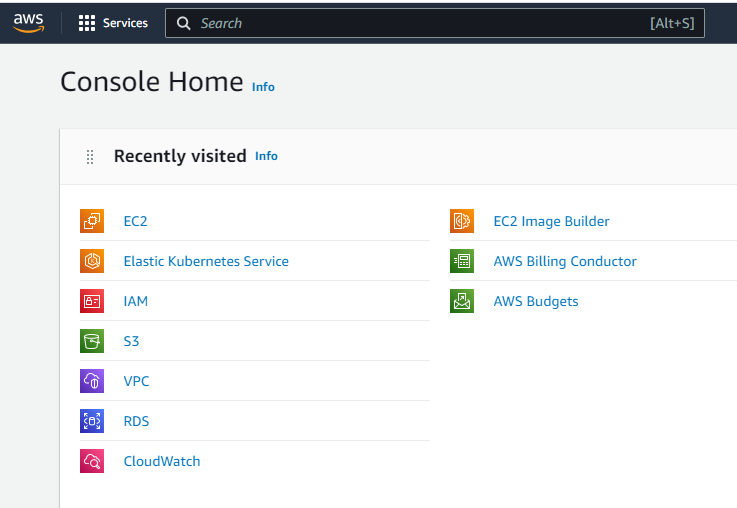
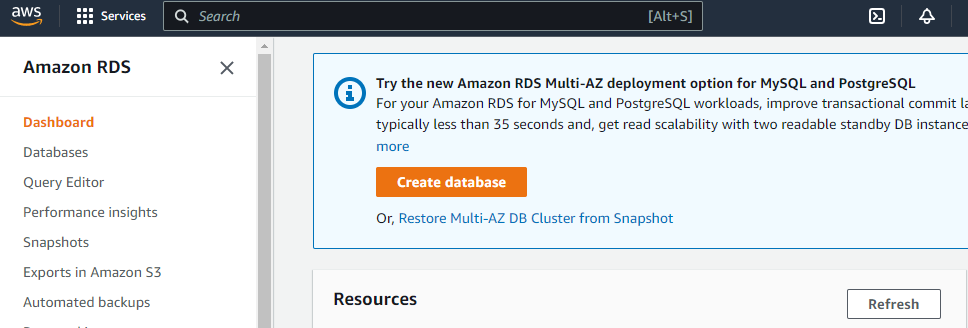
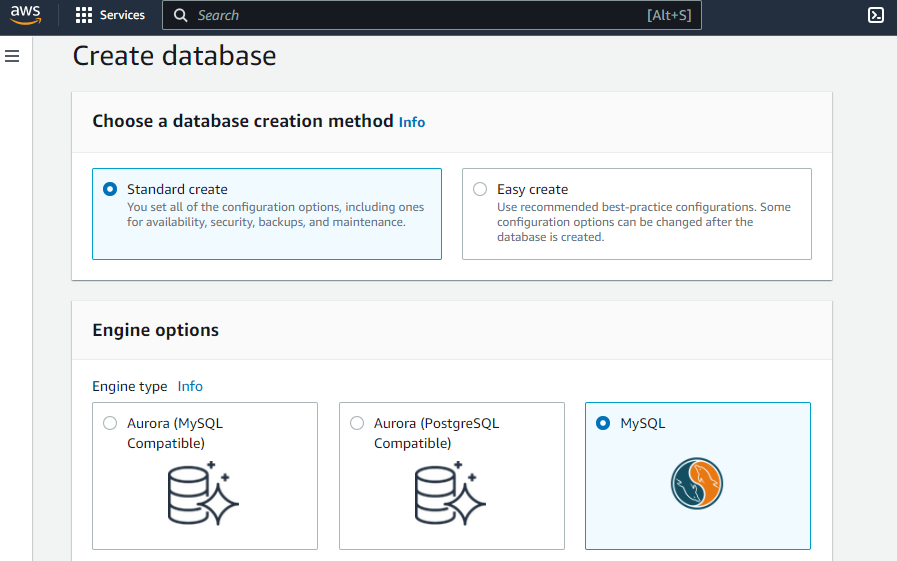
1. Spring Boot RESTful Microservice Deploy on AWS as well as use RDS (MYSQL)& DynamoDB Demo
2. Setup MYSQL database in AWS cloud using AWS RDS
   1. First go to AWS Console and then create database by using AWS RDS Service



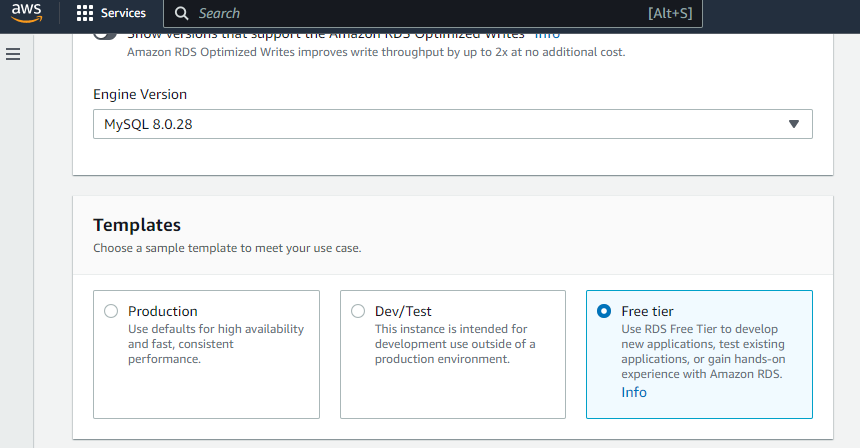
* 1. Now in RDS management console choose “Create Database”



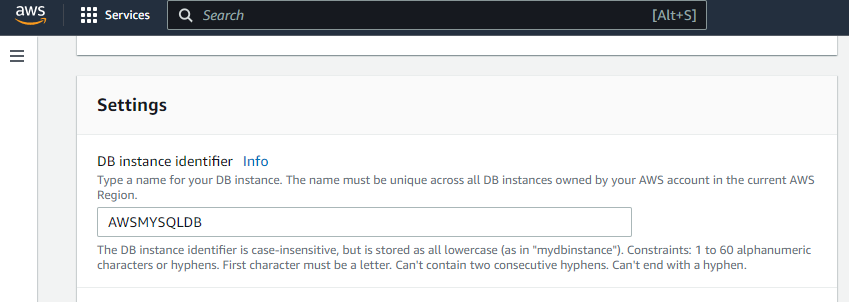
* 1. After that select standard create option and choose MYSQL database



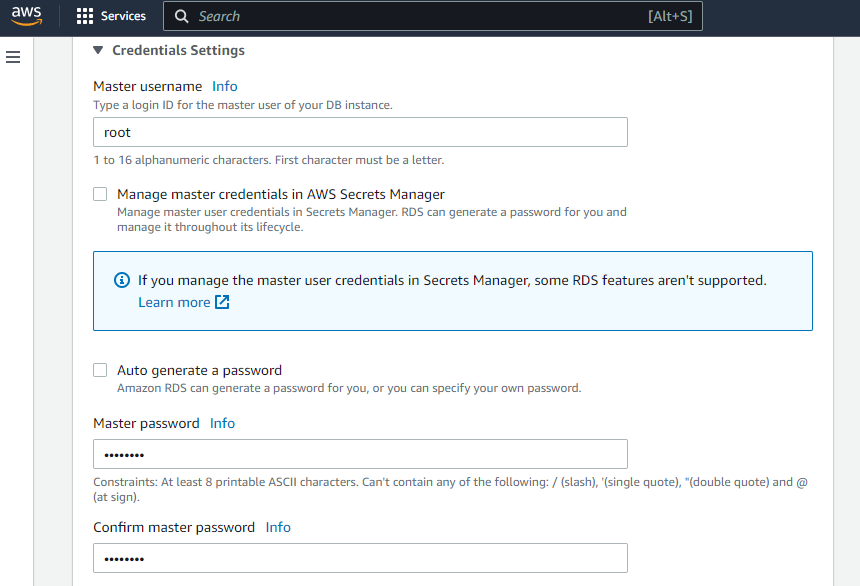
* 1. After that choose default version and then choose free tier for learning purpose



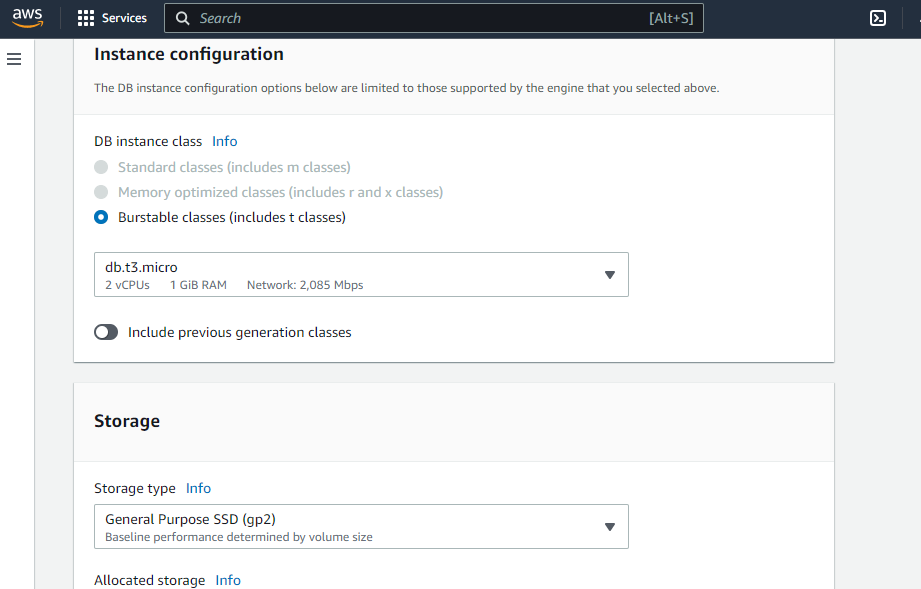
* 1. Then in settings choose database instance identifier name, you can give any name like “AWSMYSQLDB”



