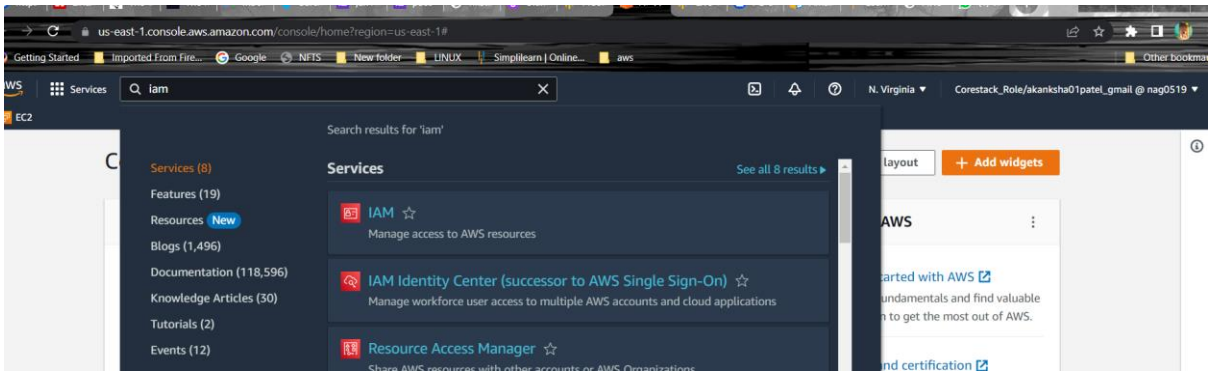
Configure and connect a MySQL Database Instance with a Web Server

To begin with any of the activity we first need to set up IAM i.e., permissions to the users on the AWS resources. So, we will provide RDB full access and EC2instance access to our user.

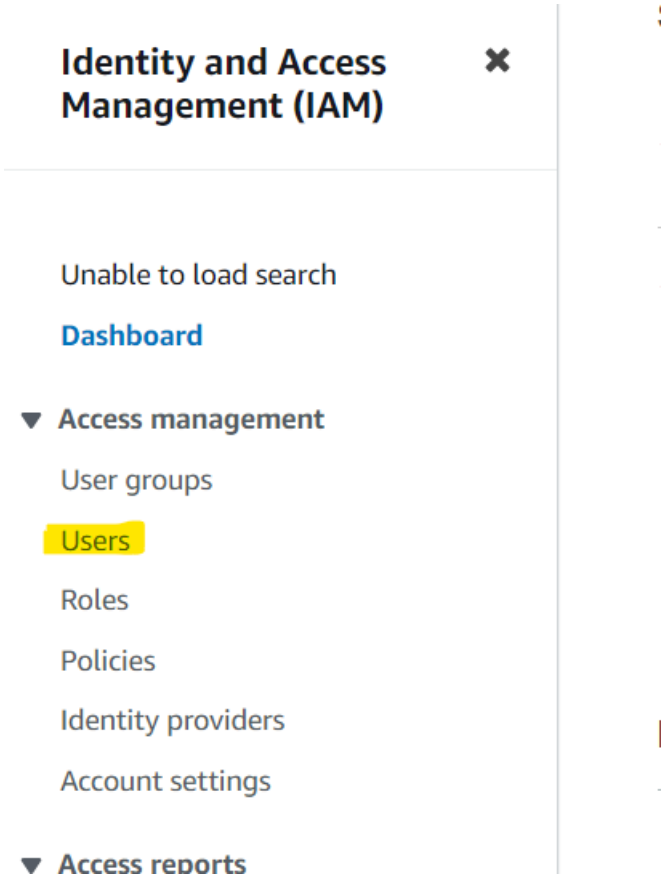
i. Navigate to the AWS lab and search for IAM:



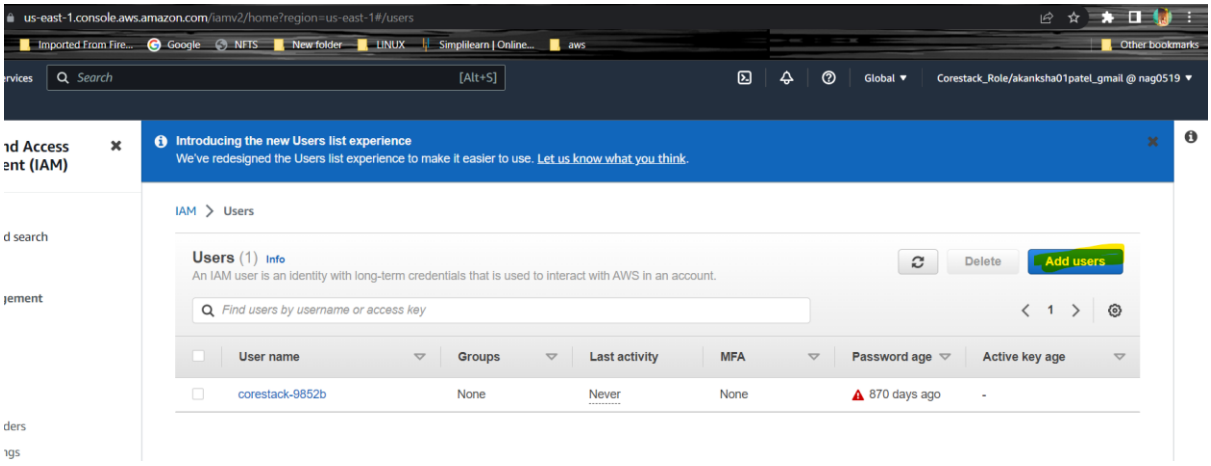
1. Select IAM.
2. IAM dashboard appears.
3. We create a user to whom we will assign Policies.
4. As IAM Policies are global so we don't need to worry of the region we have selected currently.

ii. Creating a user:

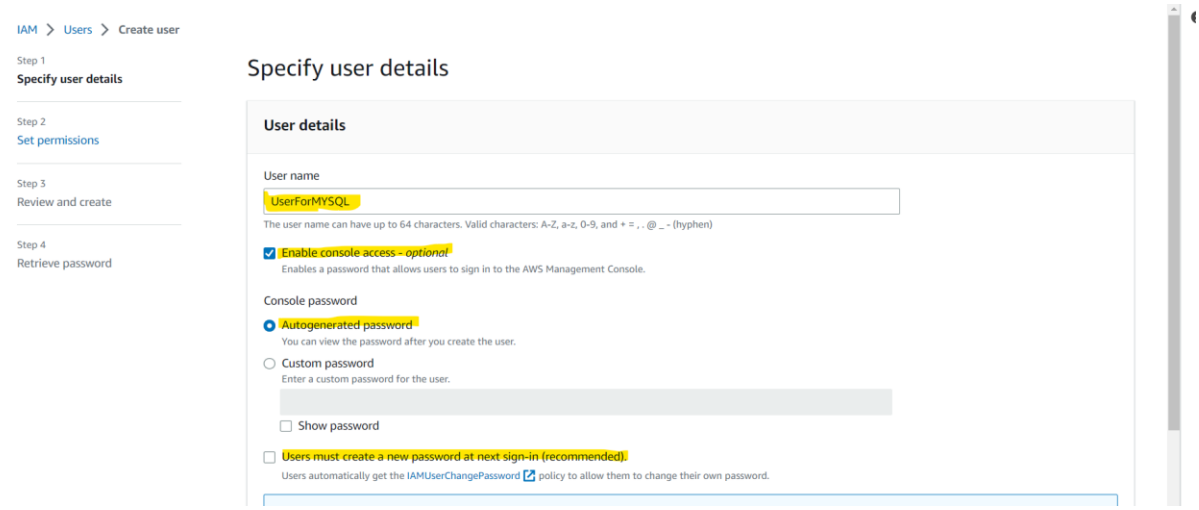
1. Click on Users on left side of IAM dashboard.



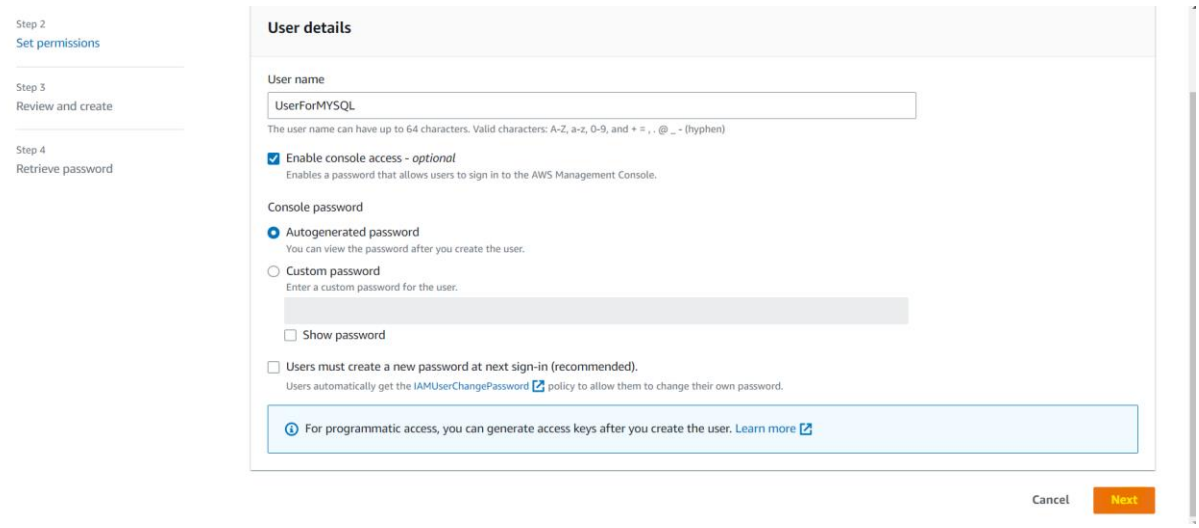
2. Click on Add Users.



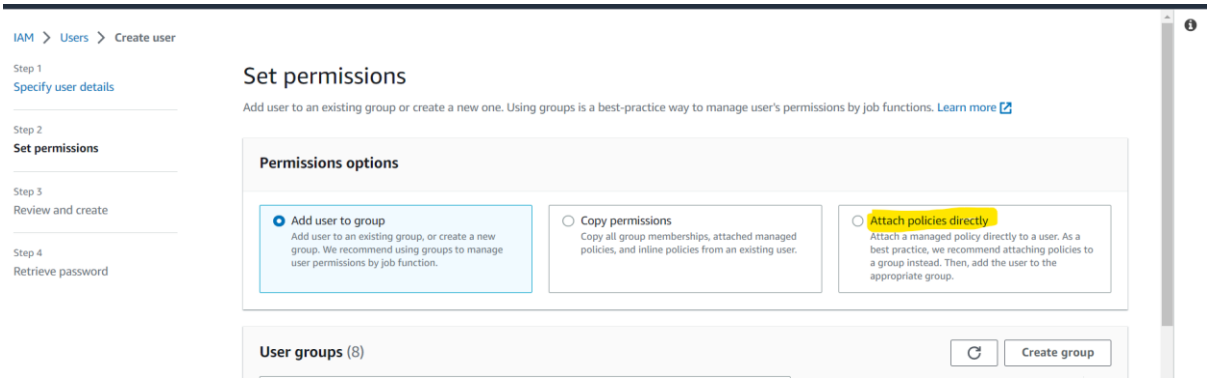
3. Enter a name for the user.
4. Enable console access.
5. Select autogenerated password.
6. And disable the option to create a new password for the user in next-sign.



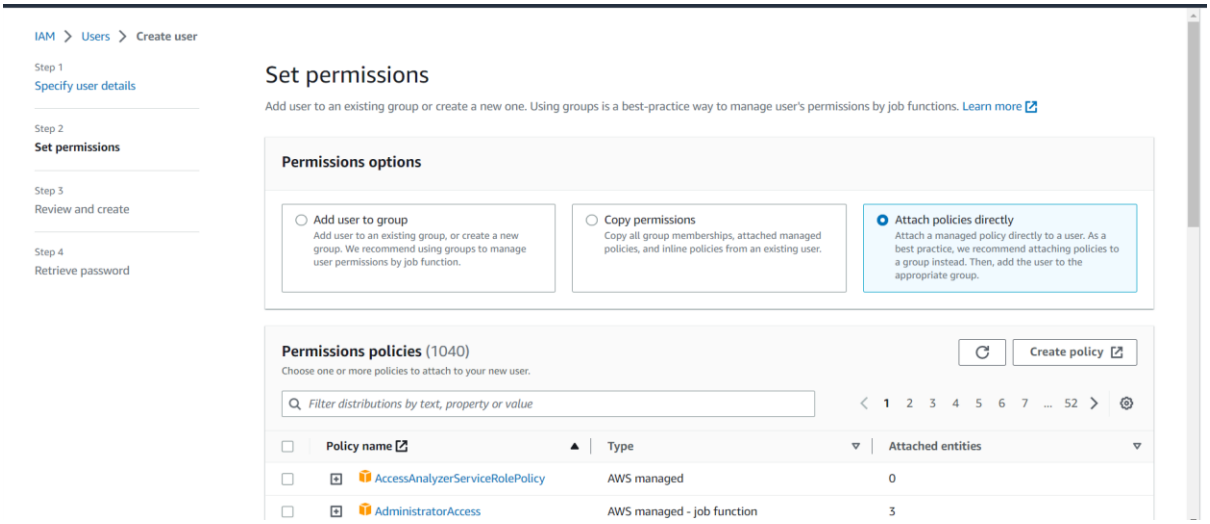
7. Then click Next.



8. Then Set Permissions page comes up. Here we need to select Attach policies directly as we are aware of the policies we need to assign to our user.

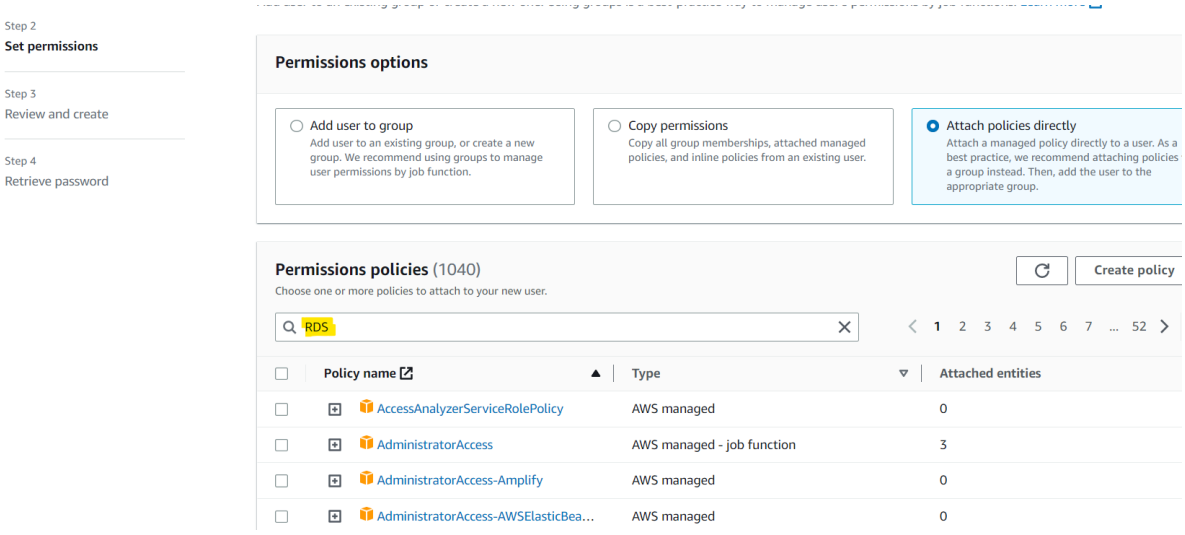


Now we see a page with various Permissions policies to choose from:

