AAW

AWS ADVANCED PROJECT-2

**Deploying Amazon RDS Multi-AZ and Read Replica, Simulate Failover**

ARAVIND G KUMAR

**Introduction:**

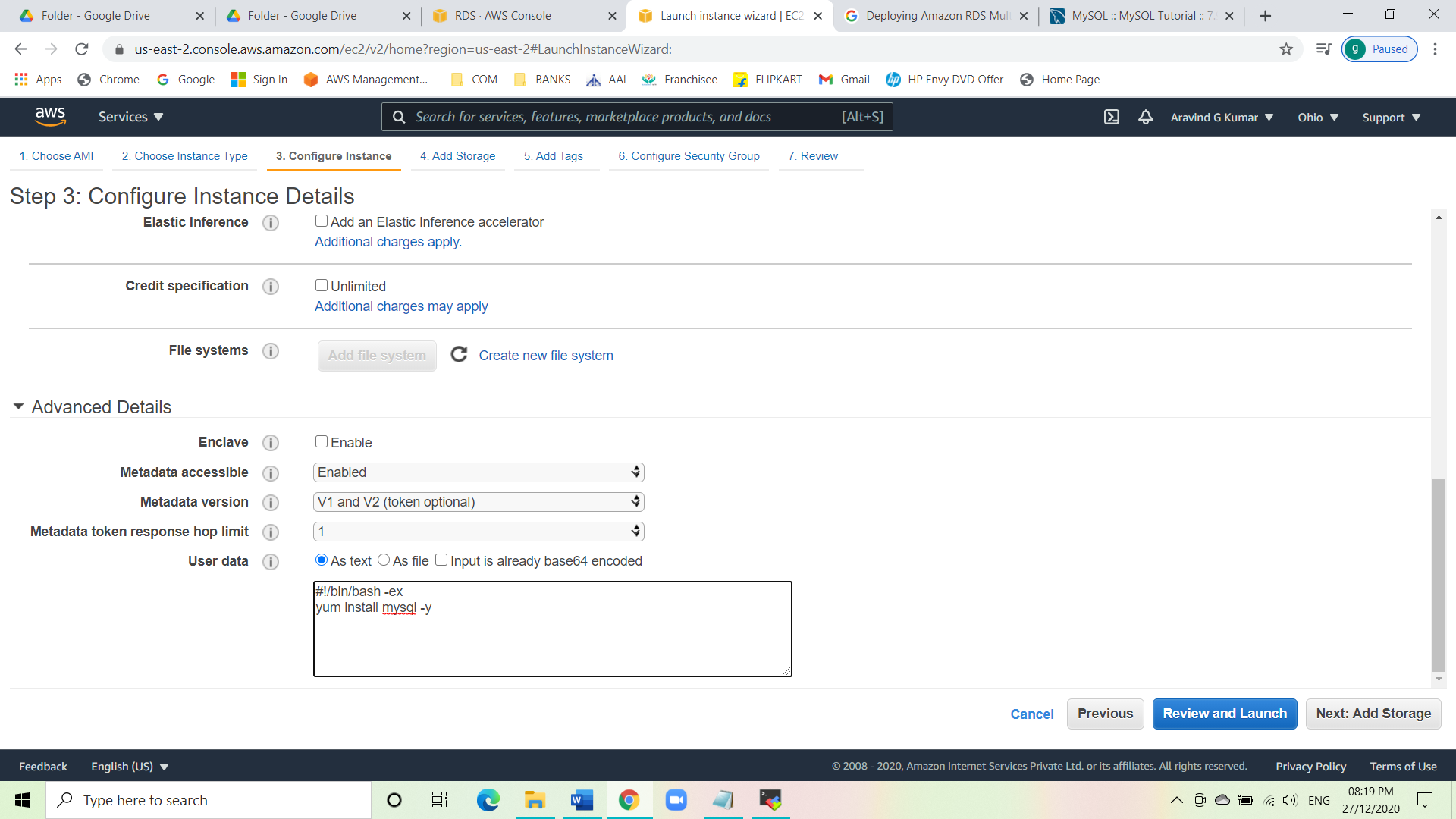
**Amazon RDS** Multi-AZ deployments provide enhanced availability and durability for RDS database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different [Availability Zone](https://aws.amazon.com/about-aws/global-infrastructure/regions_az/#Availability_Zones) (AZ). Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable. In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby (or to a read replica in the case of [Amazon Aurora](https://aws.amazon.com/rds/aurora/)), so that you can resume database operations as soon as the failover is complete. Since the endpoint for your DB Instance remains the same after a failover, your application can resume database operation without the need for manual administrative intervention.

In order to deploy Amazon RDS Multi AZ, we need to create an EC2 instance.

