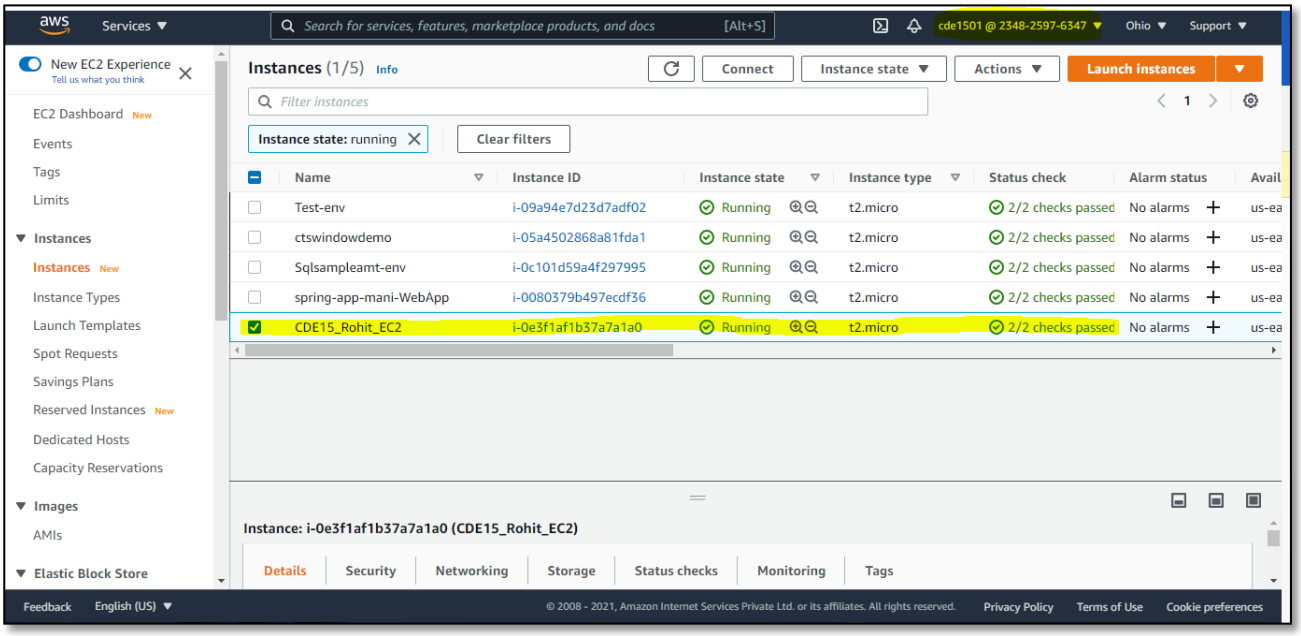
AWS [Hands-on]

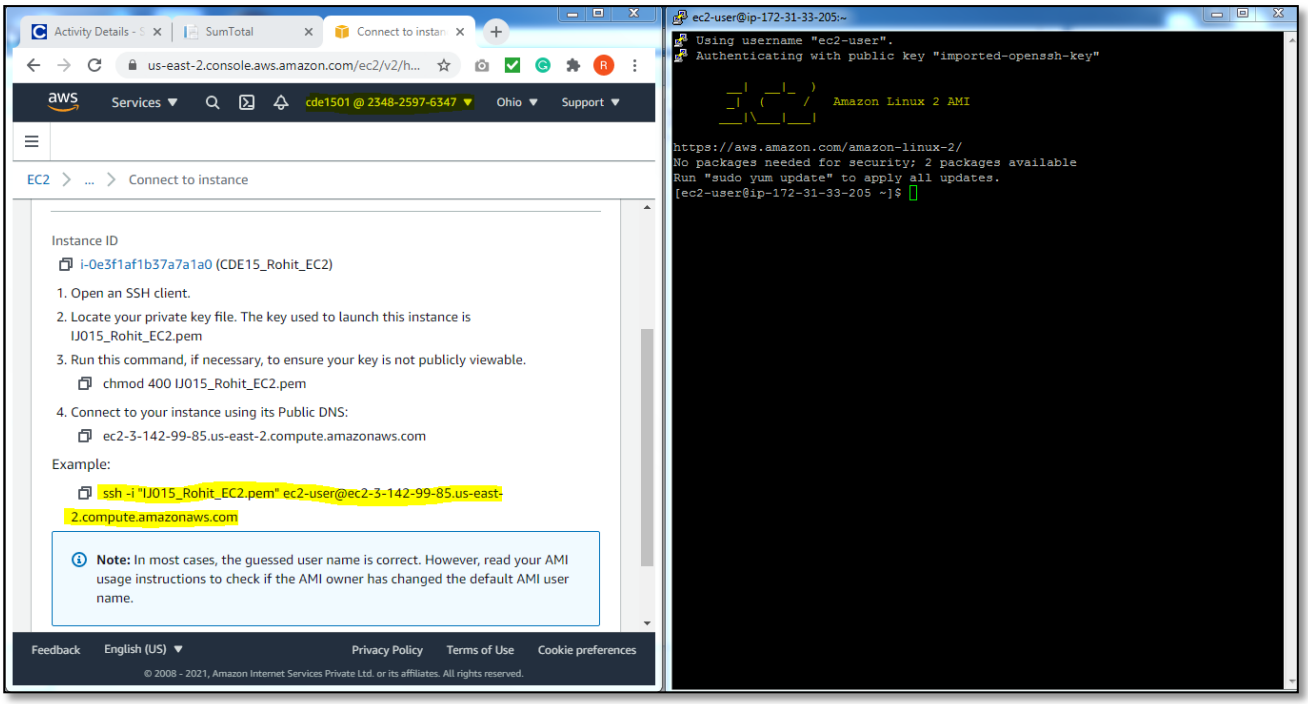
DAY-ONE

- ## [EC2-Hands-on]

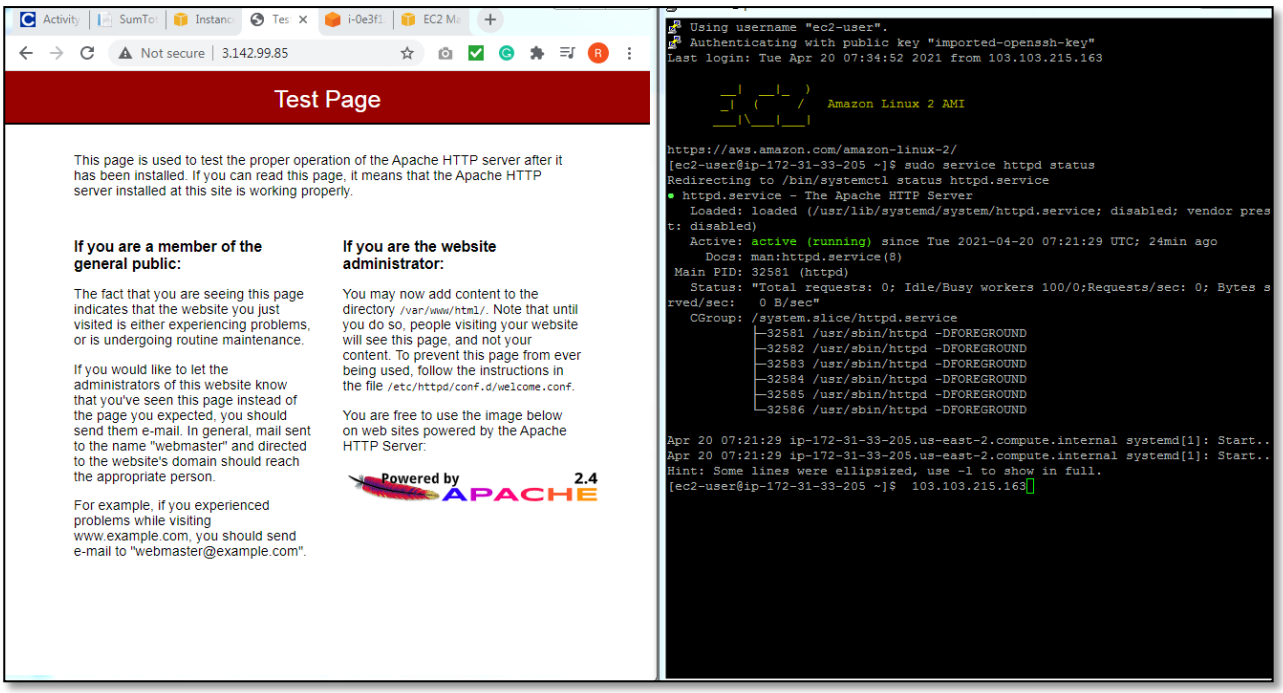
Create an EC2 instance, connect to it from your local system and install apache web server on the EC2 instance.



This indicates my instance is up and running



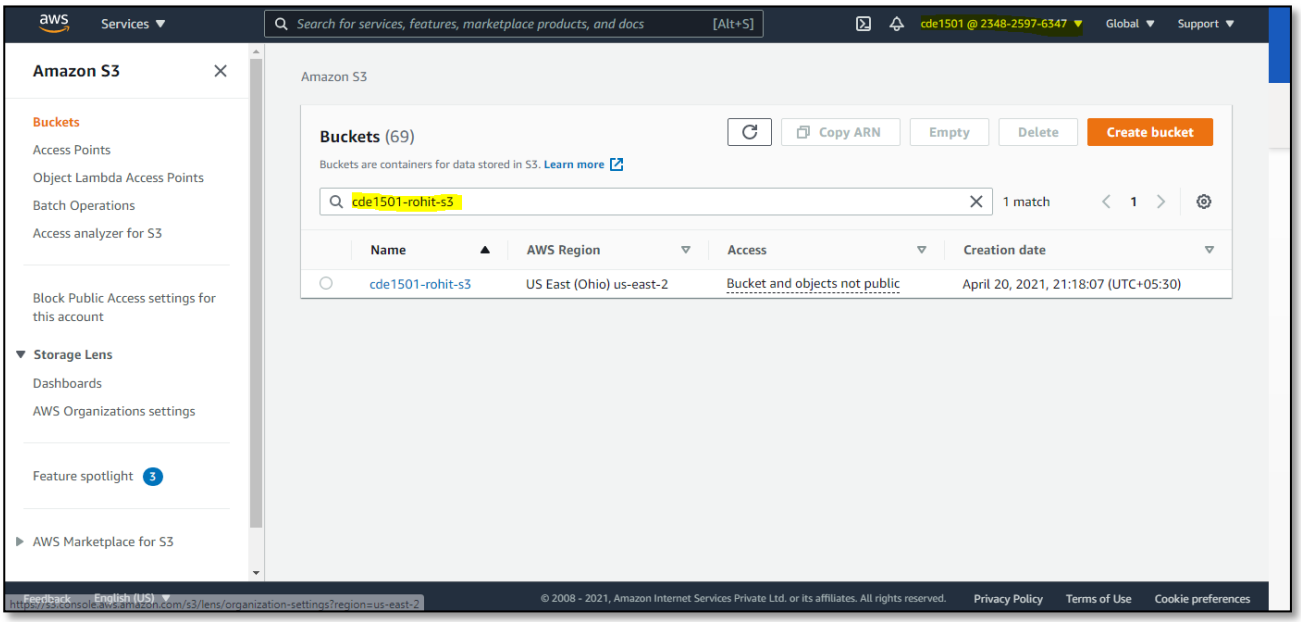
Connected to the EC2 instance



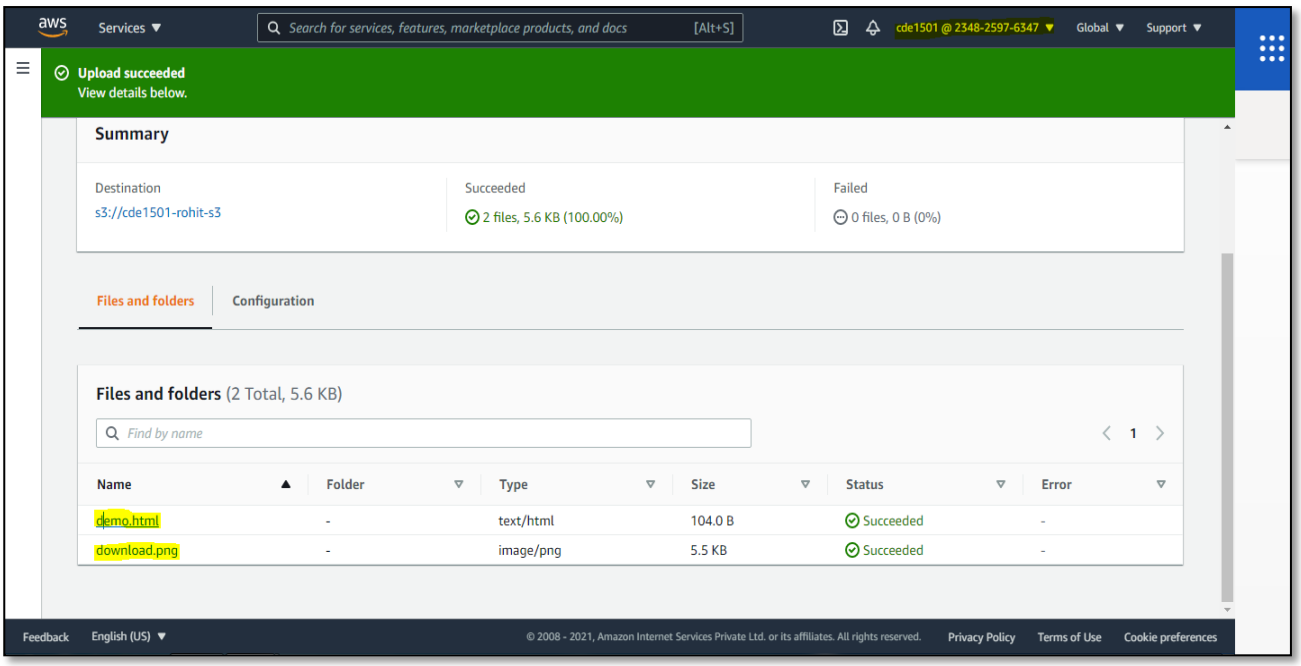
Type the public IPV4 address on the browser url bar and we should get the above shown screen