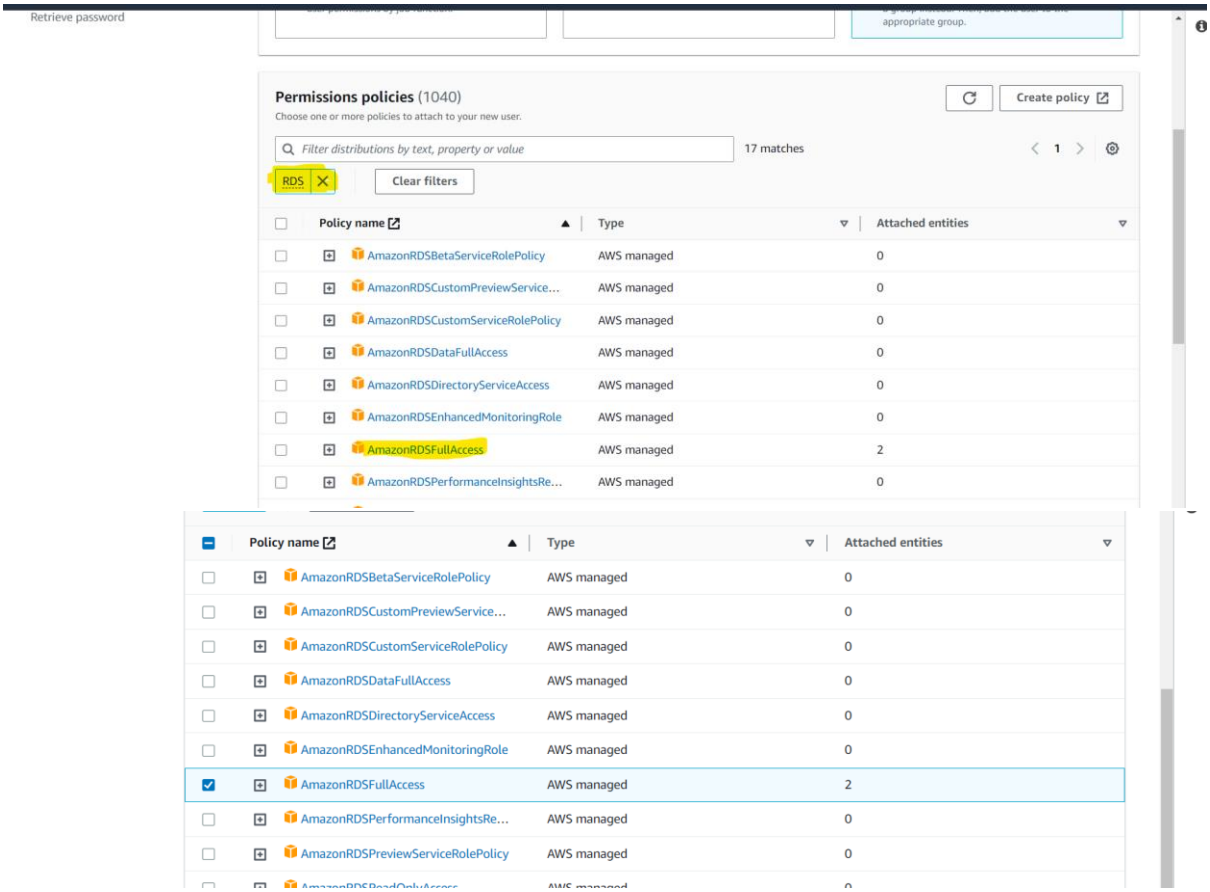


9. Now we need to choose 2 policies :
   1) RDSFullAccess

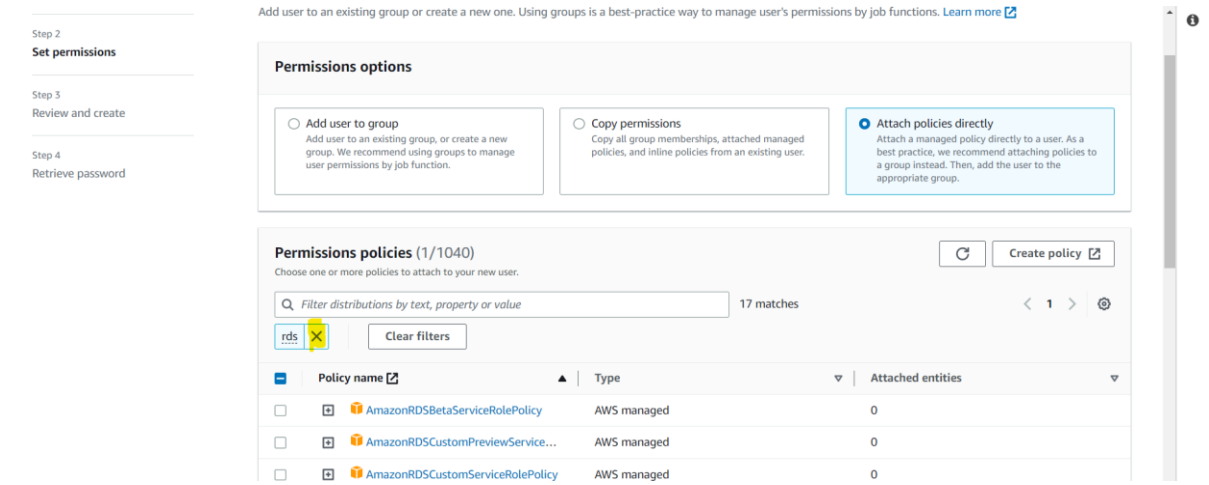
- We type RDS in search and press enter



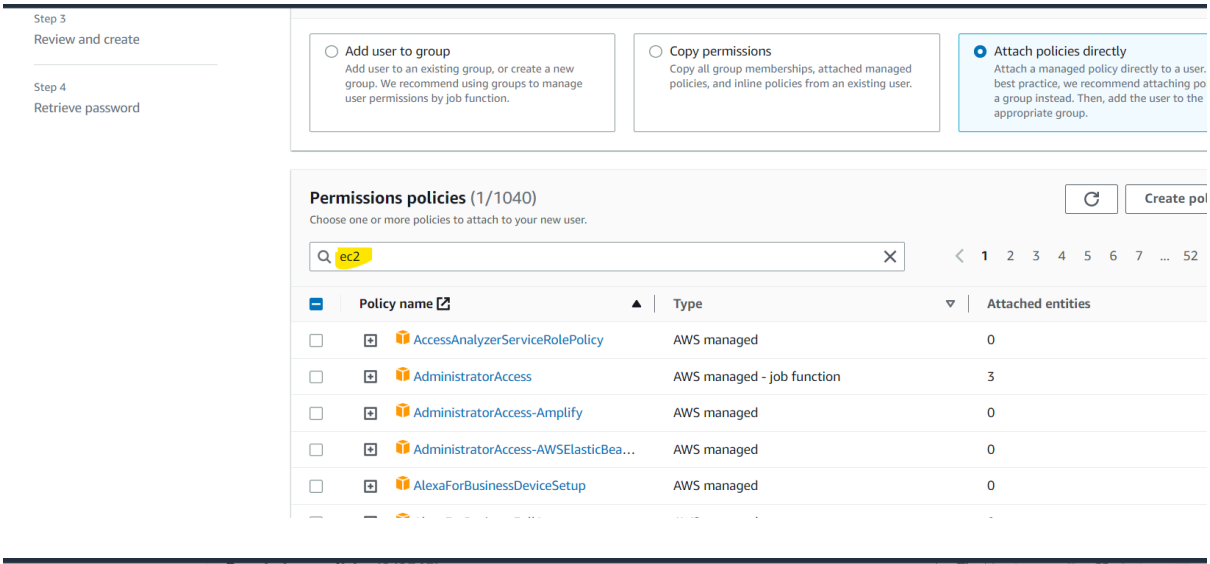
- Now we have a list of all policies related to RDS.
- We select RDSFullAccess Policy

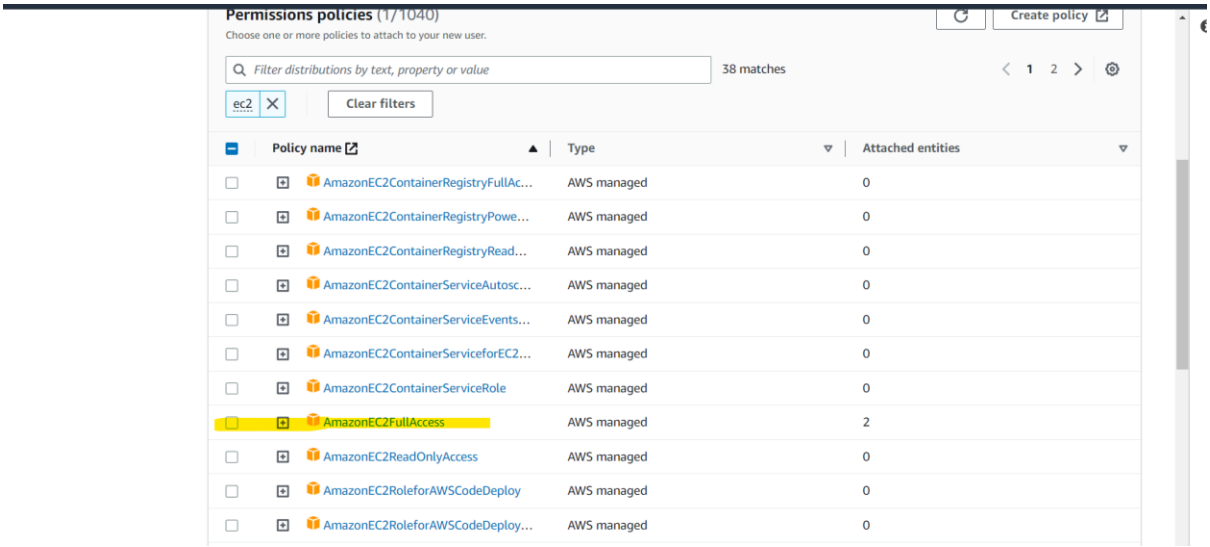


- Now we remove the RDS selected filter by clicking on the cross button .



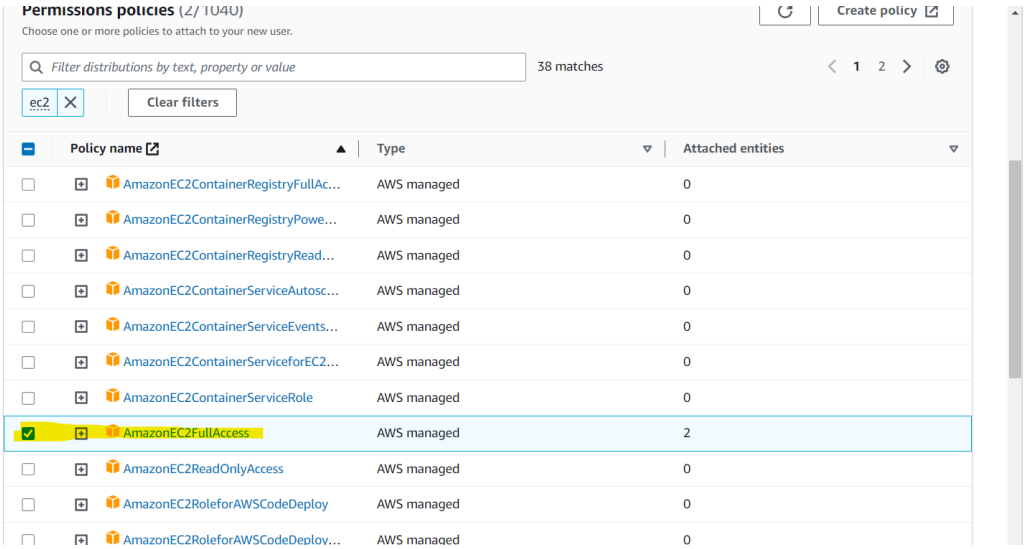
- We now proceed to search for EC2 full access policy by typing EC2 and press enter .



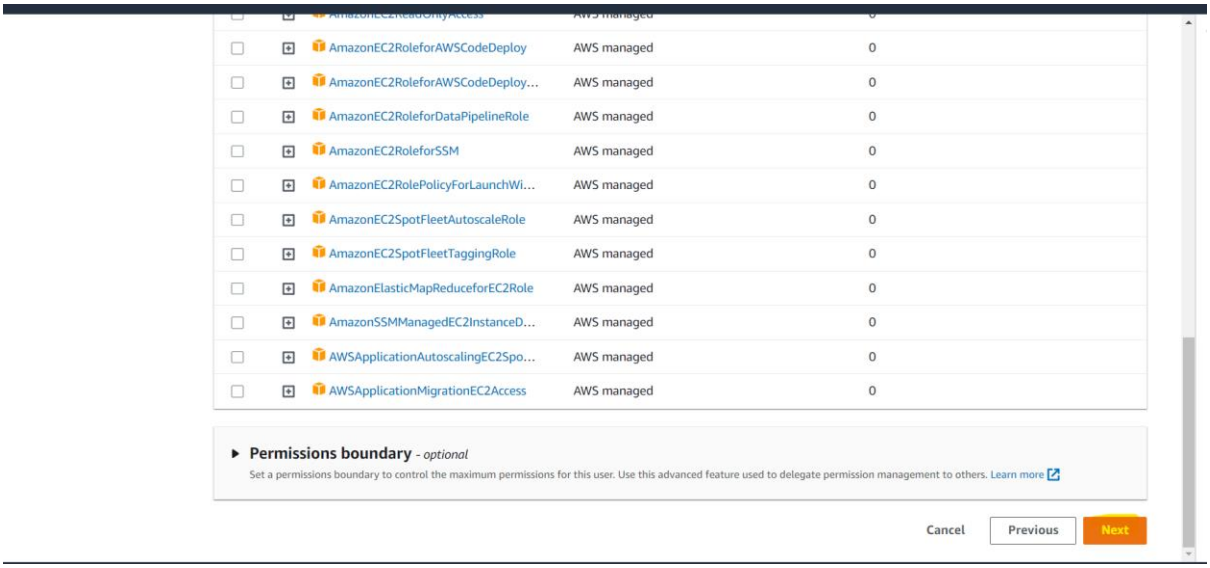


- We need to select the required policy as shown:

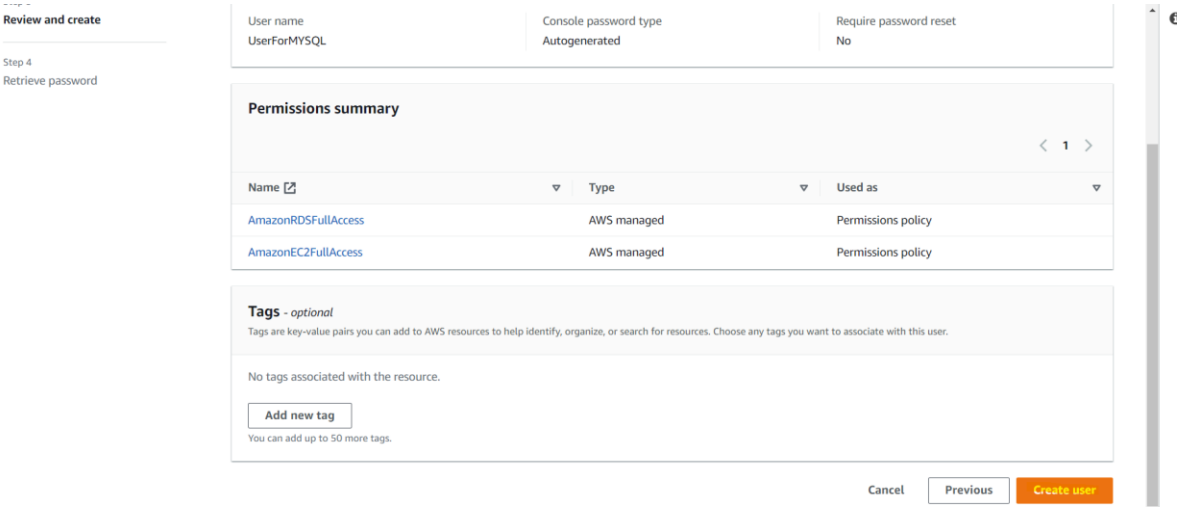
2) EC2FullAccess



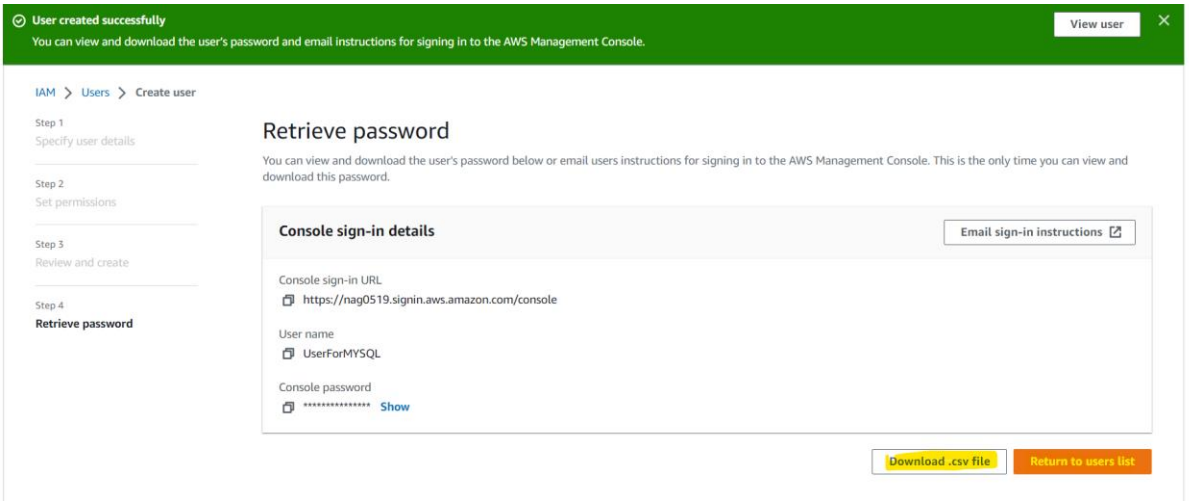
- Now we have selected all of our policies, we will scroll to the bottom and click Next.

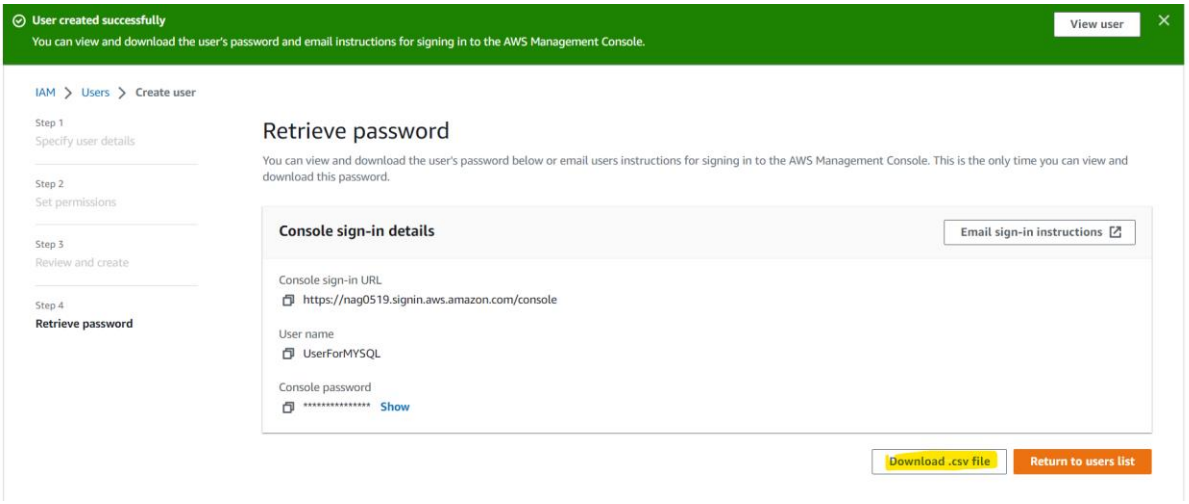


- We now need to review the details we added and click on Create User.

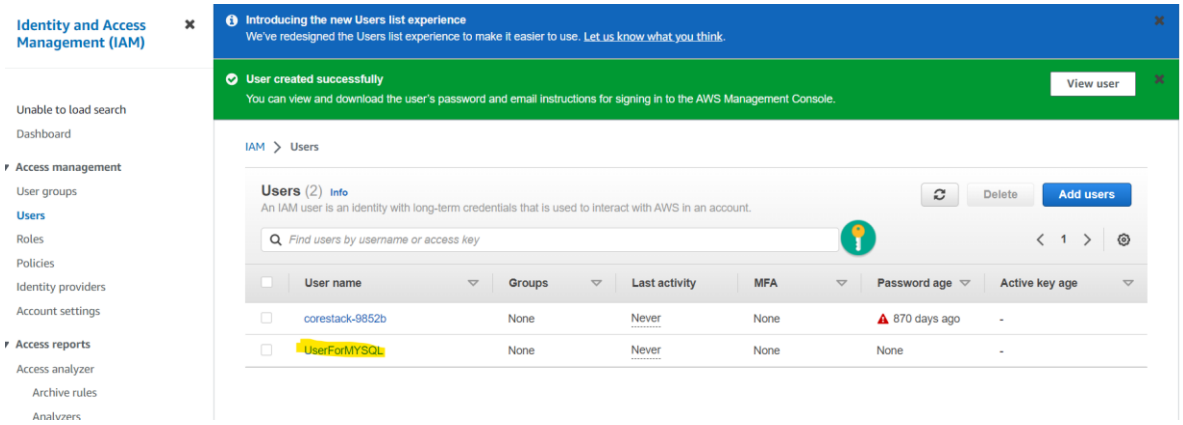


- Once user is created we need to download the csv file that contains the password details for the user to login.
- Then return to user list.





