

**Name :- Anjali Punsri**

**Class :- D20B**

**Roll No :- 57**

**Experiment no :- 5**

**Aim :-** To study and implement Database as a service on NoSQL / SQL databases like AWS RDS

### **Theory :-**

What is DBaaS?

DBaaS (also known as managed database service) is a [cloud computing](#) service that lets users access and use a cloud database system without purchasing and setting up their own hardware, installing their own database software, or managing the database themselves (not to mention hiring the high-priced talent required to do so). The cloud provider takes care of everything from periodic upgrades to backups to ensuring that the database system remains available and secure 24/7.

The market for DBaaS and [cloud databases](#) is among the fastest-growing [Software-as-a-Service \(SaaS\)](#) markets, expected to grow to [USD 320 billion by 2025](#) (link resides outside IBM). Database and data warehouse vendors have joined established cloud providers in offering hosted versions of their software, enabling customers to leverage the many [benefits of cloud computing](#) for their applications' data storage, search, and access needs.

### **Benefits**

Compared to deploying a database management system on-premises, DBaaS offers your organization significant financial, operational, and strategic benefits:

**Cost savings:** Laying down infrastructure for database management is expensive; scaling it as needed is costly and often wasteful. With DBaaS, your organization pays a predictable periodic charge based on the resources you consume—there's no need to purchase additional capacity to have on hand for hypothetical future needs.

**Scalability—up and down:** You can quickly and easily provision additional storage and computing capacity at run time if you need it, and you can scale down your database cluster during non-peak usage times to save cost.

**Simpler, less costly management:** To manage and maintain a database on-premises, you'd need an in-house administrative team. With DBaaS, the cloud provider manages everything (although you can choose to manage certain aspects yourself if you wish). DBaaS lightens the administrative burden on your existing IT staff and frees them to work on applications and

Innovation.

Rapid development and faster time-to-market: With an on-premises database system, development teams typically need to request access through IT, a process that can take days or weeks. In contrast, with DBaaS, developers can help themselves to database capabilities and spin up and configure a database that's ready to integrate with their application in minutes.

Data and application security: Cloud database providers typically offer enterprise grade security, including features like default encryption of data at rest and in-transit and integrated identity and access management controls. Some also meet specific regulatory compliance standards.

How to choose a DBaaS?

Major cloud providers offer a wide array of DBaaS options, including [relational database management systems \(RDBMs\)](#), as well as non-relational or [NoSQL databases](#), such as document and column stores. Finding the right DBaaS provider for your enterprise involves determining which database technologies will work best for your application and then, of course, ensuring that your provider supports that technology. The first half of the process can be complex since there's no one-size-fits-all DBaaS that's optimal for use with all of your applications. Trade-offs are always involved, and sometimes they can be subtle. Here are some specific factors you'll need to consider.

Steps :-

Step1 : Login to aws console and search RDS

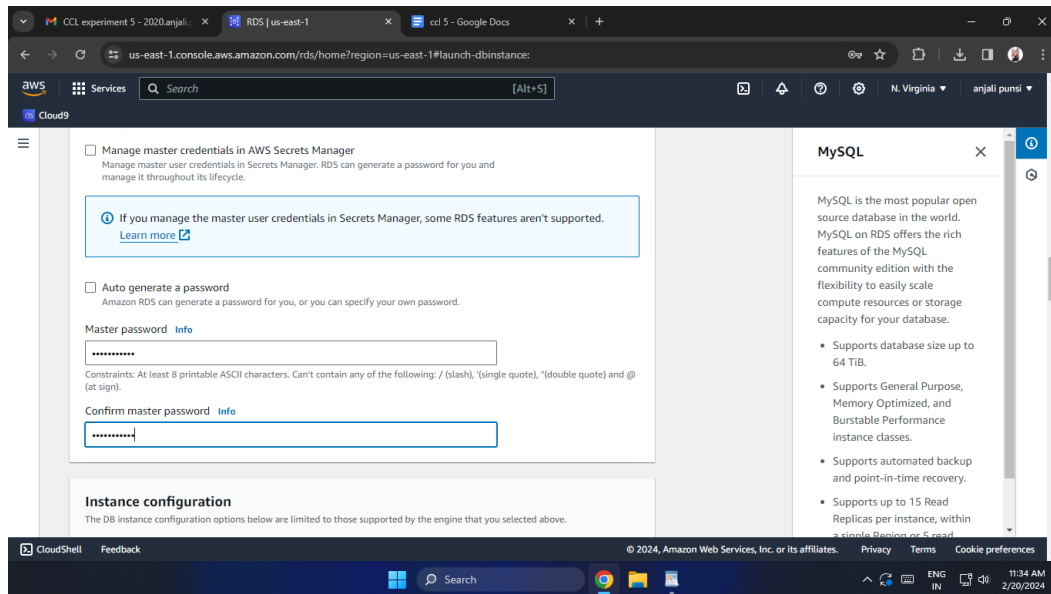
Step2: Click on to RDS and create database

