Task 2- Create 3 instances , install nginx and apply ALB

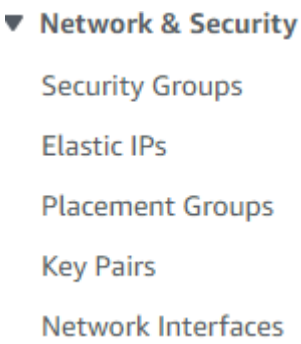
ALB – ALB abbreviates as Application load balancer.

Load balancer is used to manage the incoming traffic on servers.

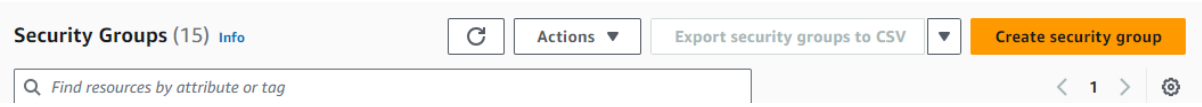
For creating Application load balancer the following steps need to be followed:-

- First, we need to create security group
- Then we need to create 3 instances with 3 different availability zones
- Install nginx in 3 instances
- Create target groups and link it with the 3 created instances
- Attach the created target group with the load balancer
- This is done because of balancing the traffic or data between 3 instances
- Let us do this in detail

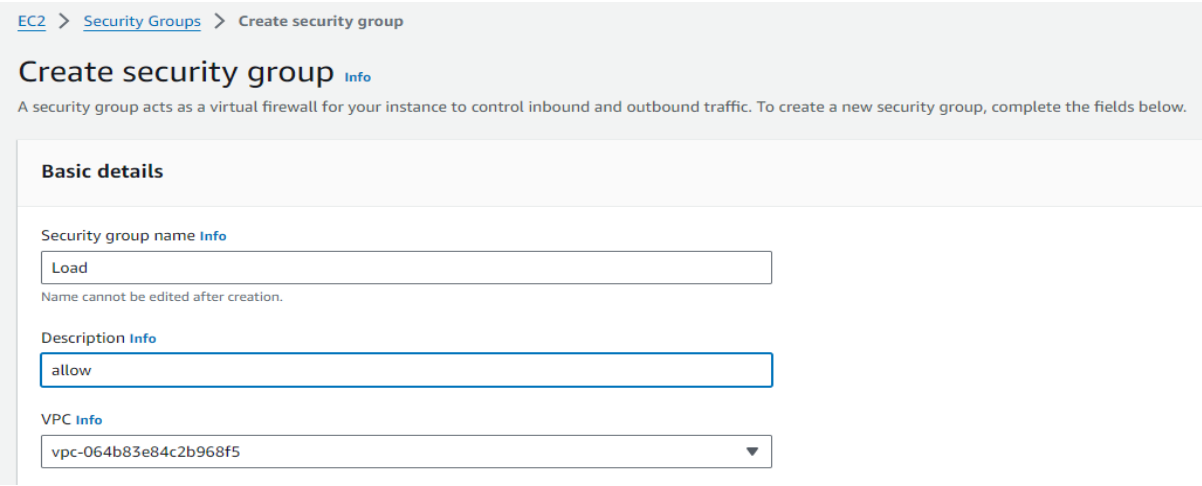
Select security groups under network and security.



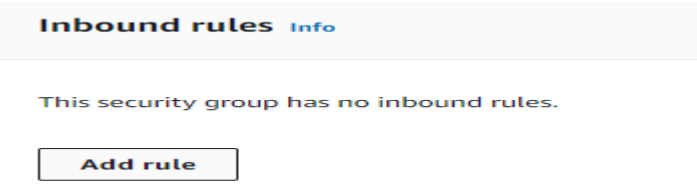
Click on create security group.



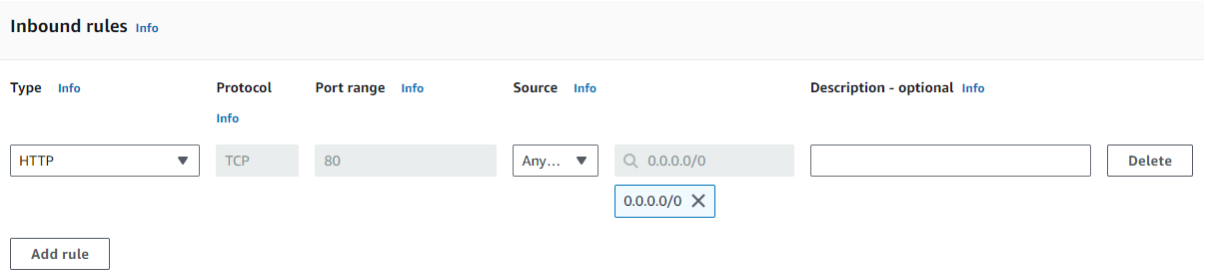
Give security name and give description as allow.