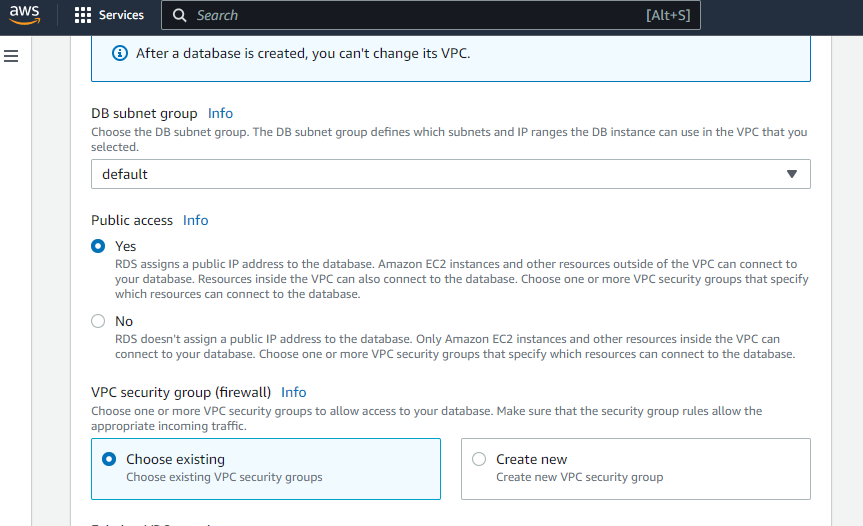
* 1. After that provide DB username and password
     1. UserName: “root”
     2. Password: “root1234”



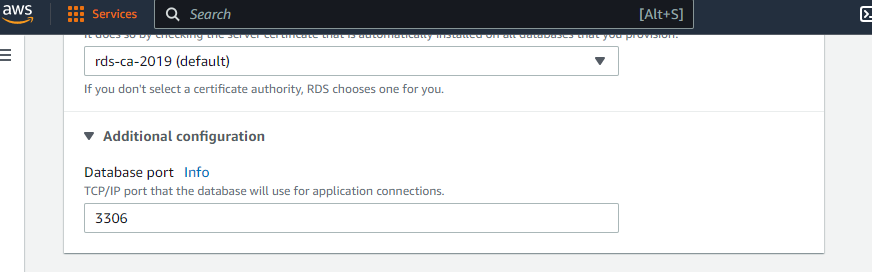
* 1. After that keep the default settings and give the public access to the MYSQL DB



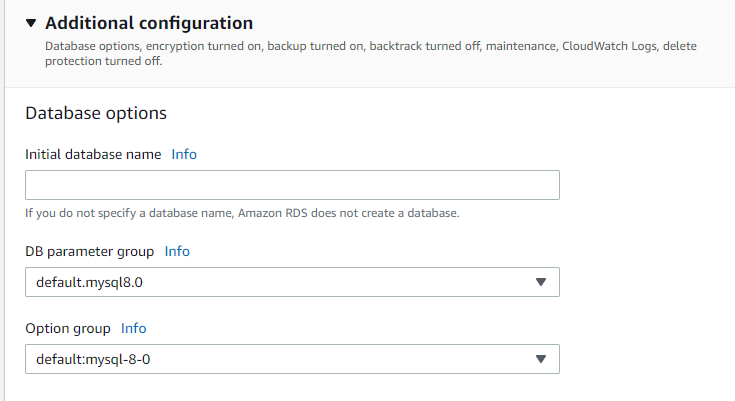
* 1. Give public access



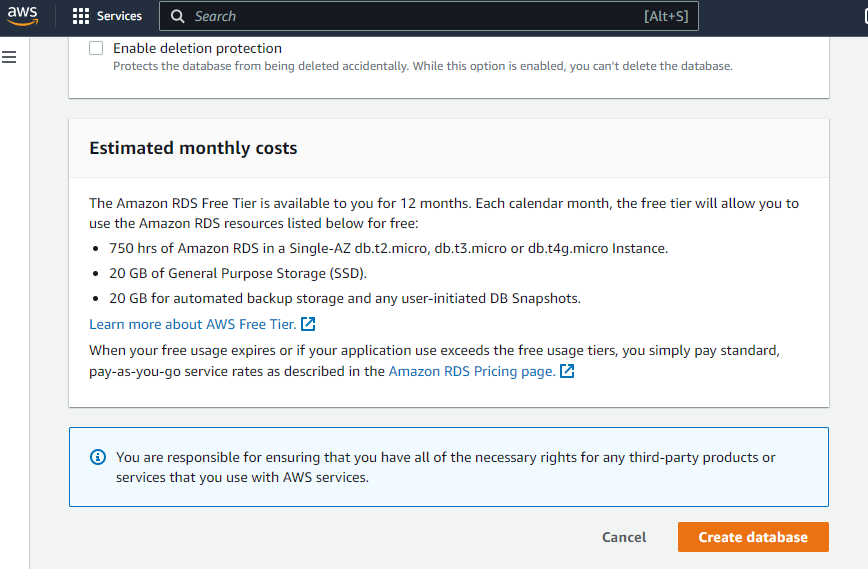
* 1. Keep the database port as it is which is mentioned inside the Additional Configuration section “3306”



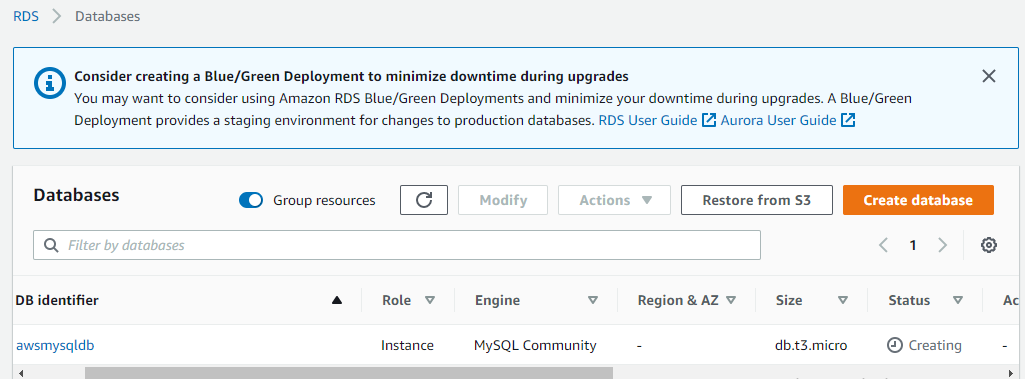
* 1. Now click on additional configuration to here you can specify the default DB name or you can also specify the db name after creating the instance by connecting through MYSQL workbench. I am keeping the database option blank. I will create the DB after connecting to DB through MYSQL workbench.

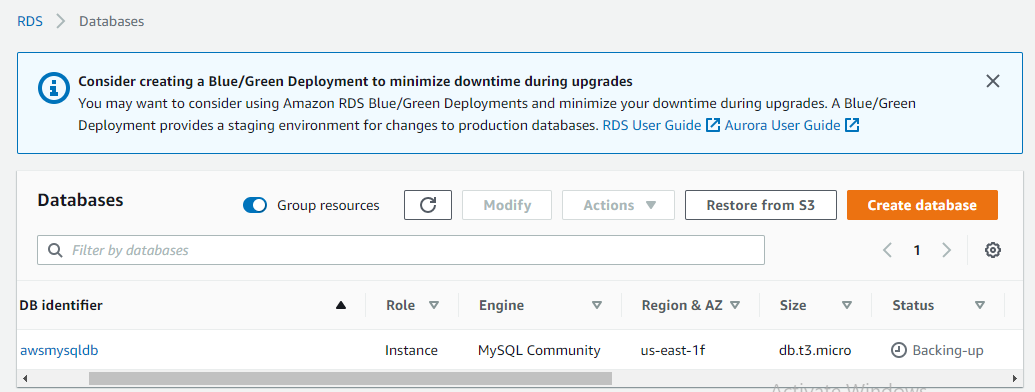


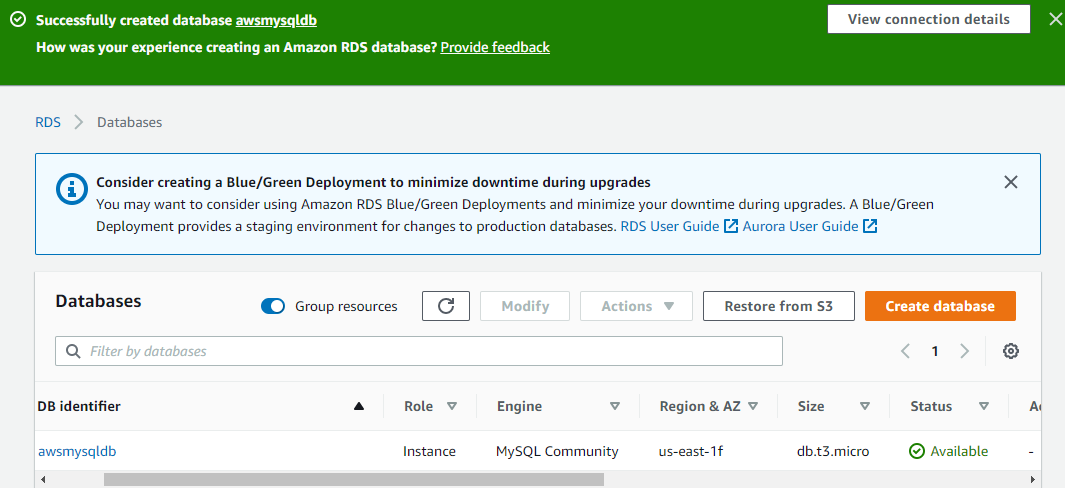
* 1. Now create all other things default and click the option create database.



