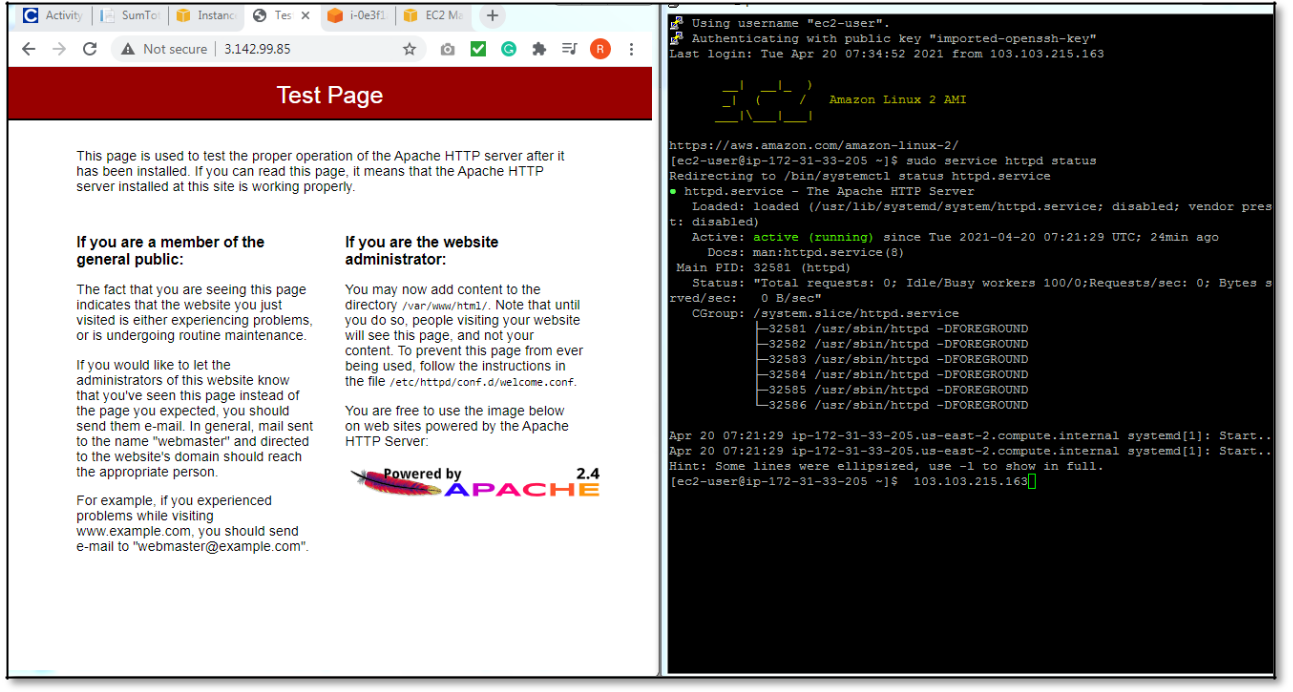
Amazon Web Service [Stage-4] Putluru Maneesha [896815]

**AWS-Hands-on**

**EC2-Hands-on**

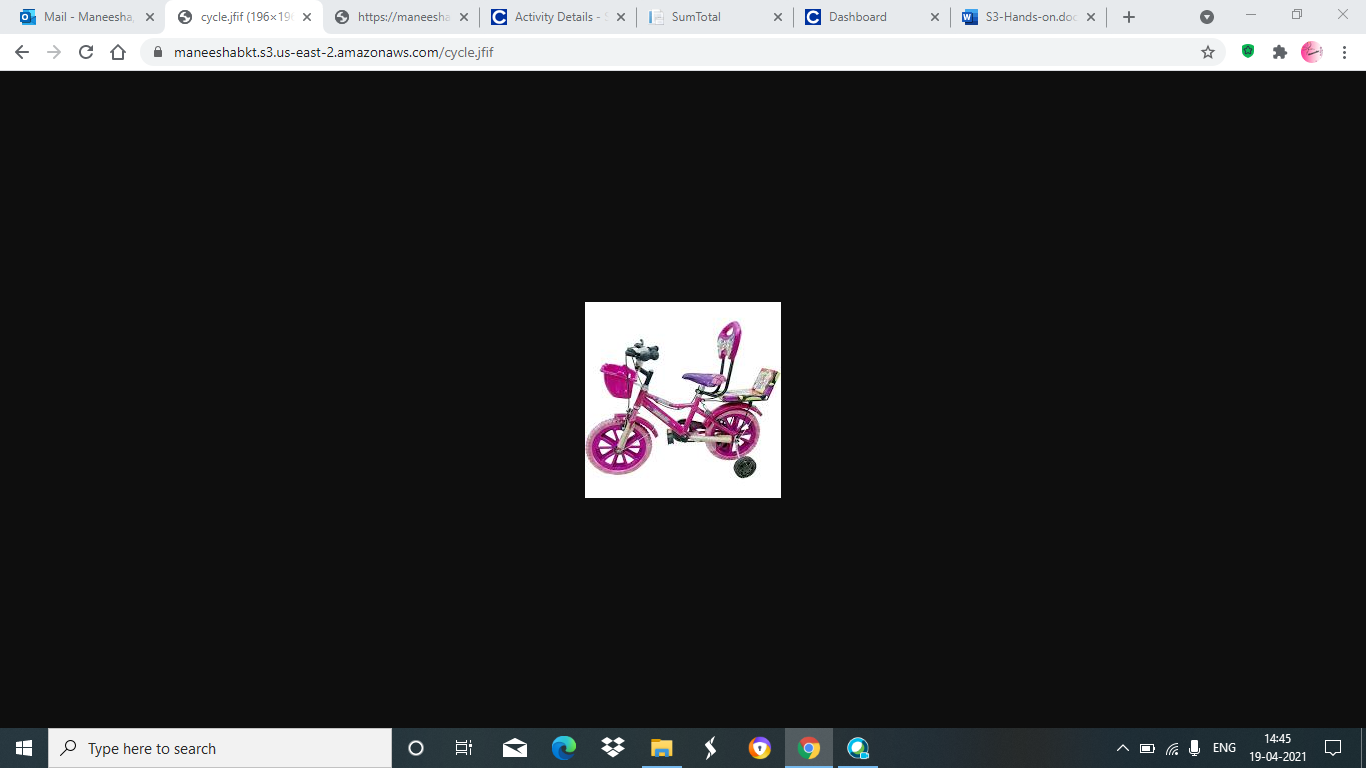
Create an EC2 instance, connect to it from your local system and install apache web server on the EC2 instance



