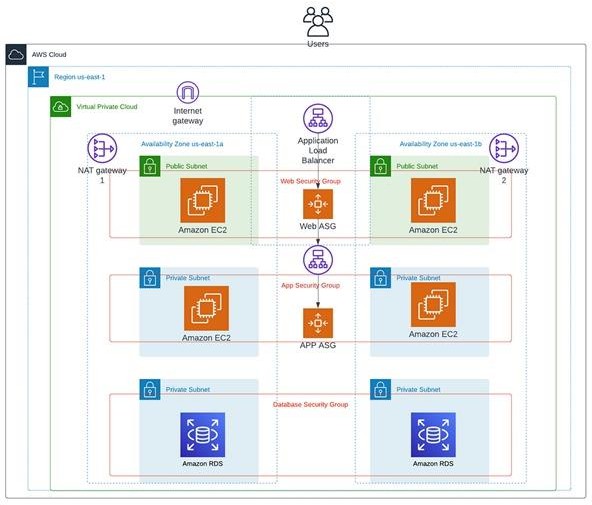
**NAME: Muthyam Rohitvarma**

**MAIL.ID: rohitvarmamuthyam123@gmail.com**

****

**The above architecture is the “Architecting 3 Tier Architecture on AWS”.**

* First tier of our architecture is a web tier. It consists of 2 public subnets in separate availability zones, and an Auto scaling group with launch

template and security group.

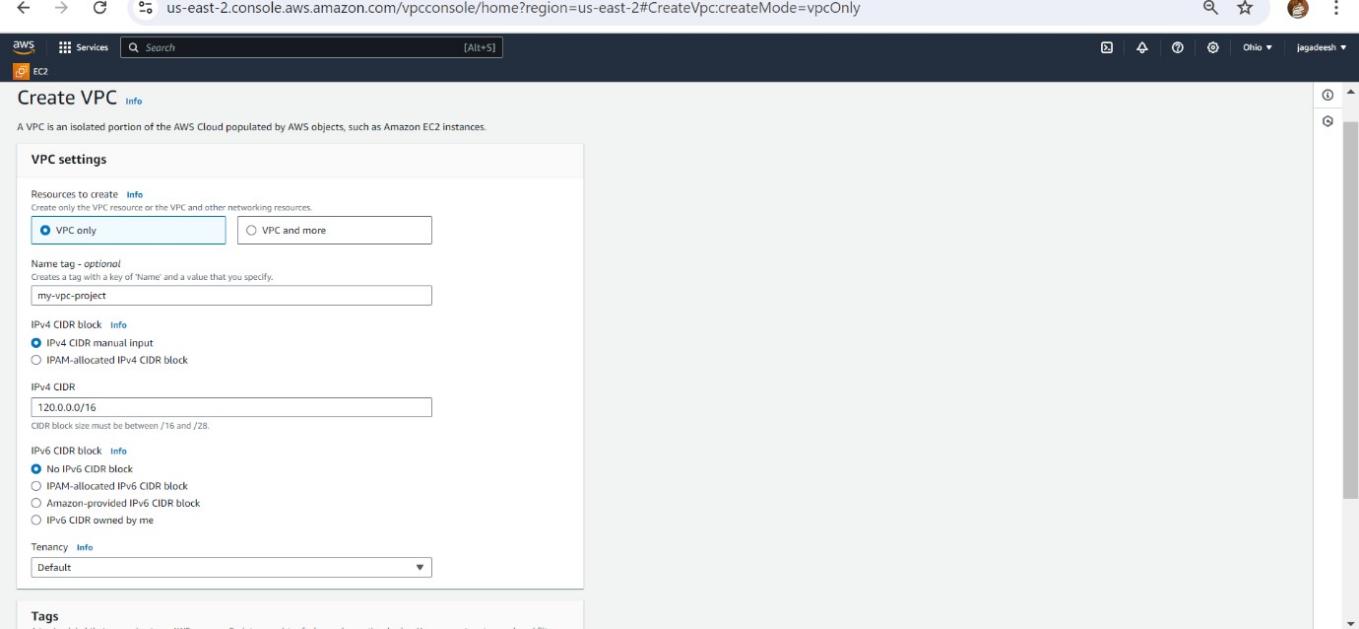
* The second tier is an Application tier. This tier will consist of 2 Private, an Autoscaling group with launch a template and same security group used in above web tier.
* The third tier is a Database tier. This tier will have an RDS(relational database service) in 2 Private subnets and an same security group used in above both the tiers.

# Creating the above architecture we have to follow the following steps:

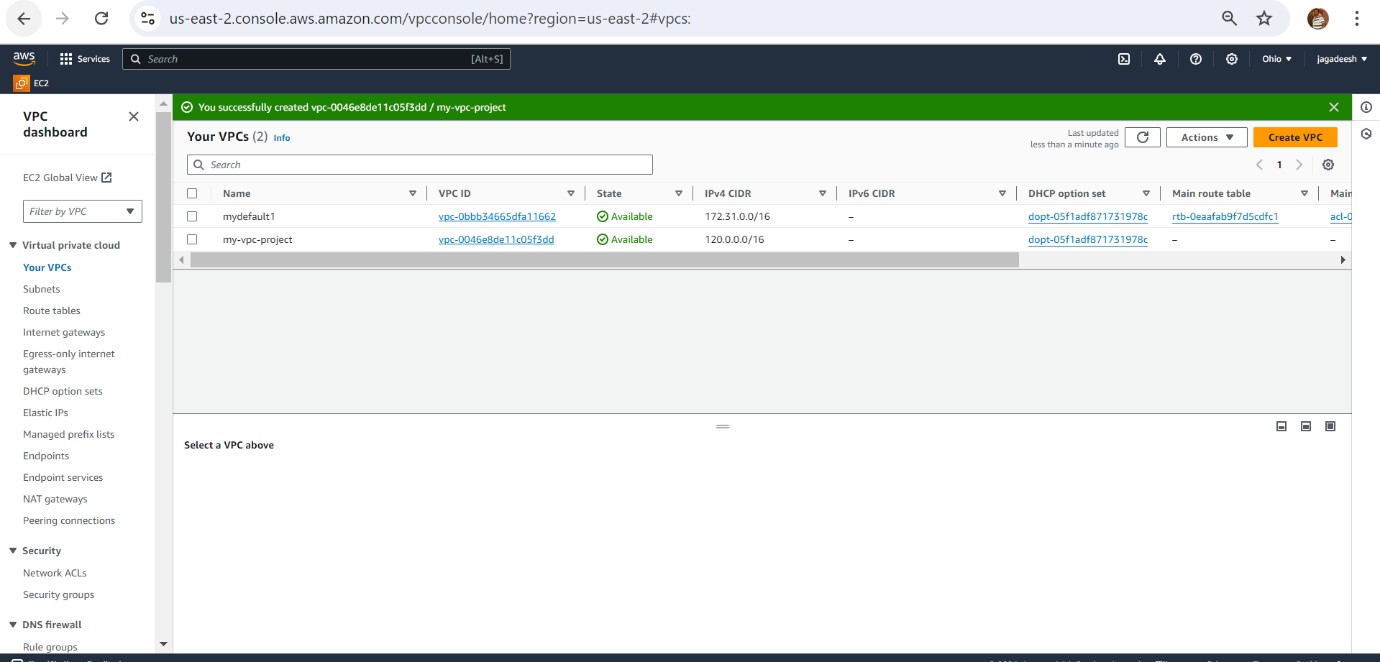
1. Create VPC, Subnets – 6, Internet gate way – 1, Route tables – 2, Nat gate way – 1.
2. Launch an EC2 instance.
3. Create Load Balancer
4. Create an AMI (image).
5. Create Autoscaling group, Create launch template.
6. Create Subnet group.
7. Create Database (RDS).
8. Establish connection.

# Step: 1

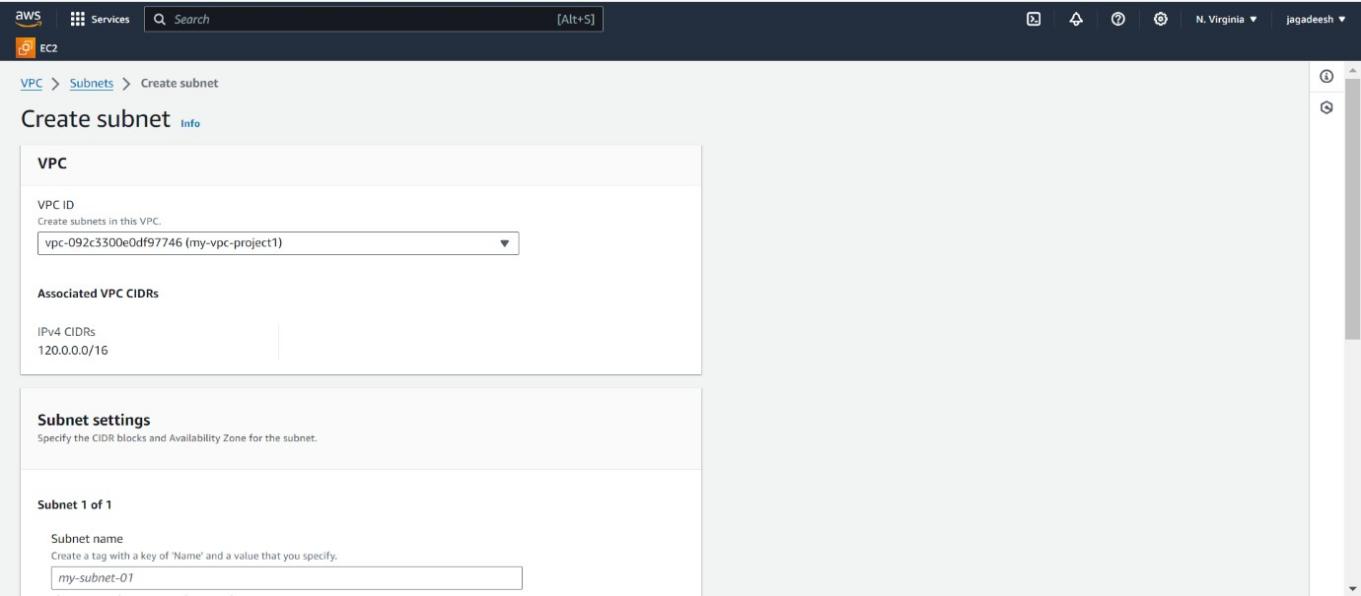
Create VPC and its components :

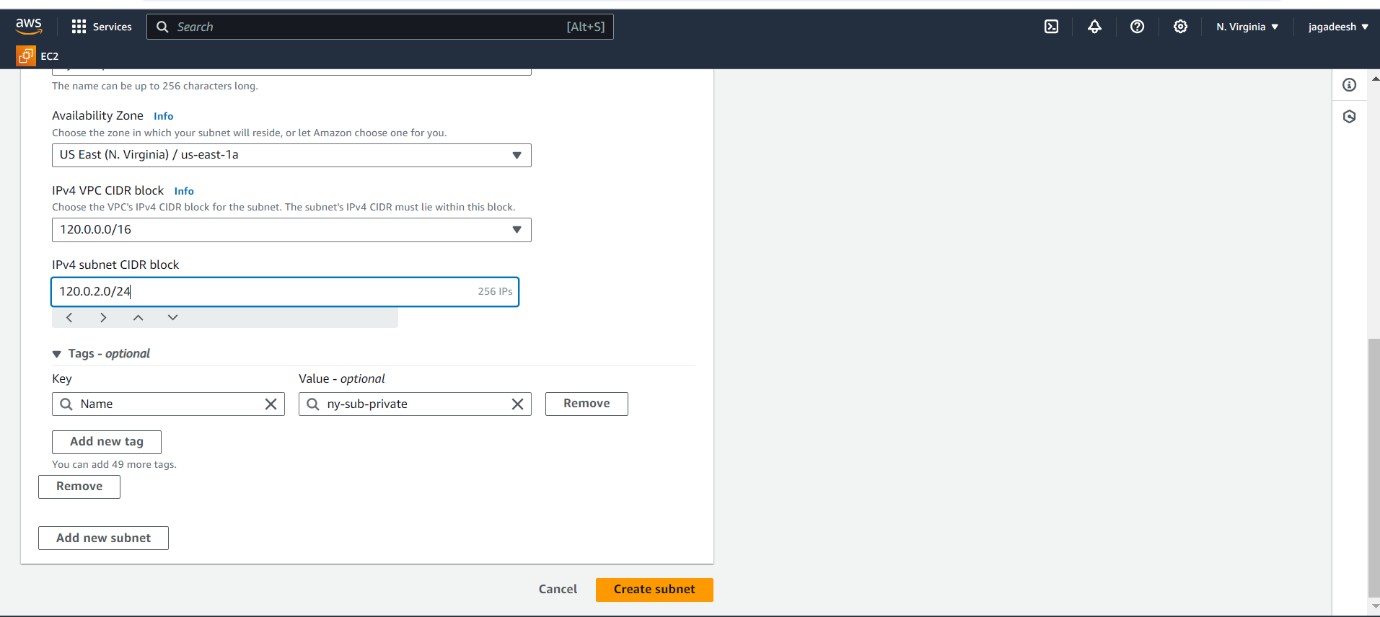


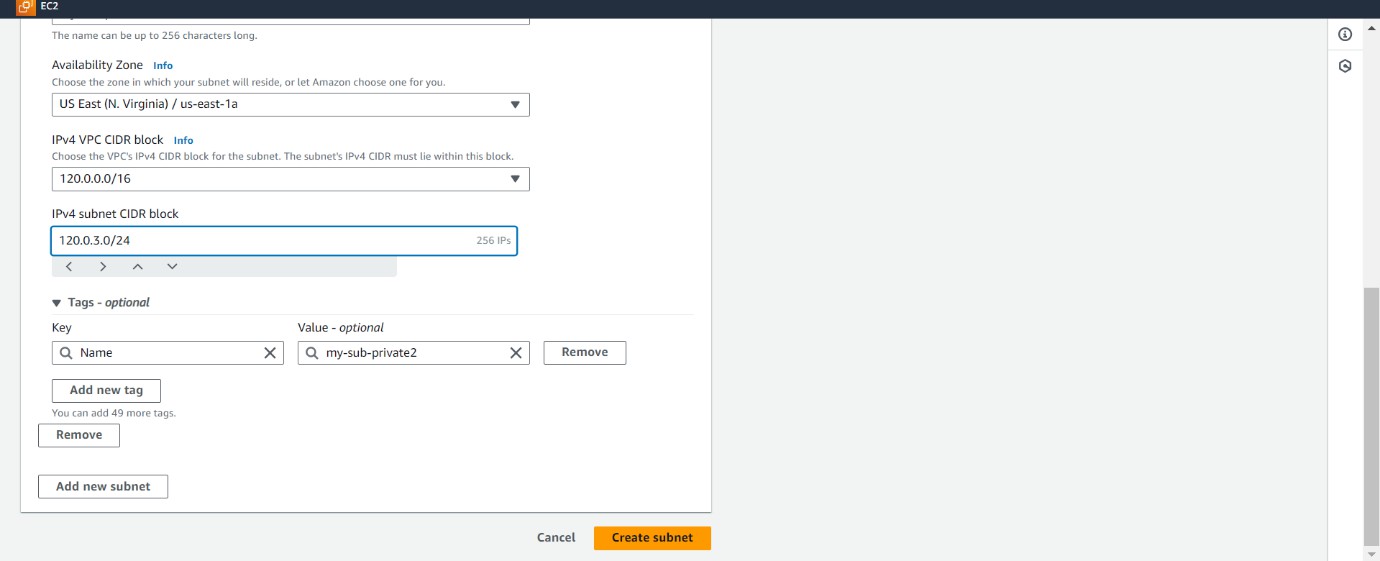
* + Go to VPC dashboard click on create VPC.
  + Click on VPC only and name tag as my-vpc-project1
  + Give IPV4 CIDR (classless inter domain routing) as 120.0.0.0/16
  + Click on VPC, it is created



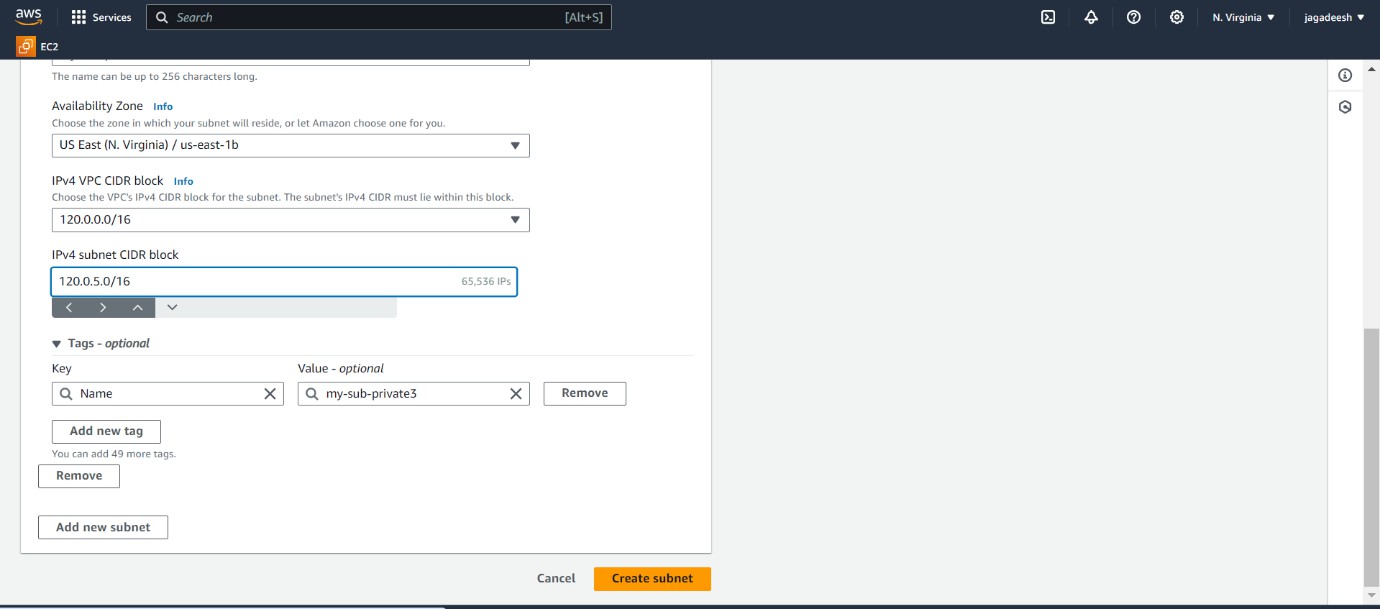
* + Create 6 subnets (2-public, 4-private).
  + Create first subnet.
  + Click on subnet, click on create subnet, select our VPC (my-project-vpc).
  + Give name tag as availability zones

