NAME: GARIKI PRATHIBHA

MAIL.ID: [prathibhagariki437@gmail.com](mailto:prathibhagariki437@gmail.com)

# **3-Tier Architecture for Web Applications in AWS**

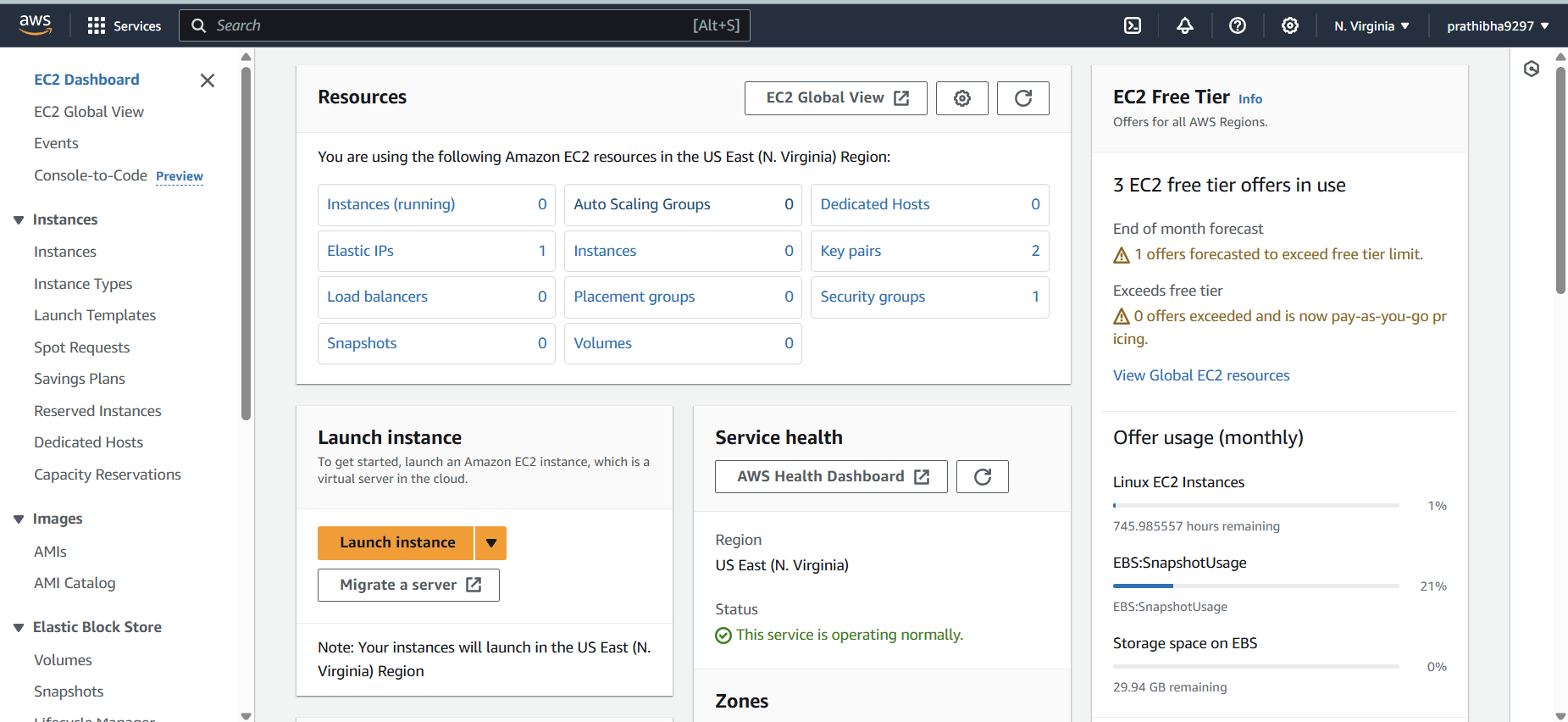
**Fig: Creating a Highly Available 3-Tier Architecture for Web Applications in AWS**

**ABOUT:**

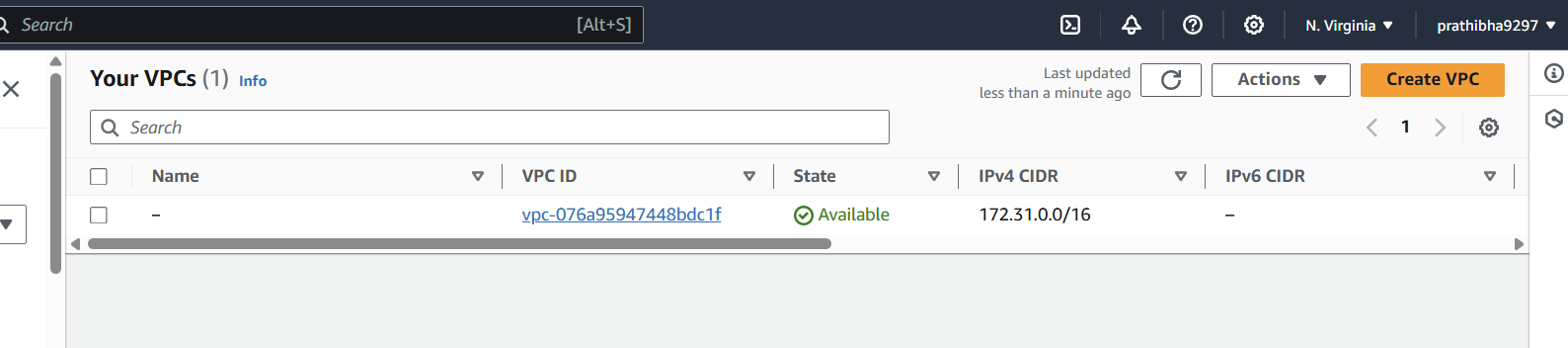
In a three-tier application architecture, each tier serves a specific role and communicates with the other tiers to deliver a complete application experience. Implementing this architecture on AWS involves using various services to handle the presentation (web), application (logic), and data (database) layers.

**VPC:**

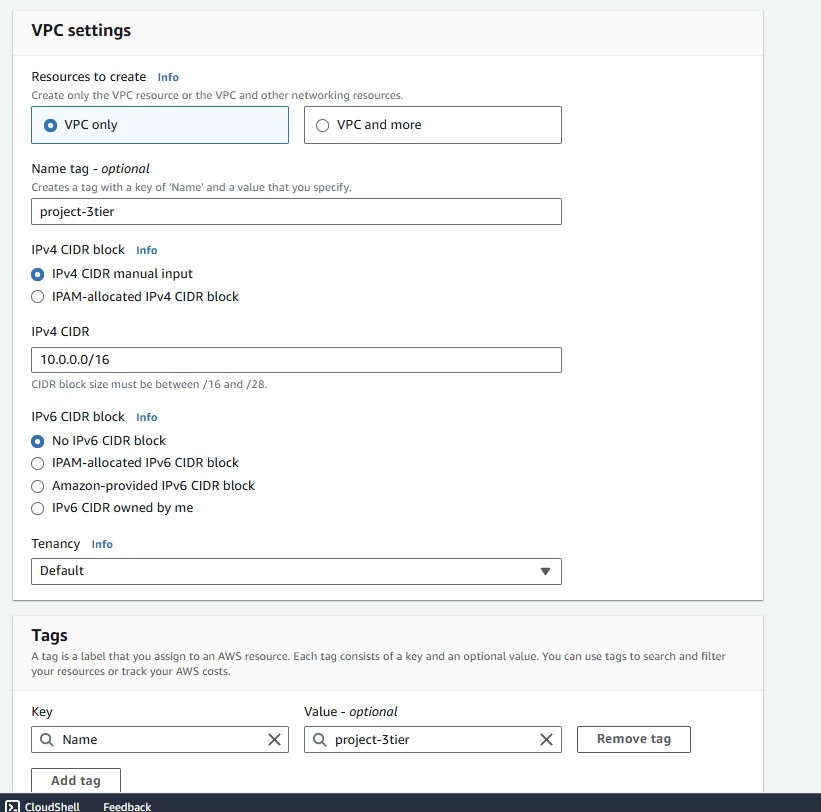
A Virtual Private Cloud (VPC) is a fundamental component in Amazon Web Services (AWS) that enables you to create a logically isolated network environment within the AWS cloud. It provides control over a virtualized network that mimics a traditional on-premises network, including IP address ranges, subnets, route tables, and network gateways. With VPC, you can securely deploy and manage your AWS resources, such as EC2 instances, RDS databases, and other services, in a private and controlled network space.

Go to AWS console, in that I was used **us-east-1a** and **us-east-1b** availability zones in **US East** **N.virginia** region 

Click on VPC in AWS console and then click on create VPC.



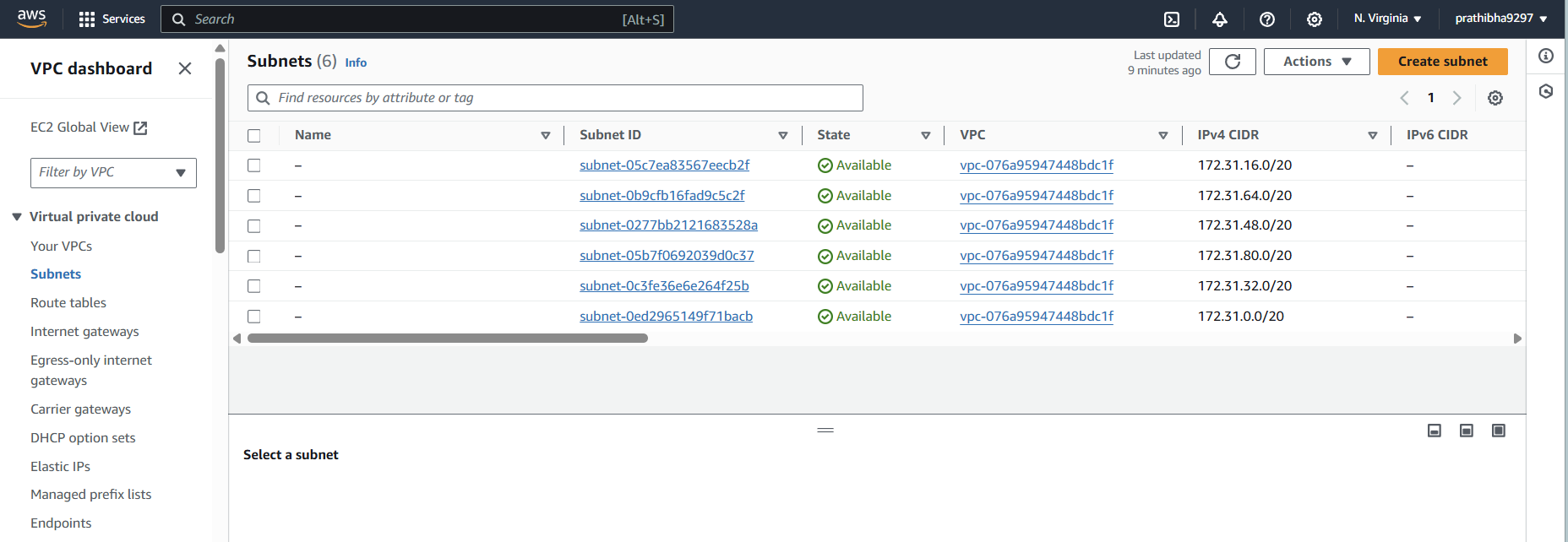
* Go to VPC dashboard click on create VPC.
* Click on VPC only and name tag as project-3tier.
* Give IPV4 CIDR (classless inter domain routing) as 10.0.0.0/16.
* Click on VPC, it is created.



**Subnets:**

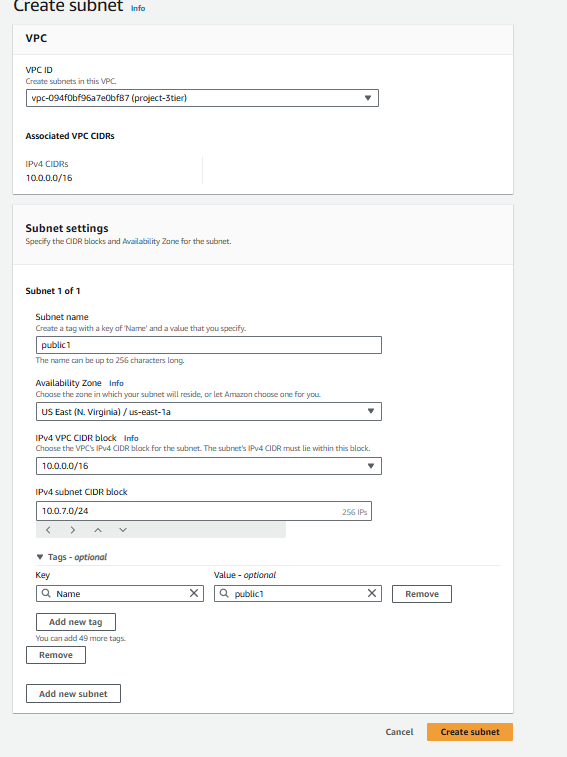
* **Public Subnets:** Subnets that have access to the internet via an Internet Gateway. Typically used for resources that need to be reachable from the outside, such as web servers.
* **Private Subnets:** Subnets that do not have direct internet access. Used for resources that should not be exposed to the internet, such as databases and application servers.

For this 3-tier architecture 6 subnets need to create, In that 2 subnets are public and remaining 4 subnets are private.



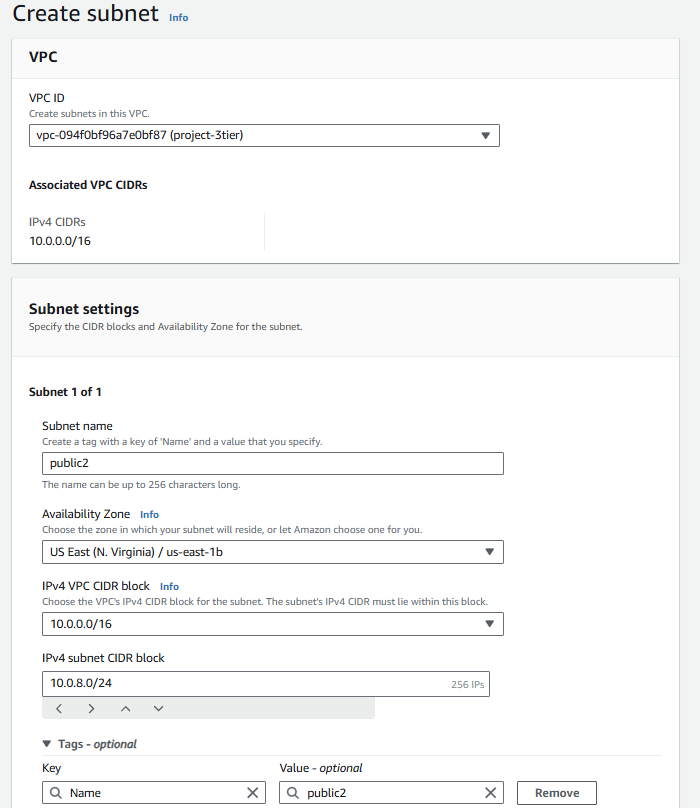
Subnet 1:

* Click on create subnet
* Select the VPC (project-3tier)
* Give the name tag as public1->select availability zone as US East(N.virginia)/us-east-1a
* Give CIDR as 10.0.7.0/24 and then click on create subnet



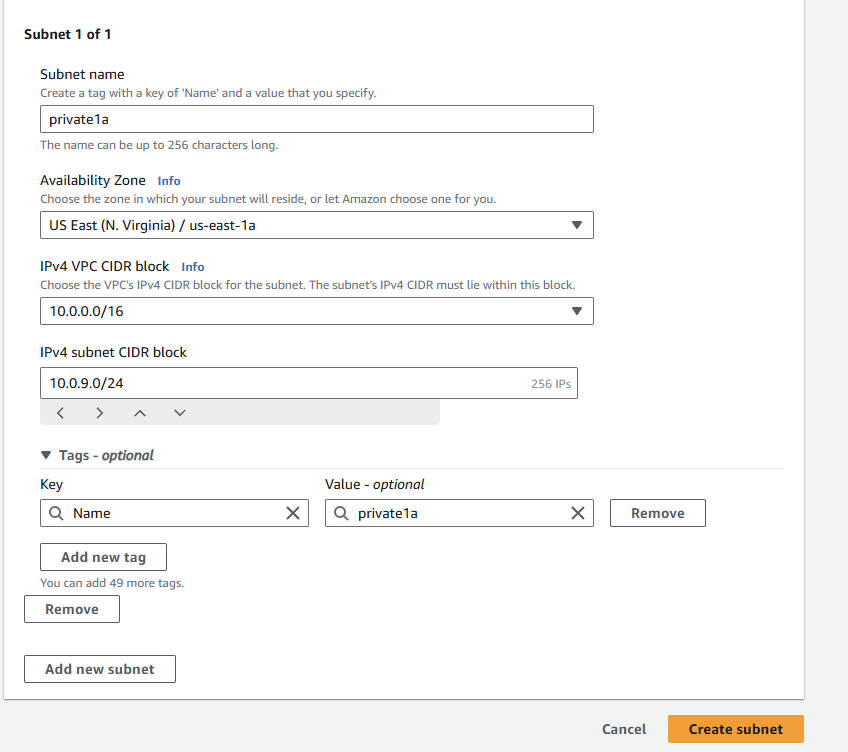
Subnet 2:

* Click on create subnet
* Select the VPC (project-3tier)
* Give the name tag as public2->select availability zone as US East(N.virginia)/us-east-1b
* Give CIDR as 10.0.8.0/24 and then click on create subnet.



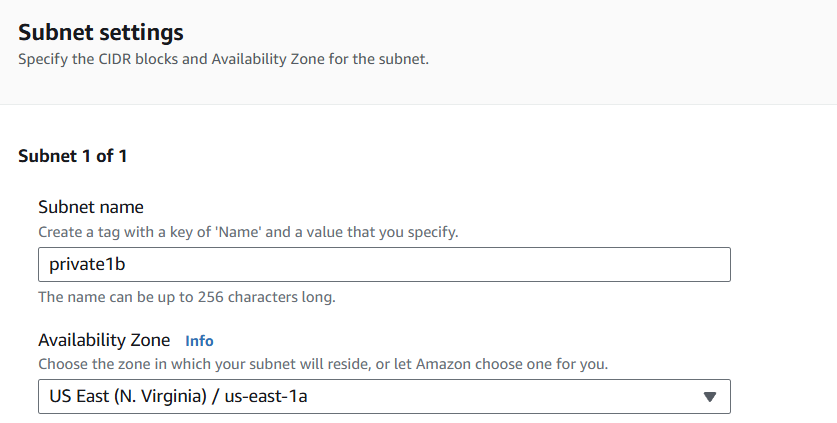
Subnet 3:

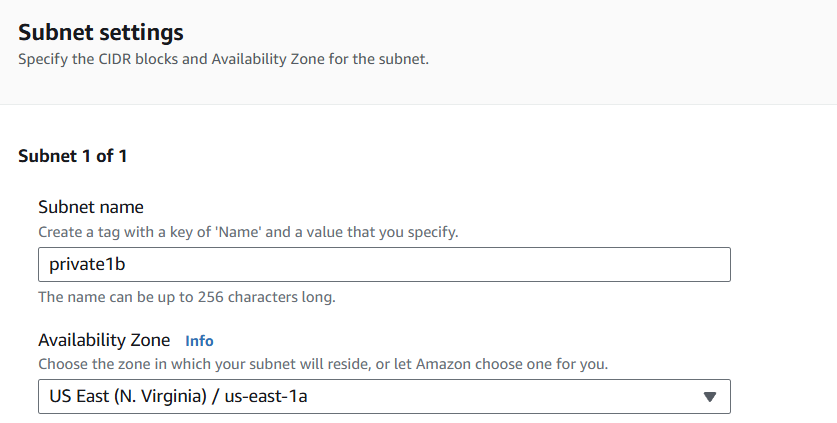
* Click on create subnet
* Select the VPC (project-3tier)
* Give the name tag as private1a->select availability zone as US East(N.virginia)/us-east-1a
* Give CIDR as 10.0.9.0/24 and then click on create subnet.