- **[S3-Hands-on]**

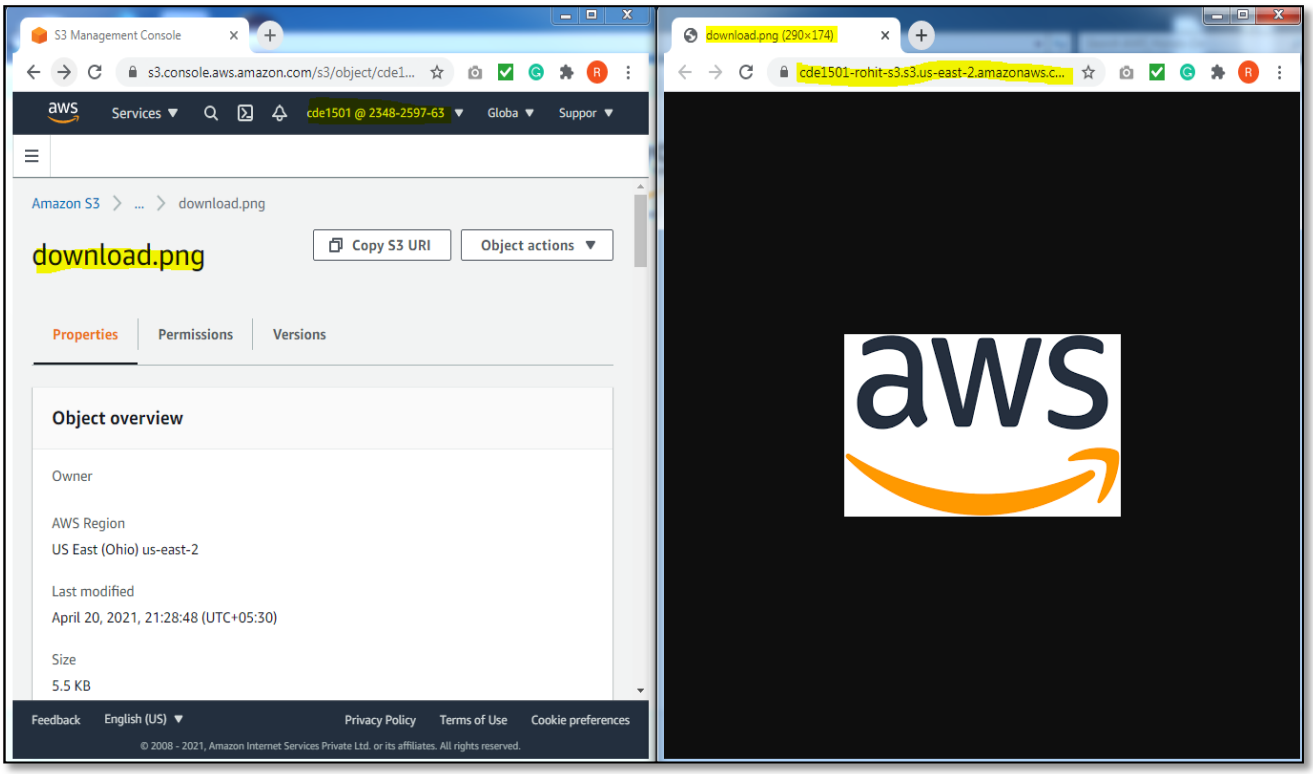
Create a S3 bucket and store an object in it. Enable to object for public access so that anyone can access it through a web browser.



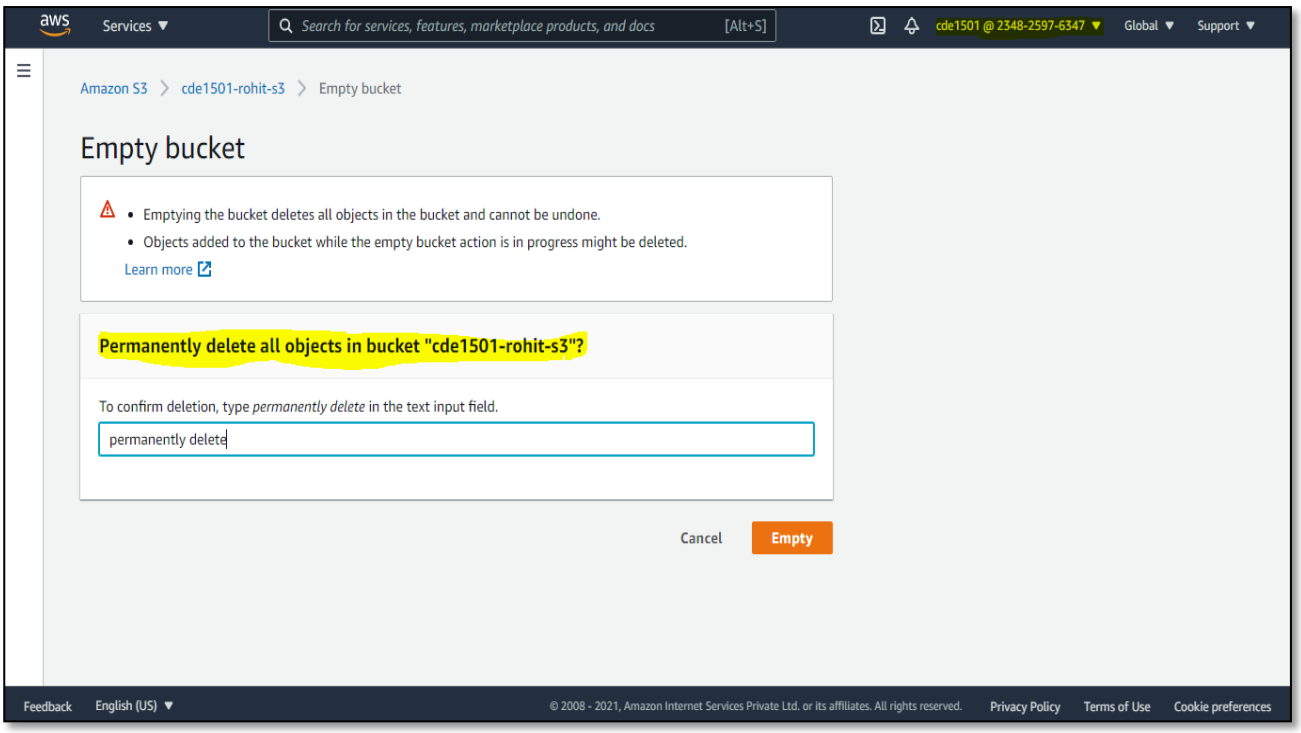
Once successfully S3 created, we will be taken to the above shown



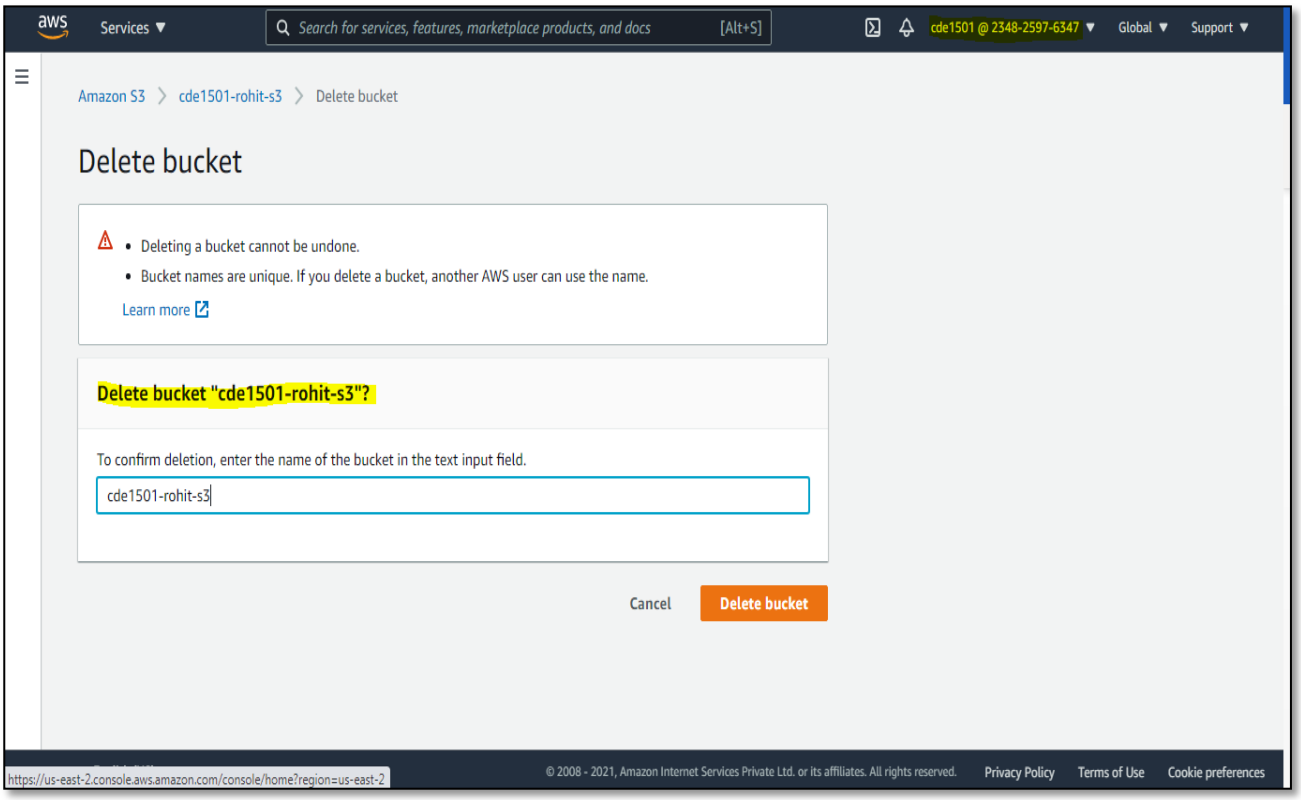
Once files and folders will be uploaded we will be taken to the status screen as shown above



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



Click on the "Empty" button to delete the contents of the bucket

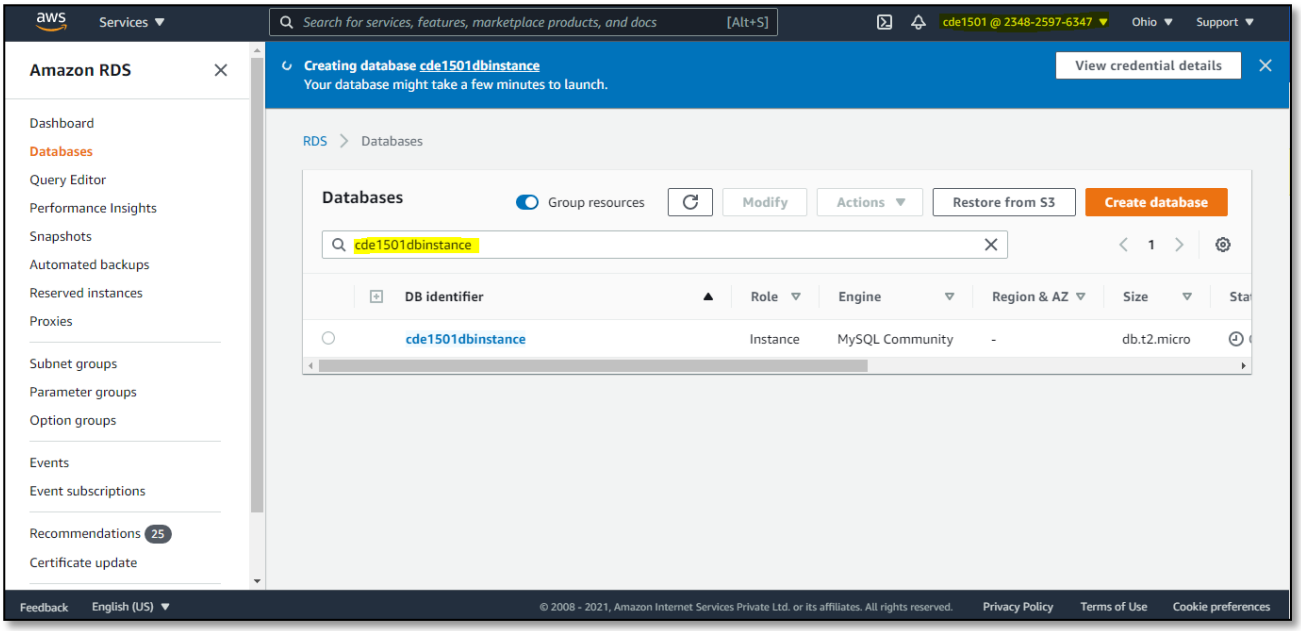


Type the name of the bucket we are deleting and click on the "Delete bucket" button