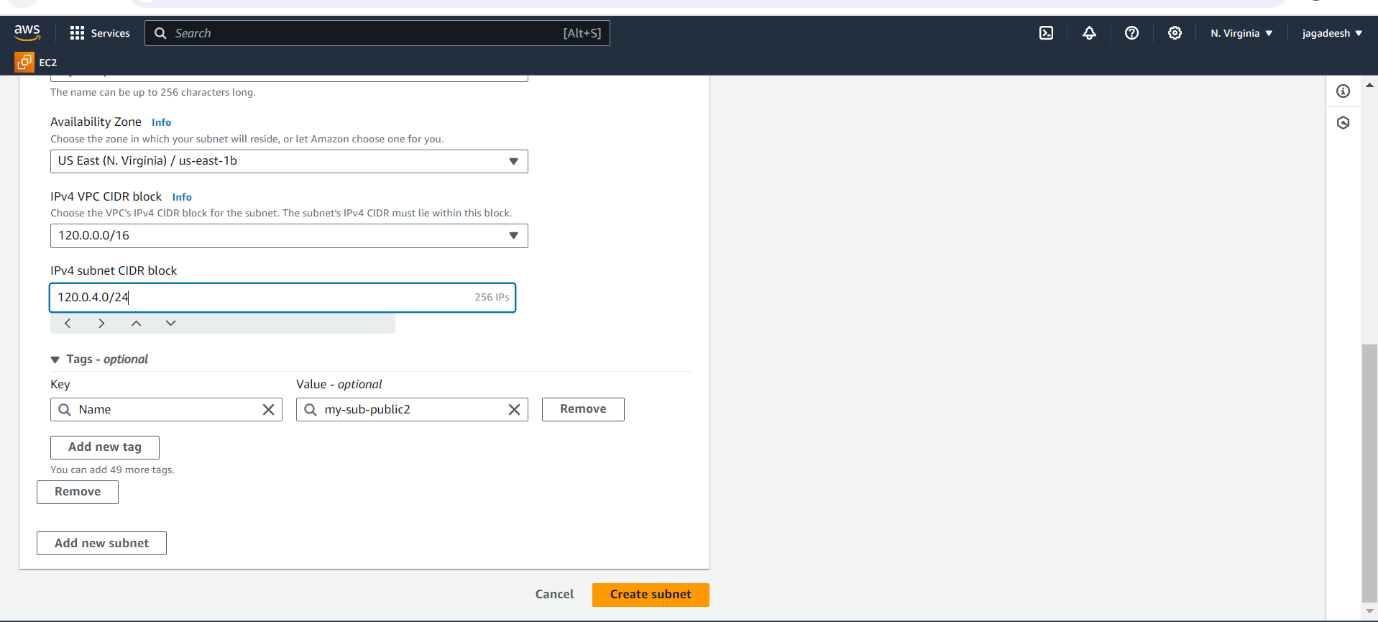




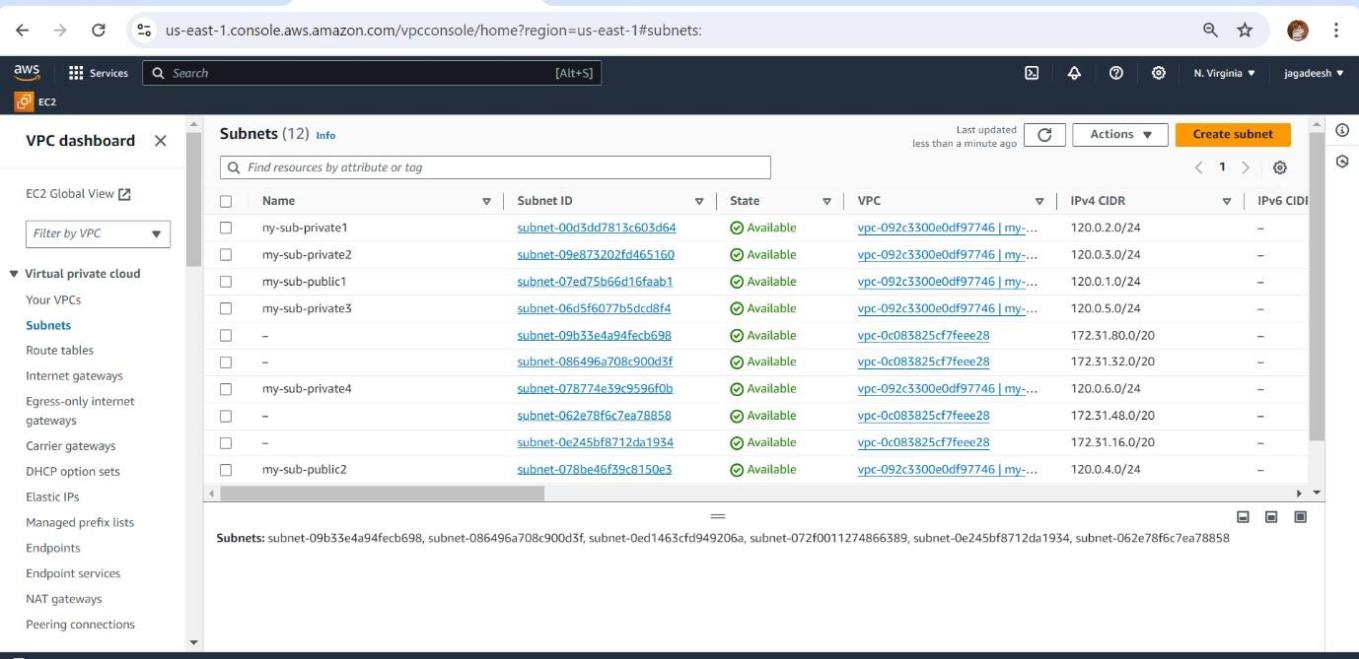




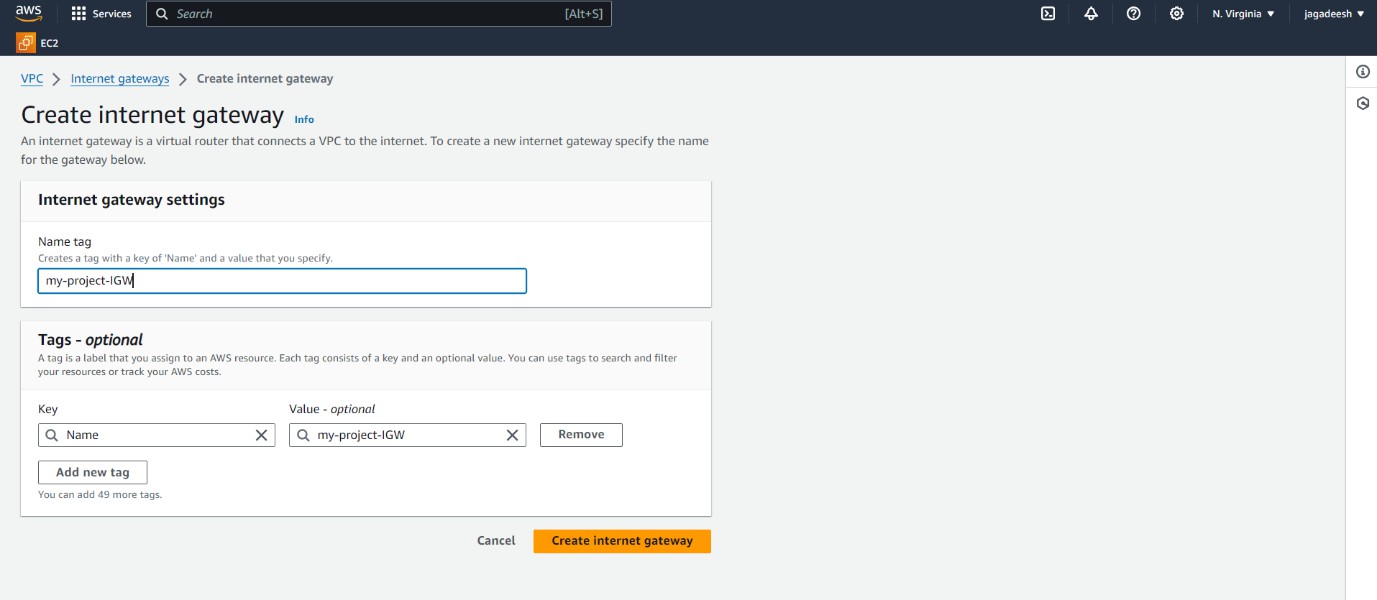




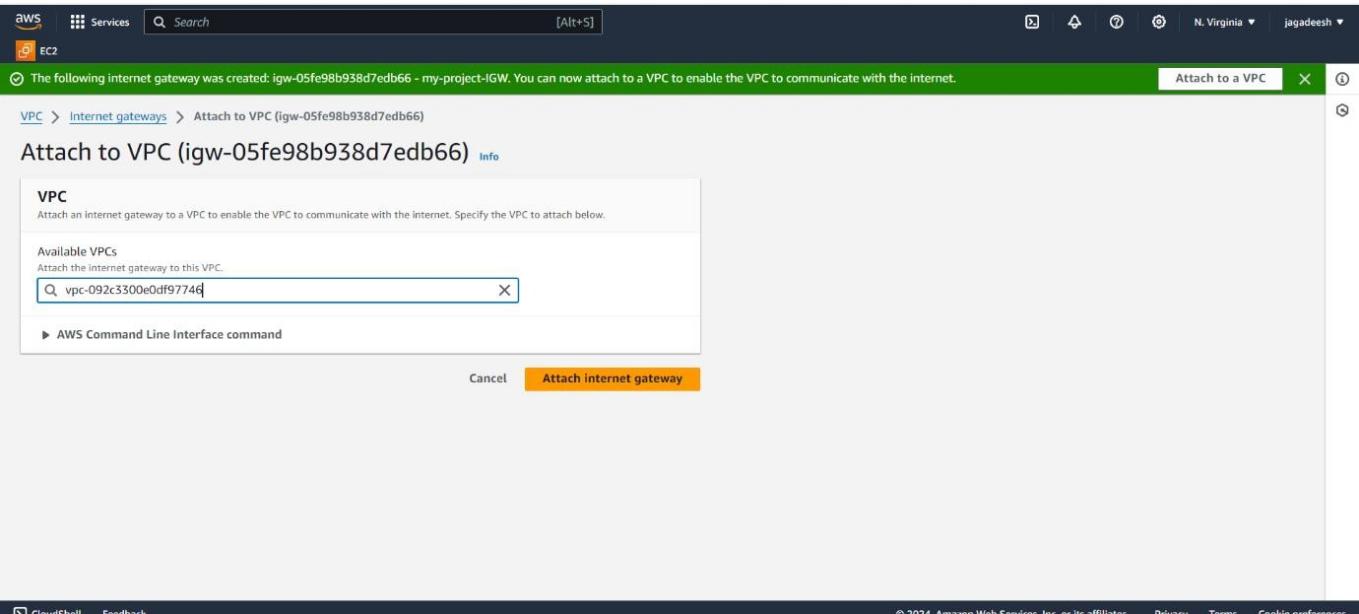
These are the subnets we created.



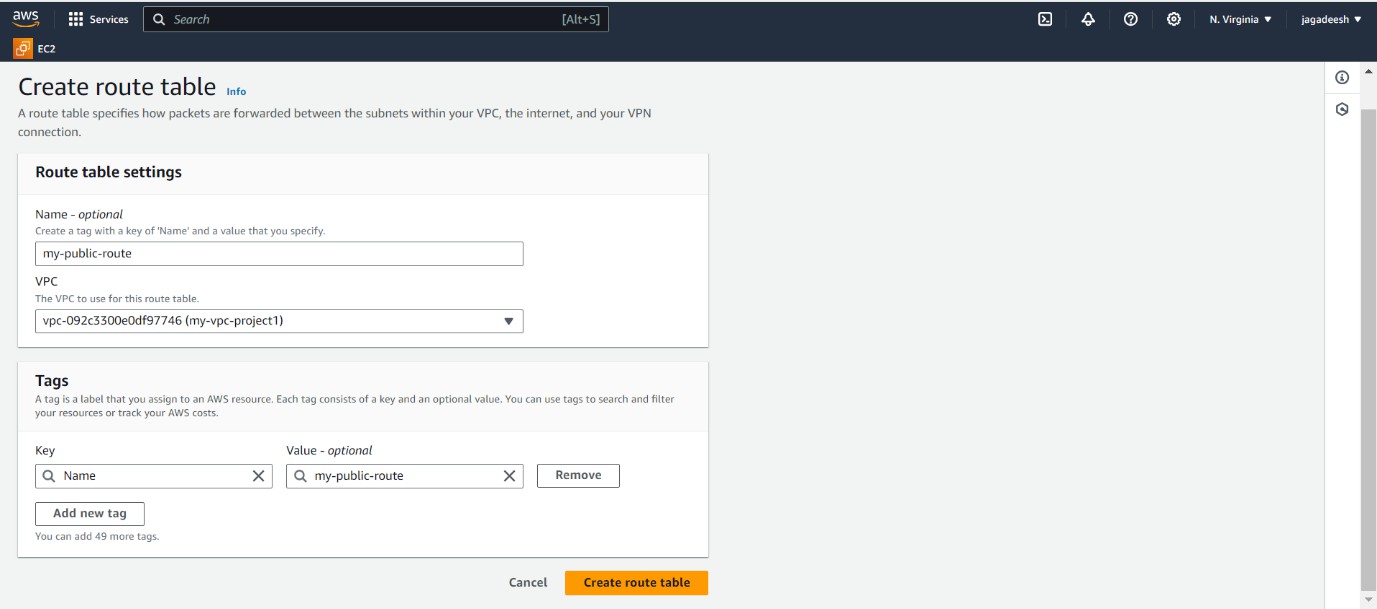
* + Create internet gateway name my-project-IGW



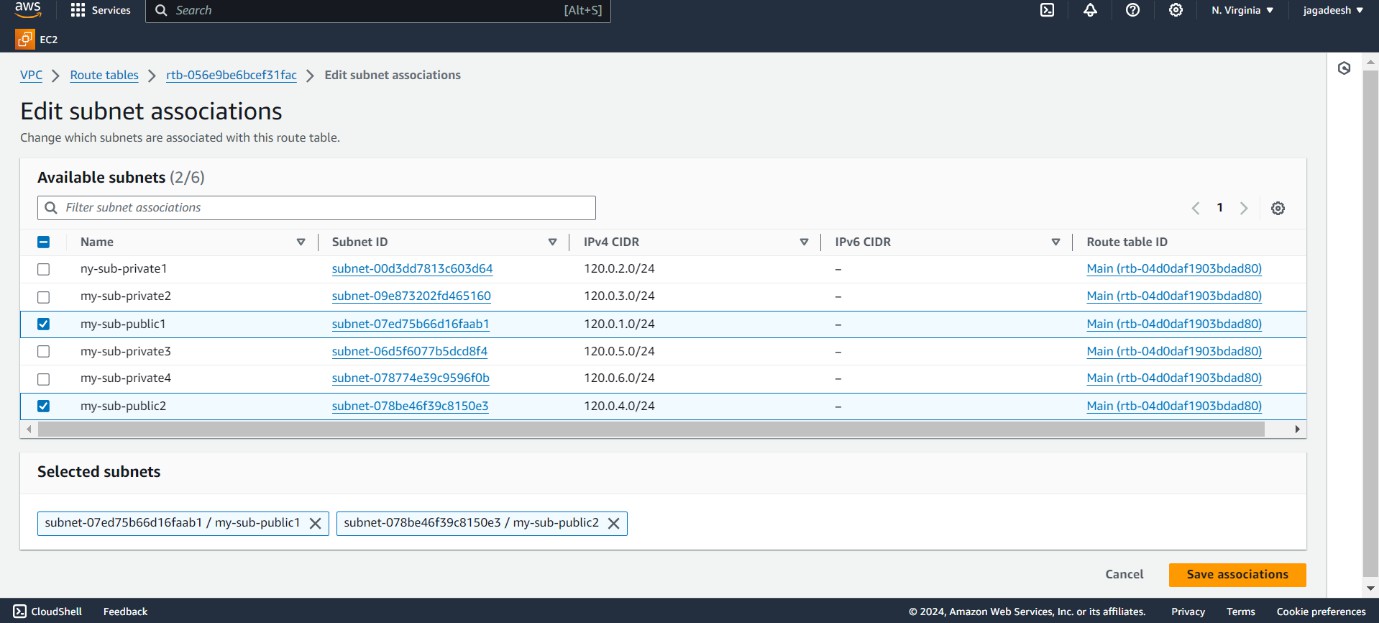
* + This igw is attached to VPC.
  + Go to actions in internet gate way and click on attach to VPC.
  + Select our VPC (my-vpc-project1). Click on attach internet gateway



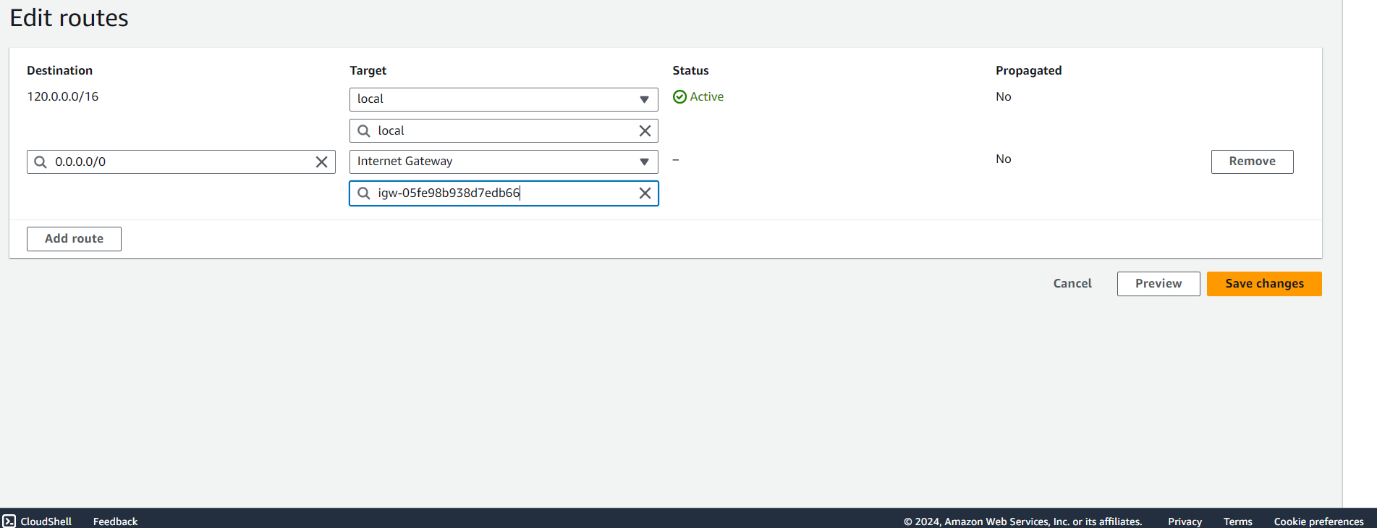
* + Create route table, give name as my-public-route
  + Select our VPC (my-vpc-project), create it.