Subnet 4:

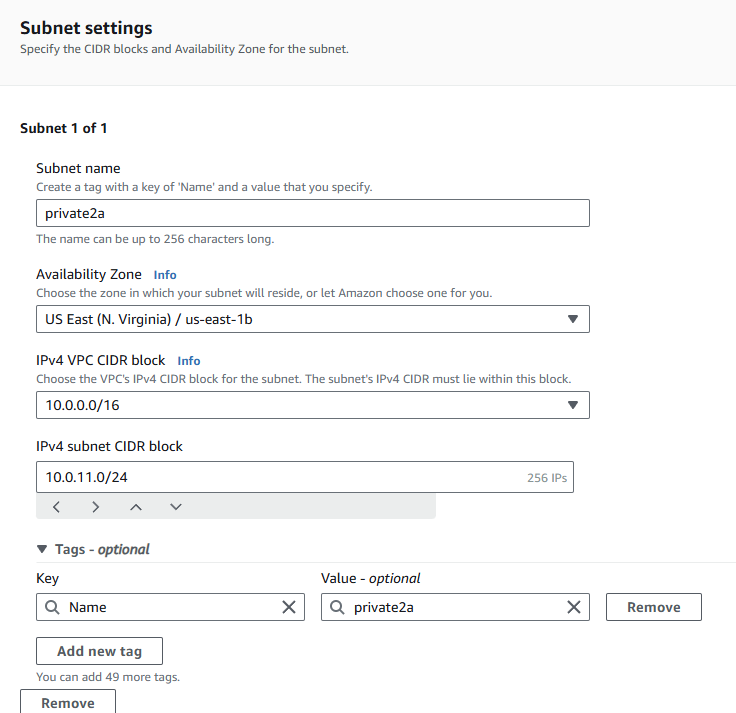
* Click on create subnet
* Select the VPC (project-3tier)
* Give the name tag as private1b->select availability zone as US East(N.virginia)/us-east-1a
* Give CIDR as 10.0.10.0/24 and then click on create subnet.





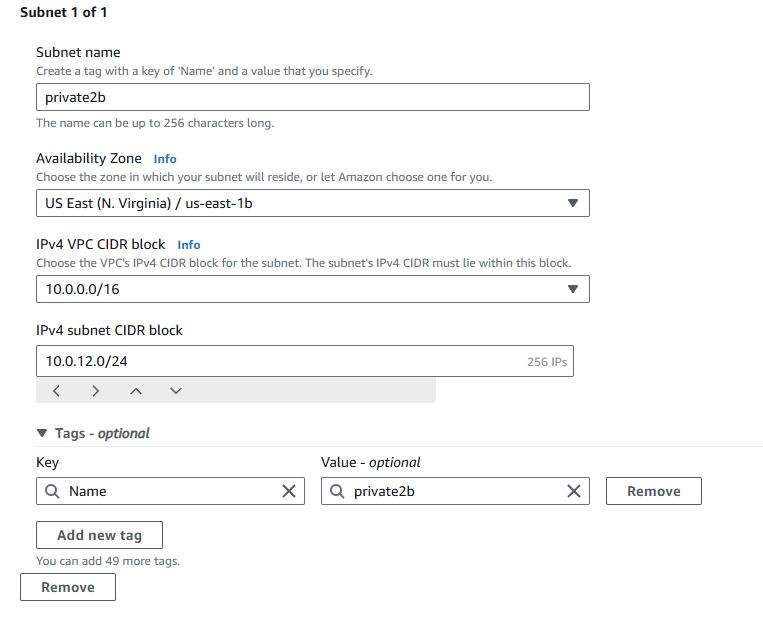
Subnet 5:

* Click on create subnet
* Select the VPC (project-3tier)
* Give the name tag as private2a->select availability zone as US East(N.virginia)/us-east-1b
* Give CIDR as 10.0.11.0/24 and then click on create subnet.

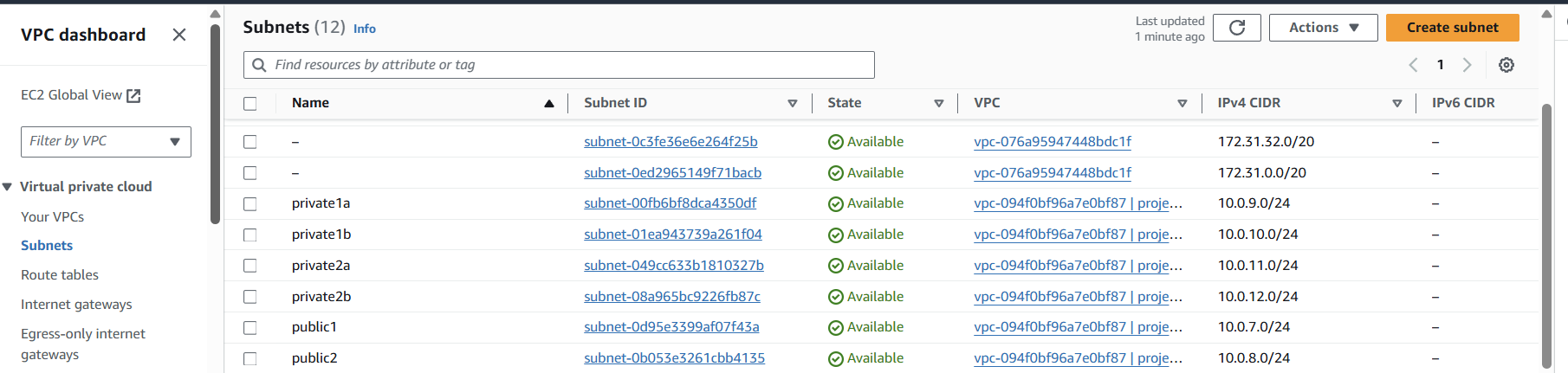


Subnet 6:

* Click on create subnet
* Select the VPC (project-3tier)
* Give the name tag as private2b->select availability zone as US East(N.virginia)/us-east-1b
* Give CIDR as 10.0.12.0/24 and then click on create subnet.



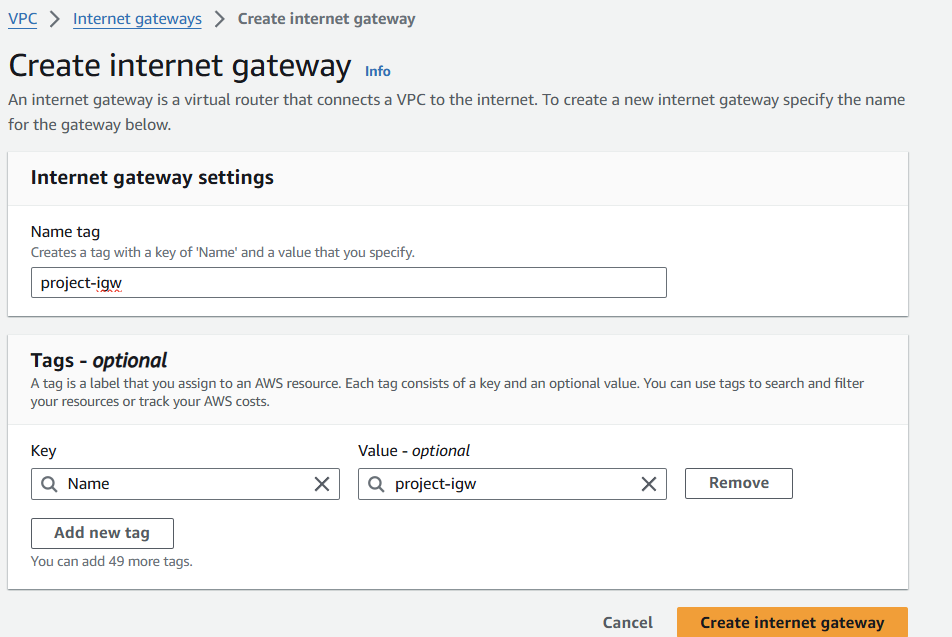
Total 6 subnets we just created as shown in below.

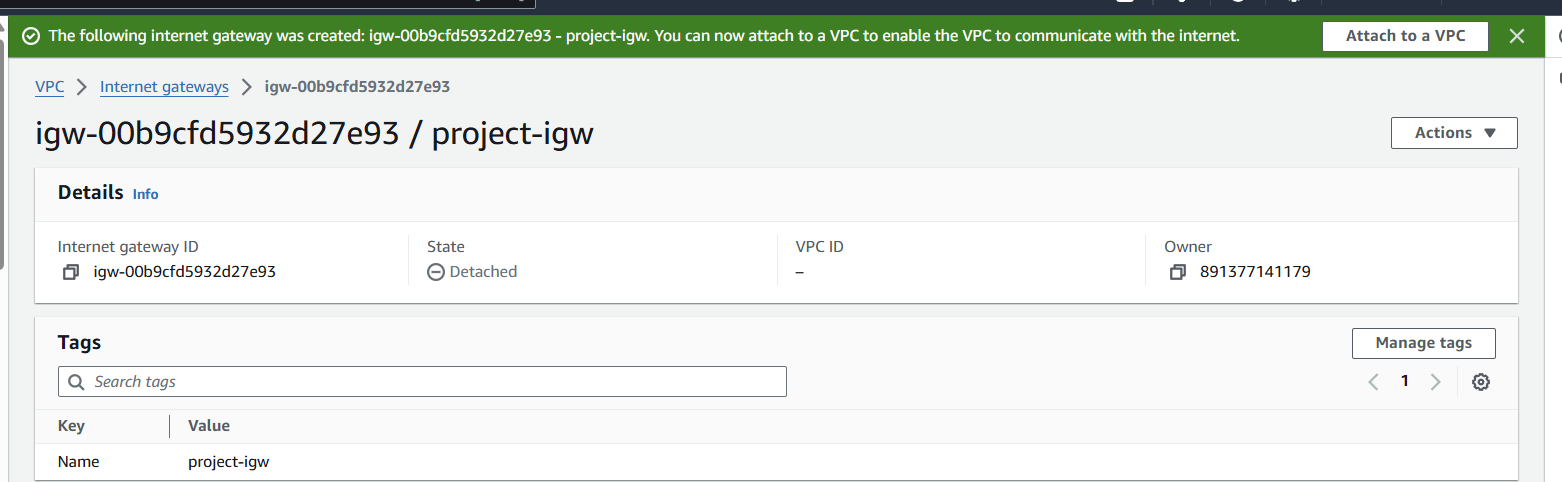


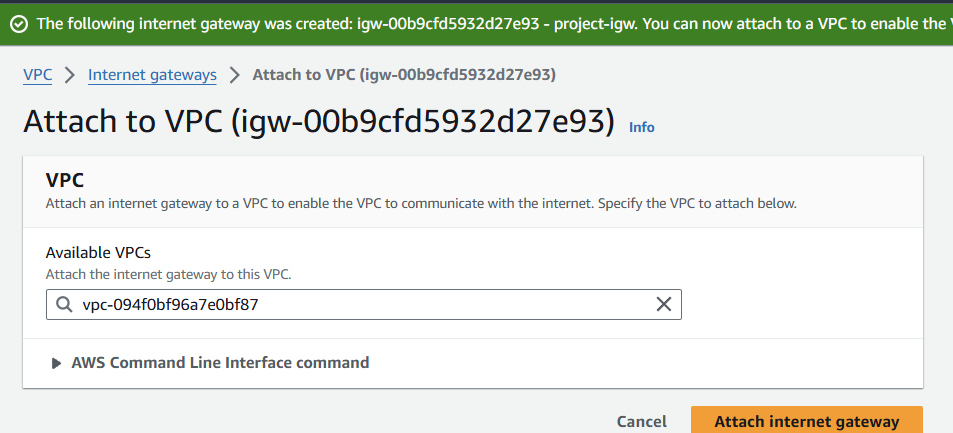
Internet gateway(IGW):

A gateway that provides internet access to resources in your VPC. It is attached to your VPC to enable communication between instances in your VPC and the internet.

* Click on create internet gateway->Give name tag as project-igw-> click on create.
* Once IGW created click on attach to a VPC
* Select the VPC and then click on attach internet gateway.



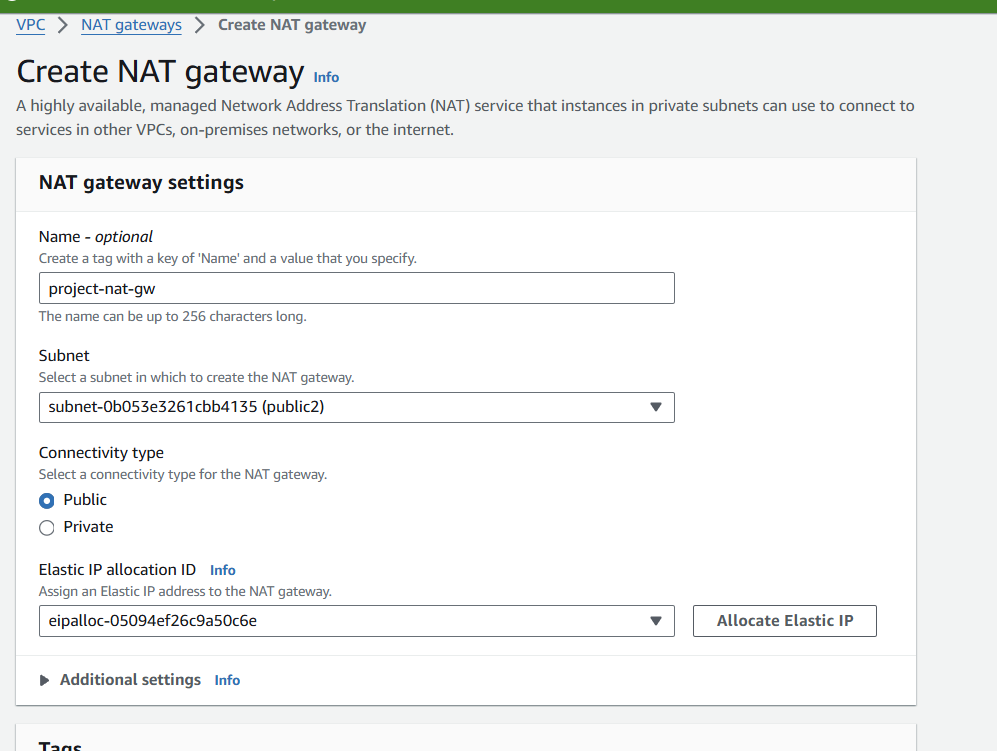




**NAT gateway**:

A managed service that allows instances in a private subnet to initiate outbound traffic to the internet while preventing unsolicited inbound traffic.

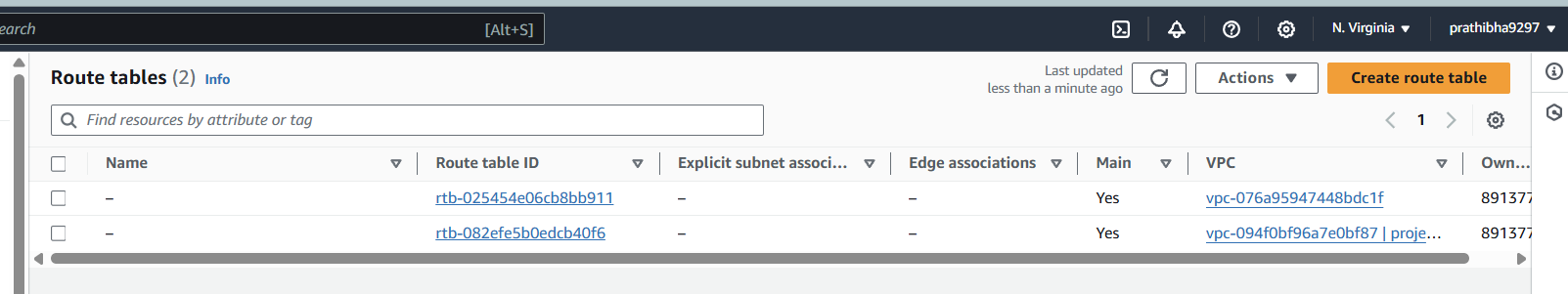
* Click on create NAT gateway
* Give name tag as project-nat-gw->select the public subnet(public2)->connectivity type should be public-
* Click on allocate Elastic IP and then click on create Nat gateway



**Route tables:**

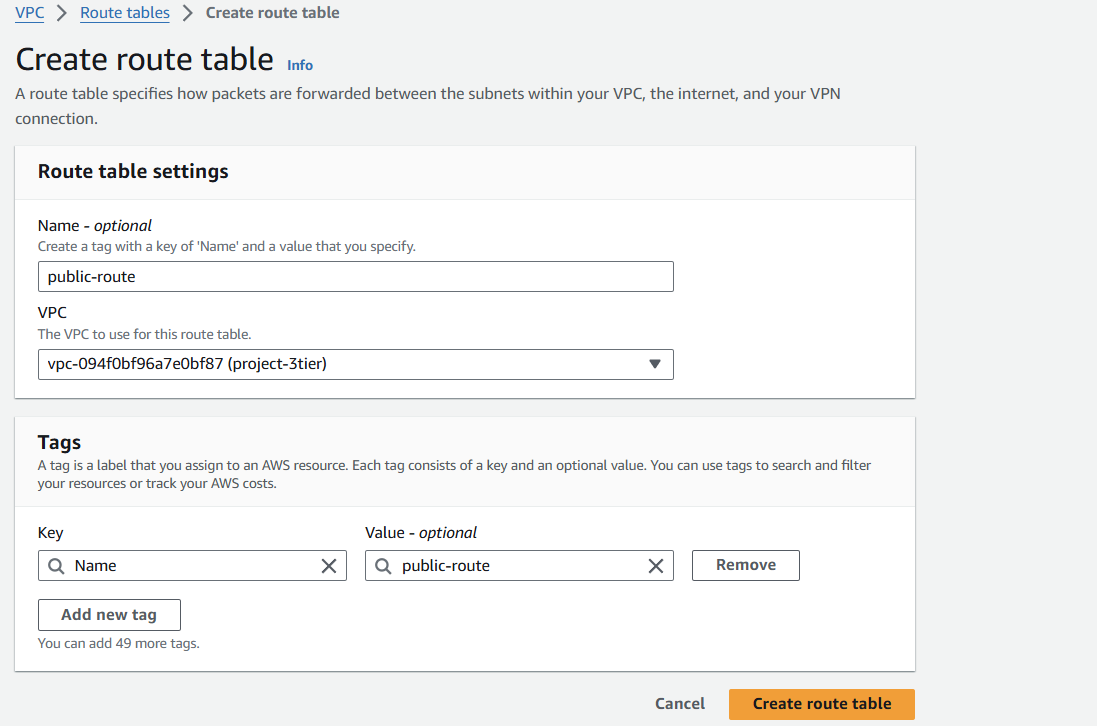
Route tables control the routing of traffic within your VPC. You can define routes for directing traffic between subnets and to/from the internet or other networks.

For 3 tier architecture we need to create 2 route tables. One is for public subnets and another is for private subnets.

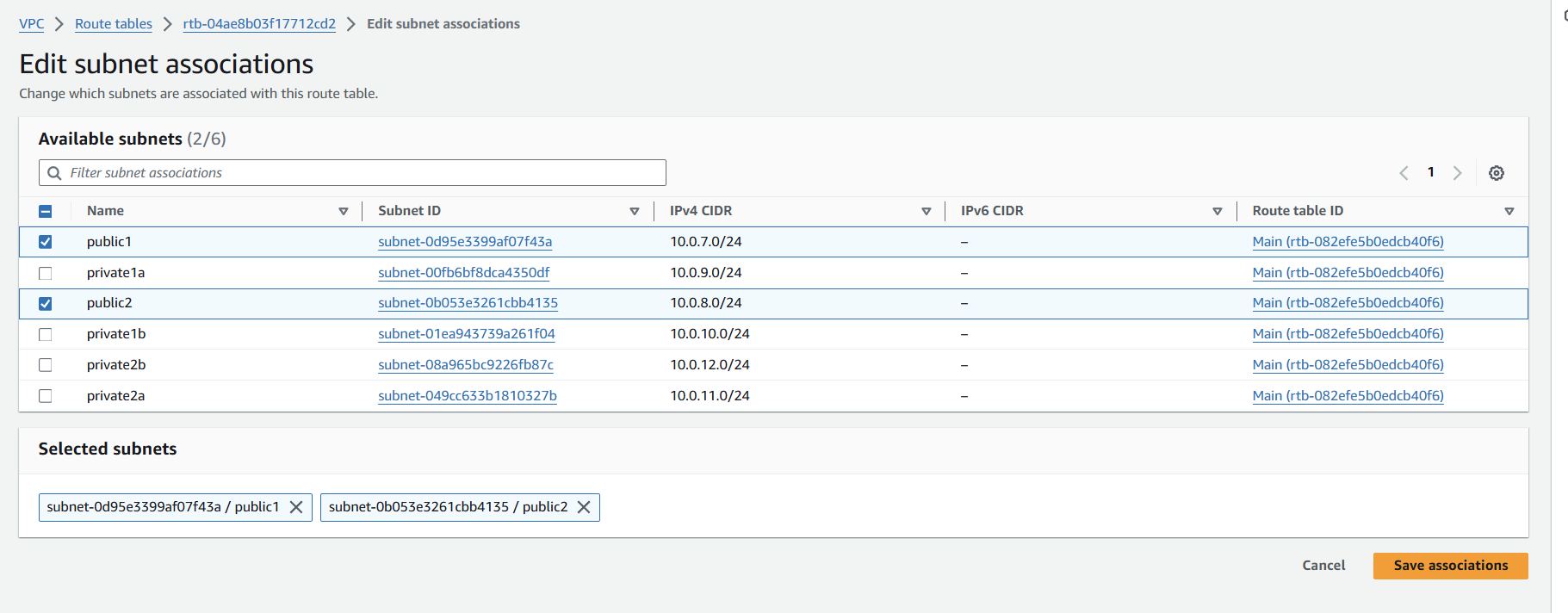


Route table1:

* Go to route tables under VPC field and click on create route table
* Give name tag as public-route->select VPC->click on create
* Once created click on action->Edit route->add IGW in add routes field->save changes.
* Click Edit subnet associations->select 2 public subnets [public1 & public2]->click on save associations.