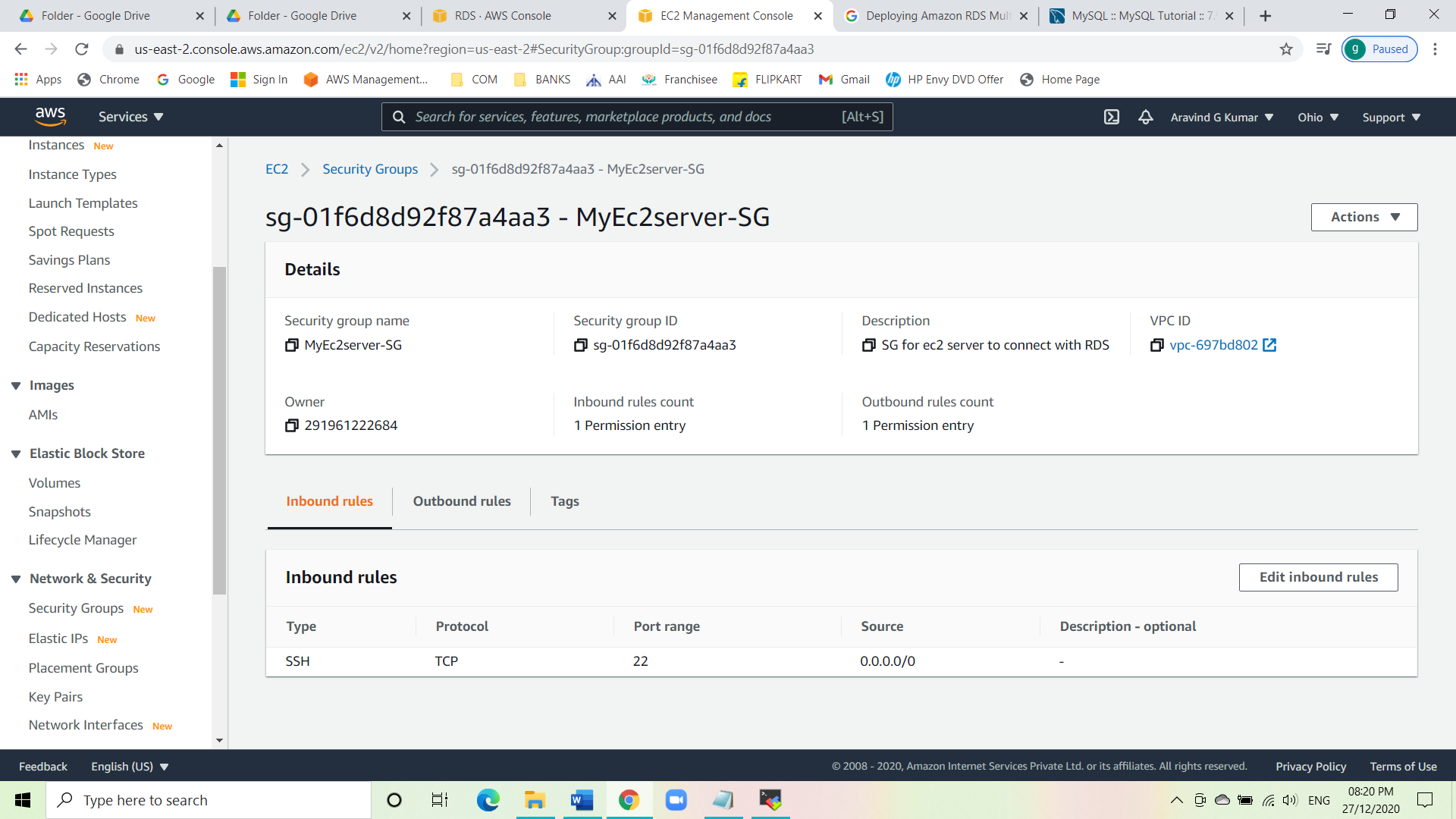
* Create an EC2 instances.
* First create an instance with Linux AMI with the default VPC and subnet attached to it.
* Click on enable the auto assign Public IP.
* Under the user data section, enter the following script.