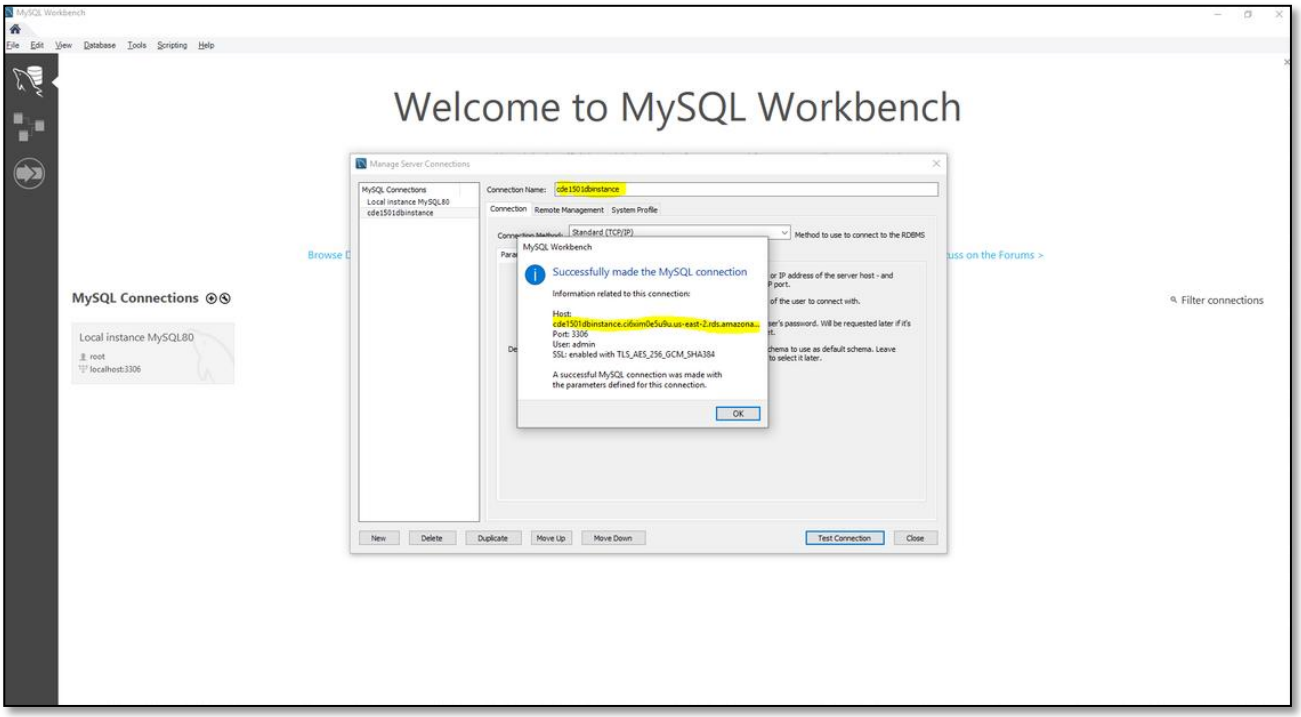
DAY-TWO

▪ [RDS-Hands-on]

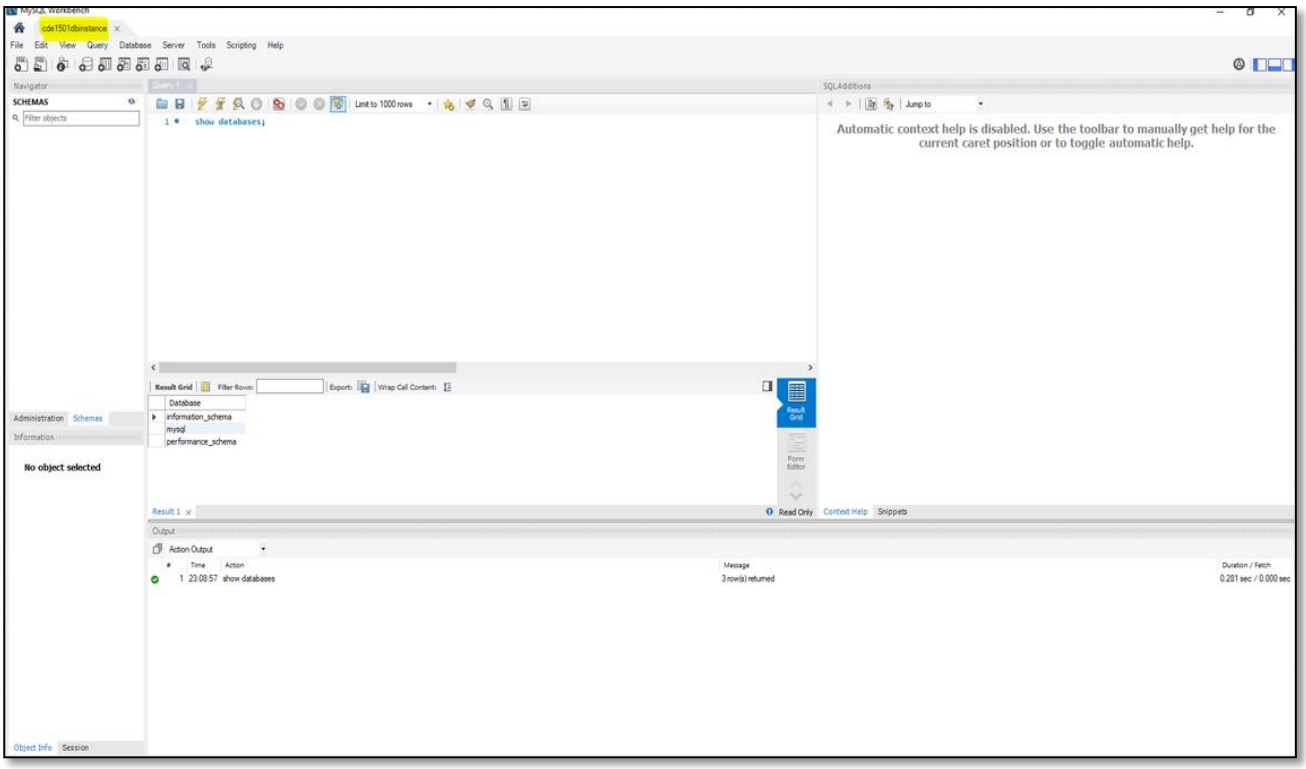
Create a RDS database in AWS and access it through the local client tool.



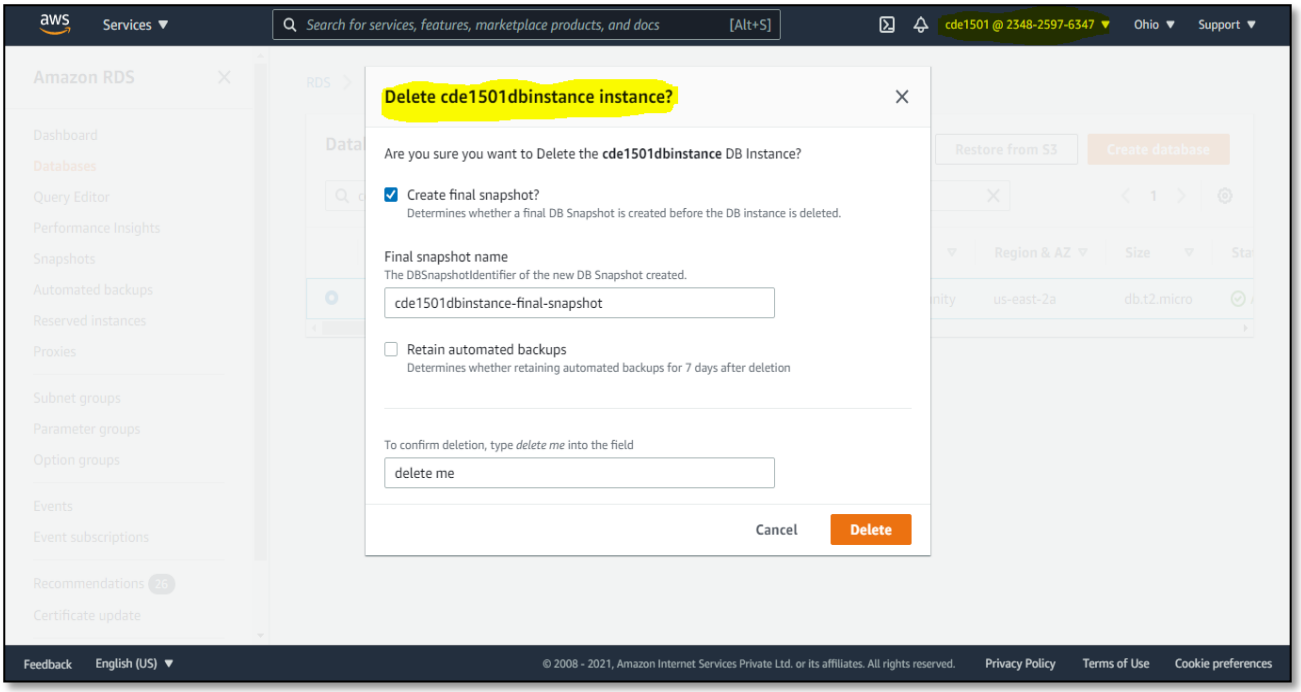
We can see above screen that our database is being created



We will be connected to the RDS MySQL Server



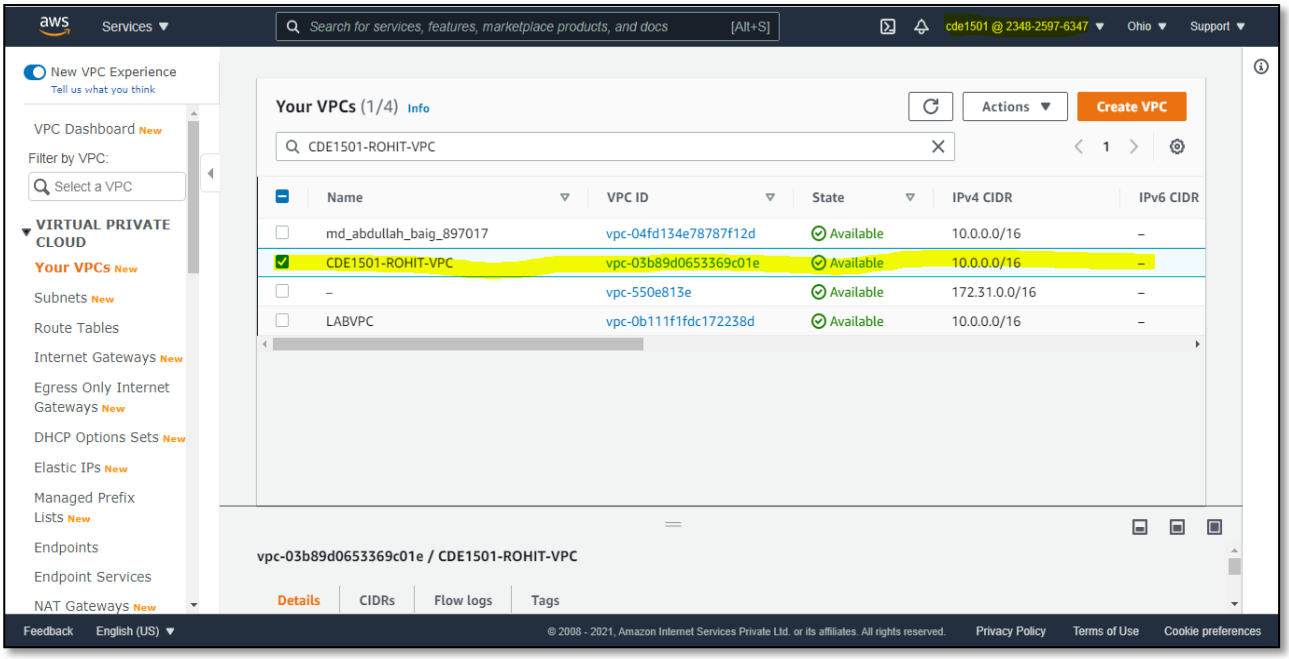
We can see above screen that we are connected to the RDS MySQL server and query window is opened. Here we can issue any sql commands we want to execute against the RDS database



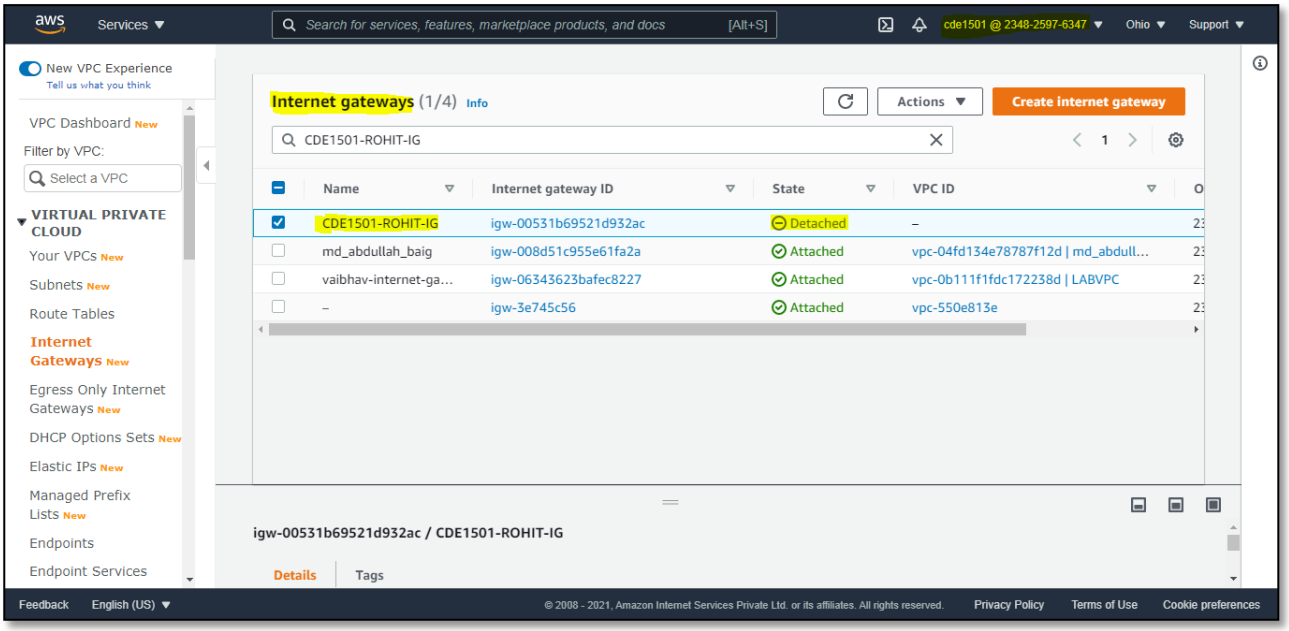
Deleting the RDS database by typing “delete me” in the confirm textbox and press “Delete” button