Step 3: Select standard database

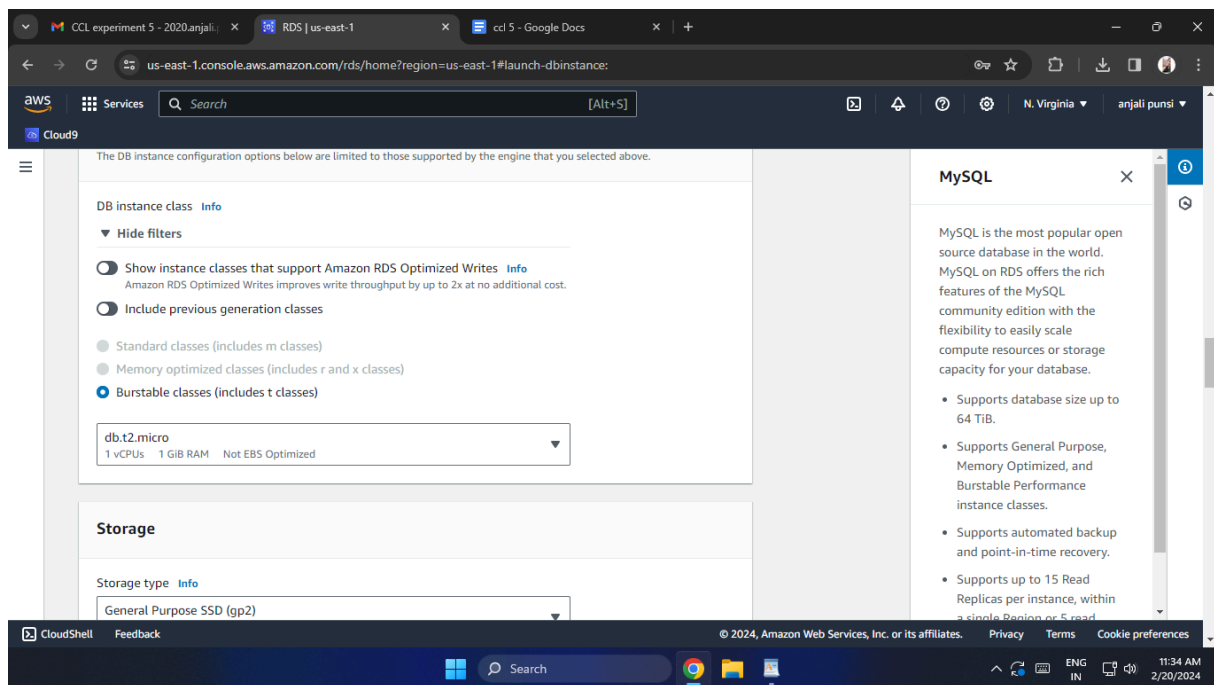
Step 4: Select MySQL and MySQL Community edition