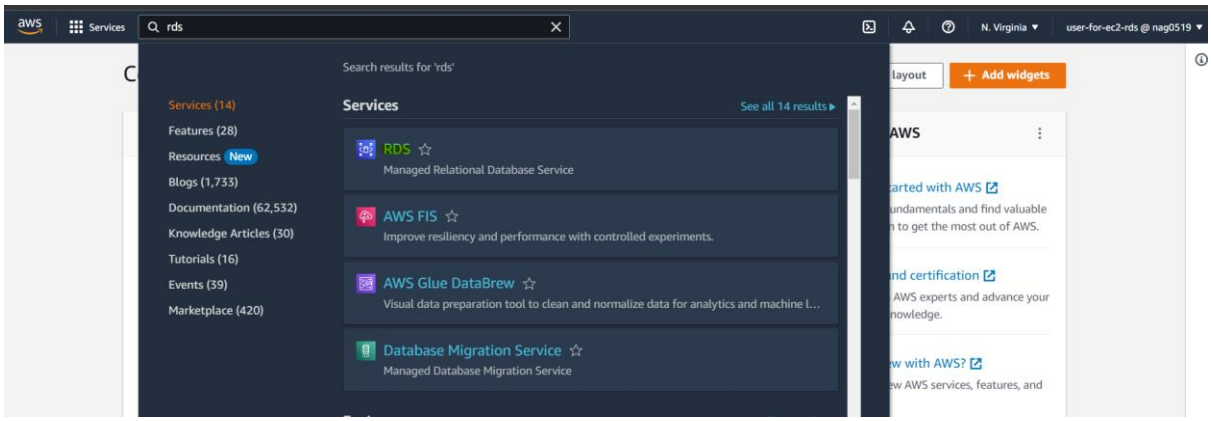
- We can now view the user we just created.



- We will create all our services in default VPC, so our region will be N.Virginia.

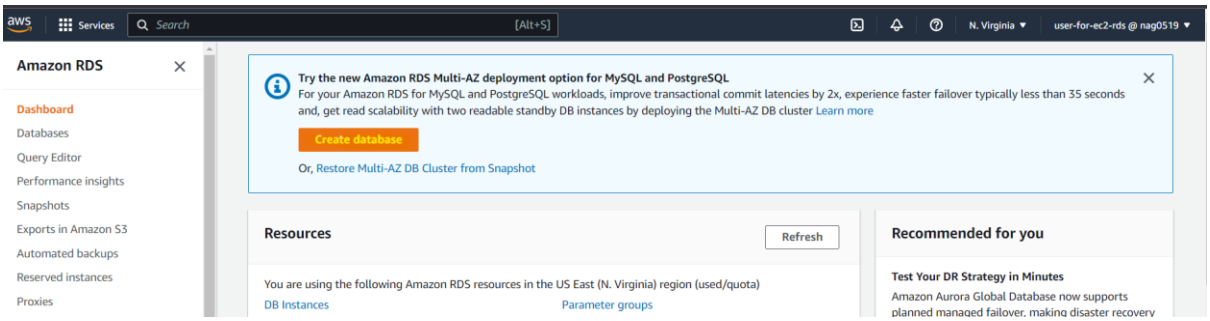
iii. Creating MySQL Database:

1. On Aws dashboard search for RDS:

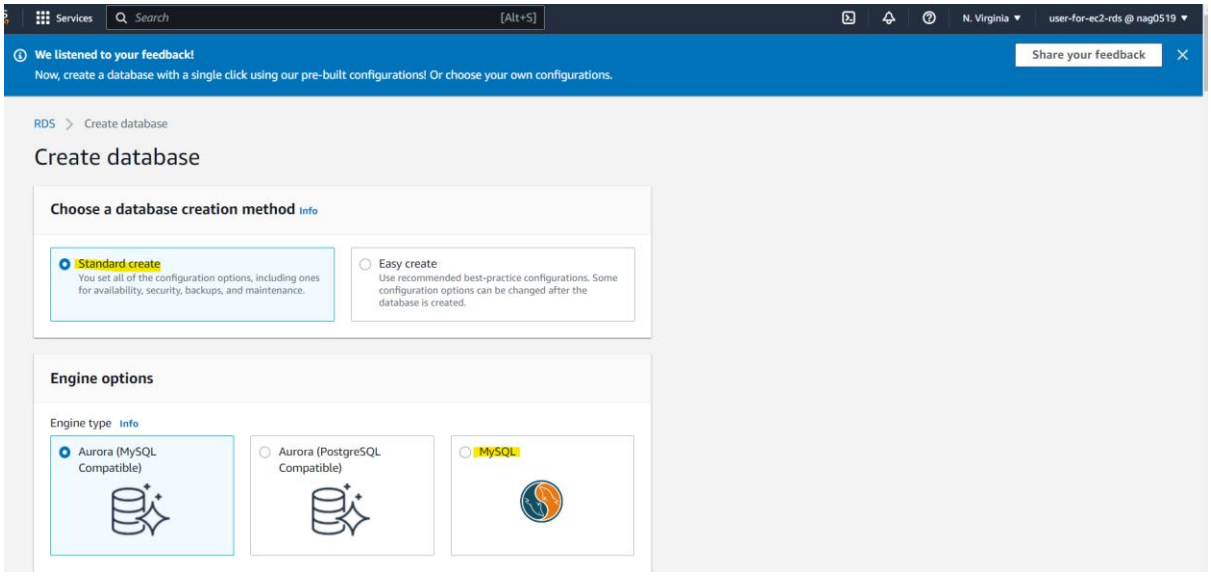


Select RDS, its dashboard opens.

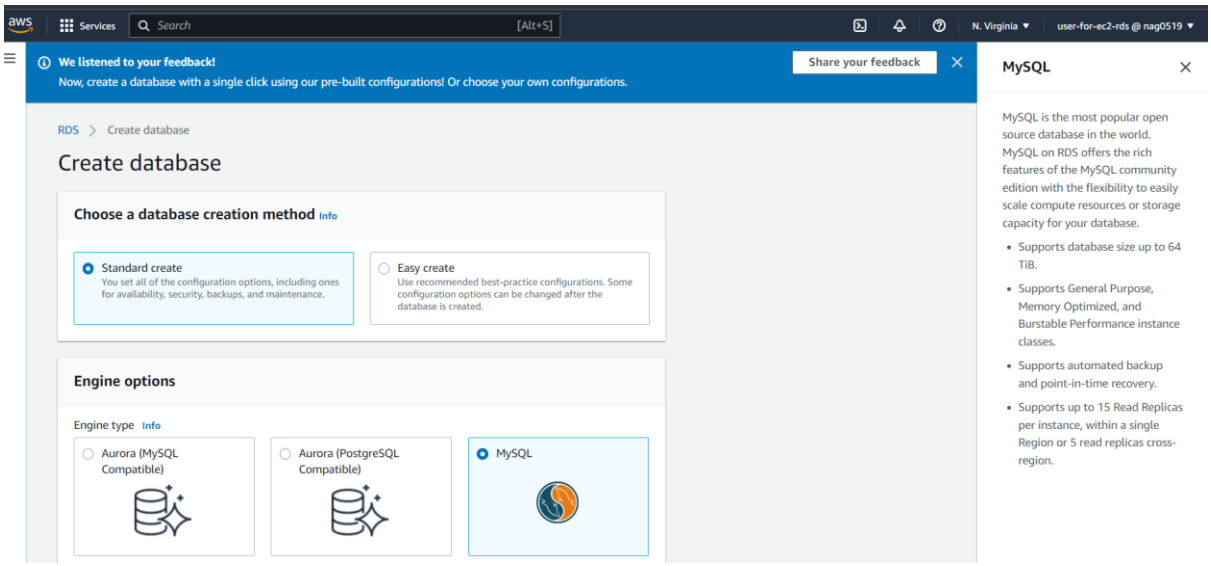
2. Select Create Database

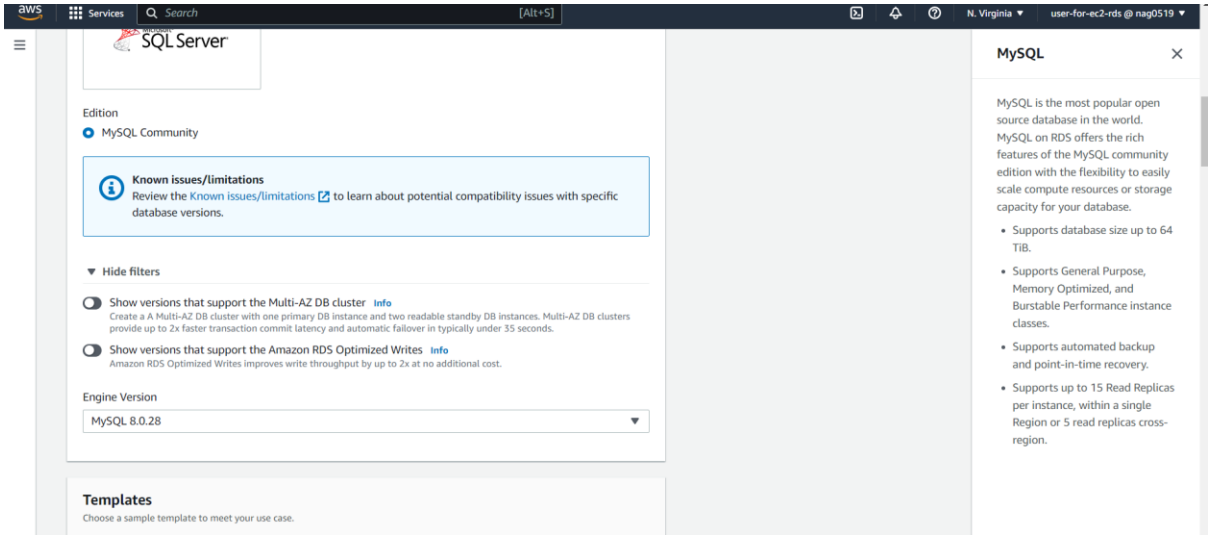


The create database screen appears.