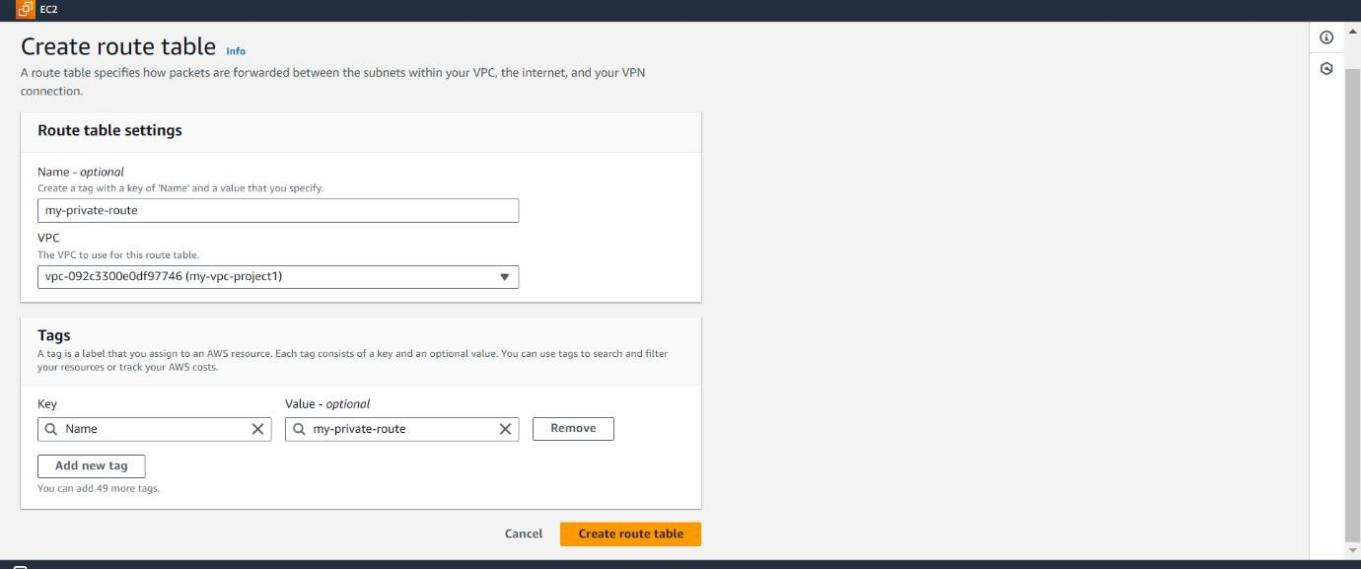
* + Click on route table id, open it.
  + Go down click on edit subnet association.
  + Select both public subnet and click on save association

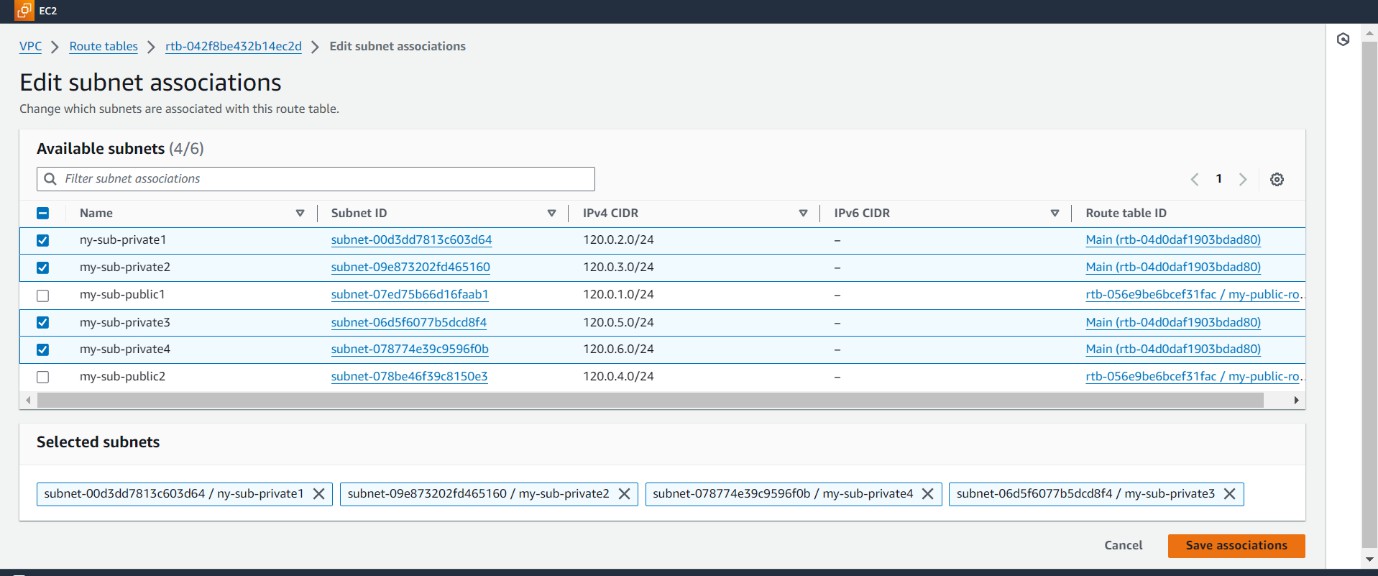


* + G0 actions and click on edit routes .
  + Click on add routes give all traffic (0.0.0.0/0) and select our internet gateway, save changes .



* + Create private route table name as my-private-route
  + Select our VPC, (my-vpc-project1), create it.
  + Click on route table id, open it.
  + Go down click on edit subnet association.
  + Select all private subnet and click on save association

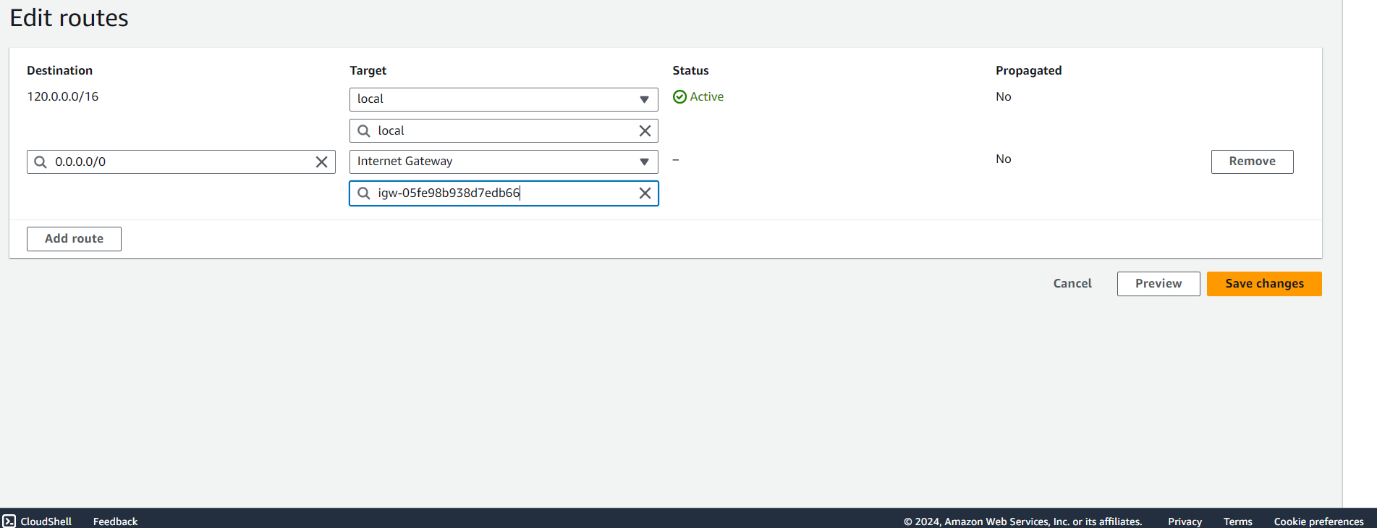




* + Create NAT gateway, give name as my-nat.
  + Select public subnet(my-sub-public1).
  + Select connectivity type as IPV4.
  + Click on allocate Elastic IP.

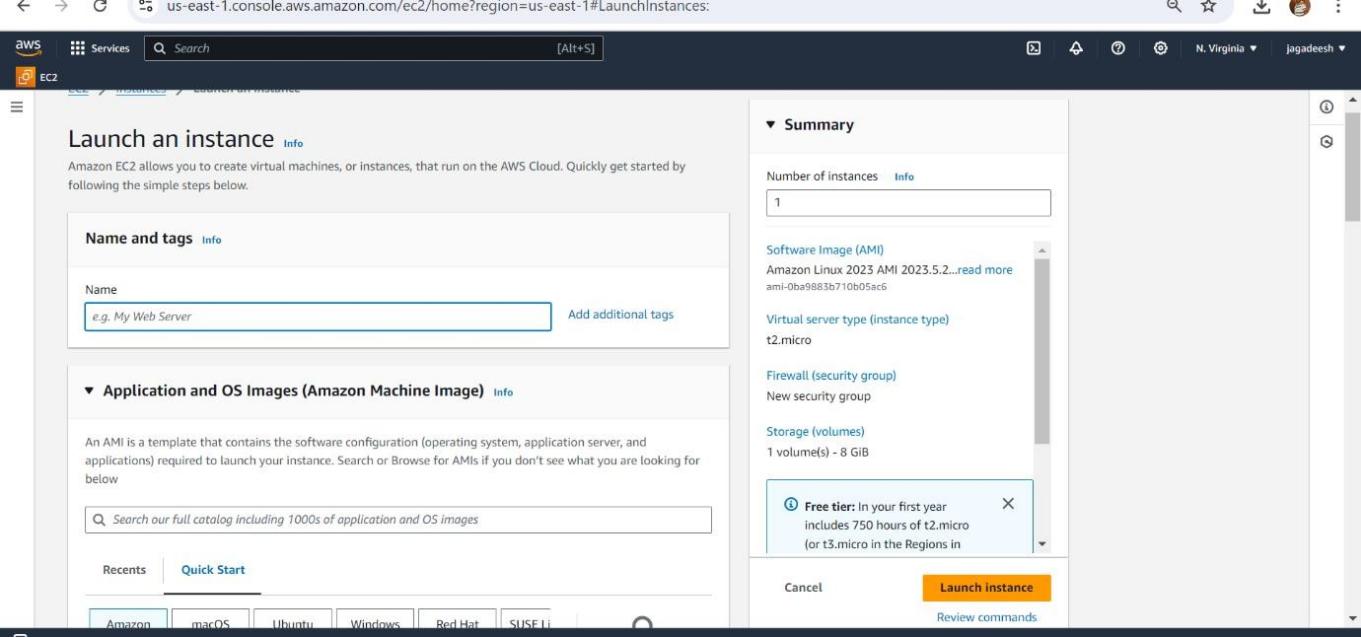


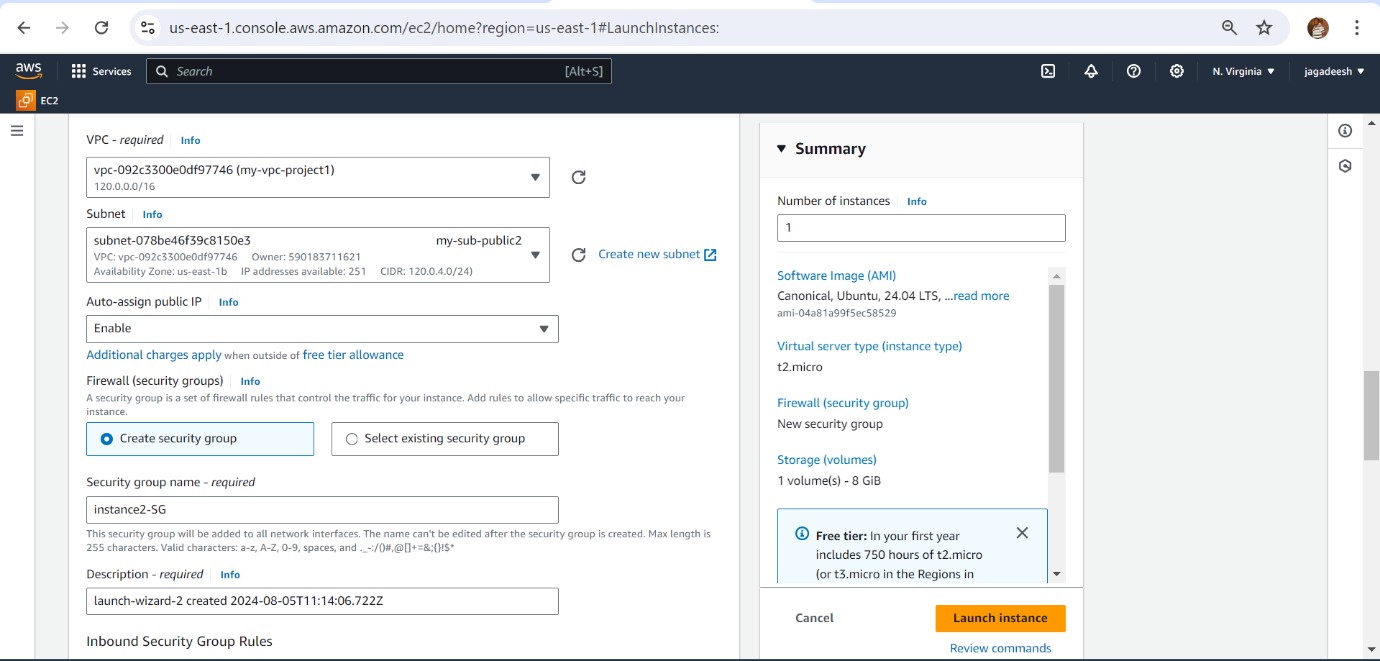
* + Now go to private route and click on actions.
  + Click on edit routes and add route.
  + Give all traffic (0.0.0.0/0) and select NAT gateway



**Step2:** Launch an EC2 instance.

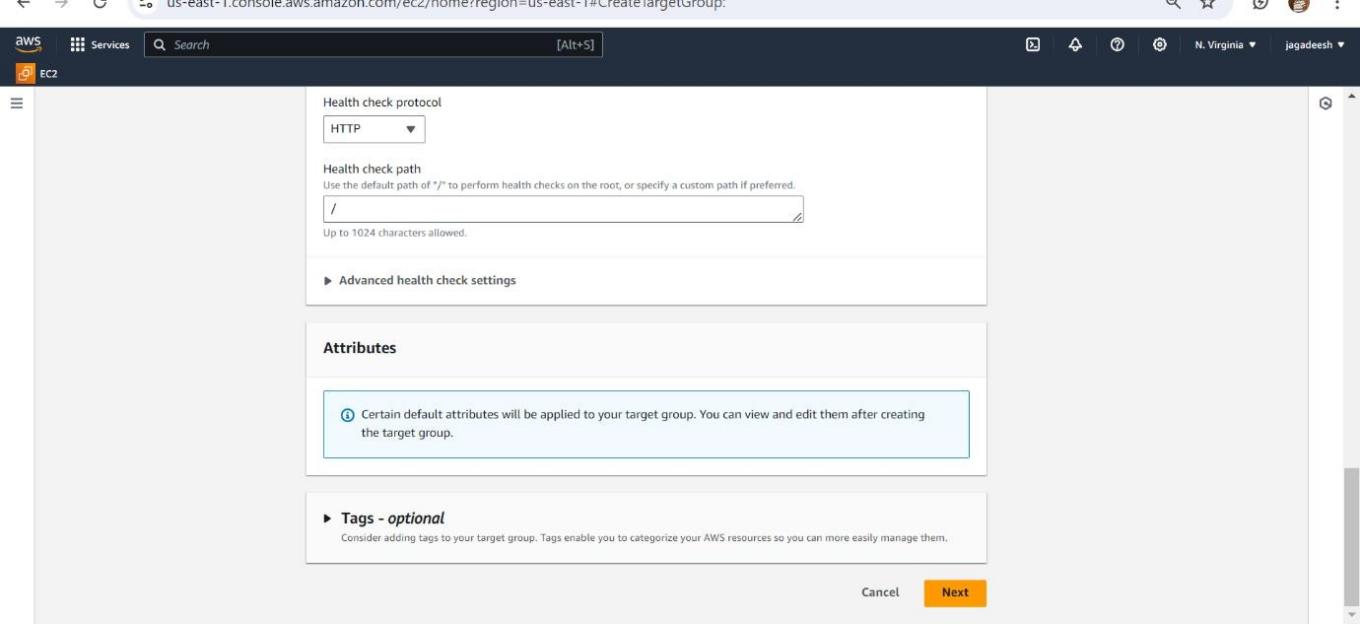
* + Go to EC2 dashboard click on launch instance.
  + Name as and select ami as ubuntu
  + Instance type as t2.micro and key pair as project.
  + Click on edit network settings, select our VPC and public subnet.
  + Auto assign IP enable and create a security group as project-sg.
  + Launch the instance.