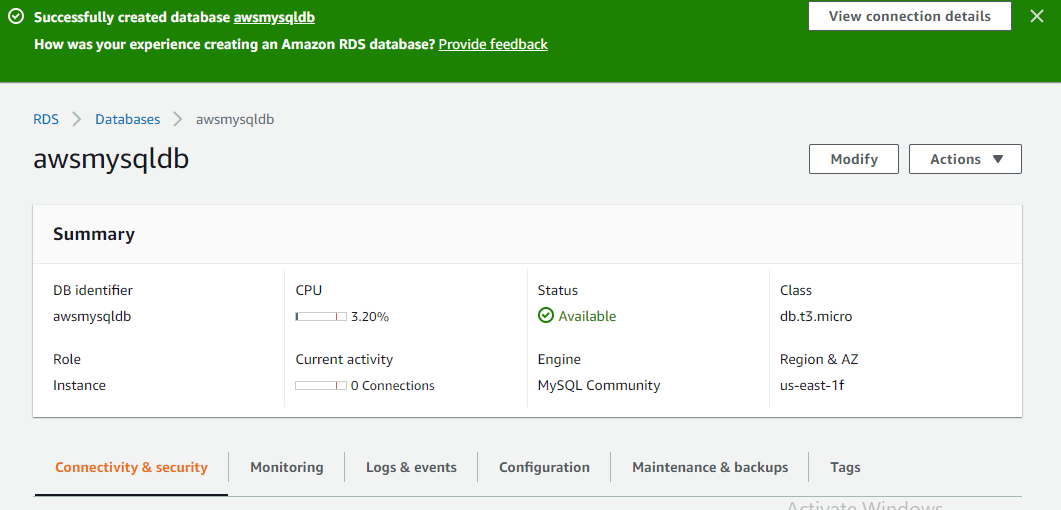
* 1. Initially the status will be “Creating” and after some time status will change to “BackingUp” and then “Available” if everything goes well.



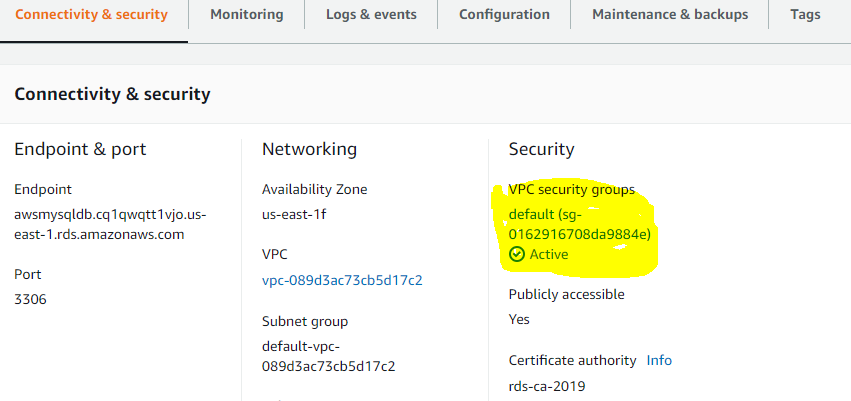




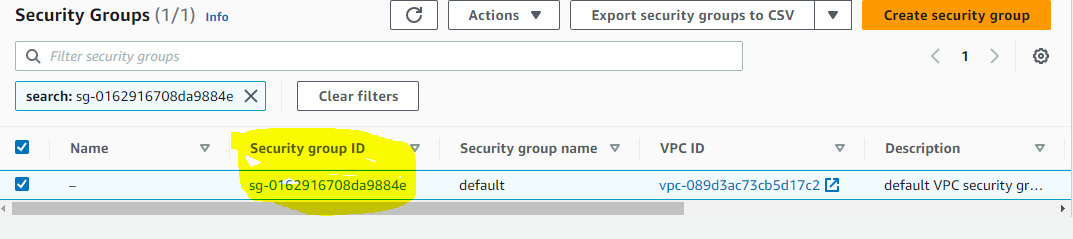
* 1. After this select the database instance by clicking the instance name and then change the security group so that it can be used from anywhere.



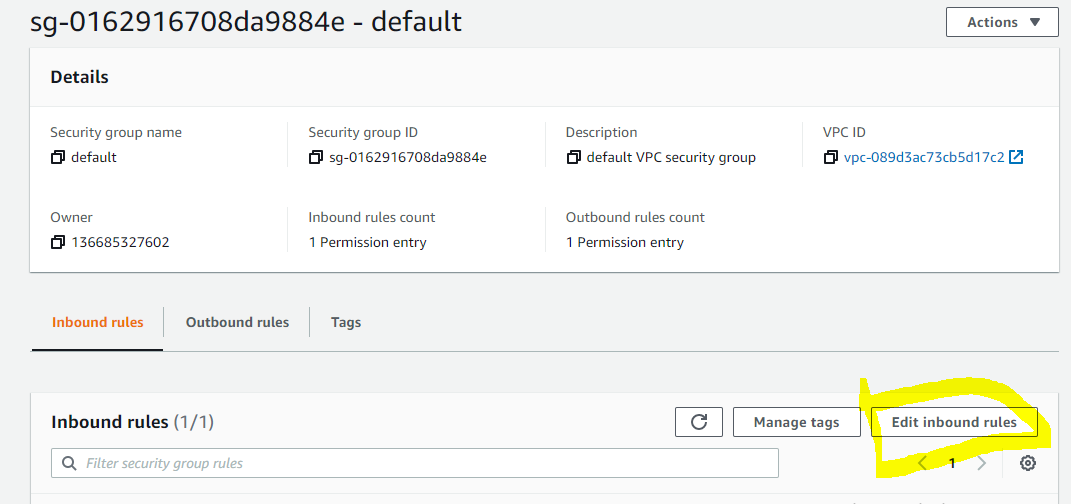
* 1. Now under the Connectivity & Security option go to security section and click the default link