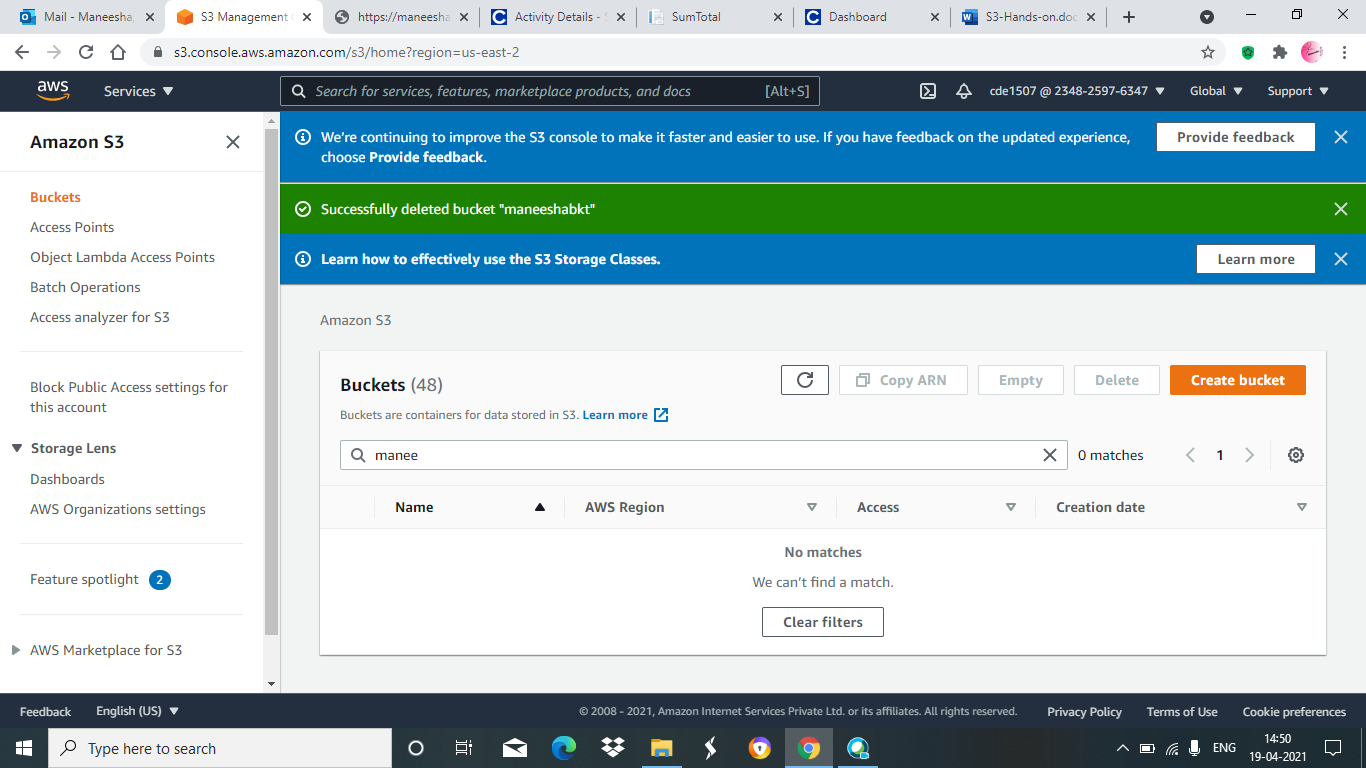
Type the public IPV4 address on the browser url bar and we should get the above shown screen

**S3-Hands-on**

Create a S3 bucket and store an object in it. Enable to object for public access so that anyone can access it through a web browser.



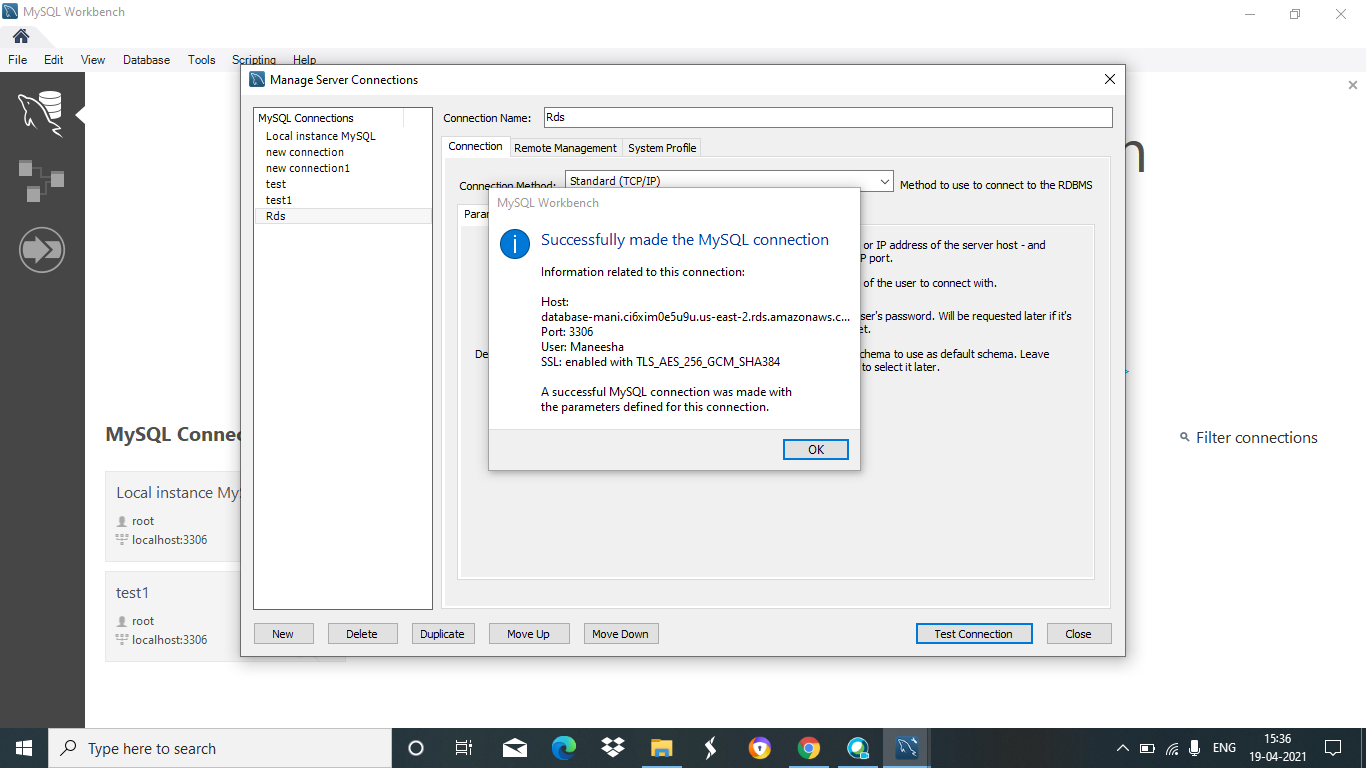
Access the object by clicking in the Object URL and we can view the object in the browser