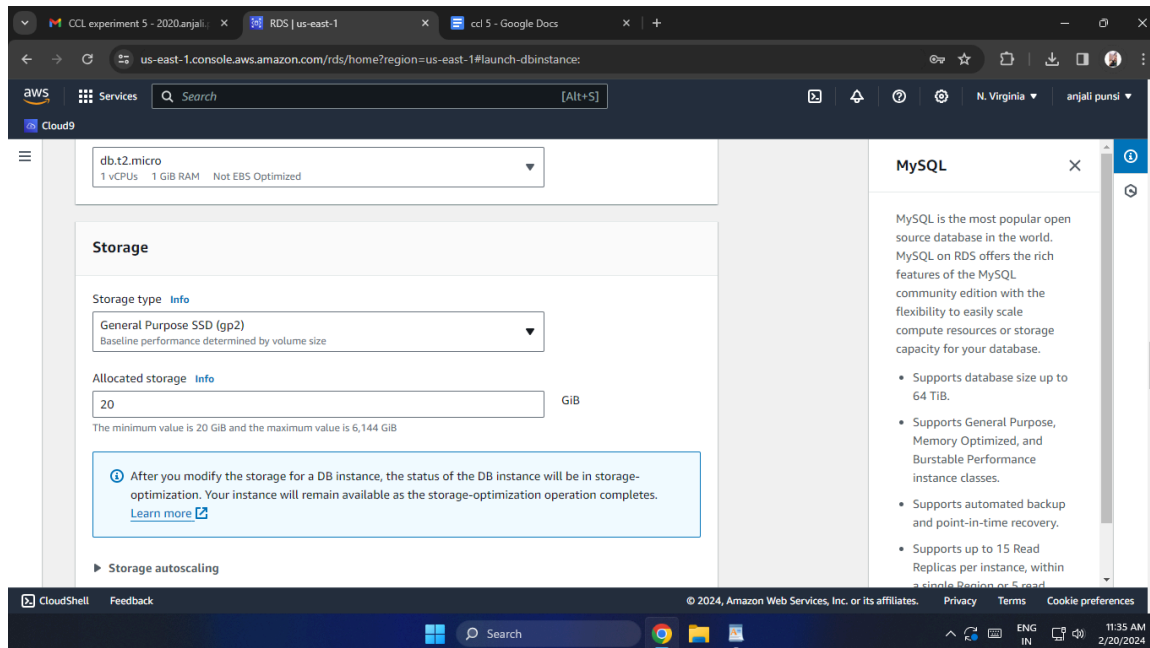
Step 5: In Templates select Free tier

Step 6: Mention database name (default is database1) and username and password

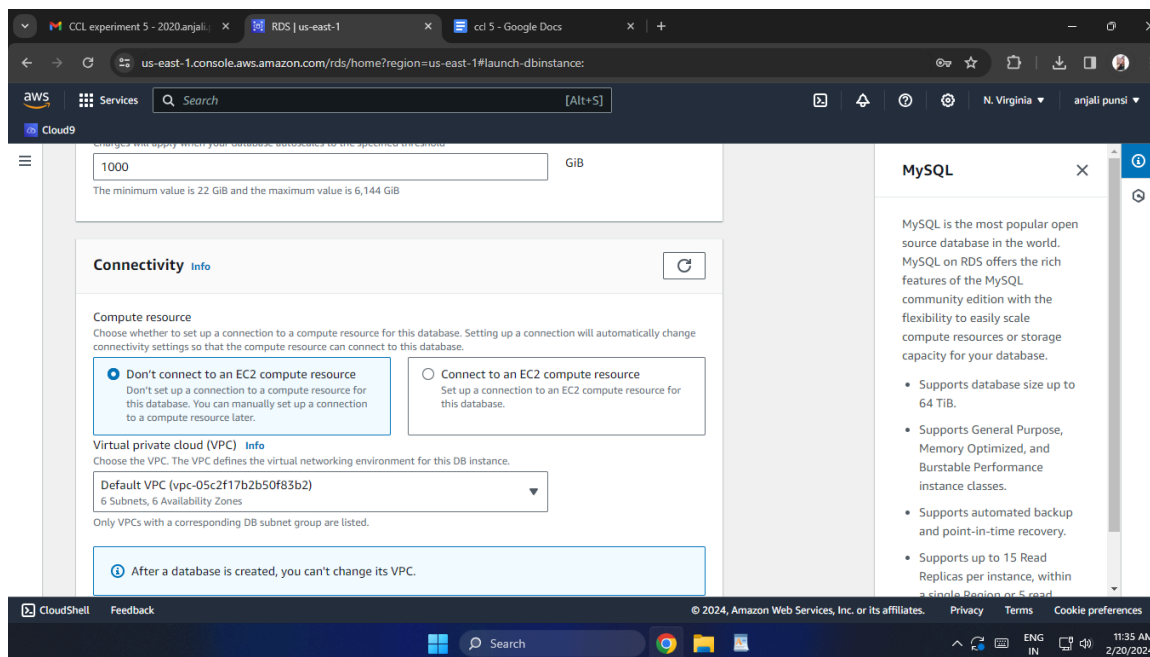


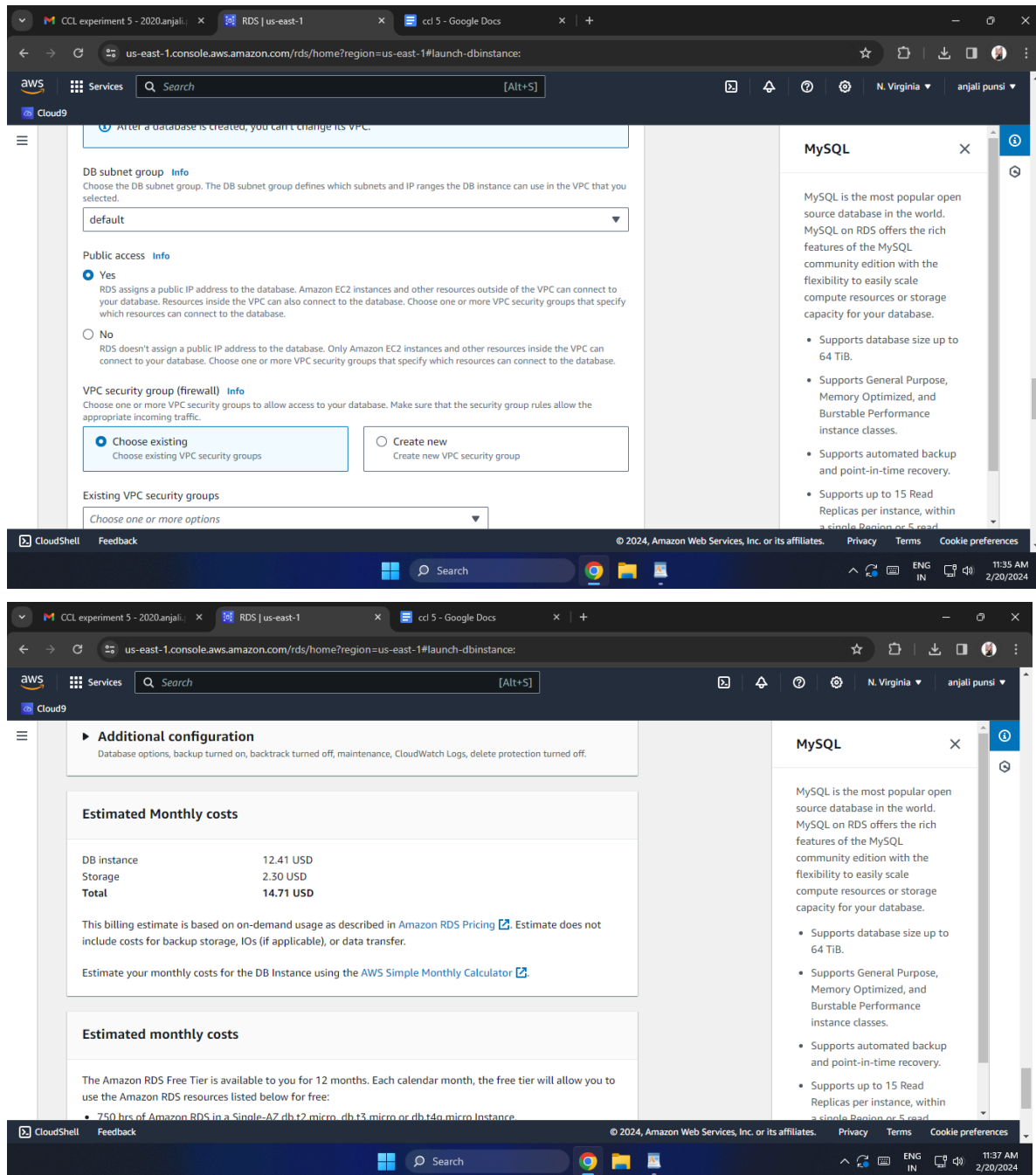