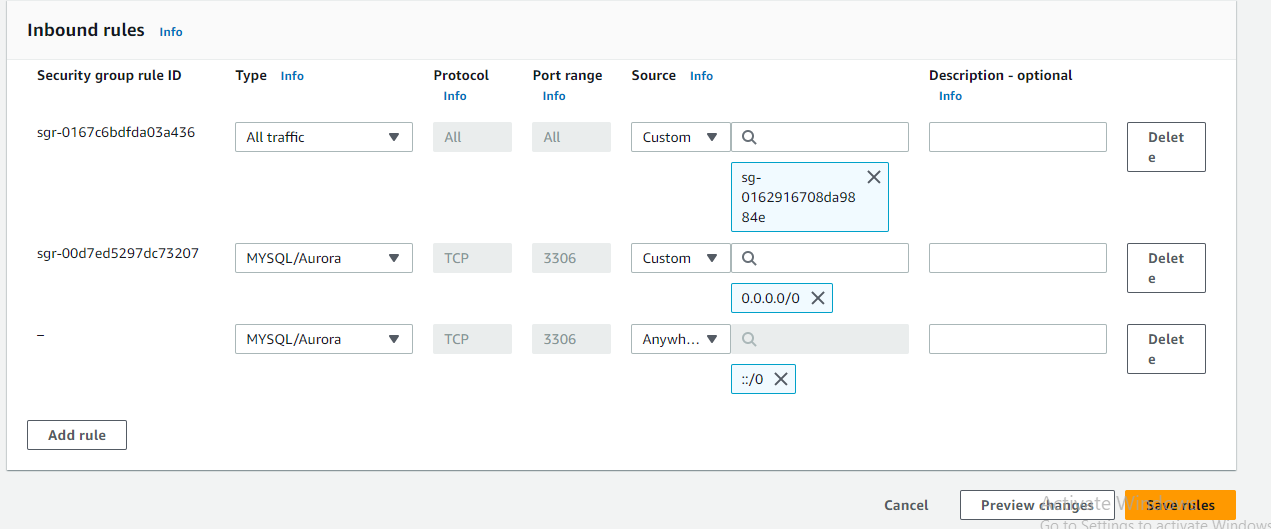
* 1. Now click the security group



* 1. And after that click Edit Inbound rule

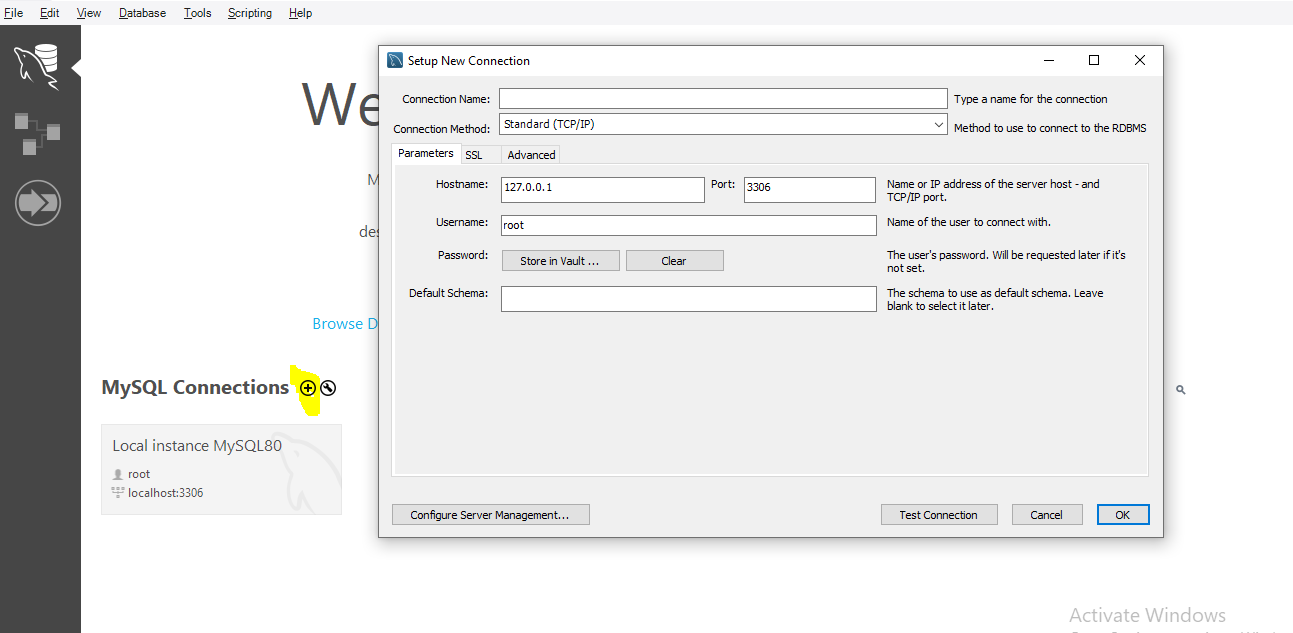


* 1. Now
     1. Click “Add Rule”
     2. Select “MYSQL Aurora”
     3. Choose “AnyWhere IPV4”
     4. Click On Save button
     5. Also repeat this for IPV6 as well

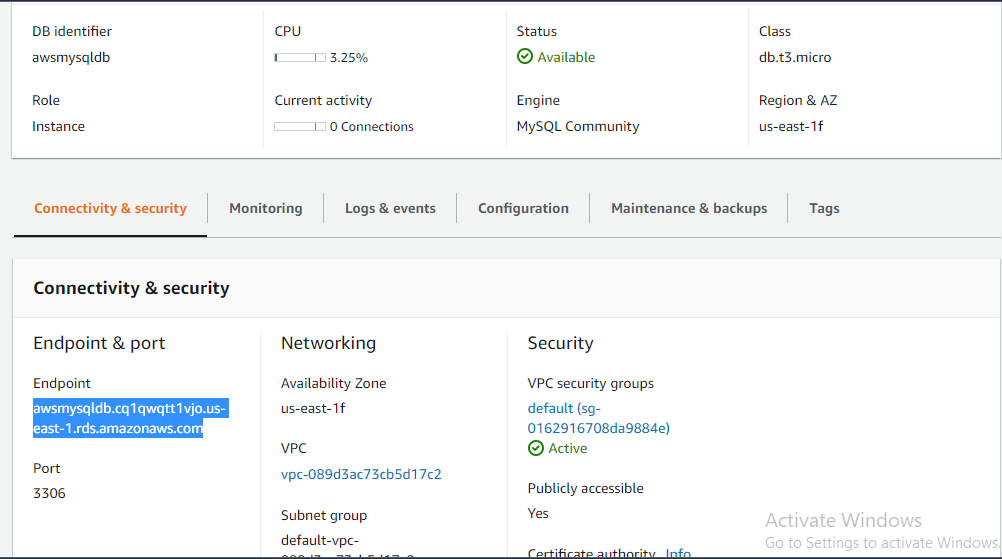


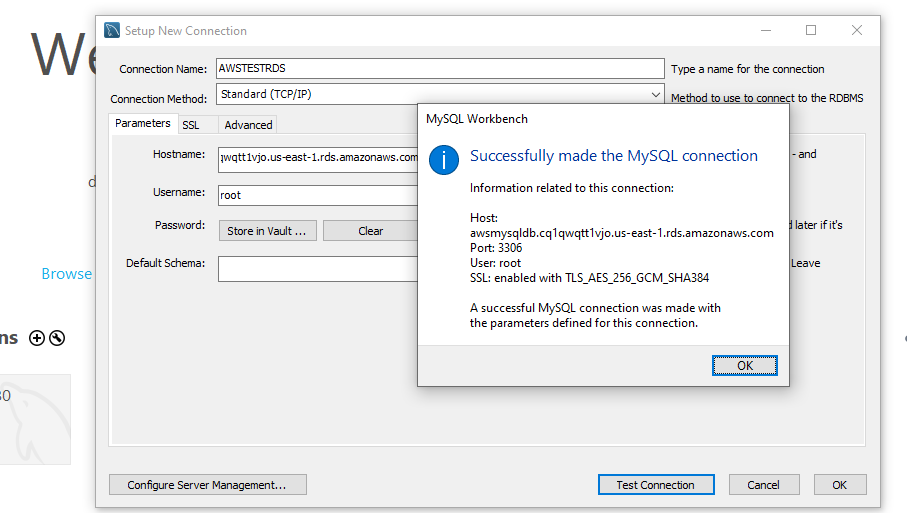
* 1. We have successfully configured RDS (MYSQL) inside AWS Cloud. Now we will connect with this DB instance by using MYSQL workbench and then we will perform DB operations.

1. Connect to MYSQL Database with MYSQL workbench to verify the connectivity
   1. Go to MYSQL workbench and then click the plus icon along with MYSQL Connections options

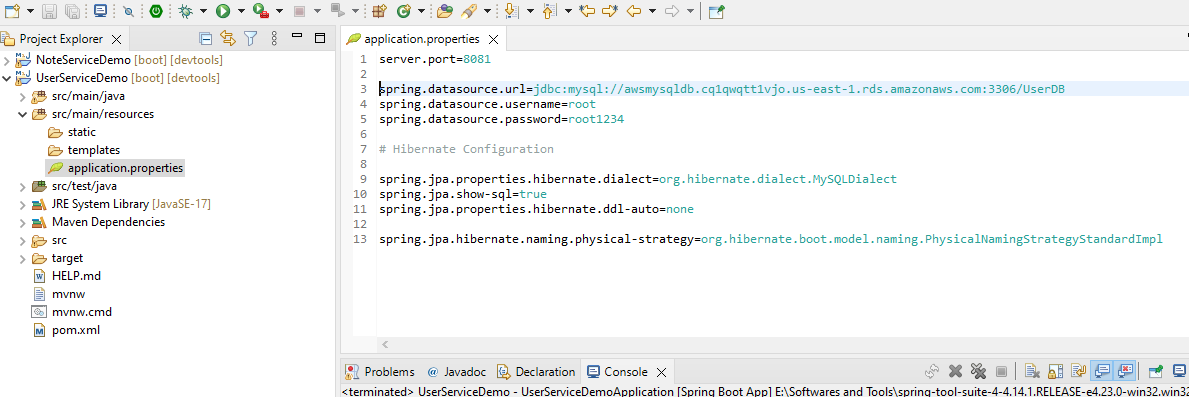


* 1. Now enter the connection details and establish the connectivity
     1. Connection Name : “AWSTestRDS”
     2. Host Name: “awsmysqldb.cq1qwqtt1vjo.us-east-1.rds.amazonaws.com” It is the endpoint name which you can get from AWS DB instance connectivity section.
     3. Password: “root1234”
     4. Now click on Test Connection