- **[AWS-lab-hands-on-practise]**

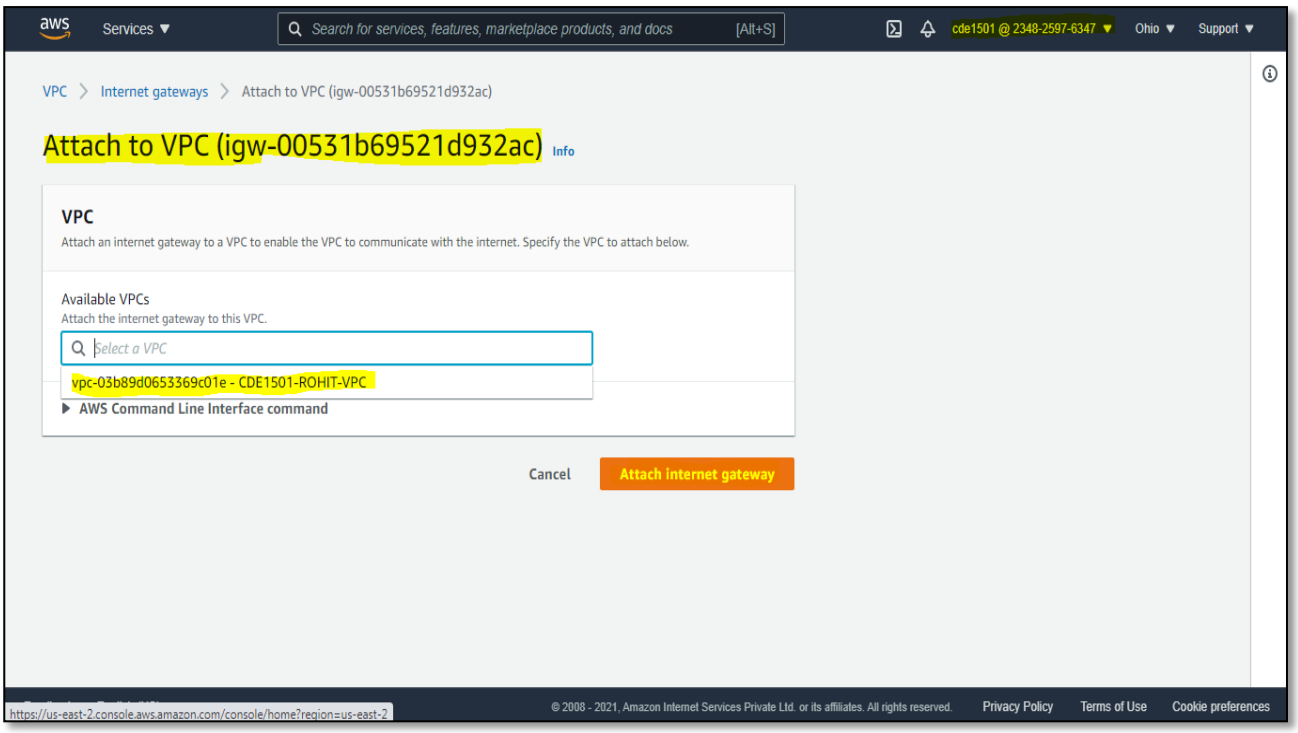
Create various infrastructure components that will be used to build a web server within the AWS cloud environment.



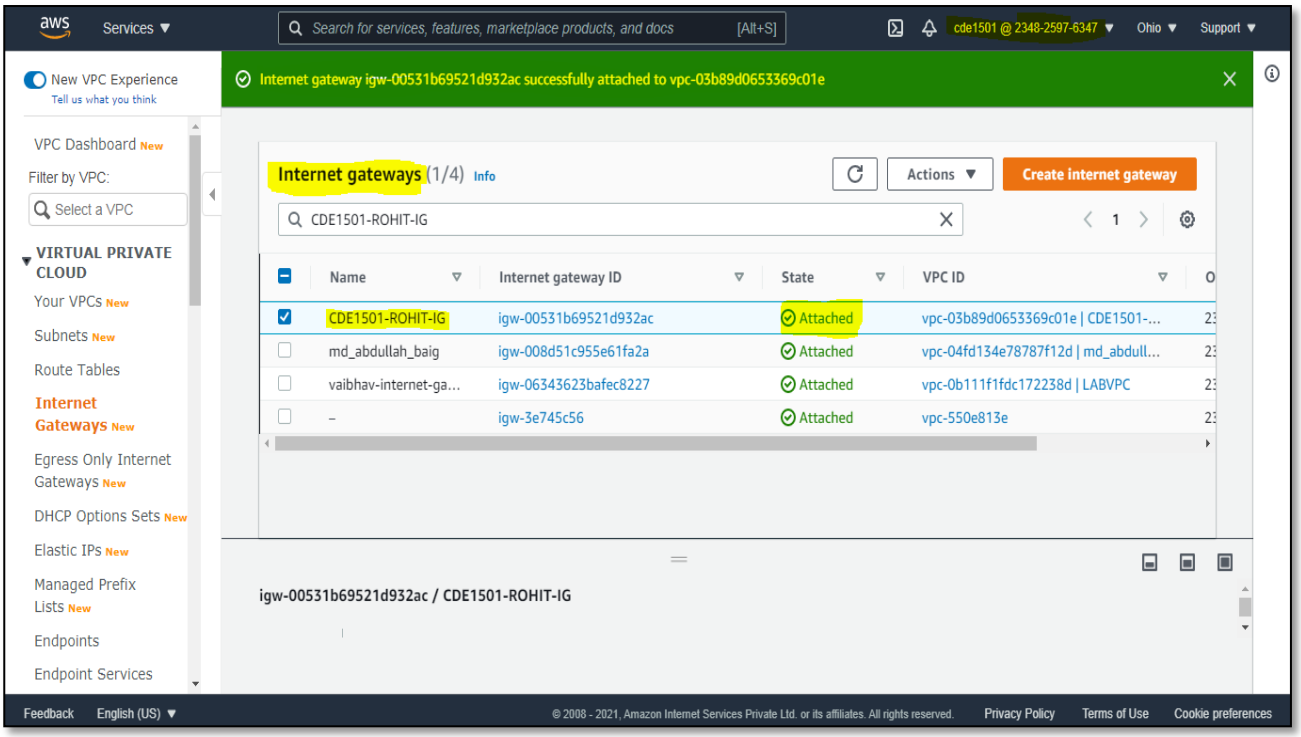
We can see above screen that VPC is created



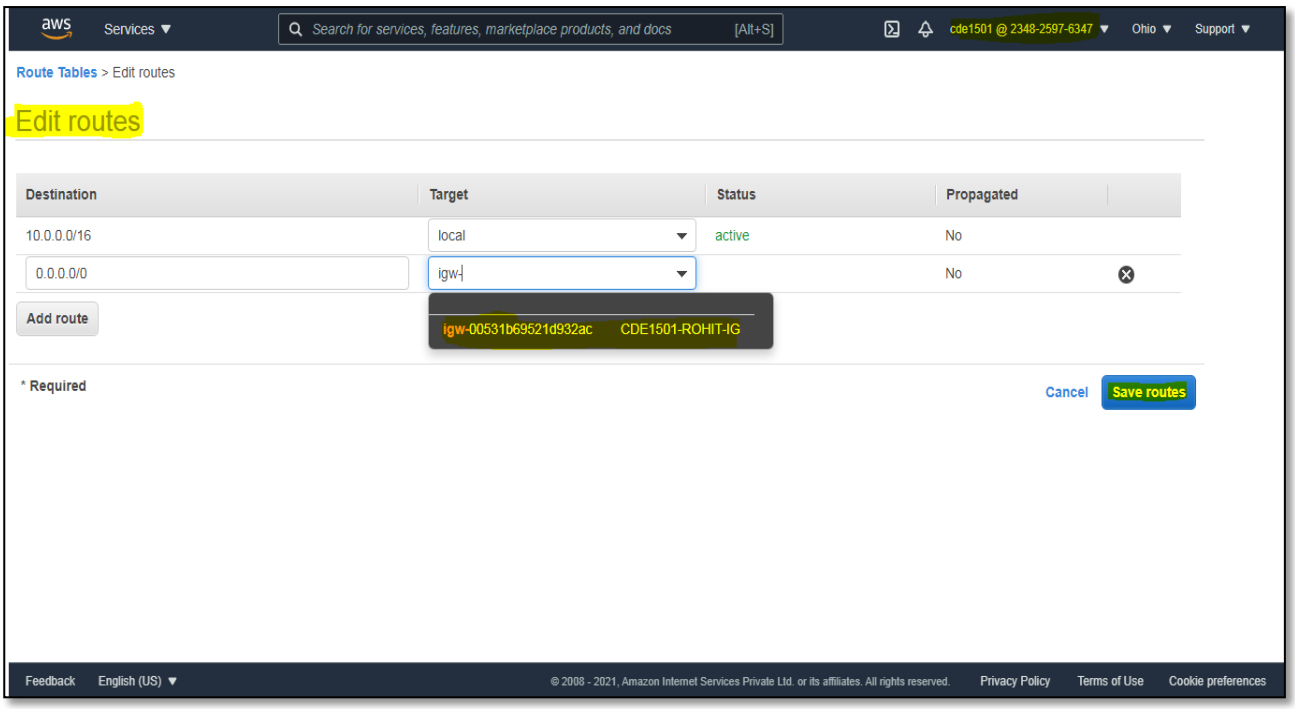
We can see above screen that INTERNET GATEWAYS is created



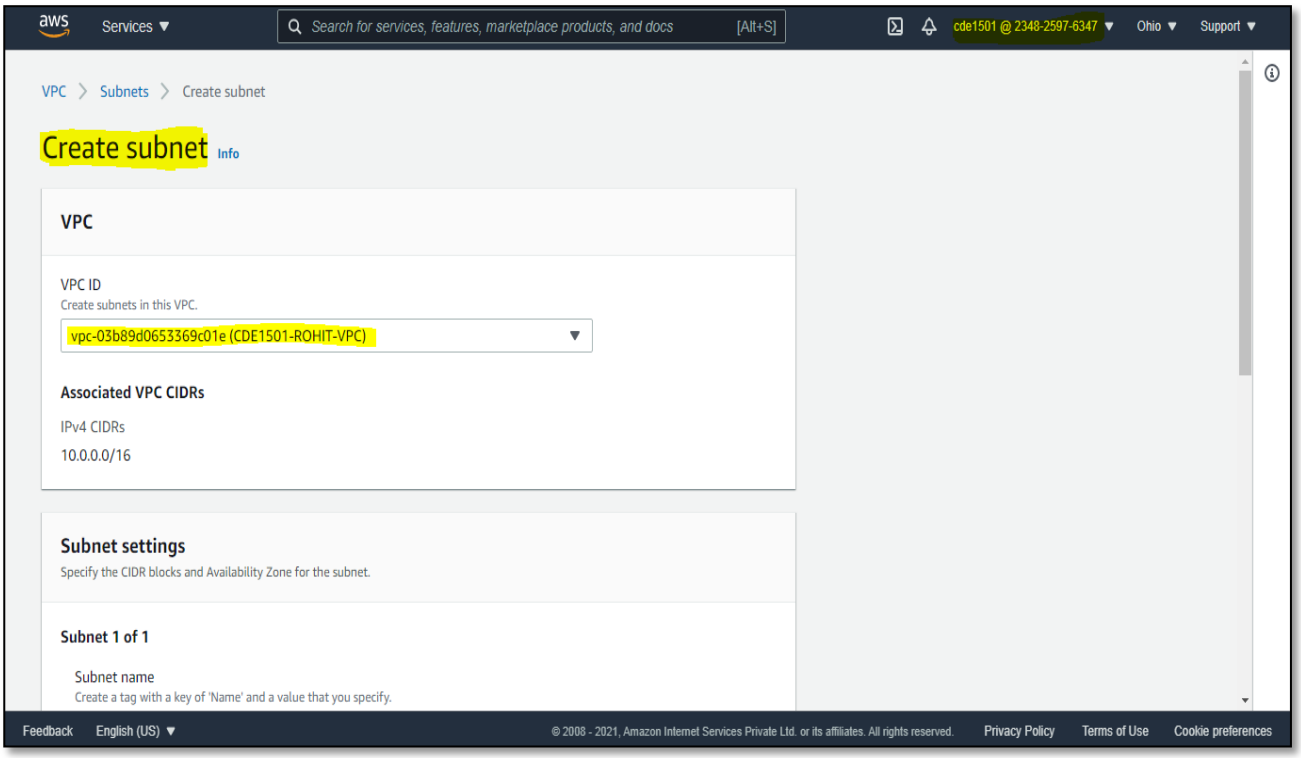
Attach to VPC



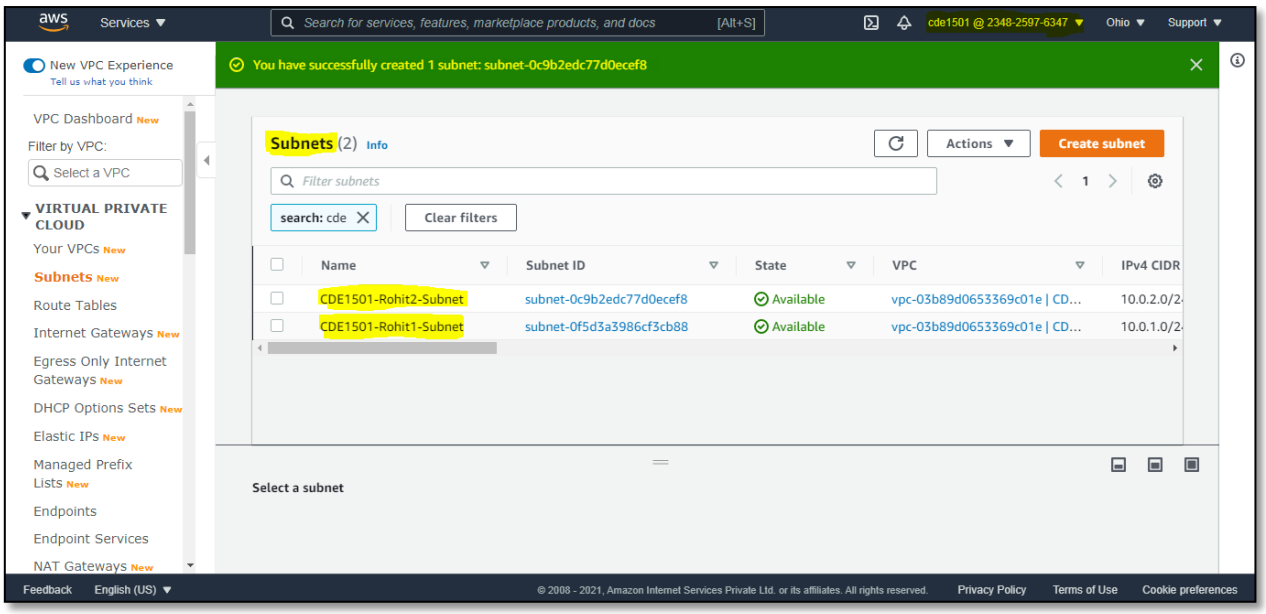
Internet gateway is successfully attached to VPC