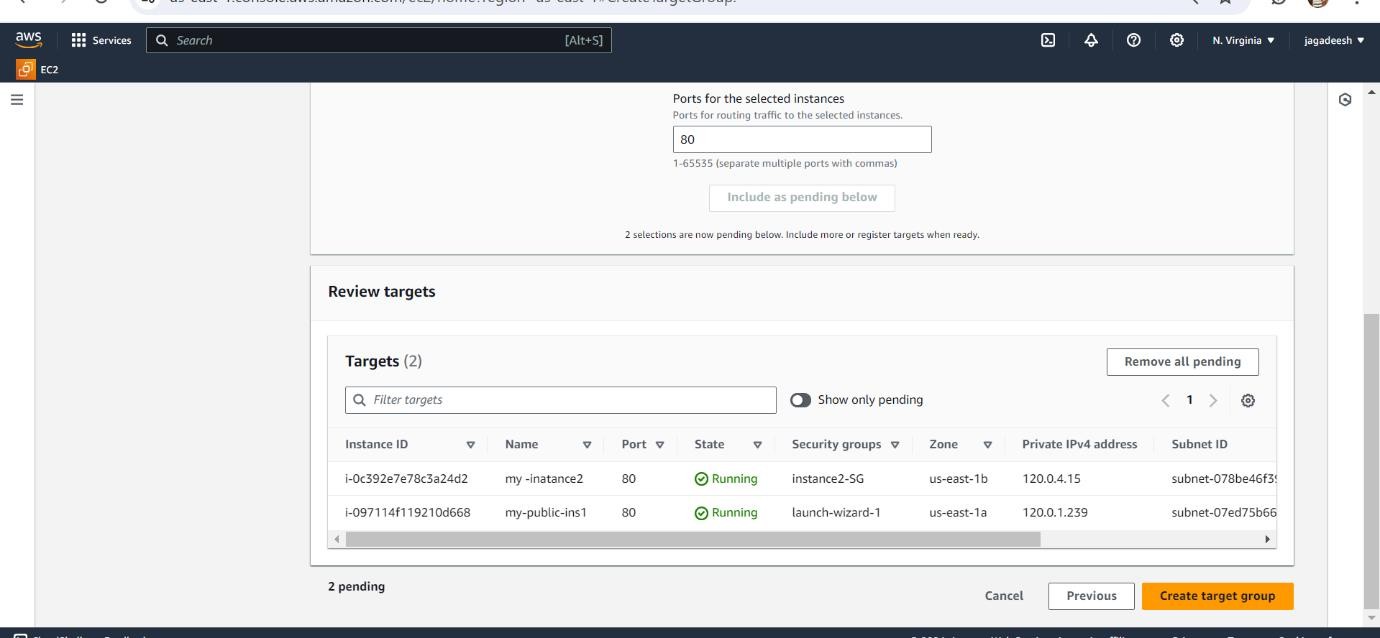


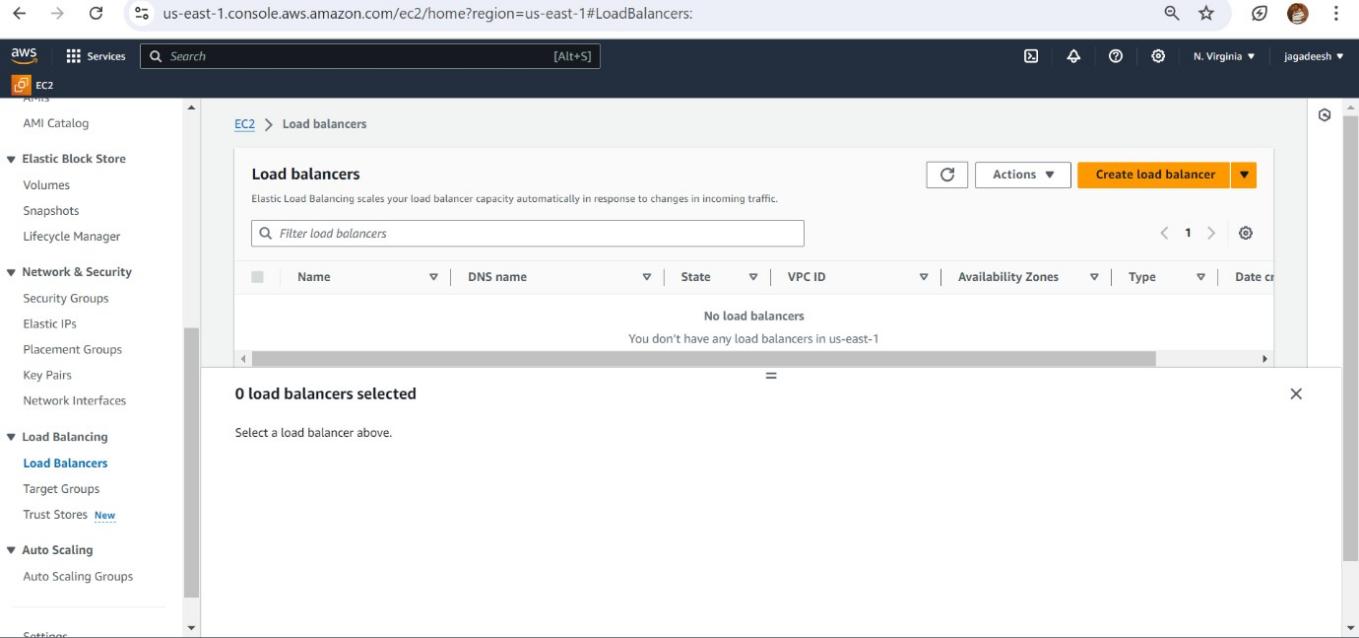


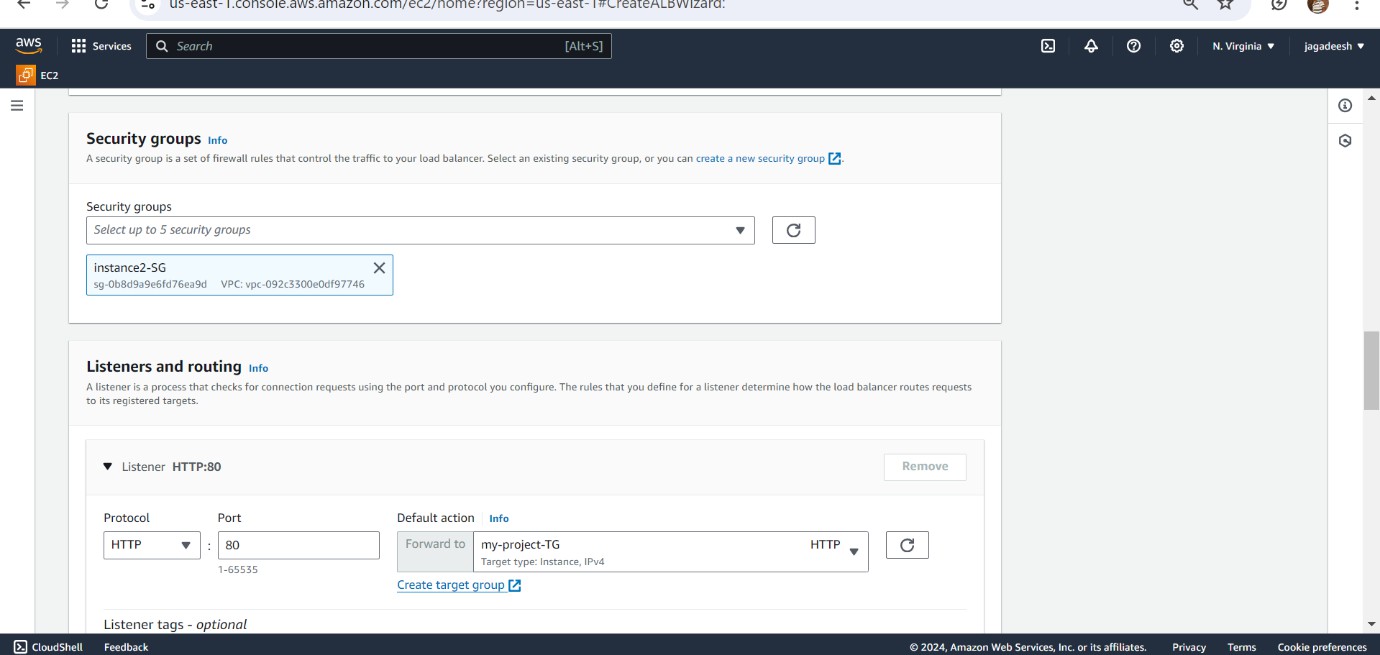
# Step 3:

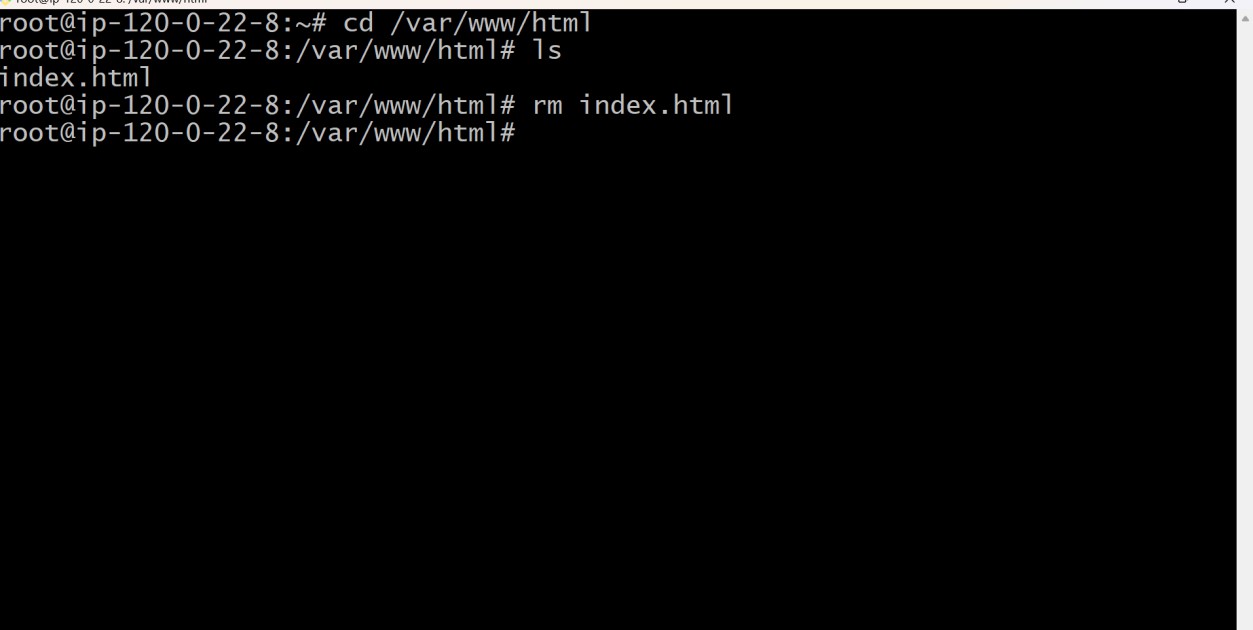
* + Now we create 2 LoadBalancer
  + One is public instances and another one is private instances
  + Then create target groups and attached the instance and check the
  + Instance ssh -i copied and paste on the server
  + Then next apt update -y apt install -y cd /var/www/html
  + Systemctl restart apache2
  + We take another instance like same procedure

