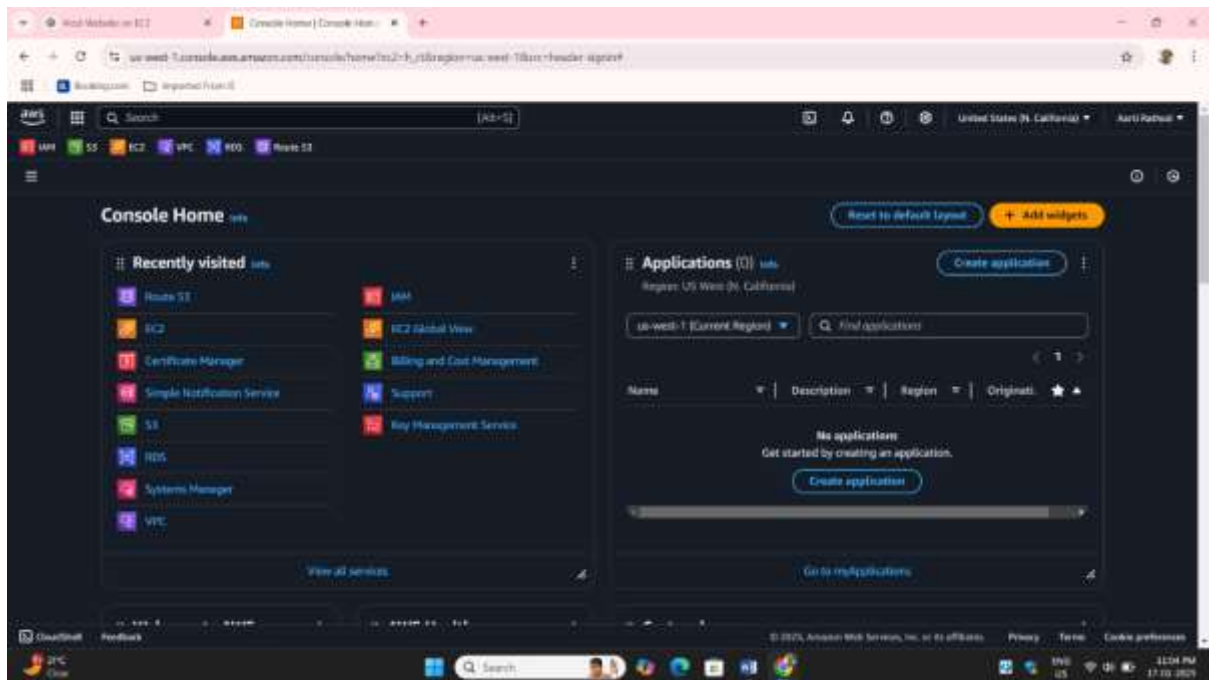
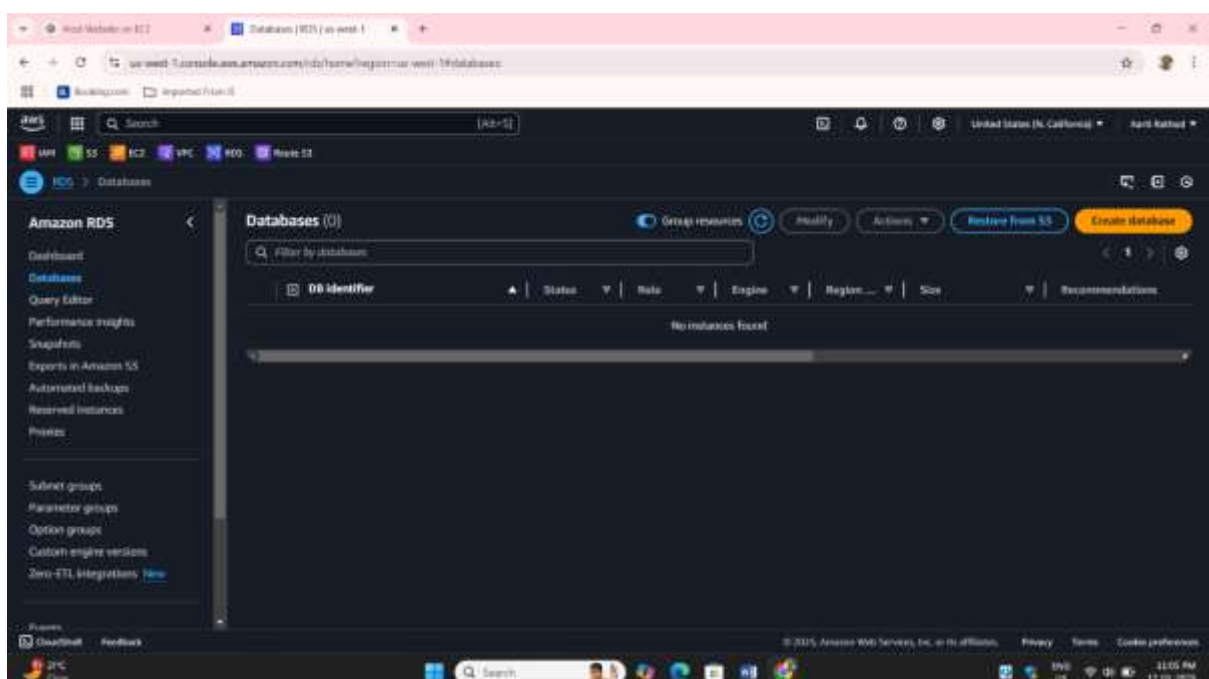


Successfully deployed a 2-tier application on AWS Set up an RDS database, configured Tomcat on an EC2 instance, and connected the application to the database.

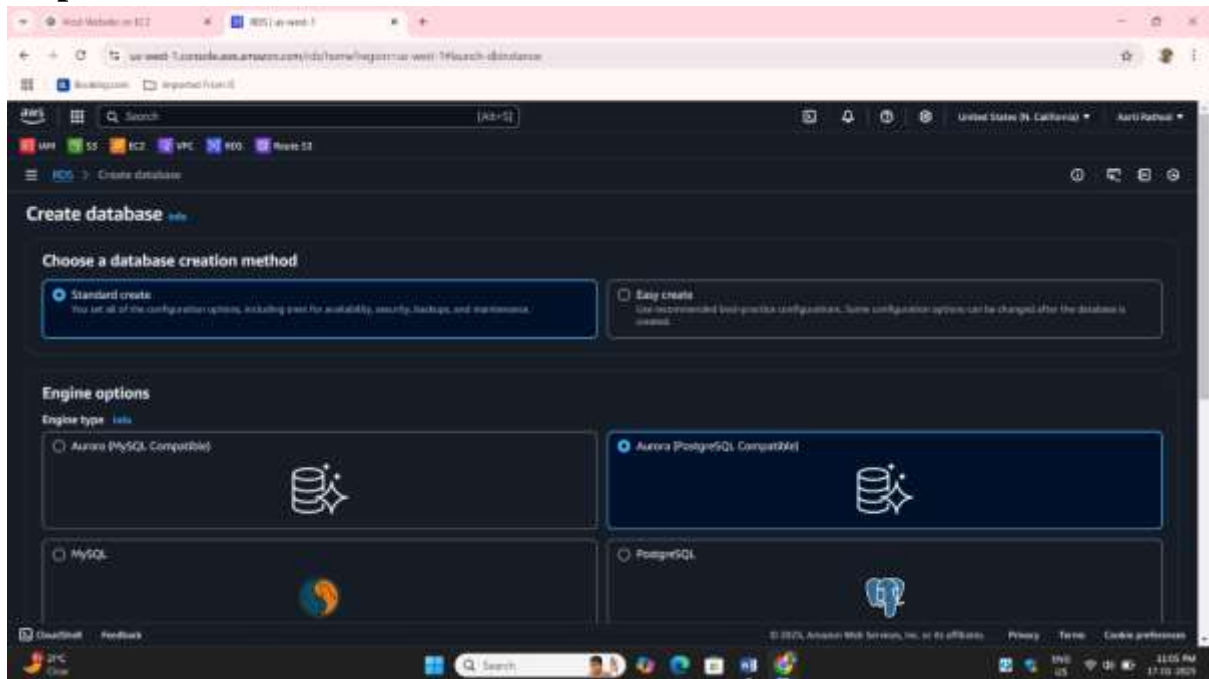
**Step1: Goto AWS Management console go to the AWS RDS service and create a MySQL database.**