After Deleting the bucket we will get the update that the bucket is deleted Successfully

**RDS Hands-on**

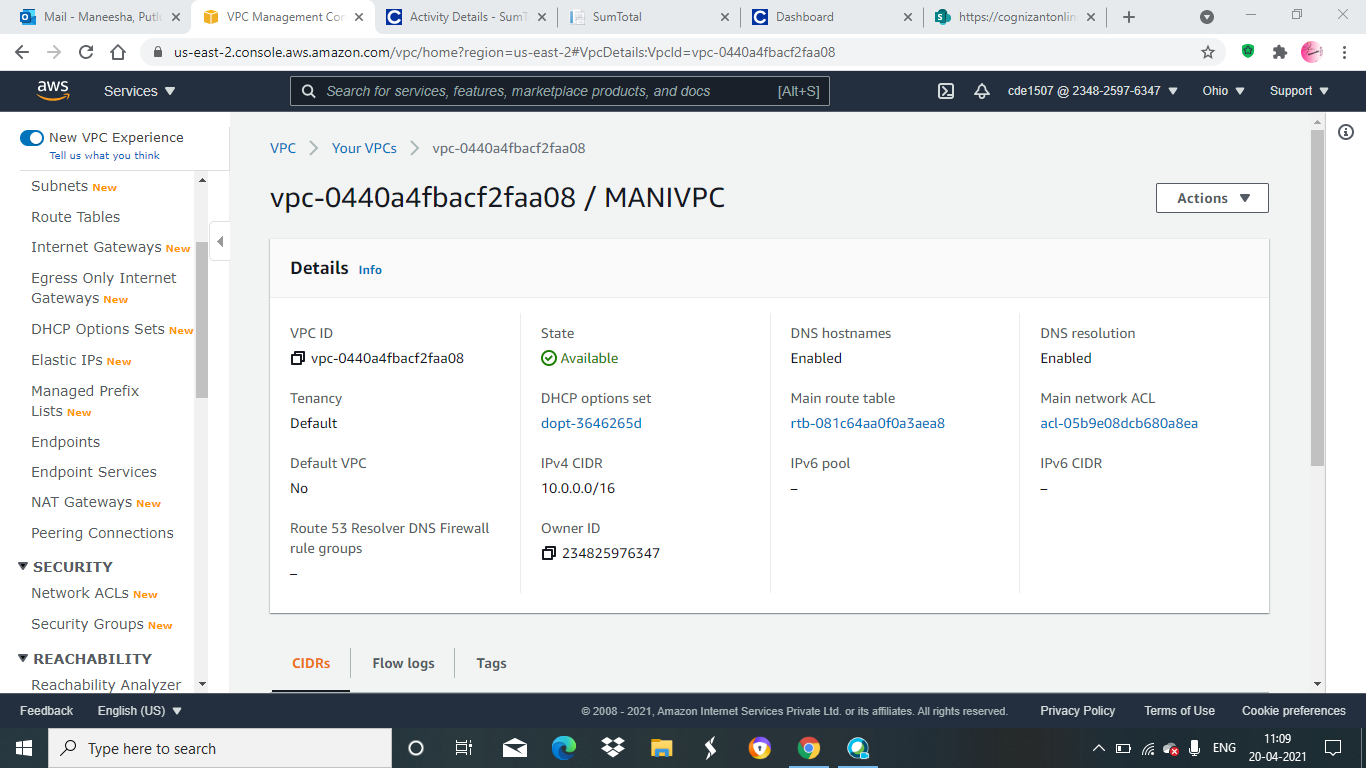
Create a RDS database in AWS and access it through the local client tool.



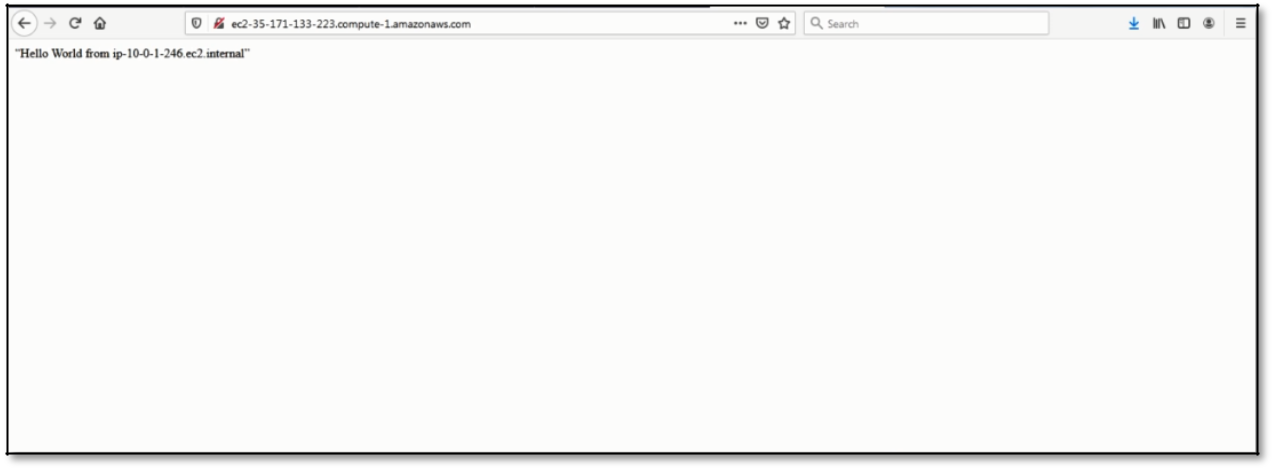
After creating database we will be used to connect RDS MySQL server

**AWS lab hands-on**

Create various infrastructure components that will be used to build a web server within the AWS cloud environment.