Now click on add rule in inbound rules.

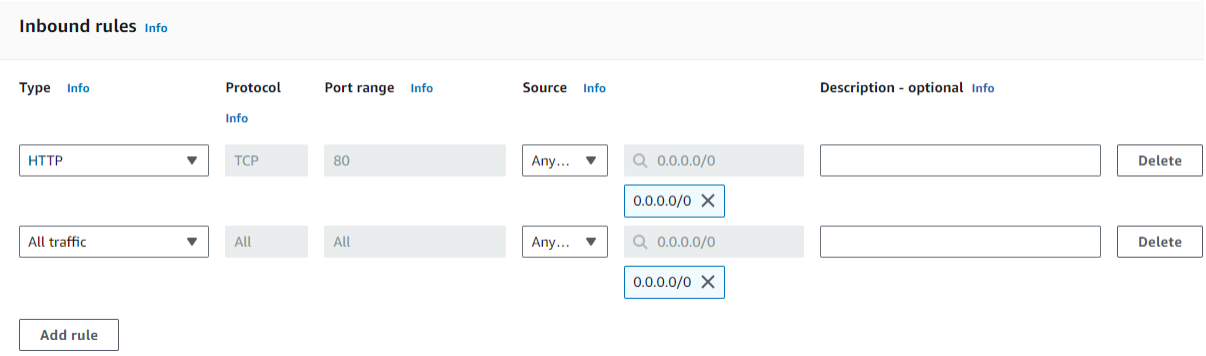


Type – HTTP, Source- 0.0.0.0/0

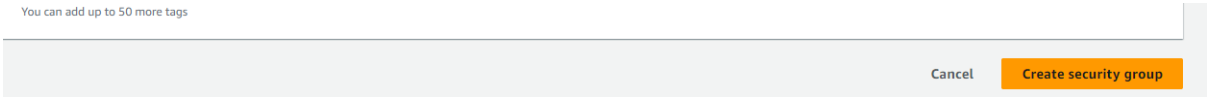


Add rule

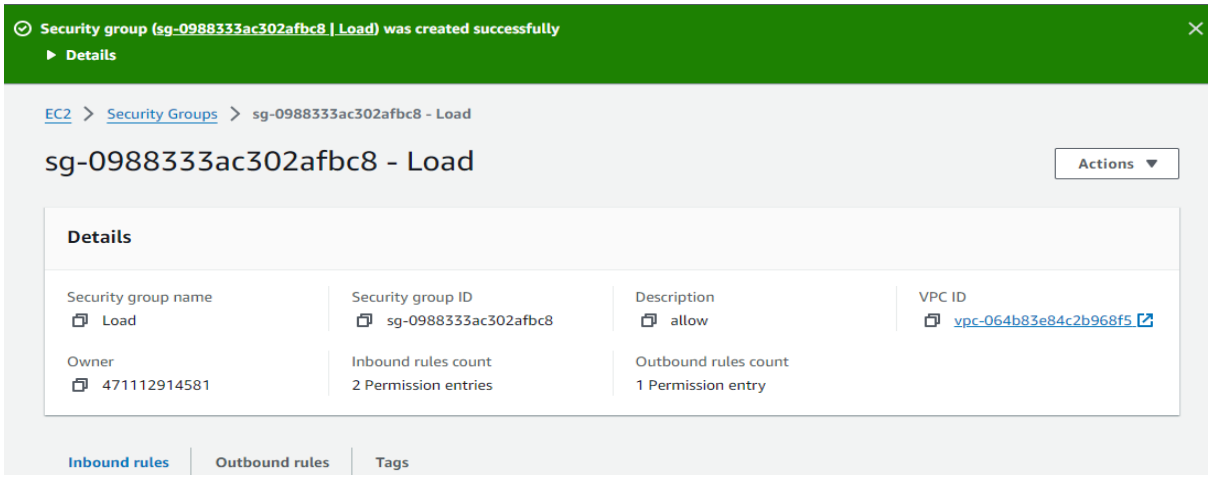
Type- all traffic, source – 0.0.0.0/0



Click on create security group.