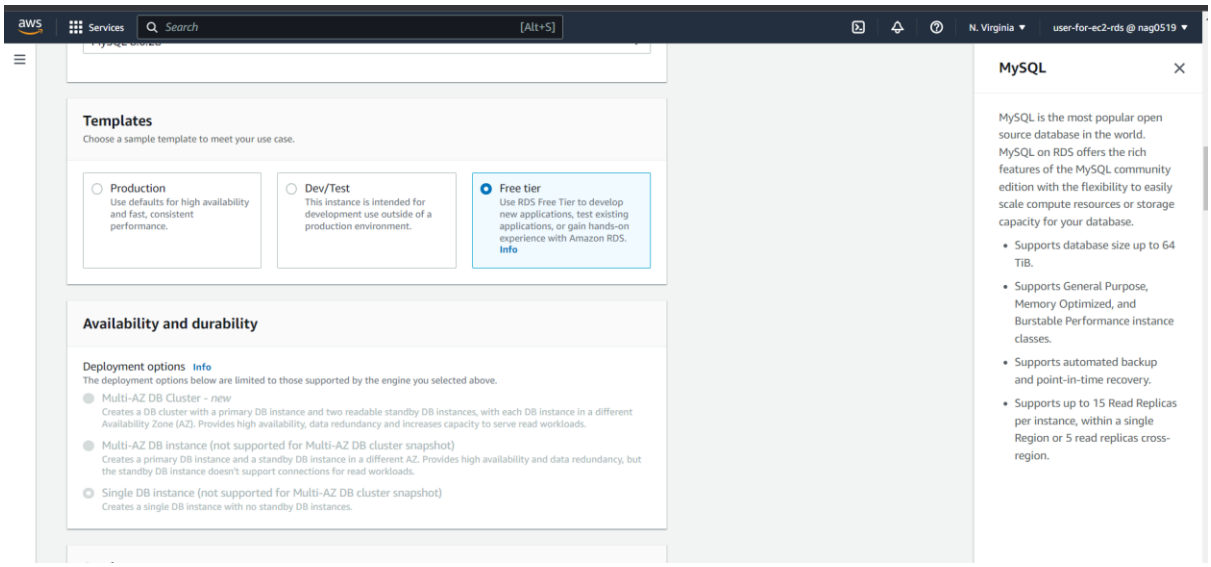


3. Select Standard Database
4. In Engine options choose: MySQL

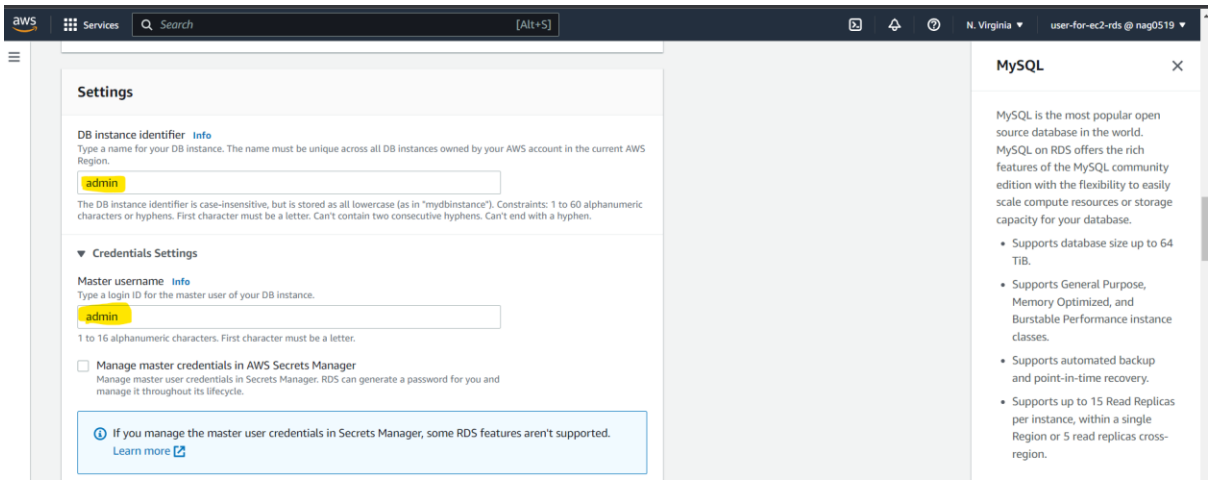




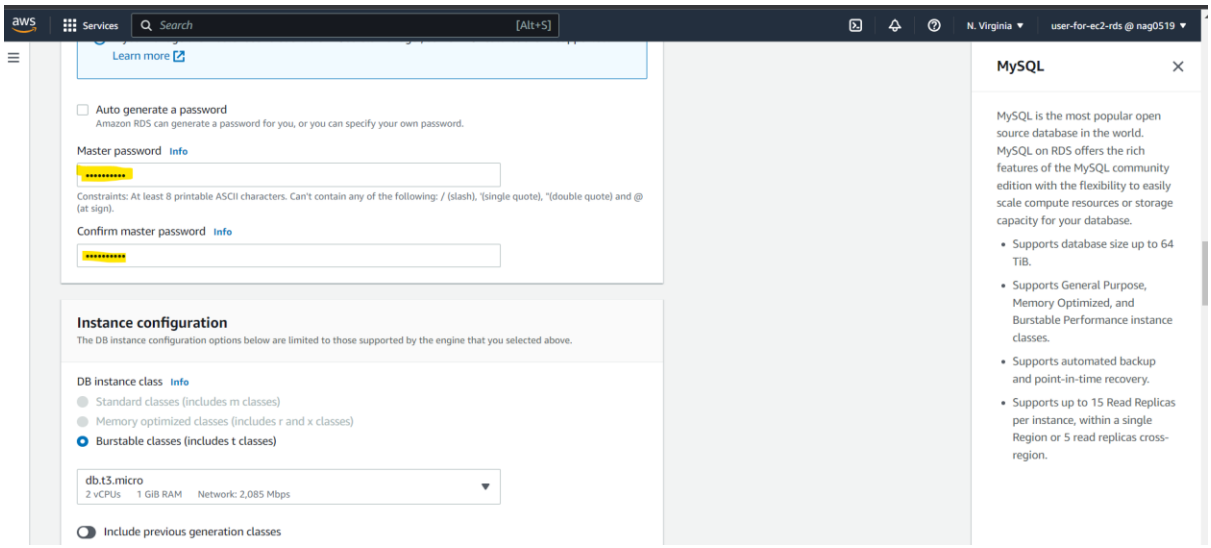
5. In Availability Choose: Free Tier



6. In DB Instance Identifier type a name.
7. In Credential Setting type a username of your choice/requirement.

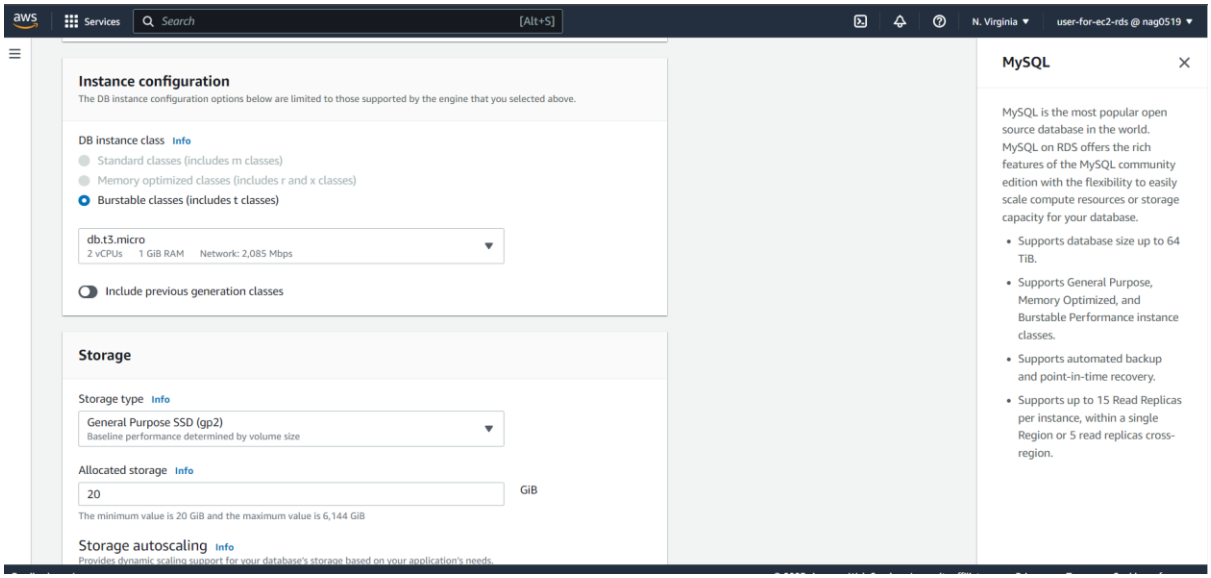


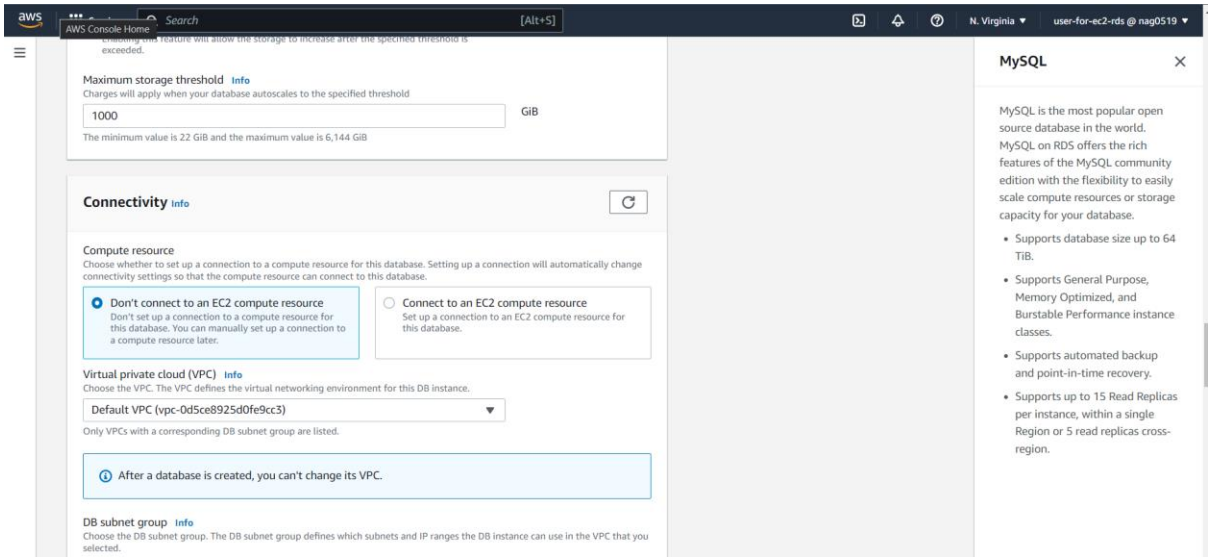
8. Type a password and retype the same



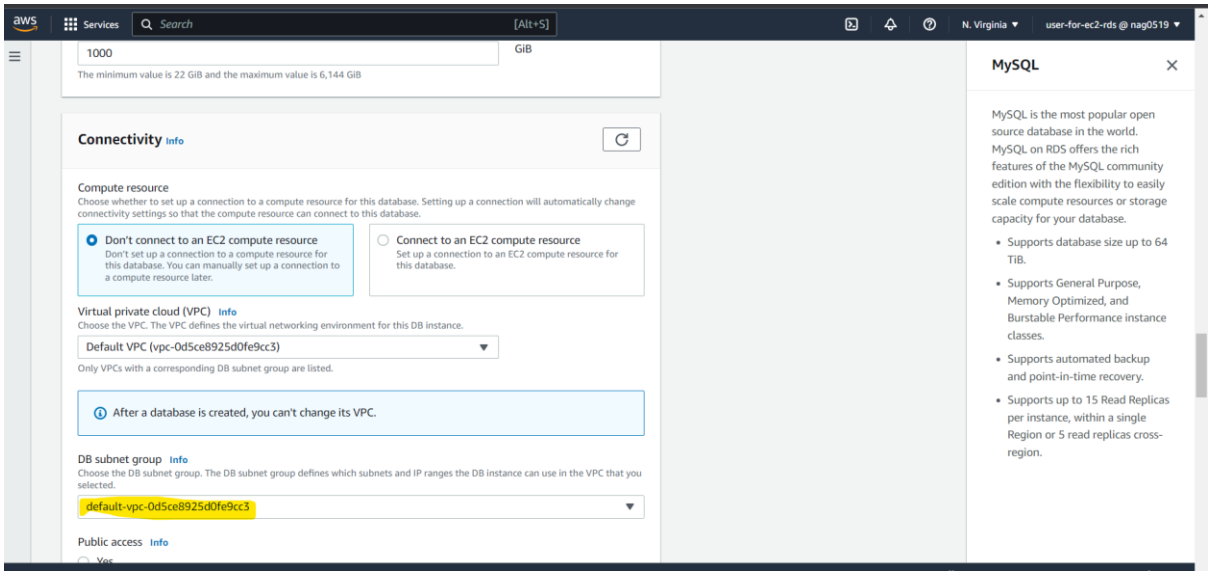
9.

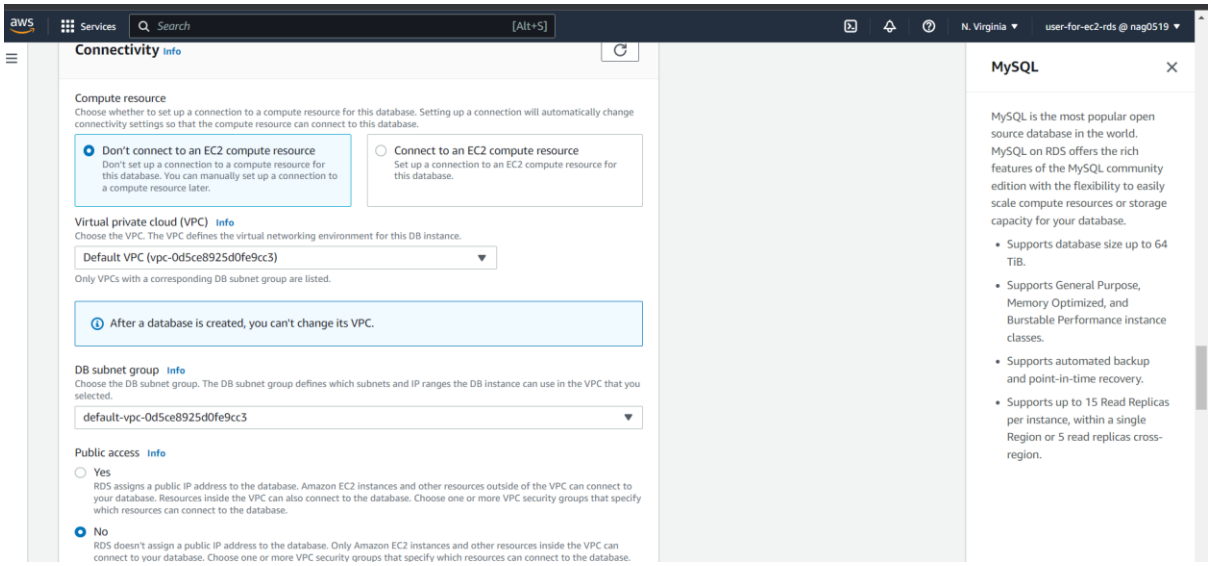
10. Selecting the default configurations

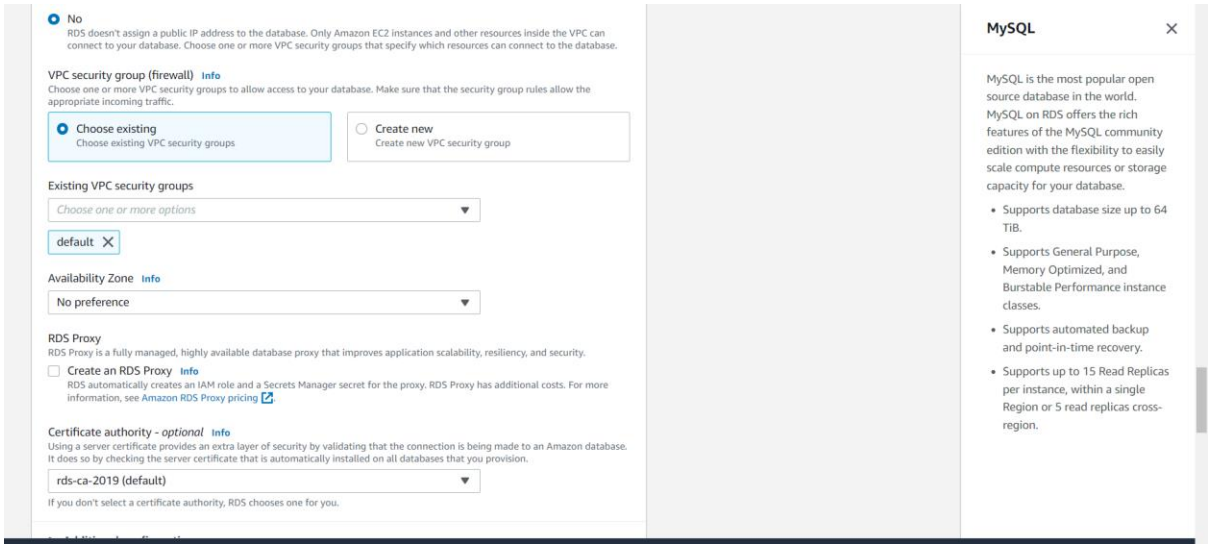


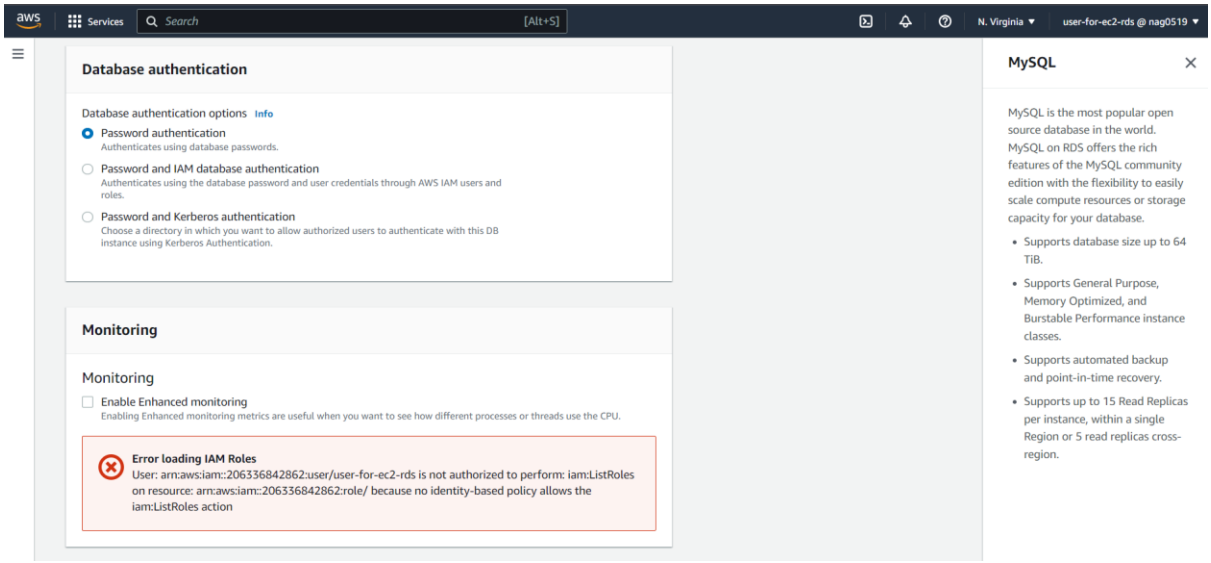


11. Selecting default VPN and proceeding:

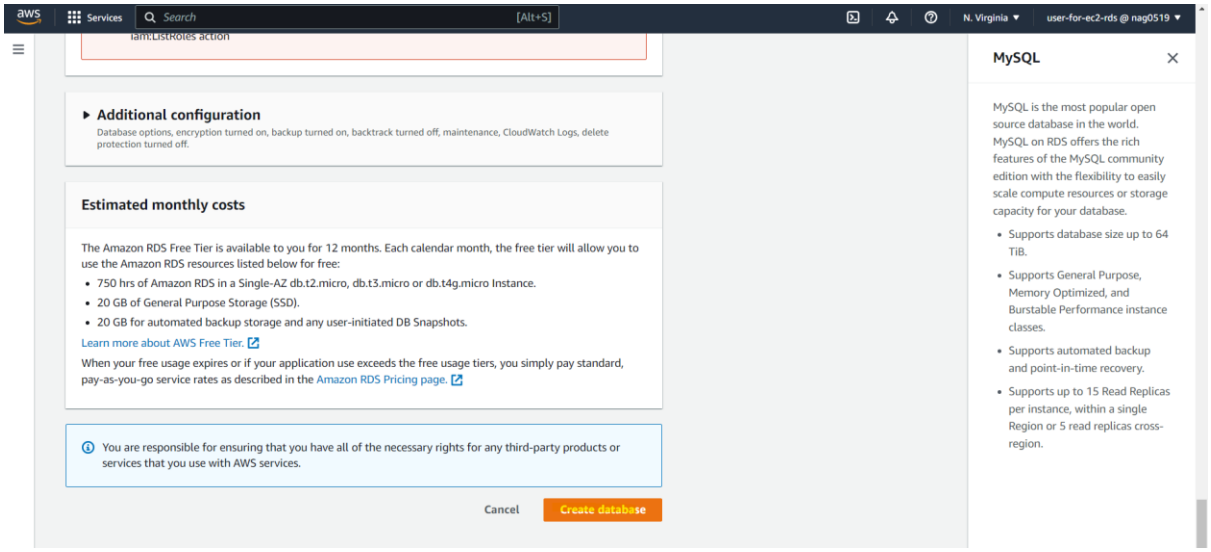








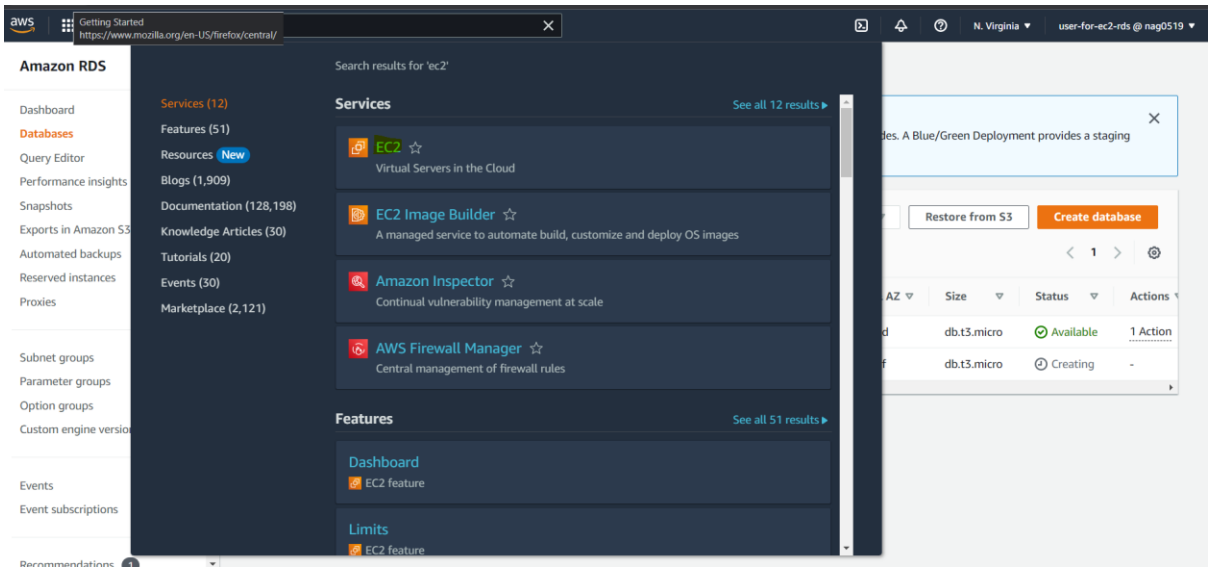
12. Select Create Database



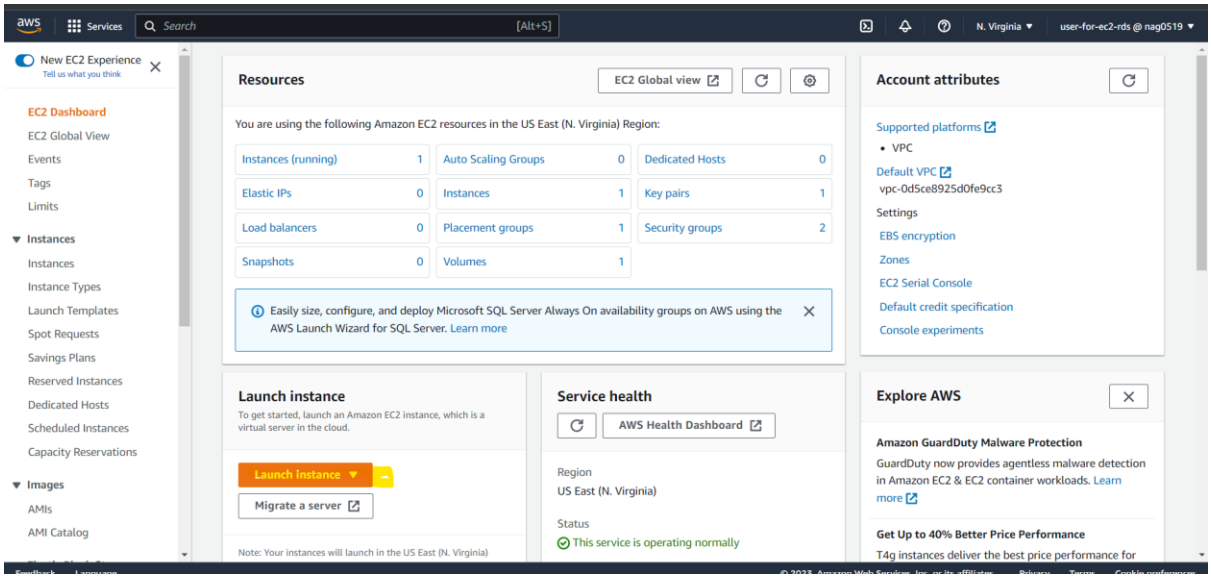
Once we hit Create Database it takes some time to get the database Status as **Available**.