#!/bin/bash -ex

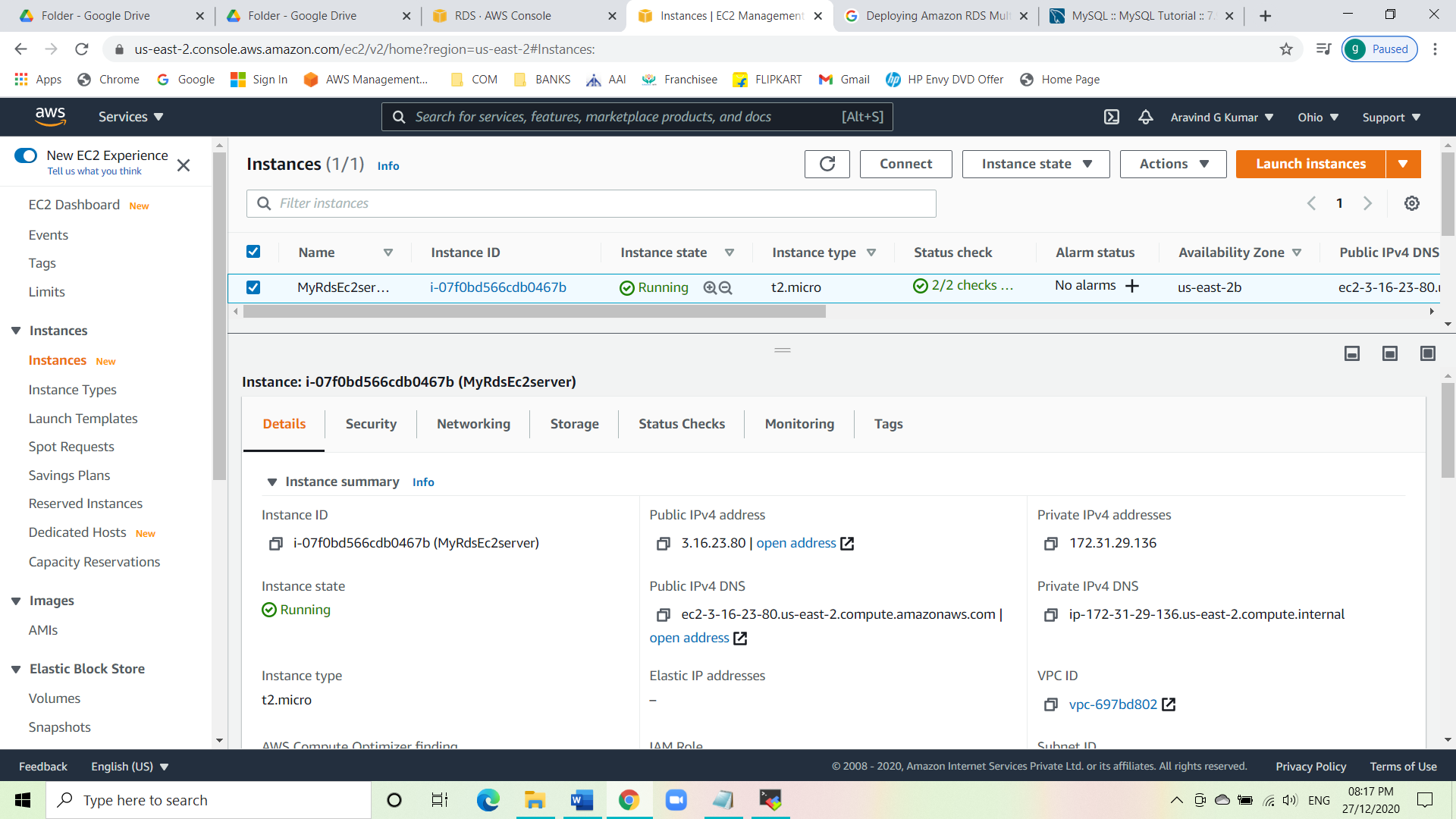
yum install mysql -y



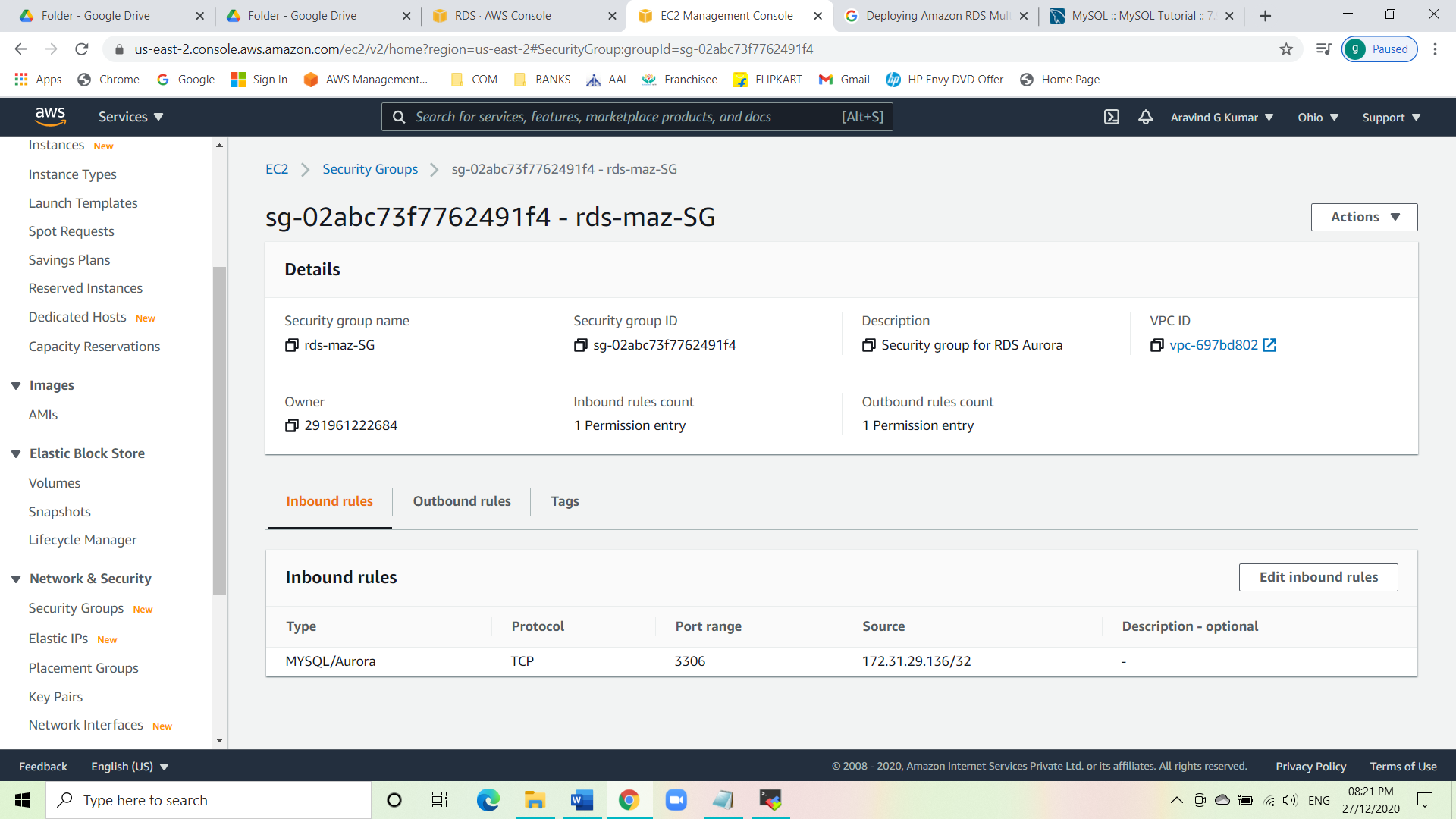
* Give a tag as MyRDSEC2server.
* Now create a security group with the name MYEC2server-SG, and set the type as SSH and source as custom.



* Launch the instance.