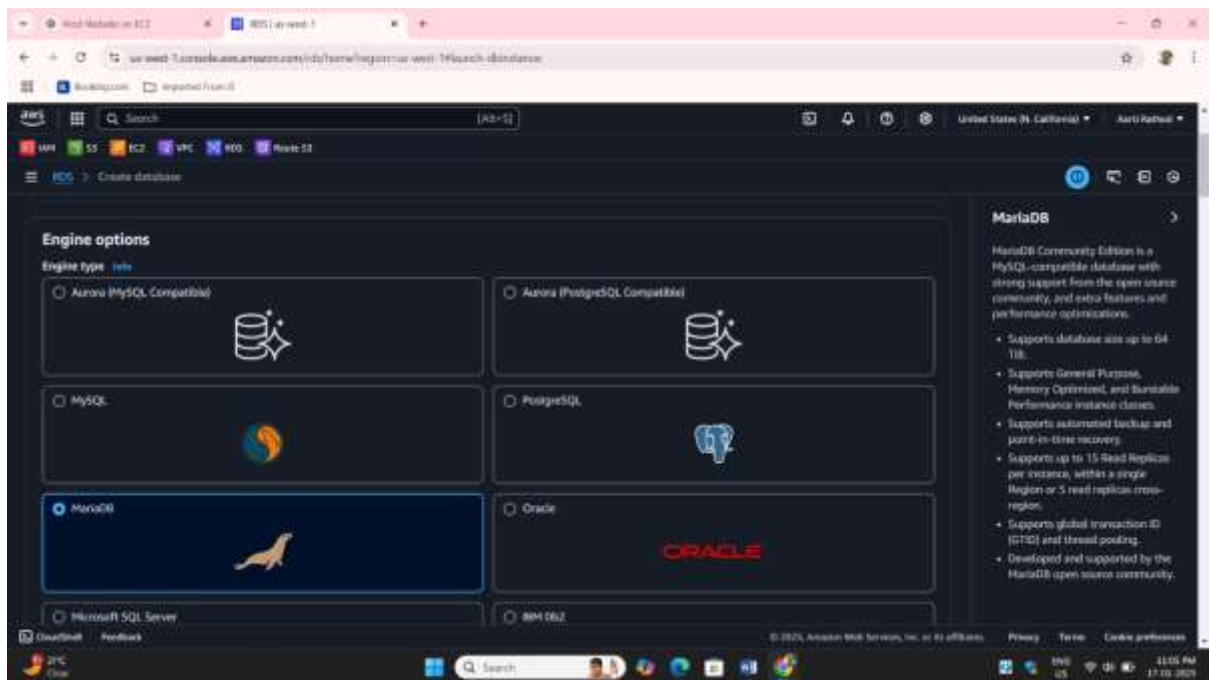
**Step 2 : Click On Create Database**



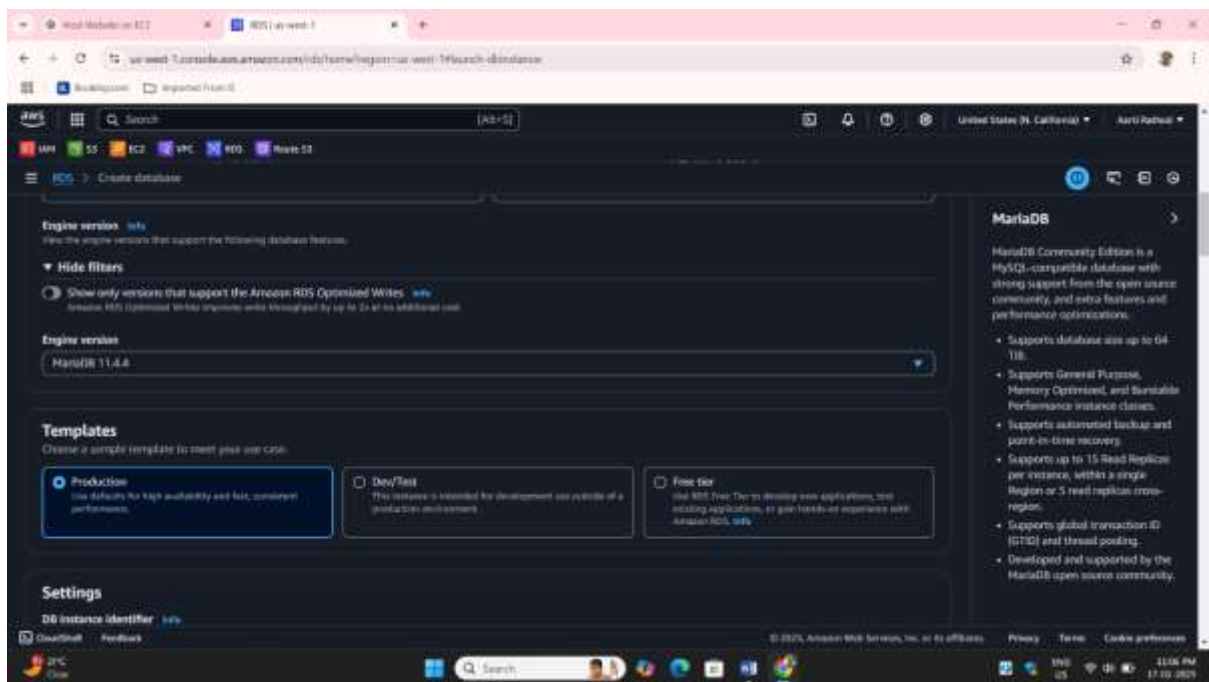
### Step 3 : Choose Standard database creation method



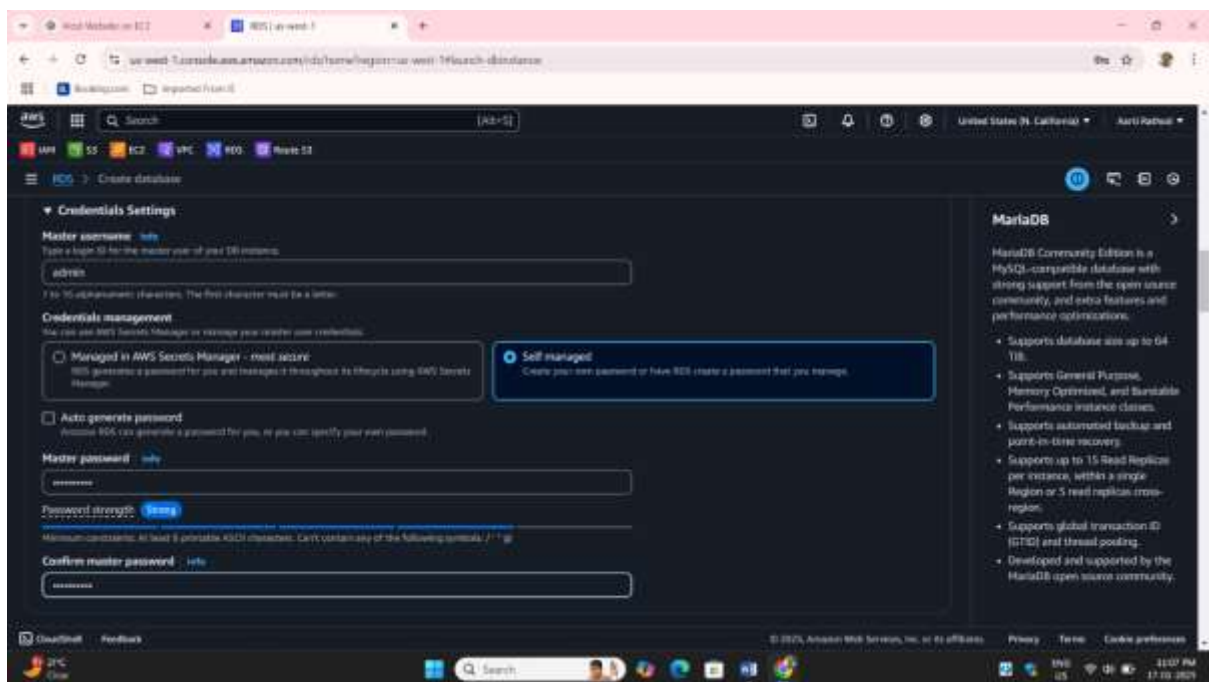
### Step 4 : Choose mariadb engine options



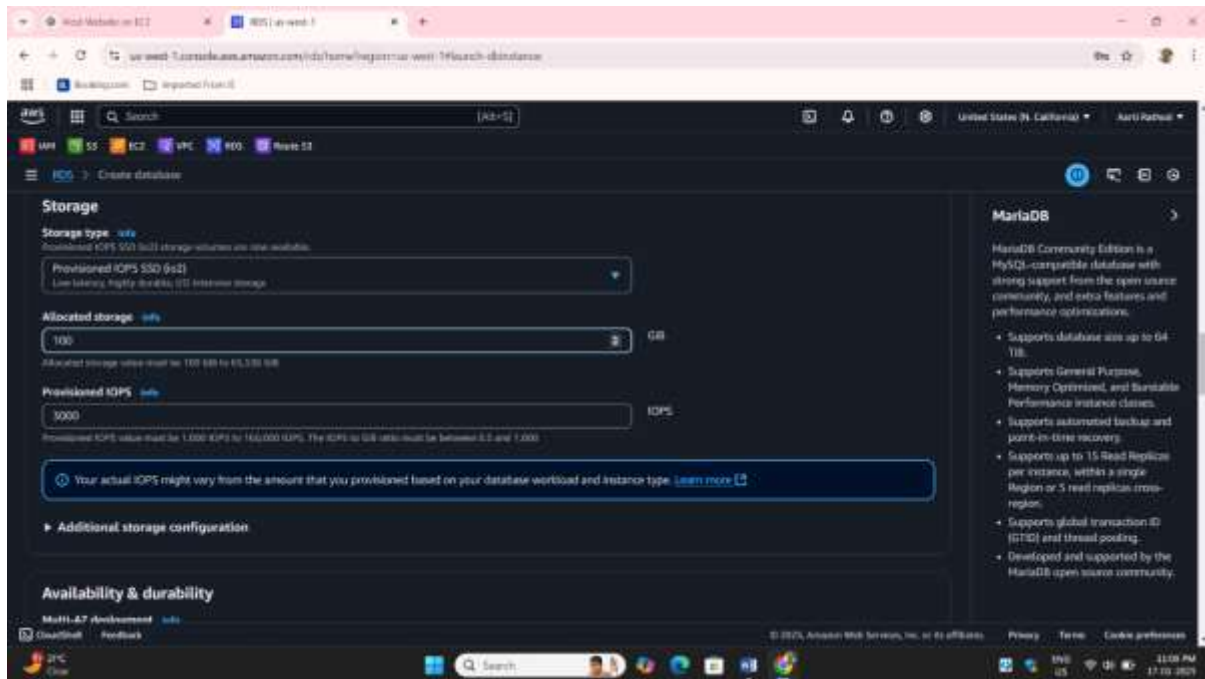
## Step 5 : Choose template Production



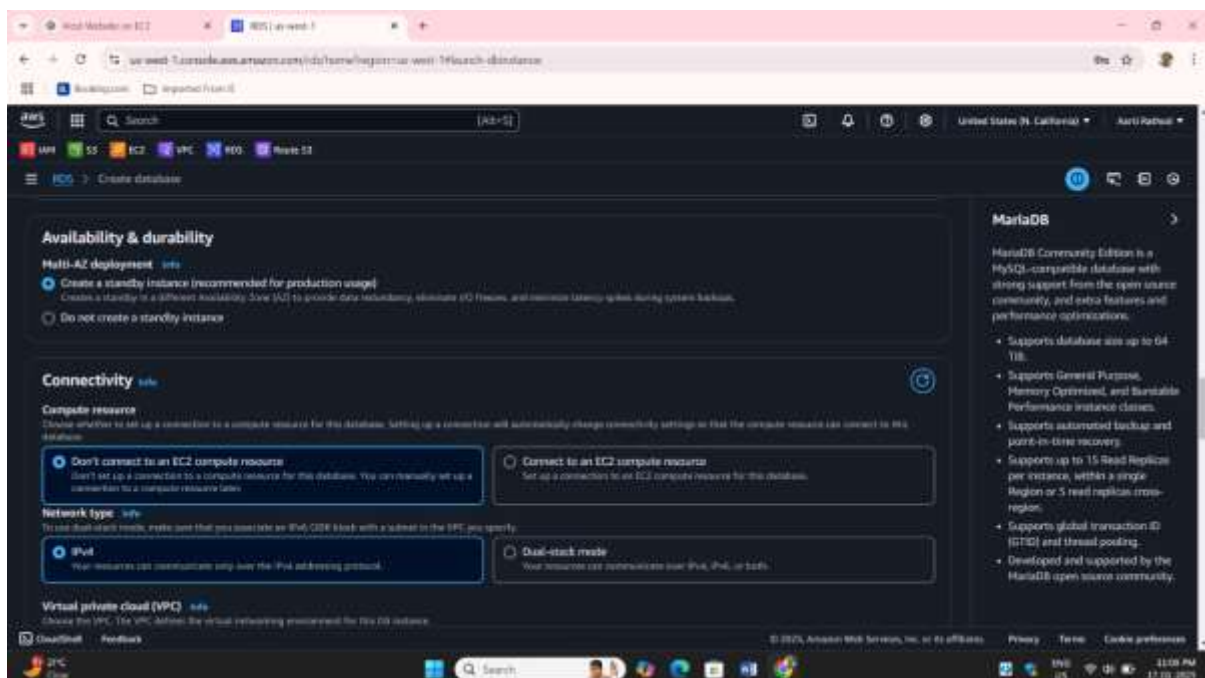
## Step 6 : Give master Username Choose self managed credentials give some strong master password