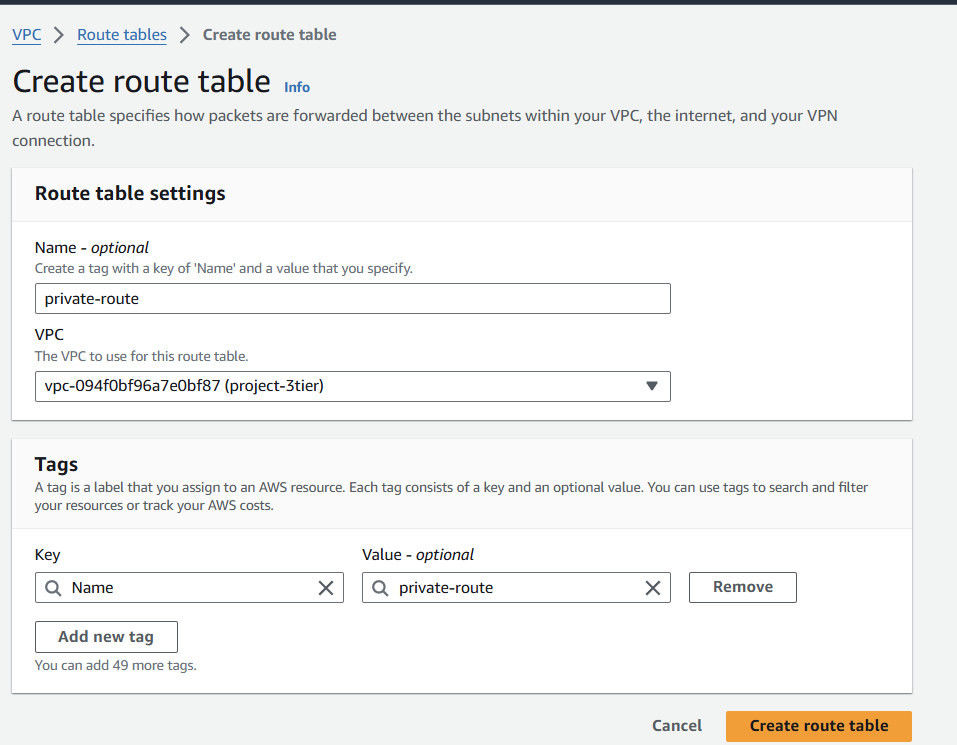




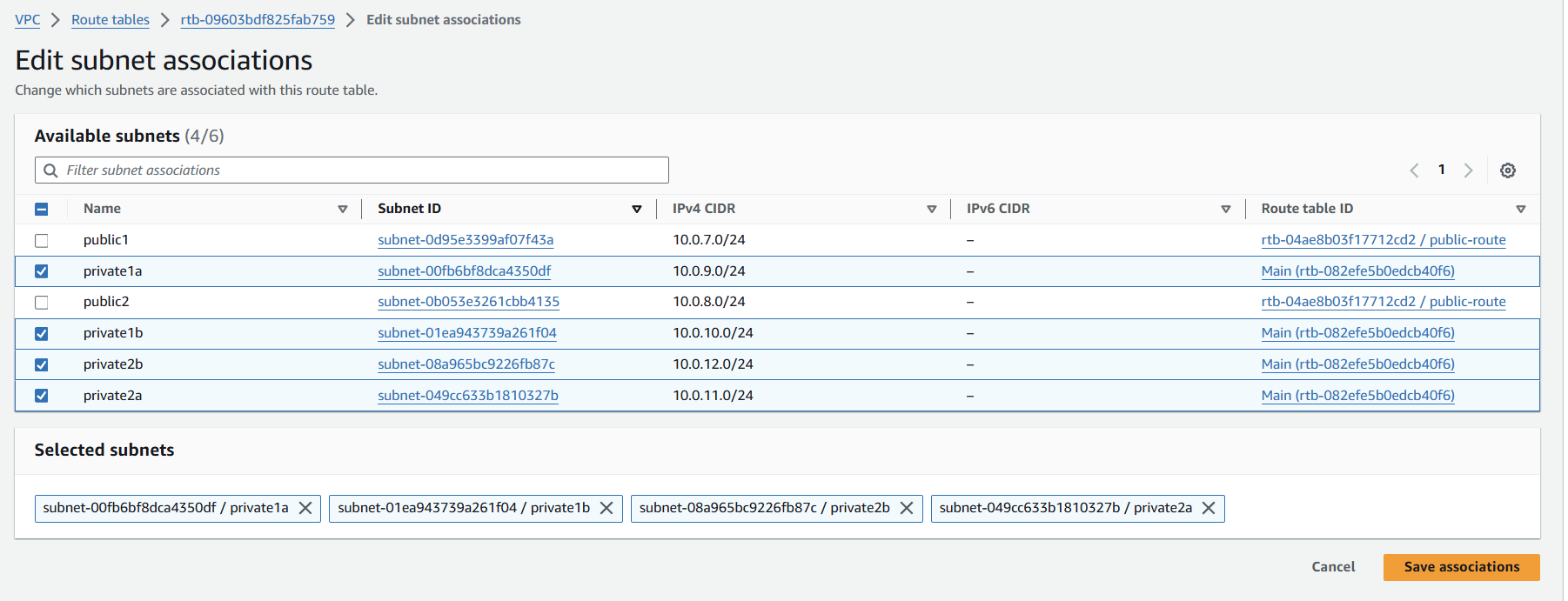


Route table2:

* Click on create route table
* Give name tag as private-route->select VPC->click on create
* Once created click on action->Edit route->add NAT gateway in add routes field->save changes.
* Click Edit subnet associations->select 4 private subnets [private1a,private1b,private2a & private2b]->click on save associations.





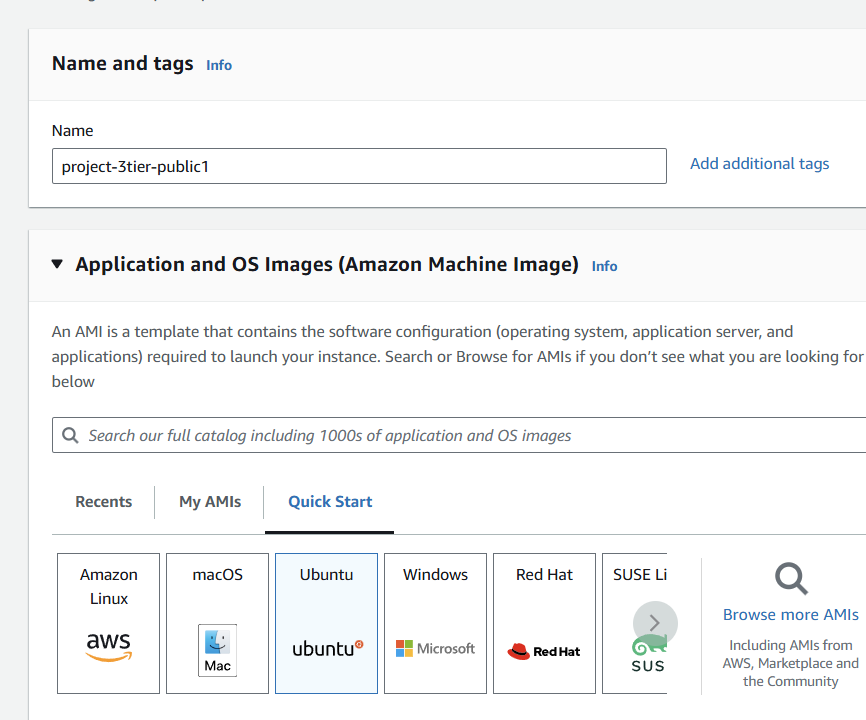


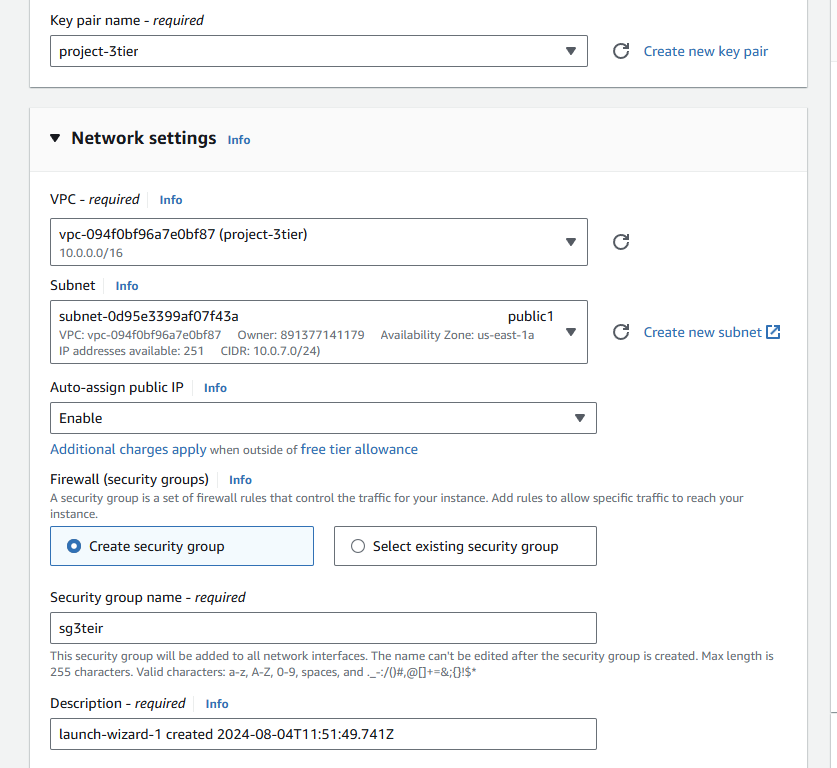
**EC2 instance:**

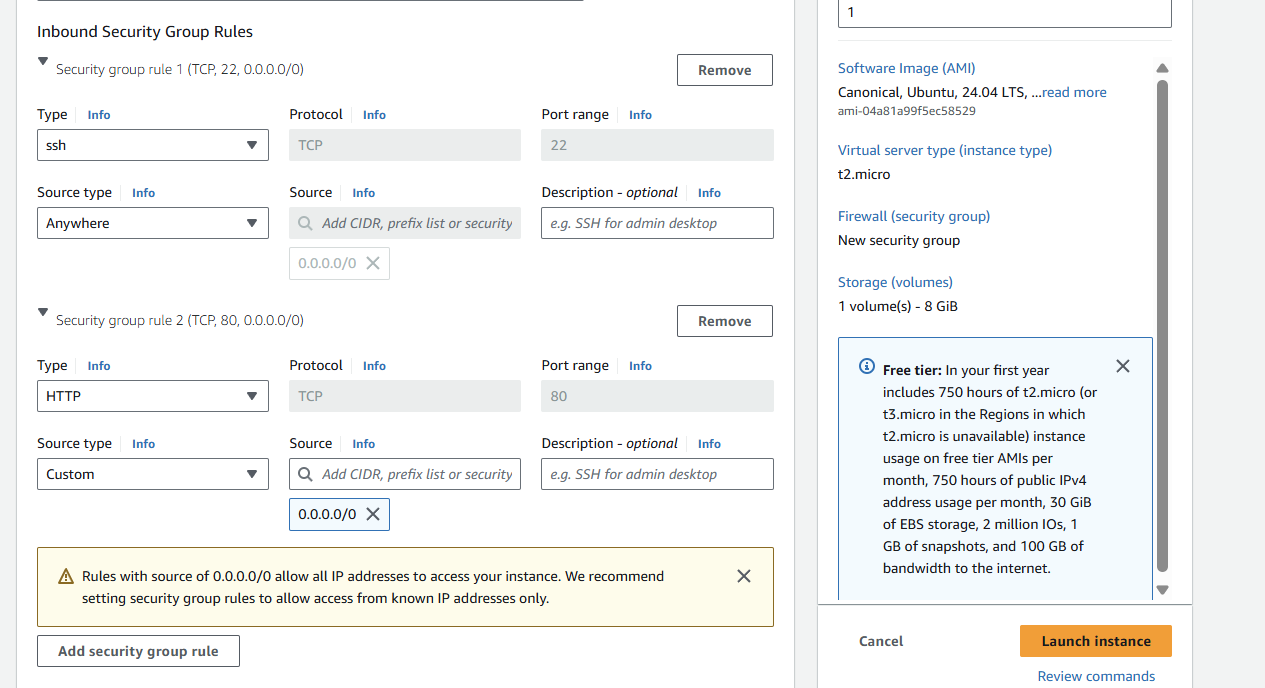
Amazon EC2 (Elastic Compute Cloud) is a core component of Amazon Web Services (AWS) that provides scalable computing capacity in the cloud. It allows you to launch and manage virtual servers, known as instances, which can run various applications and workloads.

Instance1:

* Go to EC2 dashboard and click on launch instance
* Give name tag as project-3tier-public1->select ubuntu -> create key pair and named as project-3tier
* Click on network settings->select VPC as project-3tier->select subnet public1->Enable the Auto assign public IP->create security group name as sg3tier->add http port number 80 for security group-> click on launch instance.
* The overall configuration as shown following slides.

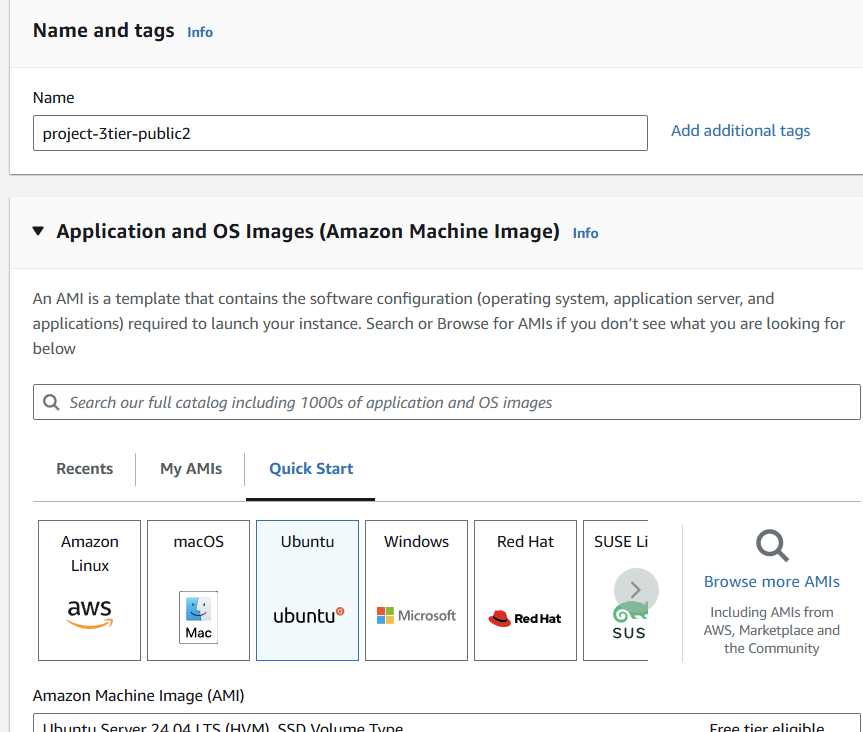


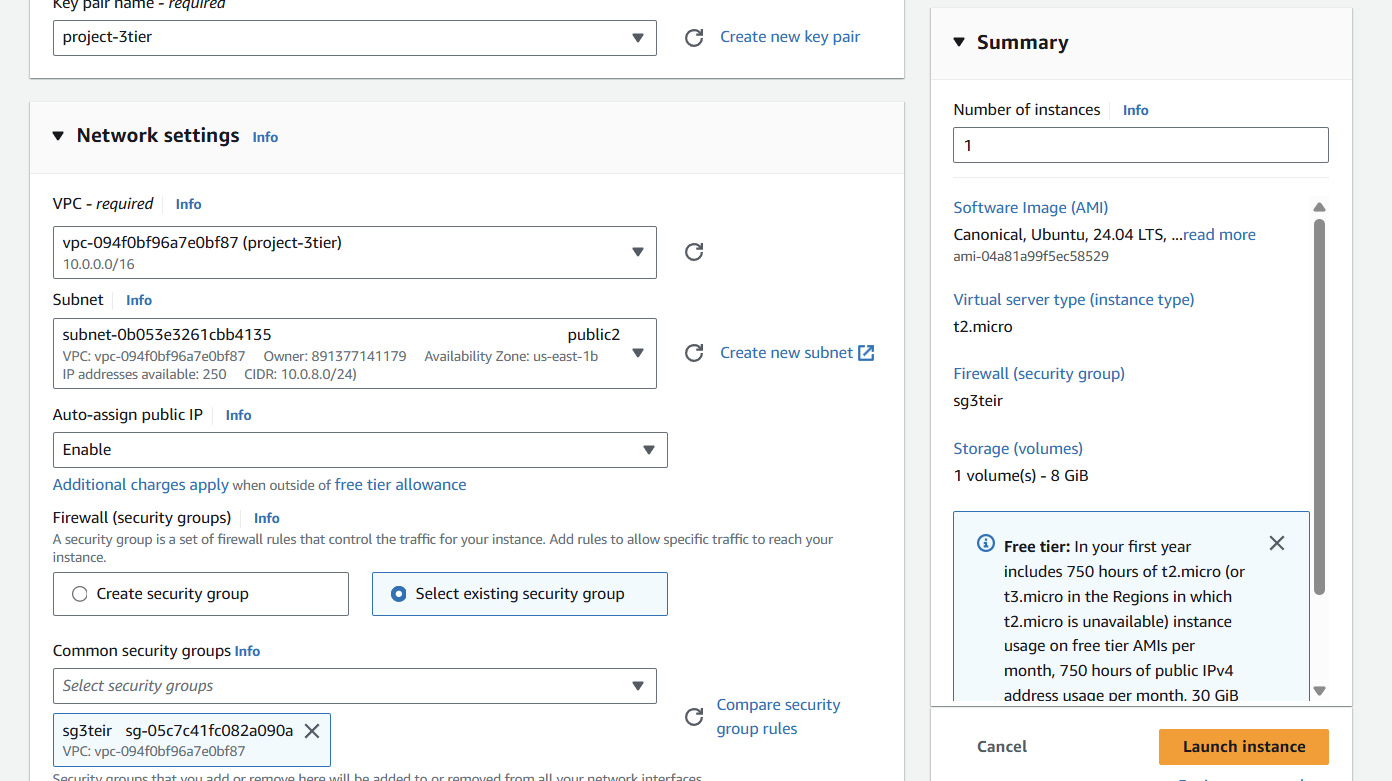


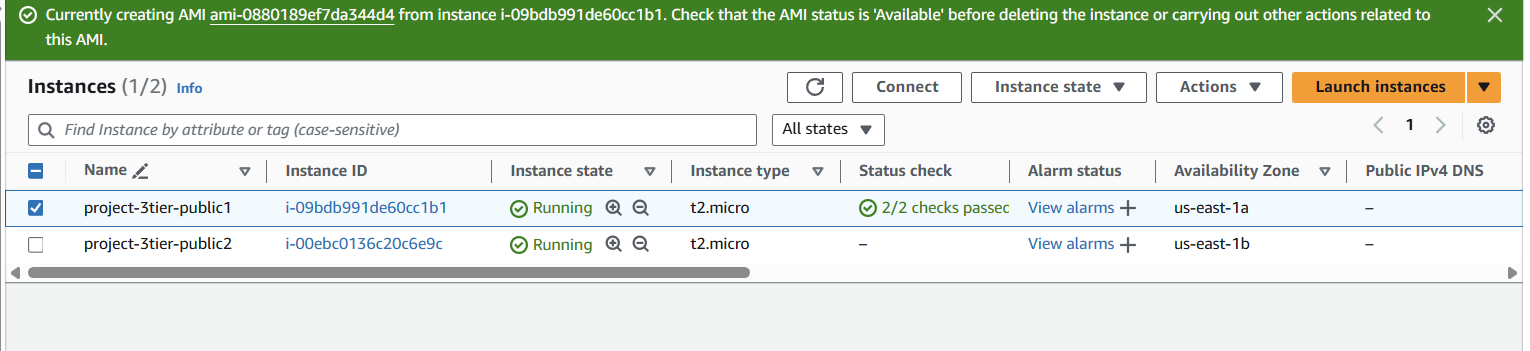


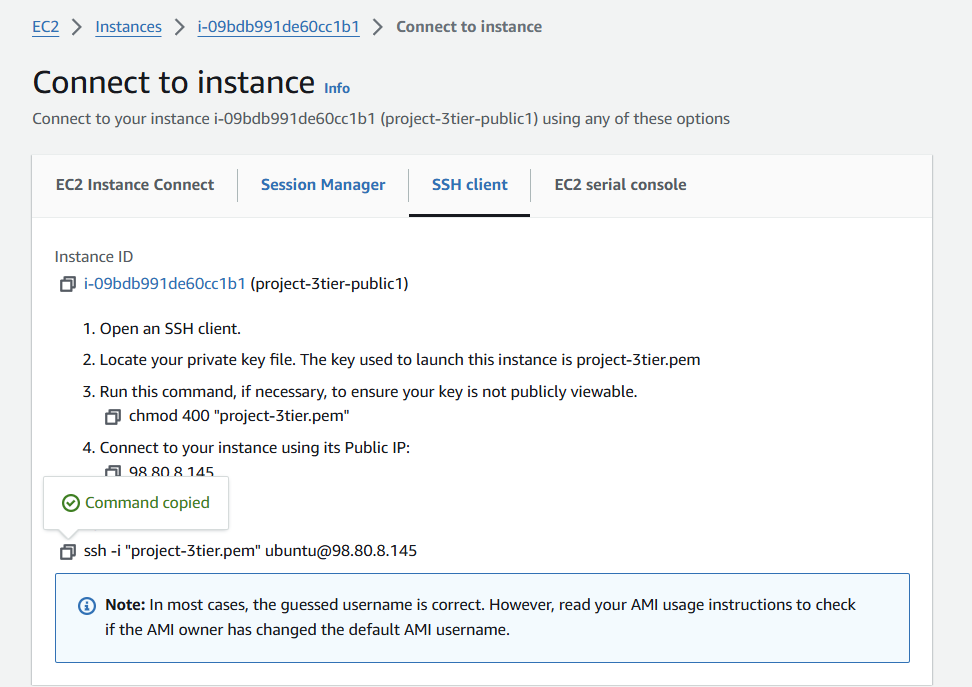
Instance2:

* click on launch instance.
* Give name tag as project-3tier-public2->select ubuntu -> select key pair as project-3tier
* Click on network settings->select VPC as project-3tier->select subnet public2->Enable the Auto assign public IP->select the existing security group-sg3tier-> click on launch instance.
* The overall configuration as shown following slides

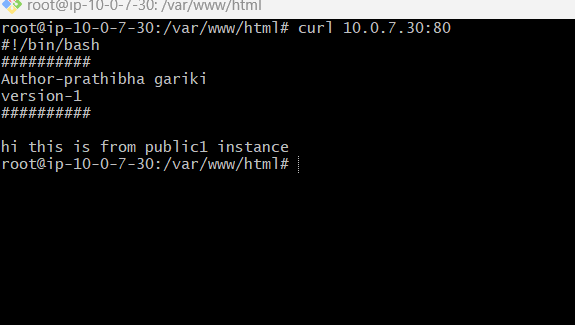




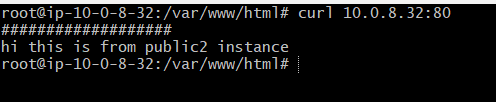




* Once created the instance click on connect and copy the ssh command in SSH client field and then go to git bash and connect to the server.
* Once connected change to root user->sudo -i
* Then update the server->apt update -y
* Install apache2->apt install apache2 -y
* Change directory path->cd /var/www/html
* Remove the default index.html and create new file.
* Restart->systemctl restart apache2
* Output shown in below slide.