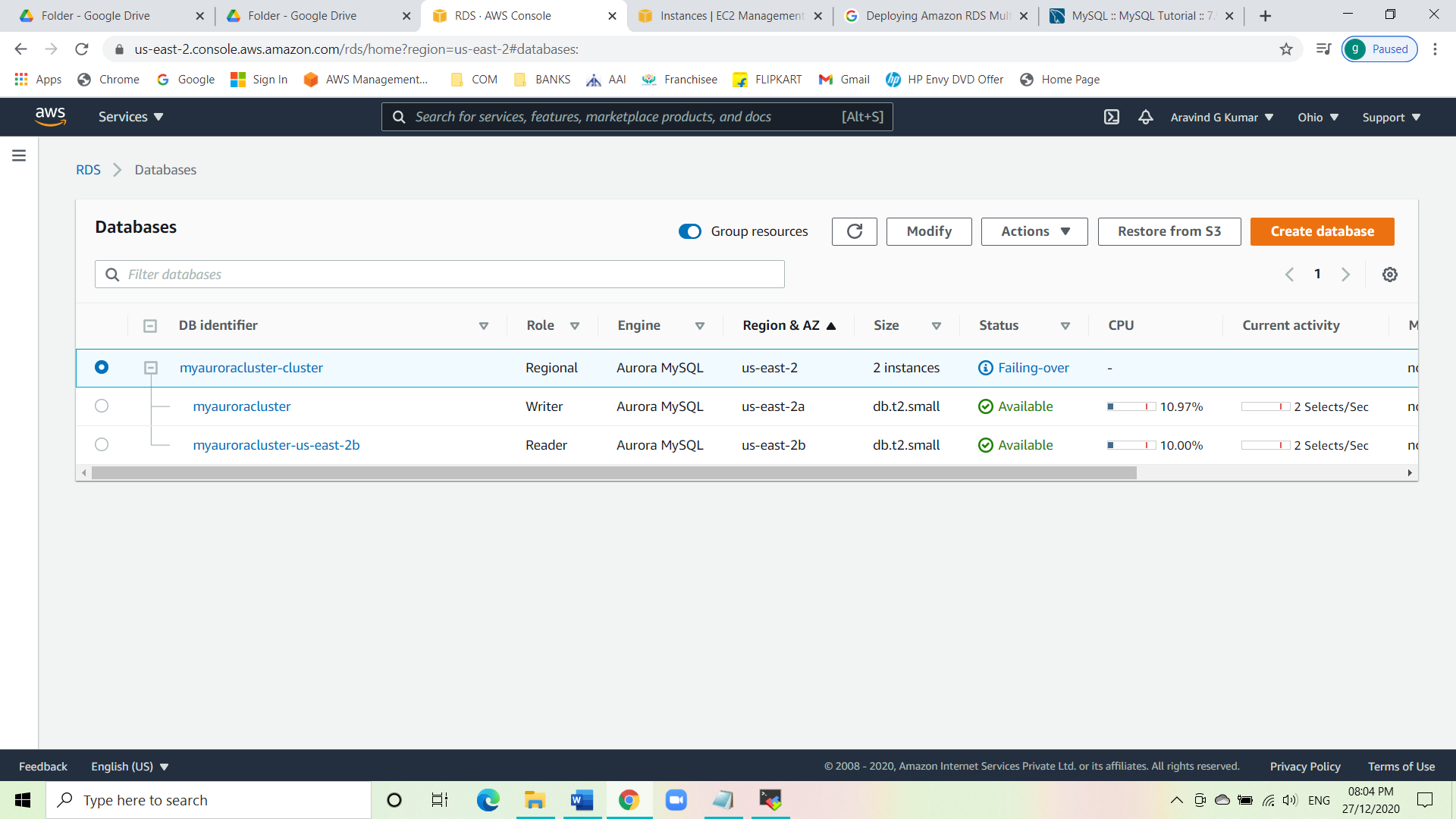
* Now create a SG for RDS instance.
* Give the SG name as rds-maz-SG.
* Under inbound rules select the type as MYSQL/Aurora and the source as custom.
* Create the SG.



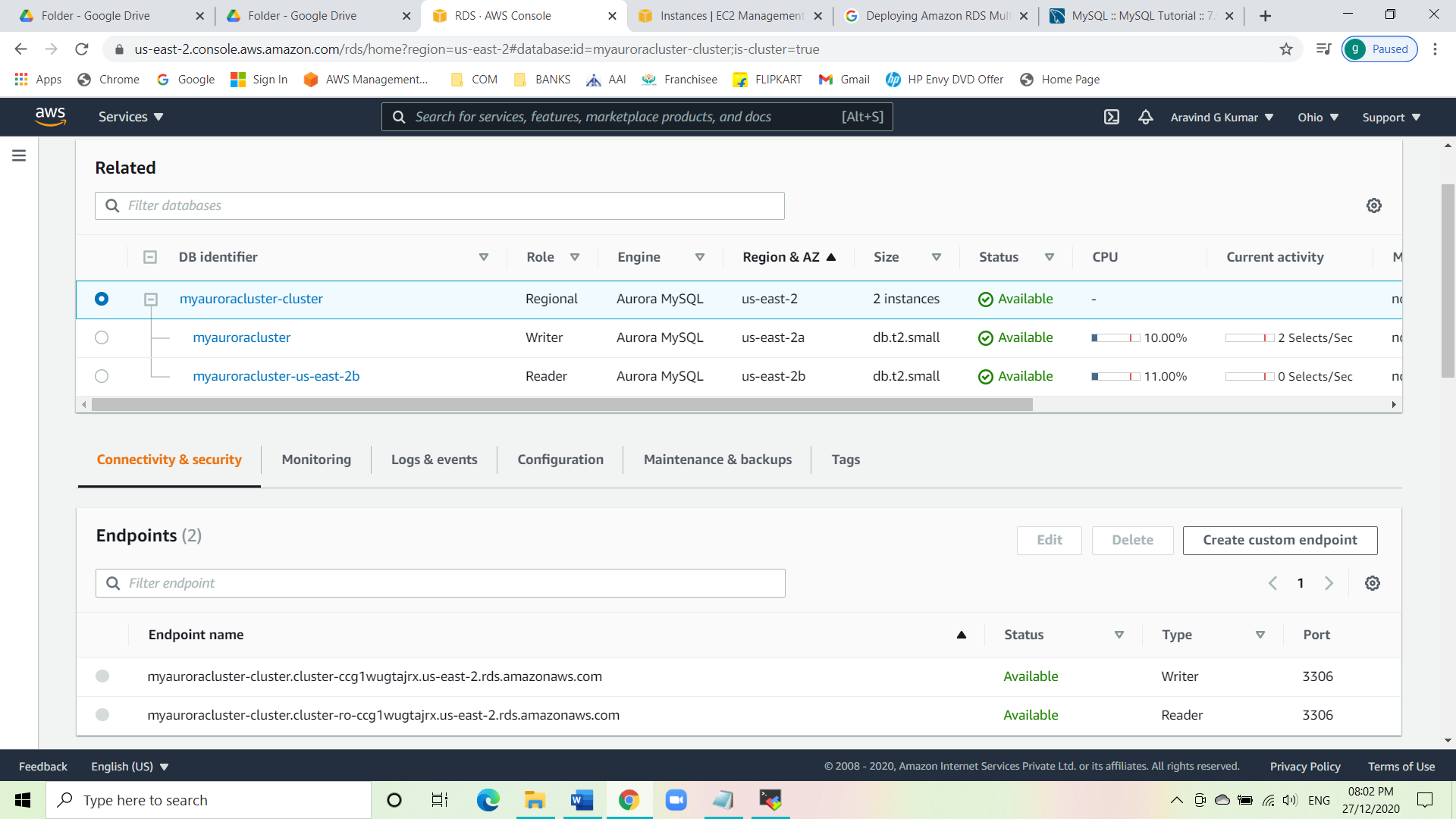
* Creating Amazon RDS database with Multi-AZ enabled.
* Navigate to RDS under the service menu.
* Click on create database.
* Select the engine type as Amazon Aurora and edition as MYSQL 5.7.
* Give the DB cluster identifier name as **MyAuroraCluster**.
* Give the Master name as labsAdmin and Master pass as labs12345.
* While choosing the DB instance size choose db.t2.small.
* Set the replication feature as default.