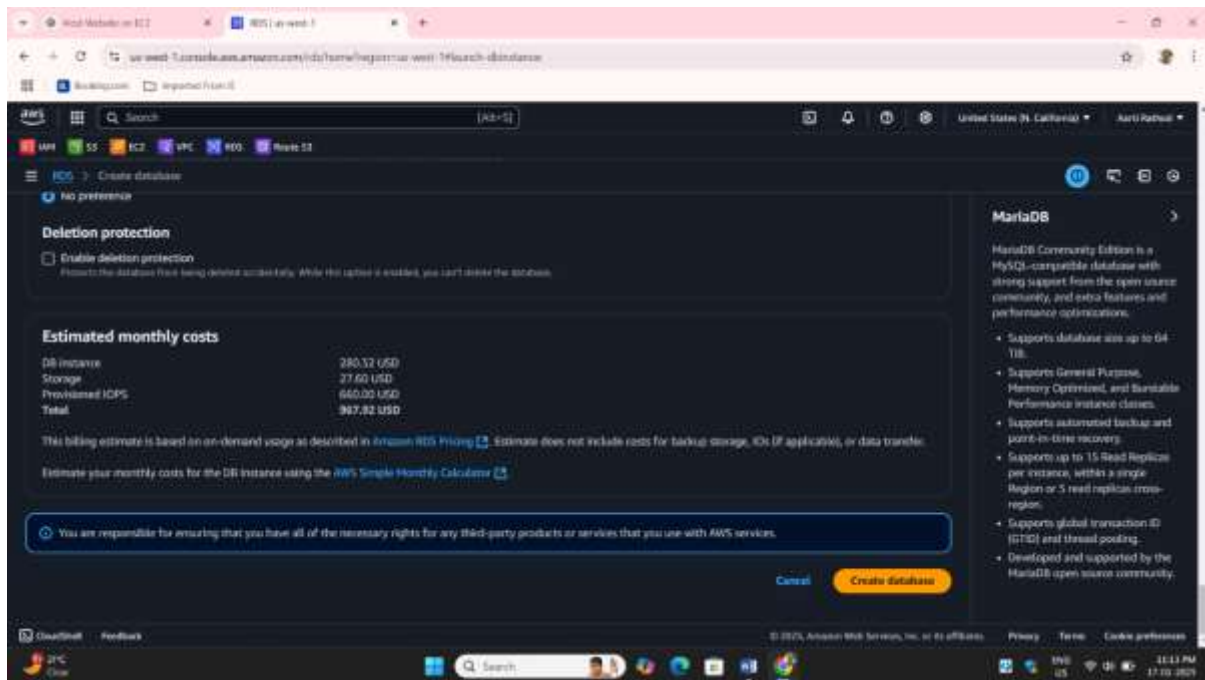
## Step 7 : Choose Storage type,Allocate Storage as much as you want give Provisioned IOPS



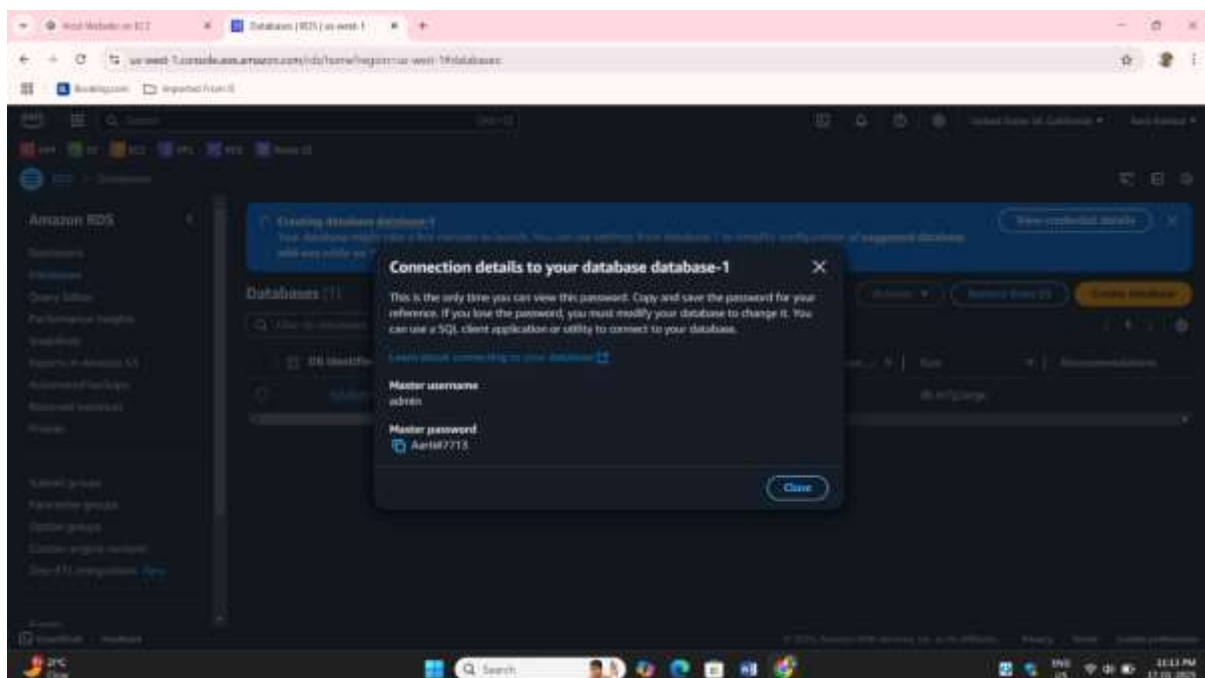
## Step 8 : Create Standby instance if you want Select don't connect to ec2 compute resource select network type IPV4



