**Java app deployment using Lambda:**

**RDS Database Creation:**

First create a Database in Aws (eg. Mysql).

**Steps:**

* Go go RDS service in aws console & select database
* Choose template type (production, dev) according to requirement & give database name and set the password for db.
* Select DB instance class and storage.
* Select availability zone and & vpc & subnet group to deploy rds.
* And click on create Database.
* Open the Database security group and add port 3306.

Integrate the RDS Database with Java application.

* Create lambda function, with appropriate ‘Handler’ & select the run time application.
* Upload the code to s3 bucket, and import the code to lambda.
* Create an event and test the function.

**APG gateway creation:**

* Create an API gateway 🡪 Create resource with name and enable API gateway cors 🡪 create resource 🡪 create method and integrate target lambda function 🡪 deploy API and test the Invoke URL by accessing it.

**Nodejs app deployment using Lambda:**

**Steps:**

* Go go RDS service in aws console & select database
* Choose template type (production, dev) according to requirement & give database name and set the password for db.
* Select DB instance class and storage.
* Select availability zone and & vpc & subnet group to deploy rds.
* And click on create Database.
* Open the Database security group and add port 3306.

**IAM Role:**

* You need to create a role for Lambda. As a best practice, define policies that follow the principle of granting least privilege.

1. AmazonRDSFullAccess
2. AmazonVPCFullAccess
3. AWSLambdaVPCAccessExecutionRole

**Create Lambda Function:**

* To create a Lambda function, go to AWS Console and select Lambda.Upload the code to s3 bucket, and import the code to lambda.
* Click create function and choose Node.js 12.x. Then, choose existing role we created above.
* In order to support a MySQL dependency, we need to create code first on your local computer and then upload it to the Lambda console.

#npm init

* Create a folder and create package.json
* Once the file is created, you can find the following JSON in the folder.

{

"name":"test",

"version":"1.0.0",

"description":"test",

"main":"index.js",

"scripts":{

"test":"echo \"Error: no test specified\" && exit 1"

},

"author":"Vaquar khan",

"license":"ISC"

}

}

* Create an event and test the function.
* You need to add MYSQL node dependency in project using following command

# npm install --save mysql

* Now, create a zip file and upload it.
* Now, add a handler into the app, create index.js (index.handler), and add Lambda logic.
* Now, add a handler into the app, create index.js (index.handler), and add Lambda logic.
* Create a lambda env variable with its key-value pairs.

RDS\_HOSTNAME

RDS\_PORT

RDS\_PASSWORD

RDS\_USERNAME

* Now, configure Lambda with RDS and VPC, go to the Security group of the RDS instance.
* Configure VPC inside the Lambda function network to Lambda function work. First, go to the Lambda function, click on Network, and then select the VPC and copy the IP address. If no VPC is selected, select a VPC that's the as the DB function and copy the IP address. Add this IP in RDS inbound settings.
* Save it and test your Lambda function.

**APG gateway creation:**

* Create an API gateway 🡪 Create resource with name and enable API gateway cors 🡪 create resource 🡪 create method and integrate target lambda function 🡪 deploy API and test the Invoke URL by accessing it.