iv. Creating a web Server:

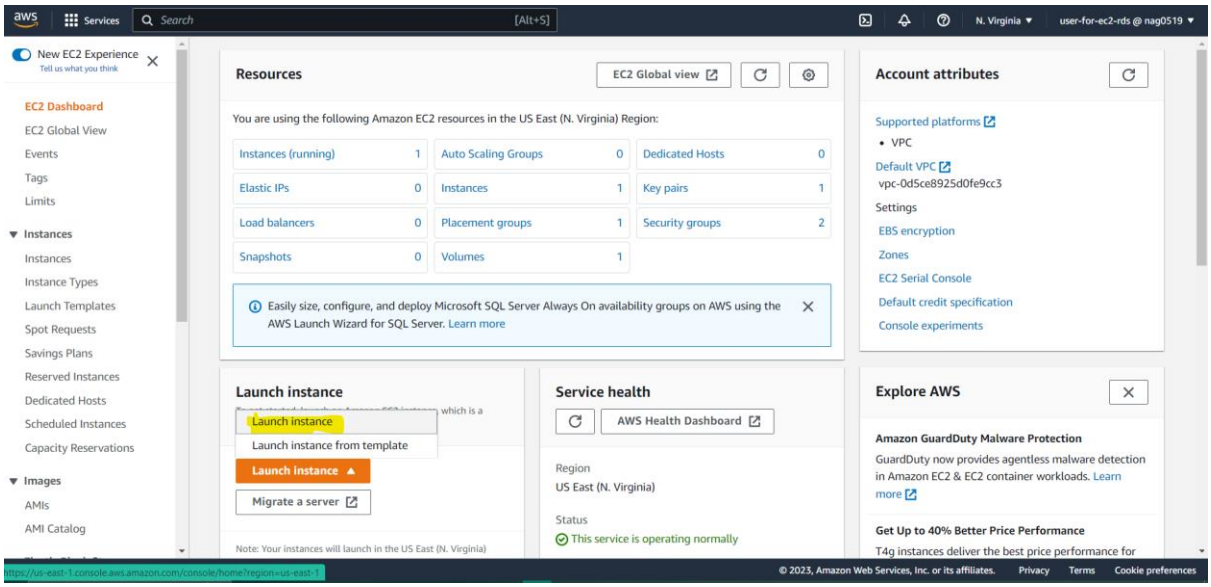
1. Let's search for EC2 in aws search bar:



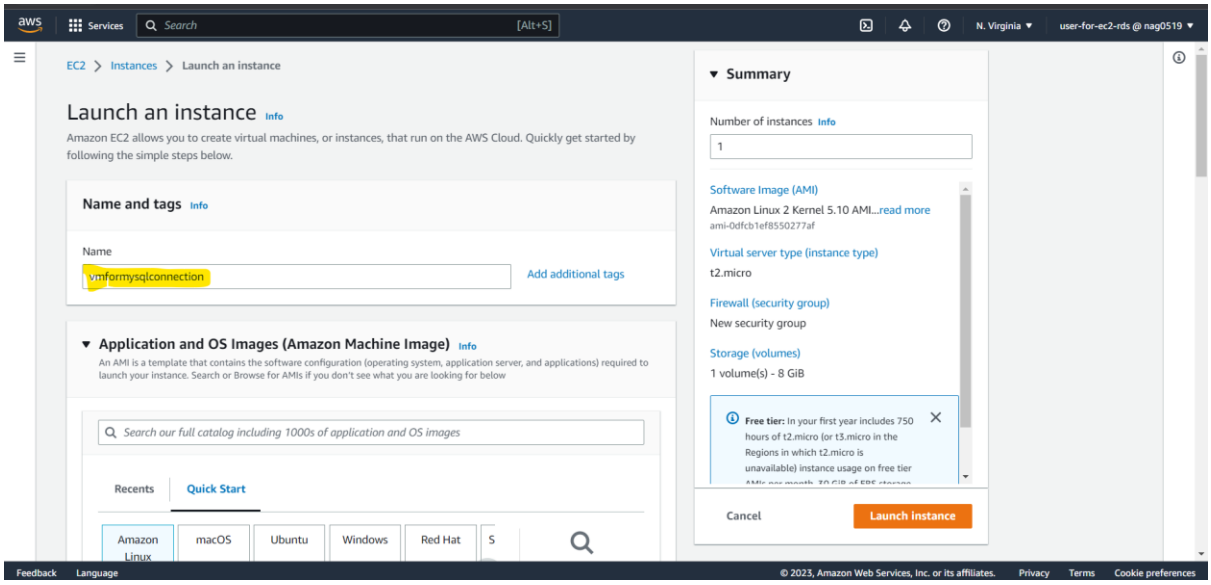
2. Click on Launch Instance.



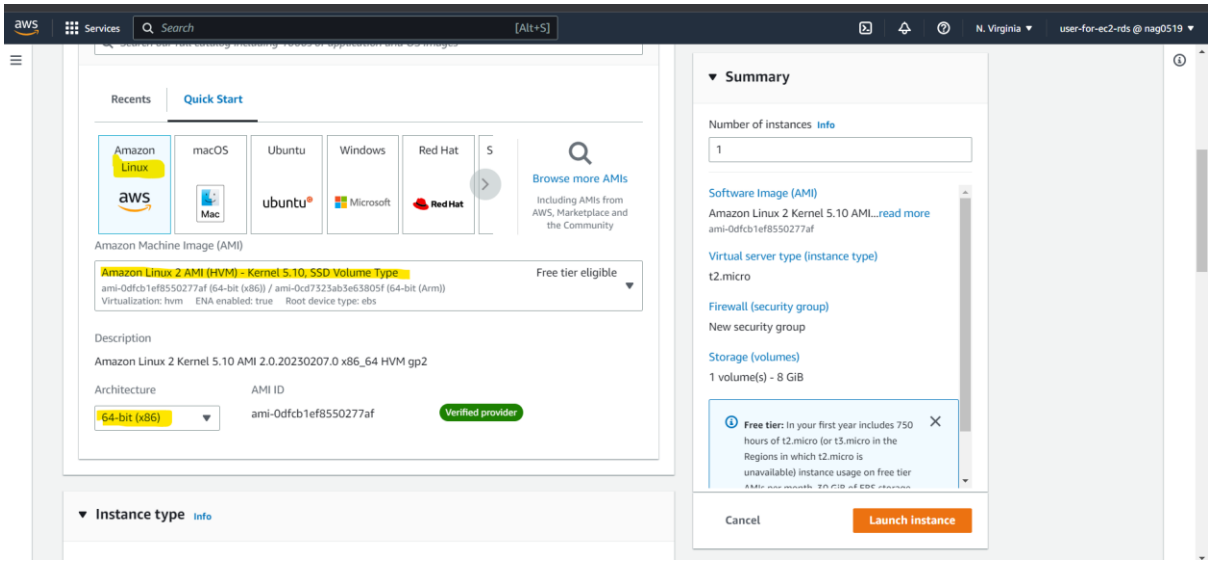
3. Click Launch Instance



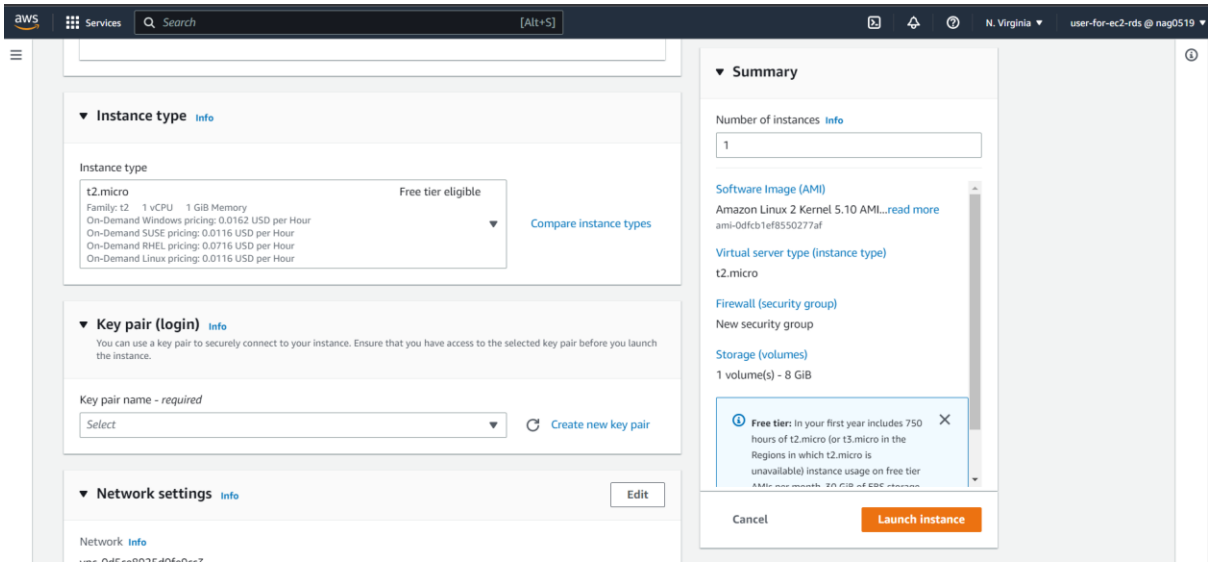
4. Enter the VM name:



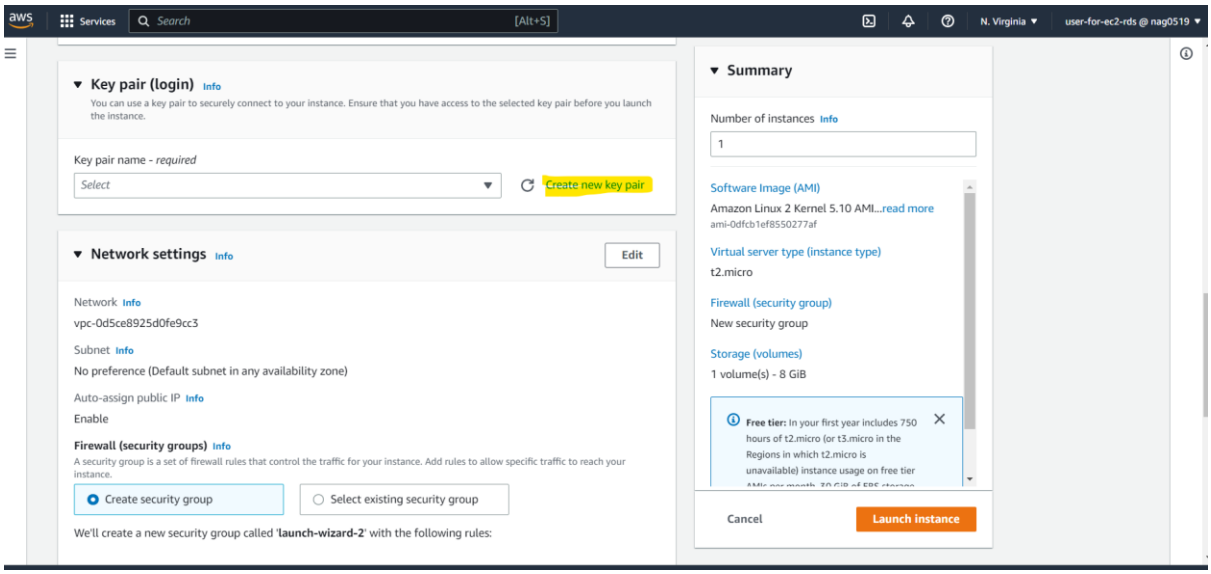
5. Select the VM Configuration as per your requirements, I will choose Linux VM with default settings.



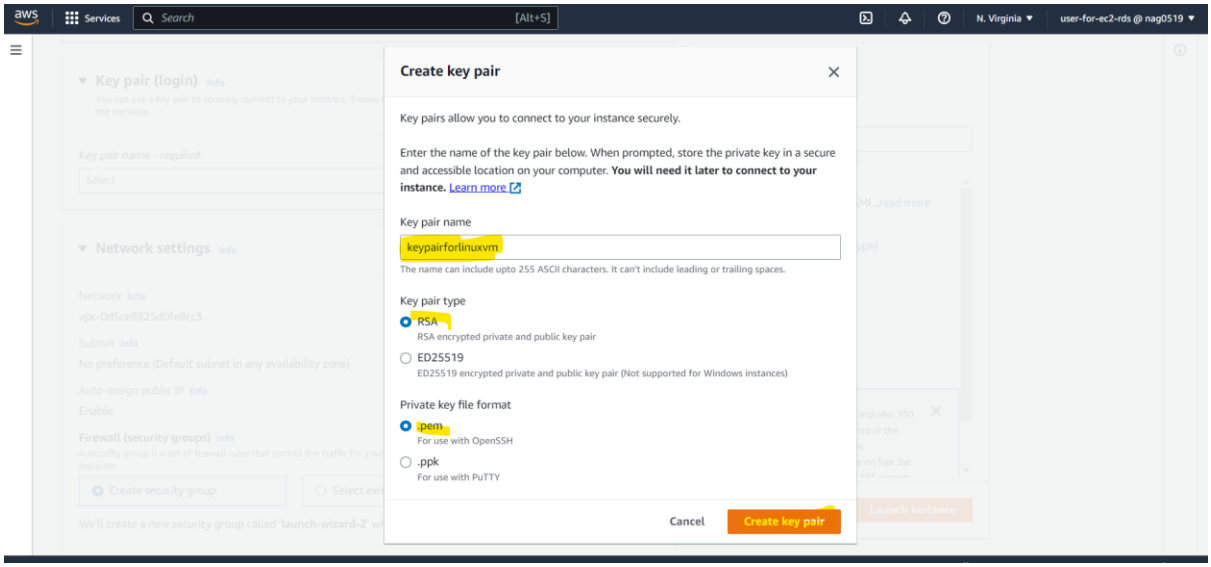
6. Choose the instance type as t2 micro:



7. Create Key-pair for VM:



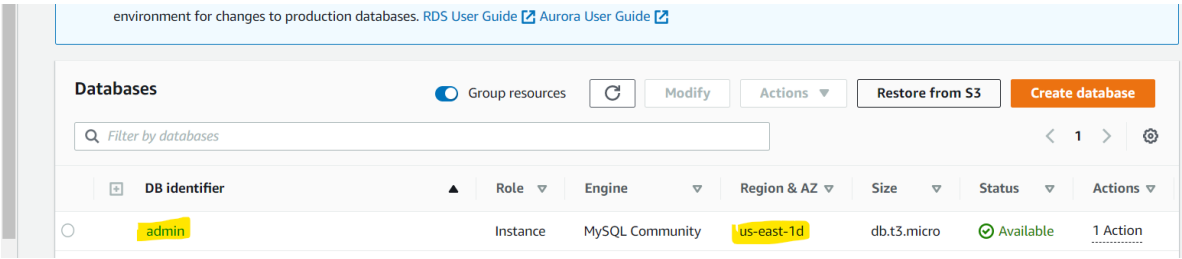
- Type name for your key pair.
- Select .**pem** file as file format

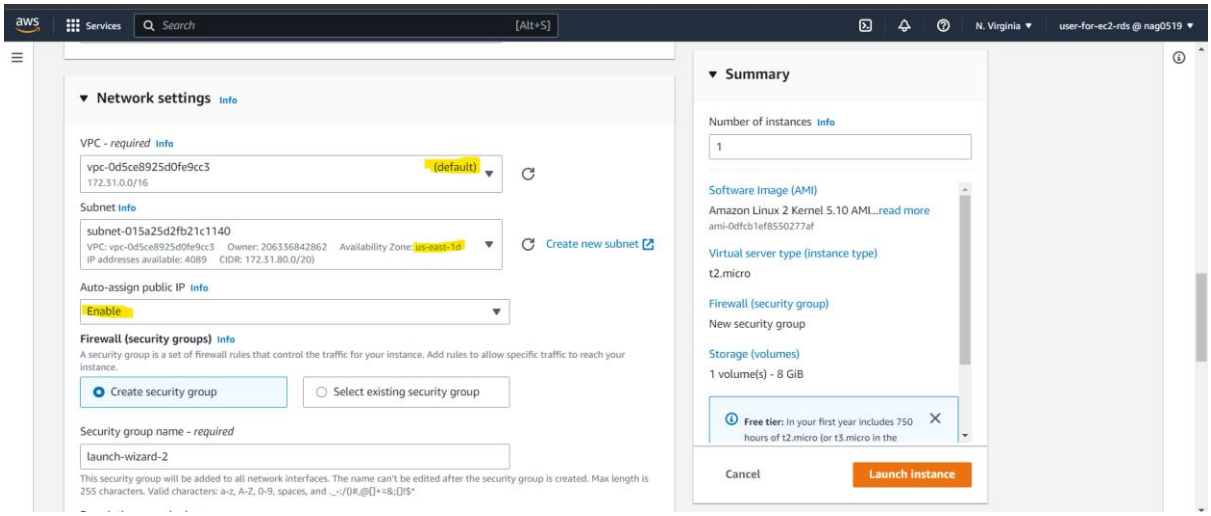


- Then click Create key pair.
- The key pair file gets downloaded in our local system.