## Step 9: If you want to enable deletion protection to can tick on the box and click on create database

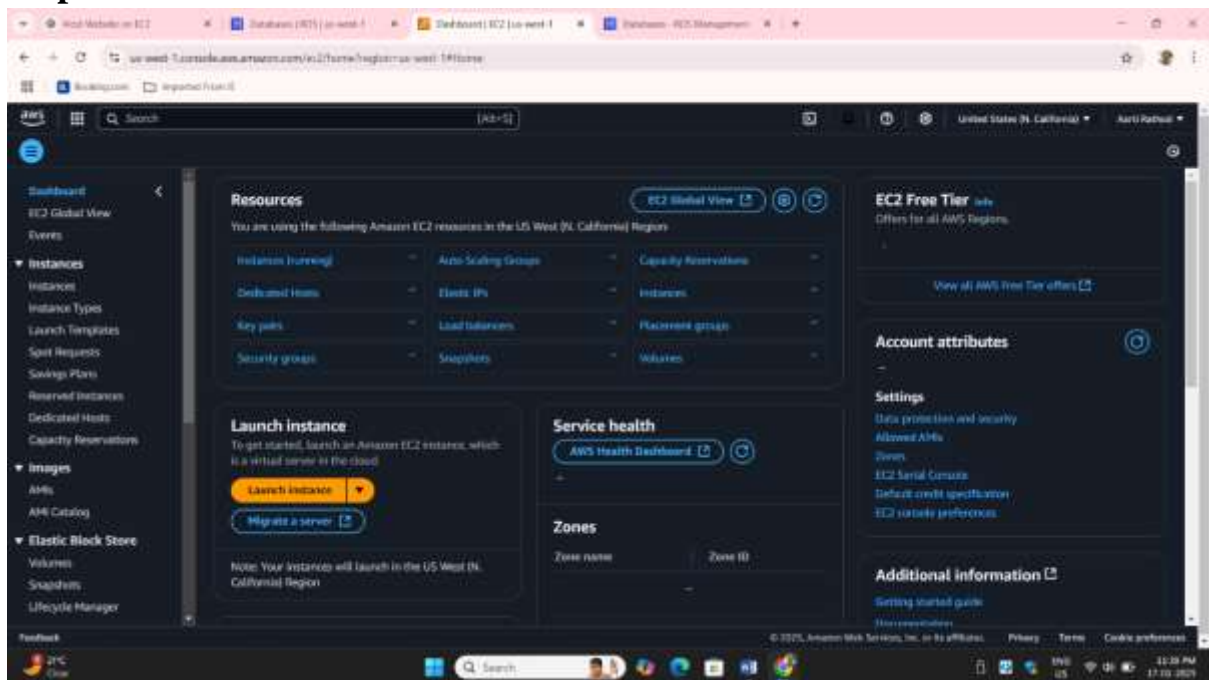


## Step 10 : Check your credentials

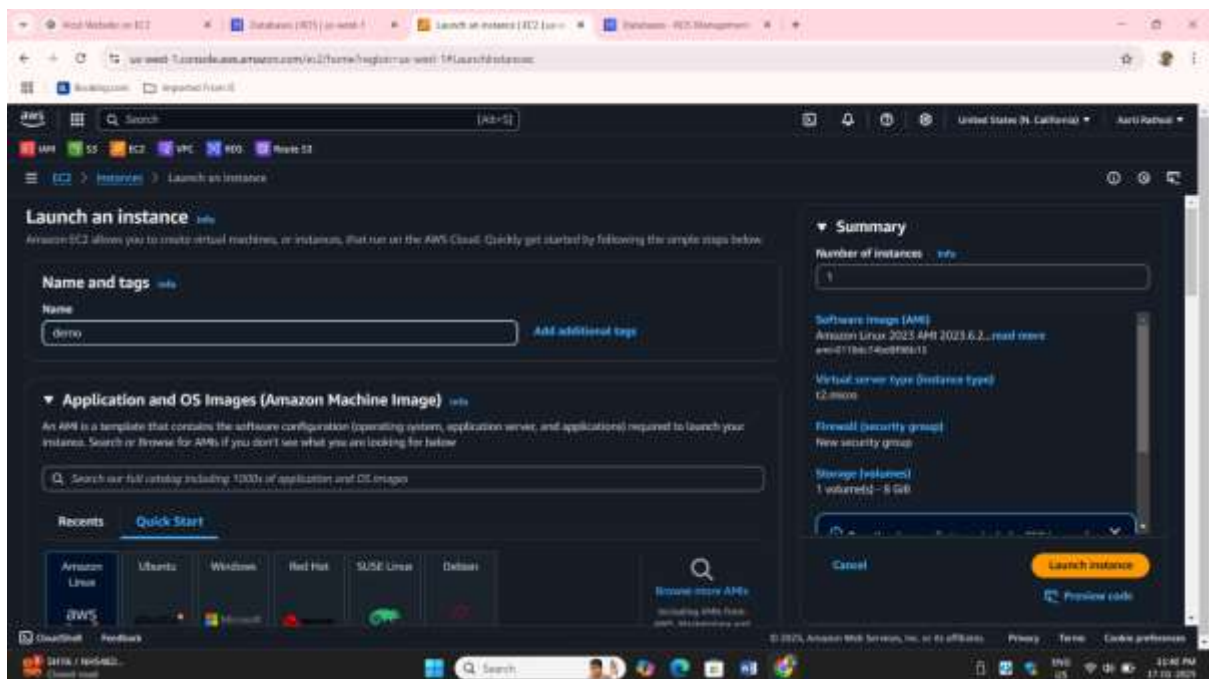




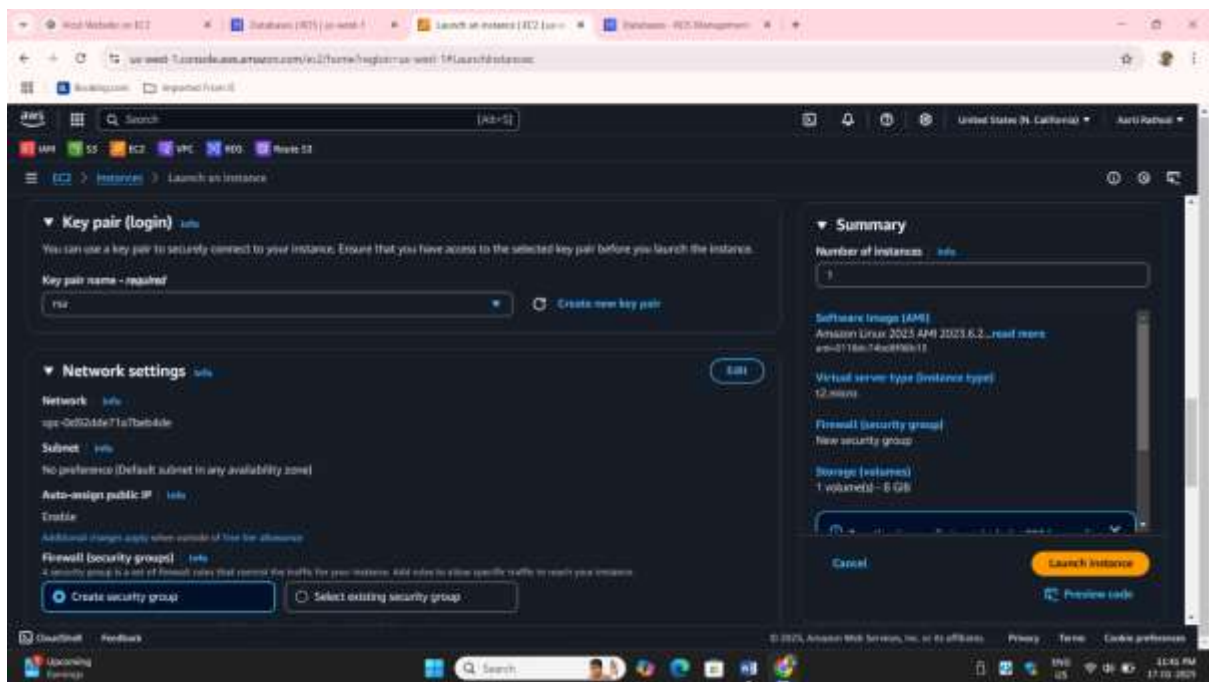
## Step 11: Go to EC2 instance click on Launch an EC2 Instance



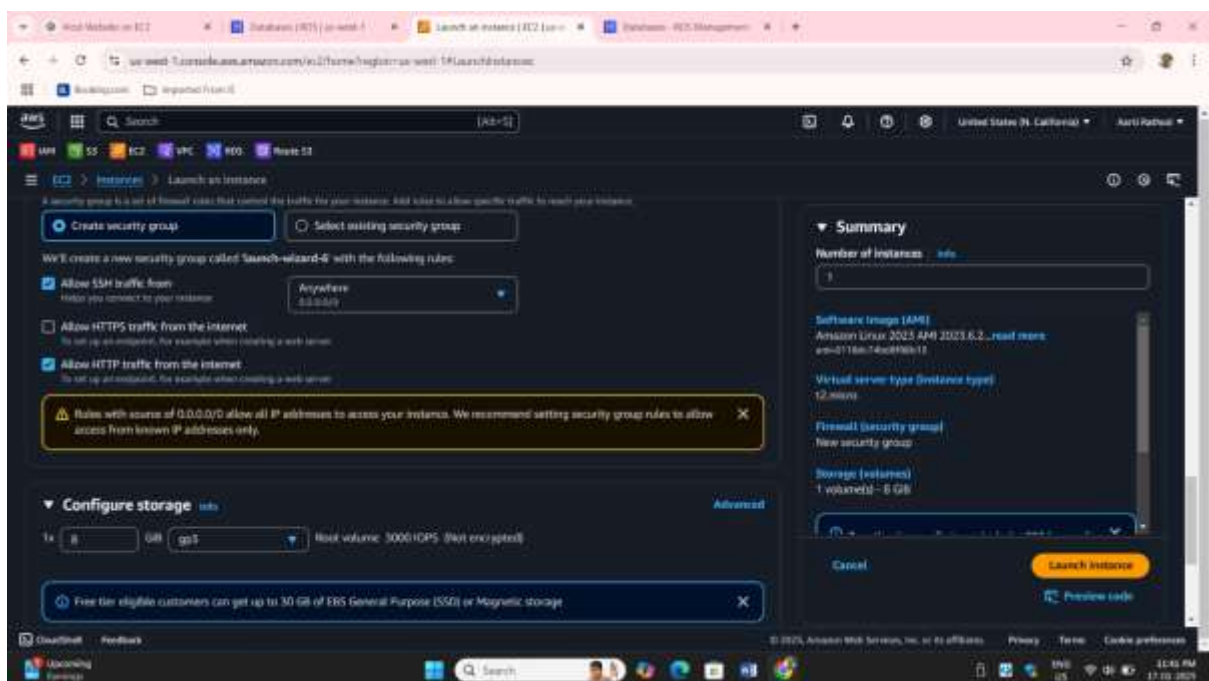
## Step 12 : Give a name and select OS Image Amazon Linux



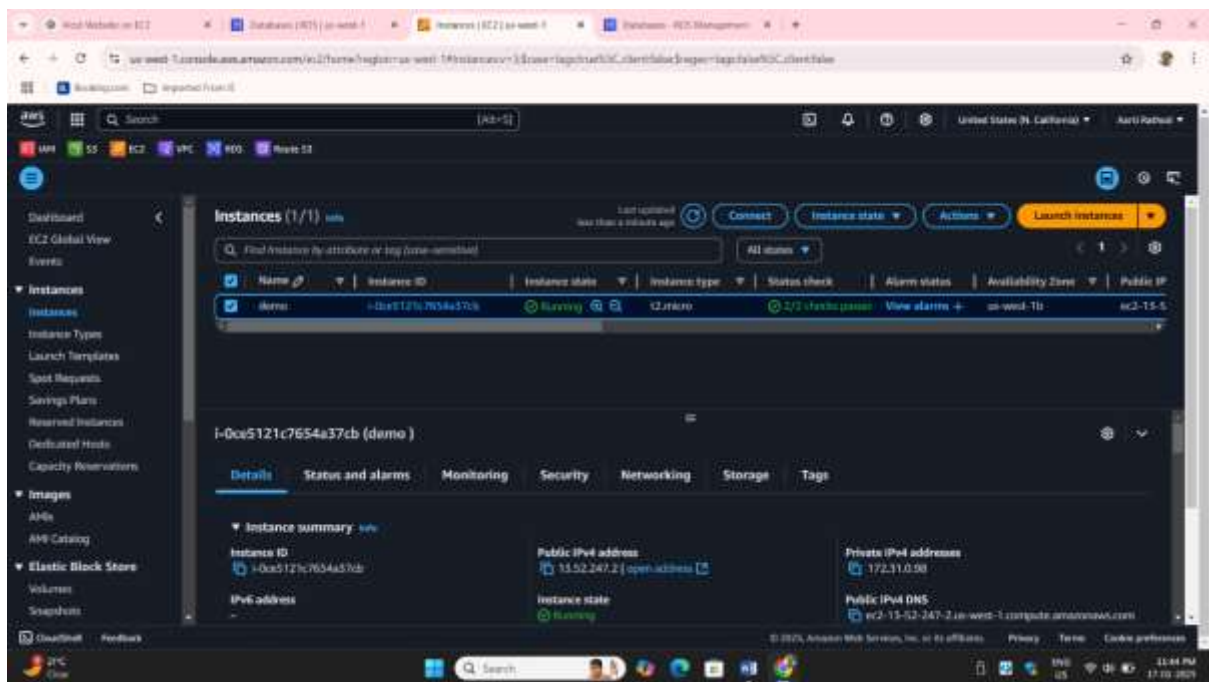
## Step 13 : Select key pair and create security group



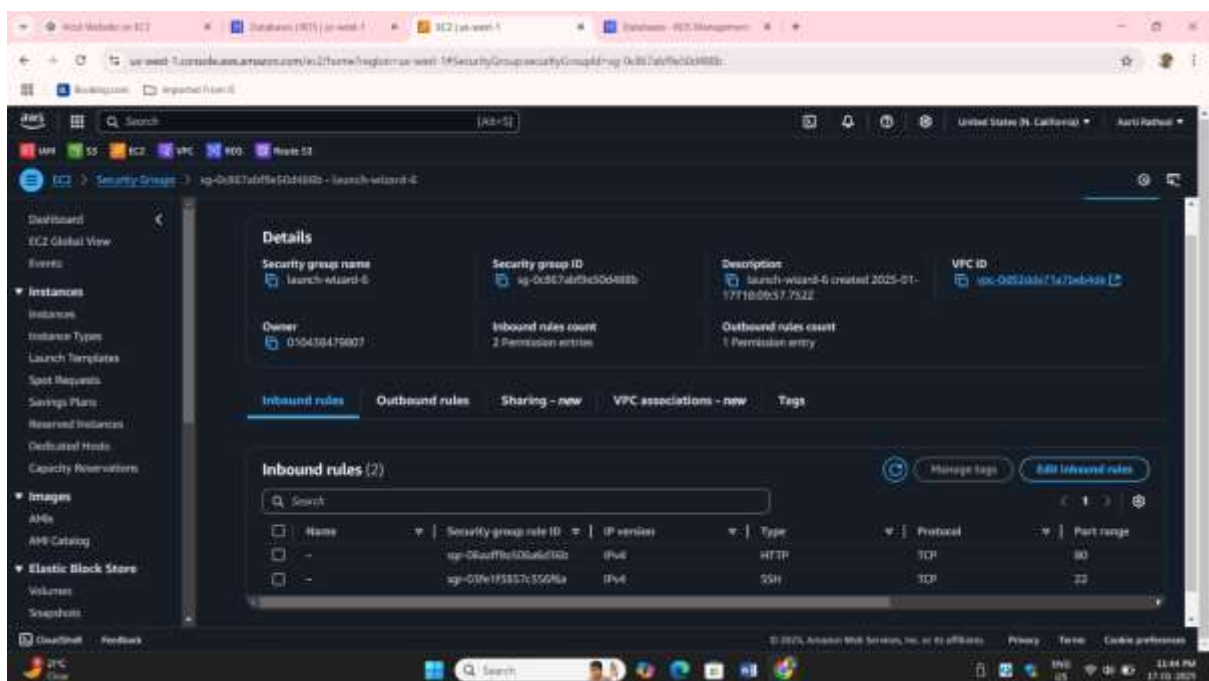
## Step 14 : Allow http and click on Launch Instance