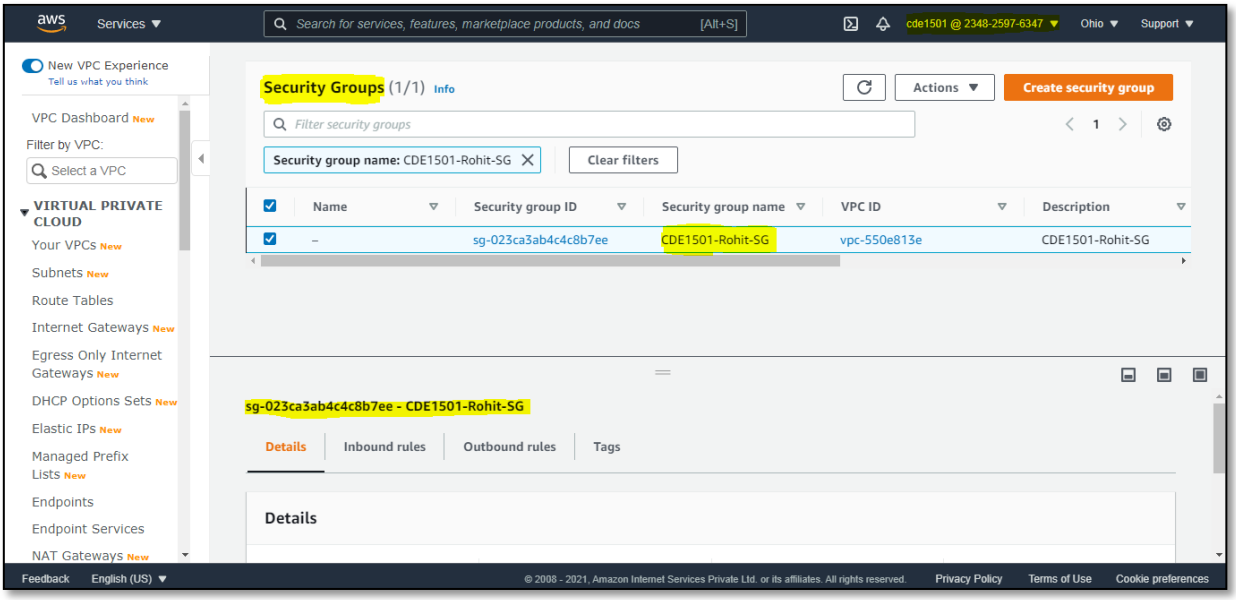
We can see above screen that ROUTES is created



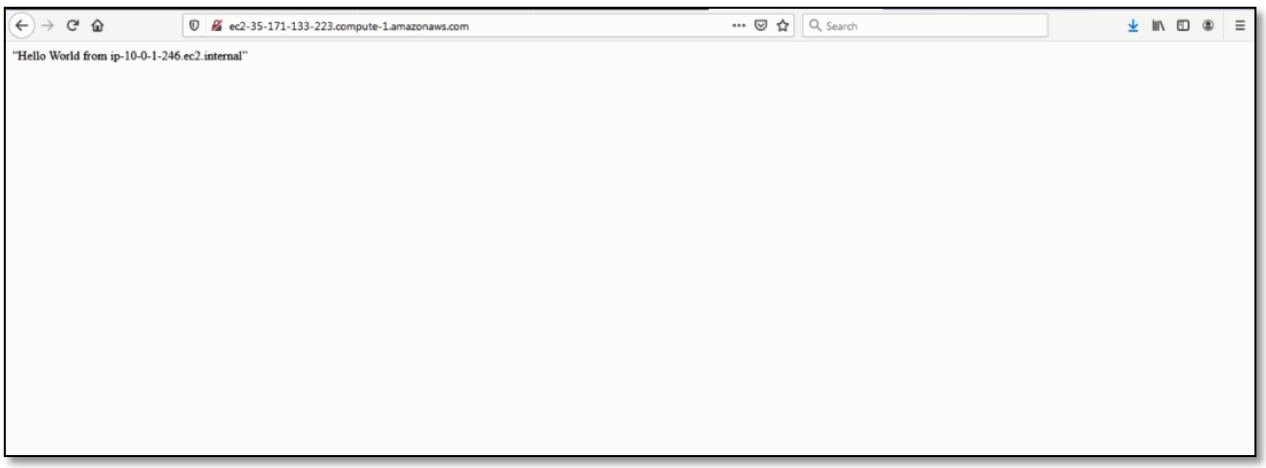
Create Subnet



We can see above screen that SUBNETS is created



We can see above screen that SECURITY GROUPS is created

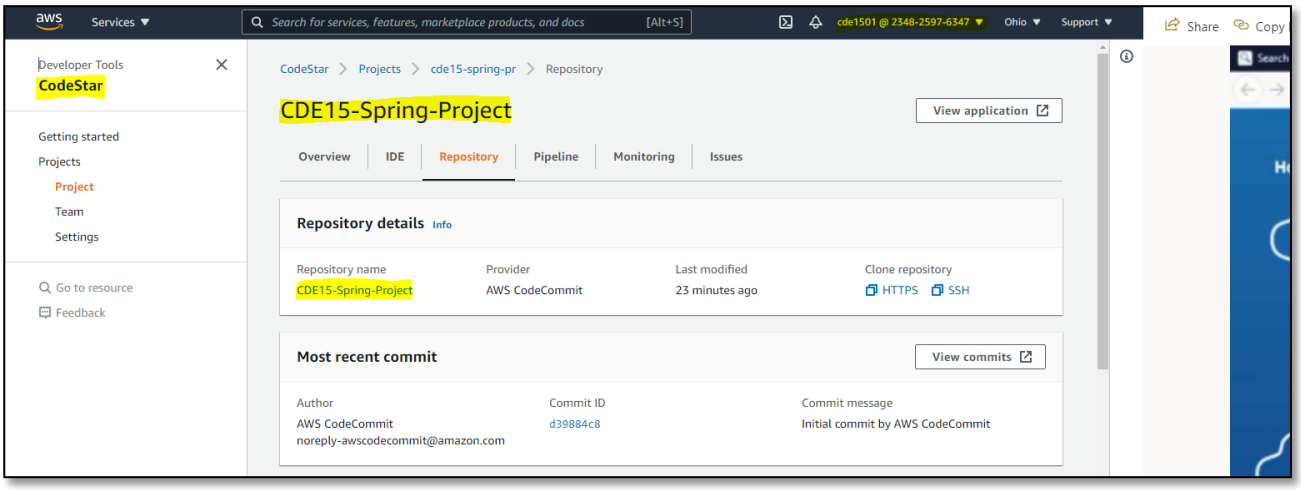


We can see above screen that browser is display HELLO WORLD

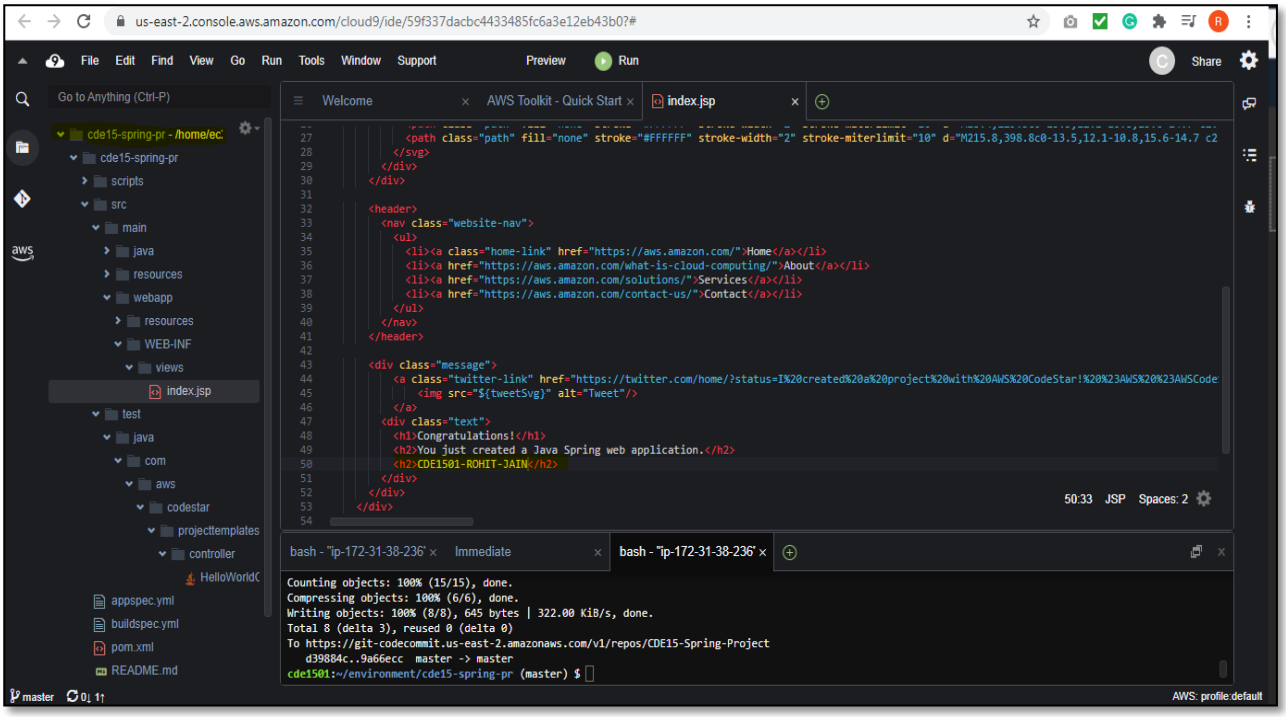
DAY-THREE

- **[CCID-lab-hands-on-practise]**

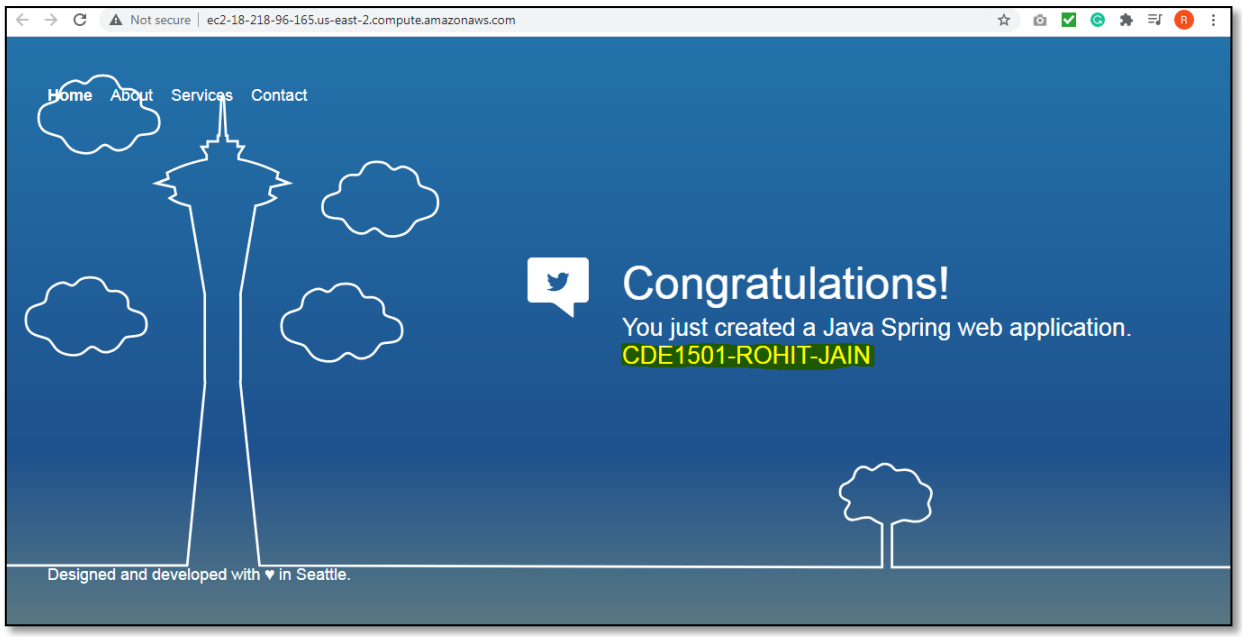
We will able to deploy a spring web application using a Continuous Integration (CI)/ Continuous Delivery (CD) pipeline and the IDE provided by AWS.