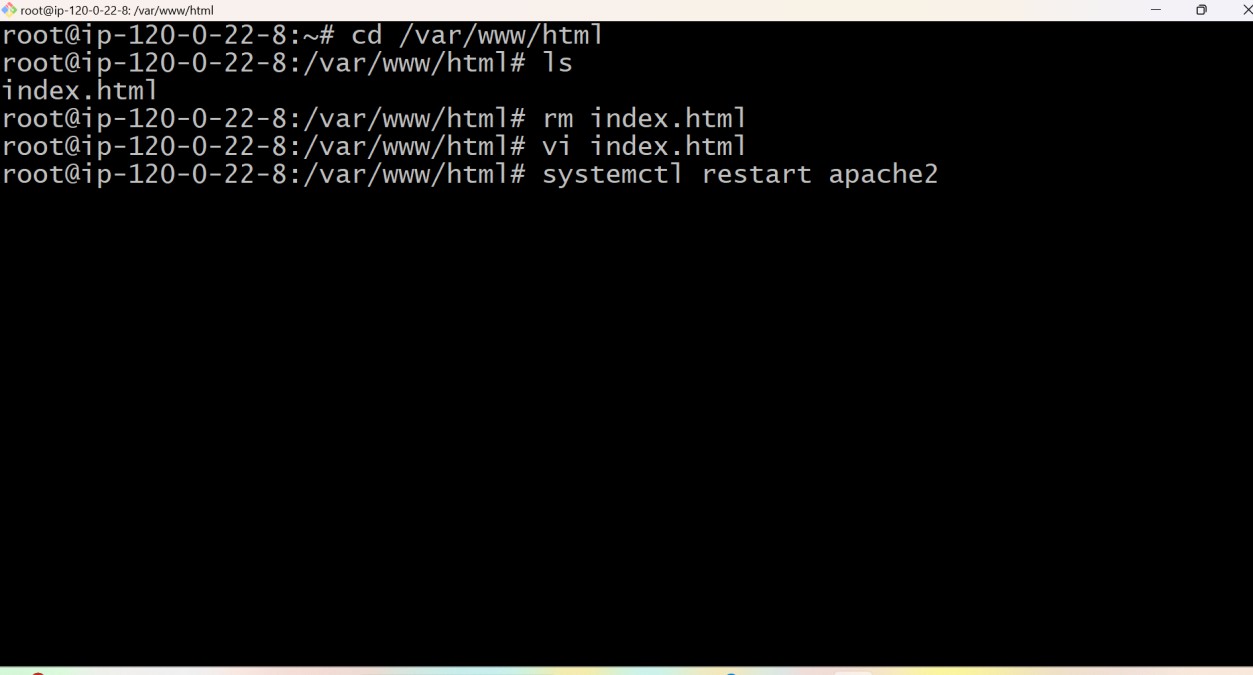


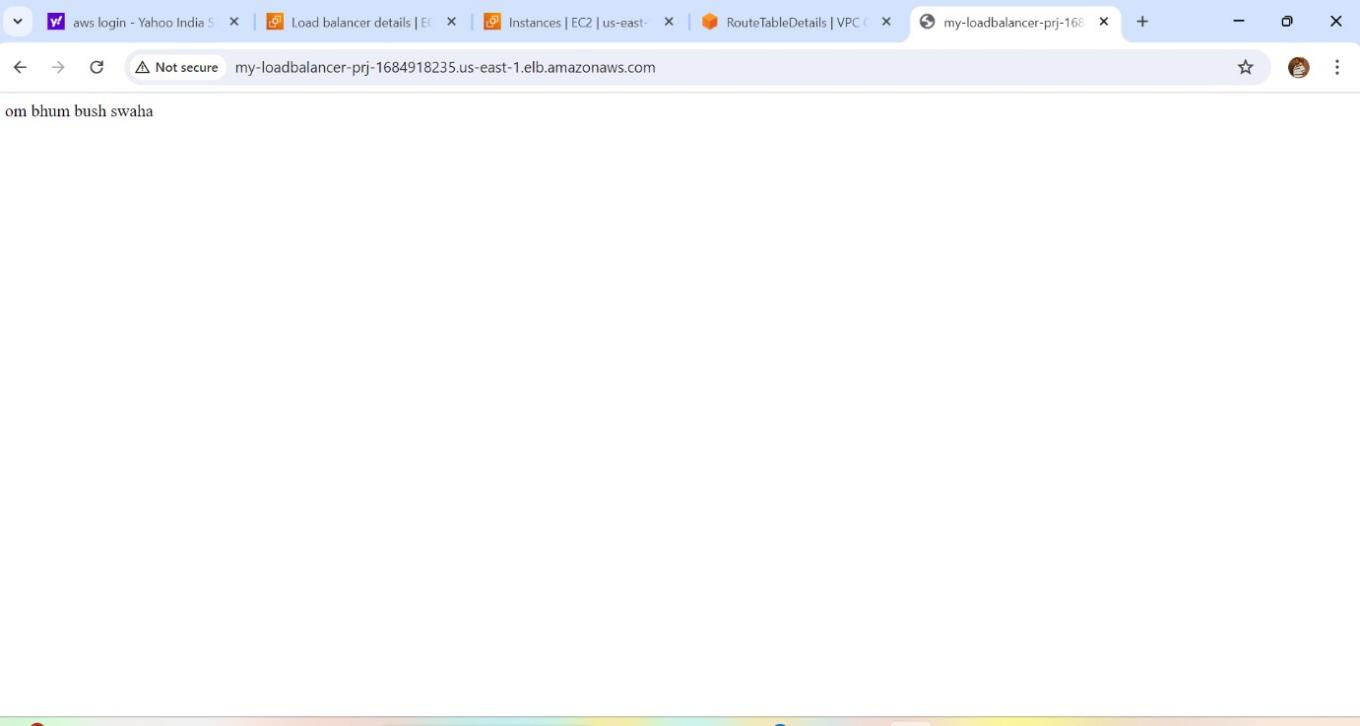


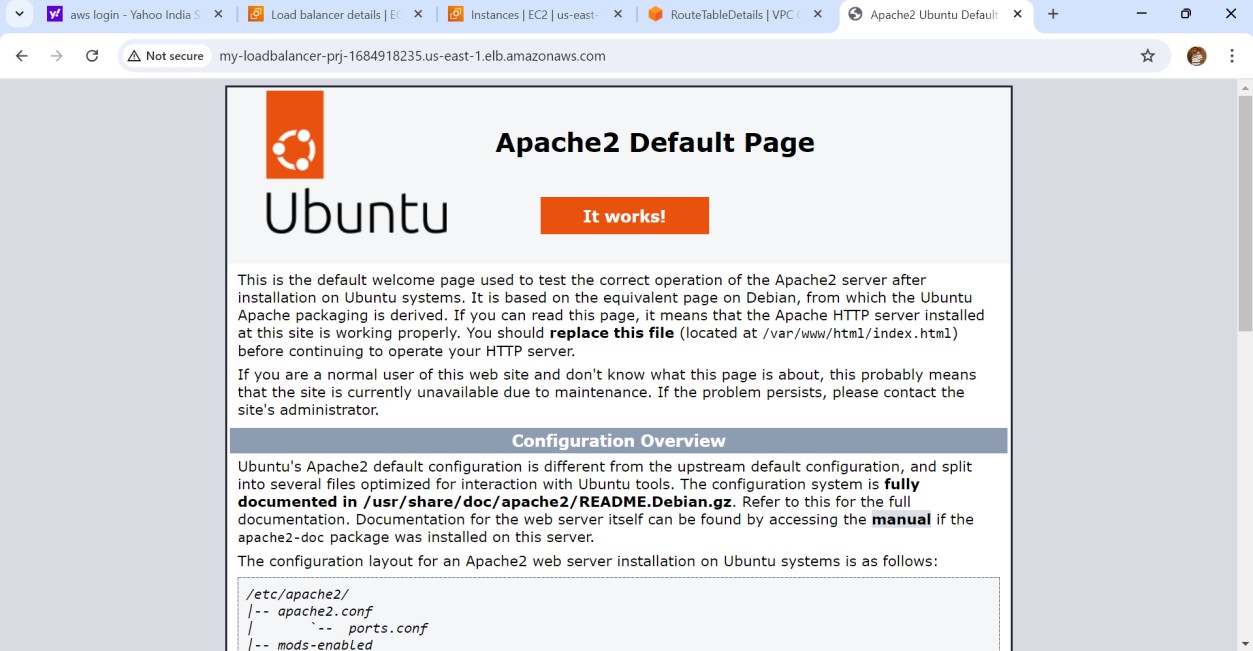






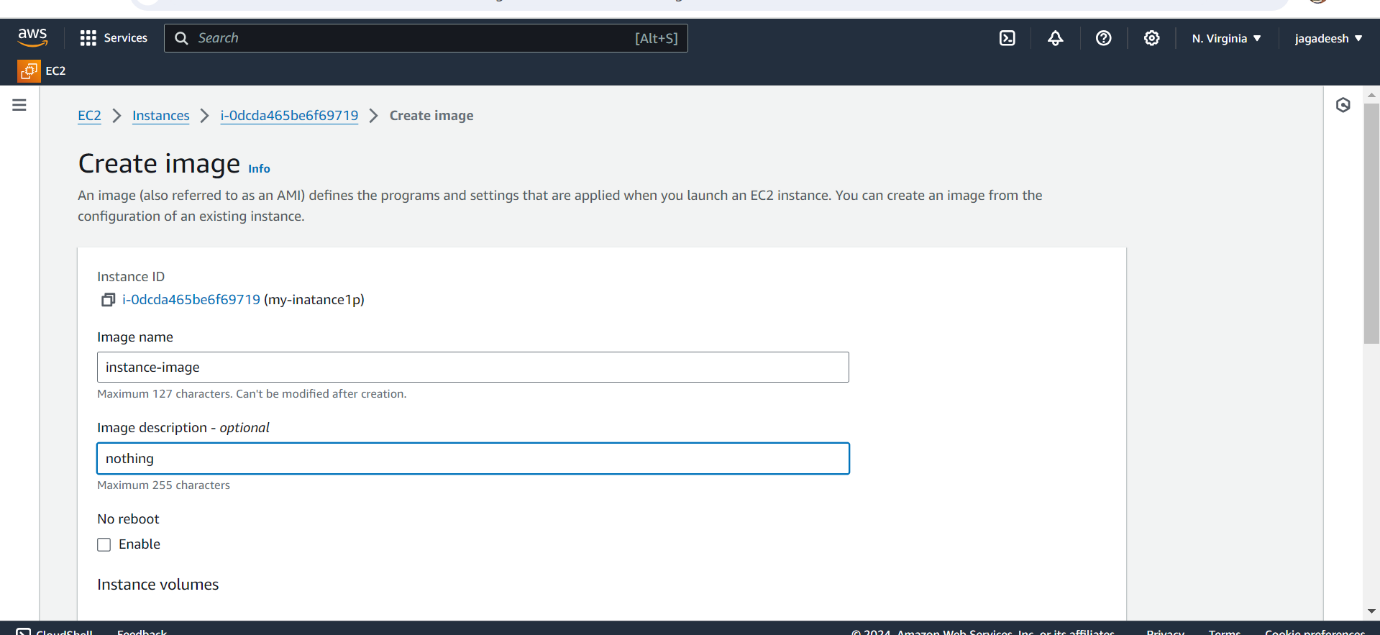


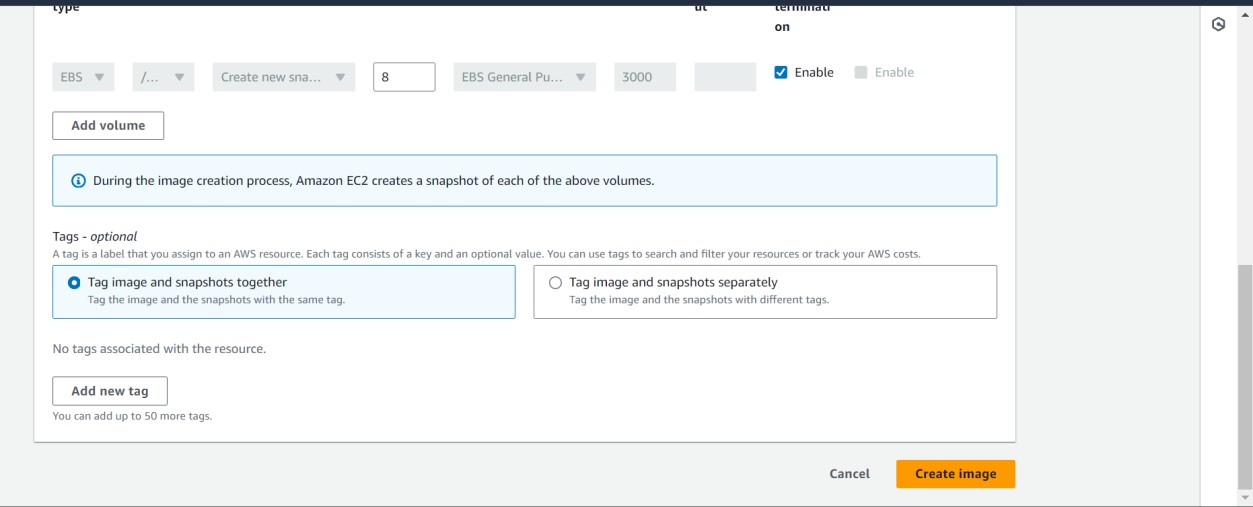




**Step 4:** Create an AMI (image)

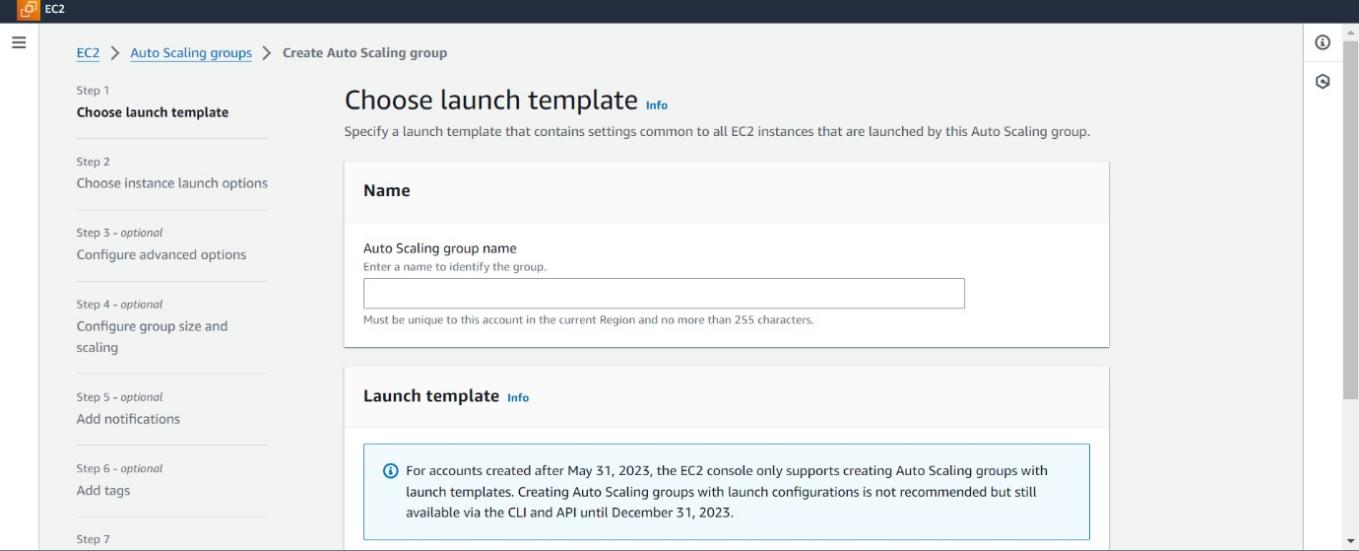
* After running the instance, click on actions.
* Click on image and templates and click on create image.
* Give image name as my-image.
* Wait until the image is available.

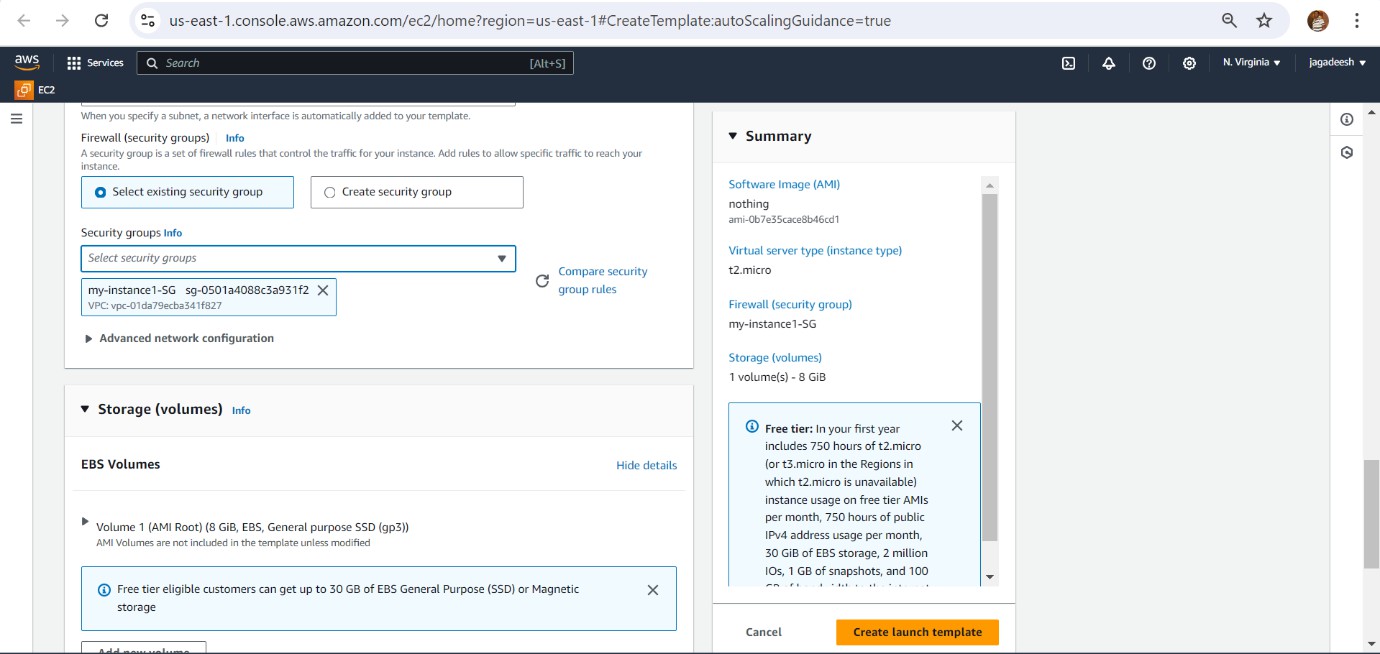




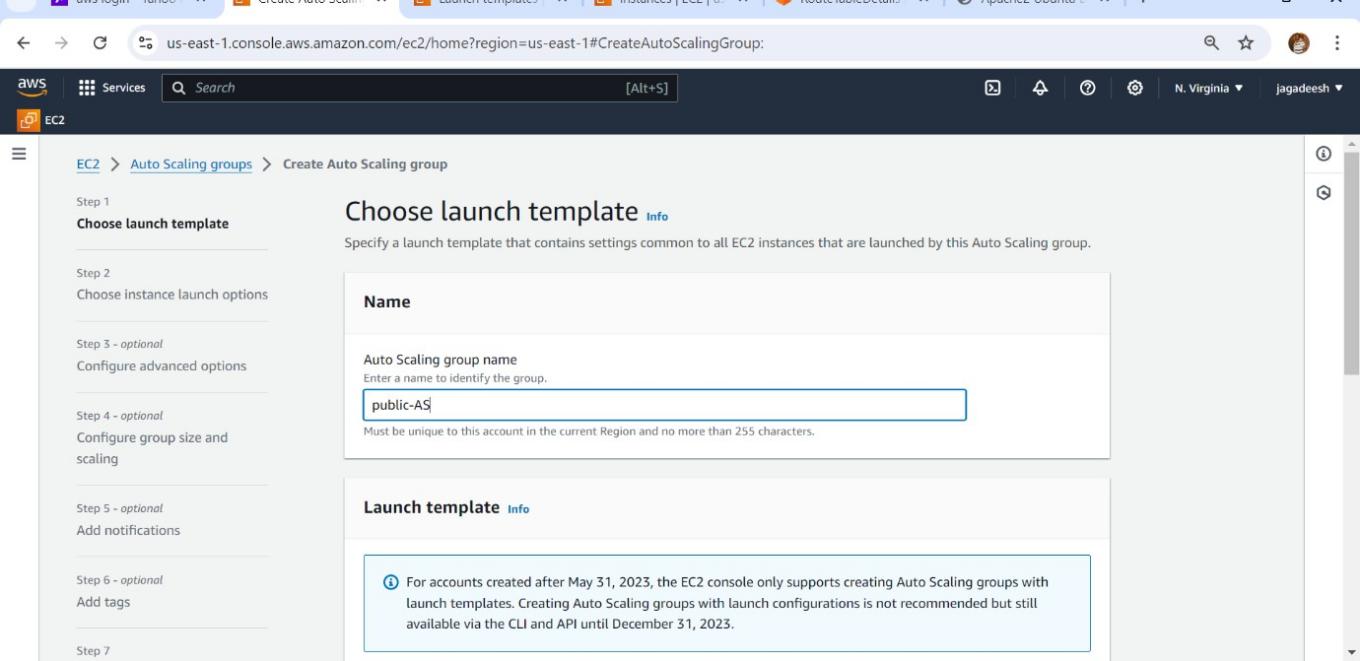
**Step 5:** Create Autoscaling group.

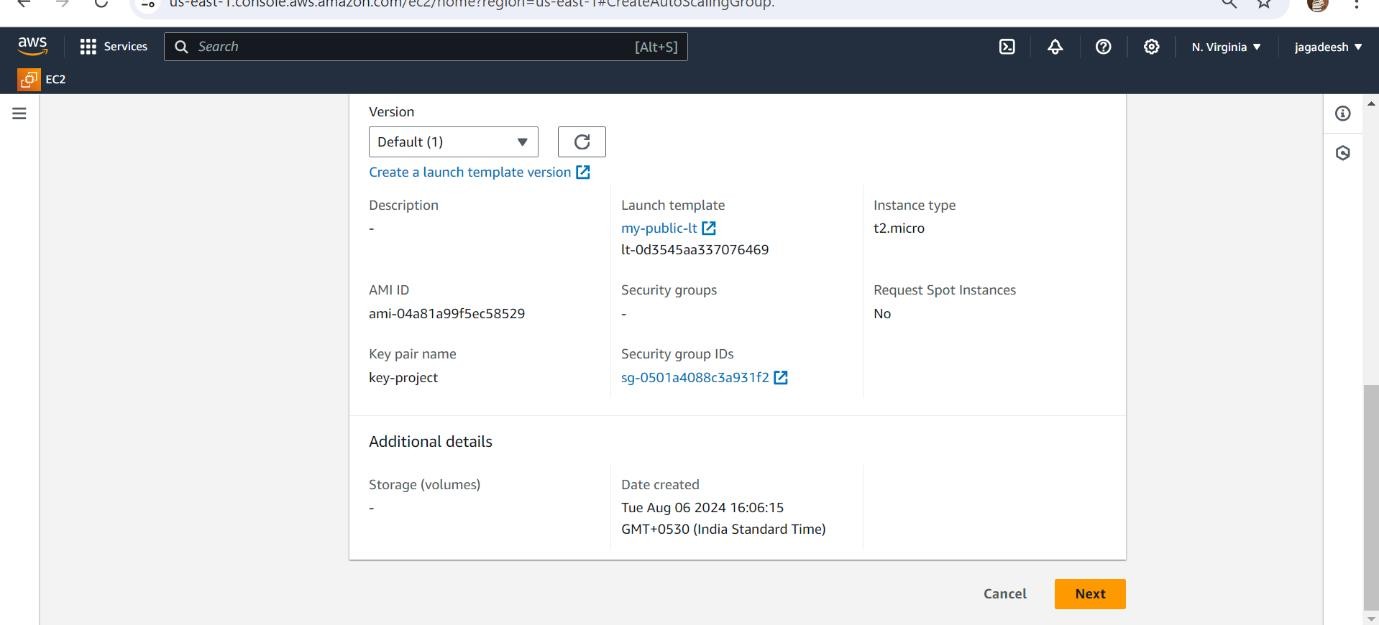
* For creating autoscaling group we need to create an launch template.
* After available of image. Click on create a launch template.
* Template name as my-public-template, description as nothing.
* Select AMI’s as share with me, select my-image.
* Instance type as t2.micro and key pair as project.
* Select existing security group (my-instance-SG) which is used to launch an EC2 instance
* Now click on create launch template.

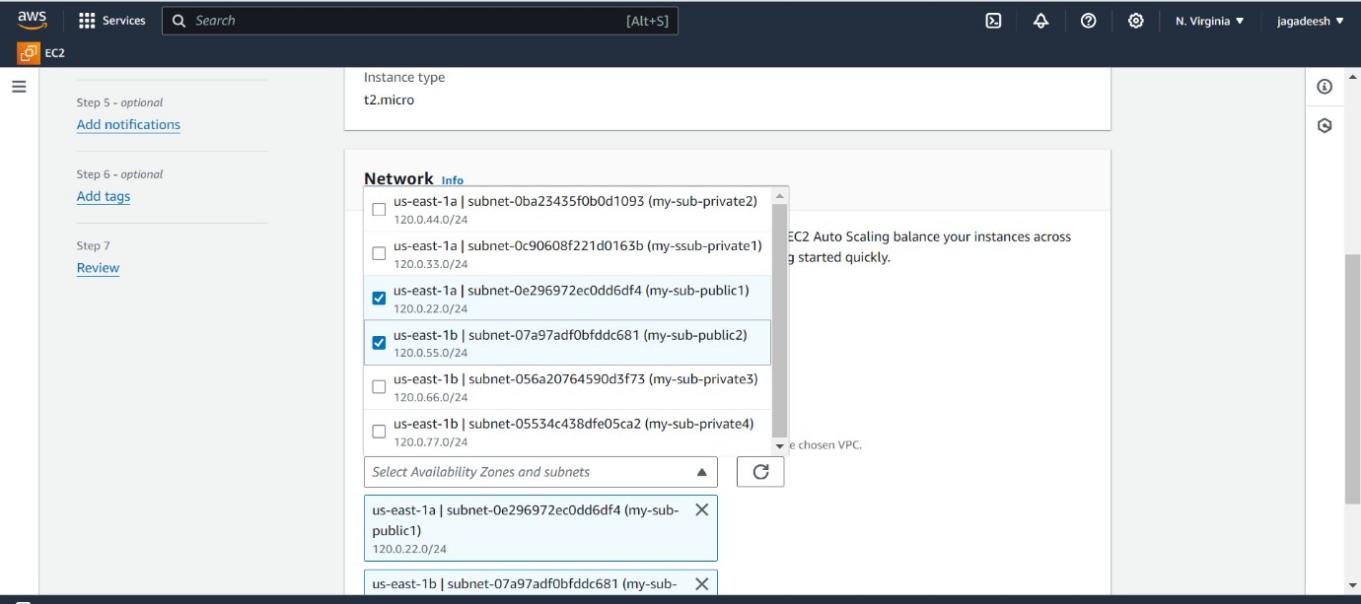


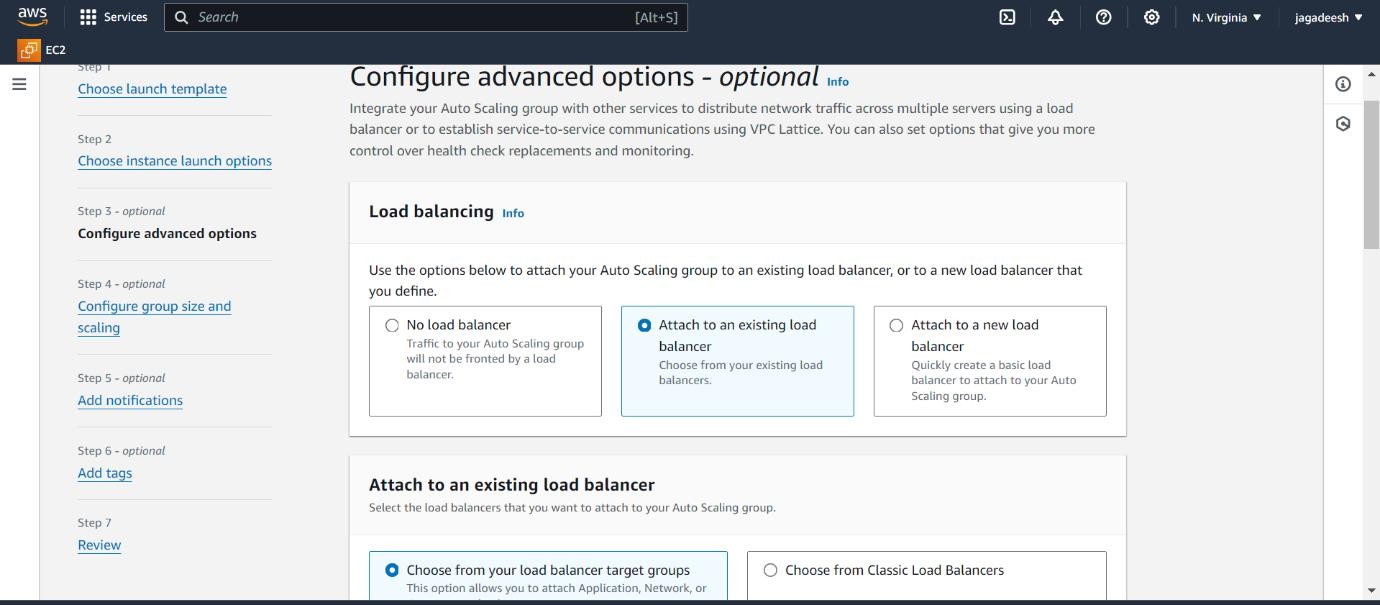


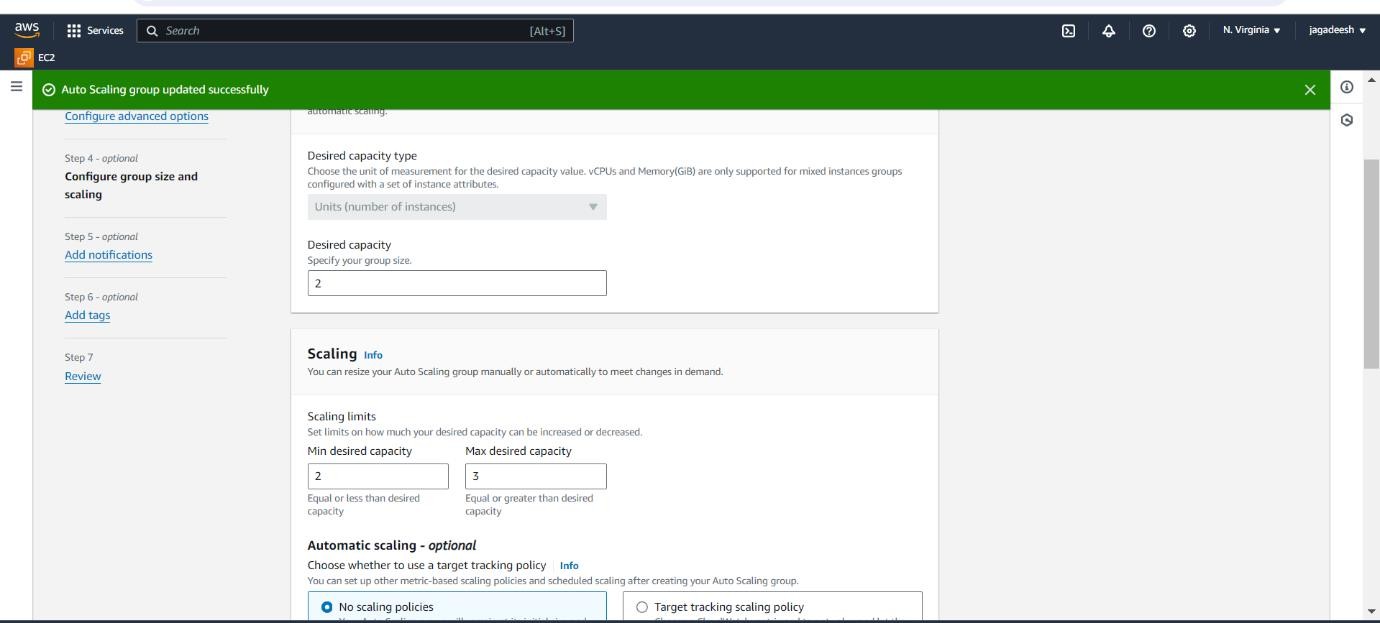
* Open autoscaling group.
* Click on create autoscaling group.
* Give name as public-AS
* Choose the created launch template (my-public-tm) and click on next.
* Select our VPC (my-project-vpc), and both public subnets.
* Click on next and click on no load balancer
* Give desired capacity as 2 in sizing desired capacity min – 2 and max – 3
* and click on next.
* Click on add notification, give name as my-topic and give email.
* Click on next and click on create auto scaling group.