## Step 15 : Check Status and then click on security

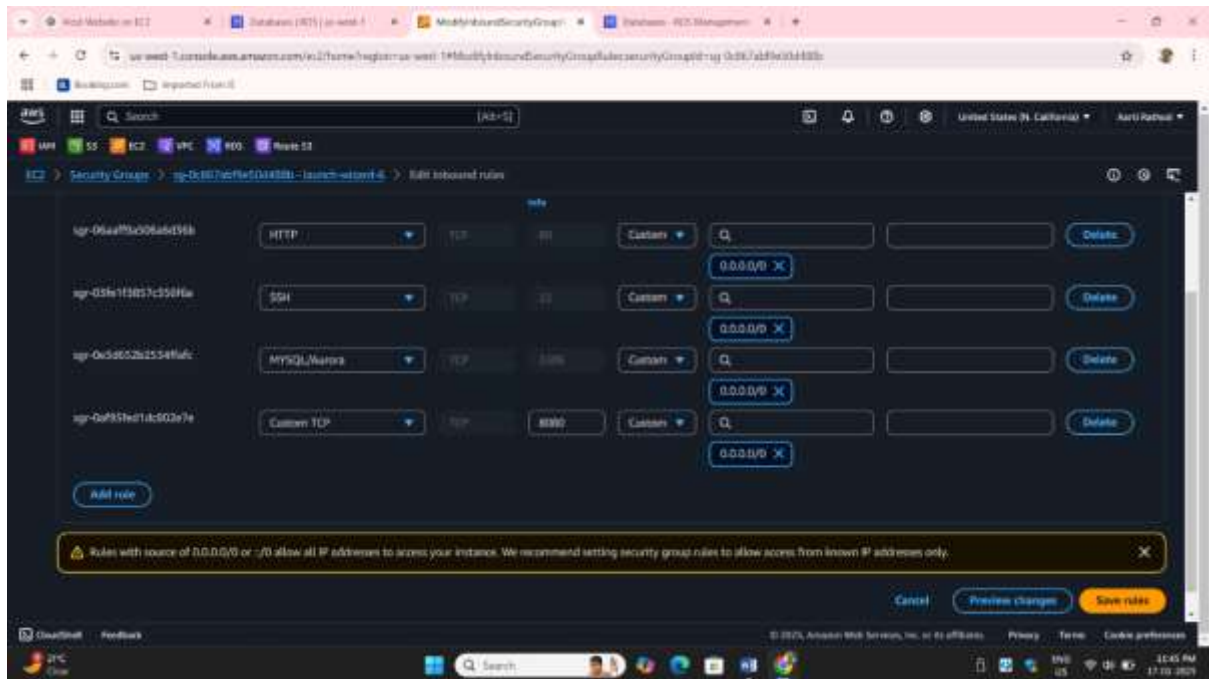


## Step 16 : Go to Edit inbound rules

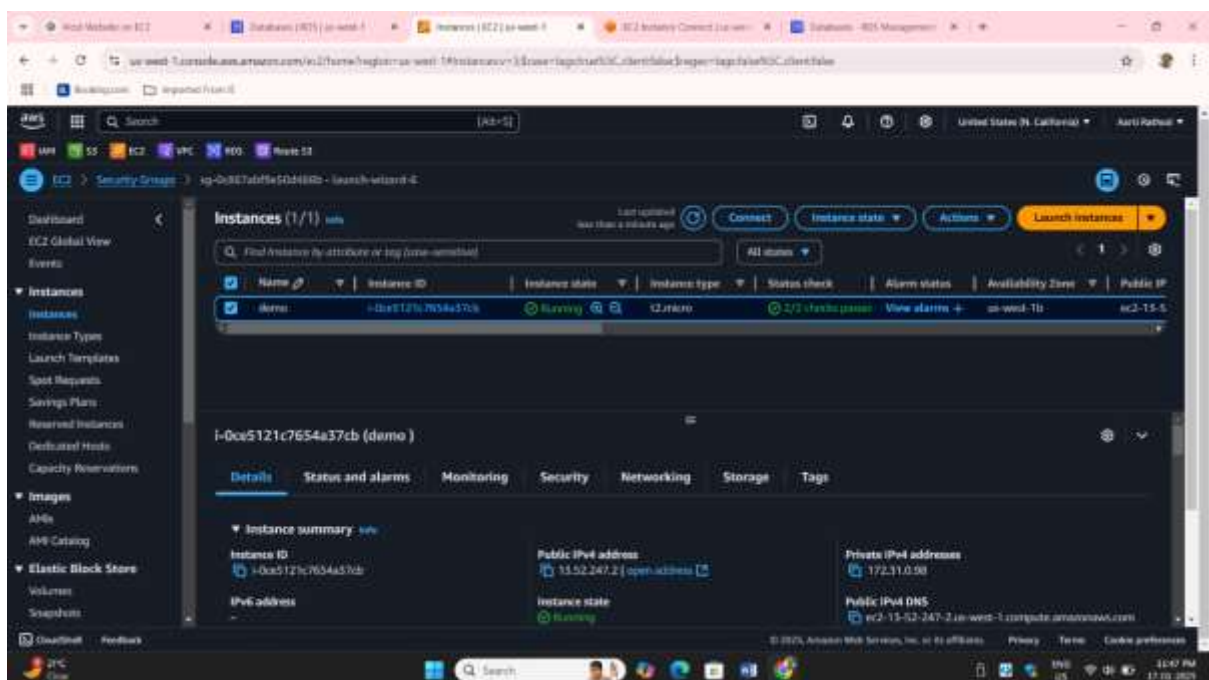




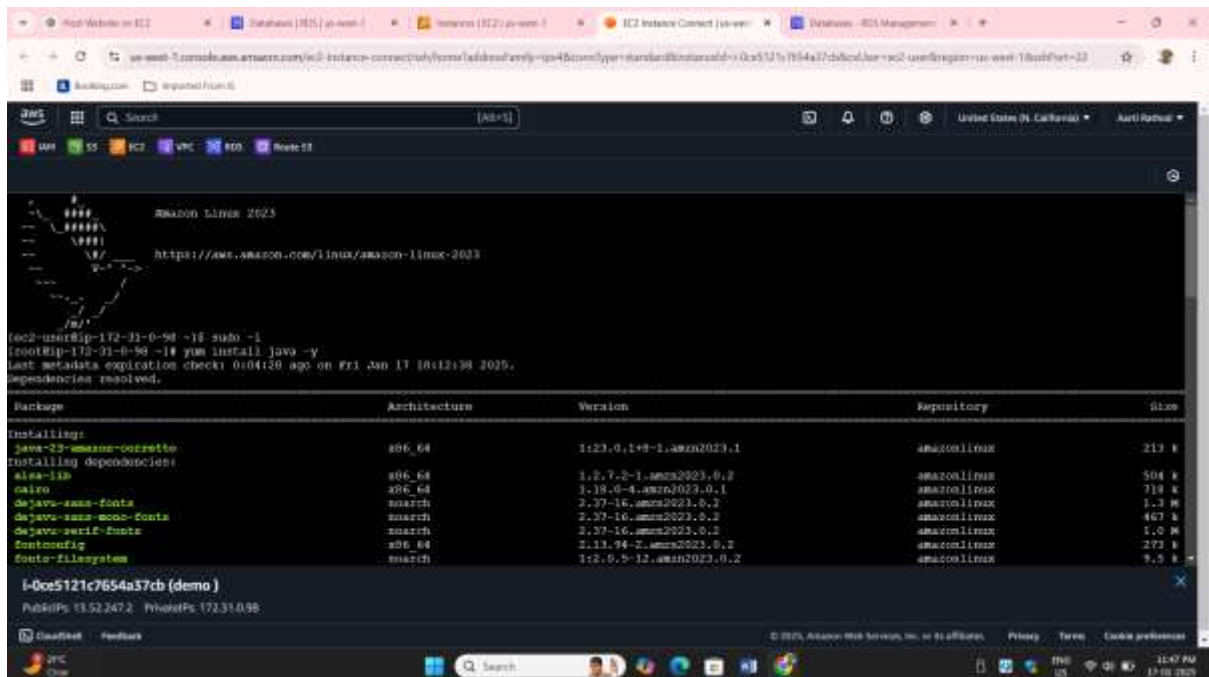
## Step 17 : Allow inbound traffic for port 8080 (for Tomcat) and 3306 (Mysql/aurora)



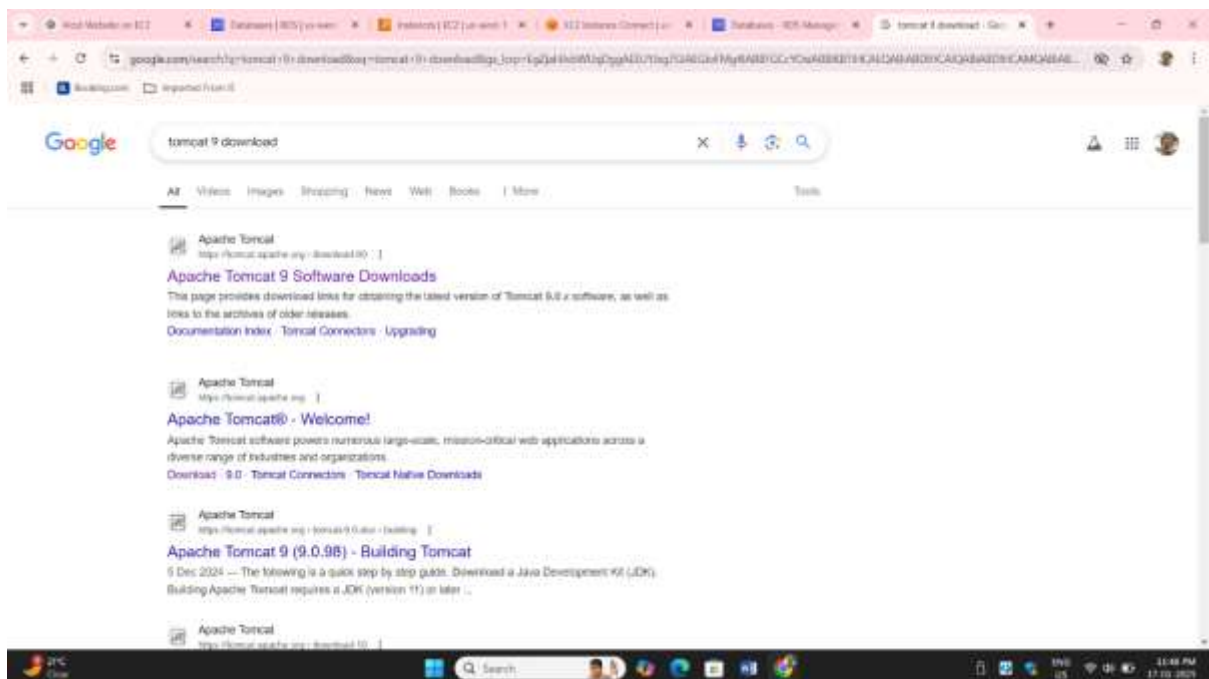
## Step 18 : Goto Instance select your instace and connect



