

# 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 \*\*

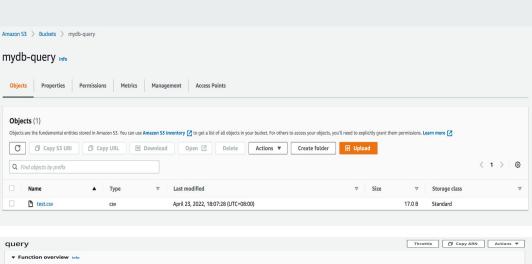
```
[(base) amrits-MacBook-Pro:rds ihealth$ ls -ltr
total 40
-rw-r--r--0 1 ihealth staff    190 Aug  8    2021 backend.tf
-rw-r--r--0 1 ihealth staff    90 Feb  7    2022 provider.tf
-rw-r--r--0 1 ihealth staff    2235 Sep    25 16:26 main.tf
-rw-r--r--    1 ihealth staff    167 Sep    25 16:50 README.txt
-rw-r--r--0 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$
```

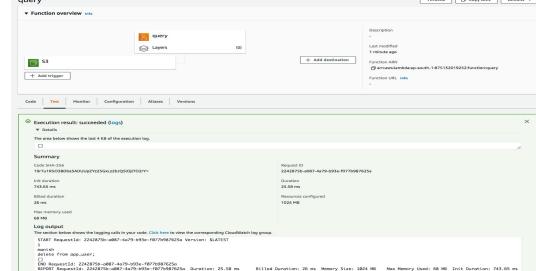
STEPS)

- 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

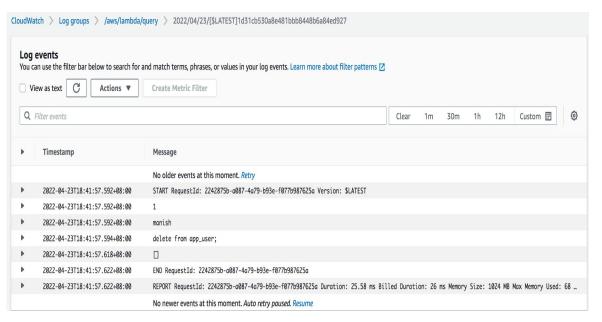
This lambda can be used to process any batch operation in serverless mode.





#### STEPS)

# 3) cloudwatch : Store logs



#### **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/_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)

#### 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.