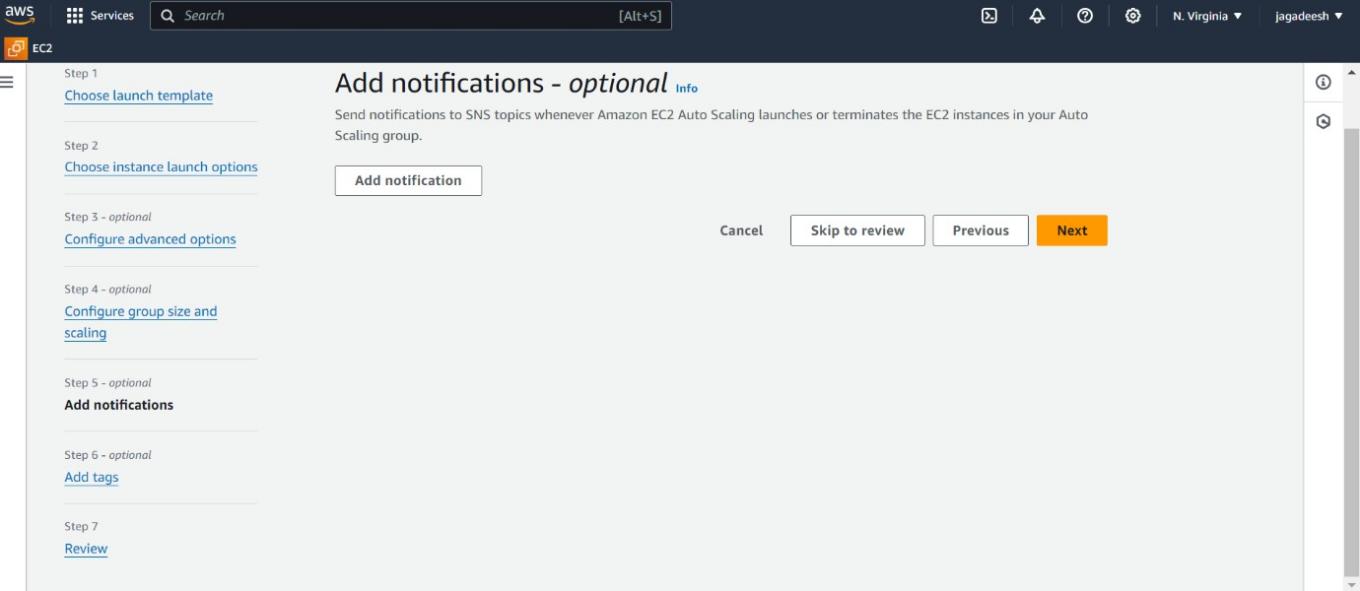




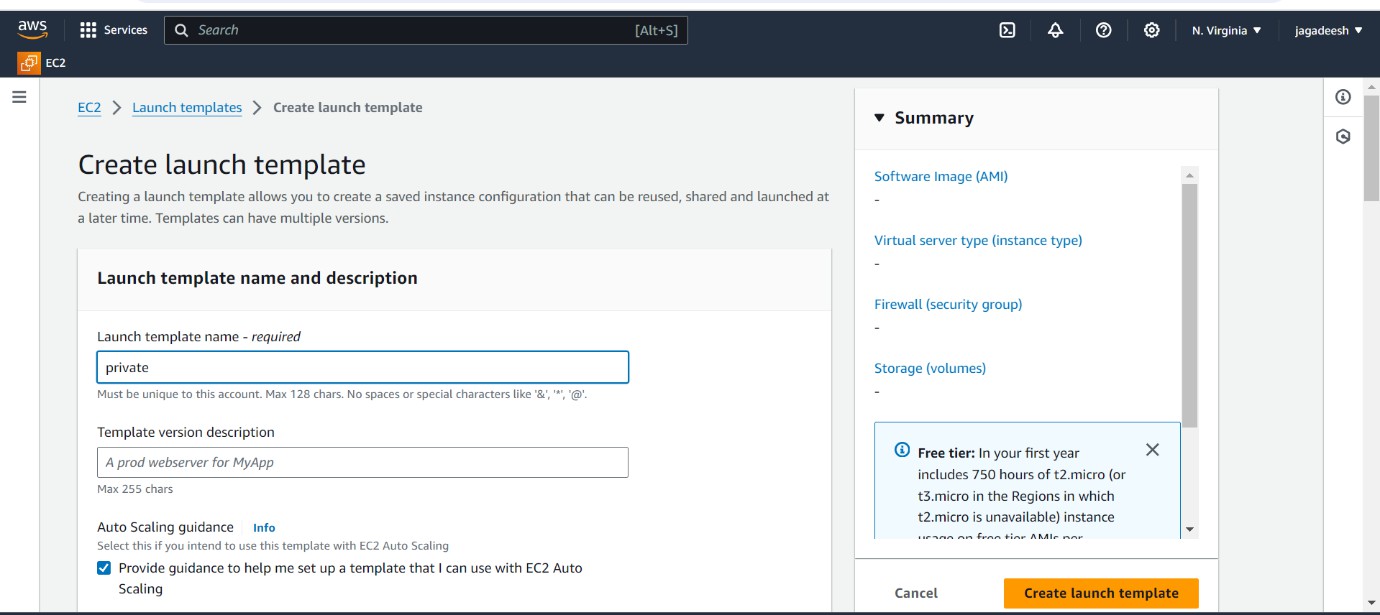


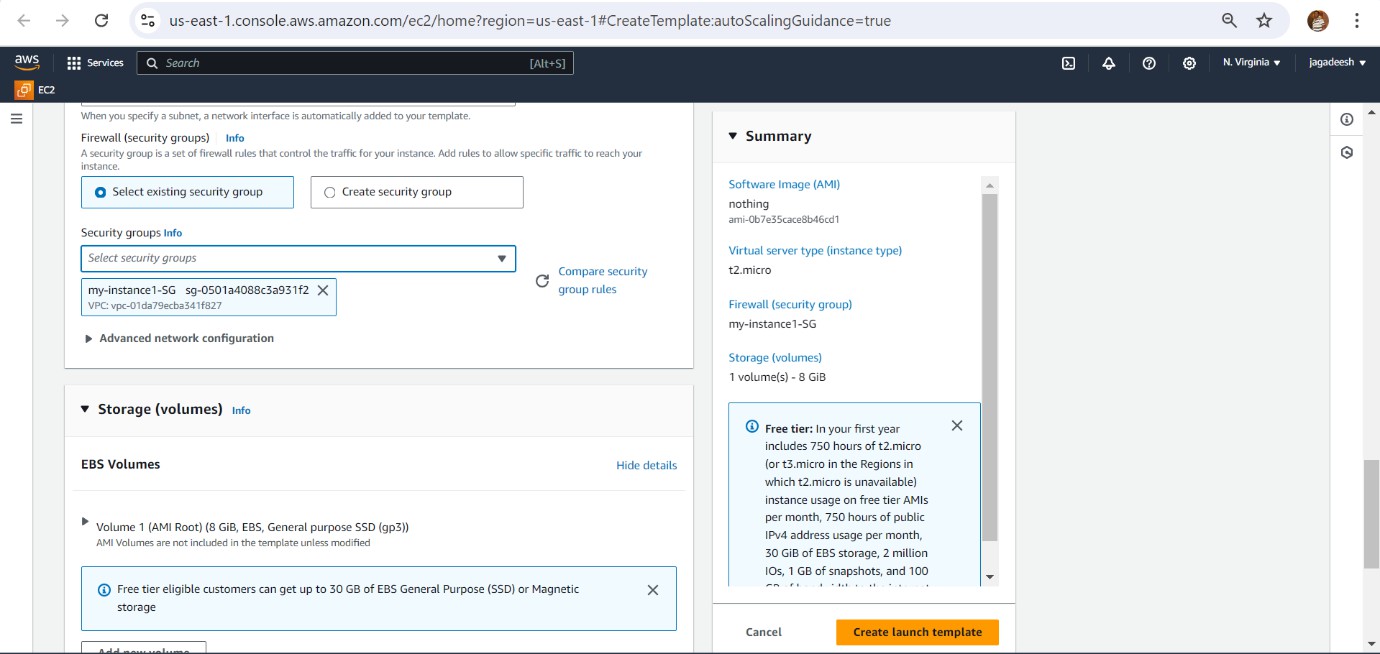




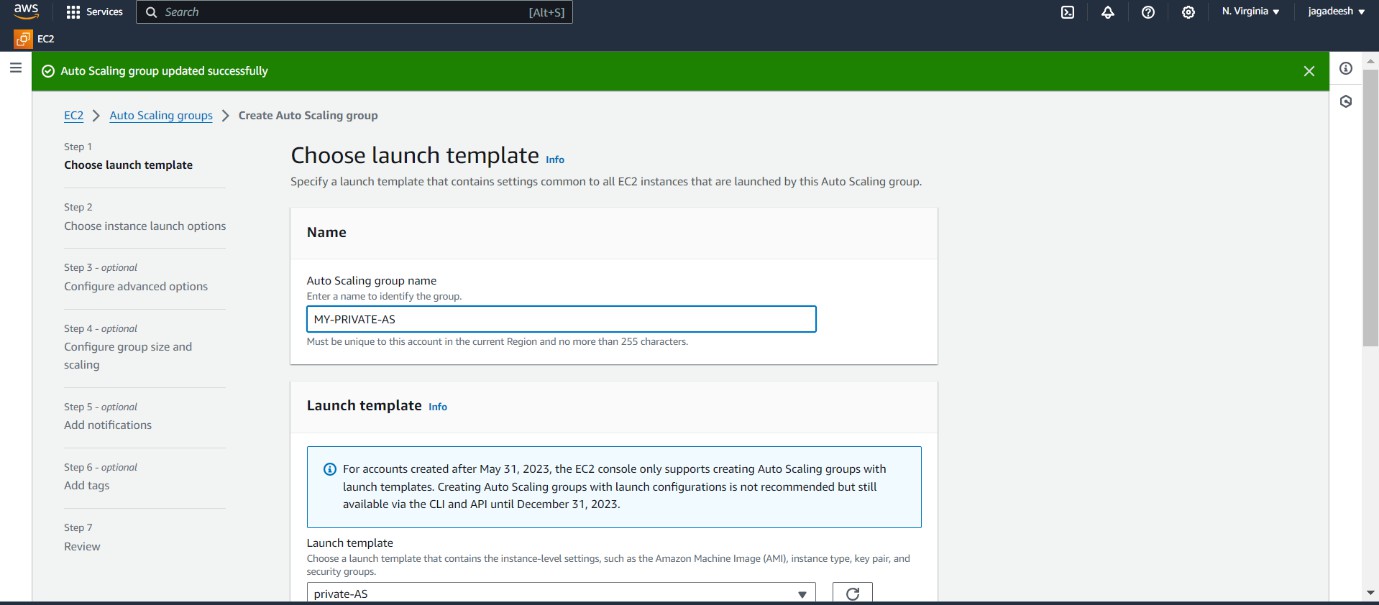


* In similar way create another launch template name as private-tm and create auto scaling group name as MY-PRIVATE-AS
* In it select create VPC and give two private subnets