We can see above screen that SPRING PROJECT is created



AWS IDE

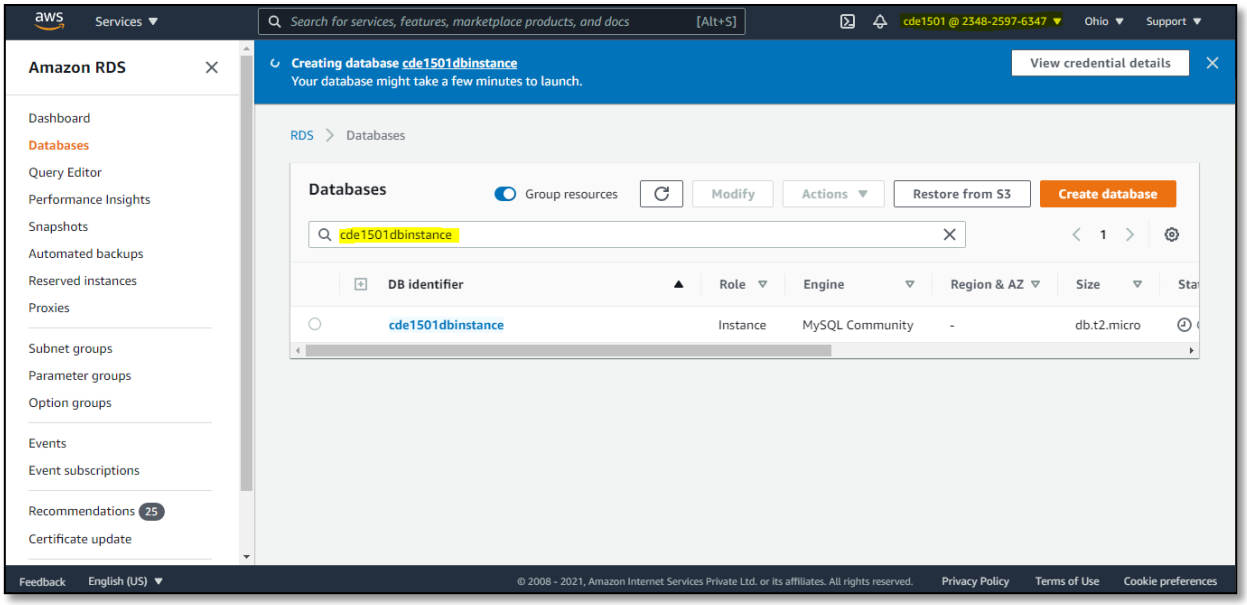


Successfully deploy a spring web application using CI/CD

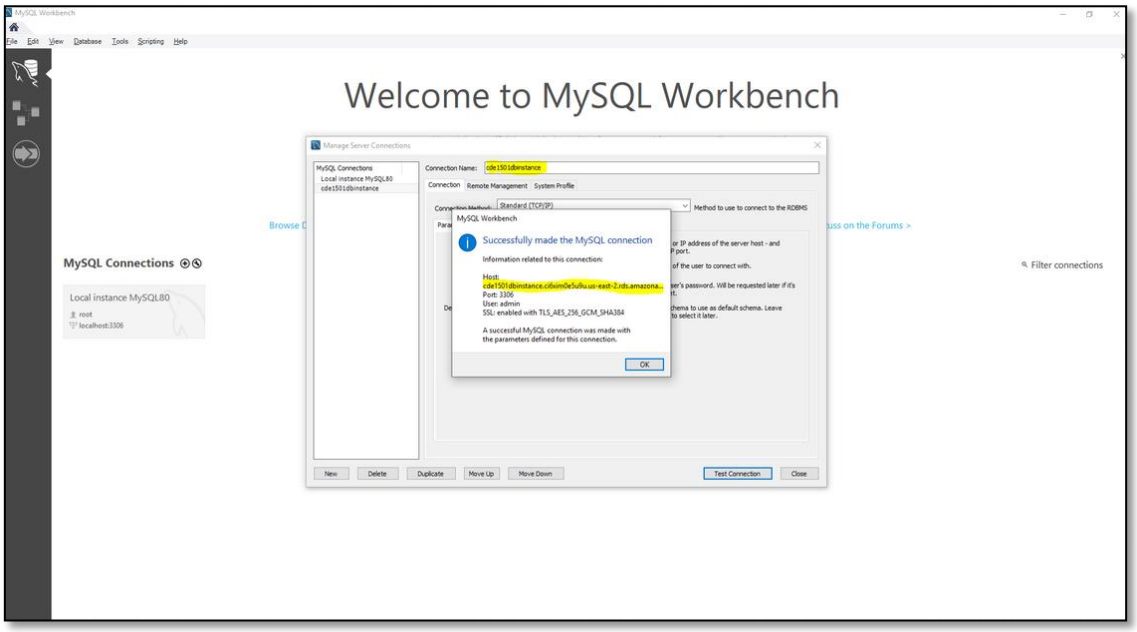
DAY-FOUR

▪ [Spring-REST-with-RDS-Backend]

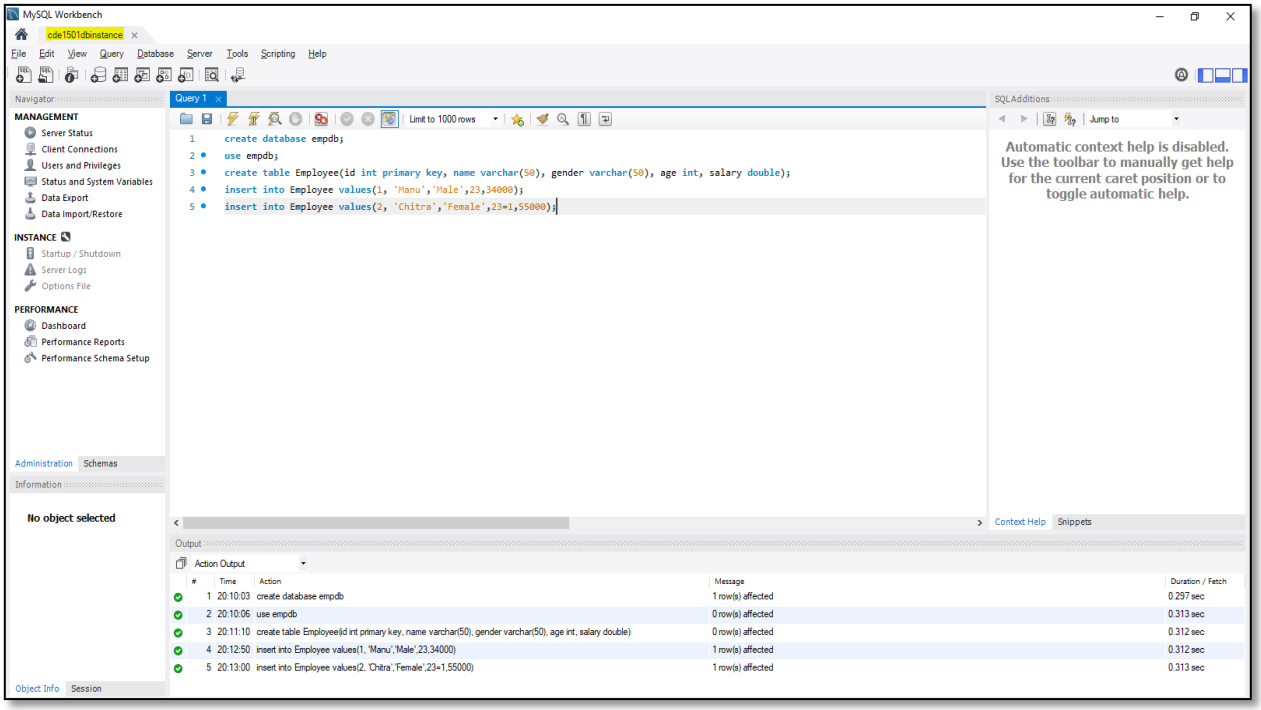
Create a Spring REST application that perform Read and Insert operation on RDS database. Deploy the application in AWS Elastic Beanstalk and access the application from anywhere.



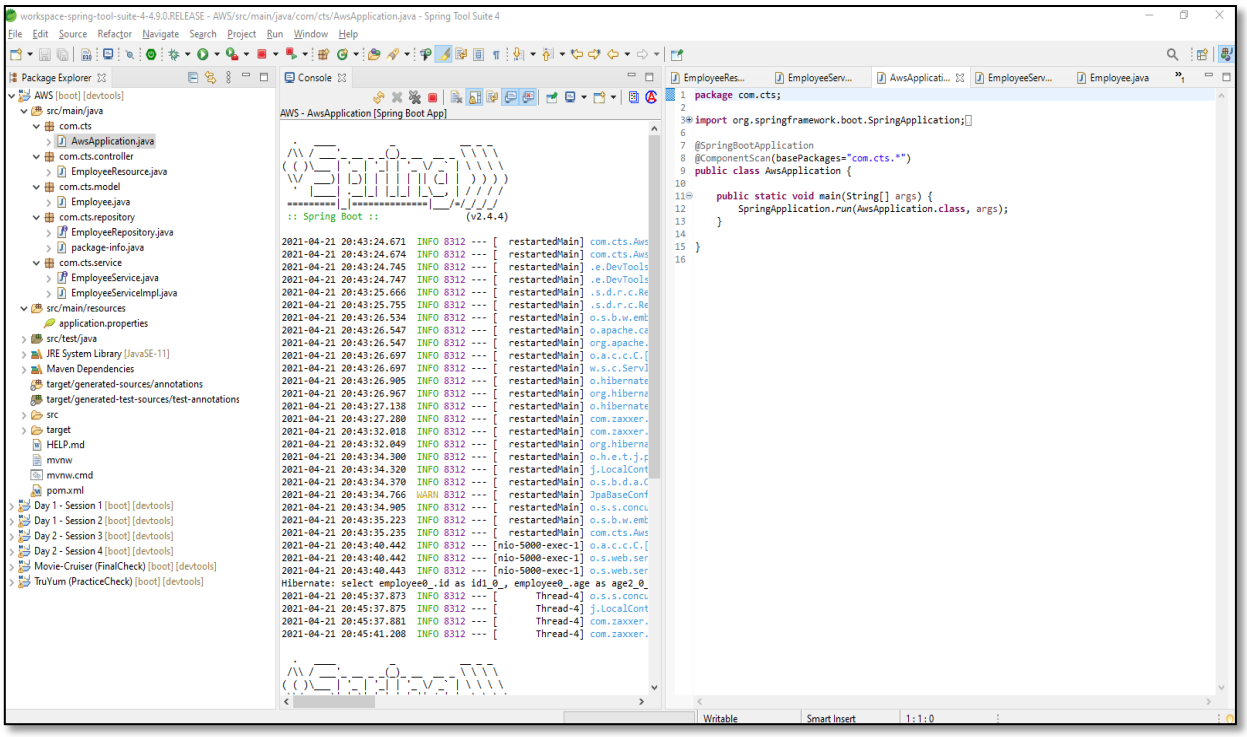
We can see above screen that our database is being created



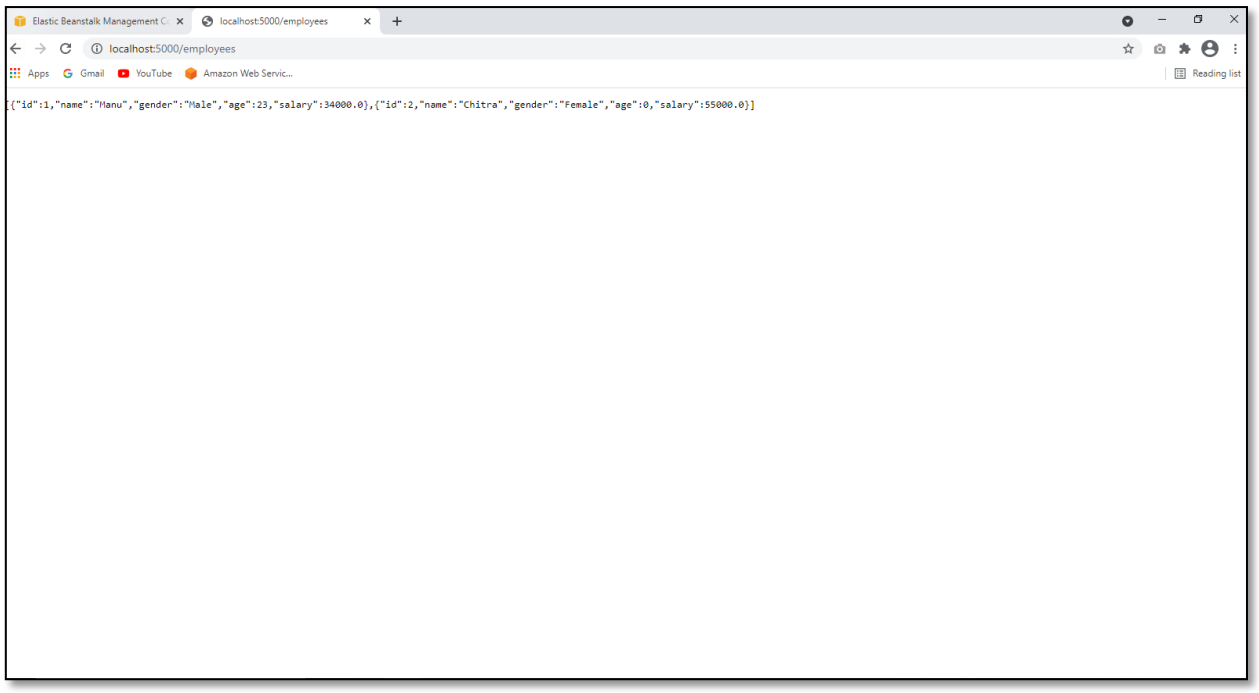
We will be connected to the RDS MySQL Server