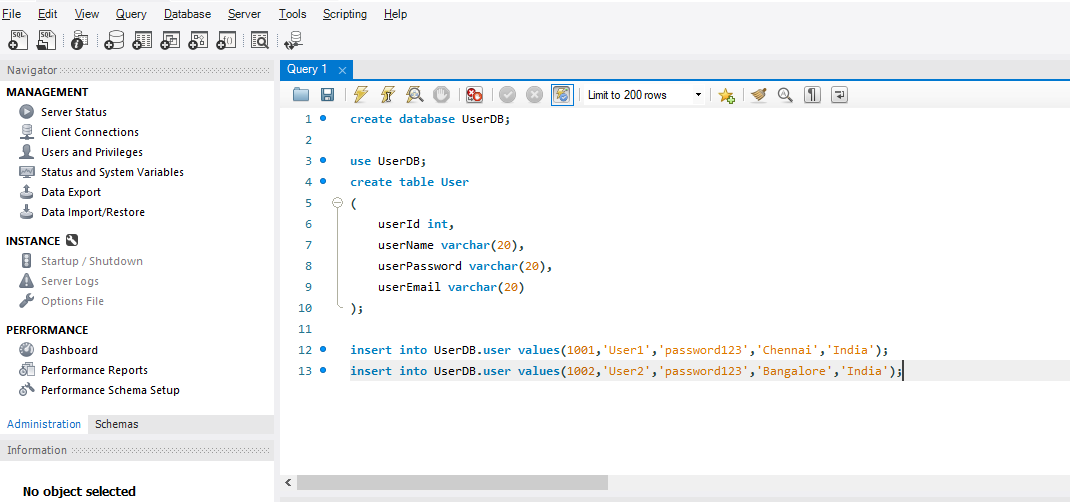




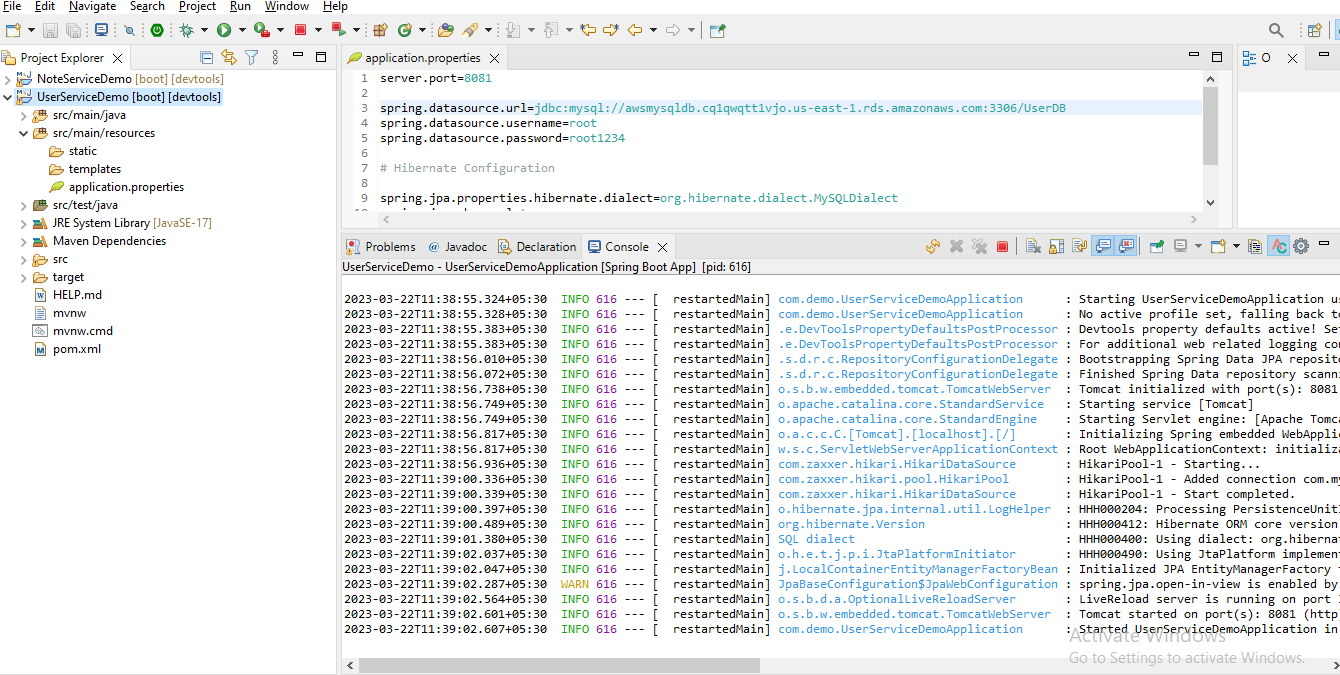
* 1. Now you can connect with database by clicking ok and implement Database queries.
  2. Now go to spring boot application.properties file and change the database connectivity information.
     1. Replace the localhost with the endpoint URL provided by MYSQL instance created inside the AWS cloud.

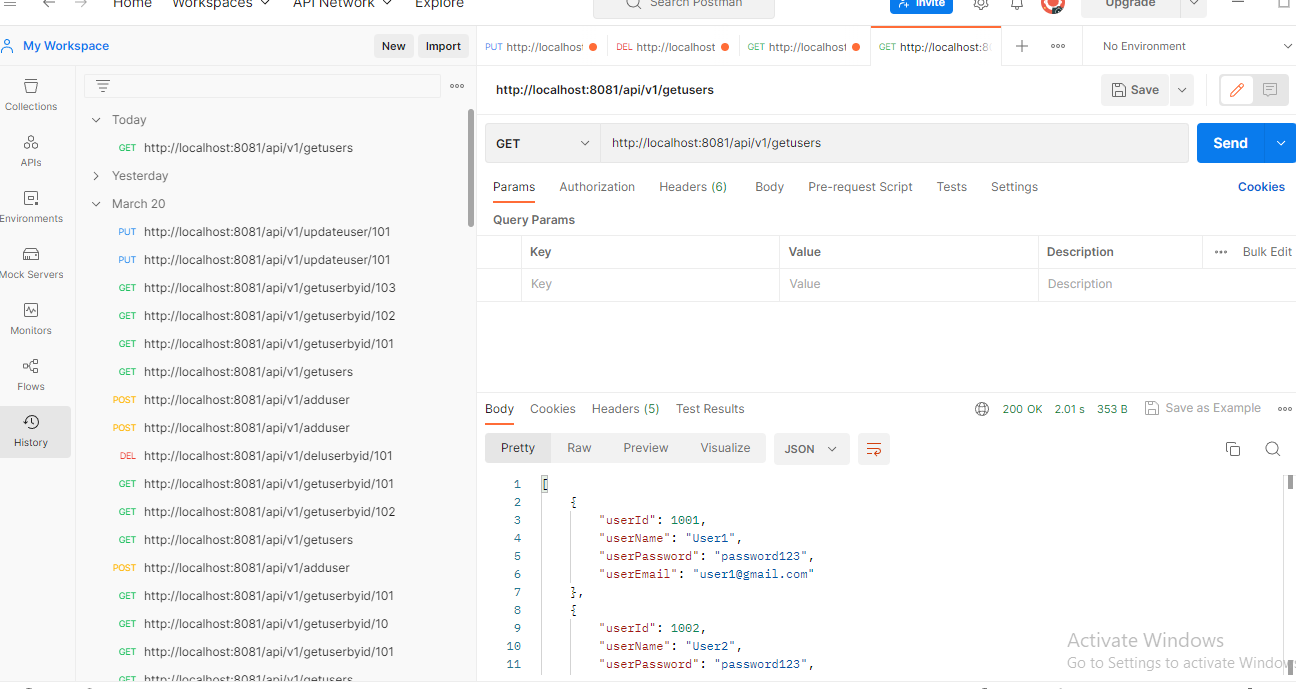


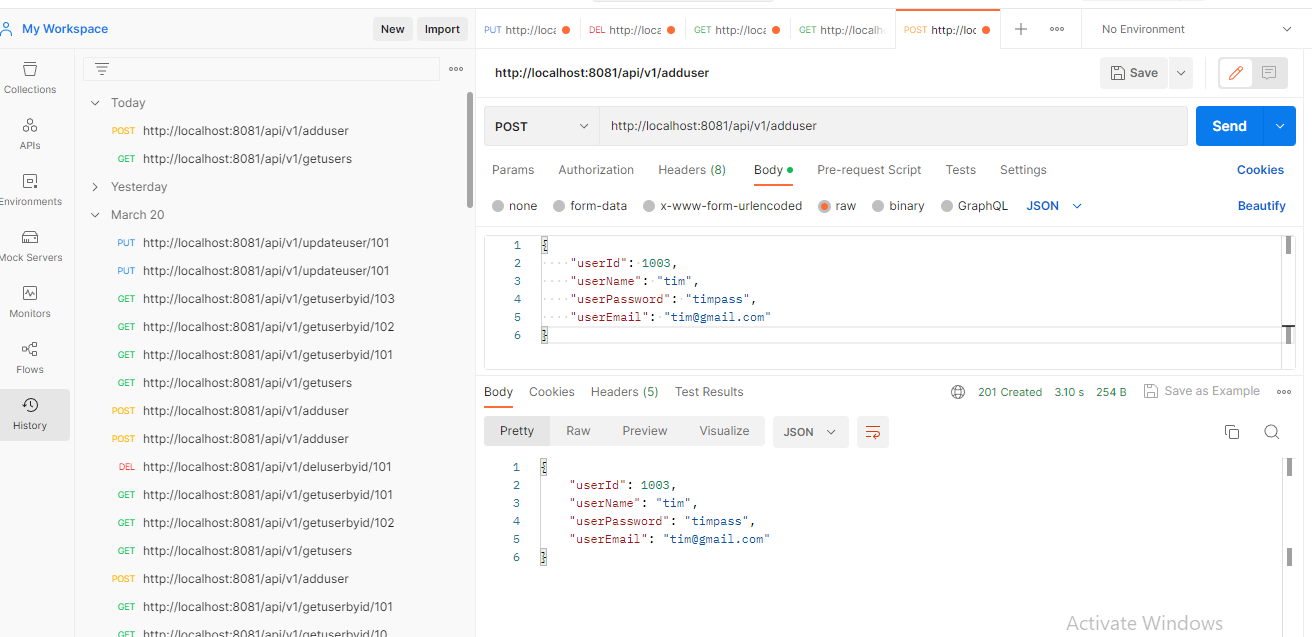
* 1. After that create the database inside the AWS RDS DB instance and then create table as well as Add few data for demo.



* 1. Now Test your SpringBoot application by running it locally first.