* Once created the instance2 click on connect and copy the ssh command in SSH client field and then go to git bash and connect to the server.
* Once connected change to root user->sudo -i
* Then update the server->apt update -y
* Install apache2->apt install apache2 -y
* Change directory path->cd /var/www/html
* Remove the default index.html and create new file.
* Restart->systemctl restart apache2
* Output shown in below slide.



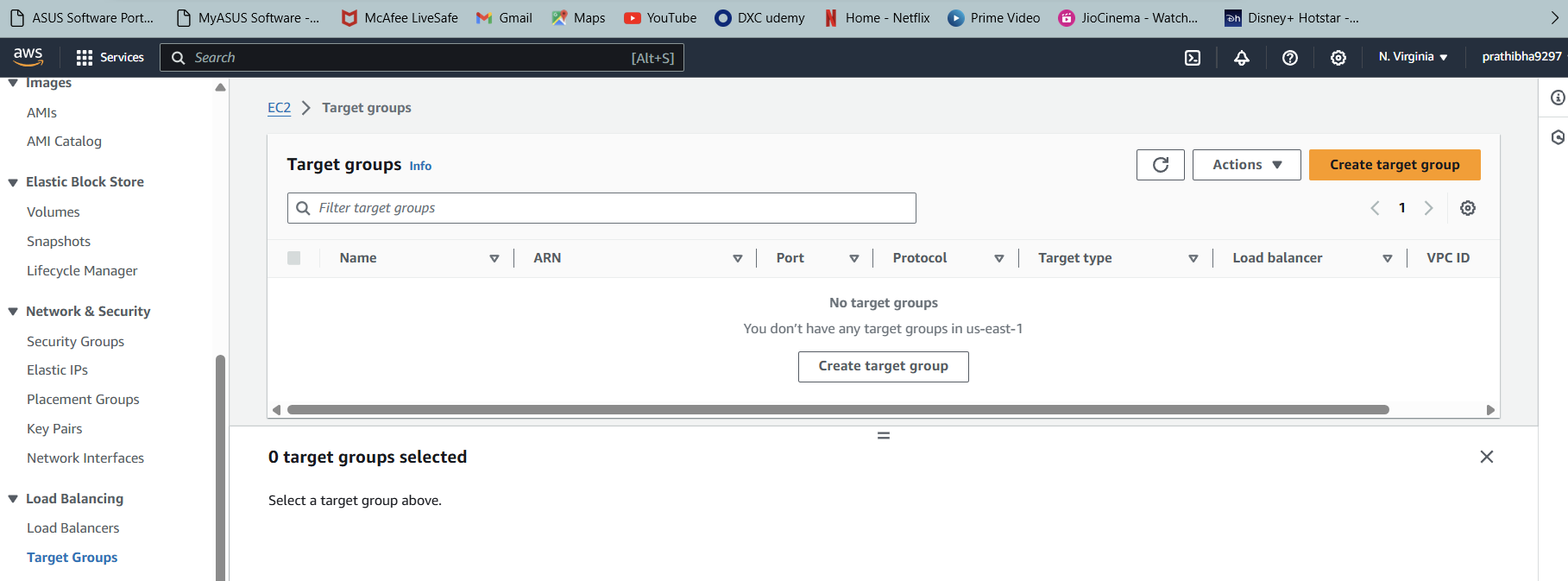
Load balancer:

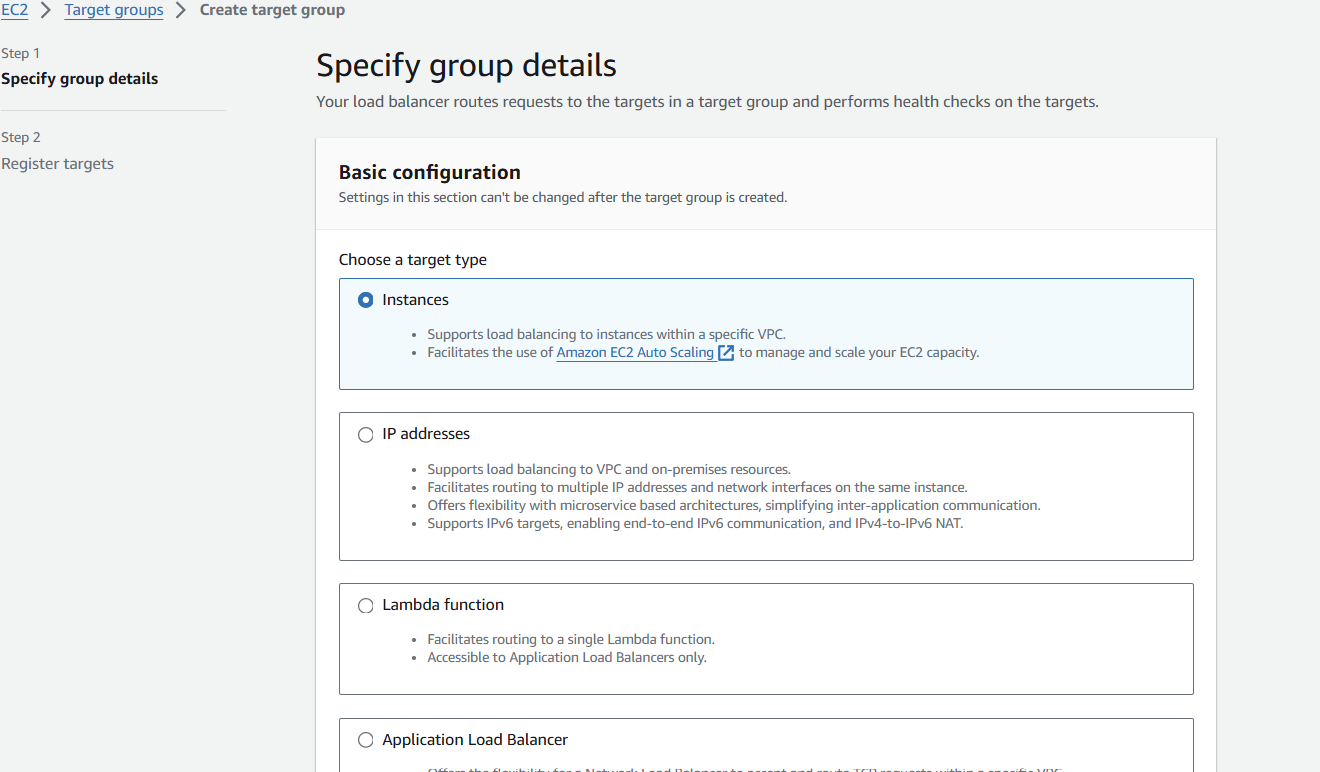
A load balancer distributes incoming network or application traffic across multiple servers to ensure no single server becomes overwhelmed. This improves the availability, reliability, and scalability of applications

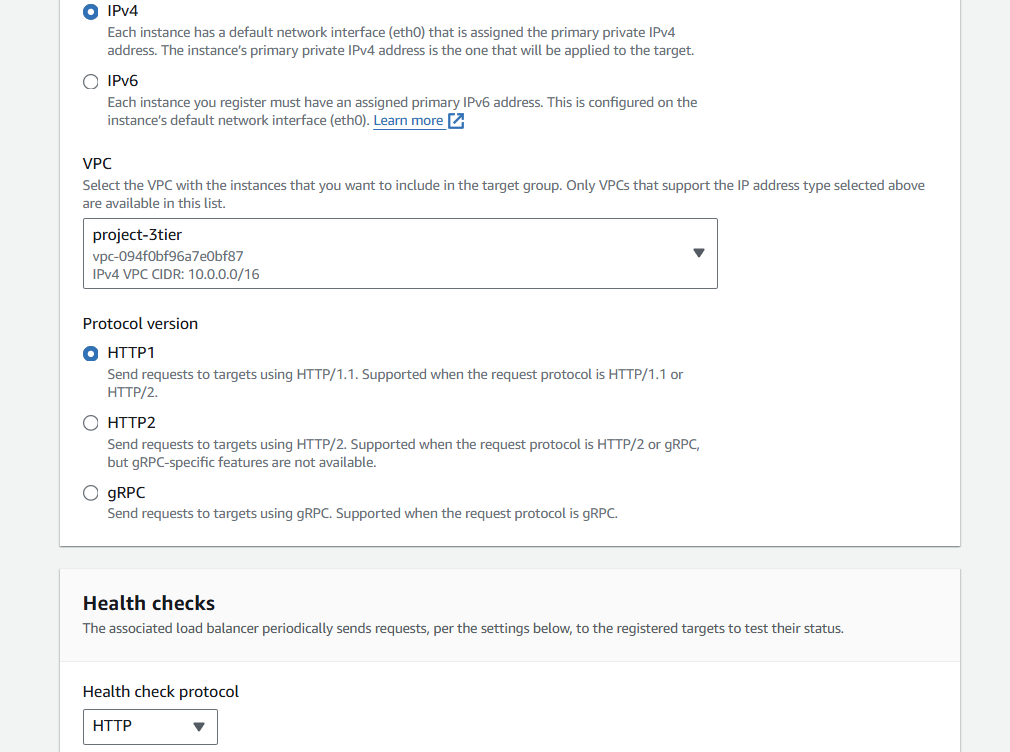
Step1:

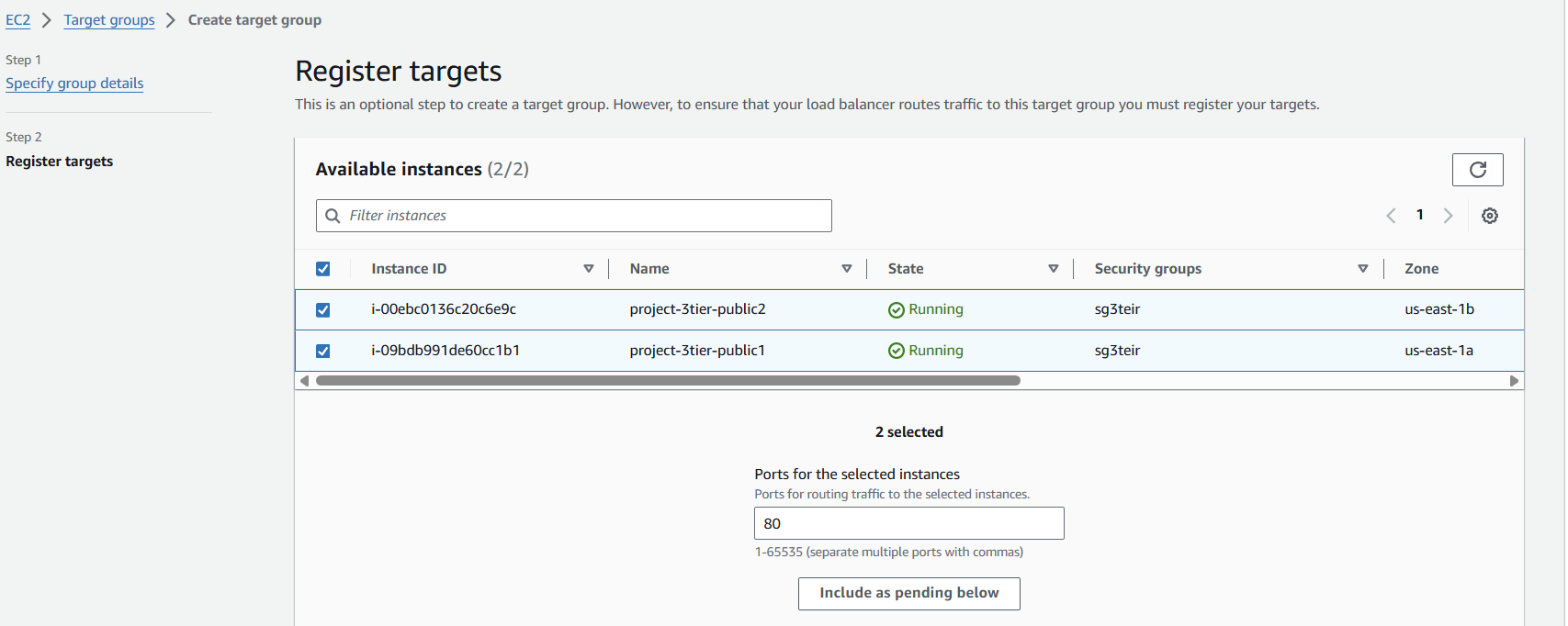
Target group:

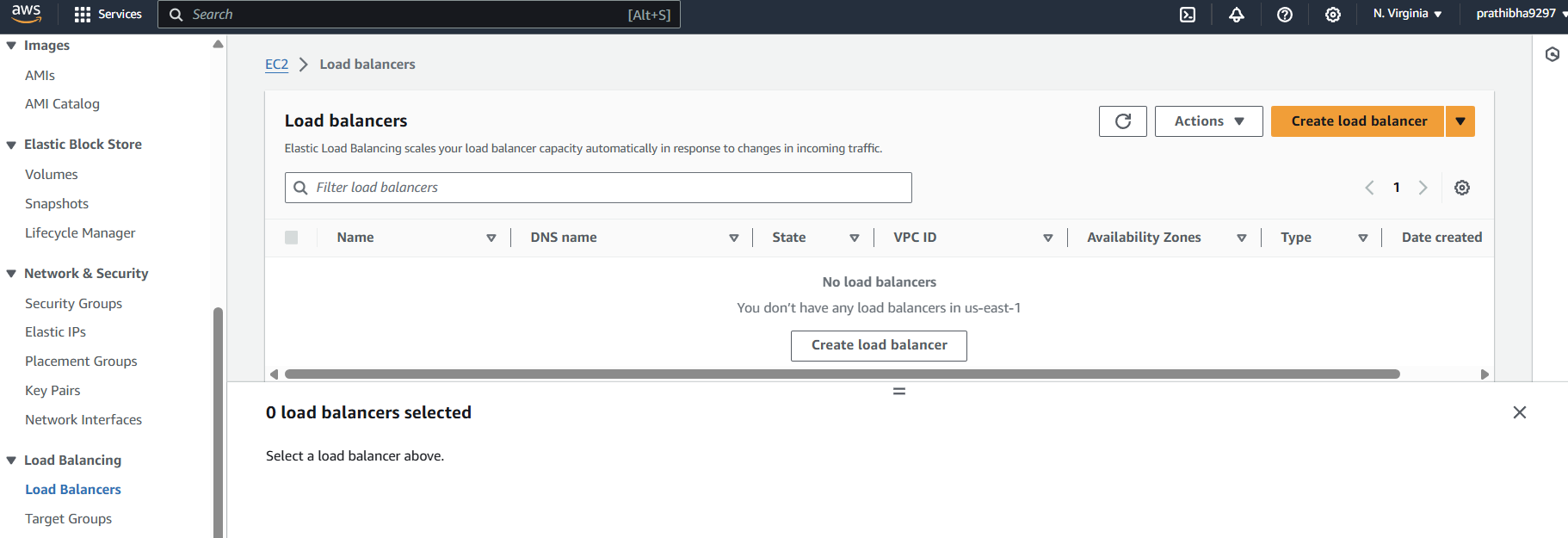
* For load balancer first we need to create Target group
* Click on create Target group->select instances under choose target type->select VPC project-3tier->select instances under register targets->click on include as pending below->click on create target group
* Once target group created go to load balancer.