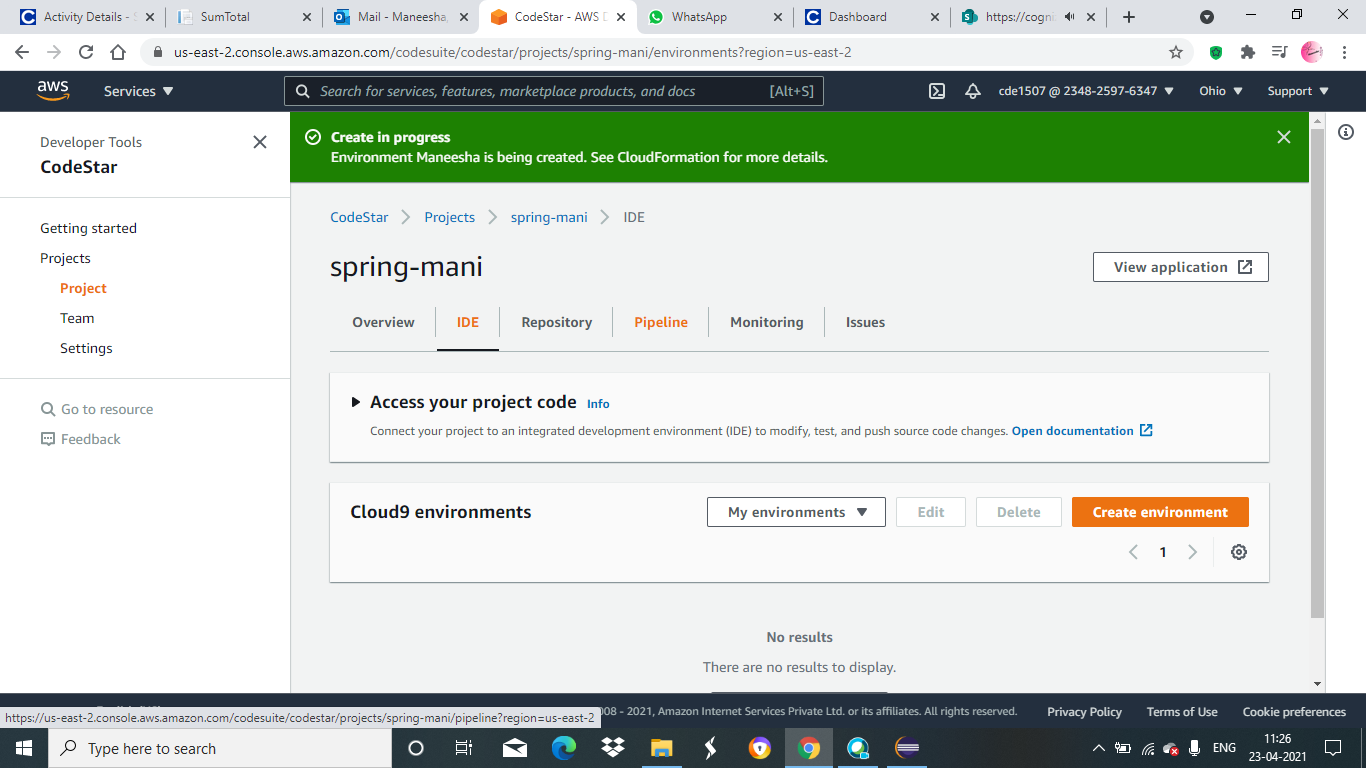
In the above image we can see the information of subnets, security groups, route tables



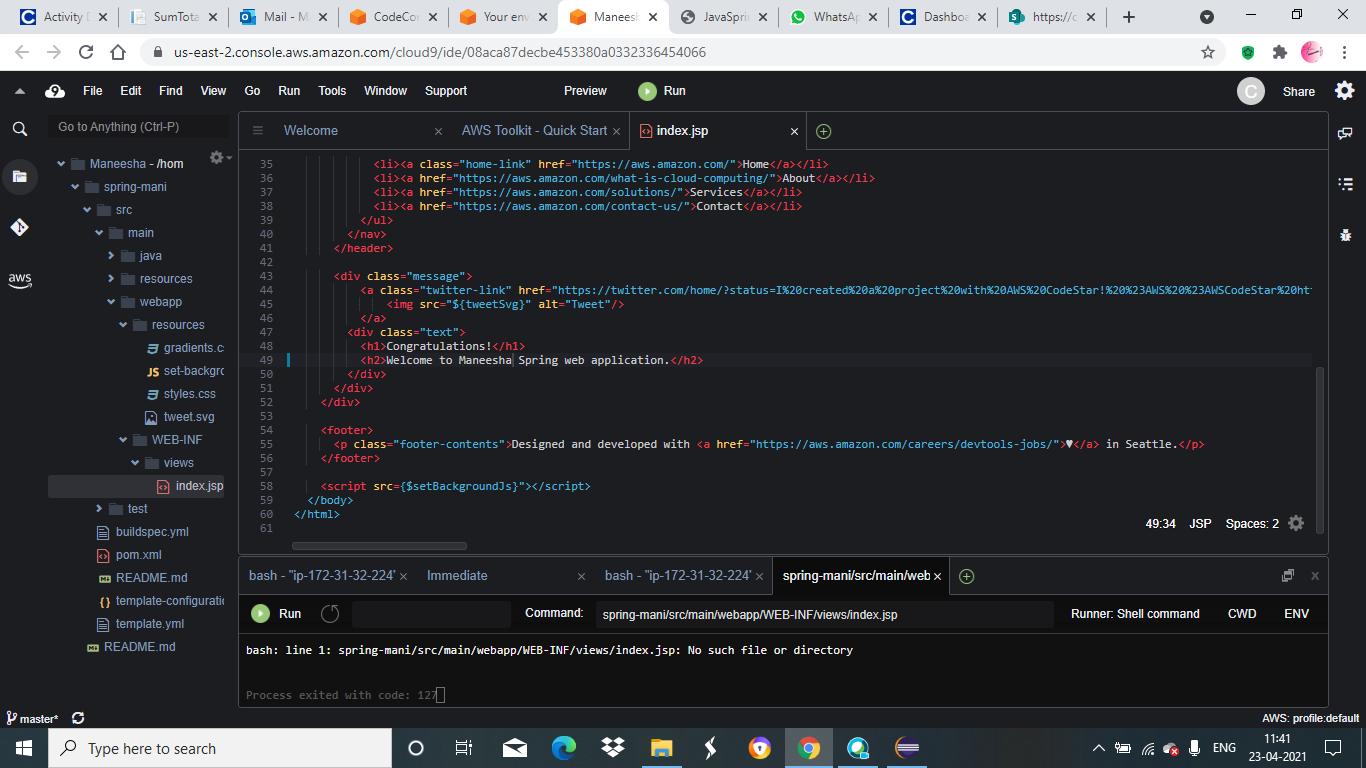
By clicking the url we will see the message in browser.