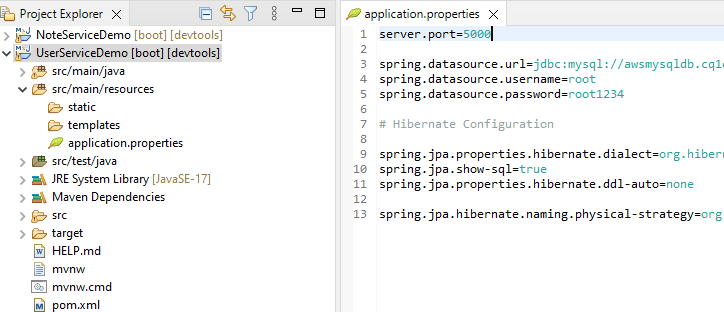




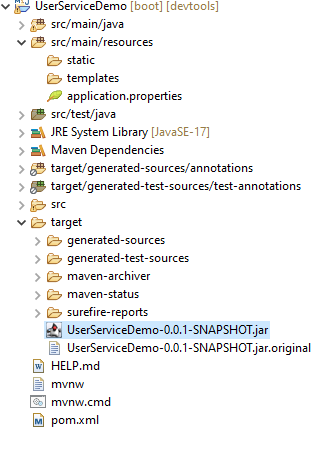


* 1. My SpringBoot Microservice is running fine in local machine. As you can see in above screen shots of postman.

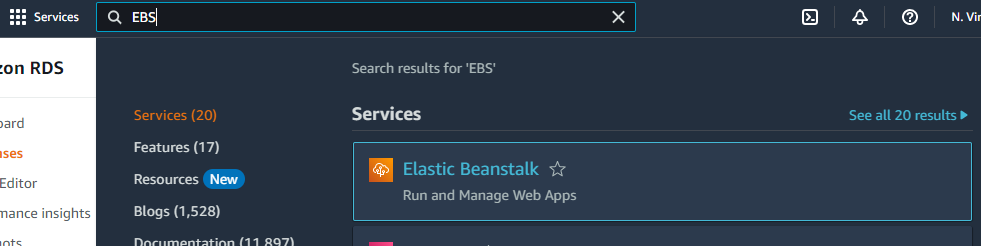
1. Deploy the application in AWS EBS service
   1. First we need to change the port to 5000 because EBS expect the springboot application to listen port 5000.



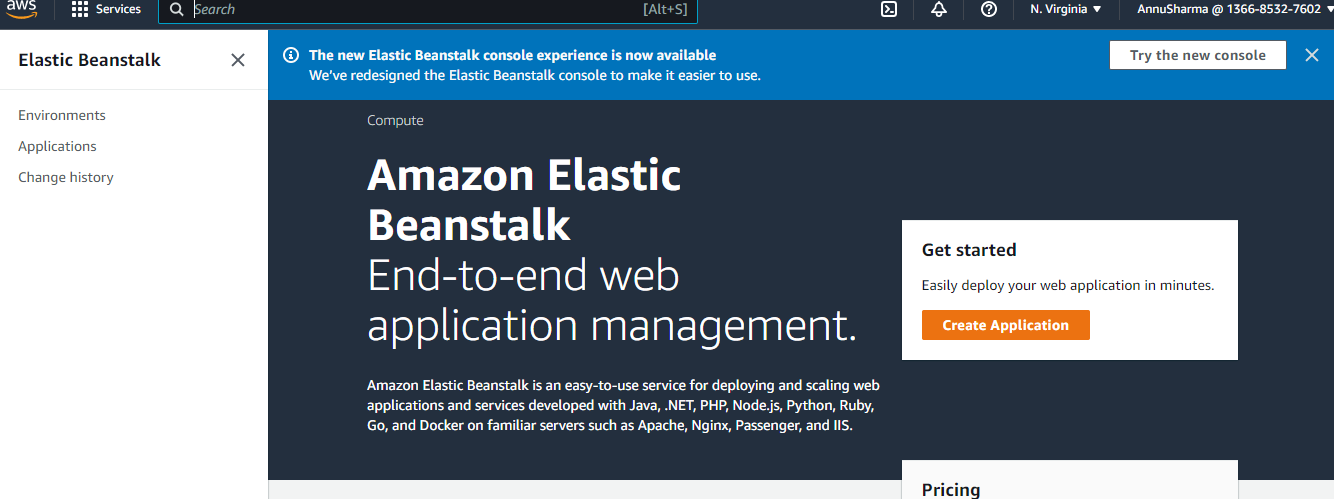
* 1. Package SpringBoot Application as JAR file
     1. Run the command mvn clean
     2. Run the command mvn build
     3. After this JAR file will be generated which we can use for deployment



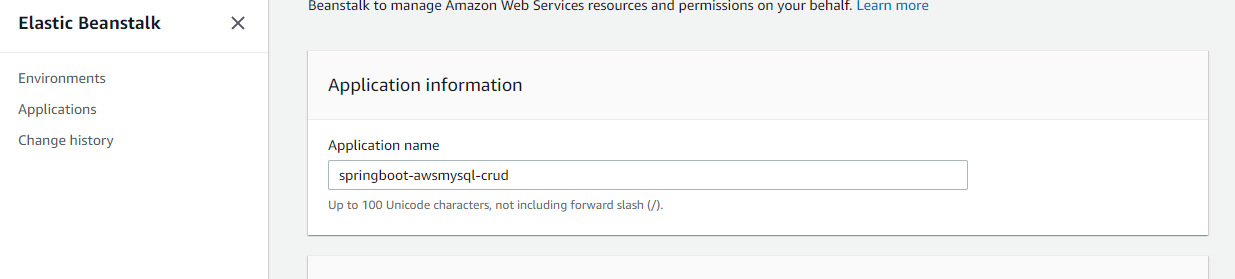
1. Deploy SpringBoot JAR file on AWS using Elastic beanstalk service
   1. Go to AWS Console and choose EBS service



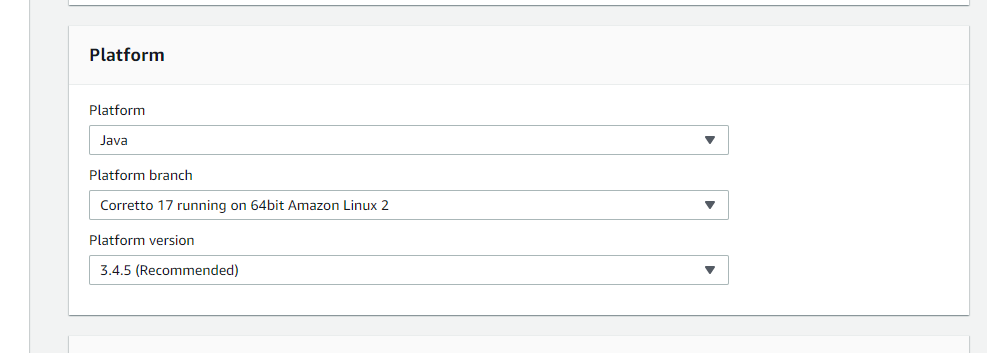
* 1. Now click create application



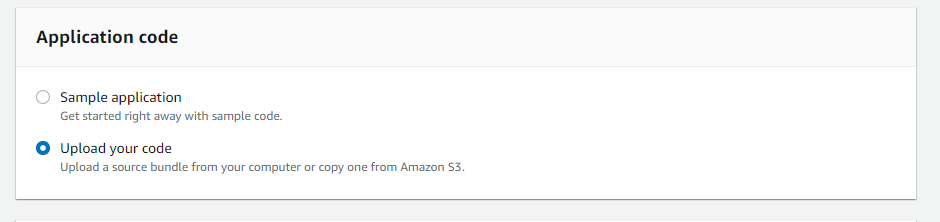
* 1. Give application name



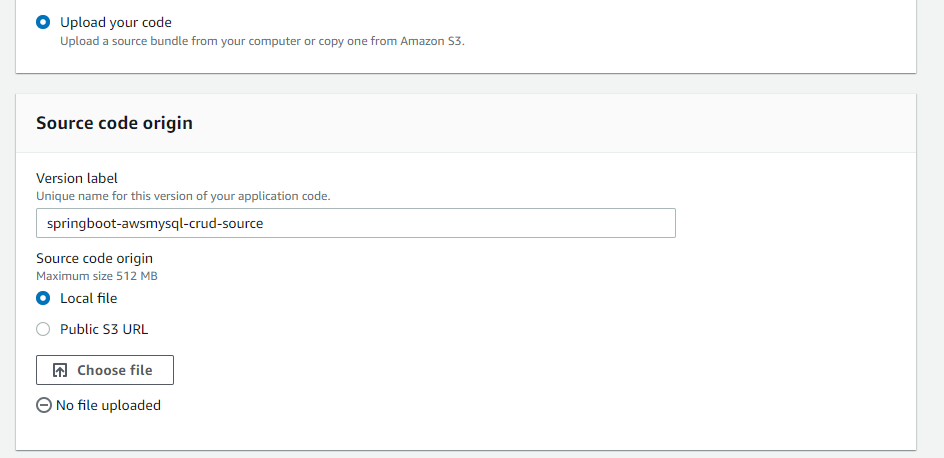
* 1. In platform choose Java



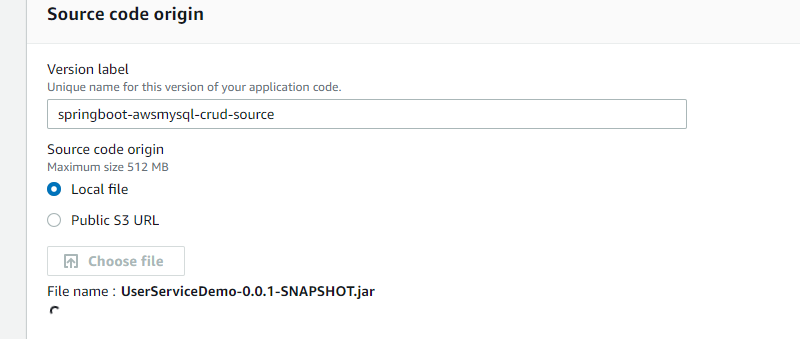
* 1. Now choose the option upload your code