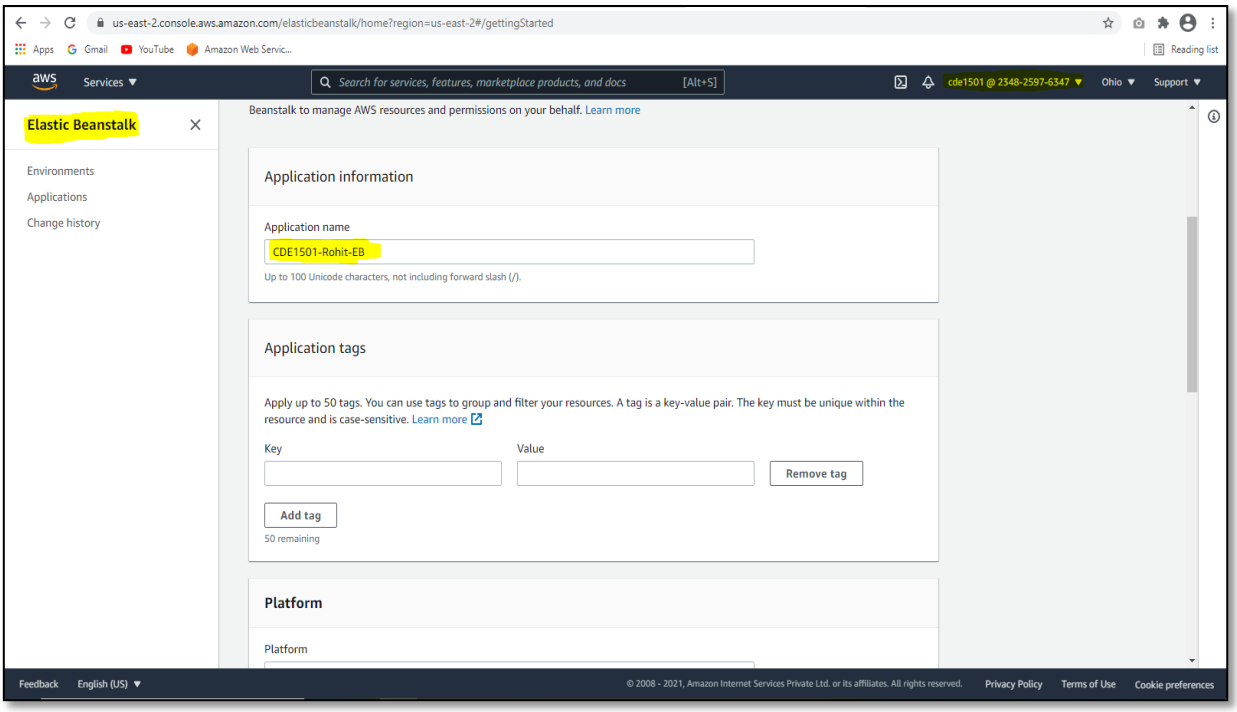
Enter the SQL commands shown in the screen above



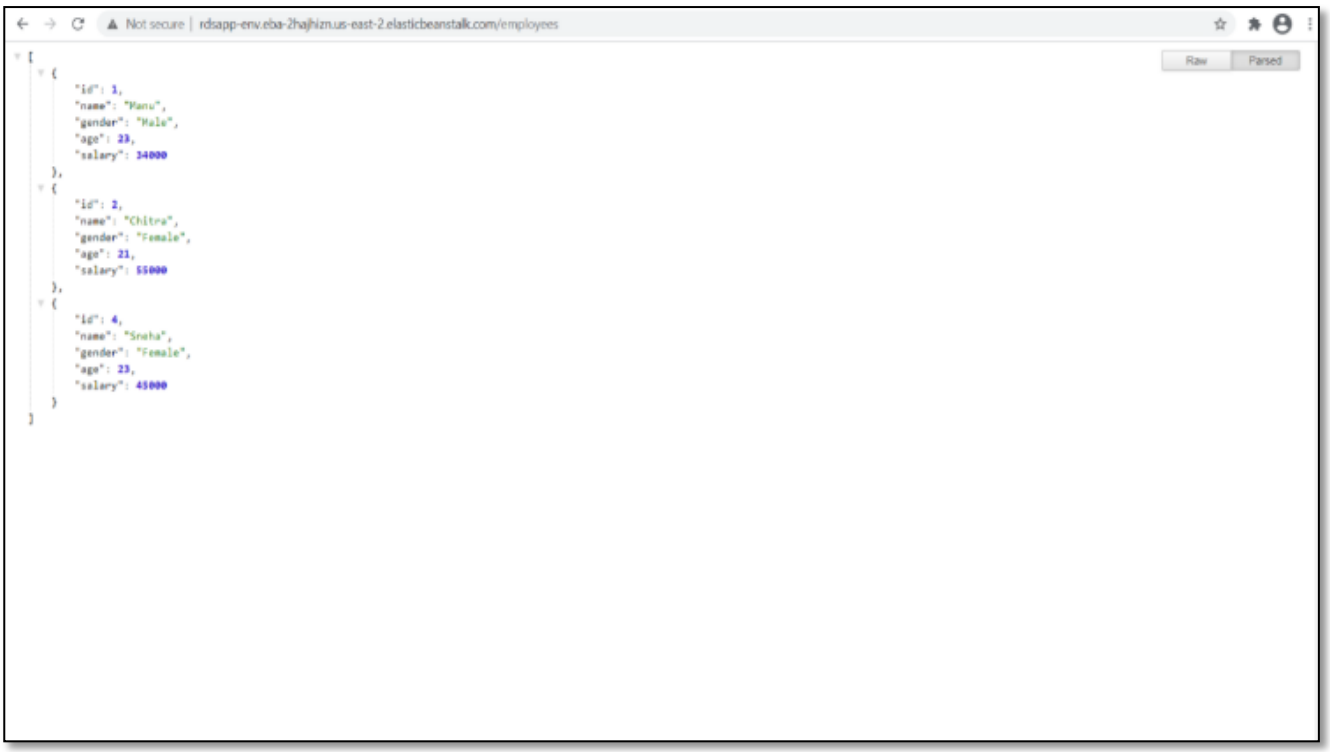
Create a spring REST application using Spring Boot.



Start the application locally and type the url in the browser window as shown above



We can see above screen that ELASTIC BEANSTALK is created

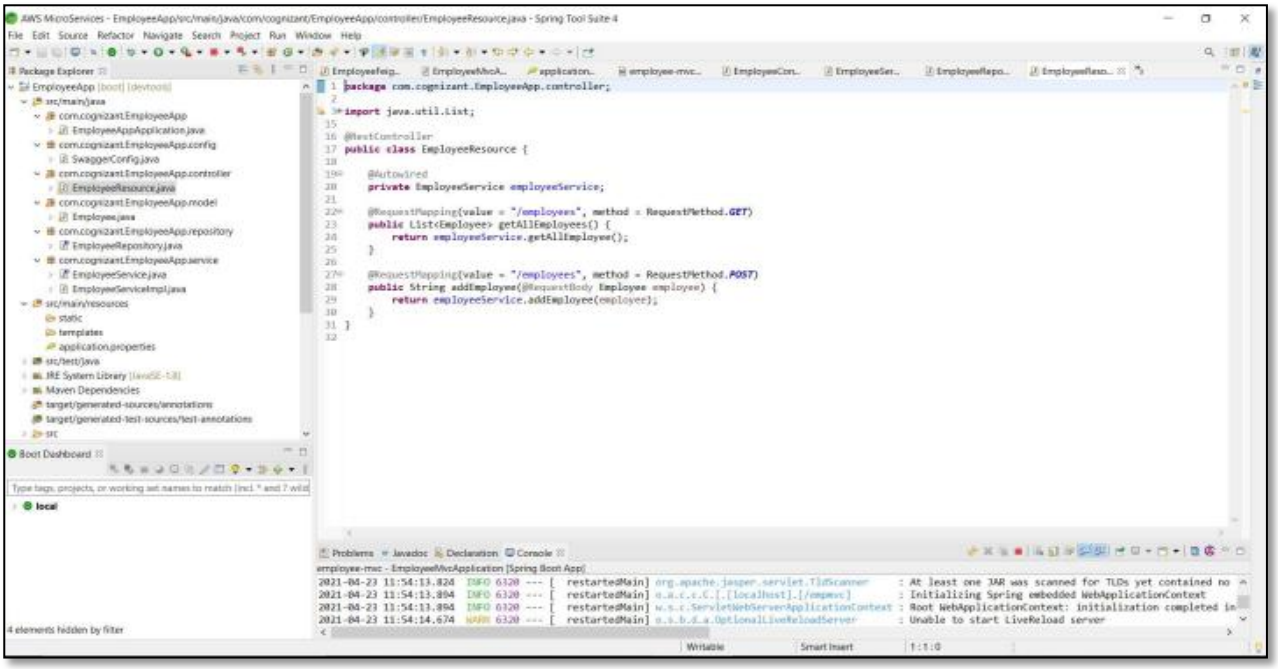


We can see that the record has been successfully inserted into the RDS database

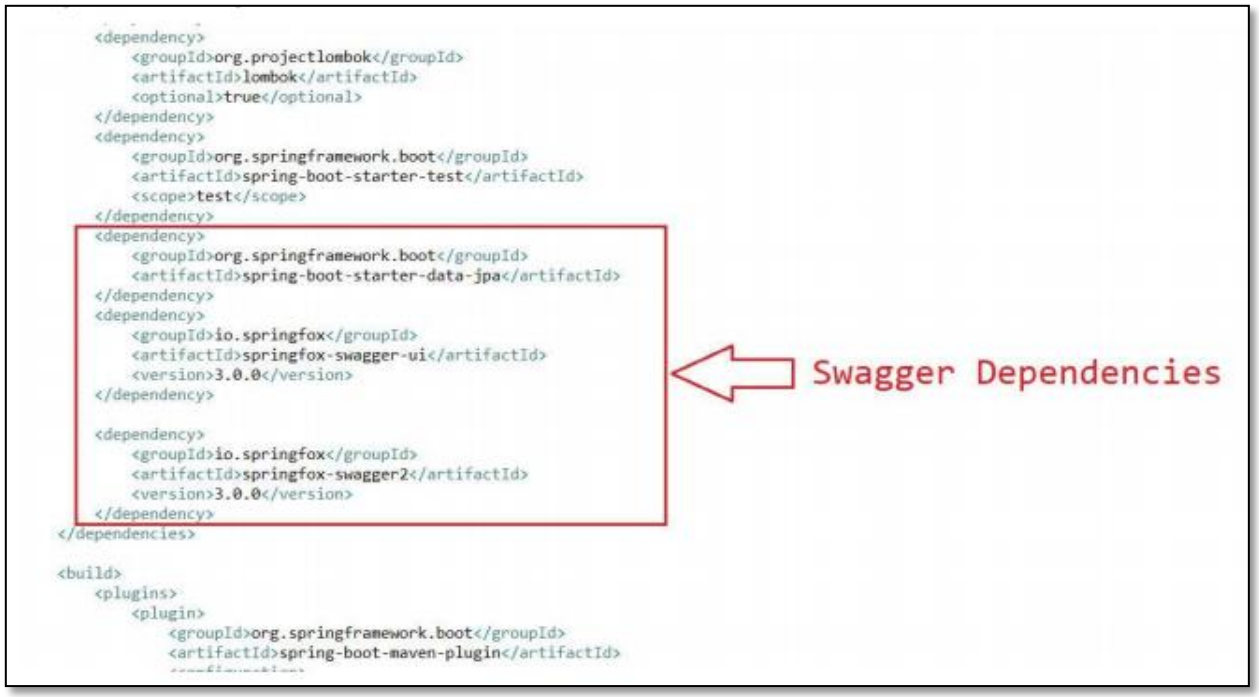
DAY-FIVE

▪ [Swagger-Hands-on]

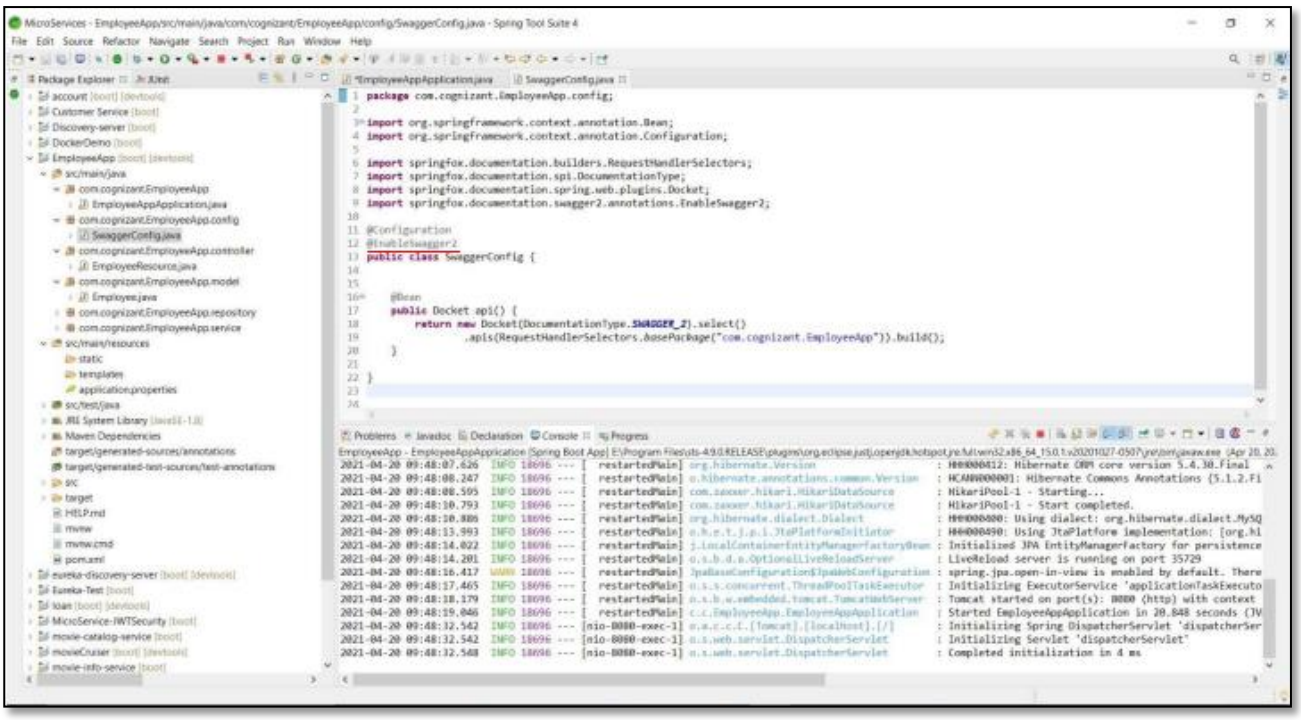
Make use of Swagger to create documentation for RESTful/microservices.