* Set everything as default under the connectivity tab.
* Create the database.



* In order to connect the Aurora DB, we need the endpoints.
* Click on the RDS cluster name and the navigate to connectivity & security to find the endpoints.
* There you can see the endpoints of both the writer and reader.

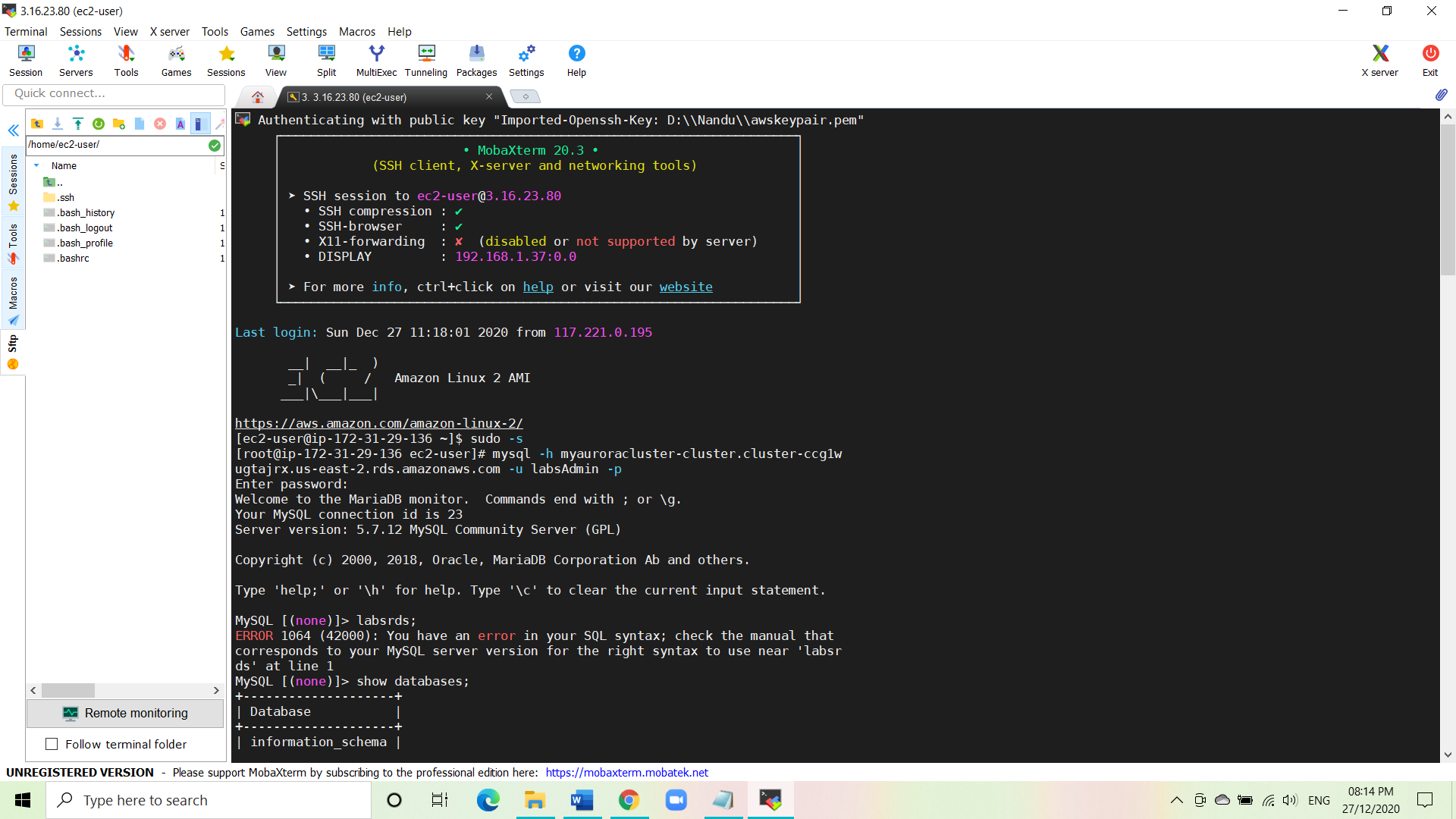


* Connecting the EC2 Server to RDS.
* Navigate to RDS and click on the writer database and click on the SG name i.e., rds-maz-SG under VPC SG.
* Click on the inbound rule and edit the existing source and copy paste the EC2 instance Private IPV4.
* Execute database operations via SSH.
* SSH into the EC2 instance using Mobaxterm. Copy the Public IPV4 and connect.
* Switch to the root user using the command:

sudo -s

* Log into RDS using the command:

mysql -h <endpoint of writer> -u <labsAdmin> -p

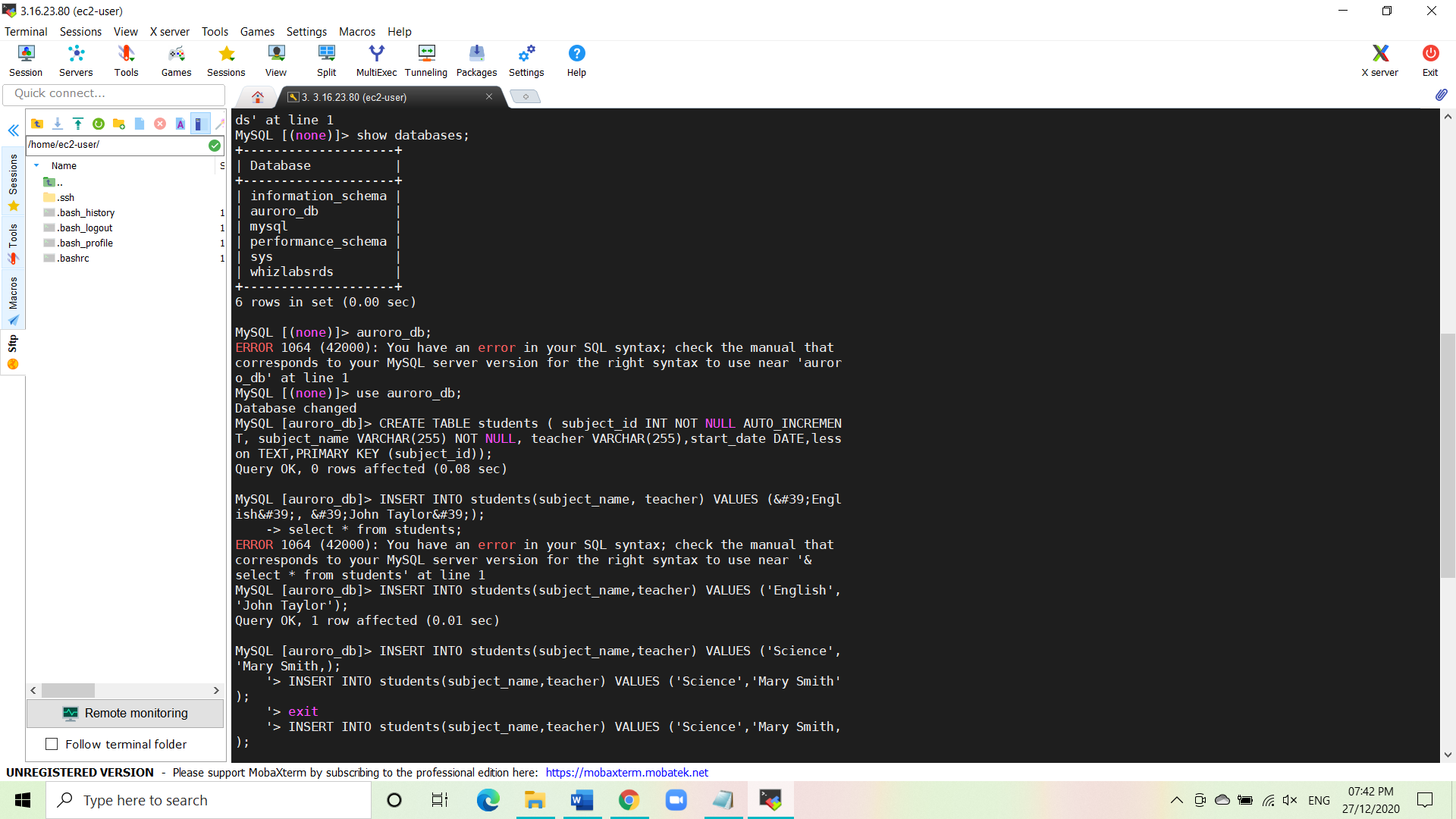


* Now give the password: labs12345.
* To list all database, give the command:

show databases;

* Now create a new database named auroro\_db by using the below command:

create database auroro\_db;



* In order to select the newly created DB, use the given command:

use auroro\_db;

* Now we’ll create a table named Students and insert few rows of data using a list of commands:

CREATE TABLE students ( subject\_id INT AUTO\_INCREMENT, subject\_name VARCHAR(255) NOT NULL, teacher VARCHAR(255), start\_date DATE, lesson TEXT, PRIMARY KEY (SUBJECT\_ID));

* To insert the data into the table, give the following commands:

INSERT INTO students(subject\_name,teacher) VALUES (‘English’,’John Taylor’);