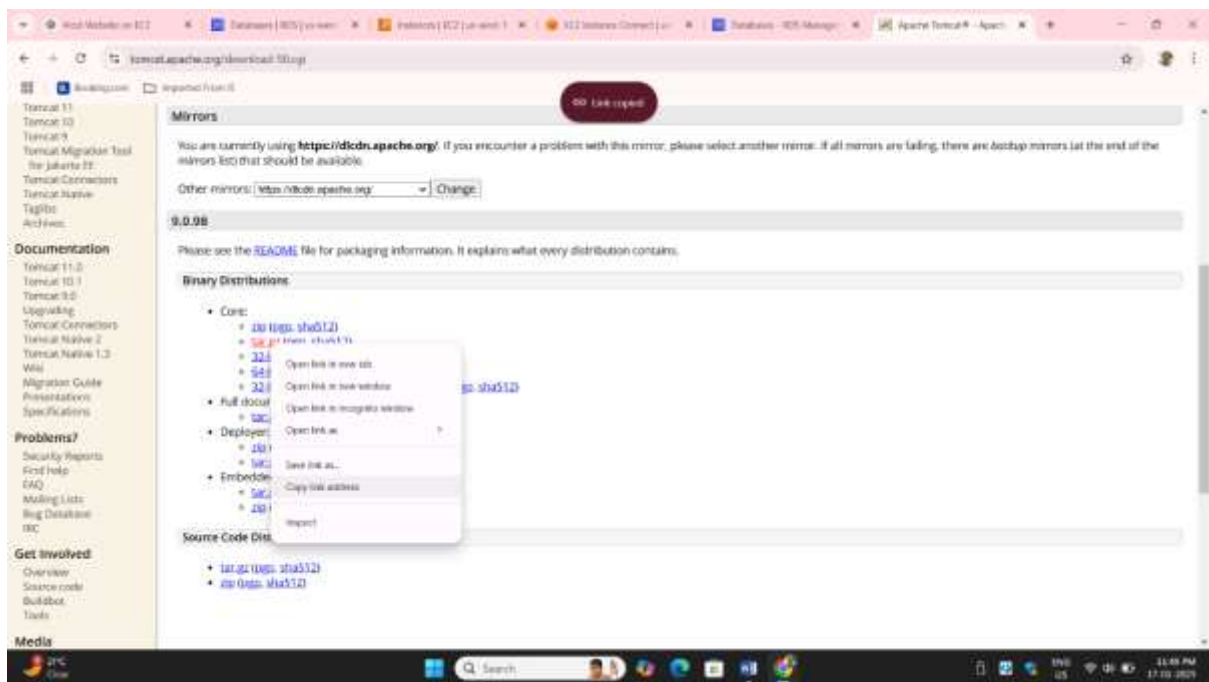
## Step 19 : Switch to root user and install java



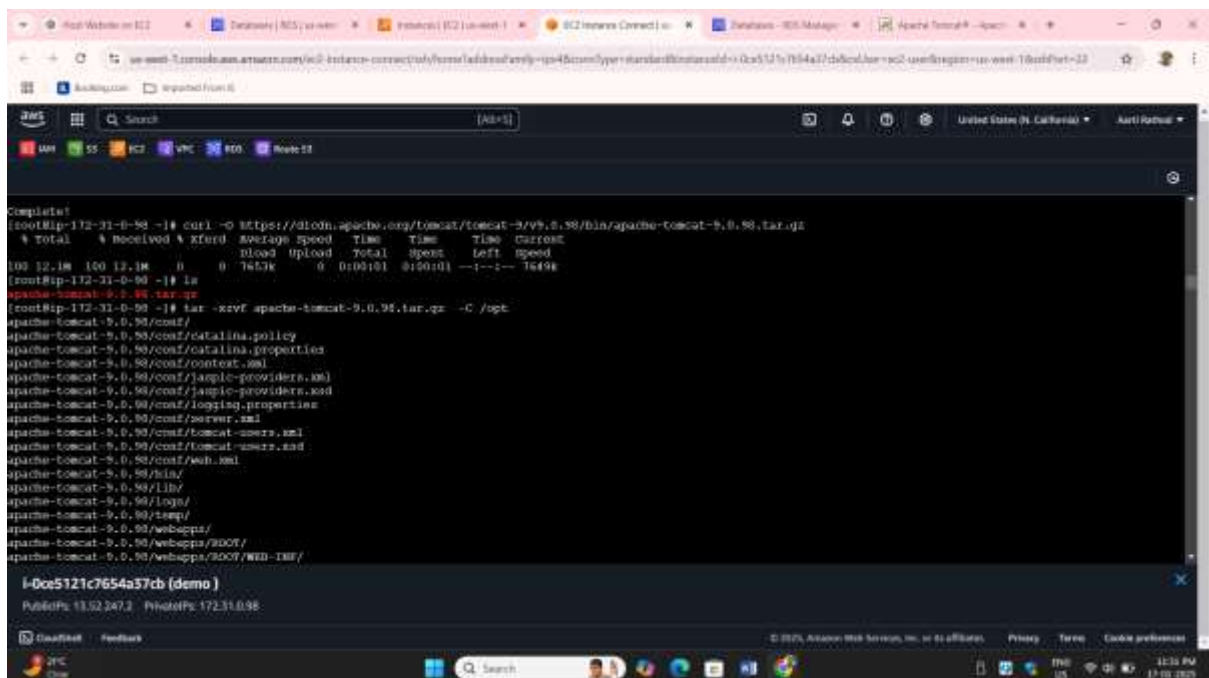
## Step 20 : Go to Google search download tomcat 9



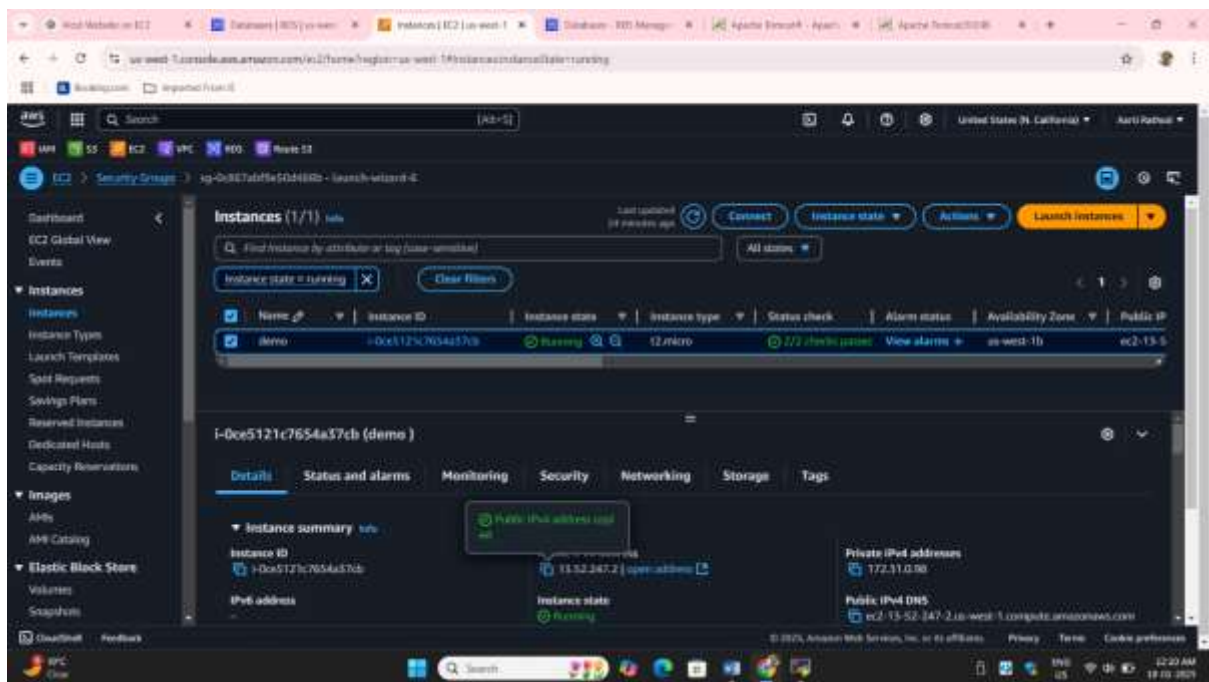
## Step 21 : Go to official link right click tar.gz file copy link address



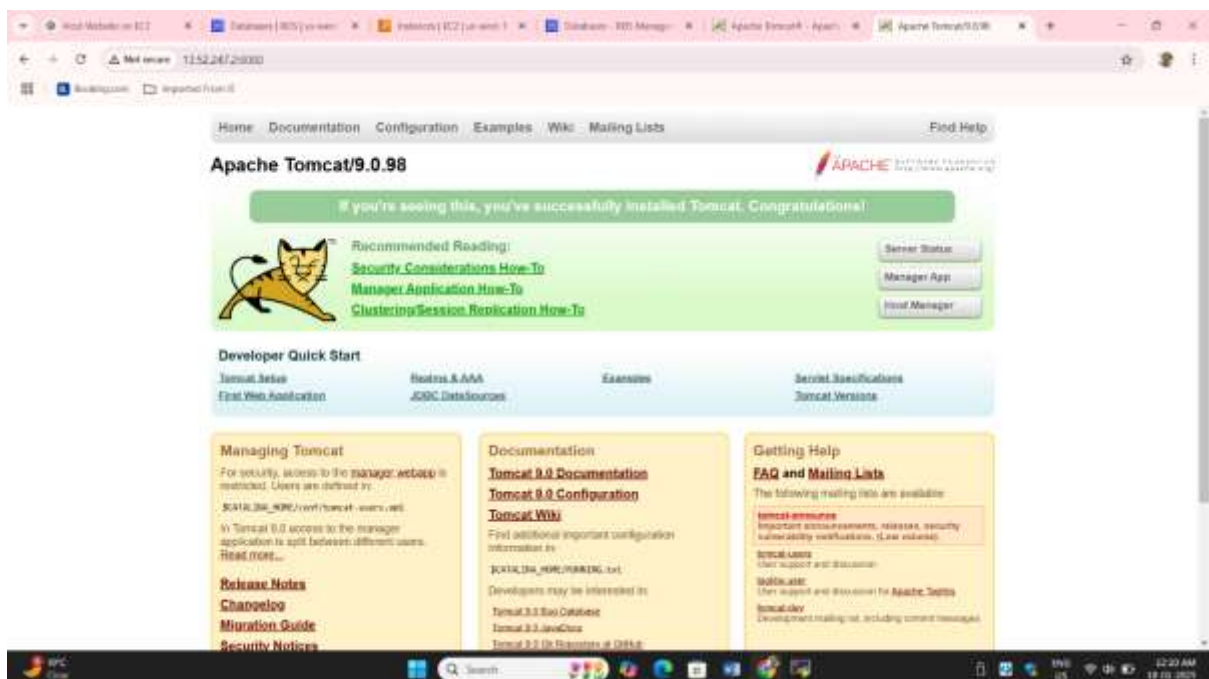
## Step 22 : Paste the click and download using curl -O command and unzip with the command tar -xzf and put that unzip file on /opt directory



## Step 23 : Copy the public ip of instance



## Step 24 : Paste on the tab and see if tomcat page is showing put :8080 port after your ip if its showing tomcat page then its working properly





**Go to cd webapps/ paste .war file curl -O <https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war>**

[illegible]