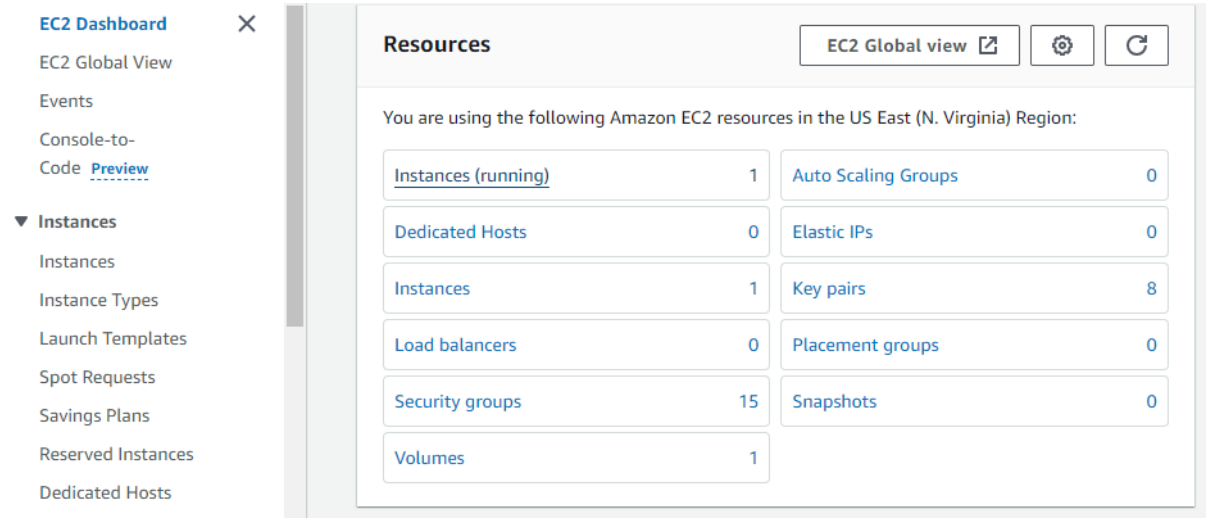


The security group is created now and it shows as follows:-

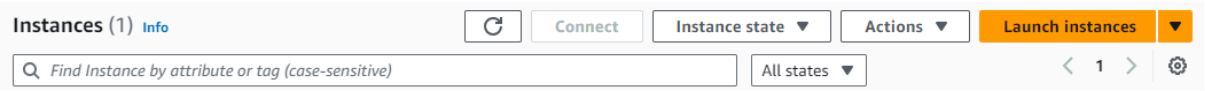


After creating the security group we need to create 3 instances with 3 different availability zones.

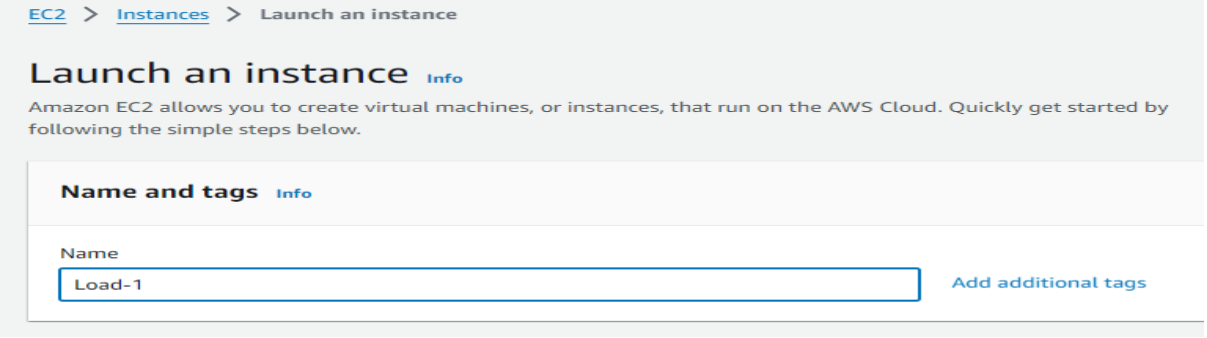
Click on ec2 dashboard and then click on instances.



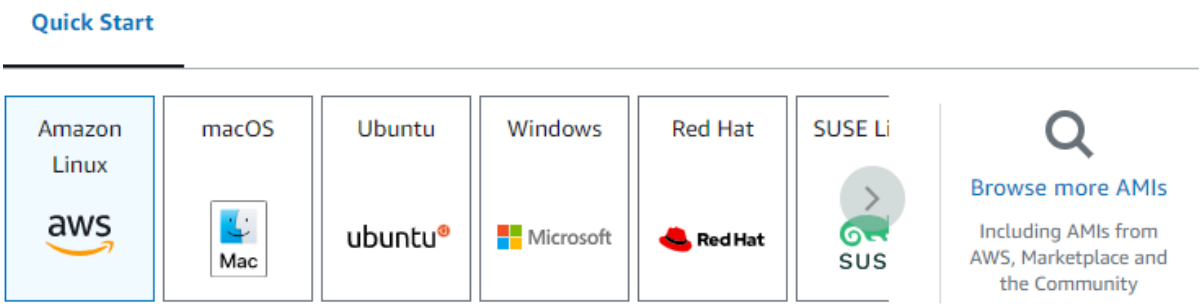
Click on launch instances.



