older events at this moment. Retry
▶	2022-04-23T18:41:57.592+08:00	START RequestId: 2242875b-a087-4a79-b93e-f077b987625a Version: \$LATEST
▶	2022-04-23T18:41:57.592+08:00	1
▶	2022-04-23T18:41:57.592+08:00	manish
▶	2022-04-23T18:41:57.594+08:00	delete from app_user;
▶	2022-04-23T18:41:57.618+08:00	□
▶	2022-04-23T18:41:57.622+08:00	END RequestId: 2242875b-a087-4a79-b93e-f077b987625a
▶	2022-04-23T18:41:57.622+08:00	REPORT RequestId: 2242875b-a087-4a79-b93e-f077b987625a Duration: 25.58 ms Billed Duration: 26 ms Memory Size: 1024 MB Max Memory Used: 68 ...
		No newer events at this moment. Auto retry paused. Resume

Lambda :

STEPS) Execution Sequence

- 1) Terraform-fargate-example part should be executed which will create base, i.e vpc , subnet , route etc.
- 2) rds part need to be executed to created the postgresql first
Needs input from terraform-fargate-example provide to fetch vpc and subnet details and replace in vars.tf
run terraform13 init && terraform13 apply
- 3) Need to move to directory serverless-project/serverless-application/lambda-db-query-src and create lambda.zip

```
(base) amrits-MacBook-Pro:lambda-db-query-src ihealth$ ls -ltr
total 32
drwxr-xr-x  19 ihealth  staff   608 Feb 20  2022 boto3
drwxr-xr-x  78 ihealth  staff  2496 Feb 20  2022 botocore
-rw-r--r--   1 ihealth  staff  1059 Feb 20  2022 db_util.py.o
-rw-r--r--   1 ihealth  staff   469 Feb 20  2022 lambda.py
drwxr-xr-x  17 ihealth  staff   544 Feb 20  2022 psycopg2
-rw-r--r--@  1 ihealth  staff    17 Feb 20  2022 test.csv
drwxr-xr-x  27 ihealth  staff   864 Feb 20  2022 urllib3
-rw-r--r--   1 ihealth  staff  1964 Sep 26 23:40 db_util.py
(base) amrits-MacBook-Pro:lambda-db-query-src ihealth$ zip -r lambda psycopg2 lambda.py db_util.py boto3 botocore urllib3
  adding: psycopg2/ (stored 0%)
  adding: psycopg2/_json.py (deflated 64%)
```

zip -r lambda psycopg2 lambda.py db_util.py boto3 botocore urllib3

- 4) Copy this zip to cp -f lambda.zip ../../infrastructure/lambda-project/lambda-db-query/files/
- 5)

Lambda :

STEPS) Execution Sequence



5) Replace db_host, db_query, vpc, s3_resources, s3_bucket, bucket values in vars.tf (min)

6) Run below commands for creation of lambda with required resources

```
terraform13 init
```

```
terraform13 plan
```

```
terraform13 apply
```

Testing:

Upload a csv file with requires values, trigger will be generated and updated in postgresl db.

Else

Can simply click on lambda test to run it manually and test .