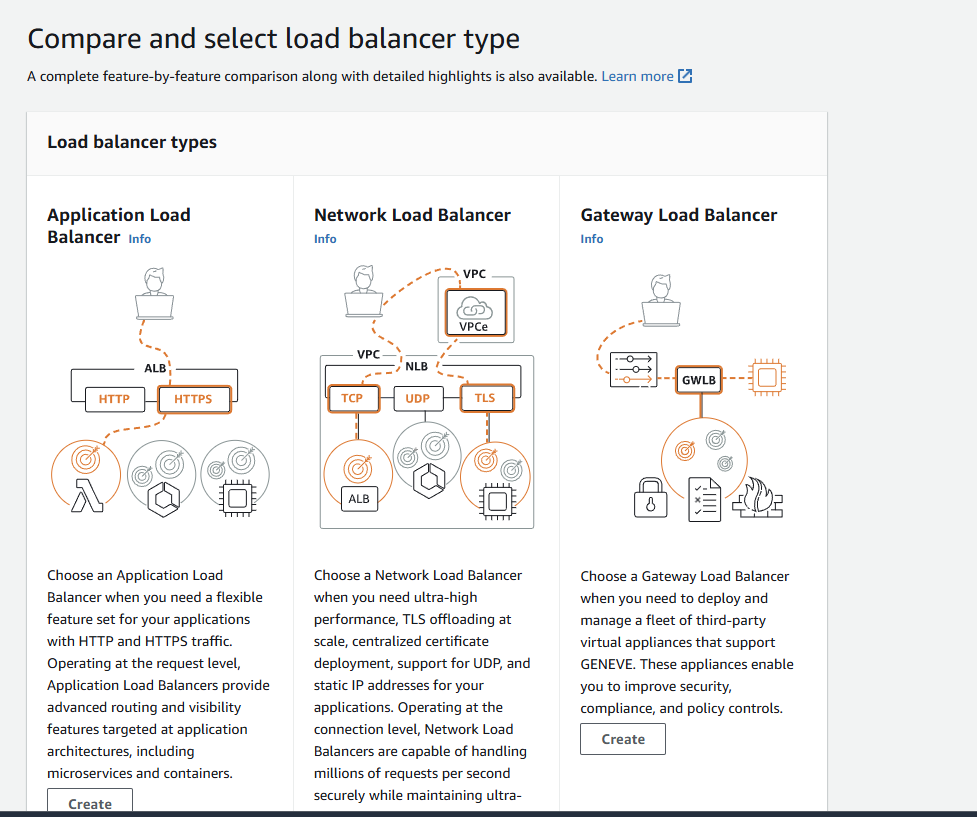




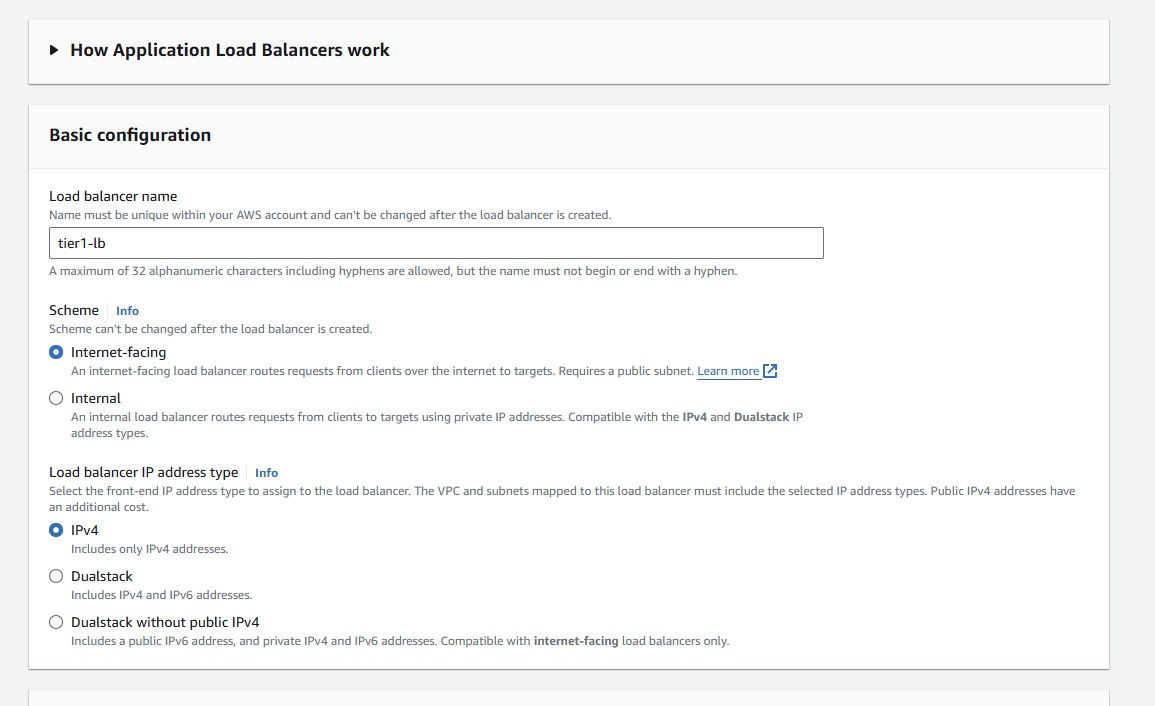


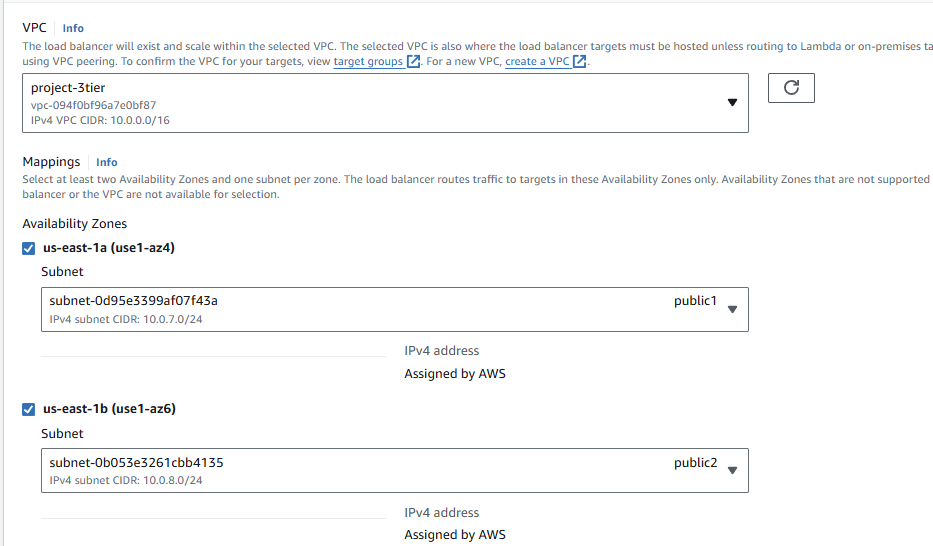


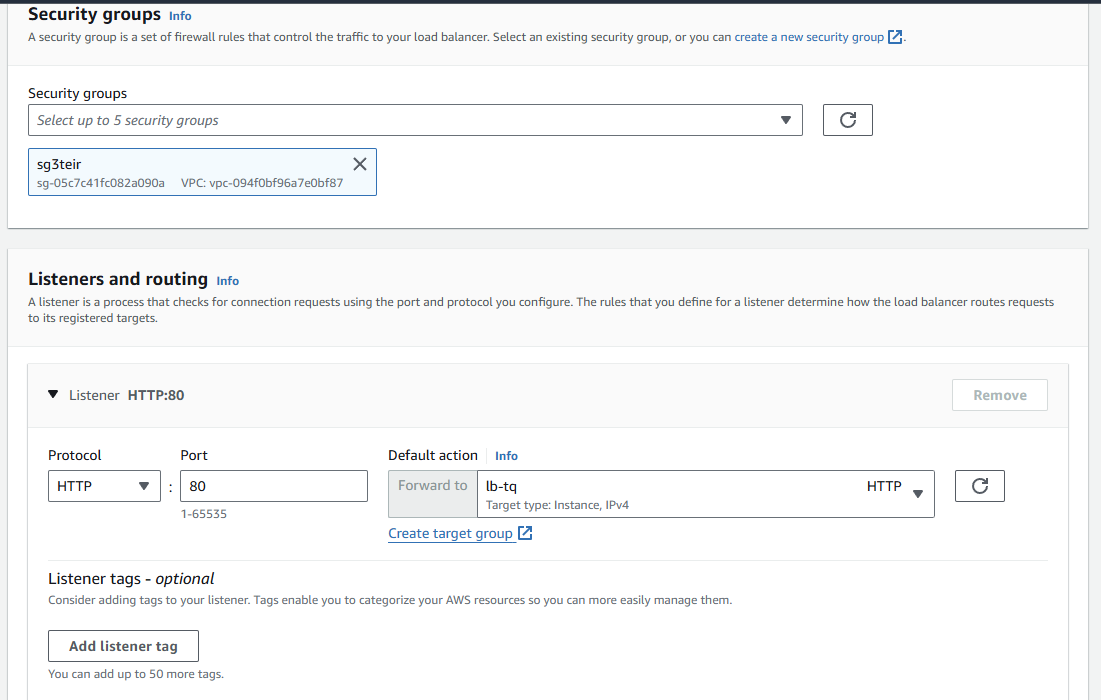
* Go to load balancer and click on create load balancer.
* Select application load balancer

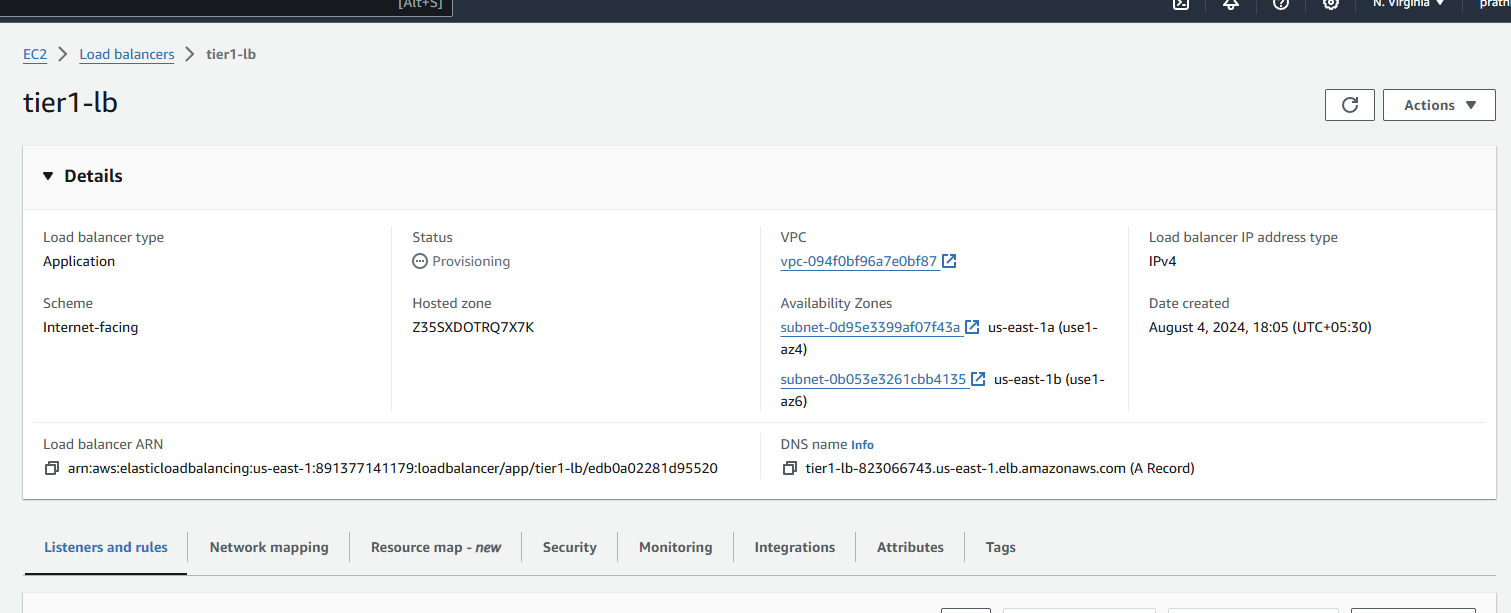


* Give name tag as tier1-lb->click on internet facing under scheme field->select VPC project-3tier->select availability zone us-east-1a & us-east-1b->select subnets public1 & public2->select security group->select target group (lb-tg)->remaining fields leave as default and click on next->click on create load balancer.

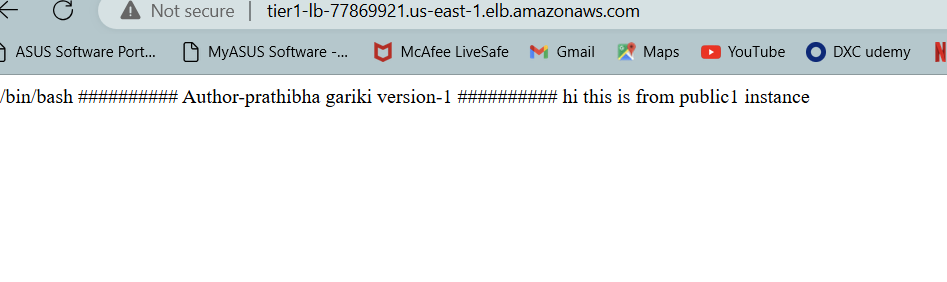


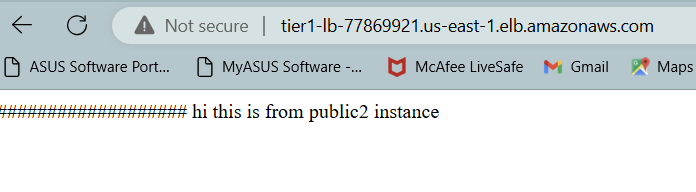






* Once load balancer created copy the DNS name and paste it in chrome.
* Out put shown in below slides.

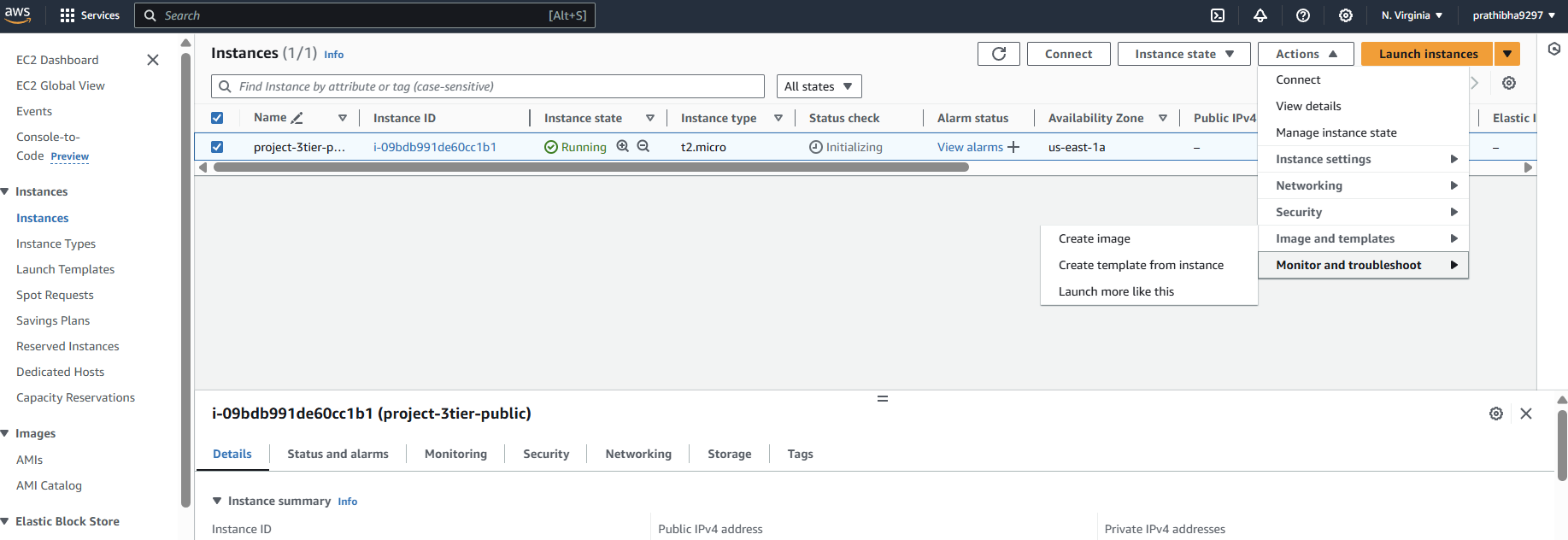


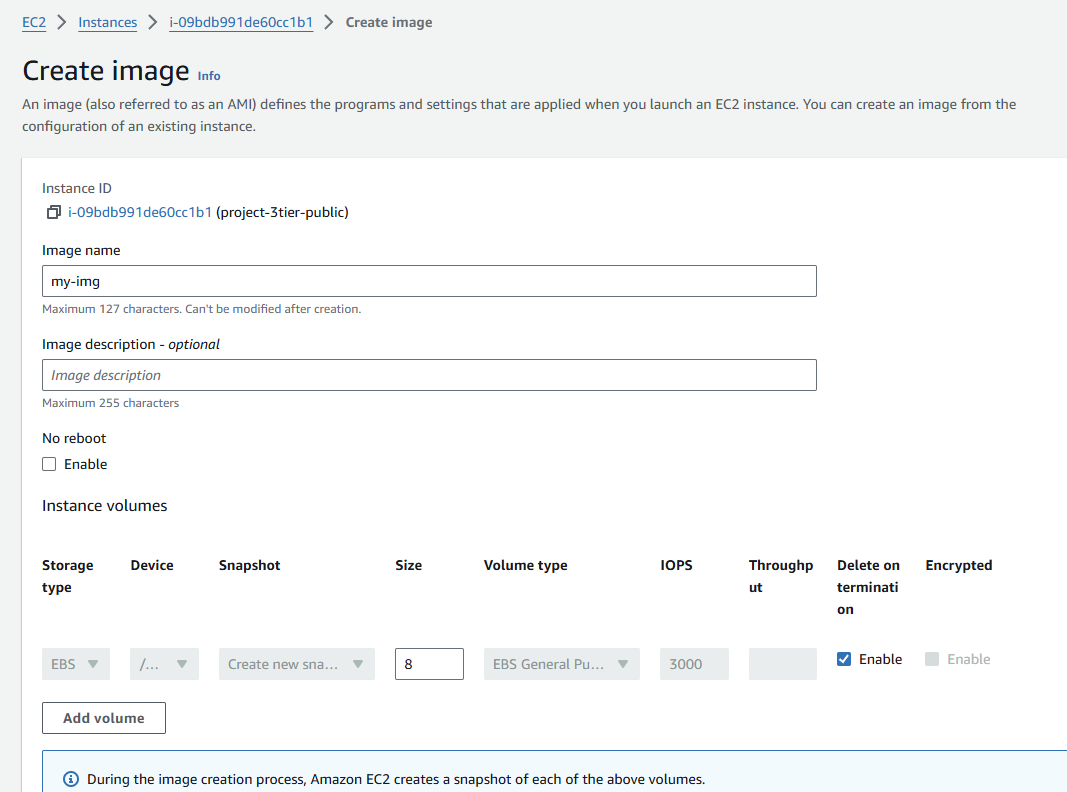


Auto scaling:

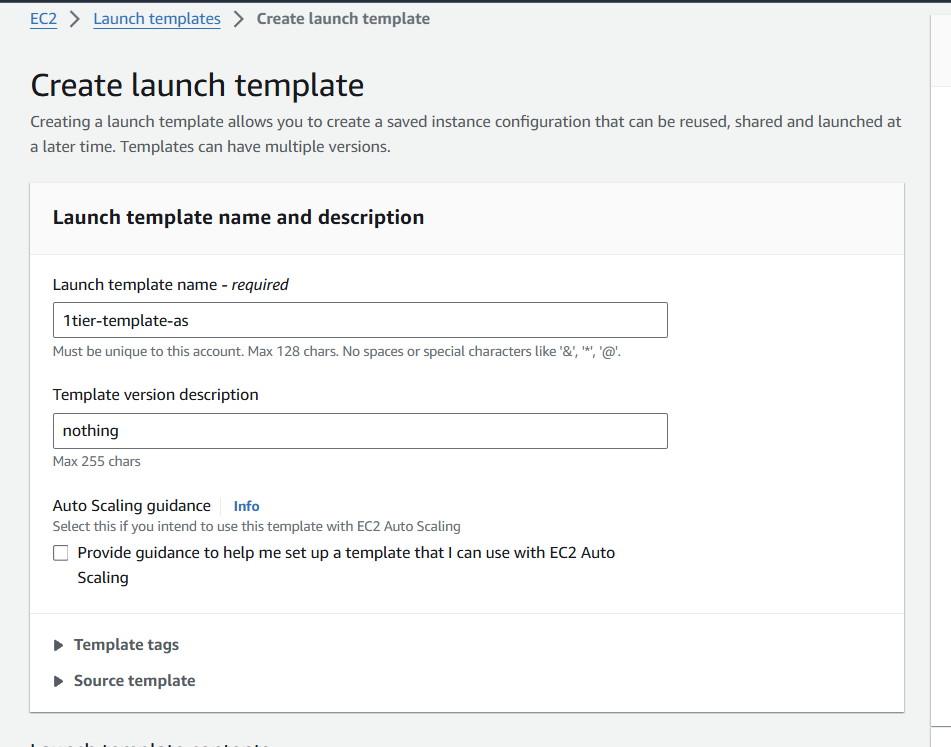
AWS Auto Scaling is a service that automatically adjusts the number of compute resources (such as EC2 instances) based on current demand. This helps maintain performance and optimize costs by scaling your resources up or down according to defined policies

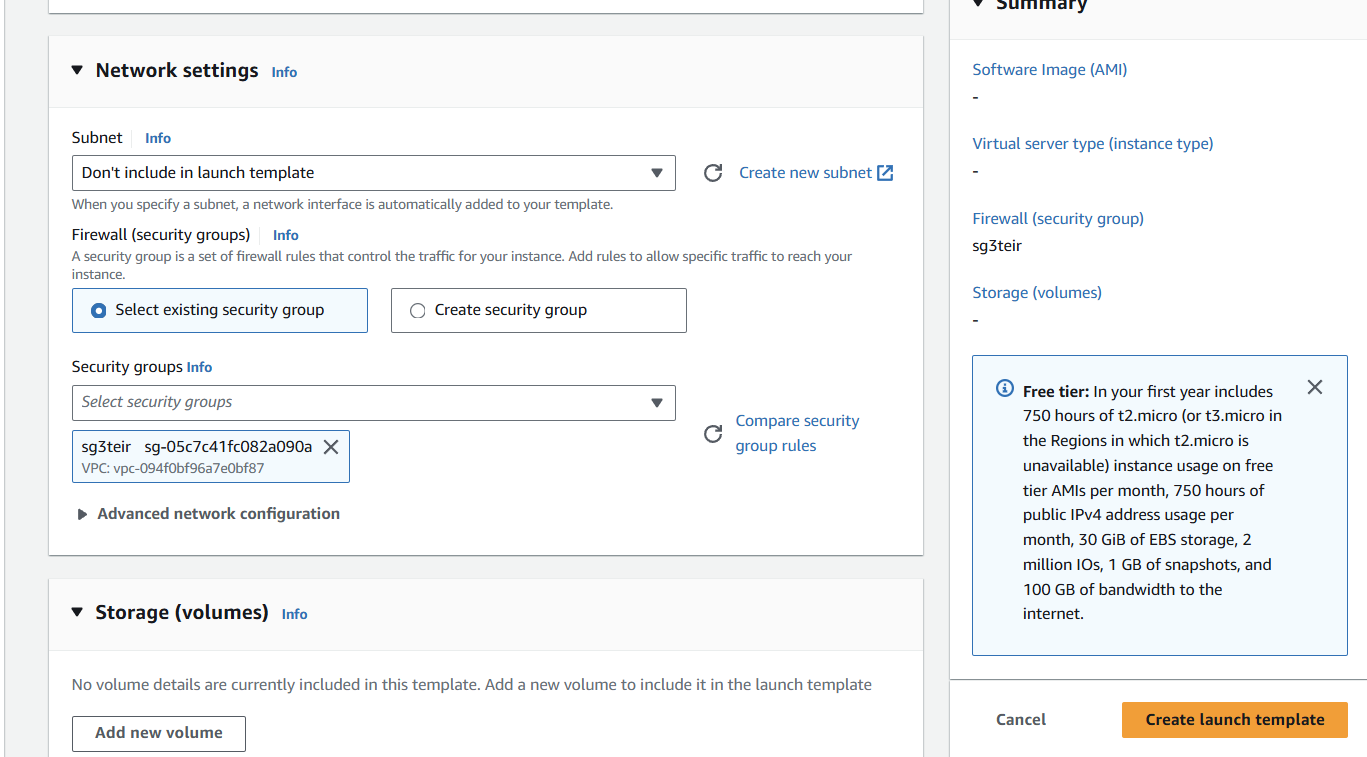
* Create image using instance->select instance and click actions->image and templates->create image.
* Give name tag as my-img under create image field as shown in below slides.