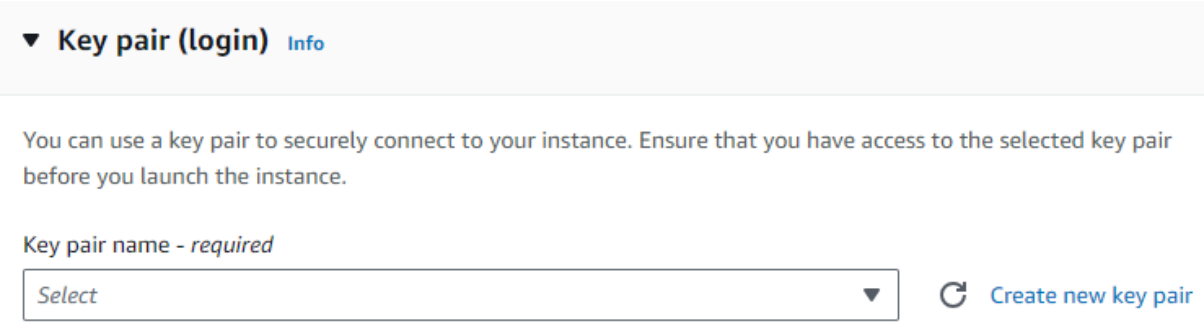
Give instance name



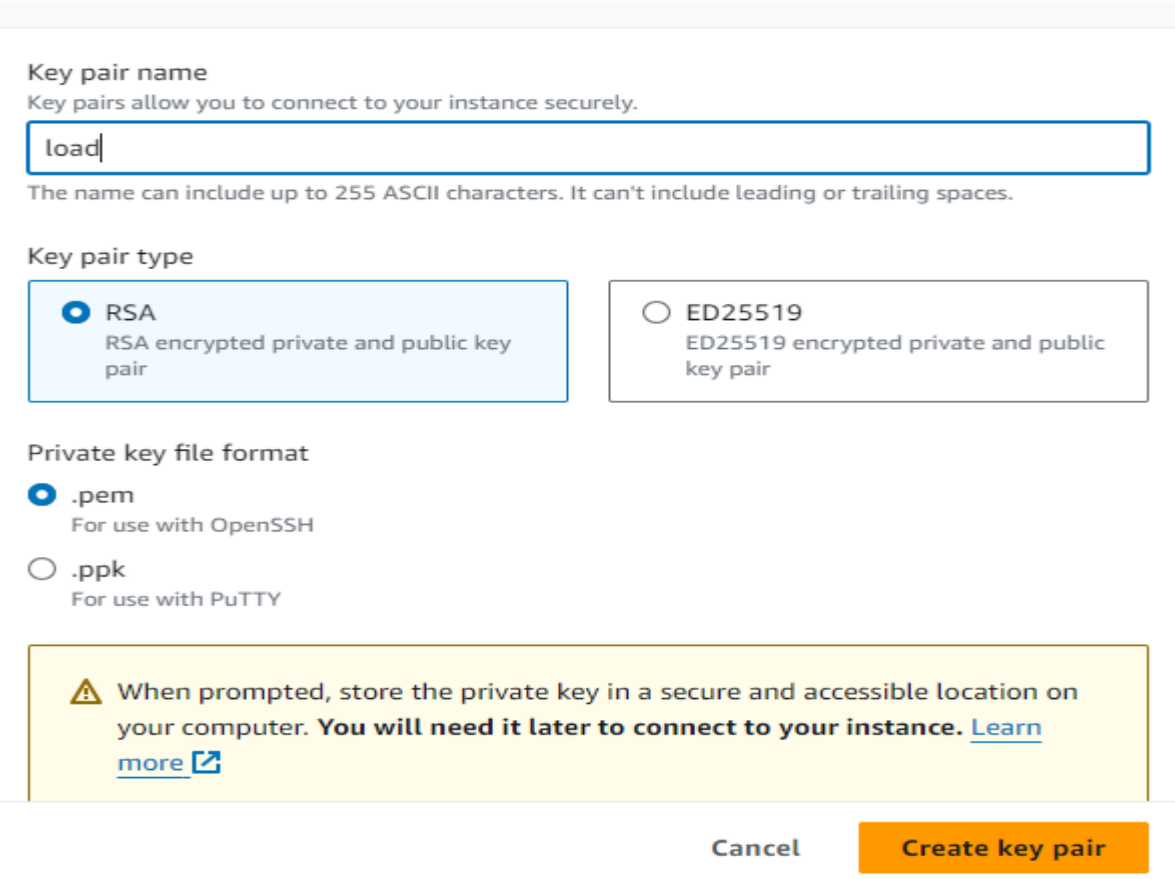
Select amazon linux



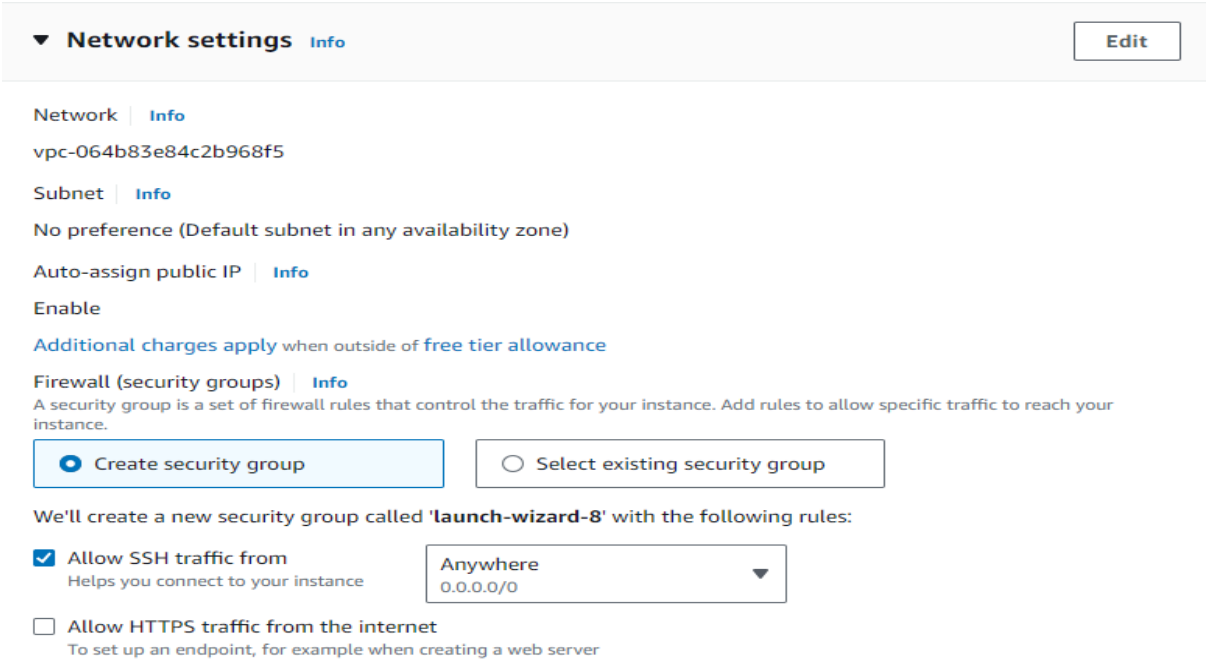