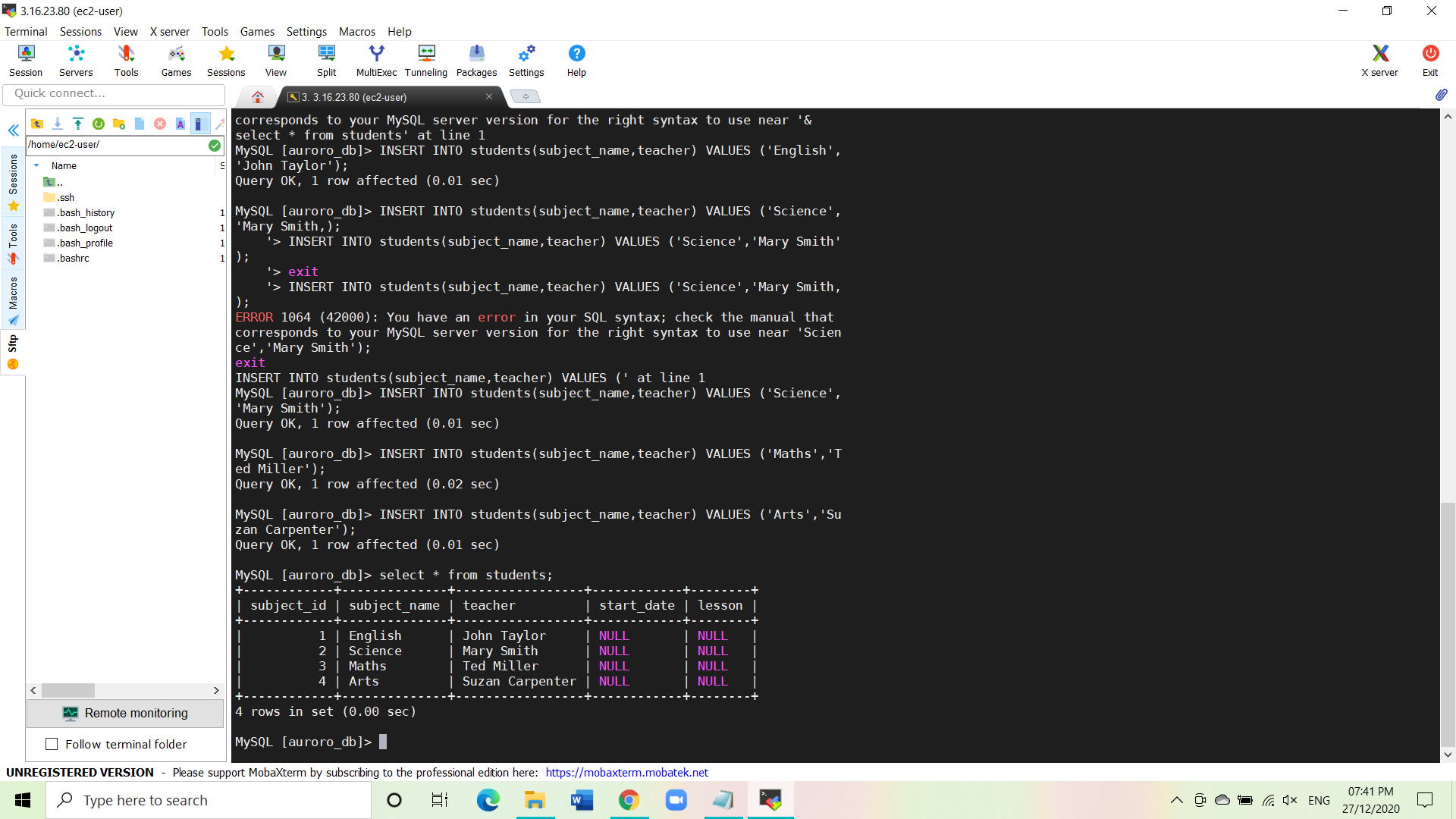
INSERT INTO students(subject\_name,teacher) VALUES (‘Science’,’Mary Smith’);

INSERT INTO students(subject\_name,teacher) VALUES (‘Maths’,’Ted Miller’);

INSERT INTO students(subject\_name,teacher) VALUES (‘Arts’,’Suzan Carpenter’);

* Now to view the contents of the table give the following command:

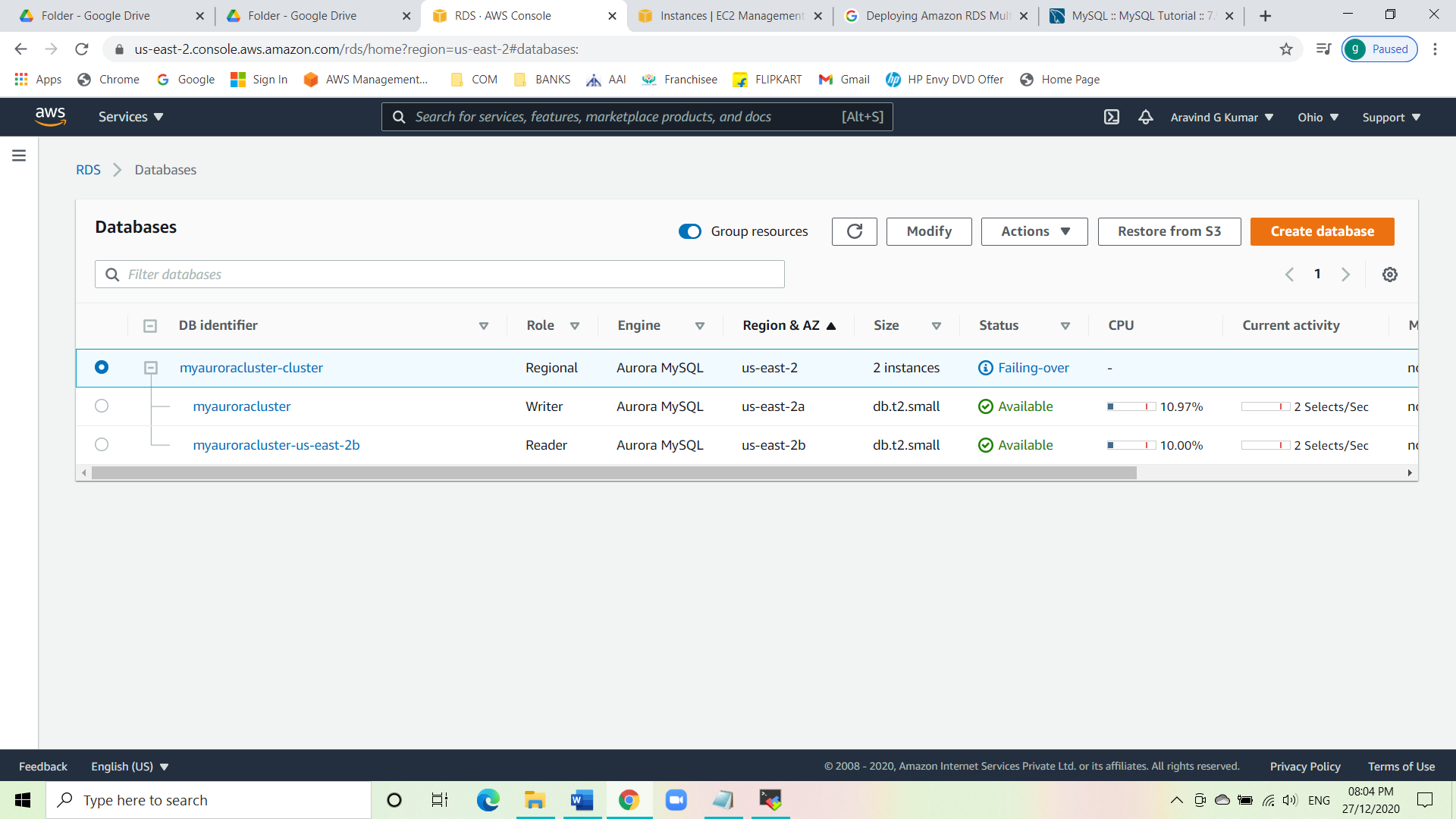
select \* from students;