## Step 7: Instance is t2.micro



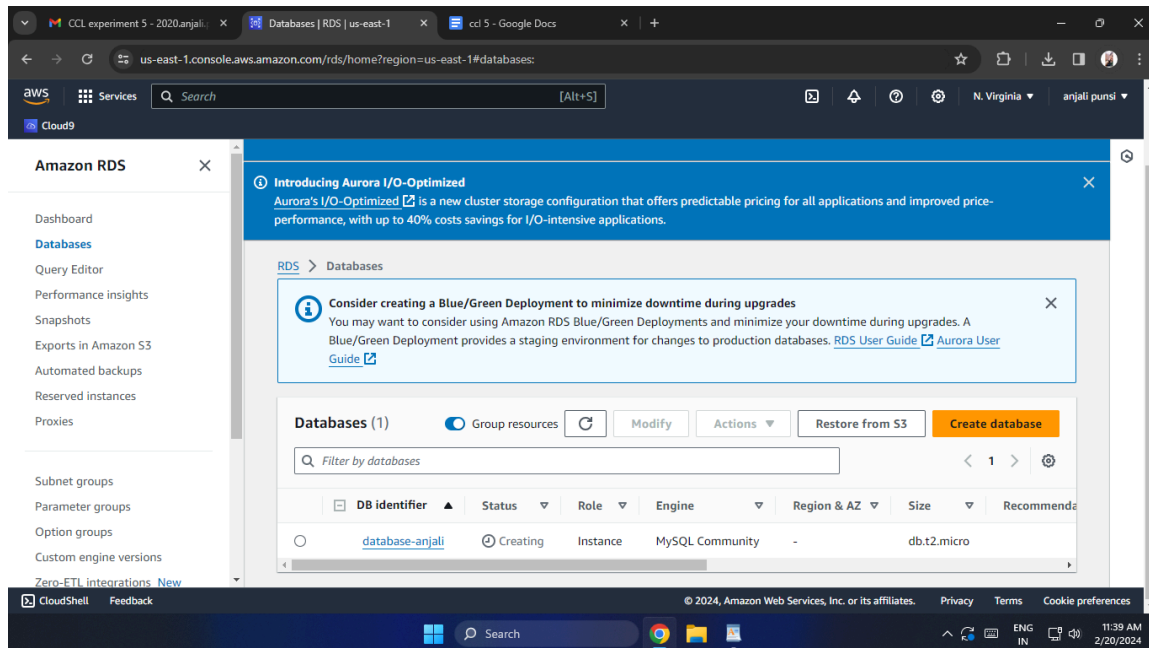


## Step 8: Select Public Access -Yes





Step 9: Click on to create Database



Step 10: It will take some time

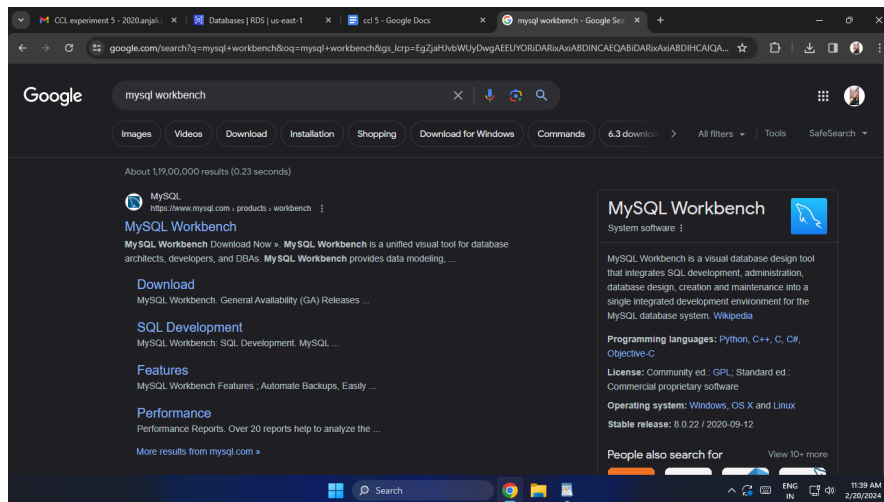
Step 11: Go to google type mysql workbench