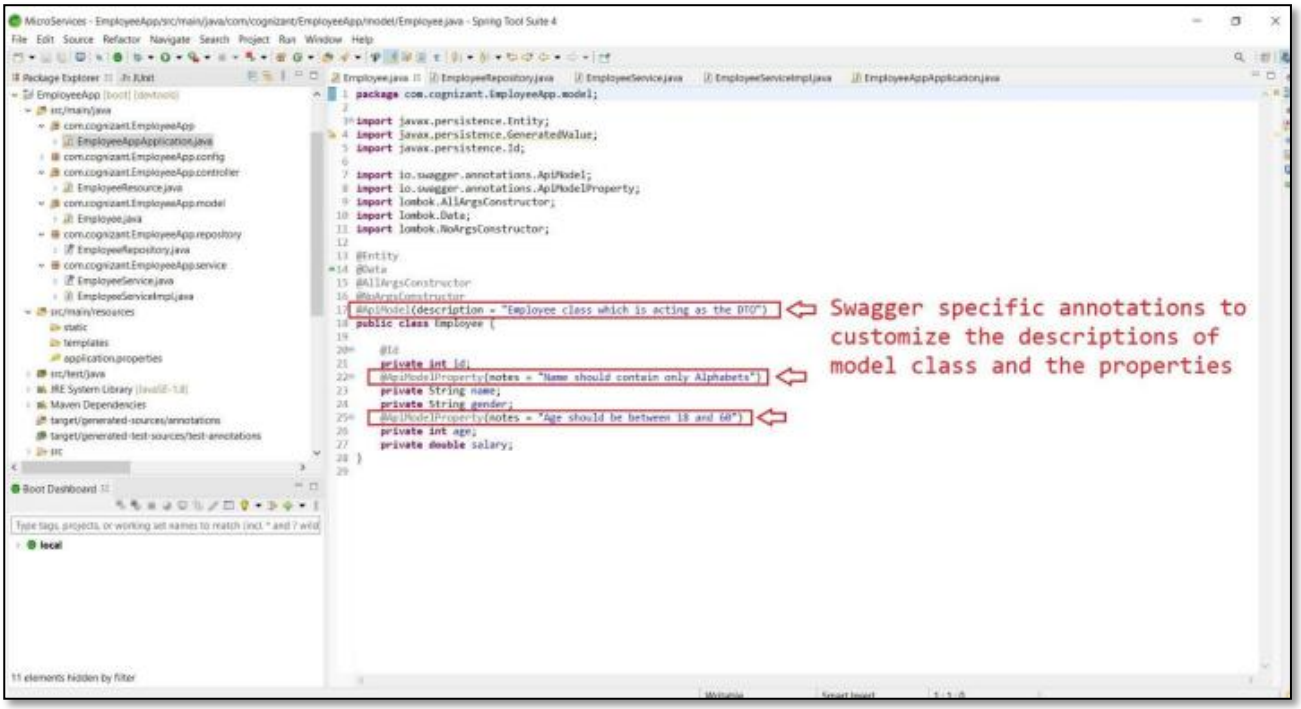
Create a simple RESTful service using Spring BOOT



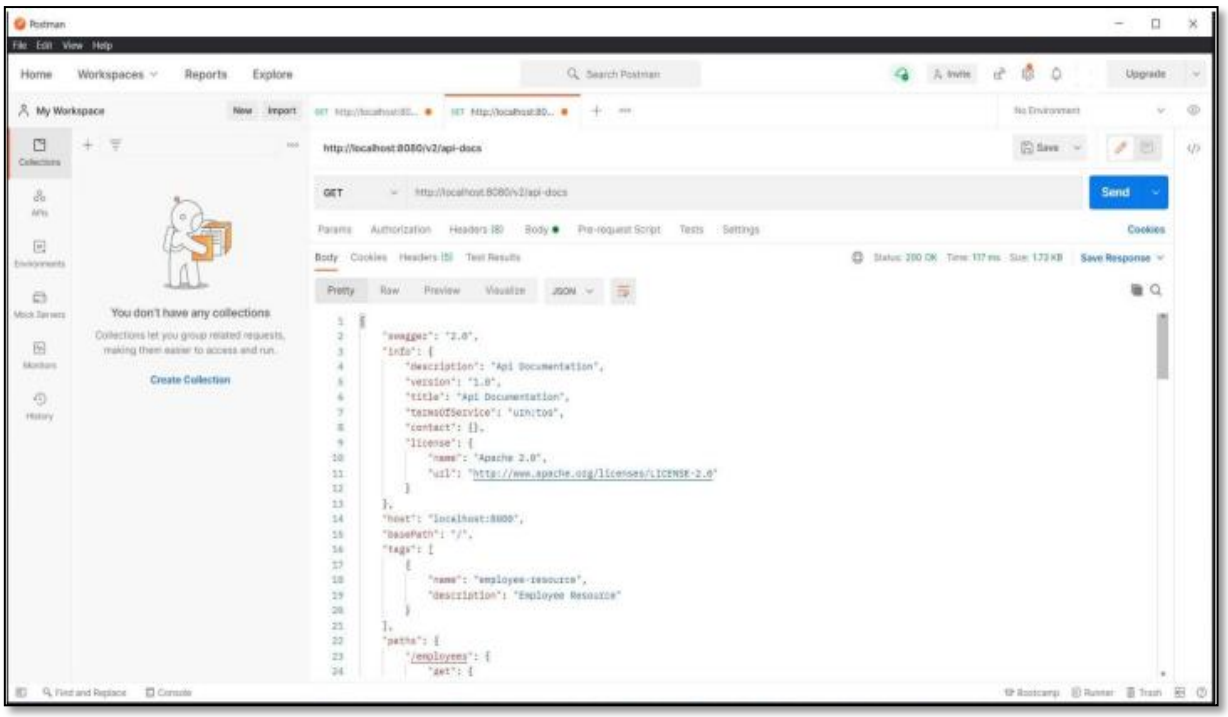
Include the Swagger dependencies



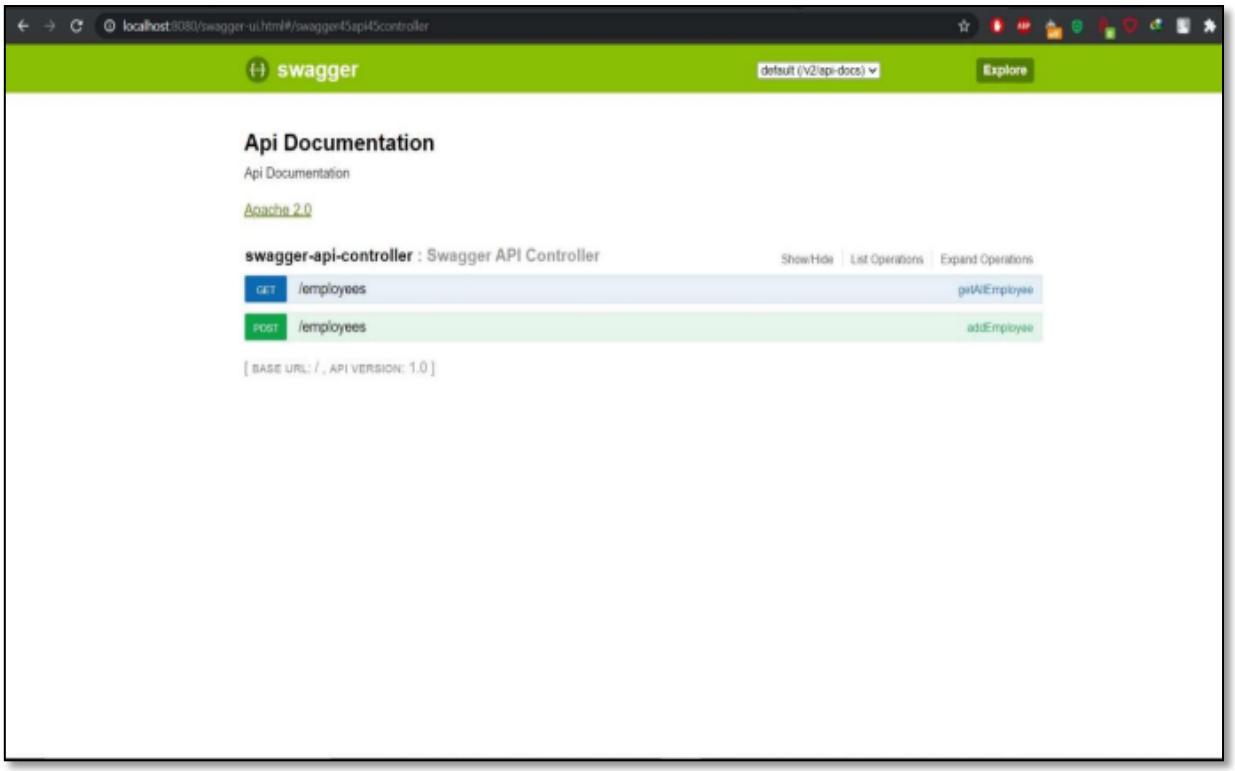
Create a Swagger configuration class as shown above



Use Swagger specific annotations to customize the descriptions of model class and the properties



Visit the api endpoint "localhost:8080/v2/api-docs" and we can see the complete API documentation of our service.

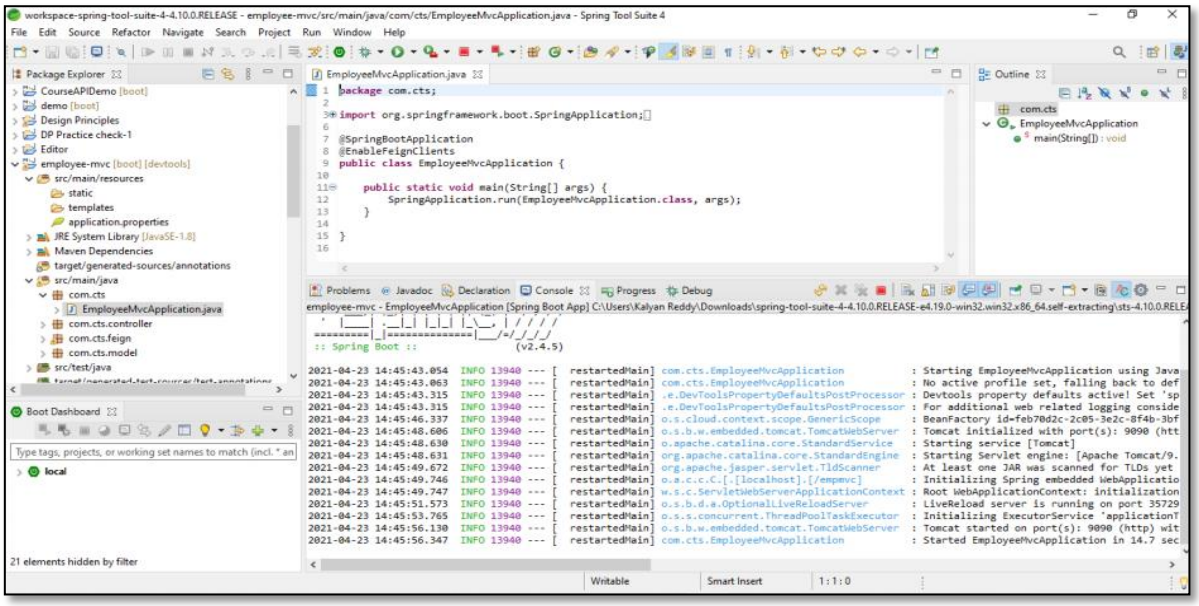


Hit the URL in our web browser and see the Swagger API functionalities

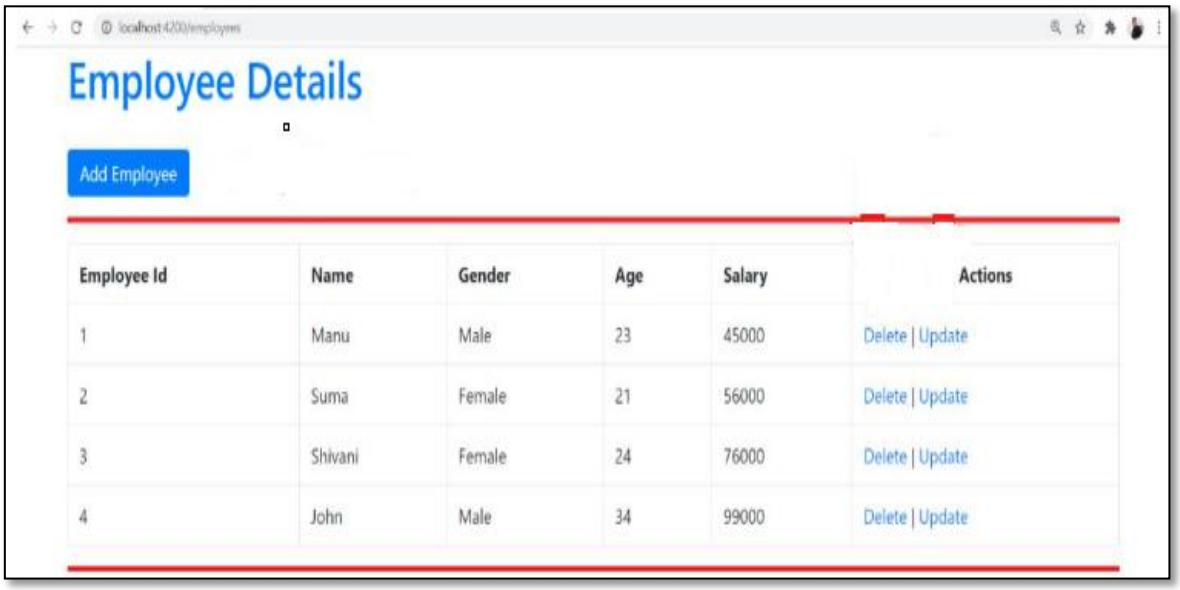
DAY-SIX

▪ [Spring MVC Client For Spring REST Service]

Create a Spring REST App which performs all the CRUD operations on an Employee table. Test the service using Postman.



Spring REST Application



CRUD operations on an Employee table

Employee record updated successfully

Employee record deleted successfully