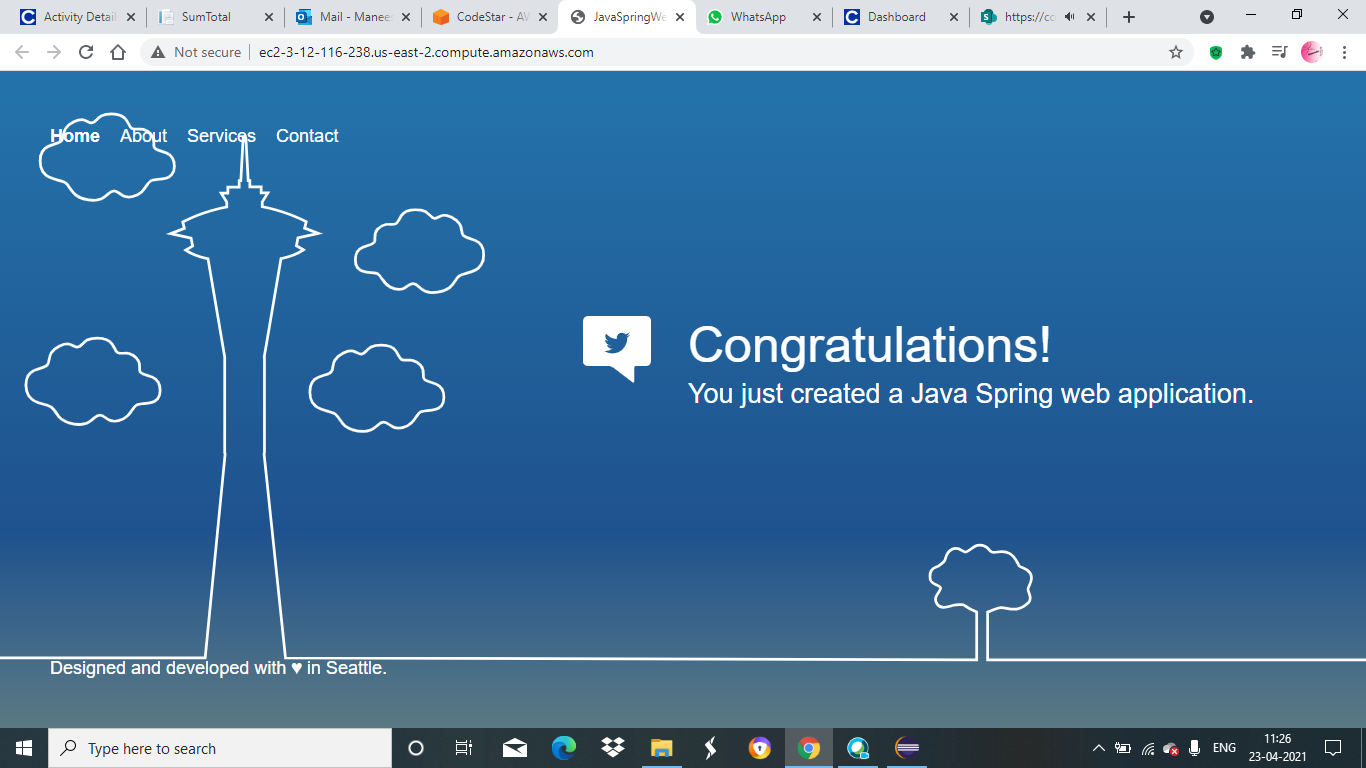
**CC ID lab hands-on**

We will able to deploy a spring web application using a Continuous Integration (CI)/ Continuous Delivery (CD) pipeline and the IDE provided by AWS.