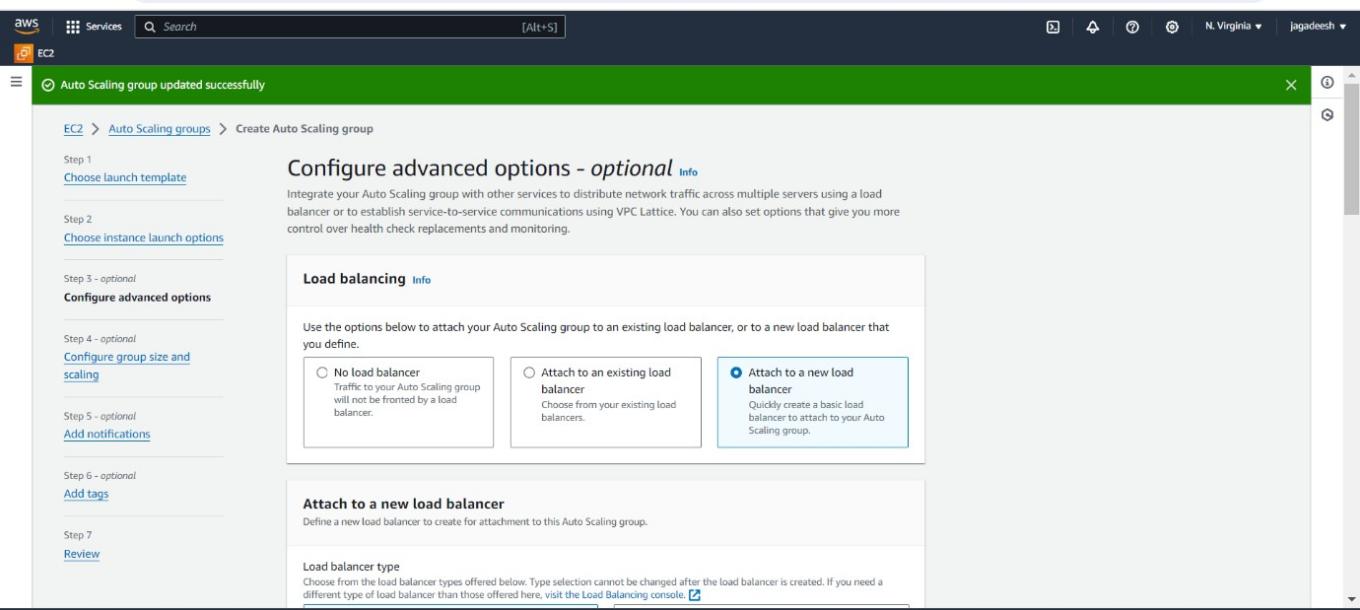


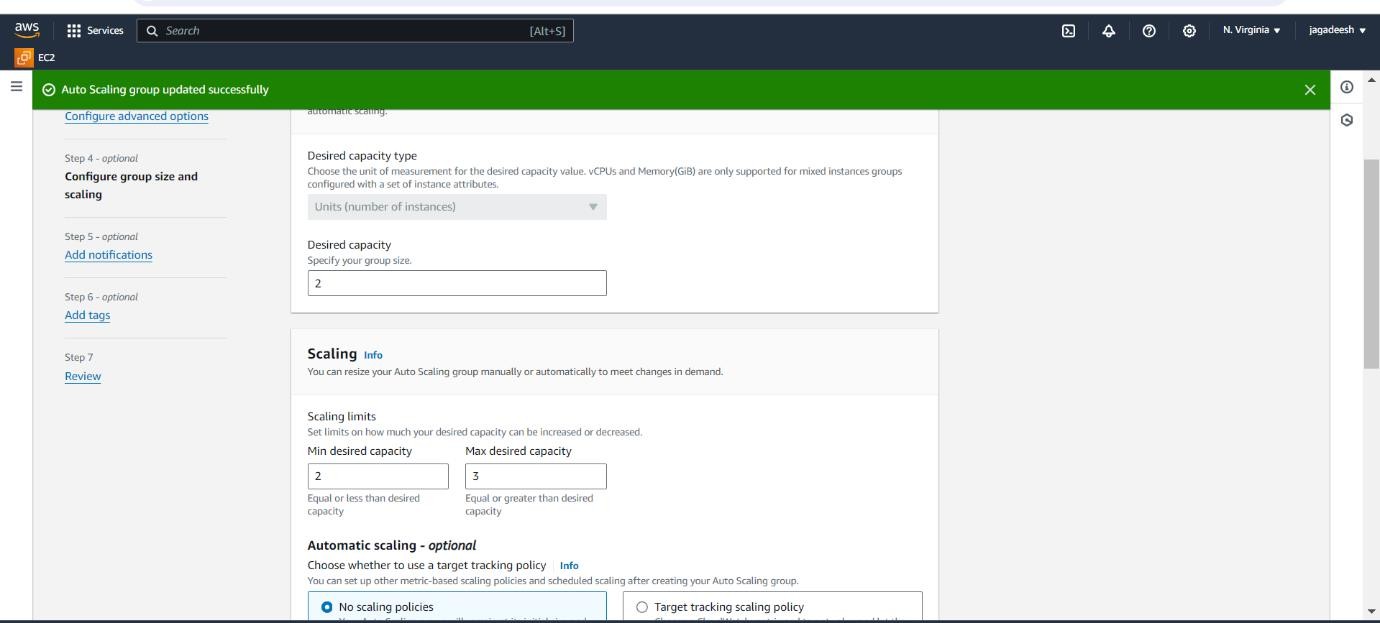


Now create private auto scaling

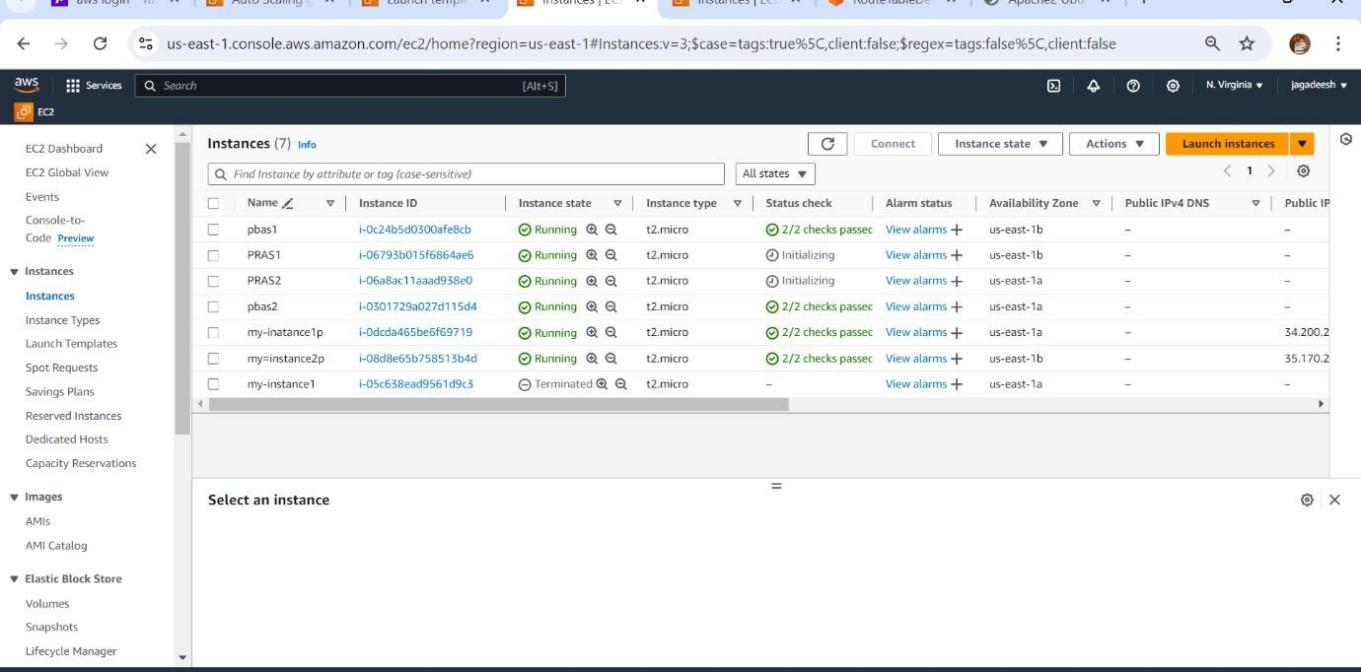






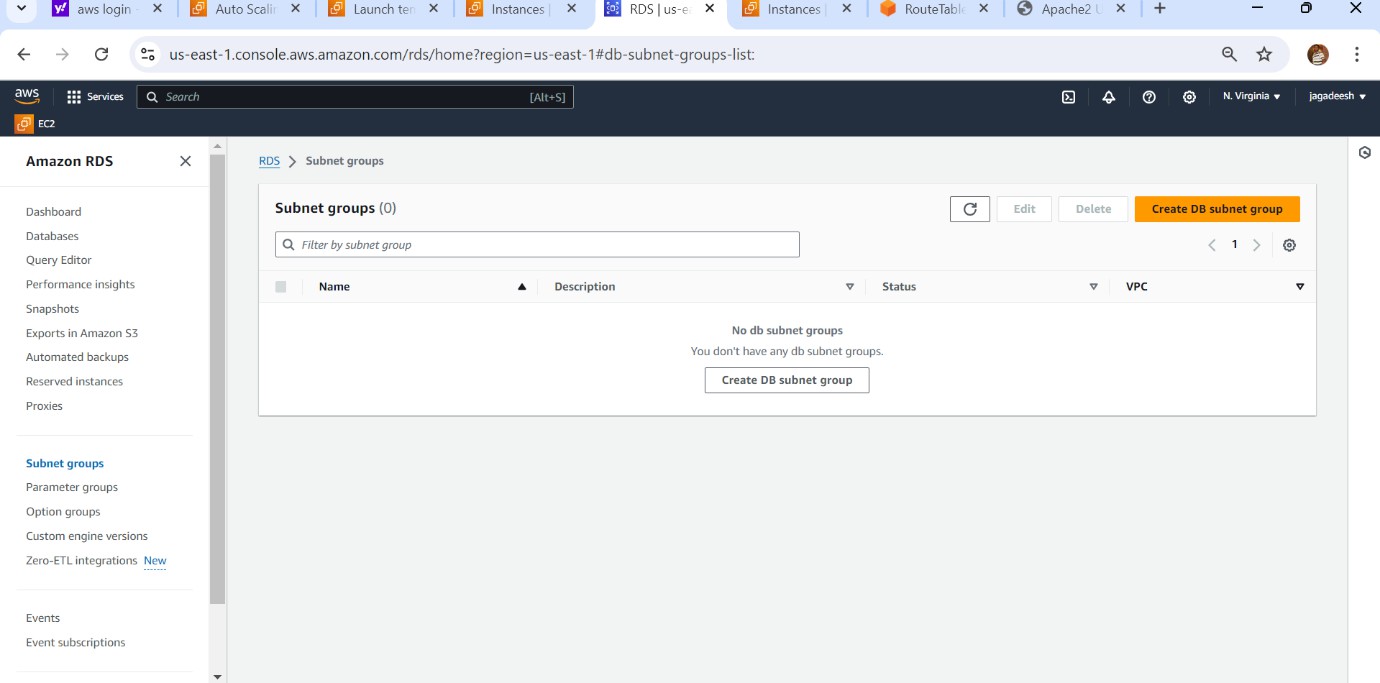


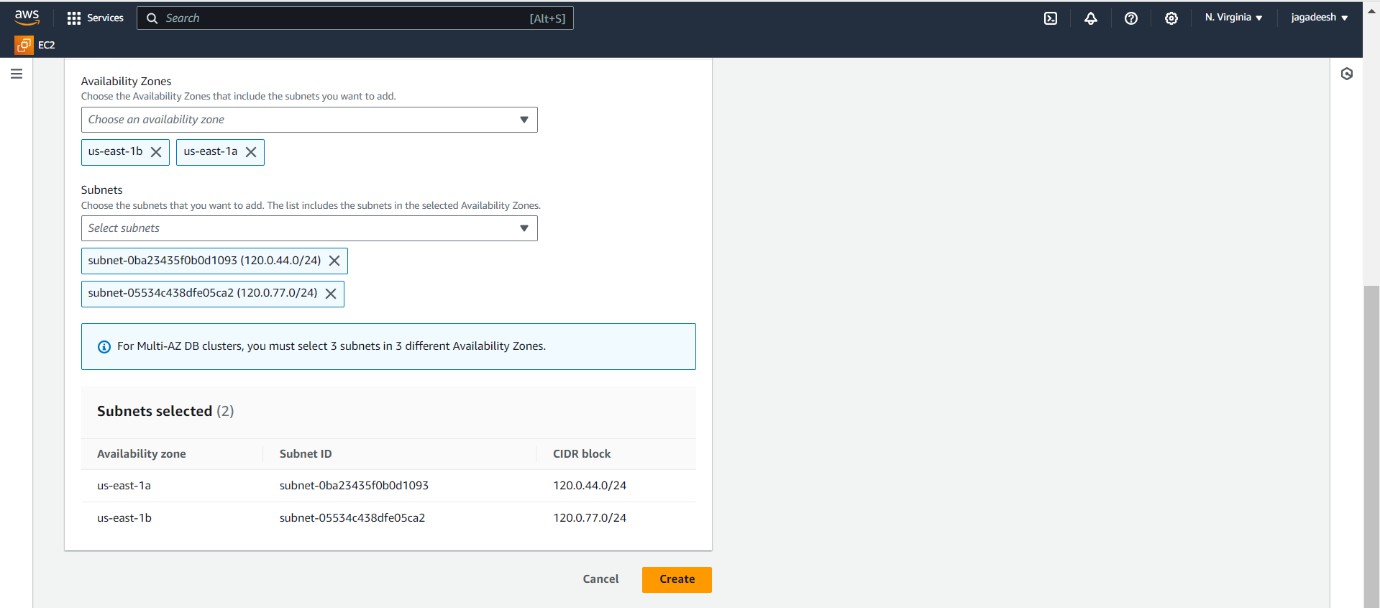
* After creating autoscaling group we can get the four extra servers from
* both public and private auto scaling group.

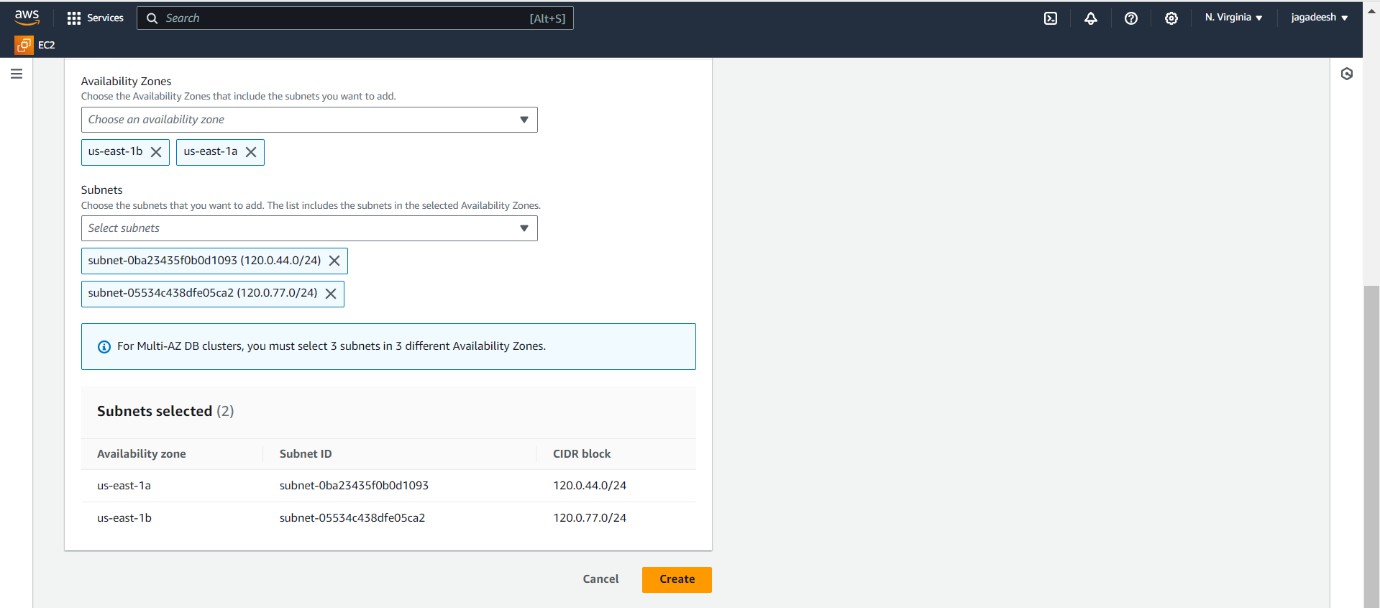


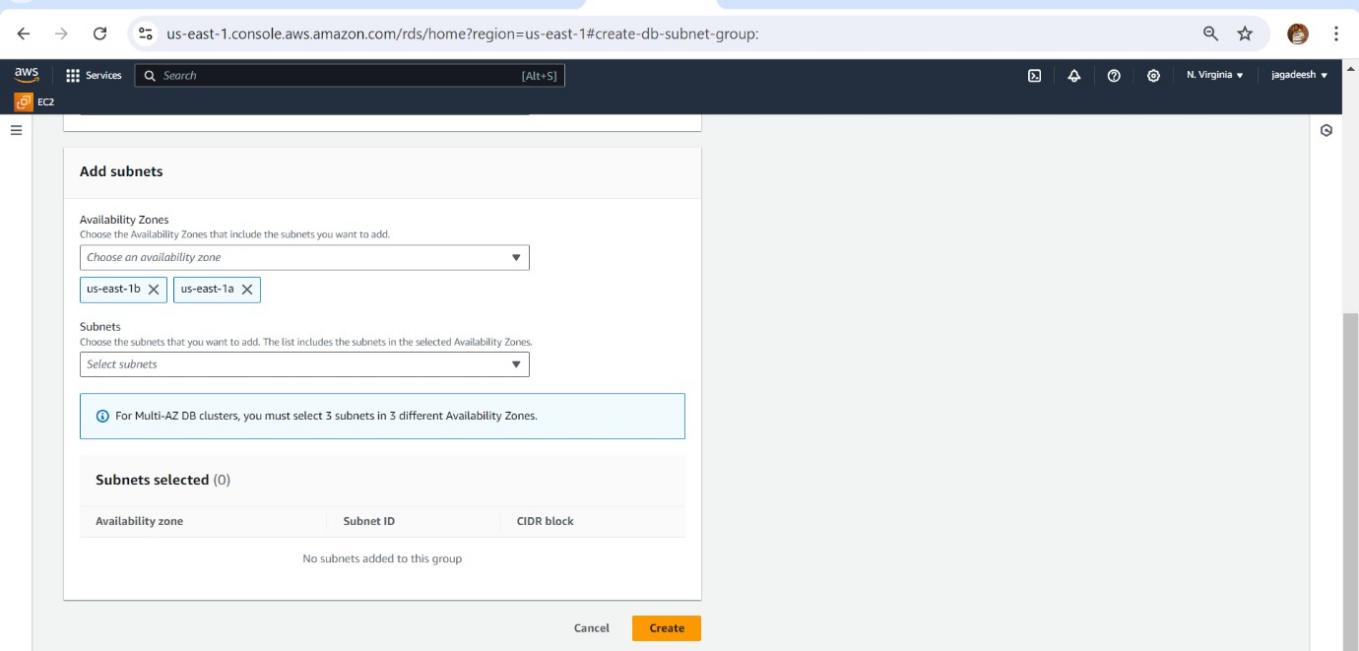
**Step 6:** Create subnet group

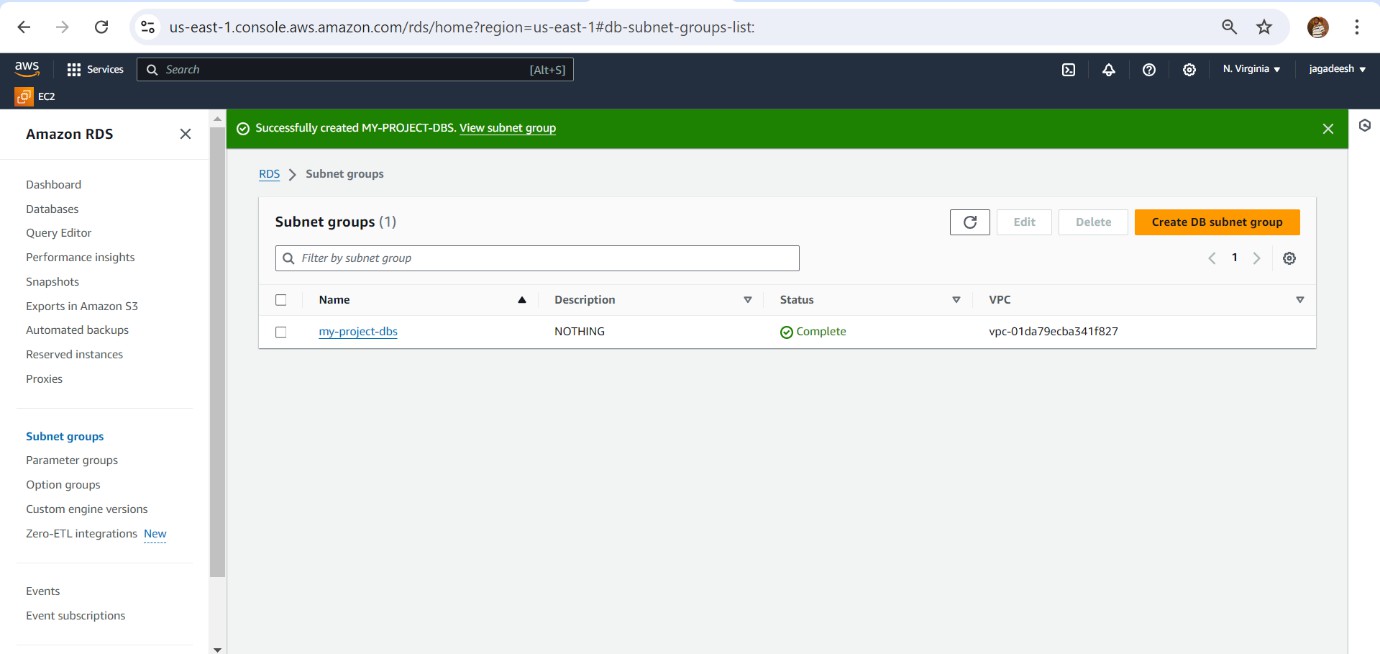
* Give name as my-subnet-grp and description nothing.
* Select created VPC.
* Give availability zones and select private subnets from each zone.
* Create the DB subnet group.