8. Under Network setting:

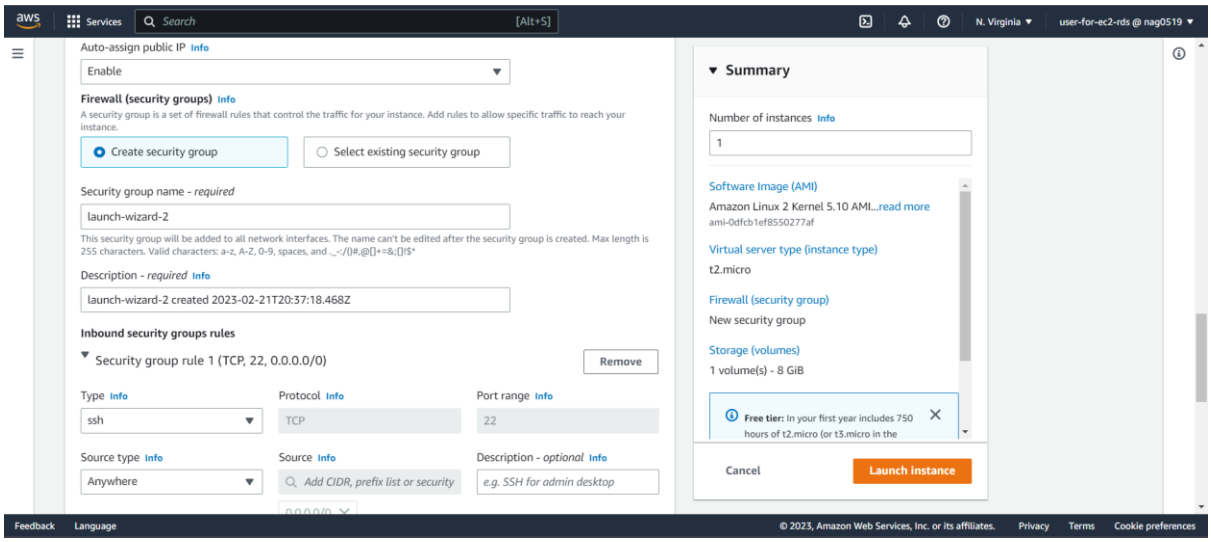
- Click edit.
- Choose Default VPN.
- For Subnet choose the subnet region which is exactly same as the region of the MySQL, RDB we created.



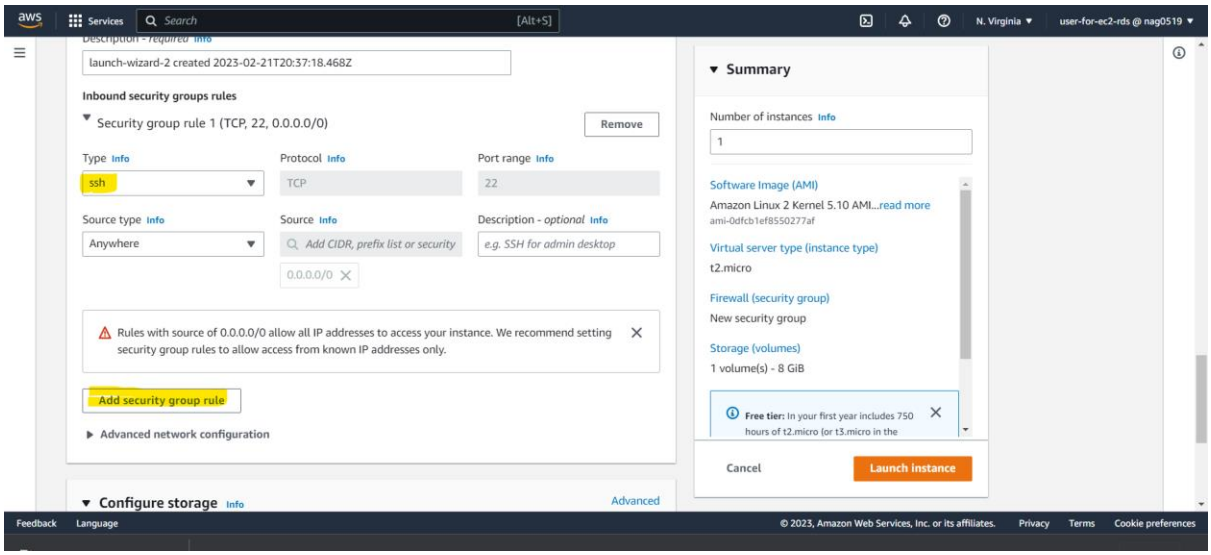


- Auto assign Public IP: Enable

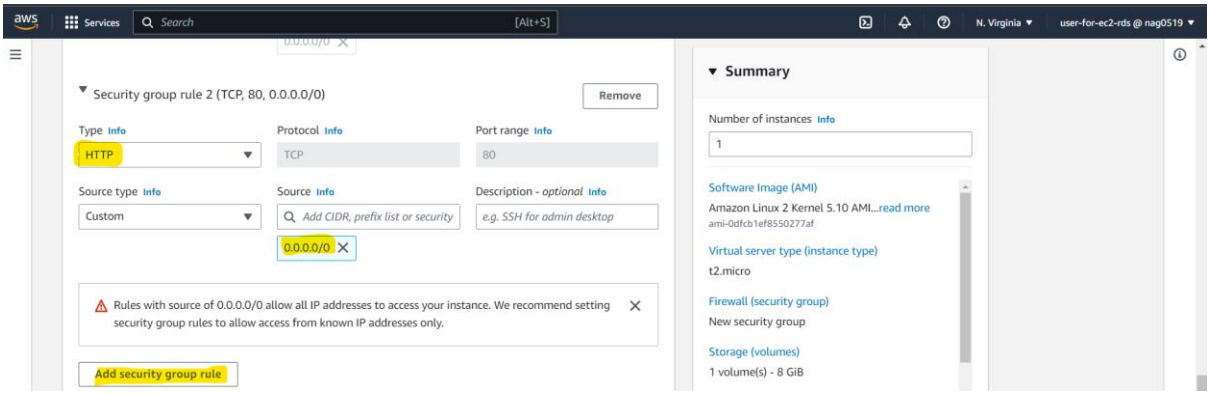
- In Firewall group:
- Give a security group name and note it aside for further use.



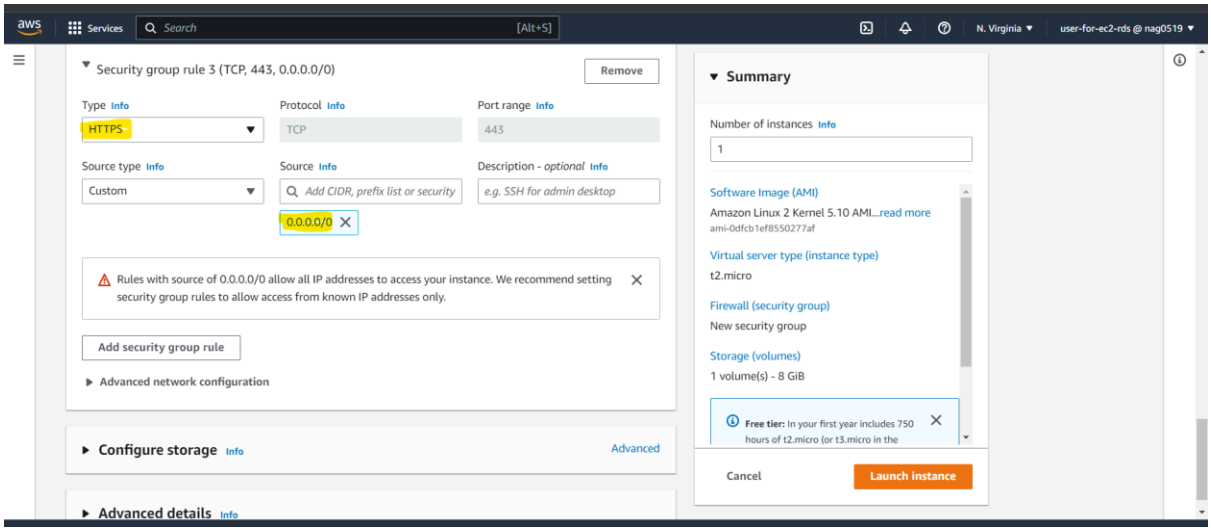
- In Inbound Security Group rules :
- Let SSH setting be present(default)



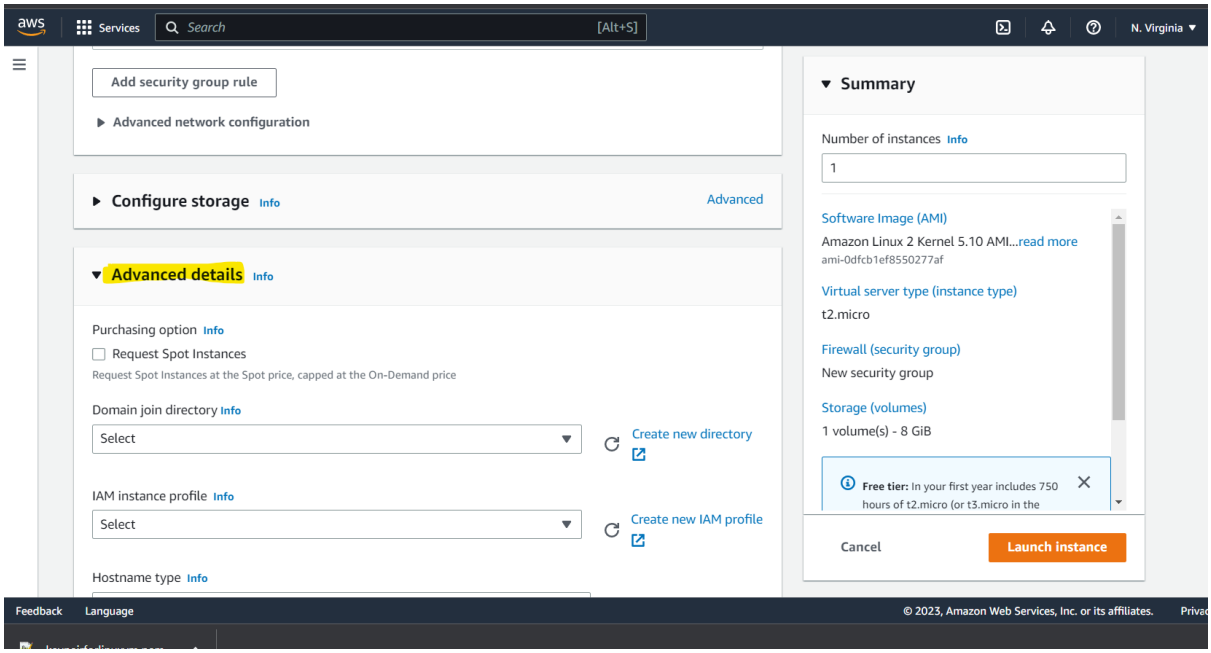
- Click on Add security group rule.
- In Type: HTTP
- Source: Internet i.e., 0.0.0.0/0



- Then add another Security group rule:
- Type: HTTPS
- Source:0.0.0.0/0

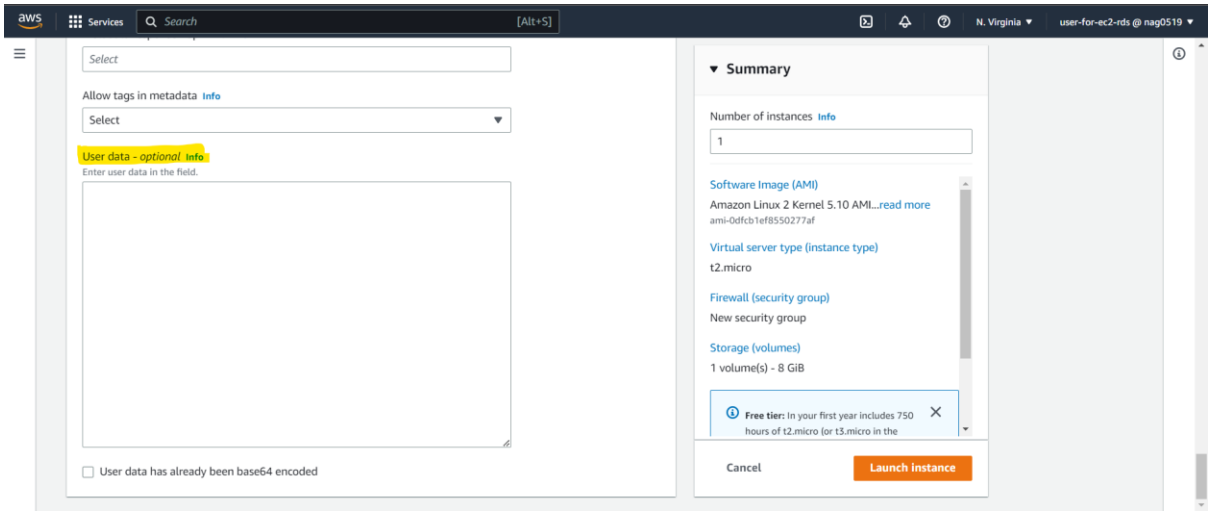


9. Then in Advance Details scroll to the bottom

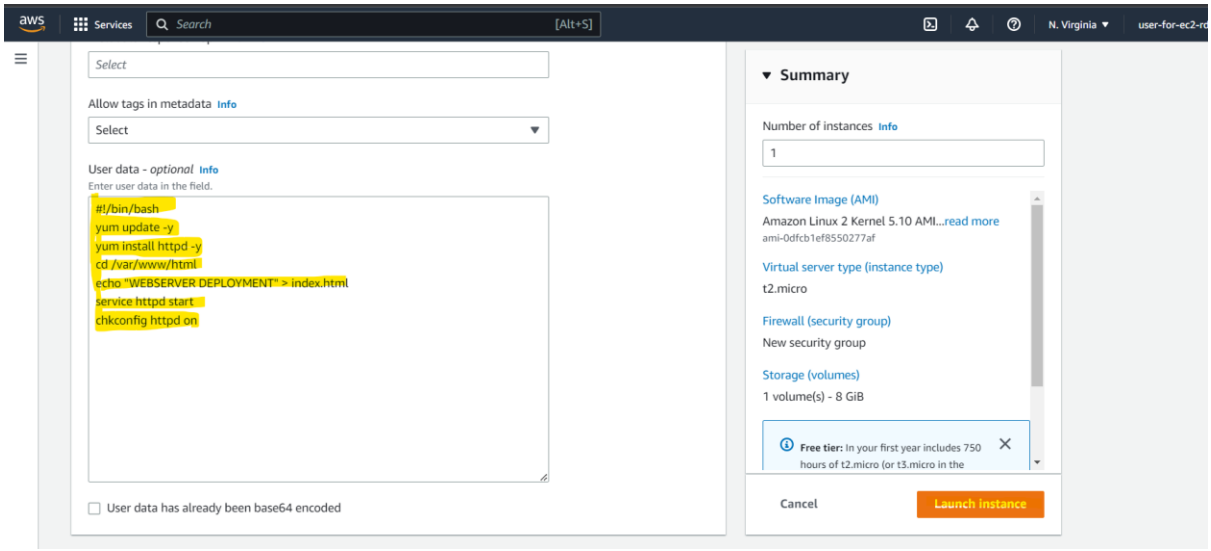


10. In User Data we need to type the following script :

```
#!/bin/bash
yum update -y
yum install httpd -y
cd /var/www/html
echo "WEBSERVER DEPLOYMENT" > index.html
service httpd start
chkconfig httpd on
```



11. Type the script and click on **Launch Instance**.



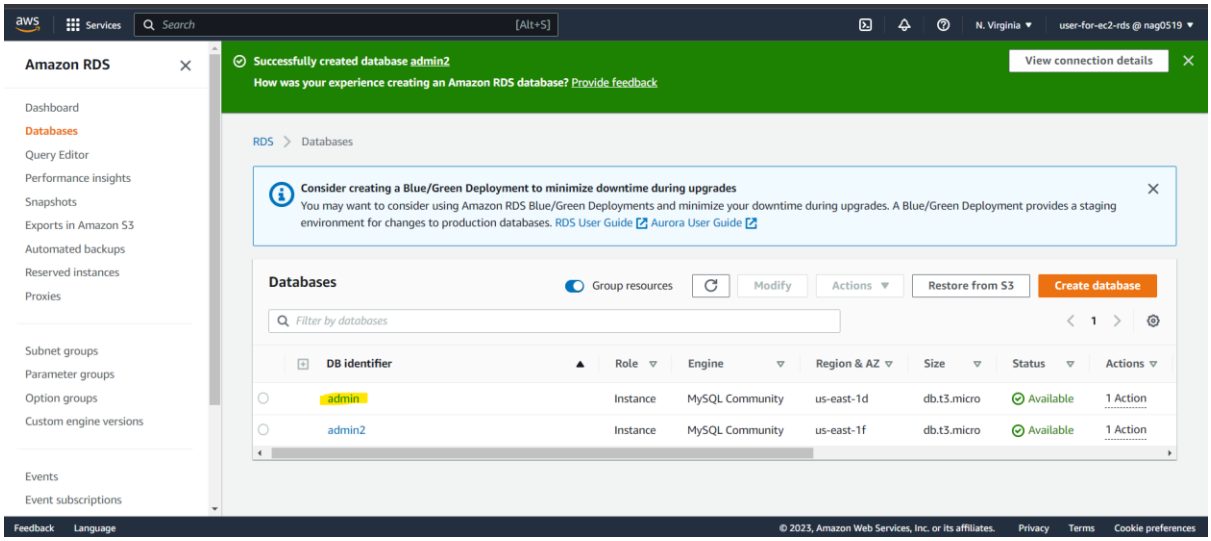
Our VM with Web Server is created successfully.

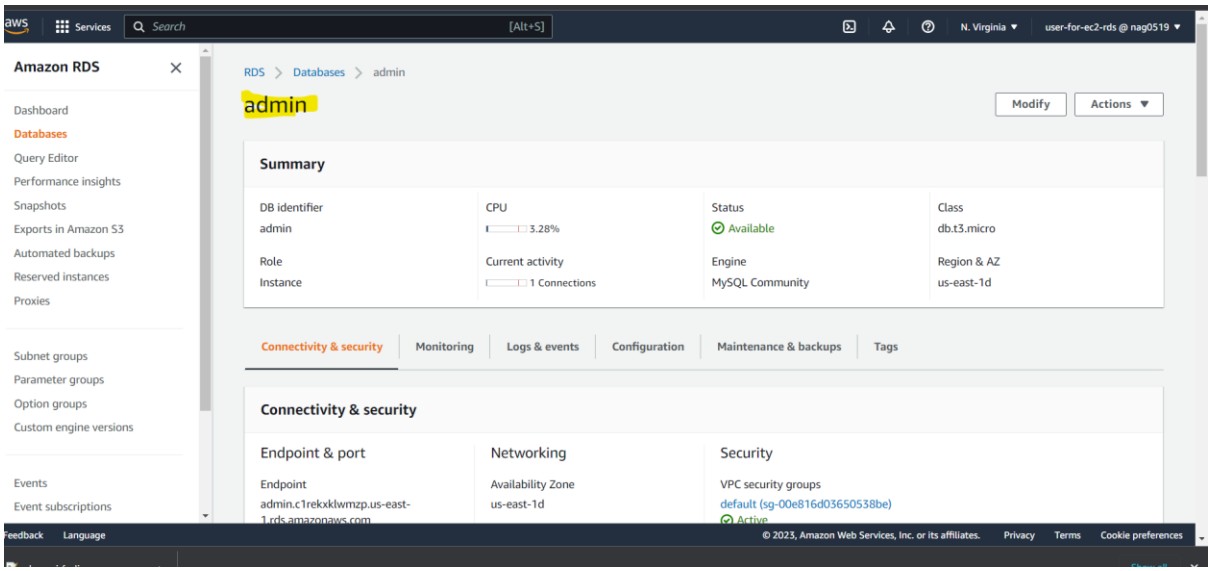
- Now we go to Instance dashboard to check the status of our VM.
- We will check our VM by Connecting it to the terminal.