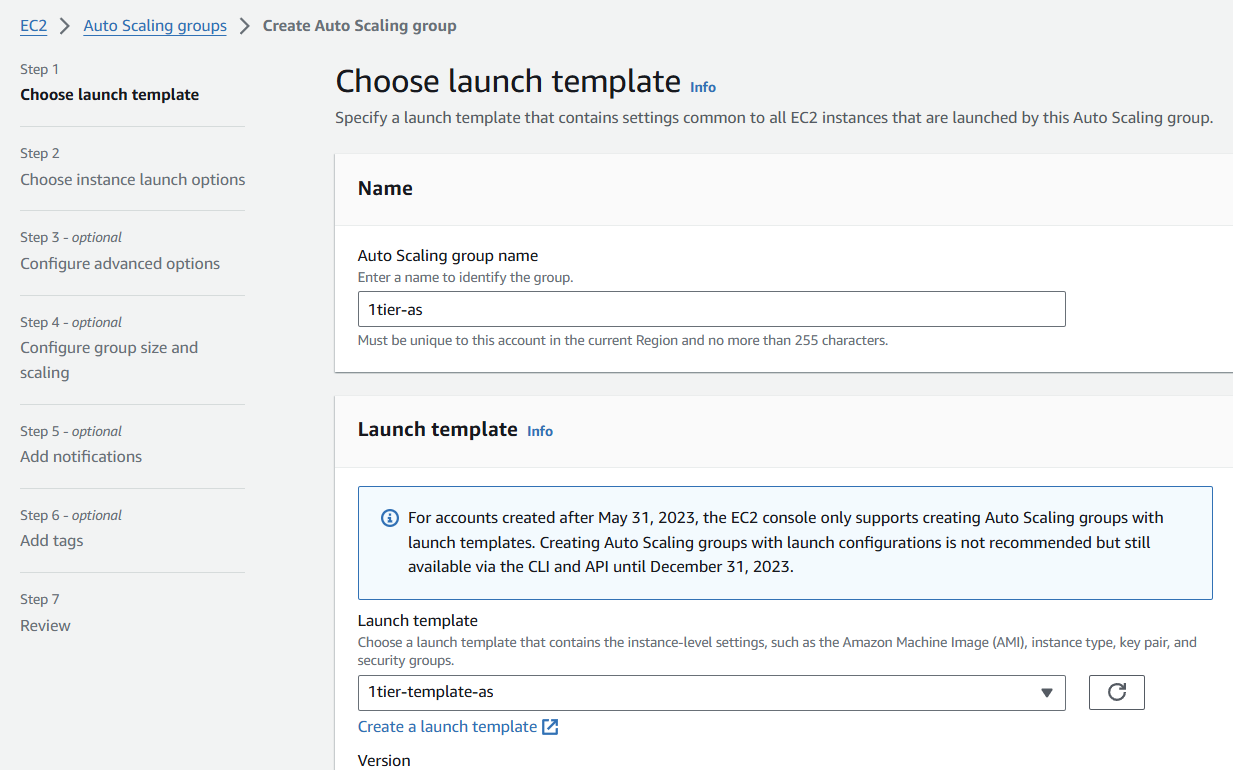


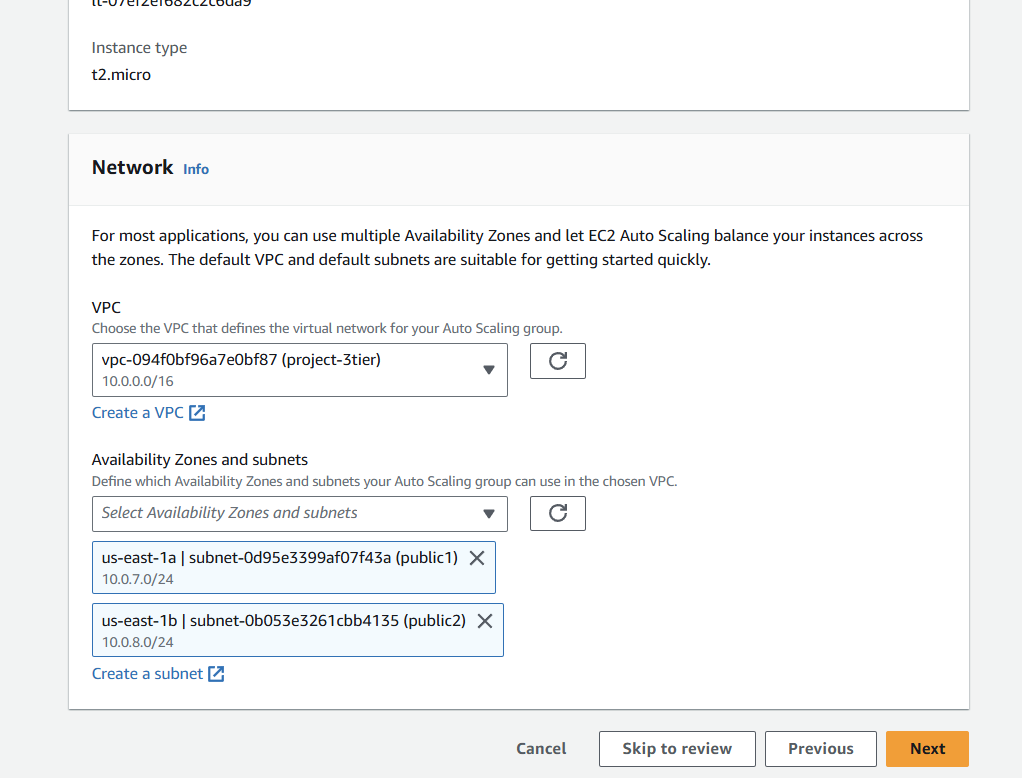
* Once image created go to EC2 dashboard and click on templates
* Give name tag as 1tier-template-as->template version as nothing
* Click the network settings and select existing security group as shown in below.

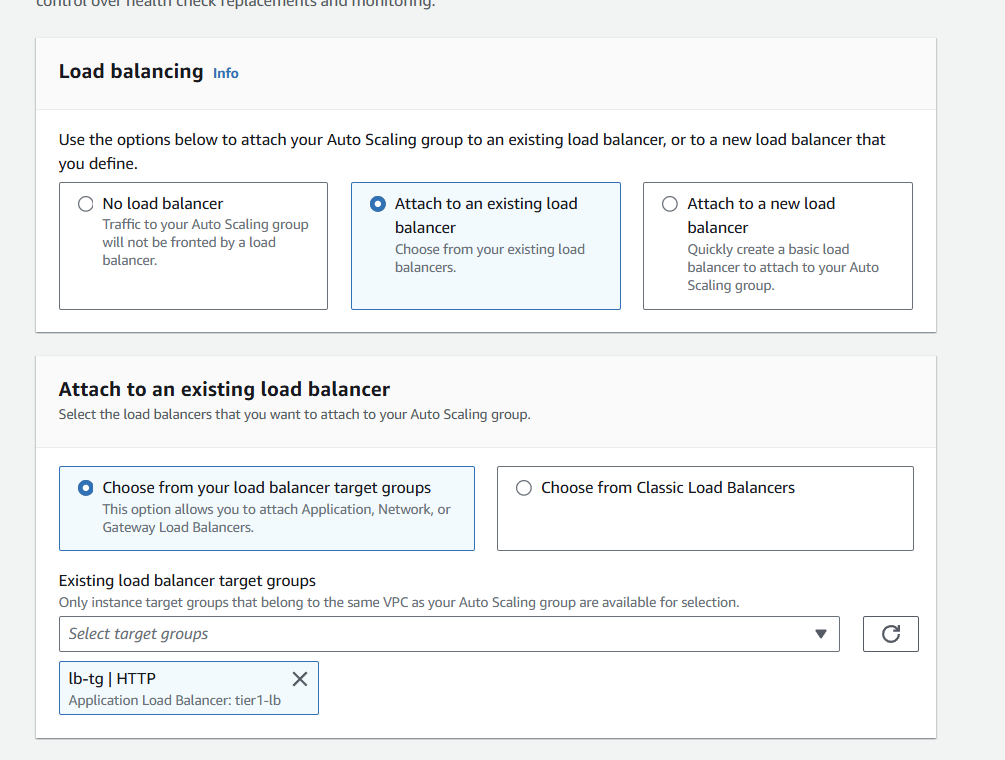


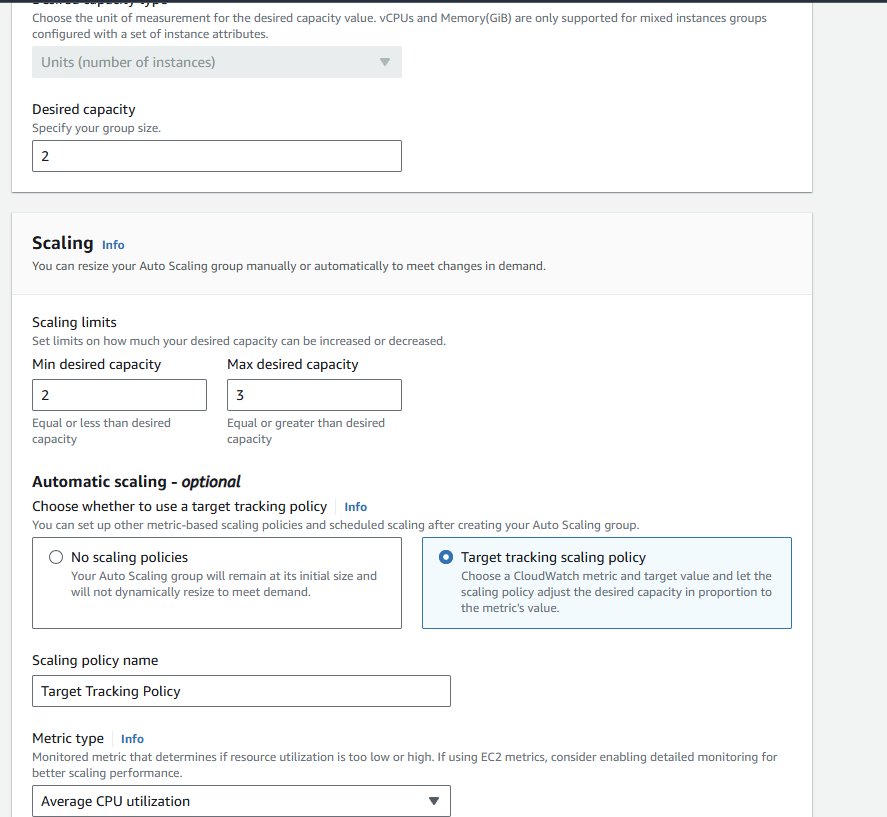


* Go to auto scaling and click on create auto scaling groups
* Give name tag as 1tier-as
* Select created target group-1tier-template-as
* Select the VPC -> select availability zone us-east-1a (public1 subnet) & us-east-1b (public2 subnet)->click attached to an existing load balancer->select existing target group.
* Give desired capacity as 2, min capacity as 2 and max capacity as 3
* Select target tracking scaling policy under auto scaling options.
* Click on create auto scaling.



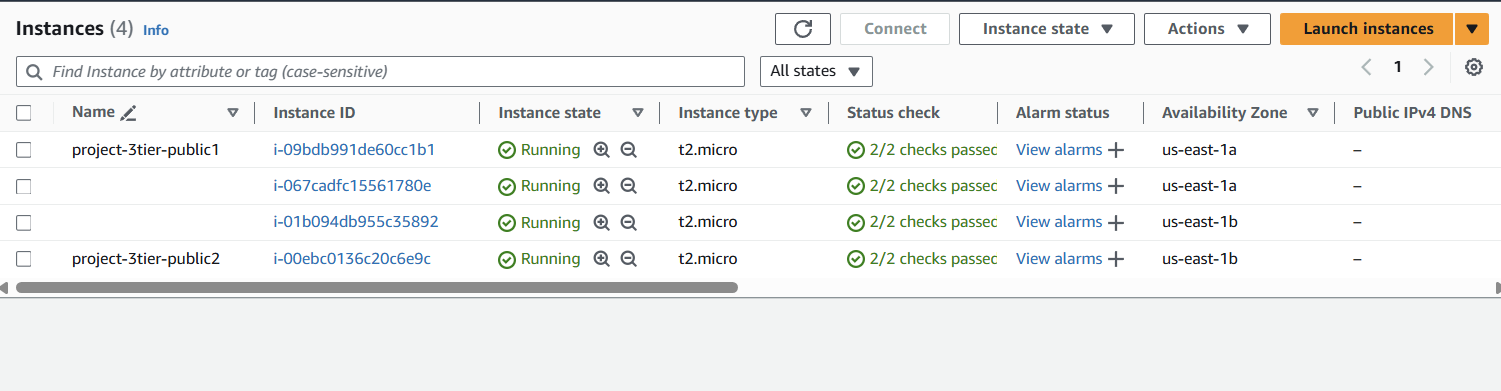






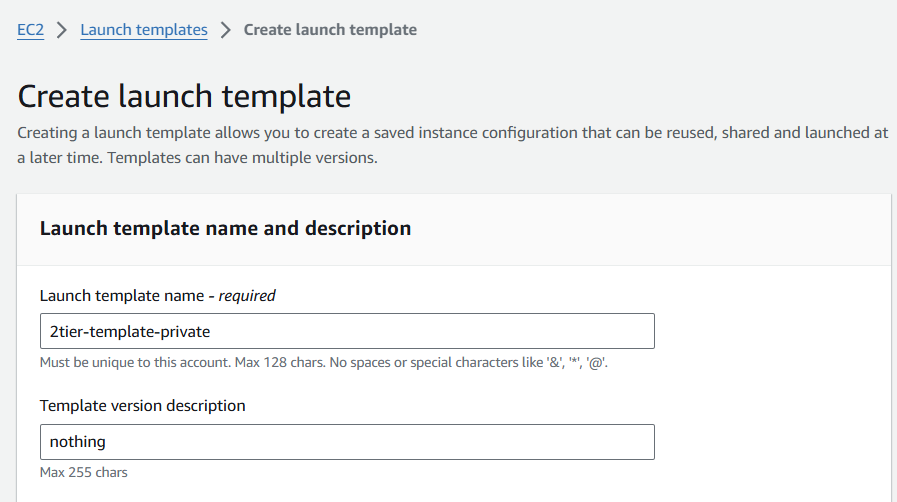


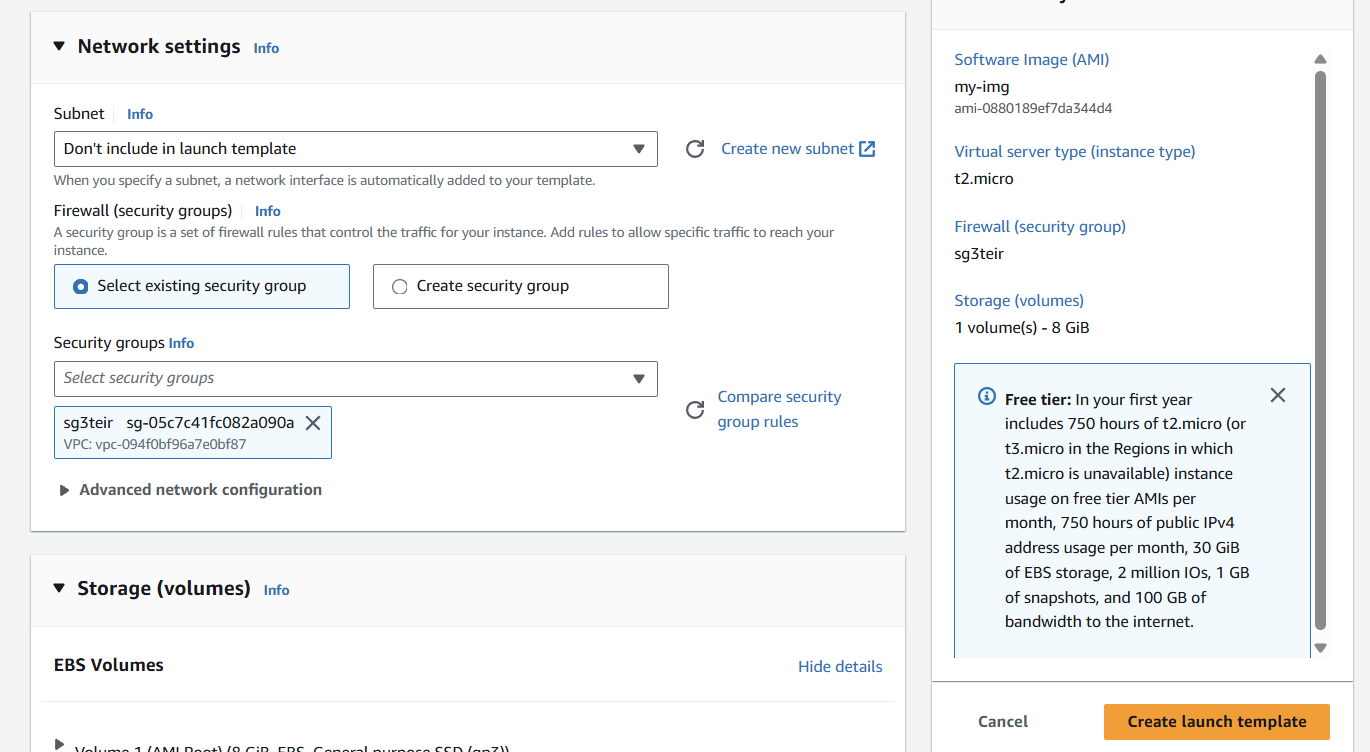
Once auto scaling created another 2 extra instances added. Previously 2 instances created and after auto scaling another 2 instances added to total 4 instances.