We can see above screen that SPRING PROJECT is creating

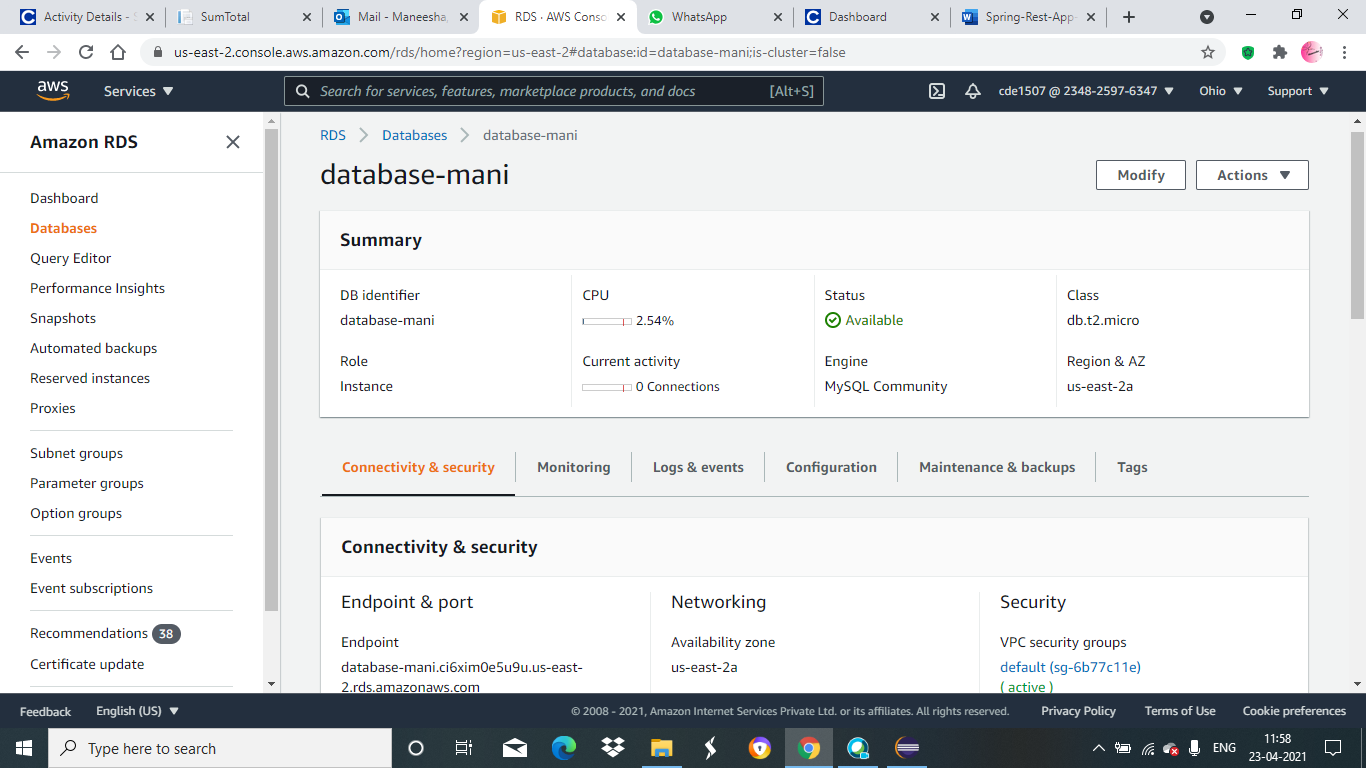
 AWS IDE



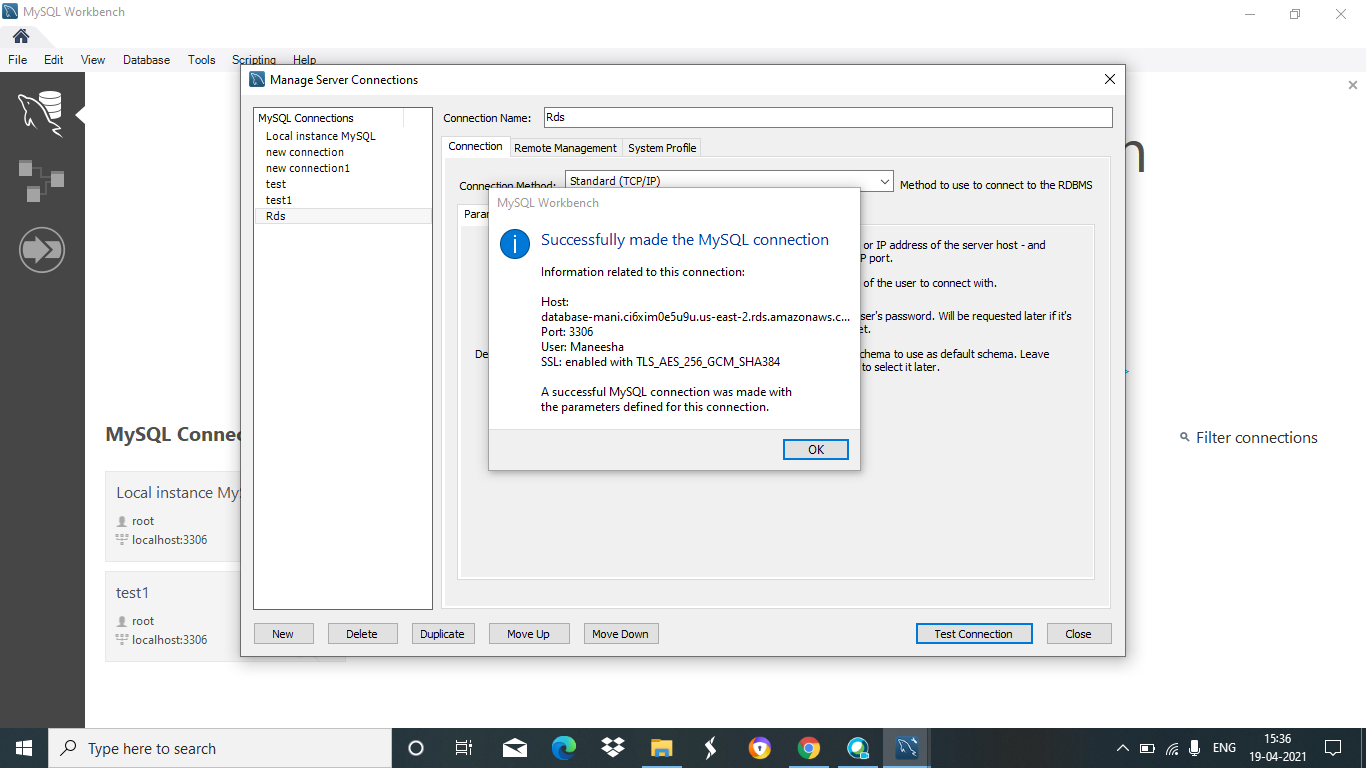
Successfully deploy a spring web application using CI/CD