Now we have created the MySQL DB and our Linux VM with Web Server.

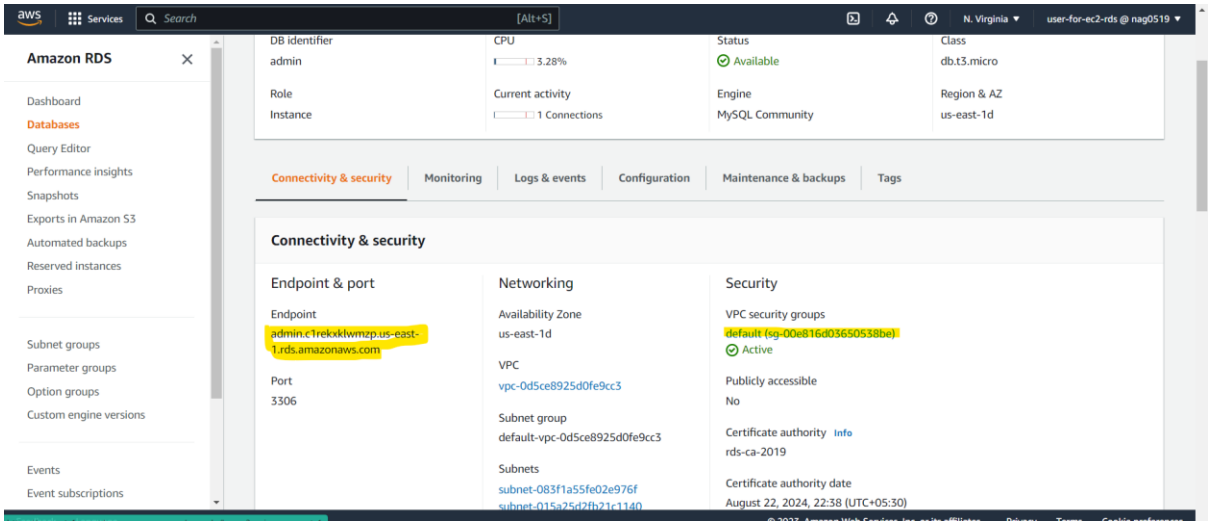
v. Creating a connection between the database and the VM.

1. Navigate to the RDS dashboard and select our admin DB.

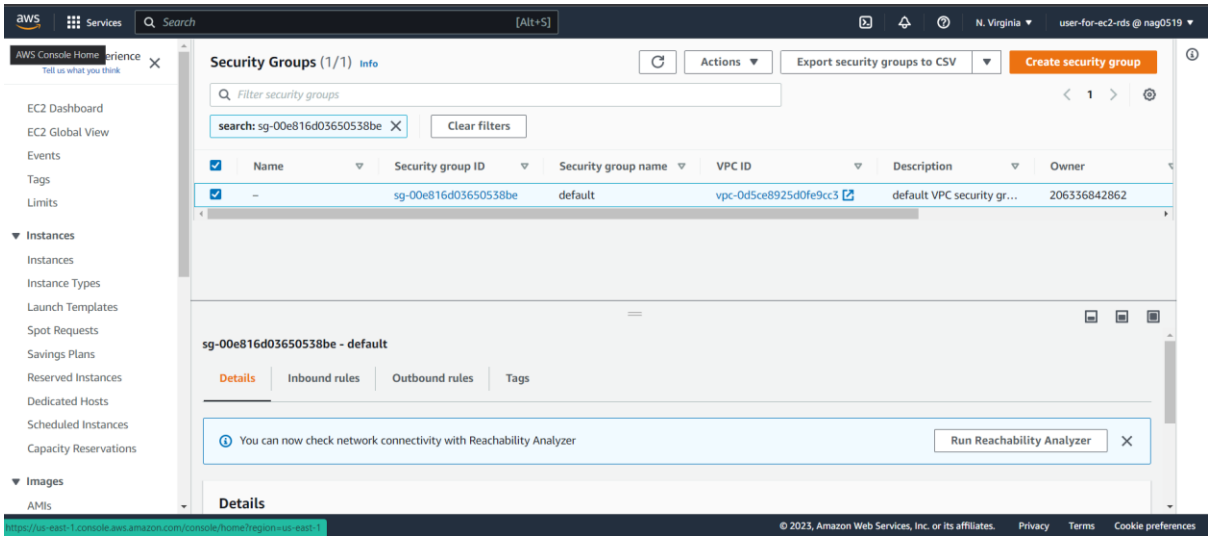




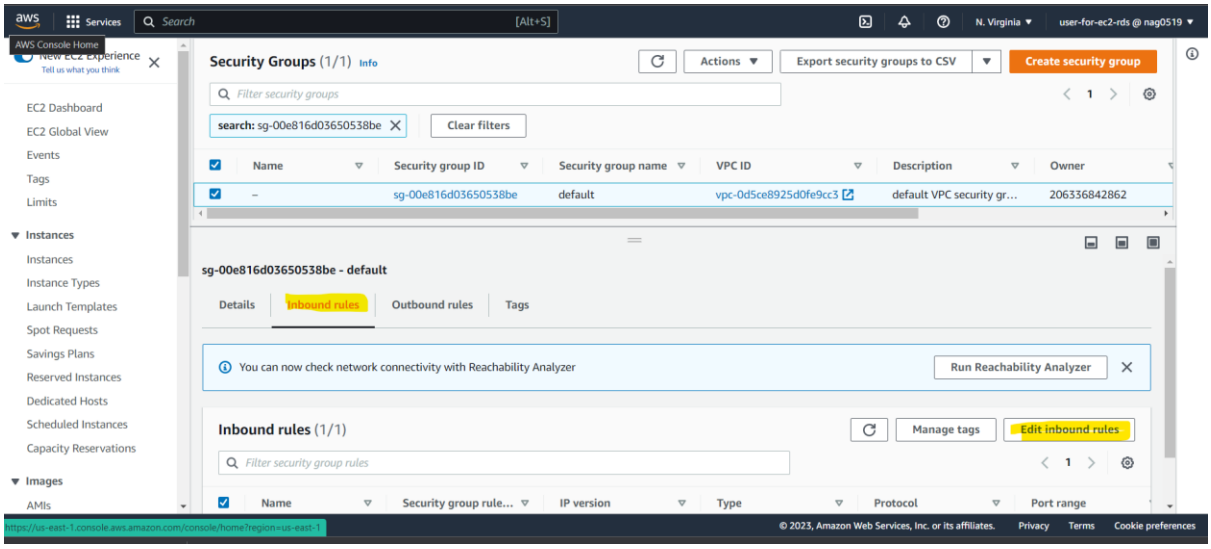
- Here we need to save our endpoint for further usage.
- We need to scroll down to get the endpoint.



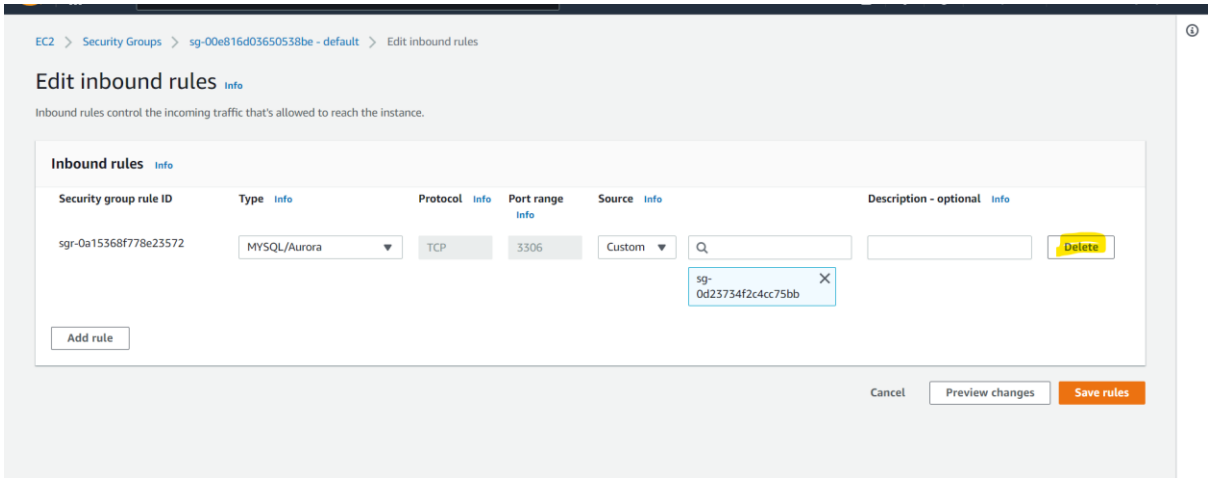
2. Once noted the endpoint we need to click the VPC Security Groups:
3. Security Groups dashboard opens:



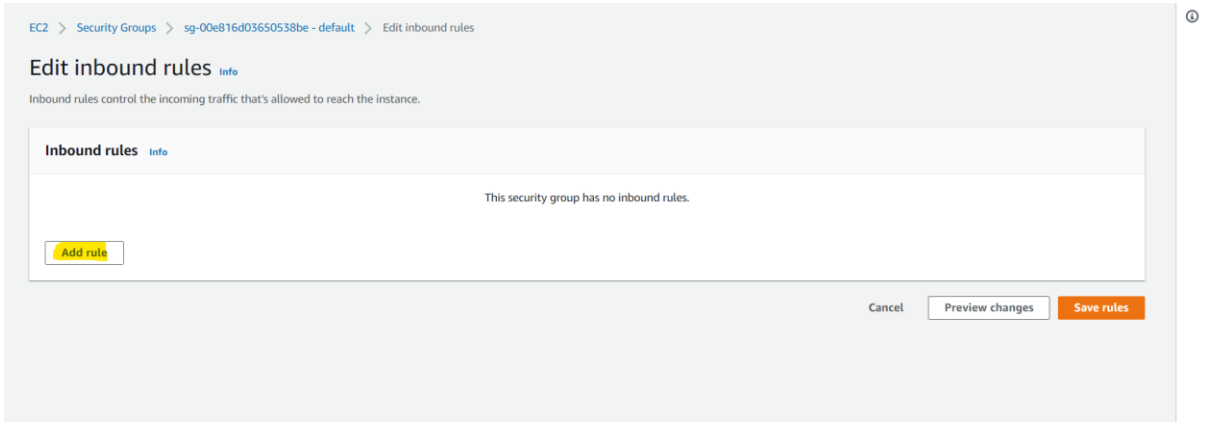
4. We need to scroll down and **edit Inbound Rules**.



Edit Inbound Rules page opens

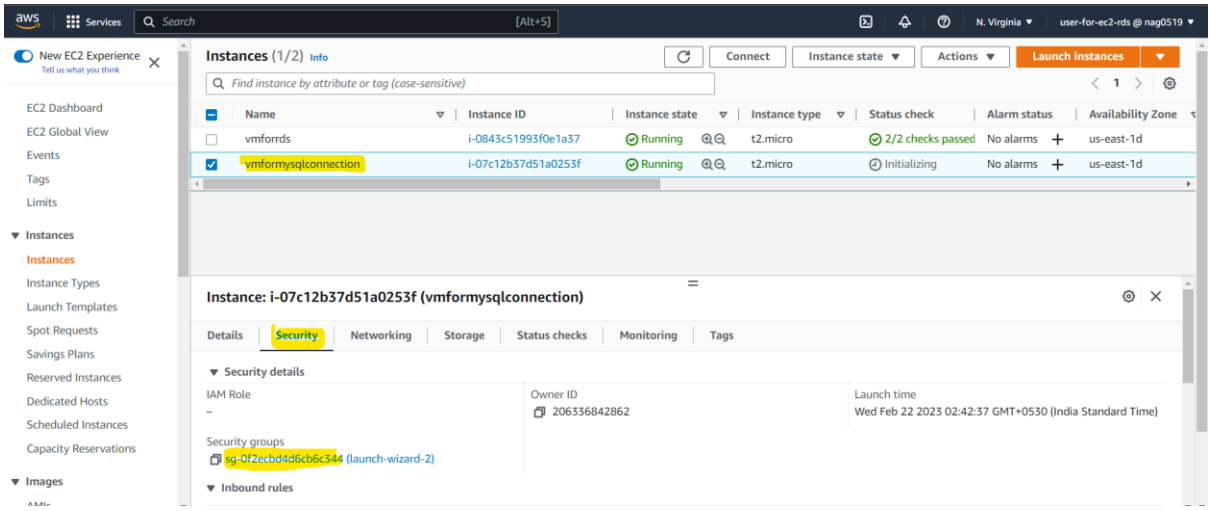


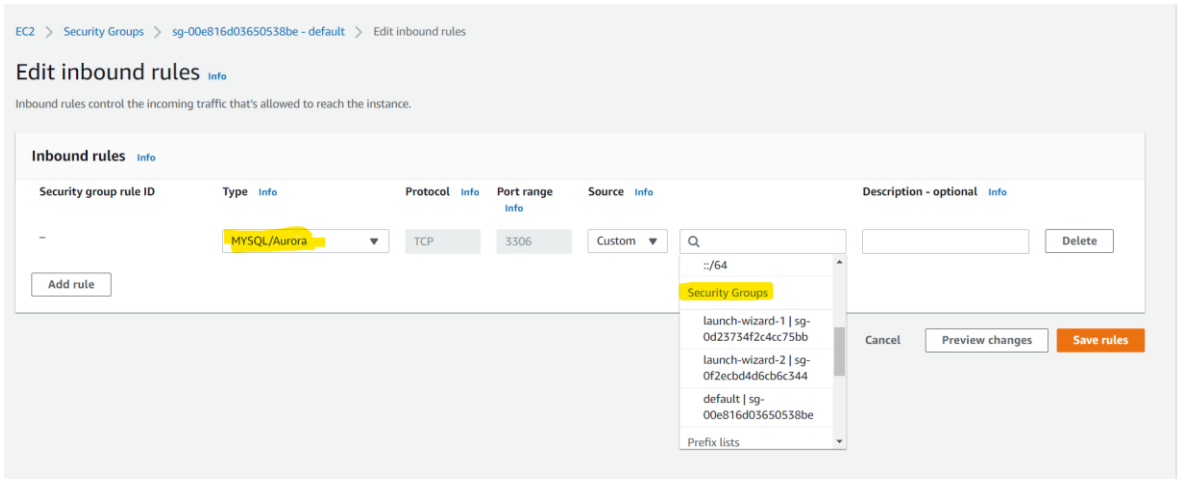
5. We need to delete all the inbound rules .
6. Now we need to click on Add Rule



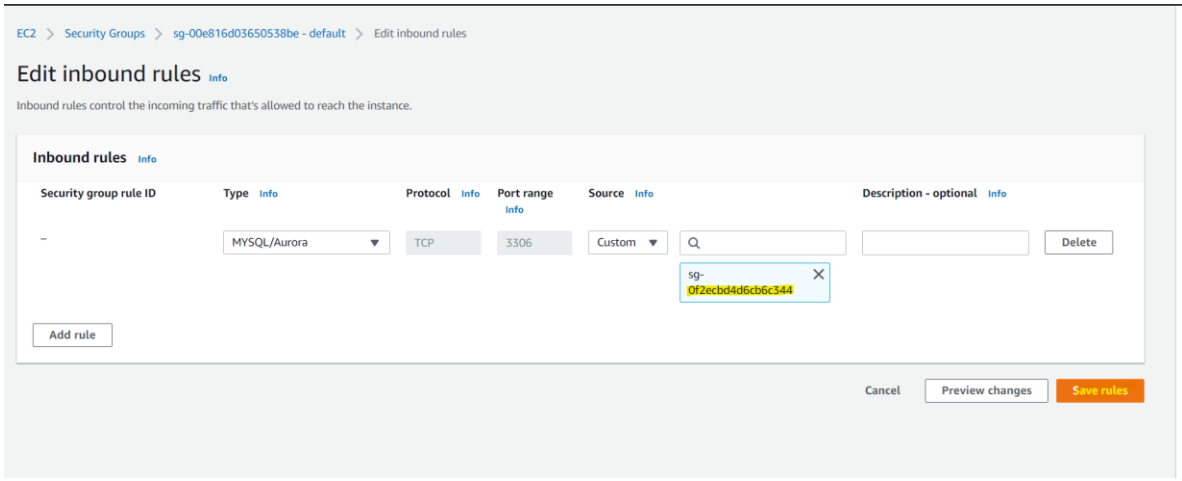
- In Type: Select MySQL/Aurora
- In Source: From the security groups present select the security group of the VM we created.