* To exit from the mysql console give this command:

exit

* Forcing a Failover to test Multi-AZ.
* To test if Multi-AZ is working, we will failover the writer by clicking on the writer and select failover.



* Wait for a few minutes for the RDS to failover.
* Testing the Failover Conditions.
* Now connect to RDS with the new writer’s endpoint:

mysql -h <newwriterendpoint> -u <labsAdmin> -p

* Give the password: labs12345.
* Now you will be able to log into MYSQL and check the failover worked successfully by giving the command:

show databases;

* Now to use the newly created database give the below command:

use auroro\_db;

* Now to check the existence of the table named Students and the data, give the following commands:

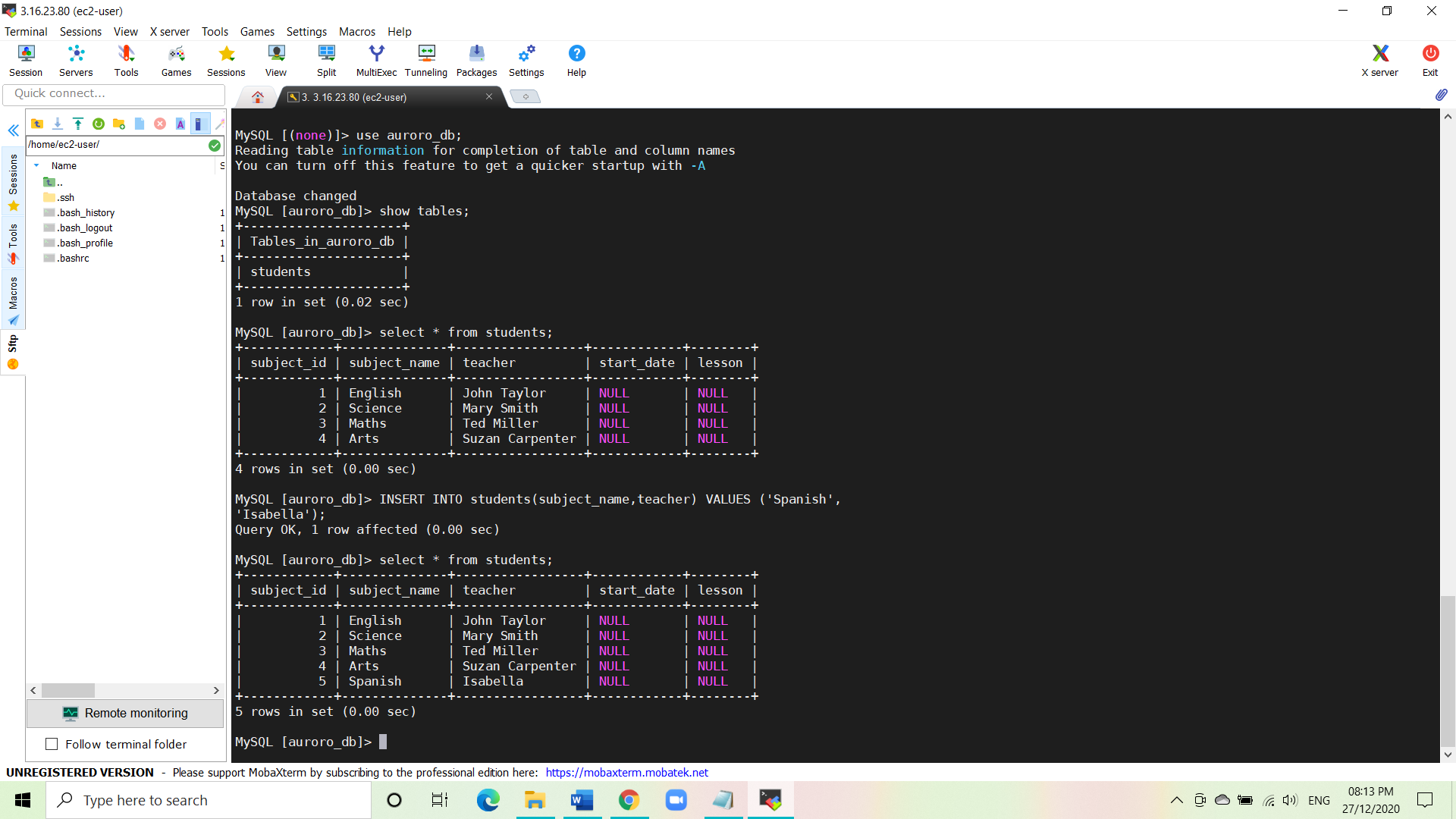
show tables;

select \* from students;

* Now insert one more row into the table by the following command:

INSERT INTO students(subject\_name,teacher) VALUES (‘Spanish’,’Isabella’);

Select \* from students;



**Conclusion:**

We have launched Amazon RDS Multi-AZ. We were able to SSH into the EC2 instance successfully. Then we created a table and added data into it. Successfully tested Failover. Using the new writer’s endpoint, we were able to see the tables and add the new data into the same created table.