**Spring-REST-with-RDS-Backend**

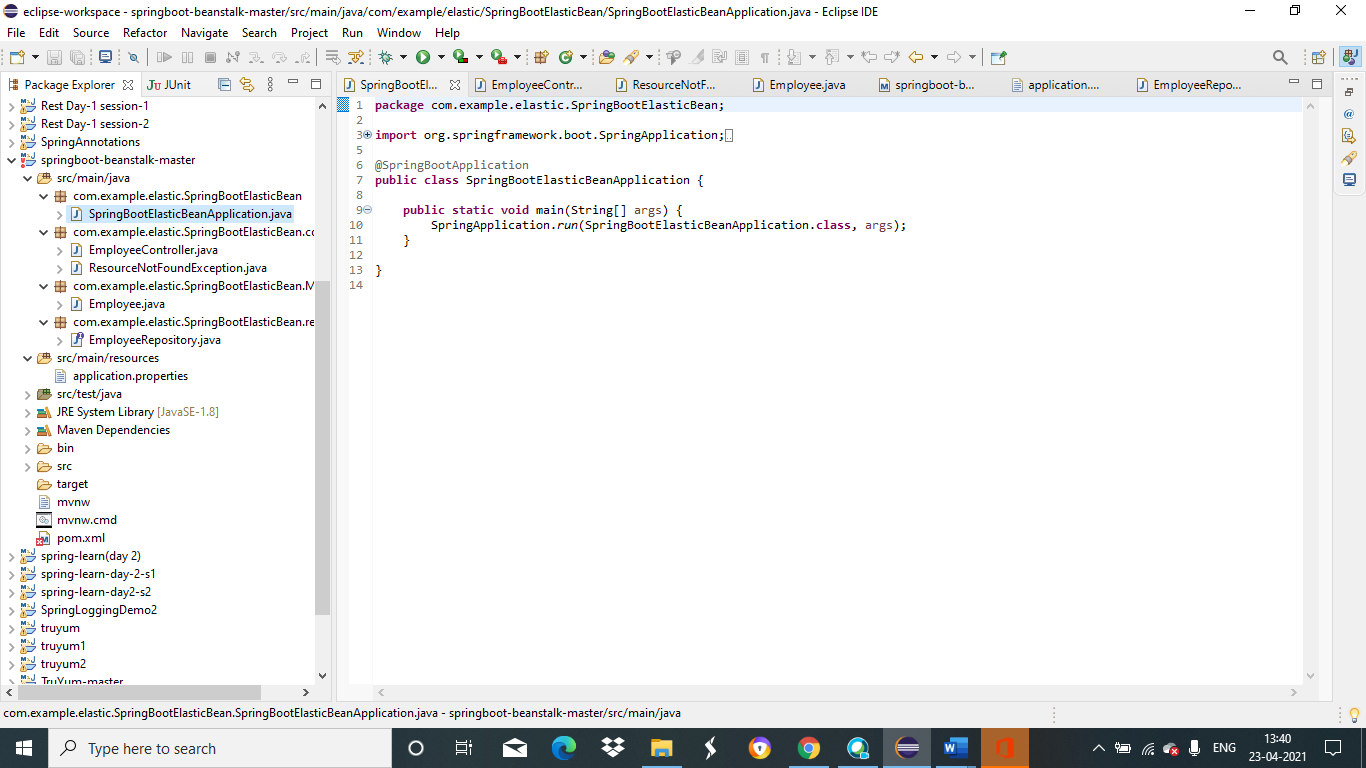
Create a Spring REST application that perform Read and Insert operation on RDS database. Deploy the application in AWS Elastic Beanstalk and access the application from anywhere.



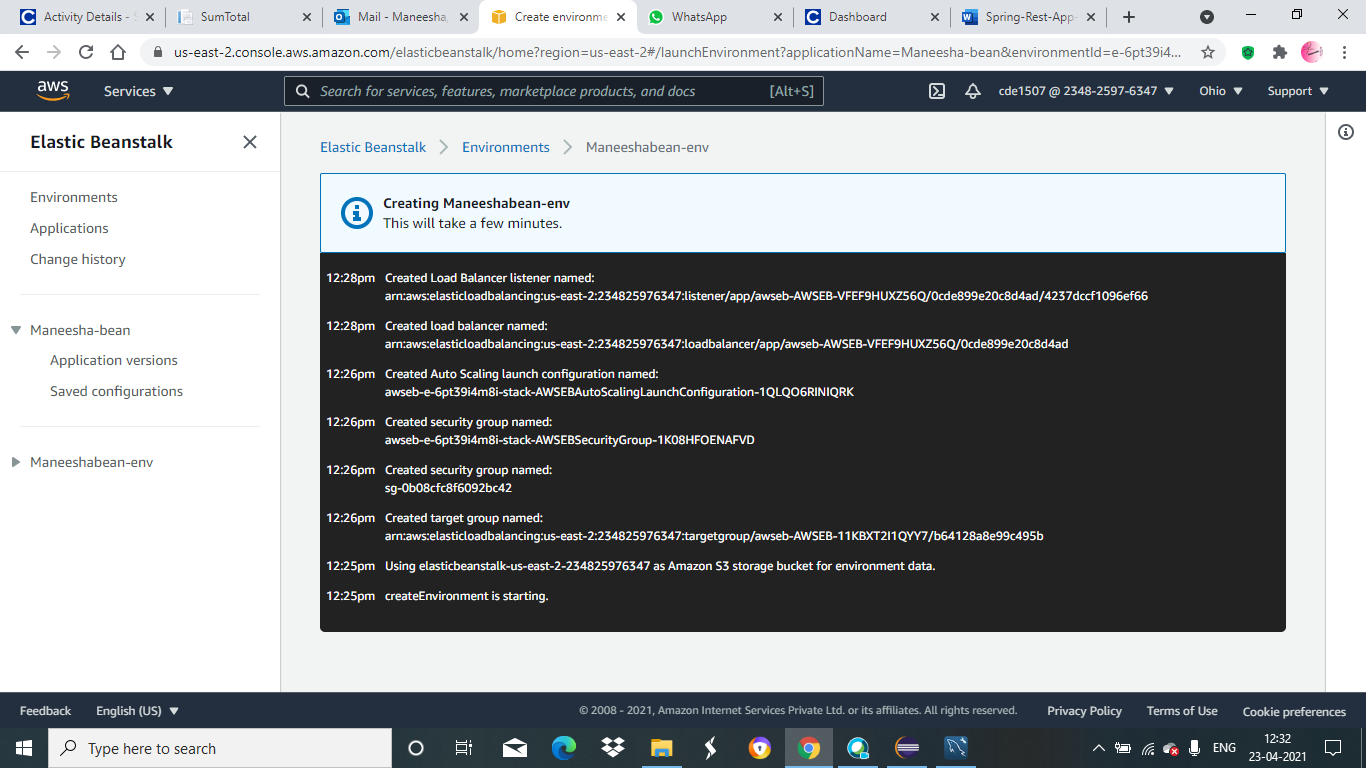
We can see above screen that our database is being created



We will be connected to the RDS MySQL Server



Create a spring REST application using Spring Boot.