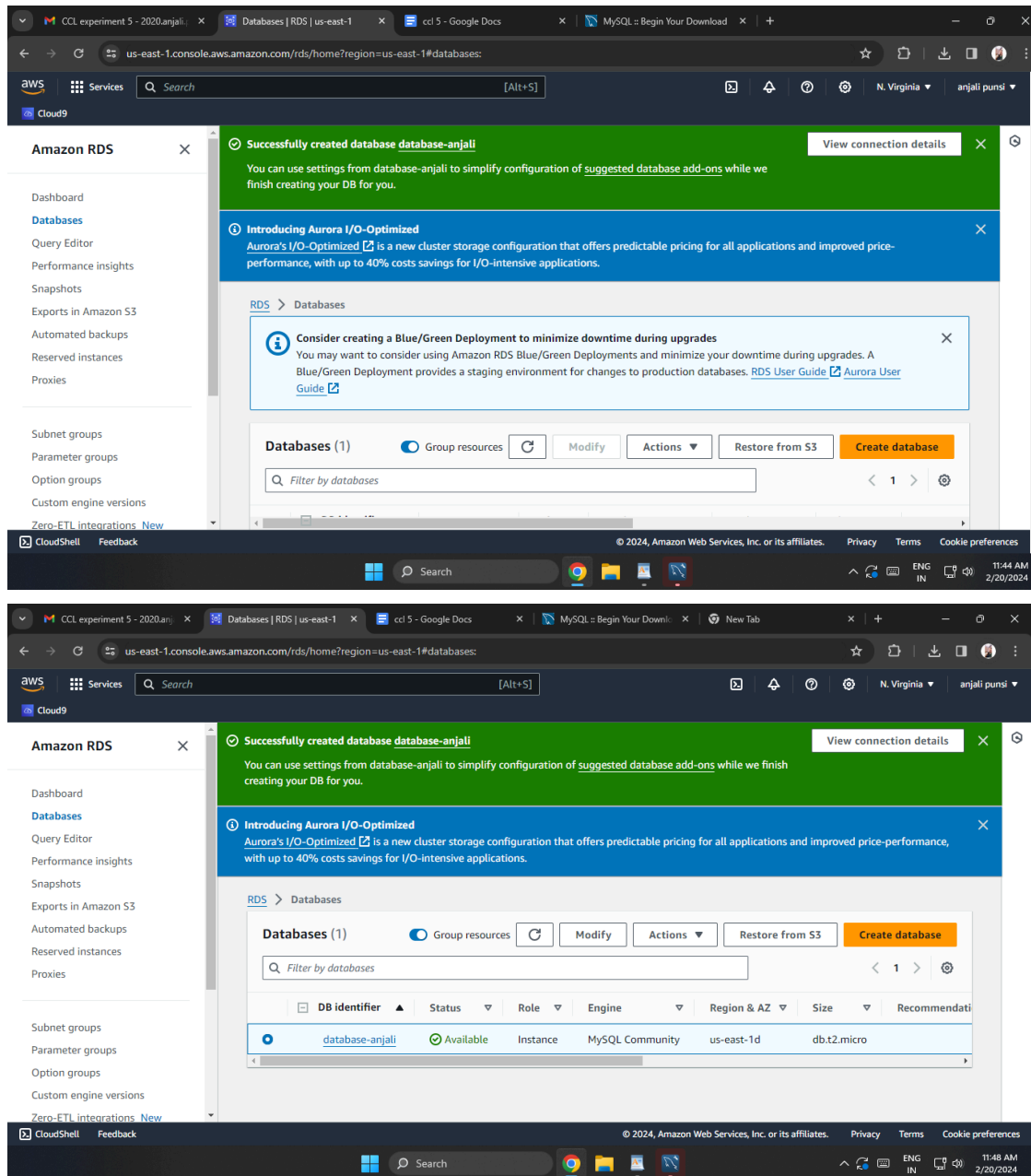
Step 12: Click on to download

Step 12: MySQL community download – Microsoft Windows

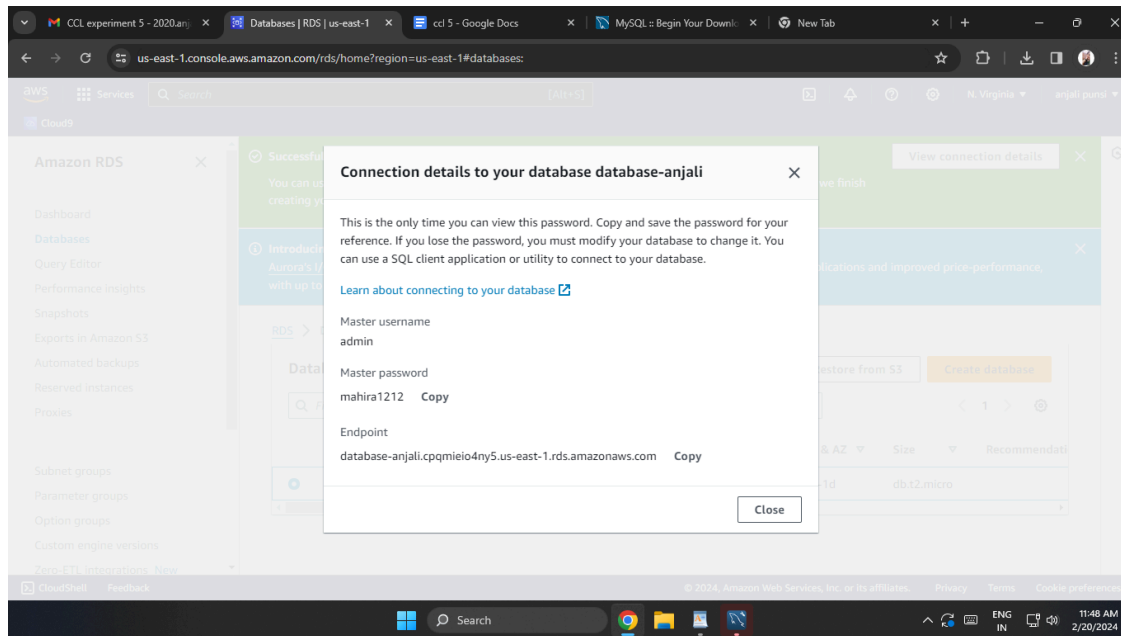
Step 13: Click on to – No thanks , just download

Step 14: Go to downloads of your machine and install it with default settings





Check your database is created and status is available  
Step 15: Click on to view credential