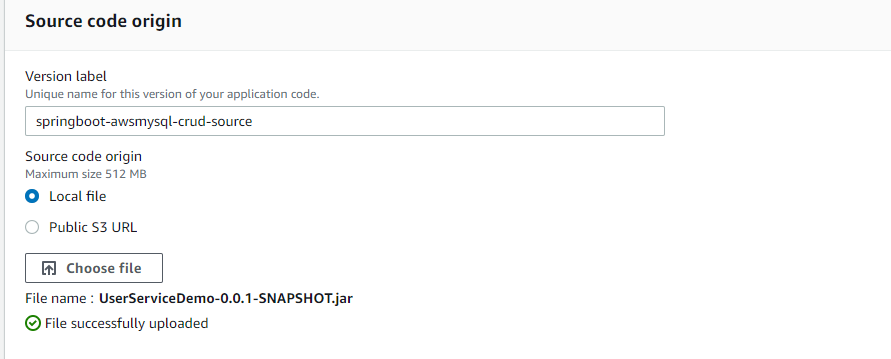


* 1. Click choose file and then upload the JAR file created for your Spring boot application

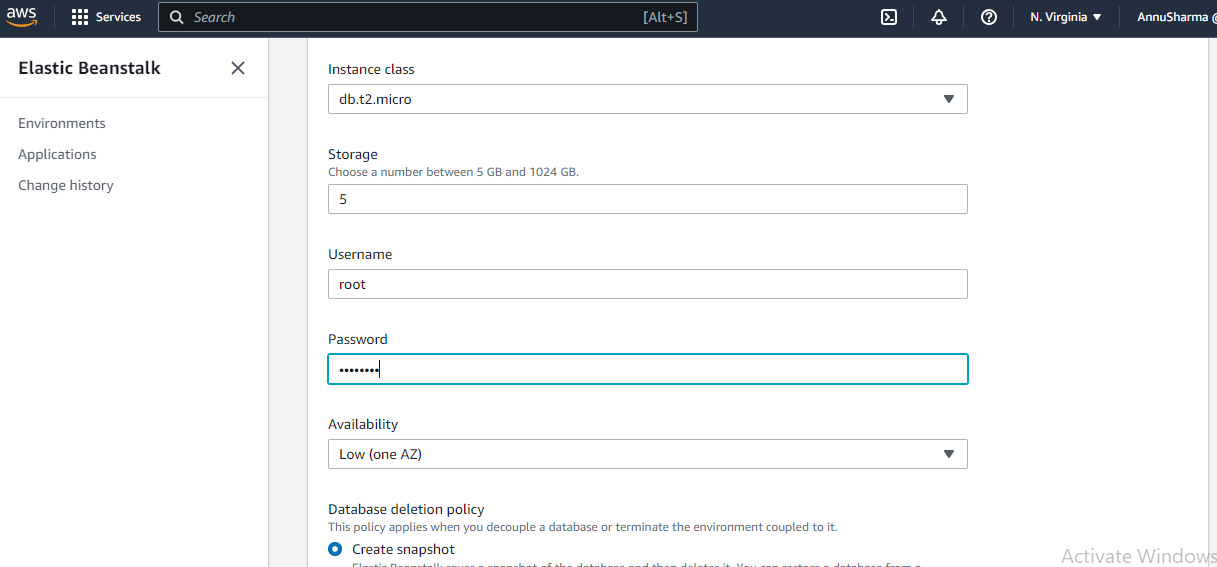


* 1. Go to JAR file path and then select the JAR file to upload

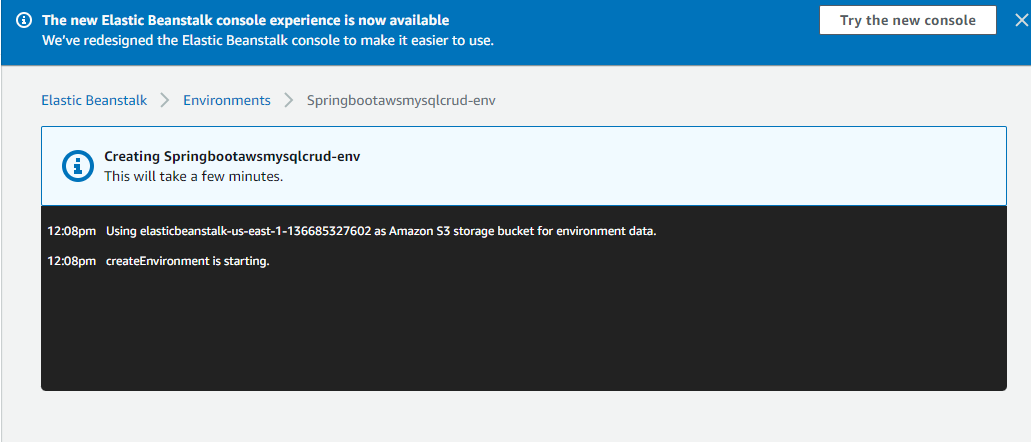




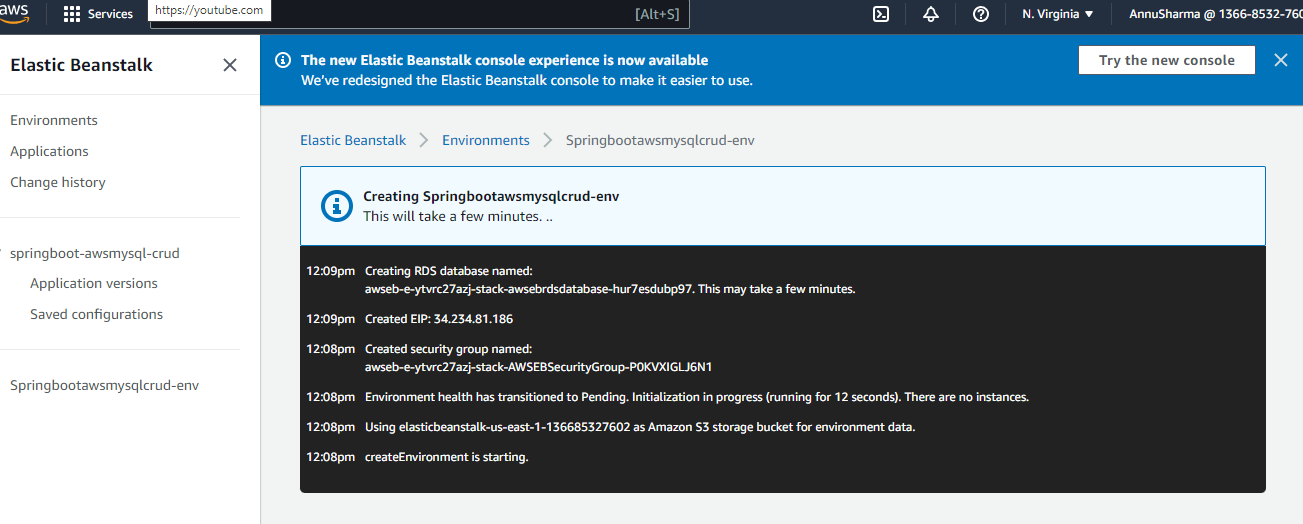
* 1. Now the JAR file uploaded successfully. EBS service will provide all the resource which is required to run spring boot application. Behind the scene it will create EC2 instance and S3 bucket and it will upload the Jar file inside the S3 bucket.
  2. Now click in Configure More Options and then go to Database options and then click on EDIT. Here provide the username and password for your database.
     1. root
     2. root1234



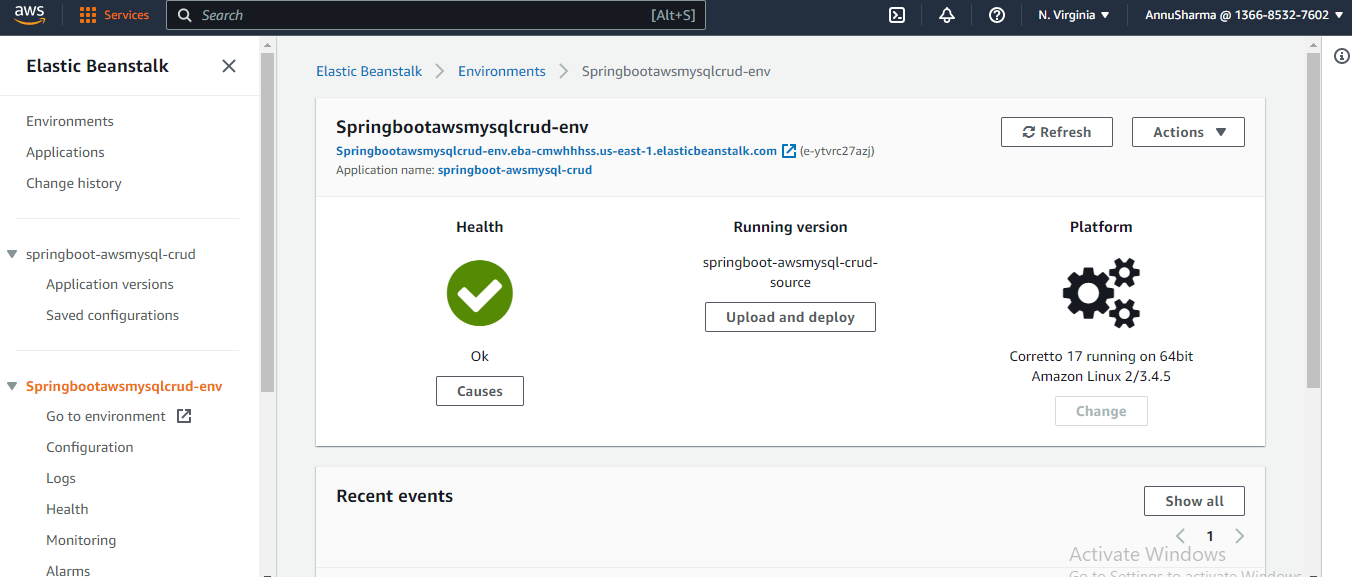
* 1. Now Click Save button
  2. After that Click the “Create App” button