VM's Security Group:

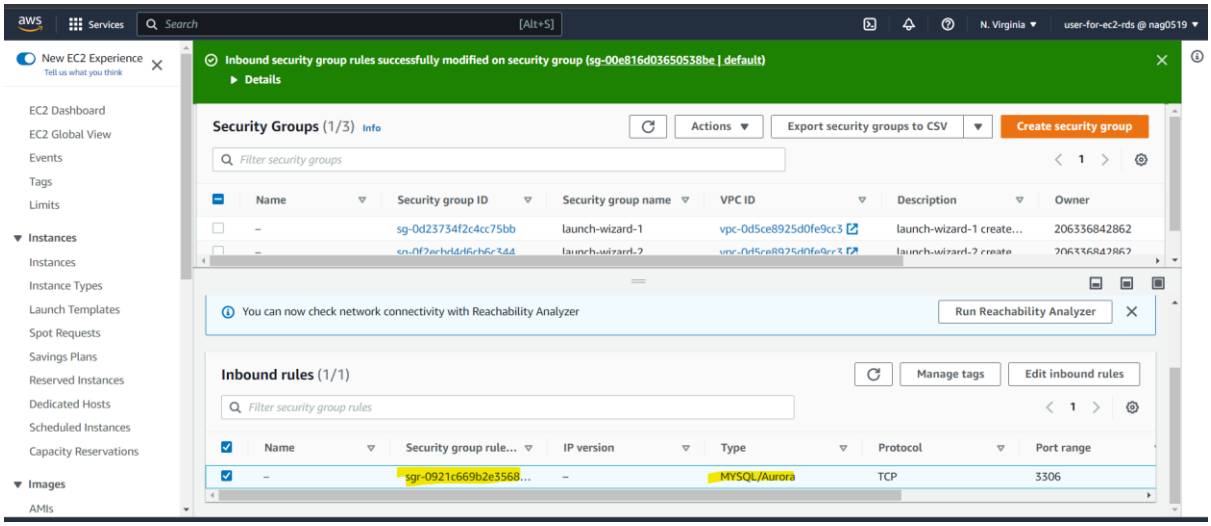




7. Now click on Save Rules



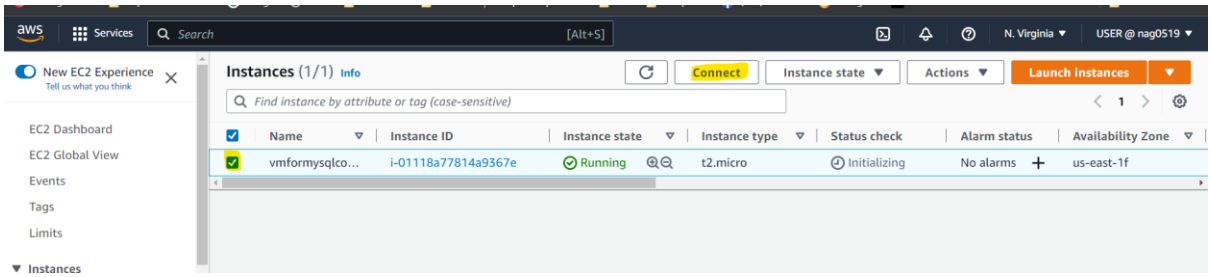
- Our Inbound Rule is displayed upon saving.



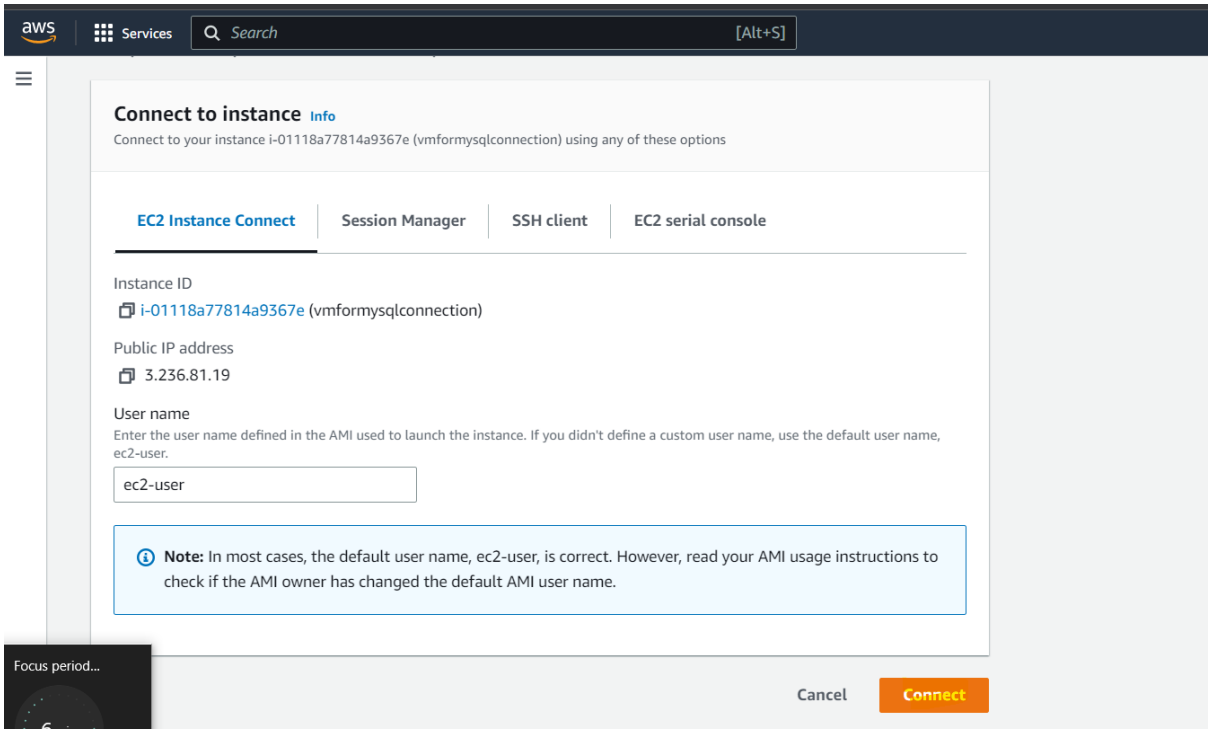
- Now we are ready to connect our VM with MYSQL and run the commands.
- Let's navigate to our VM.

vi. Launching our VM terminal:

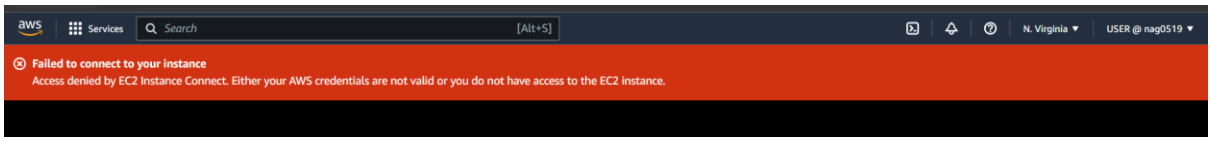
1. Select our VM and click on Connect.



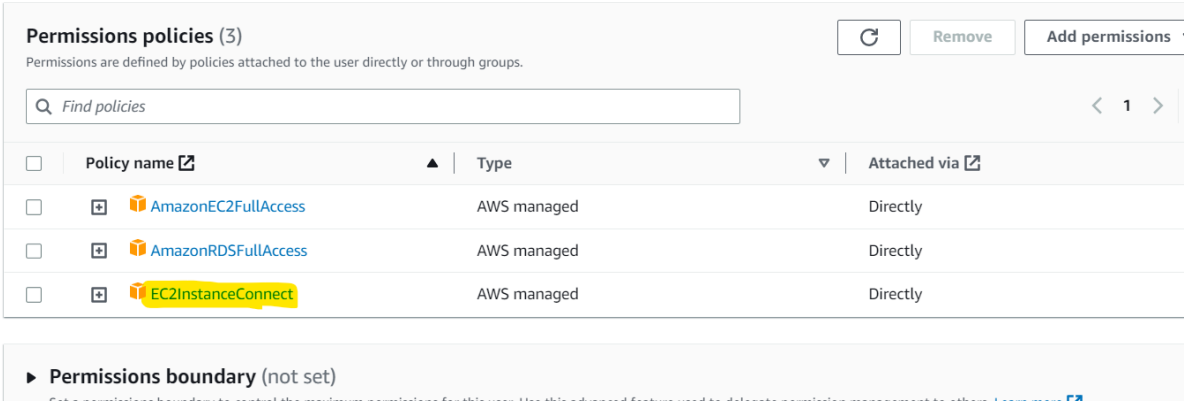
2. Again, click on connect.



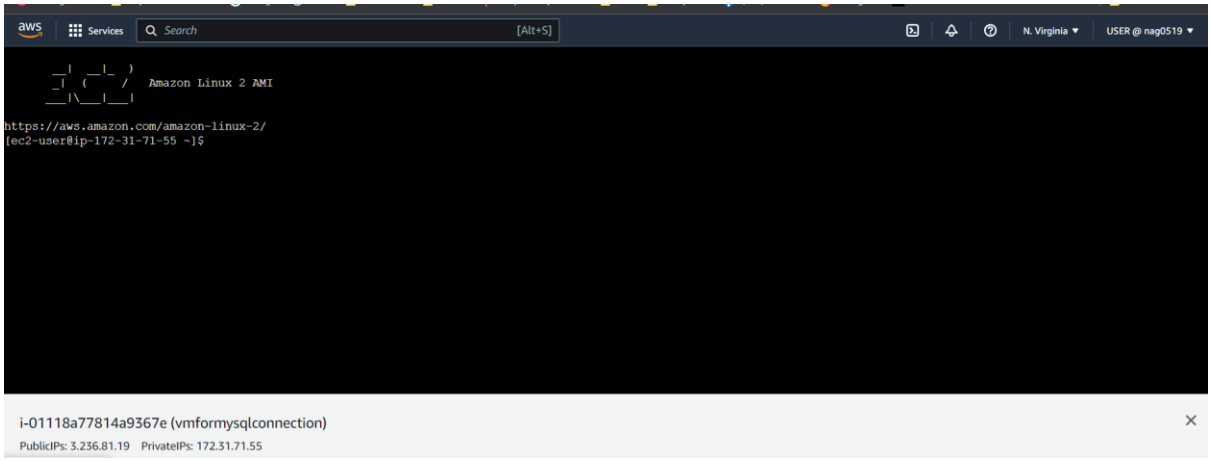
- After hitting connect we were getting the error:



3. So, then we gave user one more permission:



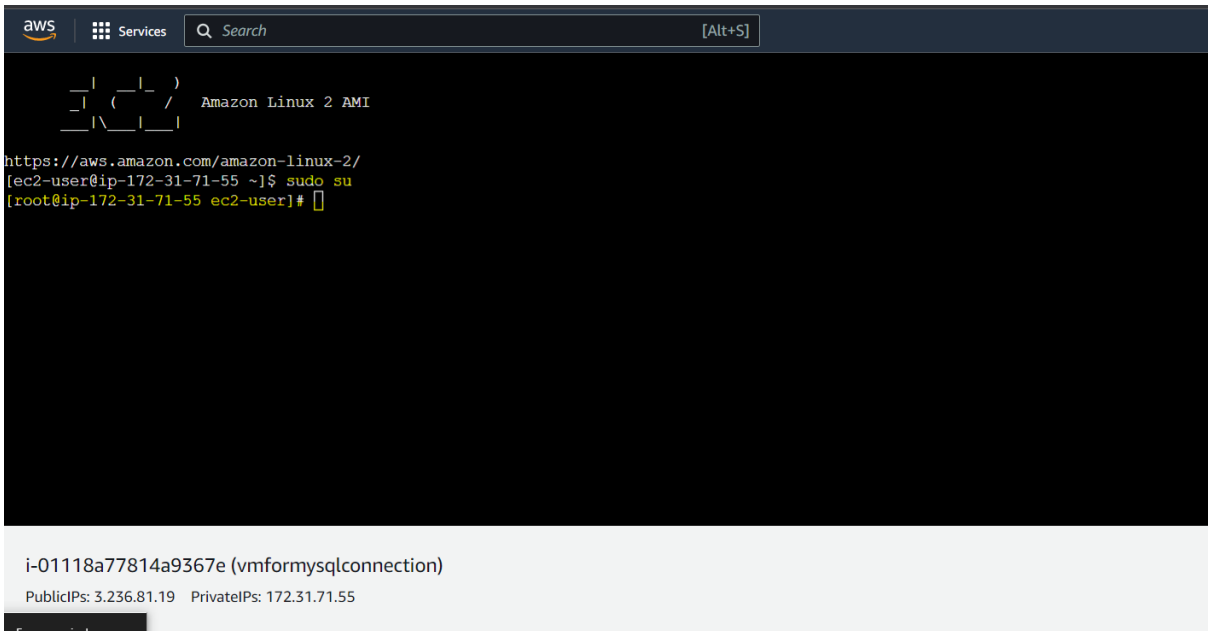
- Now we are able to connect to the VM



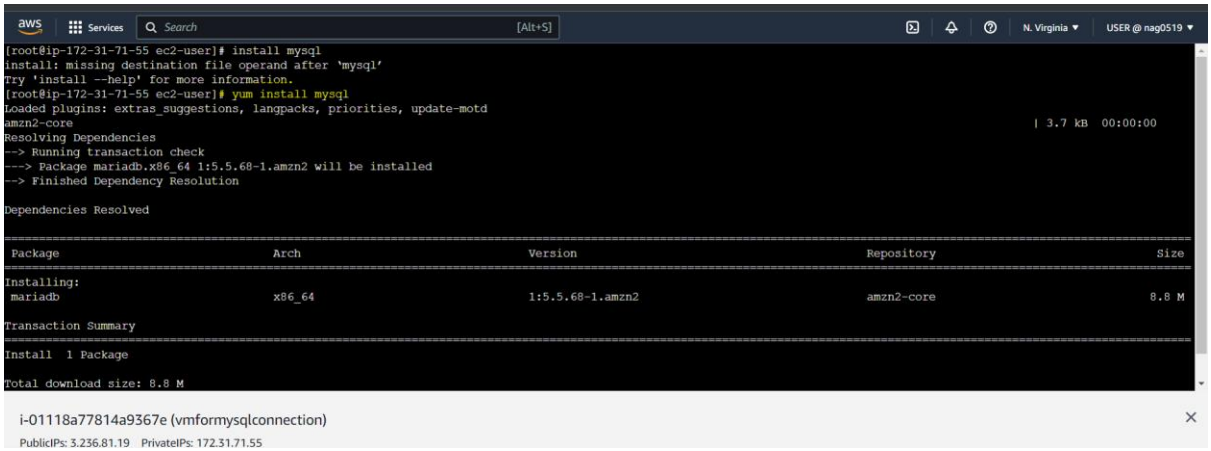
Now lets make MySQL connection .

Now we will enter the following command:

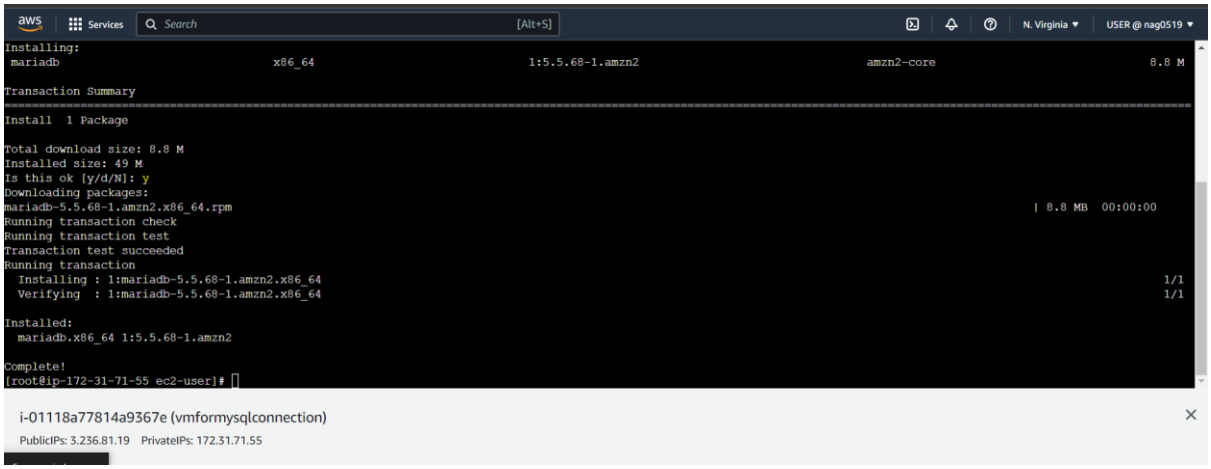
1.sudo su: This command is used to become the Super user



2.yum install mysql: to install mysql in Linux terminal.



Type y:

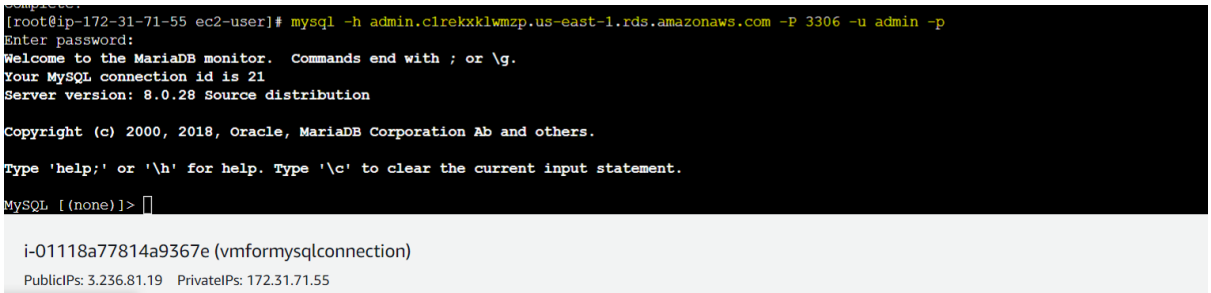


3. mysql -h endpoint_of_rds -P 3306 -u rds_username -p

This command is used to connect database with the VM.

In our case the command can be changed as:

mysql -h admin.c1rekxklwmzp.us-east-1.rds.amazonaws.com -P 3306 -u admin -p



- After we type the above command, we are prompted to enter our password.
- Now we type our RDS password but we don't see our pointer moving as we type our password.
- After typing the password, we have successfully connected our VM with the MySQL database.

- Now we can create a database inside the mysql db.

Show databases: this command is used to see all databases created.

```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

MySQL [(none)]>
```

We will create our own database.

Create database users.

```
MySQL [(none)]> create database users;
Query OK, 1 row affected (0.02 sec)

MySQL [(none)]>
```

Now we will use our database.
Use Users

```
MySQL [(none)]> use users;
Database changed
MySQL [users]> show tables;
Empty set (0.00 sec)

MySQL [users]>
```

Show tables: To display all tables inside the database.

Now we can create table:

```
MySQL [users]> create table employees( name varchar (800), empID varchar(10), city varchar(500) );
Query OK, 0 rows affected (0.02 sec)

MySQL [users]>
```

```
MySQL [users]> show tables;
+-----------------+
| Tables_in_users |
+-----------------+
| employees       |
+-----------------+
1 row in set (0.01 sec)
```

```
MySQL [users]> insert into employees values ('ANAMIKA','201','Banglore');
Query OK, 1 row affected (0.01 sec)

MySQL [users]> select * from employees
    -> ;
+---------+-------+----------+
| name    | empID | city     |
+---------+-------+----------+
| ANAMIKA | 201   | Banglore |
+---------+-------+----------+
1 row in set (0.00 sec)

MySQL [users]>
```

This is how we can configure and connect MYSQL instance to a web server in AWS