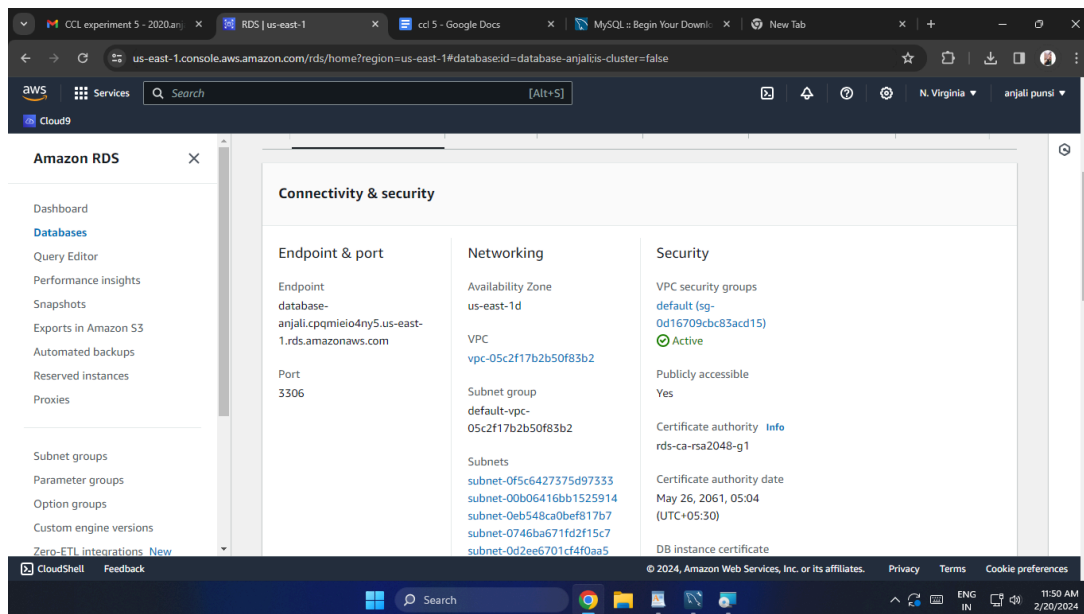


Step 18: Go back to workbench and add your admin and other details

Step 20: Click on to mysql connection

Step 16: Click on to database

Step 17: Copy Endpoint



Step 22: Go to vpc security group

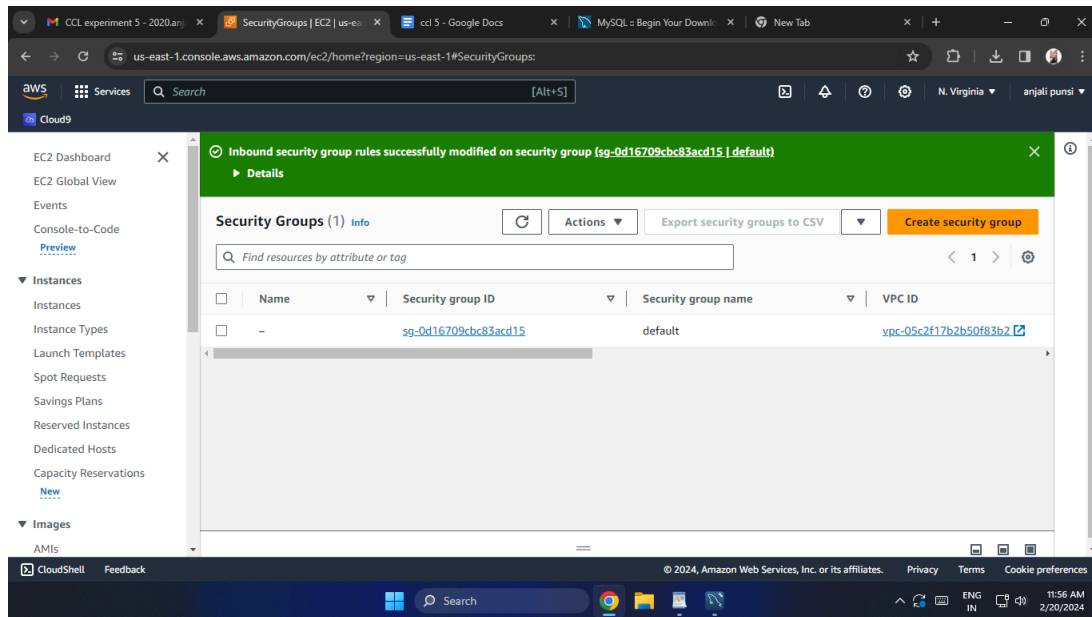
Step 23: Click on to inbound rules

Step 24: First select Click on to Edit inbound rule add rule select ipv4 --all traffic (add 0.0.0.0/0) and

save Rules

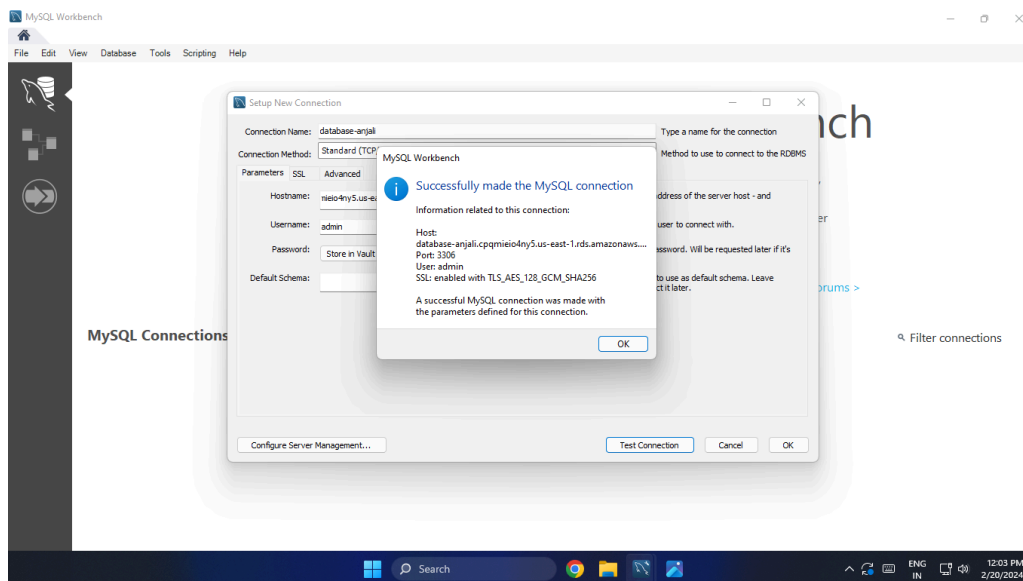
(important step to add inbound rule)





Click on Ok button

Go to workbench double click on connection(databaseshilpa)



Step 22: Goto workbench (after giving details click on to Test Connection)

Step 23:

Write query and execute

Create database tsec;

Use tsec;