Click on create new key pair



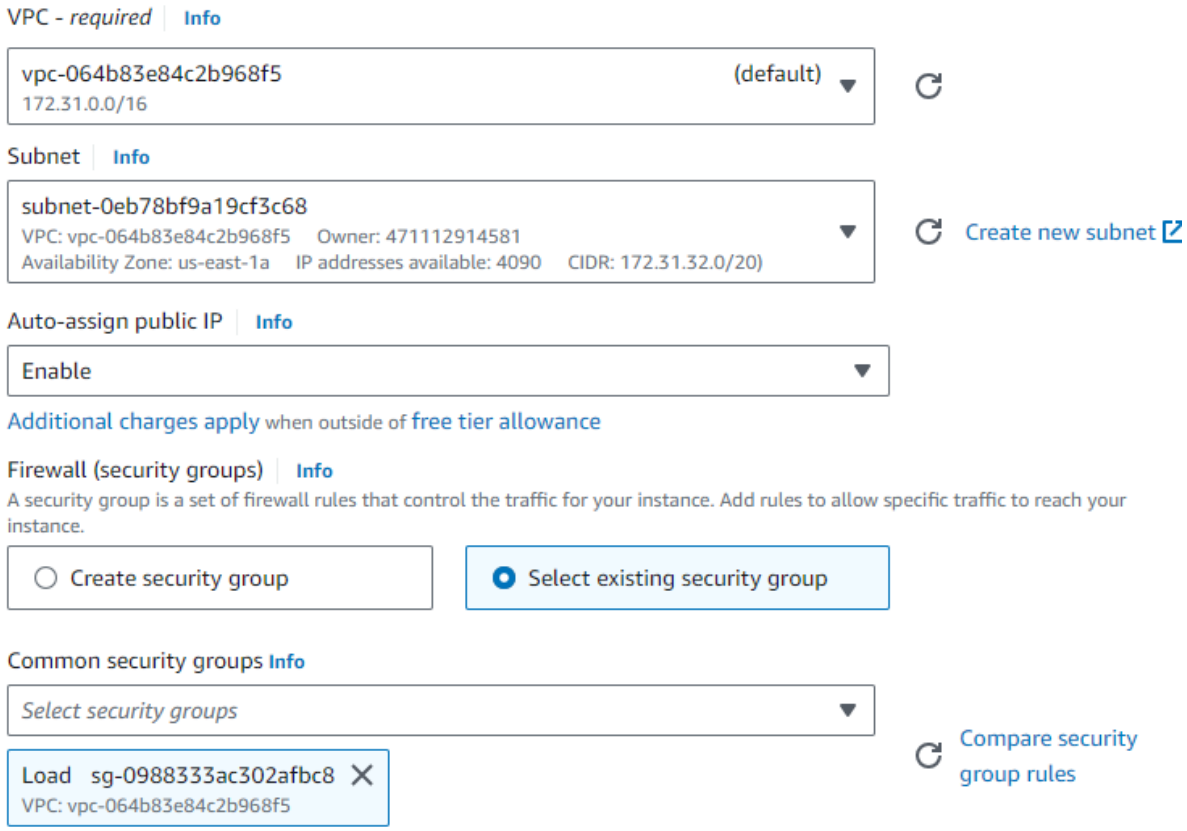
Give the key pair name and then click on create new pair