**Step 27 : Paste on tab and search 13.52.247.2:8080.student/ search and check the form is visible or not**

**Student Registration Form**

Student Name

Student Address

Student Age

Student Qualification

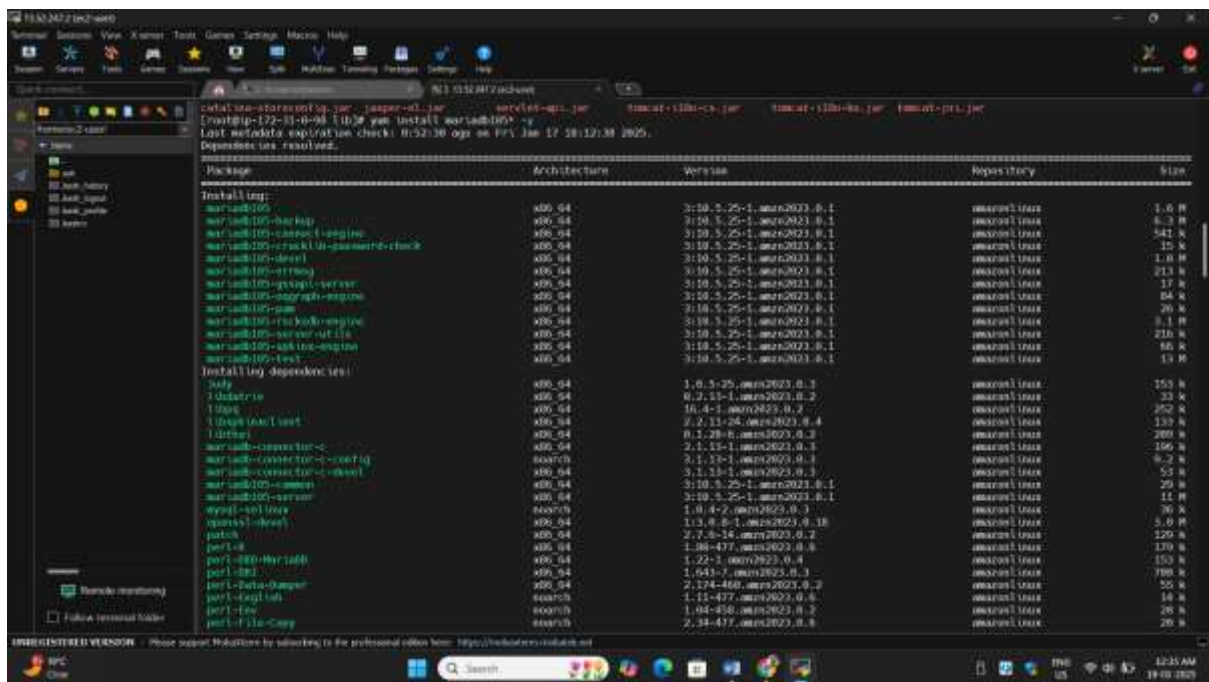
Student Percentage

Year Passed

**Step 28 : Place the JDBC connector .jar file in the /lib folder of Tomcat. go to lib file paste .jar file curl -O <https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar>**

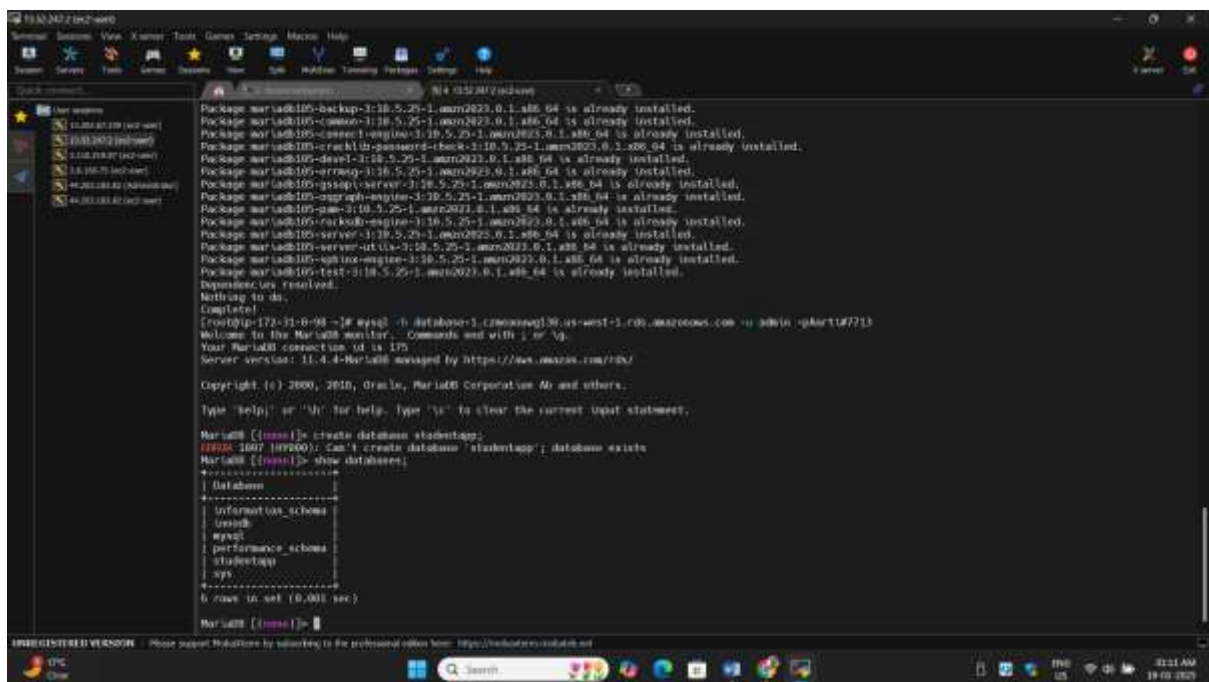
```
[root@ip-172-31-0-93 ~]# curl -O https://s3-us-west-2.amazonaws.com/studentapi-cit/student_war
curl: Remote file name has no length
[root@ip-172-31-0-93 ~]# curl -O https://s3-us-west-2.amazonaws.com/studentapi-cit/student_war
% Total % Recv'd % Xferd Average Speed Time Time Current
                                Dload Upload Total Spent Left Speed
100 88423 100 88423 0 0 458k 0 0:00:01 0:00:01 0 458k
[root@ip-172-31-0-93 ~]# ls
mysql-connector.jar studentapi-cit
[root@ip-172-31-0-93 ~]# cd /usr/local/tomcat/lib
[root@ip-172-31-0-93 ~]# mv mysql-connector.jar mysql-connector.jar
[root@ip-172-31-0-93 ~]# ls
mysql-connector.jar
[root@ip-172-31-0-93 ~]# curl -O https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar
% Total % Recv'd % Xferd Average Speed Time Time Current
                                Dload Upload Total Spent Left Speed
100 583k 100 583k 0 0 2625k 0 0:00:01 0:00:01 0 2625k
[root@ip-172-31-0-93 ~]# ls
mysql-connector.jar
[root@ip-172-31-0-93 ~]# yum install mariadb
Last metadata expiration check: 0:12:30 ago on Fri Jan 17 10:12:30 2025.
Dependencies resolved.
Package Architecture Version Repository Size
Install only:
mariadb105 x86_64 3:10.5.25-1.amzn2023.0.1 amazonlinux 1.6 M
mariadb105-debug x86_64 3:10.5.25-1.amzn2023.0.1 amazonlinux 6.3 M
mariadb105-connect-engine x86_64 3:10.5.25-1.amzn2023.0.1 amazonlinux 541 K
mariadb105-crypt-password-check x86_64 3:10.5.25-1.amzn2023.0.1 amazonlinux 15 K
mariadb105-devel x86_64 3:10.5.25-1.amzn2023.0.1 amazonlinux 1.0 M
mariadb105-libs x86_64 3:10.5.25-1.amzn2023.0.1 amazonlinux 213 K
```

## Step 29 : Install mariadb using yum



**Step 30 : This is the MySQL Client Command, used to connect to a MySQL database which we create on RDS. It allows you to run queries and manage the database directly from the terminal.**

**Create one database name studentapp.**



**Step 31 : Use studentapp. And then create one table :- create table students(student\_id INT NOT NULL AUTO\_INCREMENT, student\_name VARCHAR(100) NOT NULL, student\_addr VARCHAR(100) NOT NULL, student\_age VARCHAR(3) NOT NULL, student\_qual VARCHAR(20) NOT NULL, student\_percent VARCHAR(10) NOT NULL, student\_year\_passed VARCHAR(10) NOT NULL, PRIMARY KEY (student\_id) );**

**Show tables check if your table student is created or not .**

```

MySQL [studentapp]> REALLBACK;
Query OK, 0 rows affected (0.001 sec)

MySQL [studentapp]> create table students(student_id INT NOT NULL AUTO_INCREMENT,
-> student_name VARCHAR(100) NOT NULL,
-> student_addr VARCHAR(100) NOT NULL,
-> student_age VARCHAR(3) NOT NULL,
-> student_qual VARCHAR(20) NOT NULL,
-> student_percent VARCHAR(10) NOT NULL,
-> student_year_passed VARCHAR(10) NOT NULL,
-> PRIMARY KEY (student_id)
-> );
Query OK, 0 rows affected (0.010 sec)

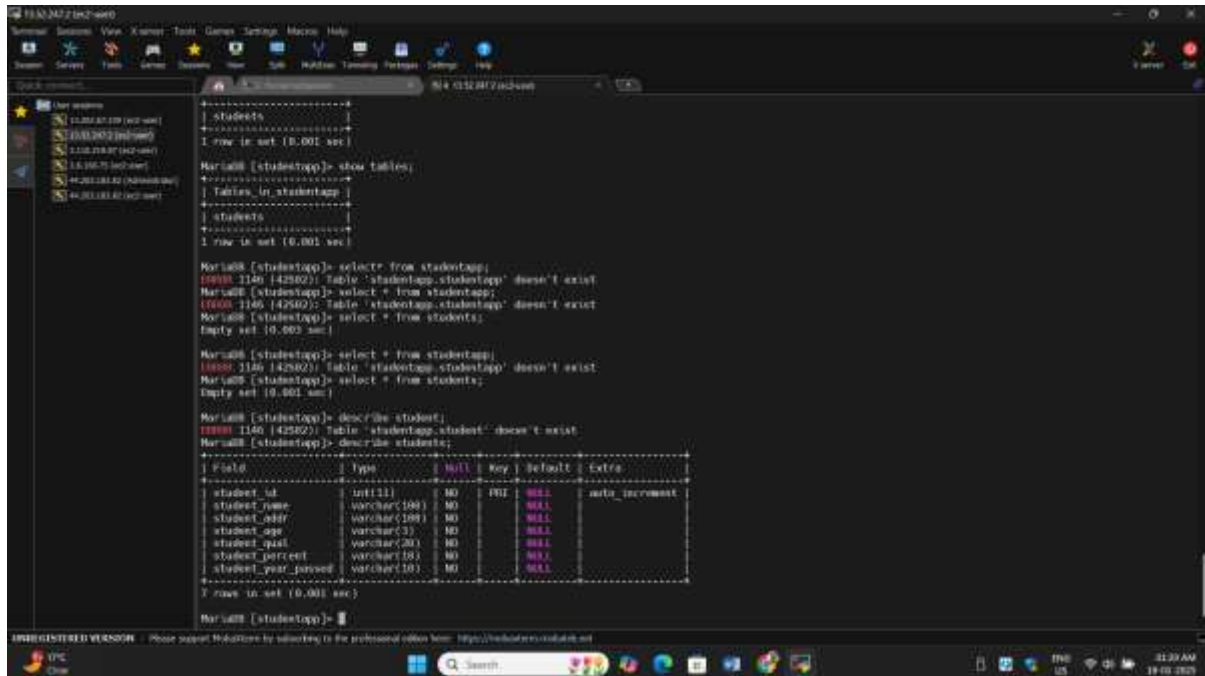
MySQL [studentapp]> show tables;
+-----+
| Tables_in_studentapp |
+-----+
| students              |
+-----+
1 row in set (0.001 sec)

MySQL [studentapp]> show tables;
+-----+
| Tables_in_studentapp |
+-----+
| students              |
+-----+
1 row in set (0.001 sec)

MySQL [studentapp]> show tables;
+-----+
| Tables_in_studentapp |
+-----+
| students              |
+-----+
1 row in set (0.001 sec)

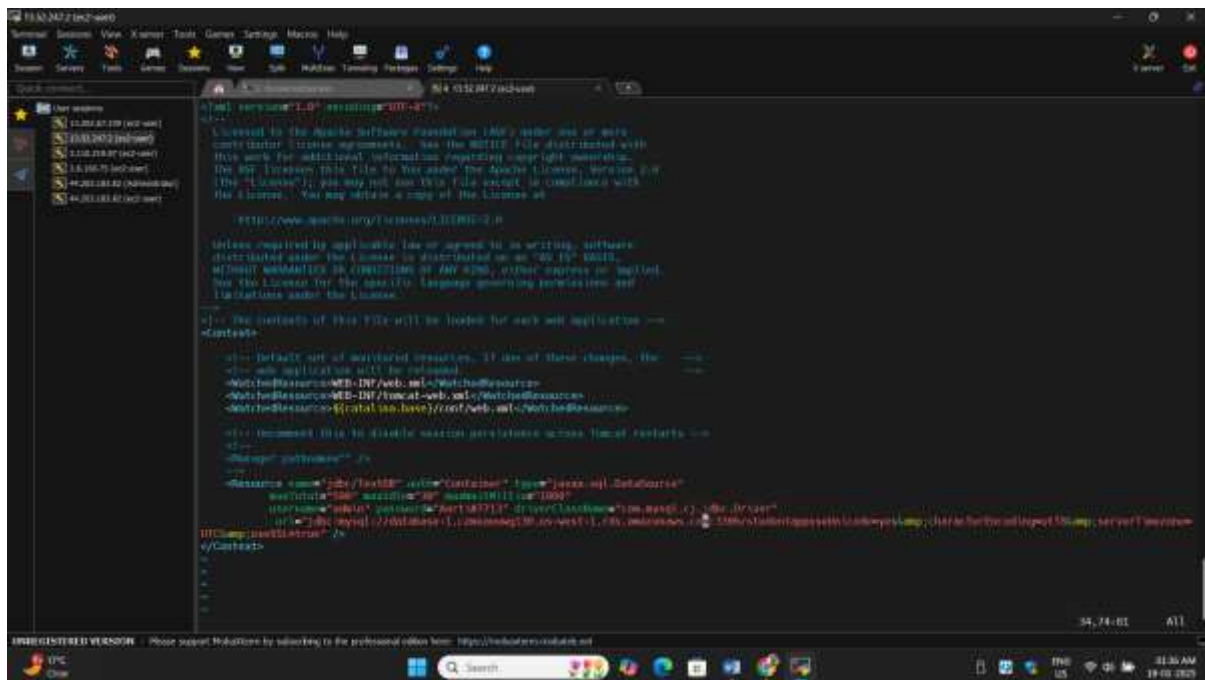
MySQL [studentapp]> select* from studentapp;
(1000, 1146, 142602); Table 'studentapp.studentapp' doesn't exist
MySQL [studentapp]> select * from studentapp;
(1000, 1146, 142602); Table 'studentapp.studentapp' doesn't exist
  