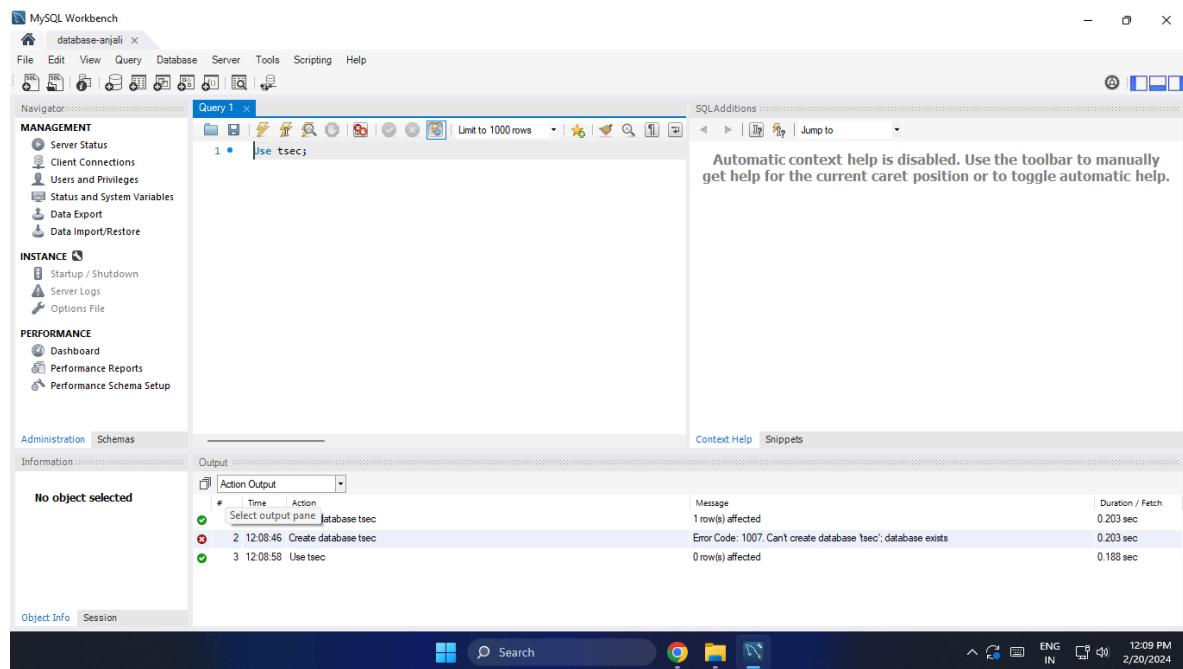
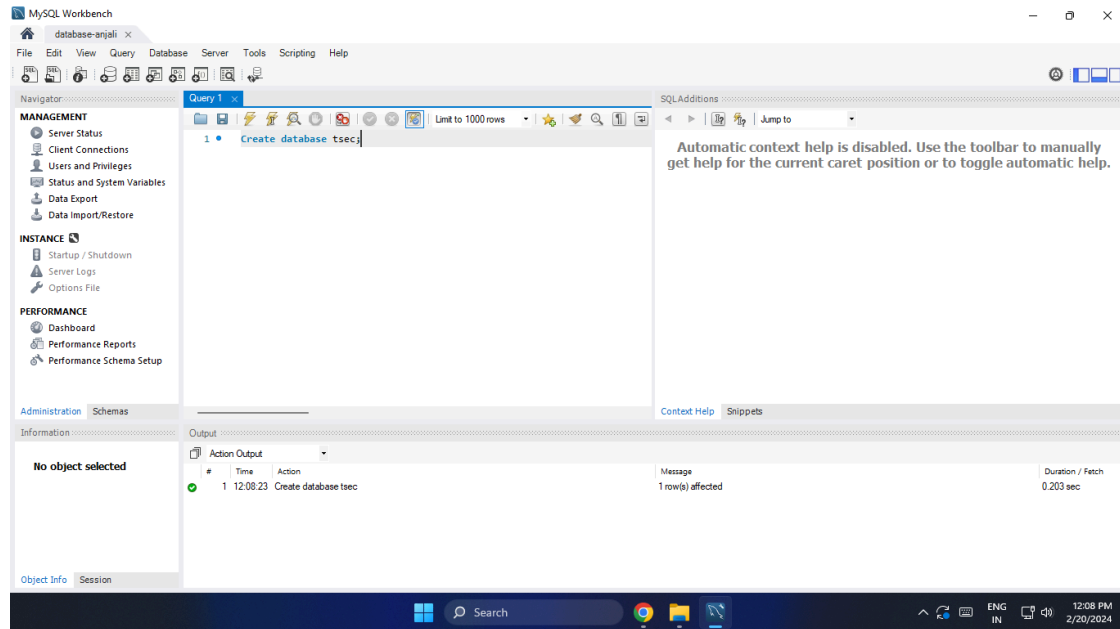
Show tables;

Create table for eg:

create table student( roll int, name varchar(10), city varchar(10));

Describe student;

insert into student values(1,'shilpa','thane'); (Perform all CRUD) operations)



create table student( roll int, name varchar(10), city varchar(10));

MySQL Workbench

database-anjali x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Object Info Session

Query 1 x

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:08:23	Create database tsec	1 row(s) affected	0.203 sec
2	12:08:46	Create database tsec	Error Code: 1007. Can't create database 'tsec'; database exists	0.203 sec
3	12:08:58	Use tsec	0 row(s) affected	0.188 sec
4	12:09:16	Show tables	0 row(s) returned	0.187 sec / 0.000 sec
5	12:09:52	create table student(roll int, name varchar(10), city varchar(10))	0 row(s) affected	0.235 sec

MySQL Workbench

database-anjali x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Object Info Session