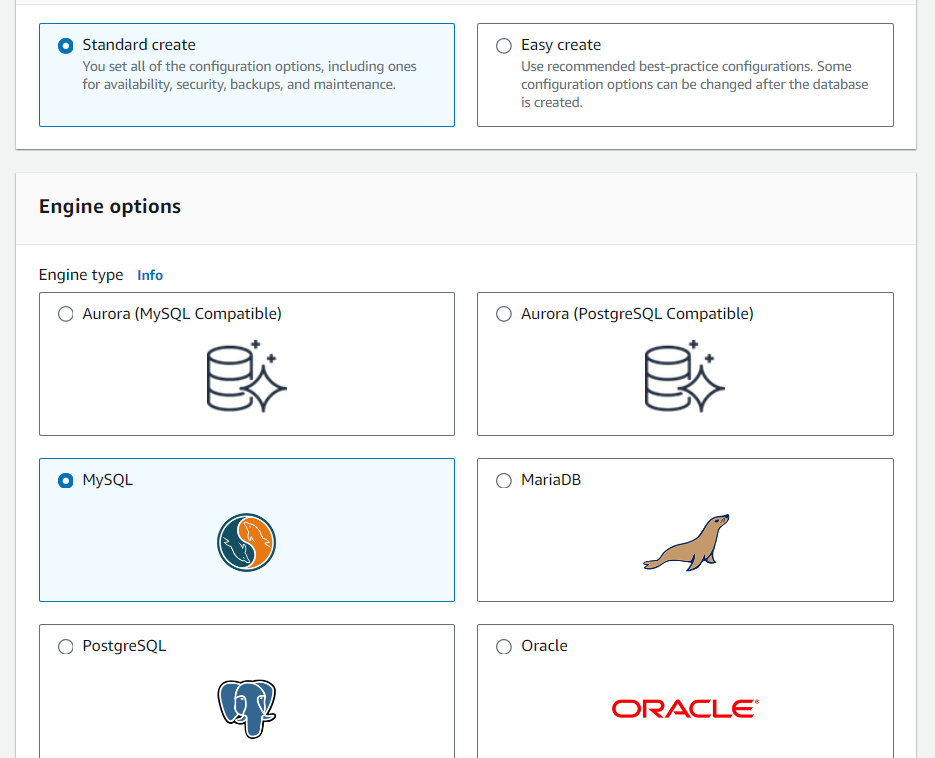


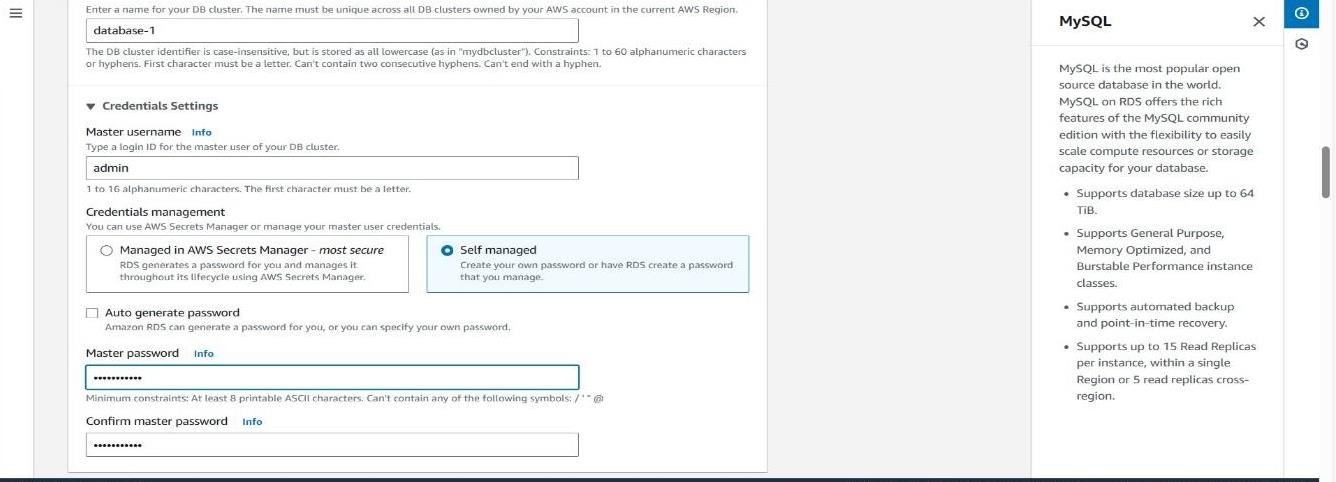


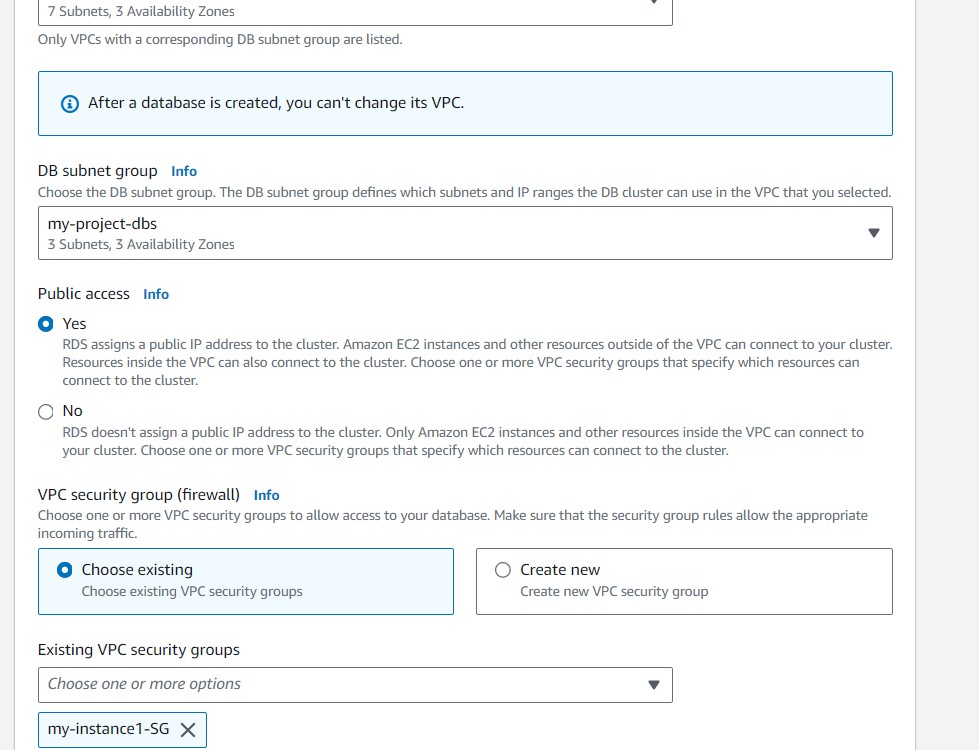


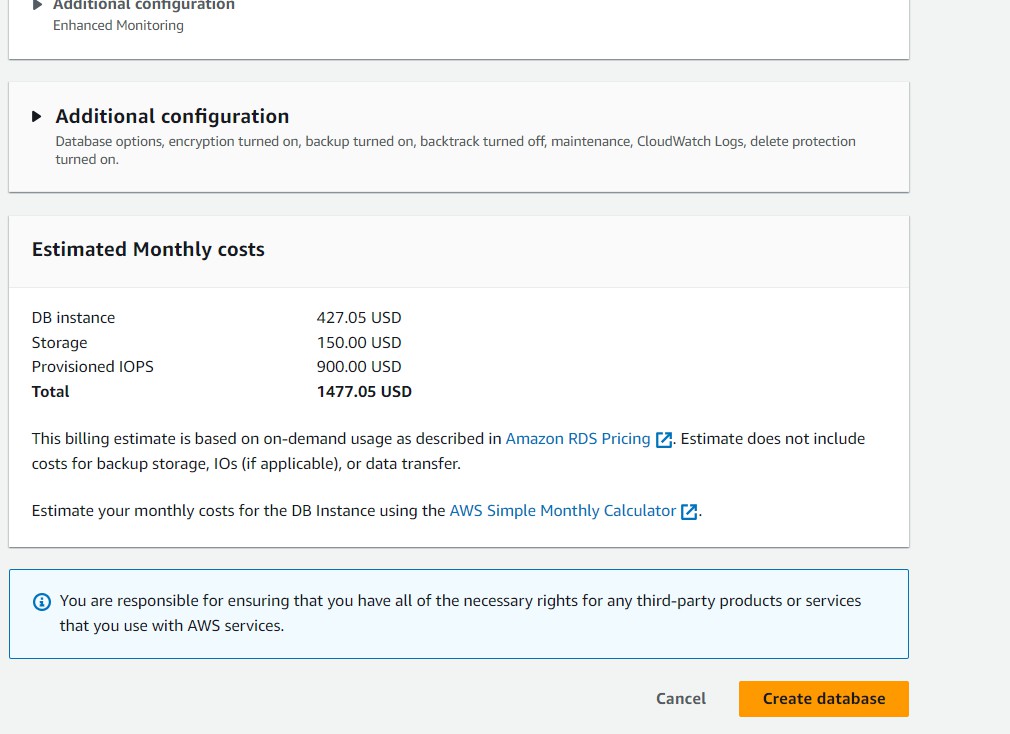
**Step 7:** Create Database(RDS)

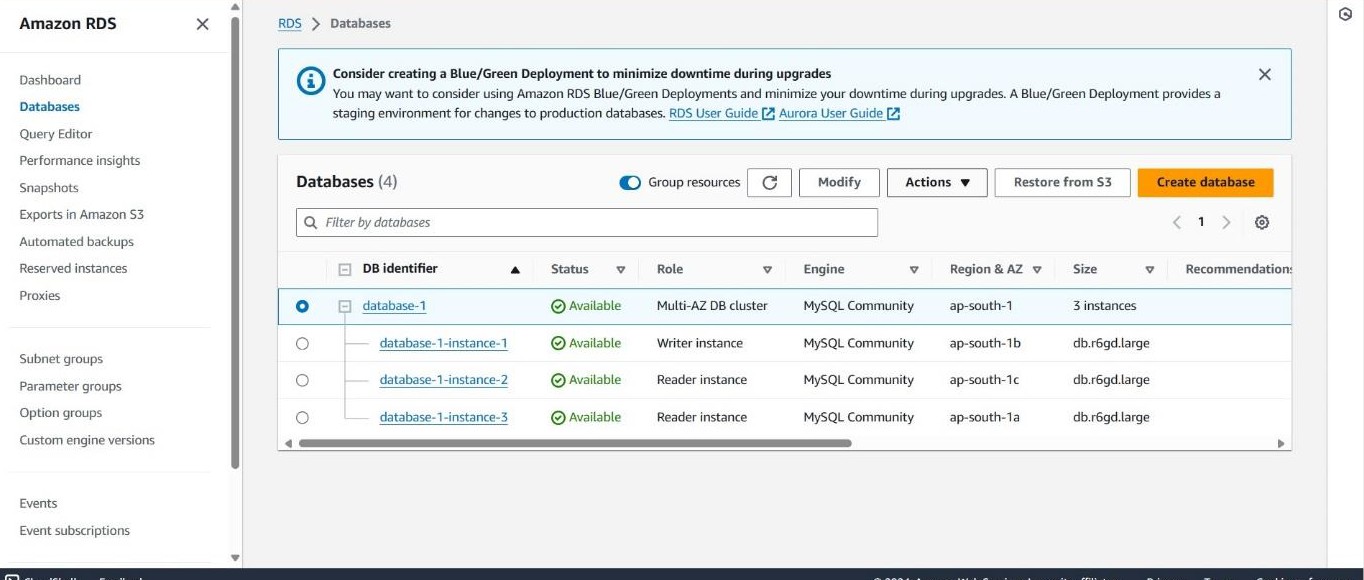
* Click on create database, select standard create, select engine type as MySQL.
* Select templates as production and select multi-AZ DB cluster.
* Select on self-managed, give password and confirm the password.
* Select memory optimized class.
* In connectivity, click on Don’t connect to the EC2 compute resource and select created vpc (my-vpc-project1).
* Select subnet group (my-project-dbs) and give public access as yes.
* Choose existing security group (instance1-SG).
* Go to VPC dashboard, click on VPC, click on actions, go to edit VPC settings and click on the enable DNS hostnames.
* Create the database.

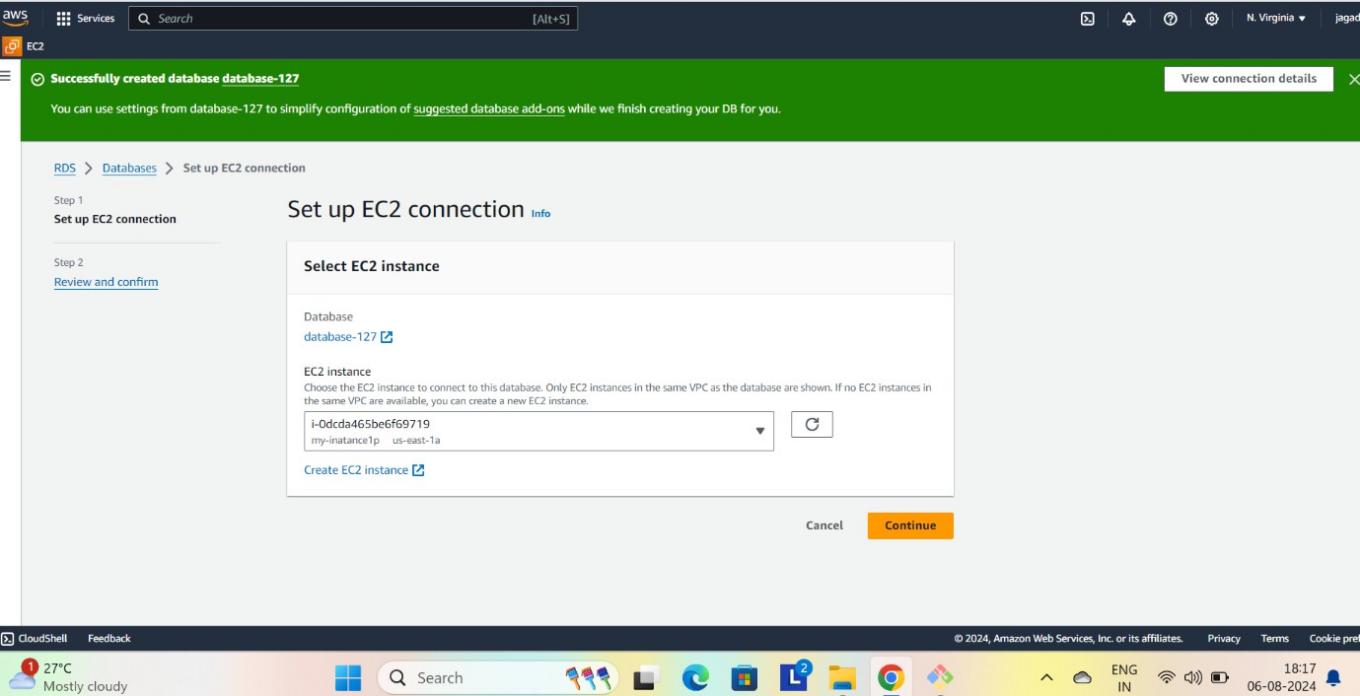










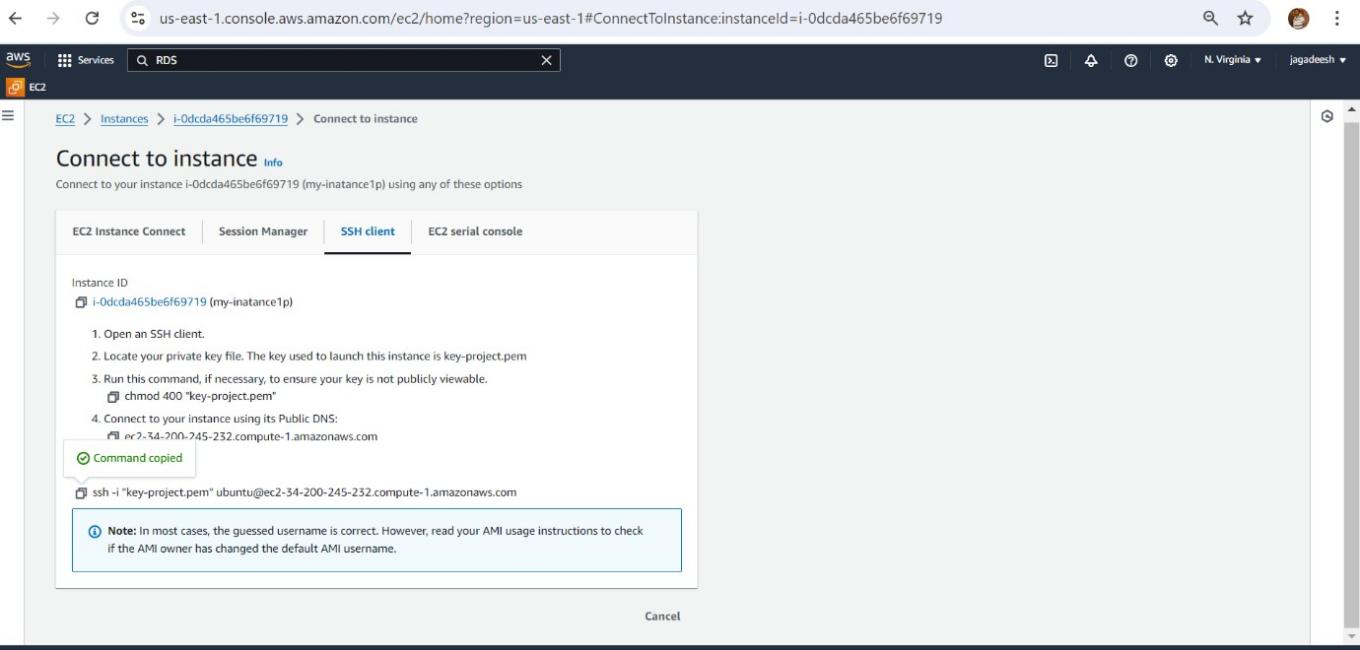


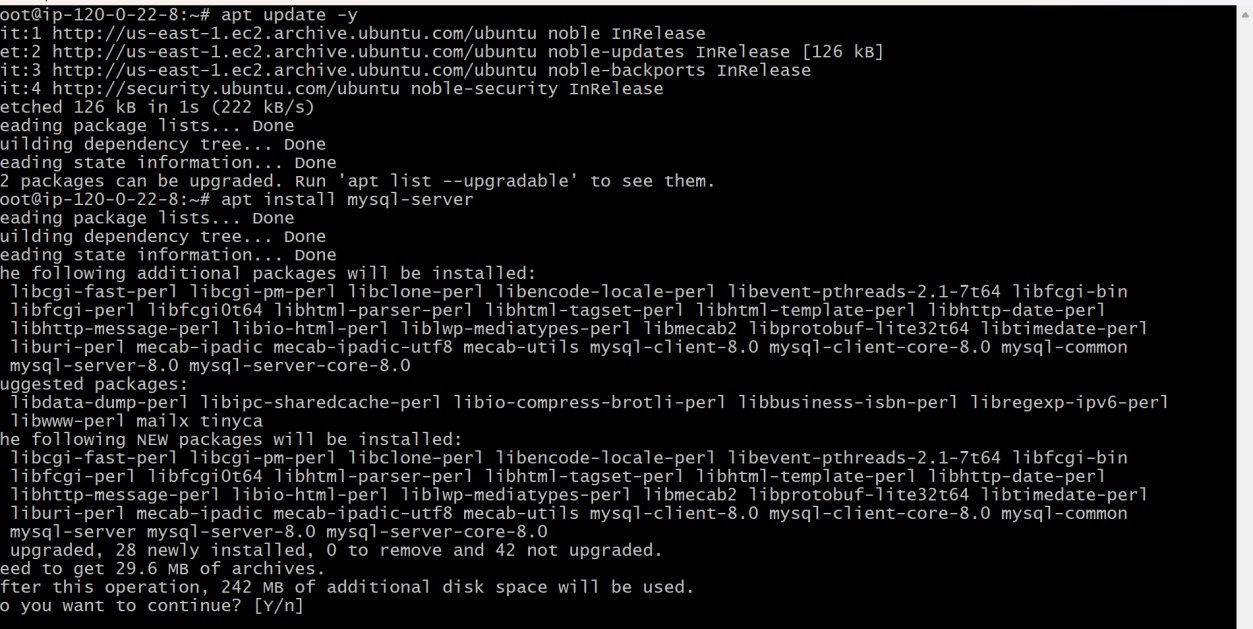
**Step8:** Establish connection

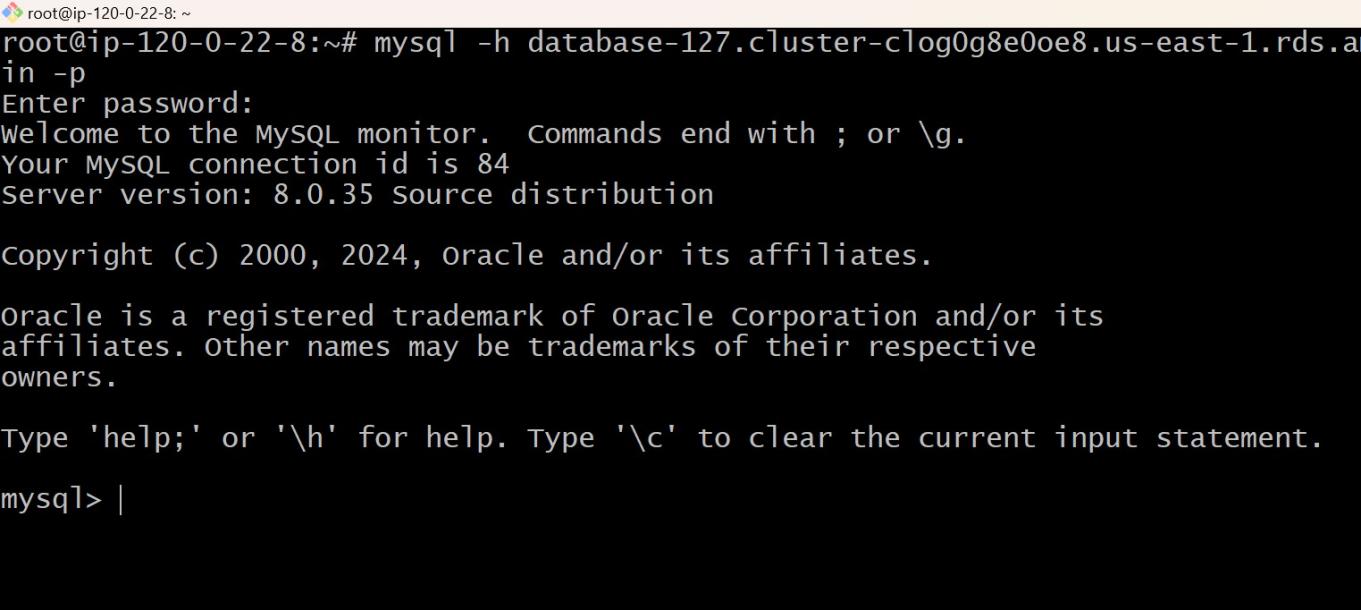
* Go to EC2 instance, click on instance1-pub, open it click on connect and again click on connect.

Give commands as:

1. sudo -i (convert from normal user to root user).
2. apt update -y
3. sudo apt install mysql-server (to install mysql)







Use commands like:

CREATE TABLE Persons (

ID int NOT NULL

LastName varchar(255) NOT NULL, FirstName varchar(255),

Age int,

PRIMARY KEY (ID)

);

* we can insert data into that table using this command
* INSERT INTO *table\_name* (*column1*, *column2*, *column3*, ...) VALUES (*value1*, *value2*, *value3*, ...);
* show tables; (to show tables in that DB)
* Select \* from Table name