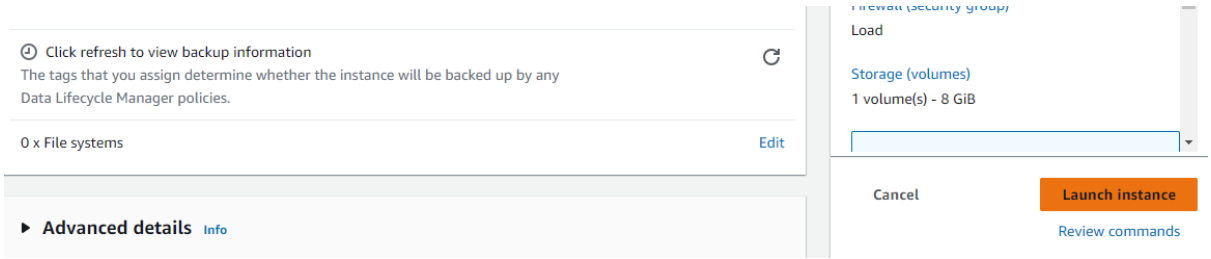
Edit network settings



Choose subnet in the availability zone in us-east-1a > enable > click on select existing security group(select the security which we have created) > select security group.

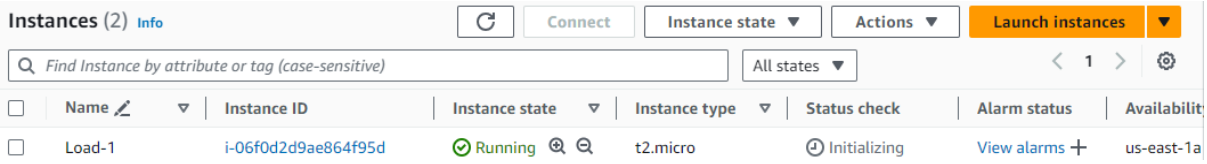


Click on launch instance.

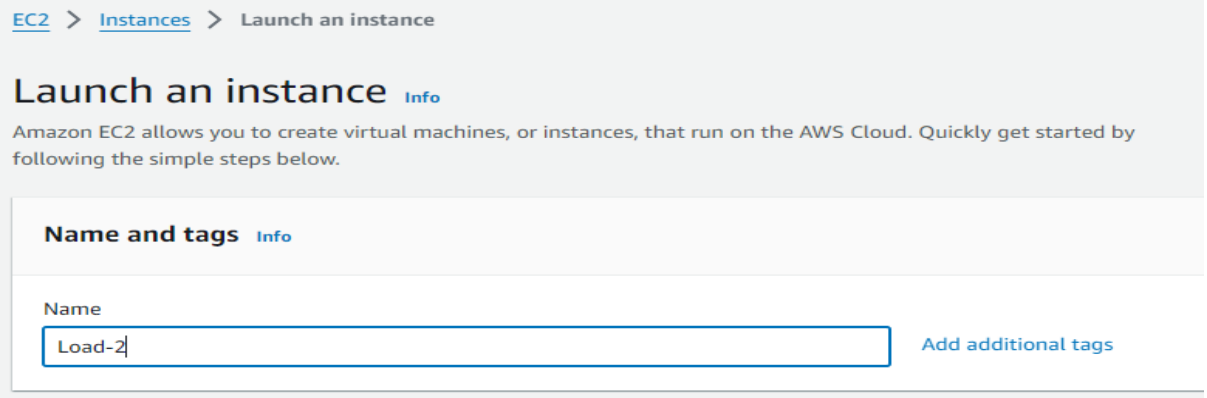


The instance is created successfully.