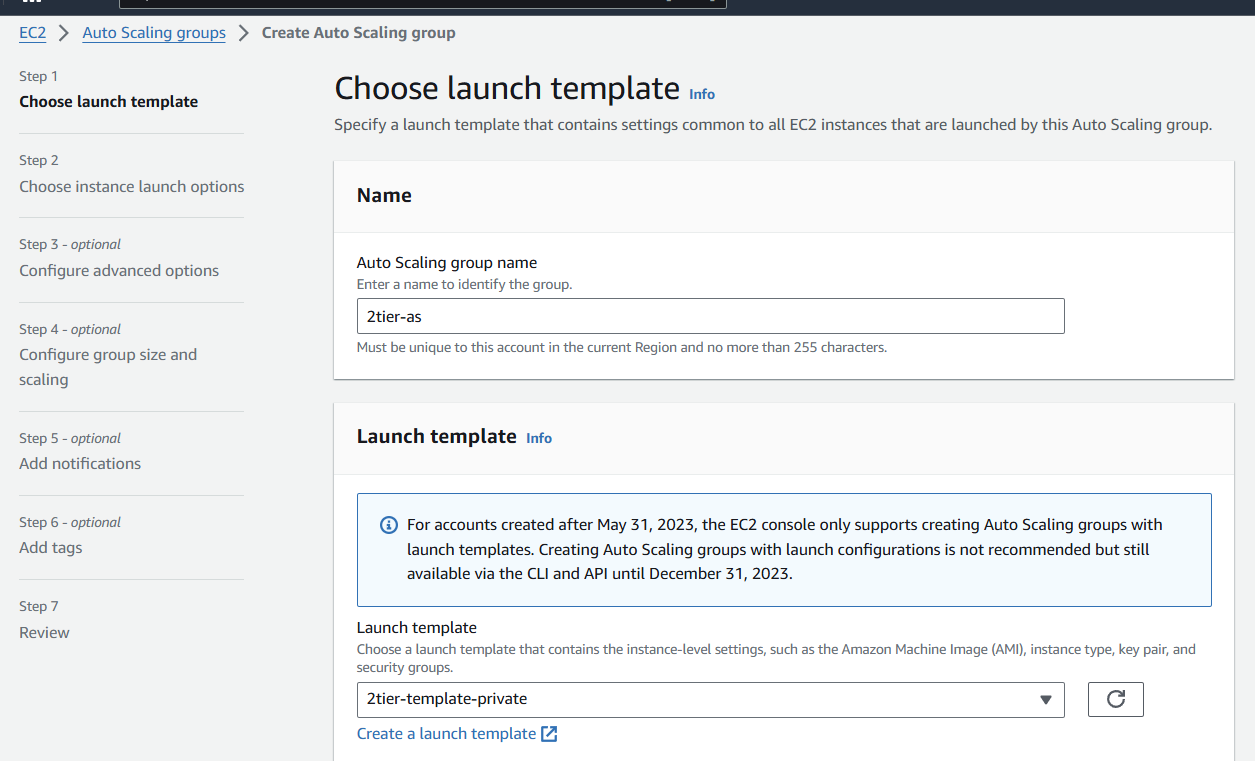
Auto scaling for private subnets:

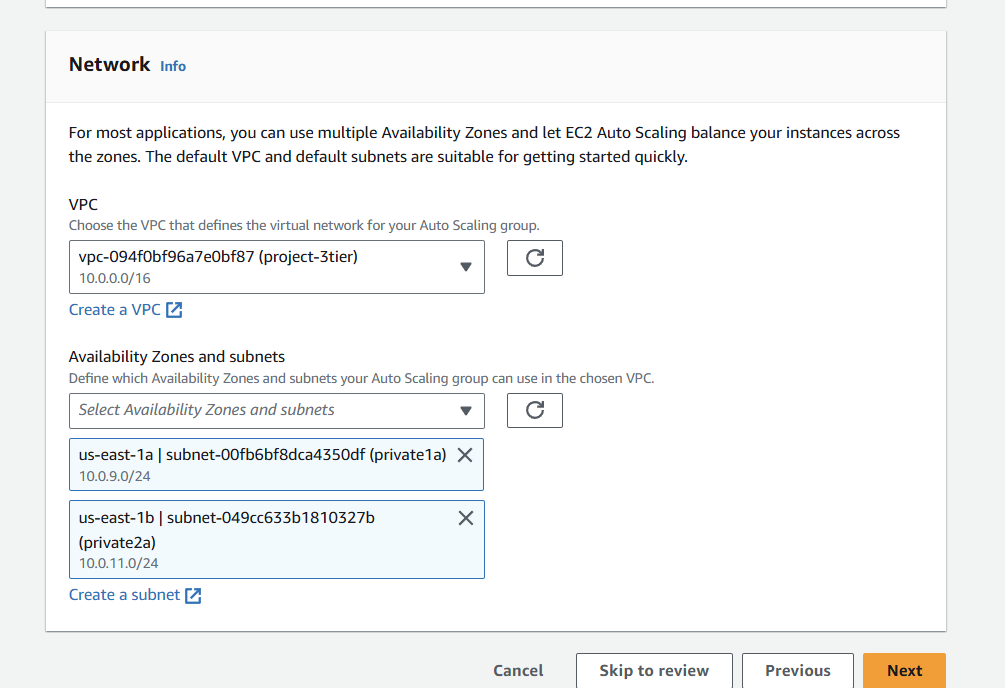
* Give name tag as 2tier-template-private->template version as nothing
* Click the network settings and select existing security group and click on create launch template as shown in below.

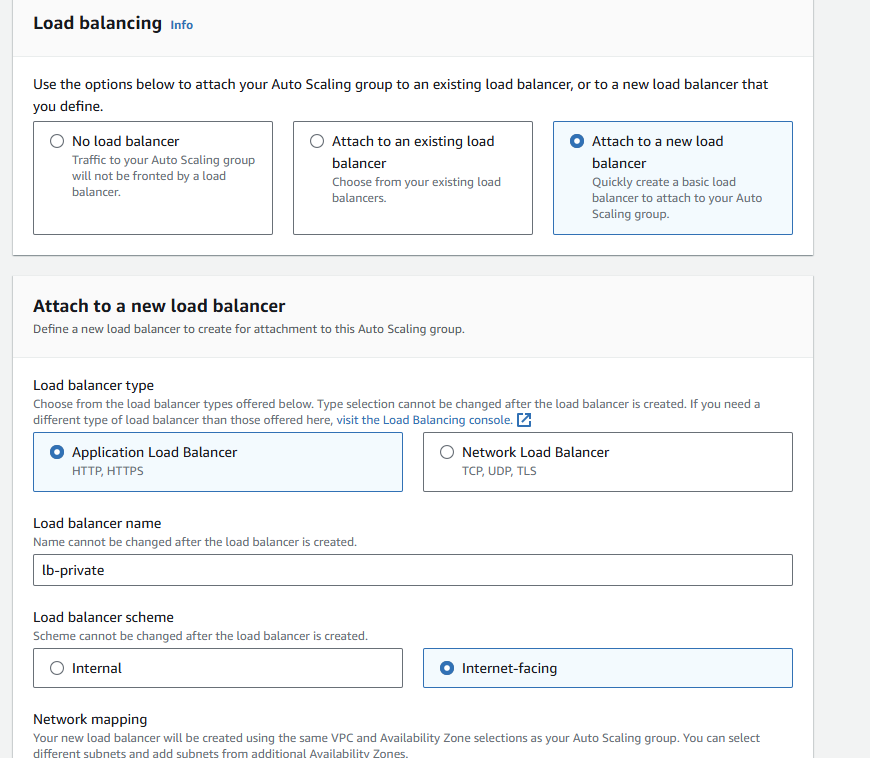


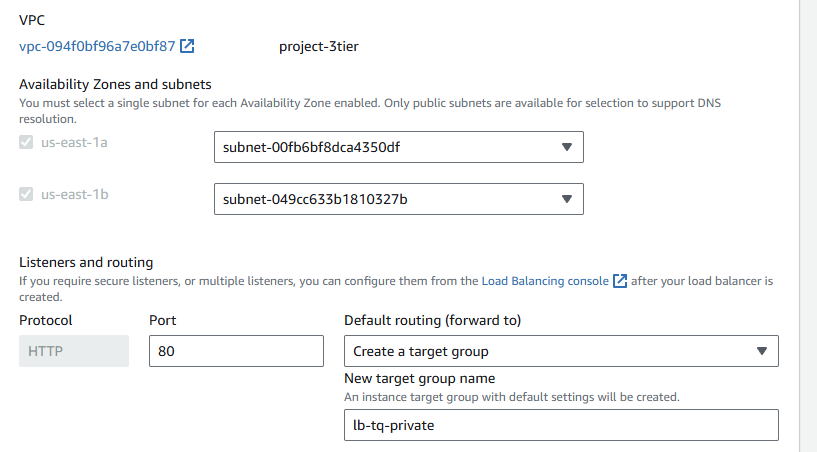


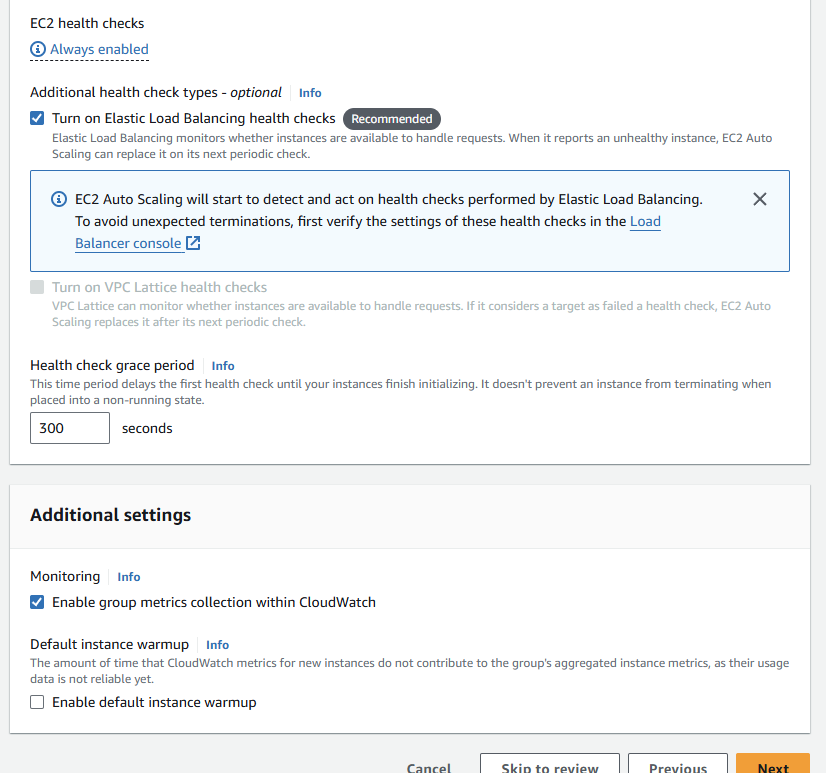
* Go to auto scaling and click on create auto scaling groups
* Give name tag as 2tier-as
* Select created target group-2tier-template-private
* Select the VPC -> select availability zone us-east-1a (private1a subnet) & us-east-1b (private2a subnet)->click attached to new load balancer->for load balancer give name tag as lb-private -> click on internet facing->select private1a & private2a subnets under availability zone->click on create target group->give name as lb-tg-private.
* Give desired capacity as 2, min capacity as 2 and max capacity as 3
* Select target tracking scaling policy under auto scaling options.
* Click on create auto scaling.

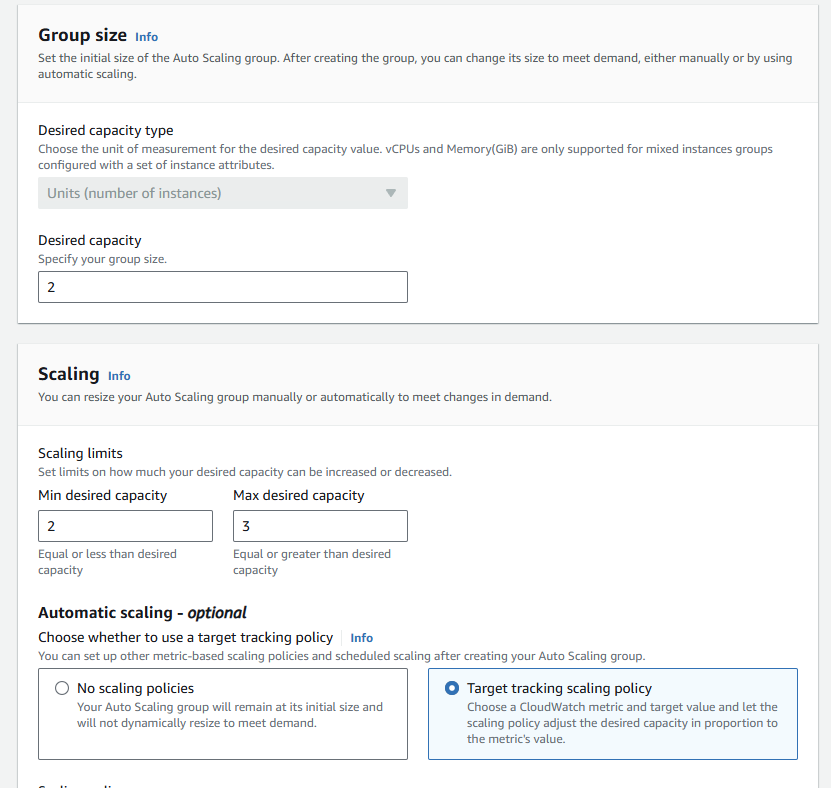




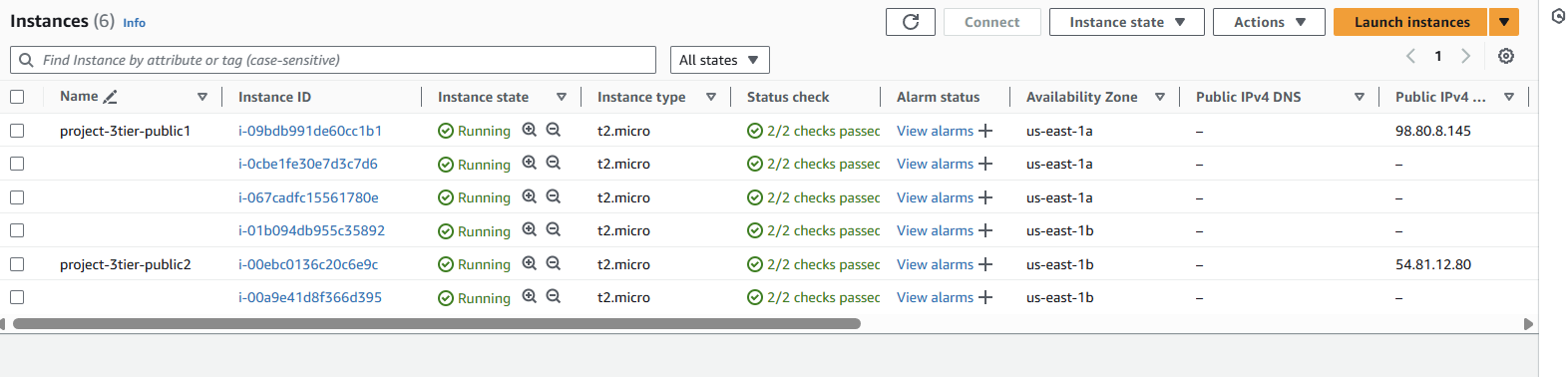








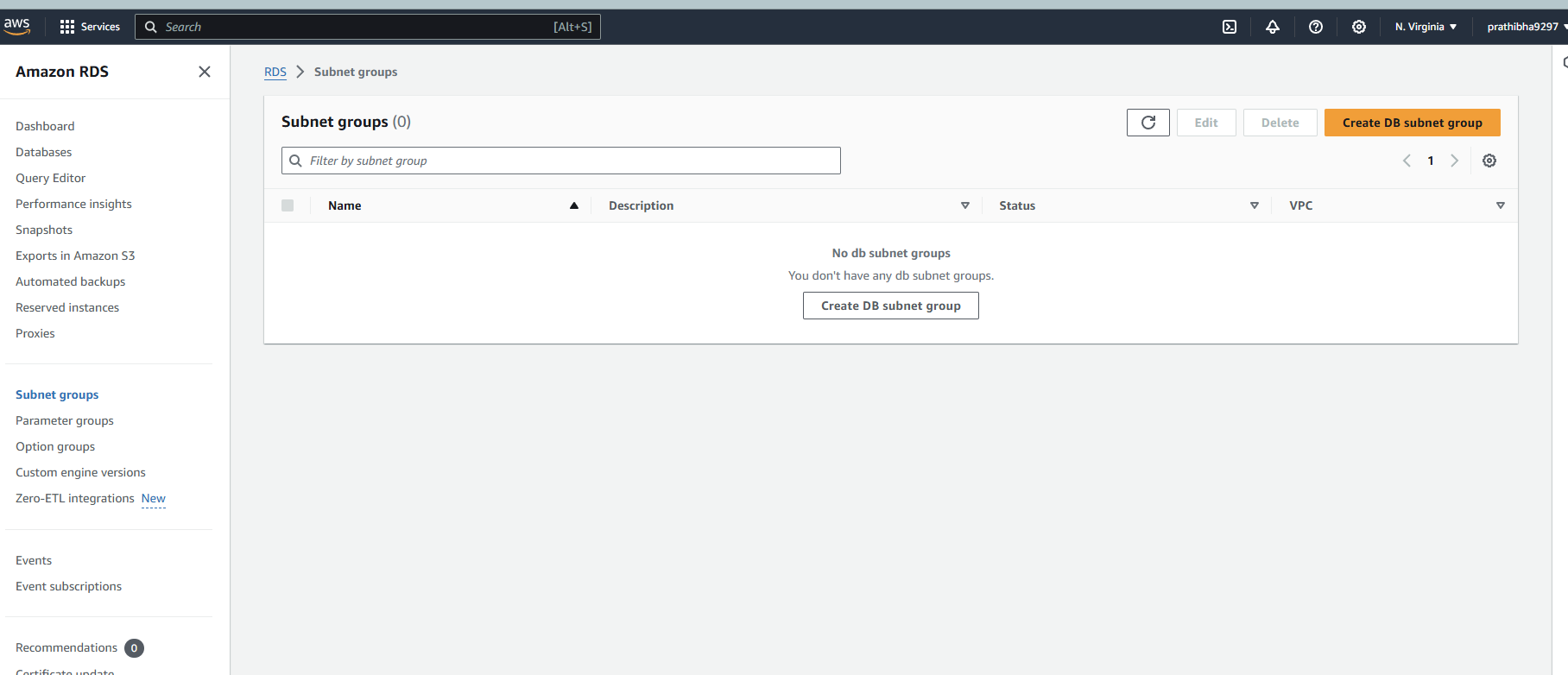
After autoscaling for private subnets (private1a & private 2a) another 2 instances added. Previously 4 instances are available after autoscaling another 2 instances added total 6 instances.

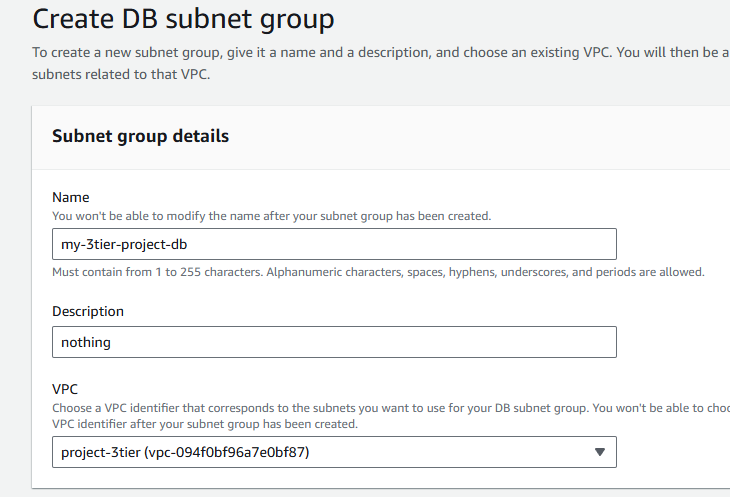


**RDS:**

1.DB subnet:

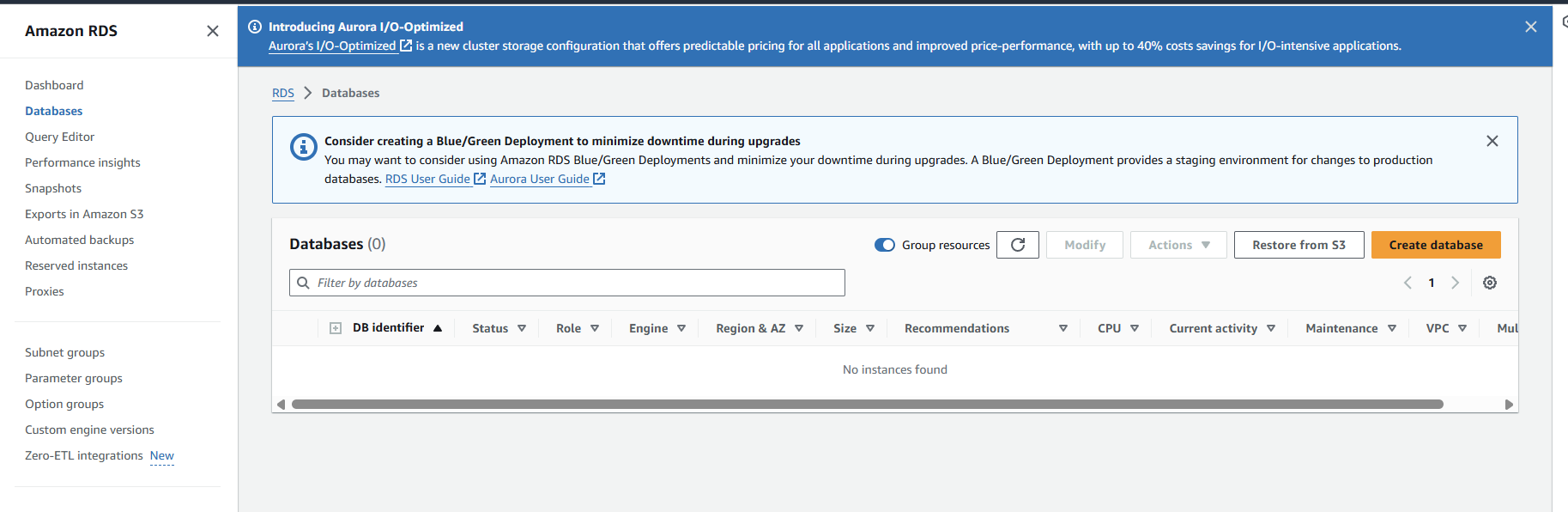
* Go to subnet group and click on create DB subnet group.
* Give name tag as my-3tier-project-db
* Select VPC project-3tier and then click on create DB subnet.