* 1. Now it will setup the environment to run the application



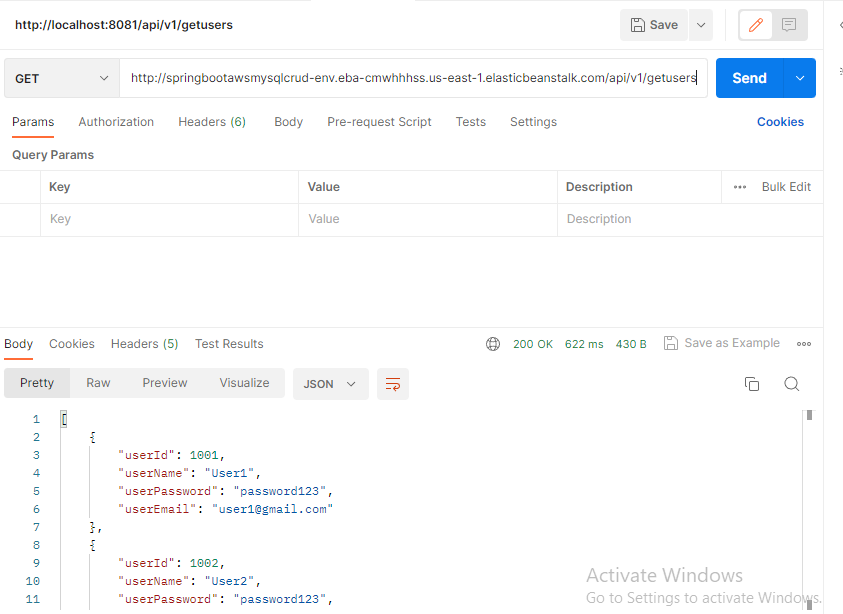
* 1. Application creation will take some time and once it is successfully deployed it will provide the link which can be used to work with the application.

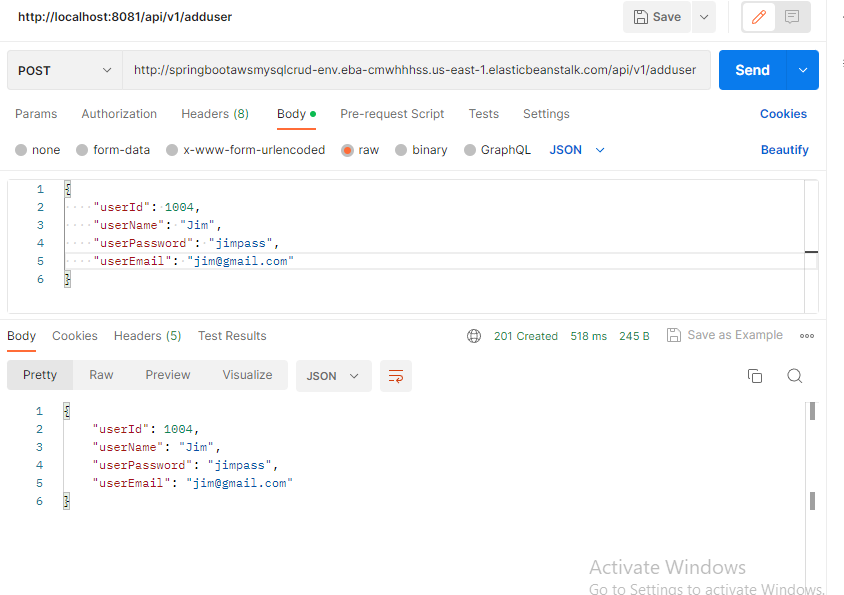


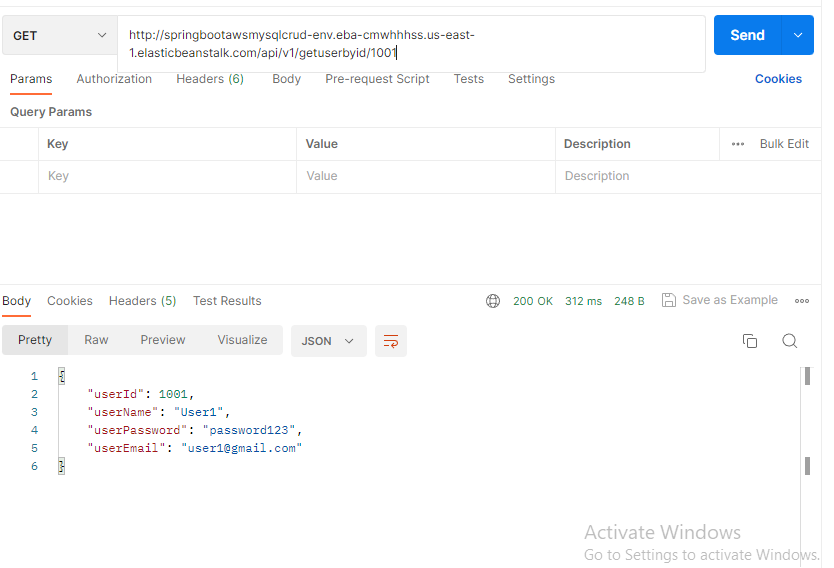
* 1. Now you can test the application by using the URL provided
     1. [**Springbootawsmysqlcrud-env.eba-cmwhhhss.us-east-1.elasticbeanstalk.com**](http://springbootawsmysqlcrud-env.eba-cmwhhhss.us-east-1.elasticbeanstalk.com/)

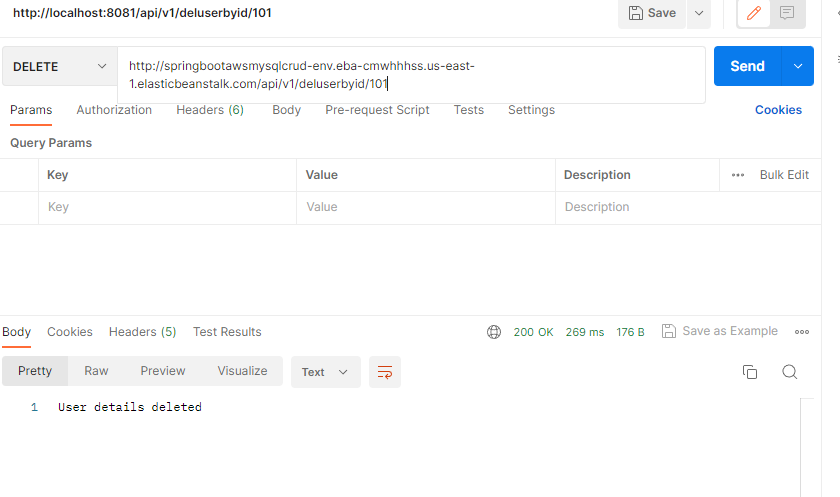


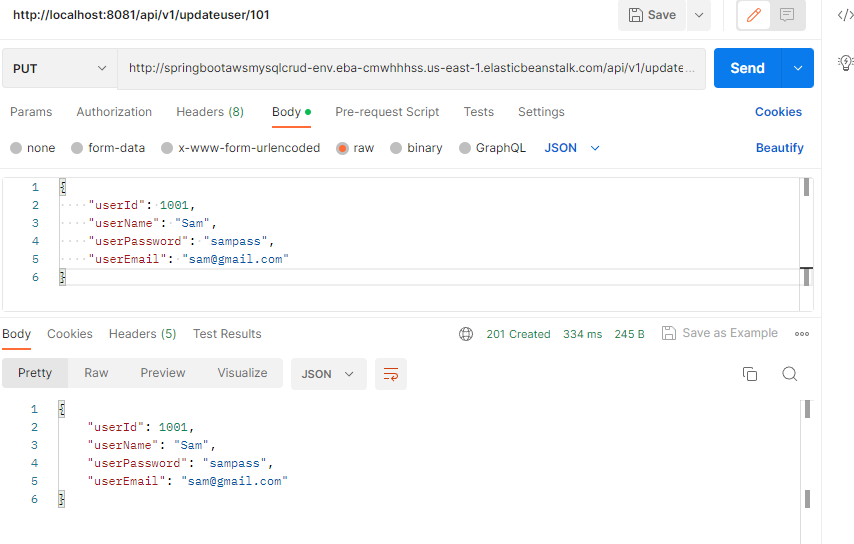
1. Test over POSTMAN client now.











1. Thank you..

For reference

* 1. Now the JAR file uploaded successfully. EBS service will provide all the resource which is required to run spring boot application. Behind the scene it will create EC2 instance and S3 bucket and it will upload the Jar file inside the S3 bucket.