```

**Run DESCRIBE student;** is used to display the structure of the student table, including column names, data types, and constraints.

[illegible]

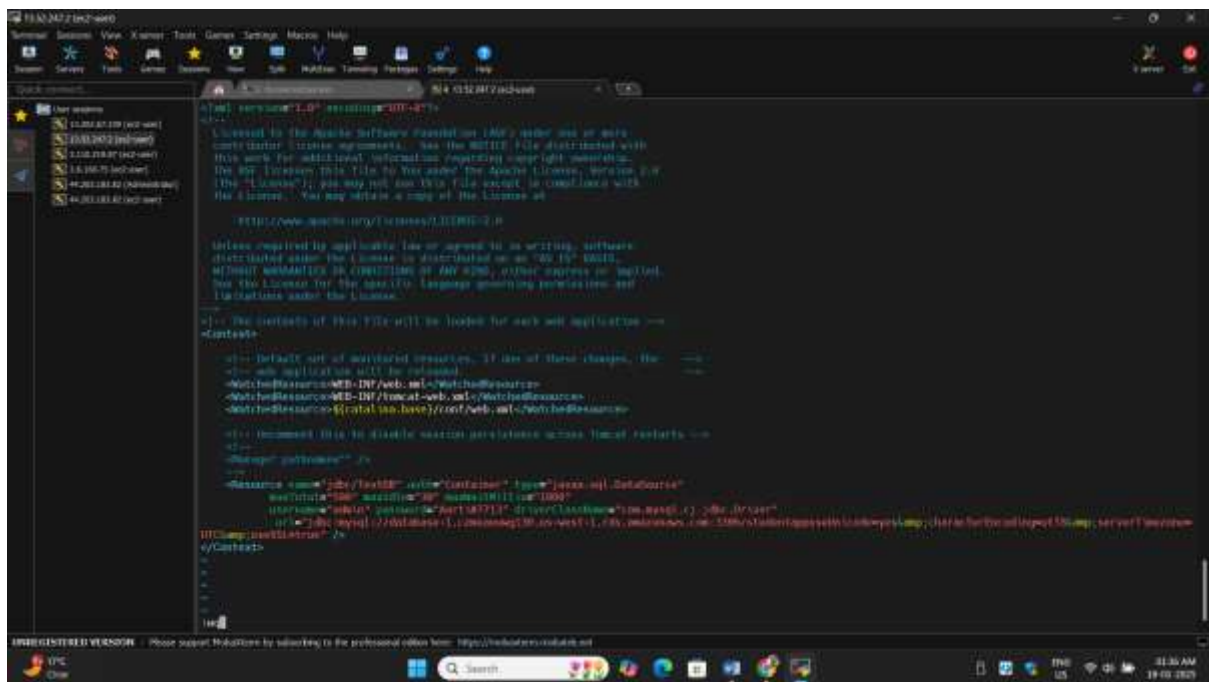
**Step 33: Edit the context.xml file in the Tomcat conf folder to add the <Resource> element for the database connection. Example:**

```
<Resource name="jdbc/StudentDB" auth="Container"
    type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="your_db_username" password="your_db_password"
    driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://<RDS-endpoint>:3306/student_app"/>
```

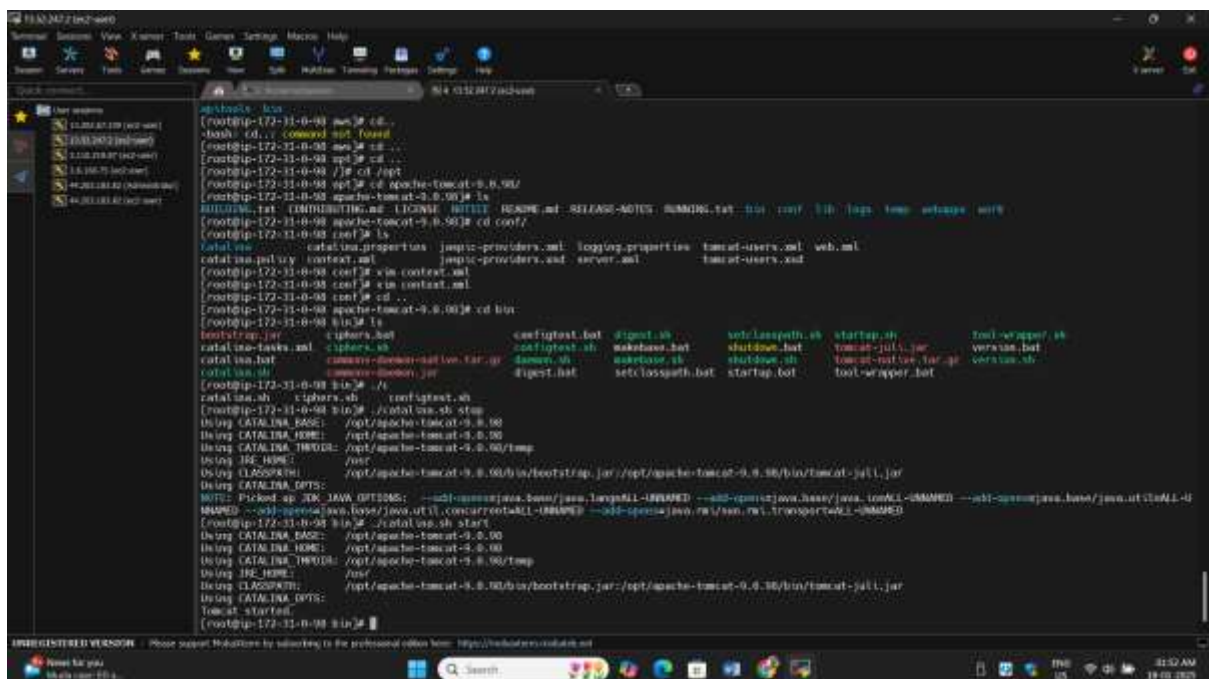





## Step 34: Save And Quit.



**Step 35: Go to cd apac-> go to cd bin -> stop catalina -> again start catalina**



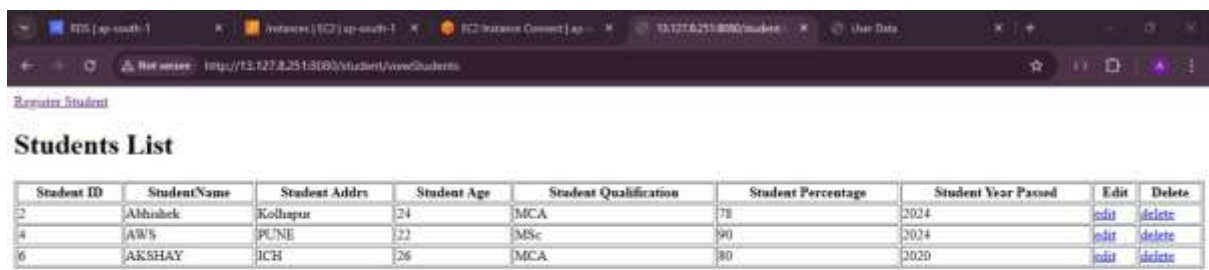
**Step 36 : Copy public ip paste on new tab run 13.52.247.2:8080/student/ fill the data and click on register**



**Student Registration Form**

Student Name	AKSHAY
Student Address	ICH
Student Age	26
Student Qualification	MCA
Student Percentage	80
Year Passed	2020
<input type="button" value="register"/>	

**Step 47 : Here your data are stored**



**Students List**

Student ID	Student Name	Student Address	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
2	Abhinav	Kollhapur	24	MCA	78	2024	<a href="#">edit</a>	<a href="#">delete</a>
4	AW'S	PUNE	22	MSc	90	2024	<a href="#">edit</a>	<a href="#">delete</a>
6	AKSHAY	ICH	26	MCA	80	2020	<a href="#">edit</a>	<a href="#">delete</a>

**This demonstrates a classic 2-tier architecture:**

**Tier 1: The web application running on Tomcat (EC2).**

**Tier 2: The MySQL database hosted in AWS RDS.**