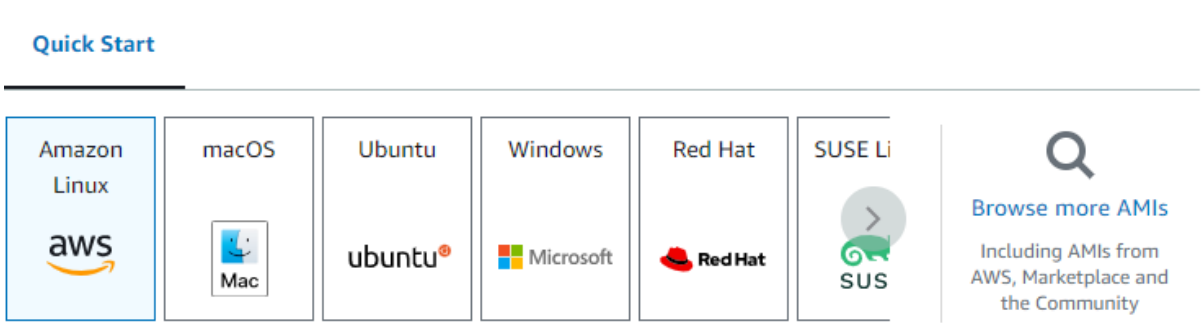
Now again launch new instance



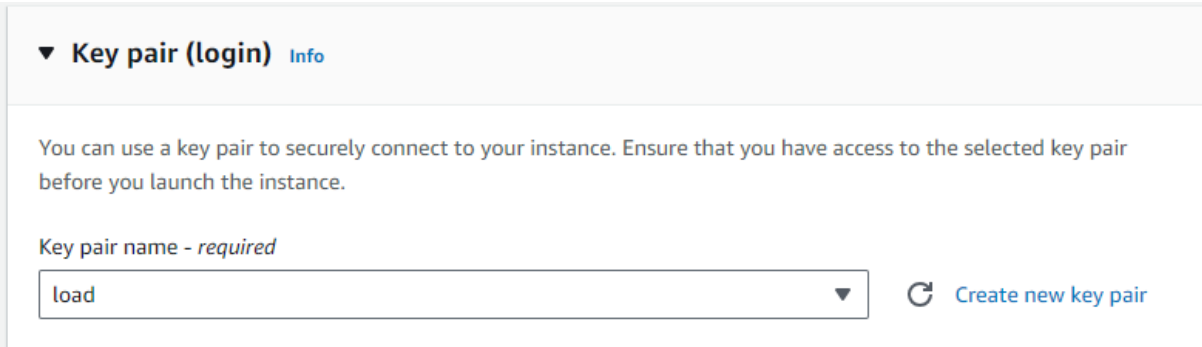
Give name to instance



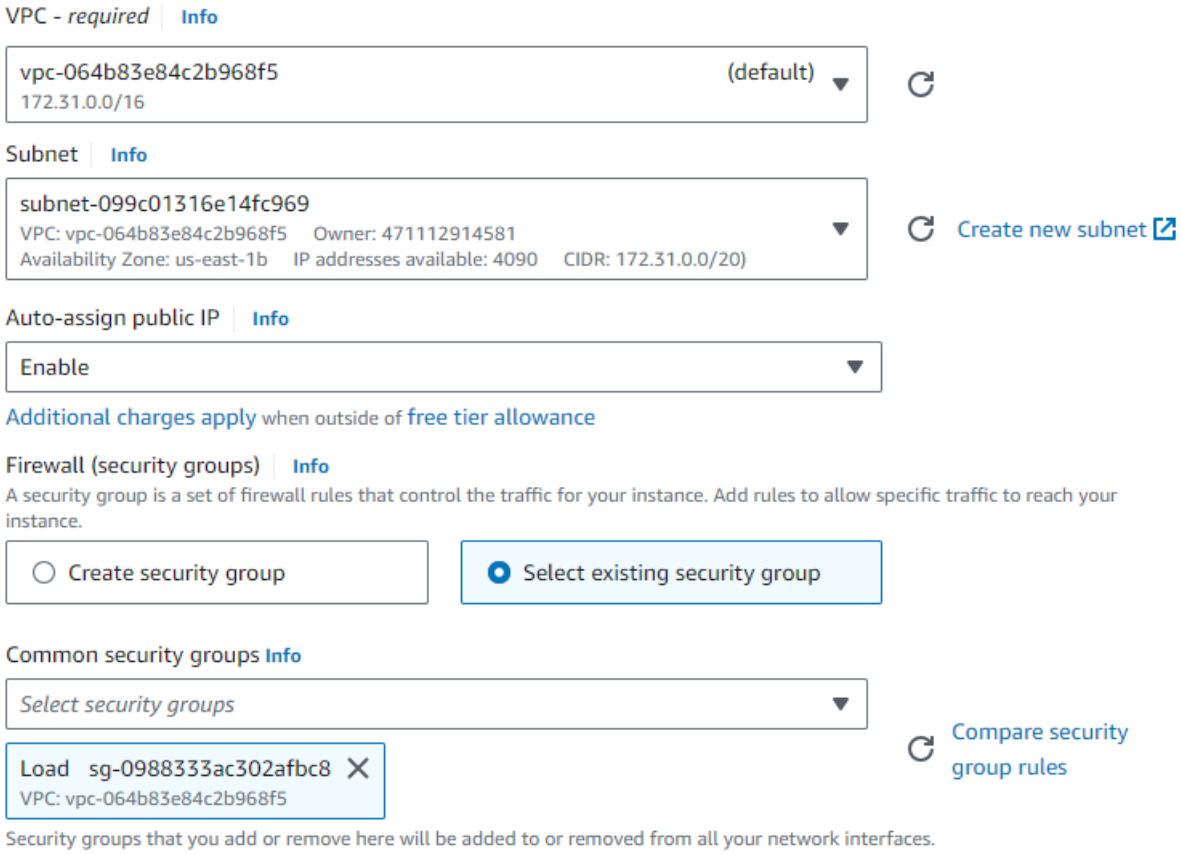
Select amazon linux



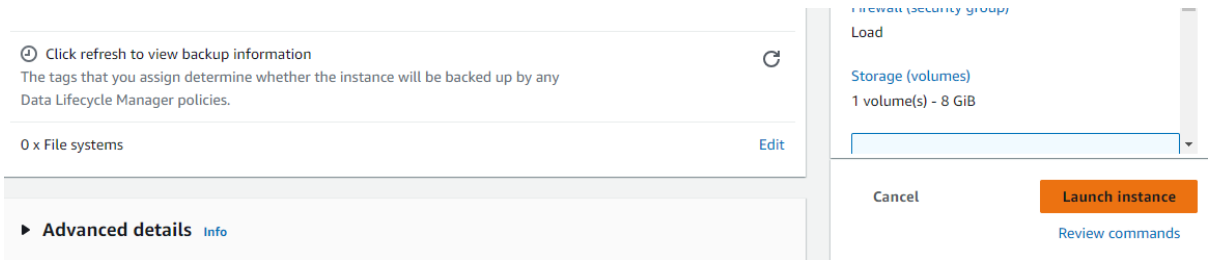