We can see above screen that ELASTIC BEANSTALK is created

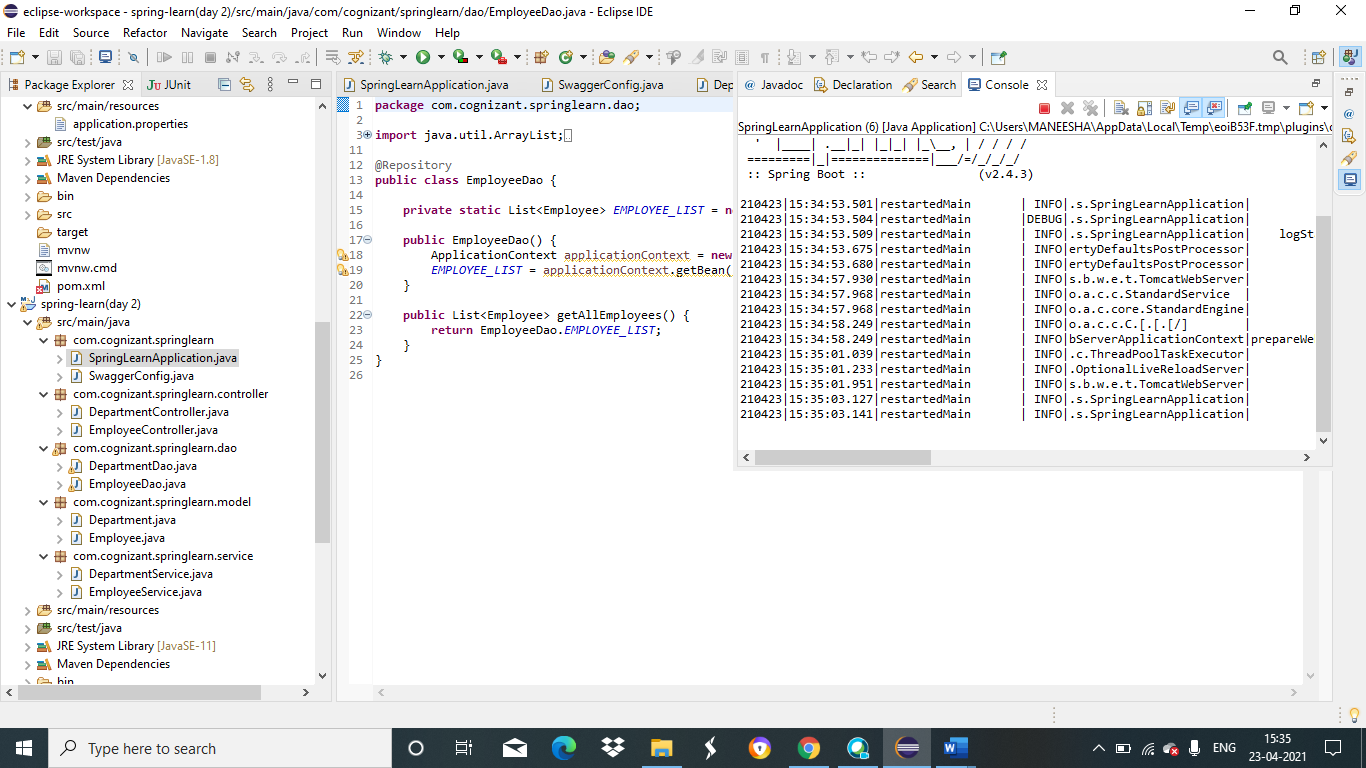


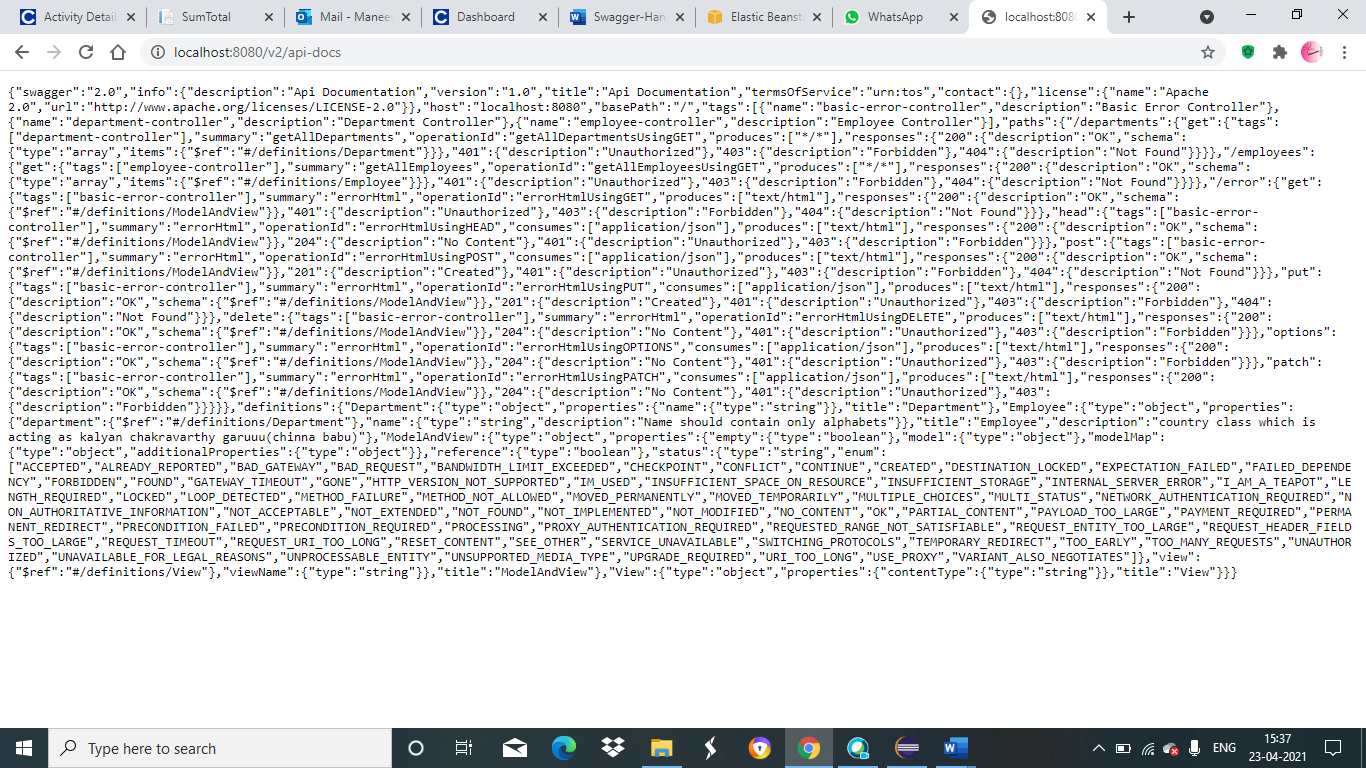
We can see that the record has been successfully inserted into the RDS database

**Swagger Hands-on:**

Make use of Swagger to create documentation for RESTful/microservices.

Create a simple RESTful service using Spring BOOT





**Spring MVC Client for Spring Rest Service**

Create a client application using Spring MVC and Spring Boot and consume the REST service created earlier from the client application.