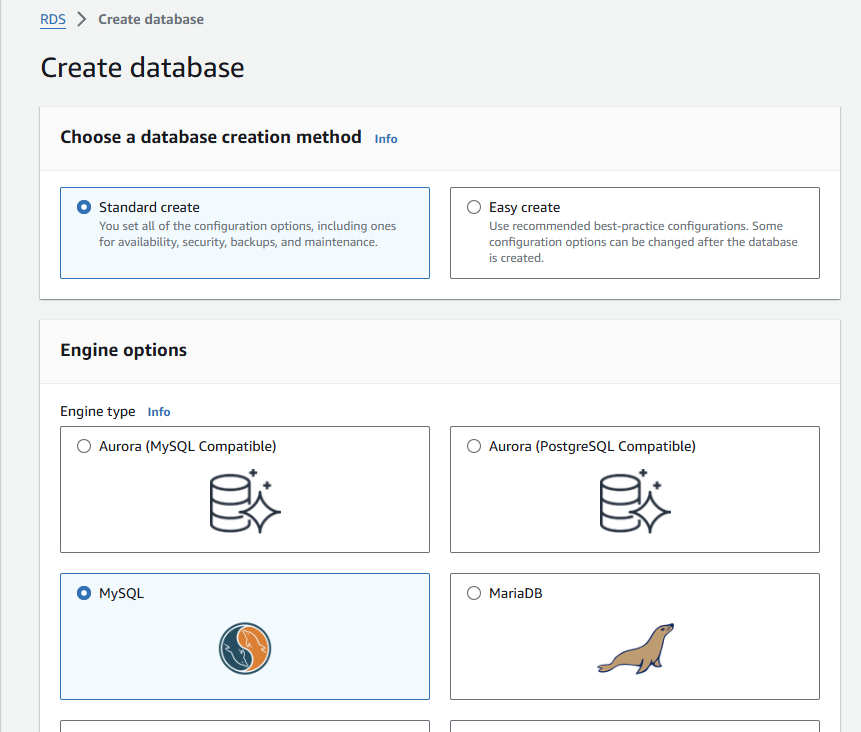


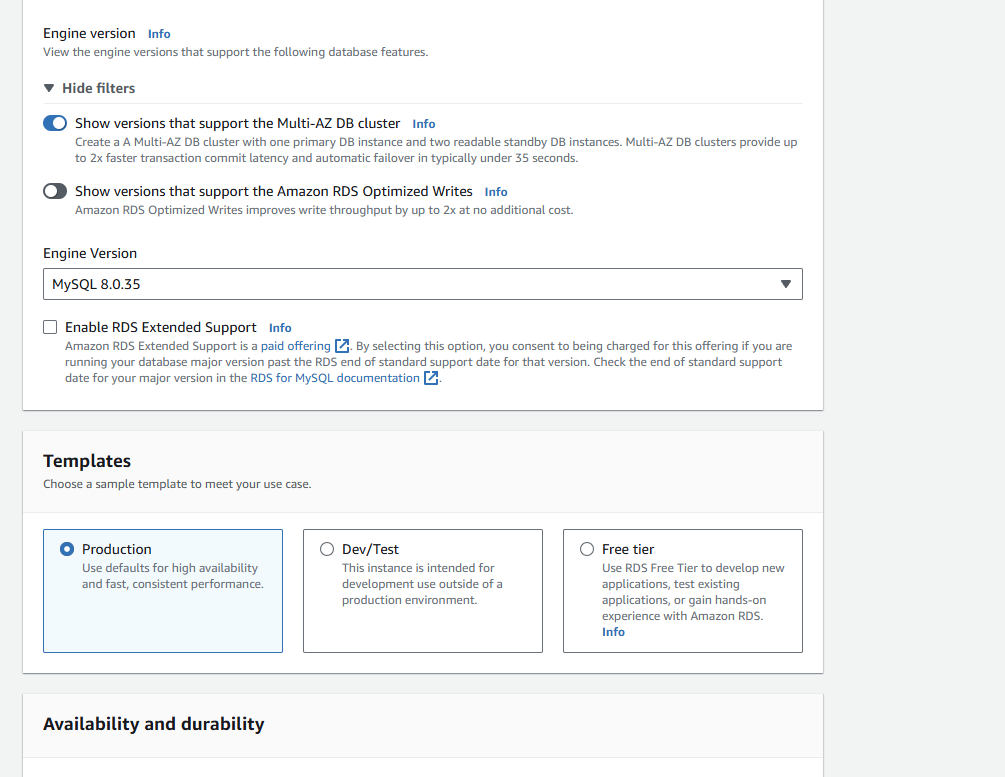


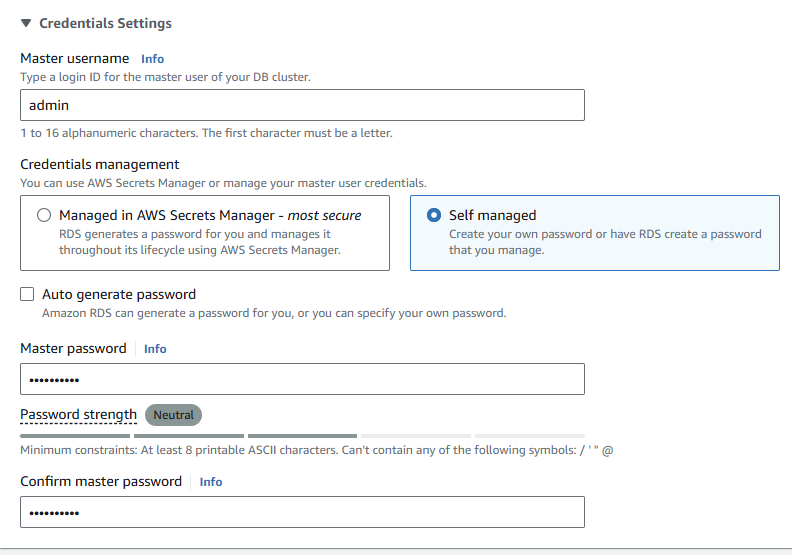
2.Create database:

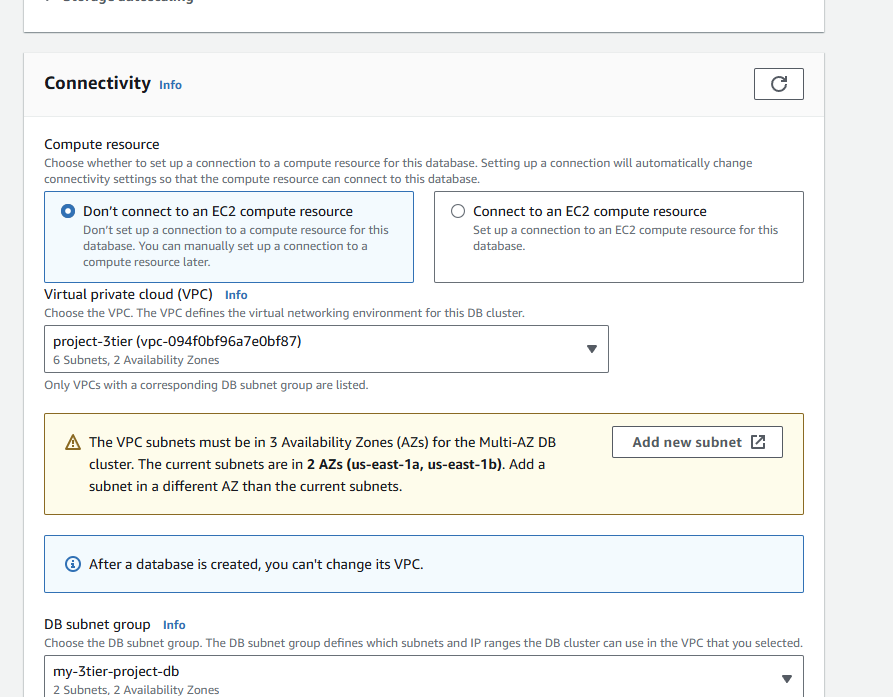
* Go to RDS and click on create database
* Click on standard create->click on my SQL engine option->click on multi AZ DB cluster->click production in templates->click on self-managed->create password->click on do not include an ec2 instance-> select VPC->select DB subnet group.
* Public access click yes->select existing security group
* Remaining fields leave as default and click on create database as shown below slides.

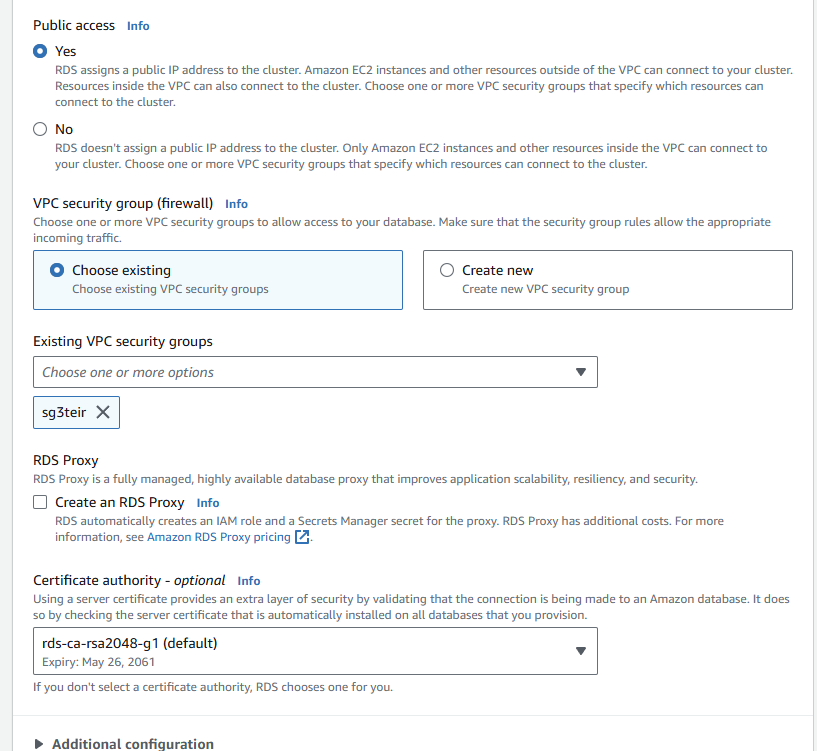


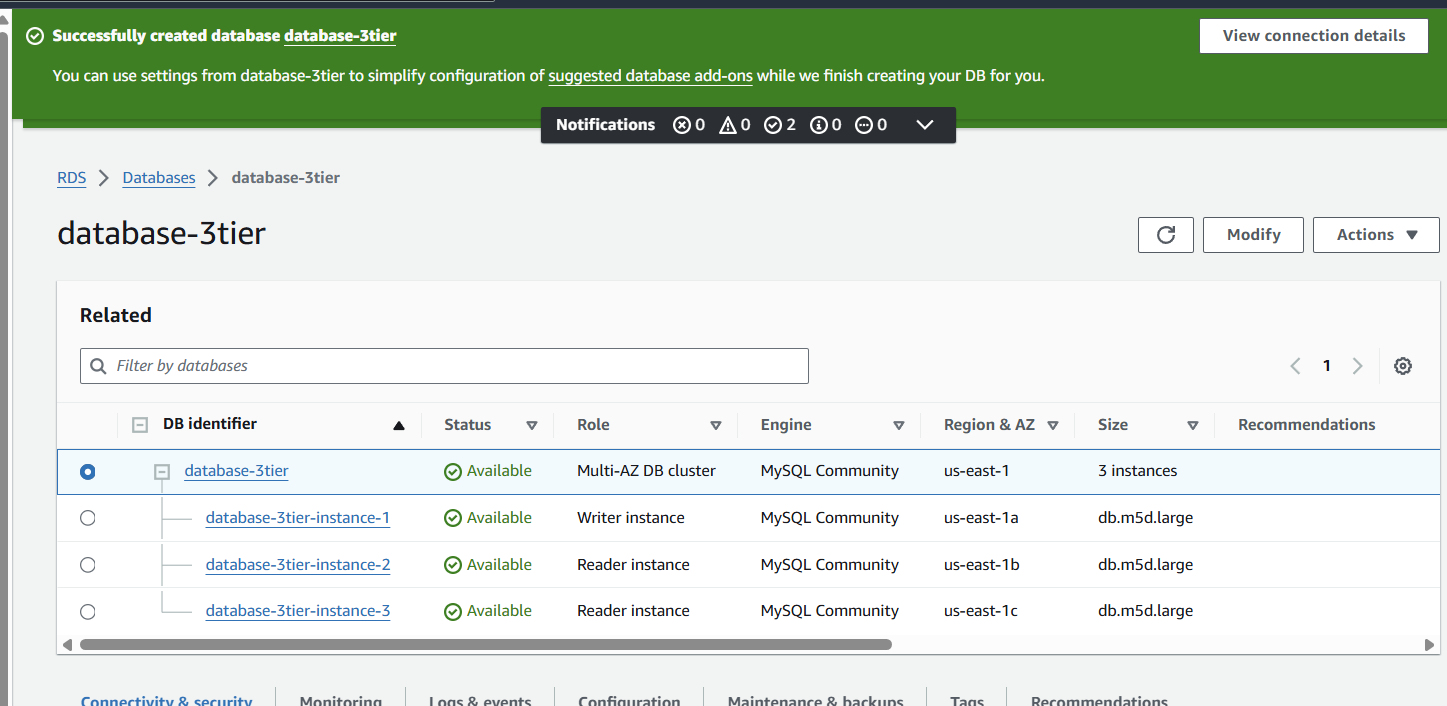






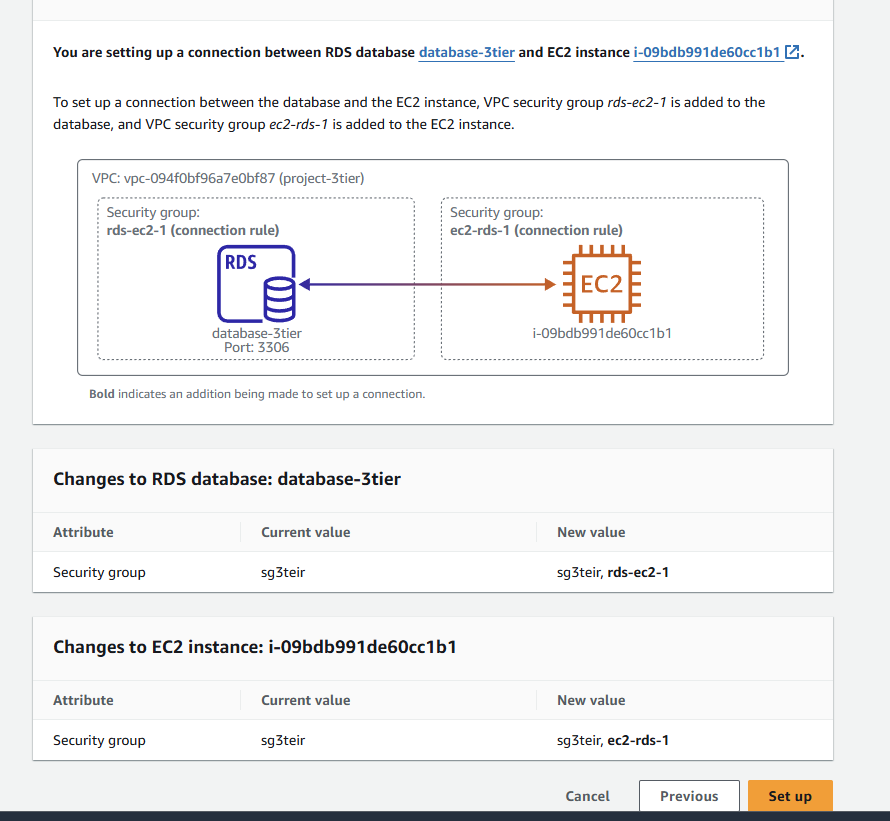


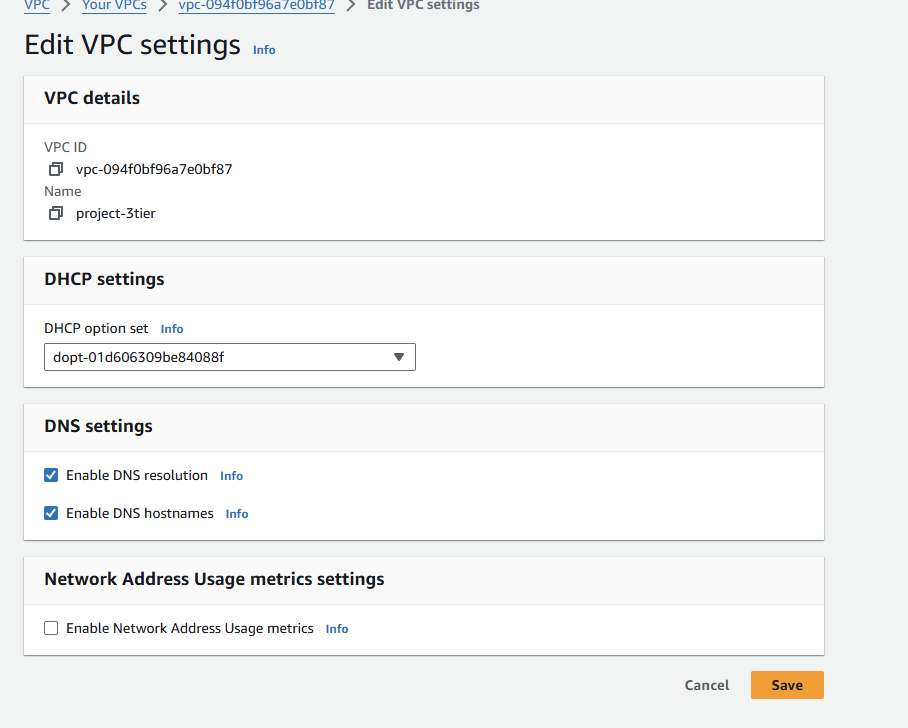


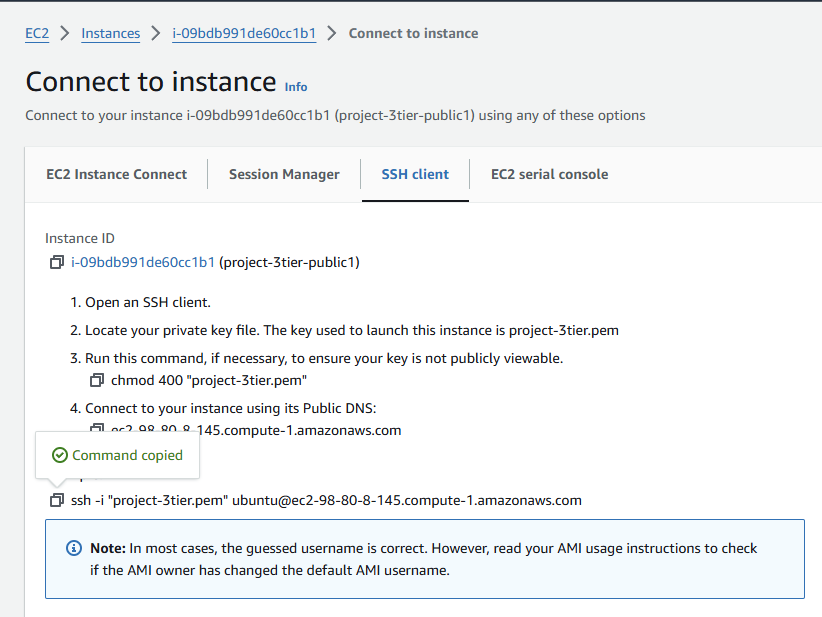








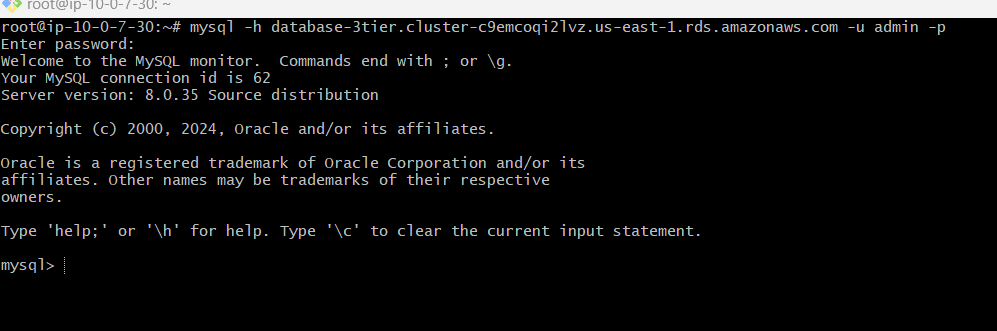




* Once created the RDS connect to the private instance through public instance.



* Install MY SQL->sudo apt install mysql-server
* Restart ->sudo systemctl start mysql service
* Mysql -h database-3tier.cluster-c9emcoqi2lvz.us-east-1.rds.amazonaws.com(rds end point) -u admin -p->Enter password-> u can connect to my SQL as shown in below



* For creating table->

CREATE TABLE Persons (  
    ID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Age varchar(255)  
);

* Create the table in in my sql
* The output of creating table as shown in below.