Query 1 x

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

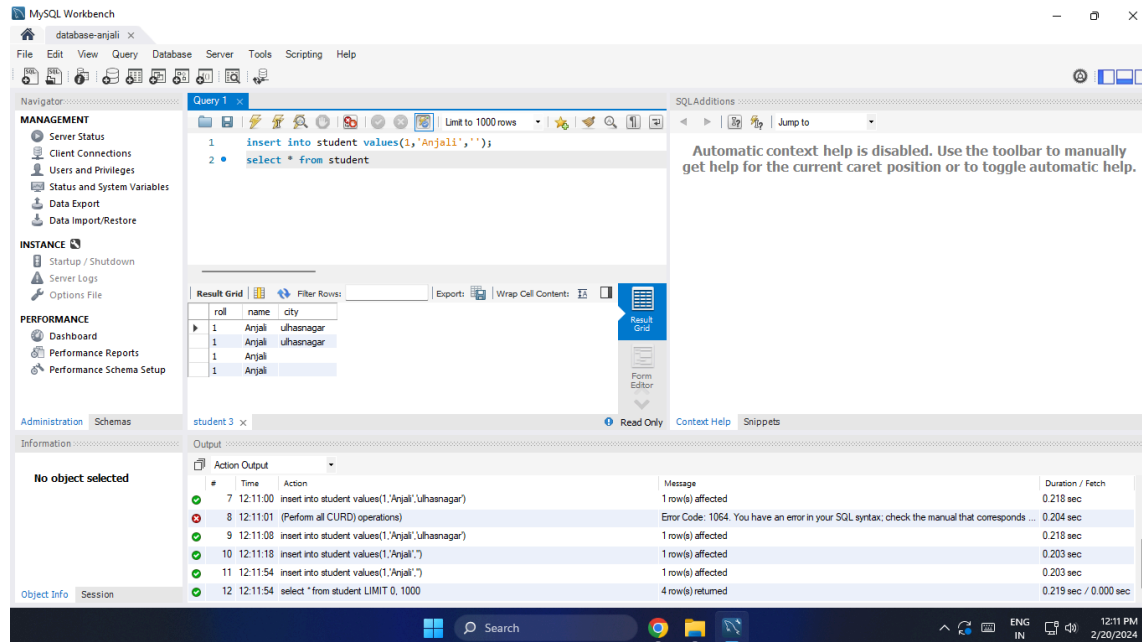
Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:08:23	Create database tsec	1 row(s) affected	0.203 sec
2	12:08:46	Create database tsec	Error Code: 1007. Can't create database 'tsec'; database exists	0.203 sec
3	12:08:58	Use tsec	0 row(s) affected	0.188 sec
4	12:09:16	Show tables	0 row(s) returned	0.187 sec / 0.000 sec
5	12:09:52	create table student(roll int, name varchar(10), city varchar(10))	0 row(s) affected	0.235 sec
6	12:10:17	Describe student	3 row(s) returned	0.203 sec / 0.000 sec

Result Grid

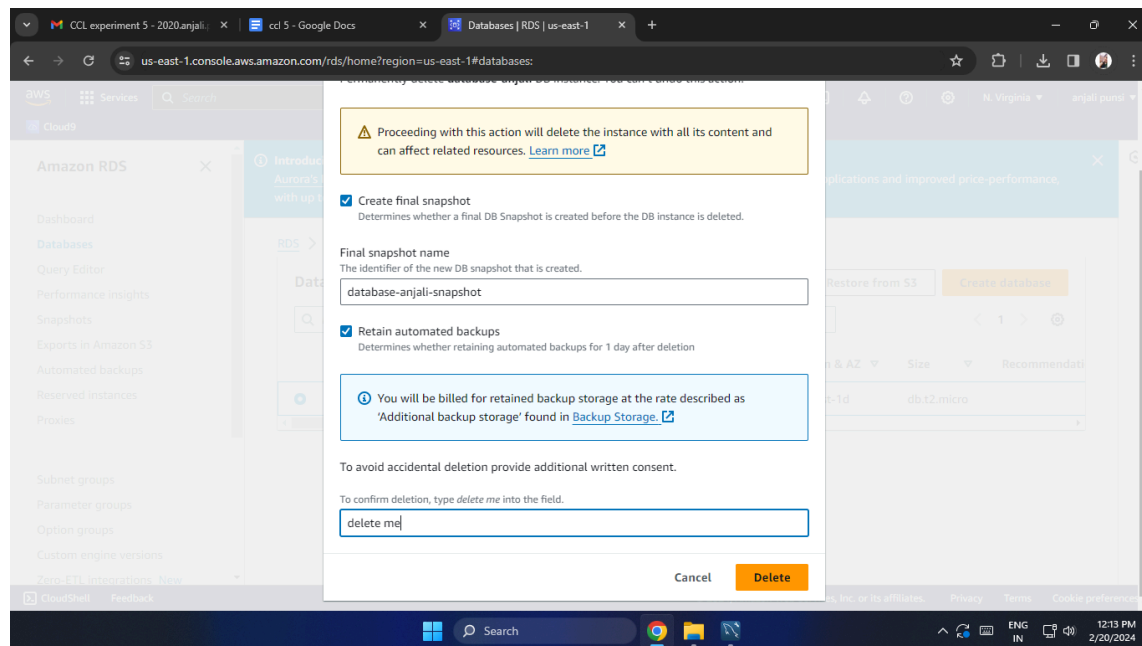
Field	Type	Null	Key	Default	Extra
roll	int	YES		NULL	
name	varchar(10)	YES		NULL	
city	varchar(10)	YES		NULL	



Step 24:

Now delete the instance (once you have done with it)

Select instance go to action stop instance and then delete instance



Conclusion :-

We have successfully studied and implemented Database as a Service on SQL/NoSQL databases like AWS RDS, AZURE SQL/ MongoDB Lab/ Firebase