Give key pair name as same as the key pair name for the first instance .



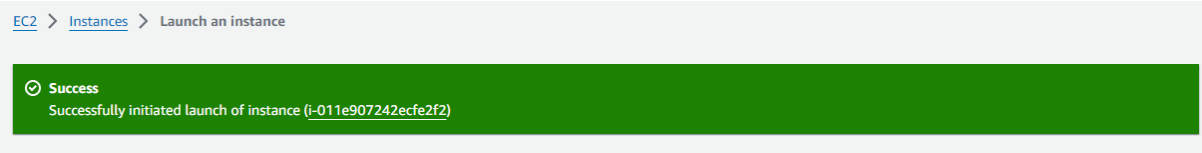
Edit network settings as the instance 1 but create the availability zone in us-east-1b



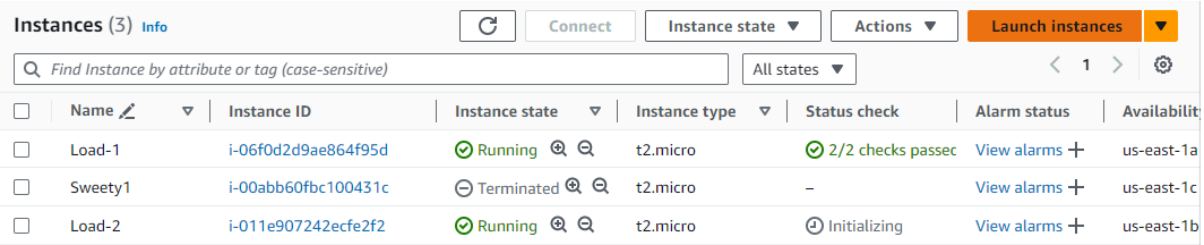
And then click on launch instance



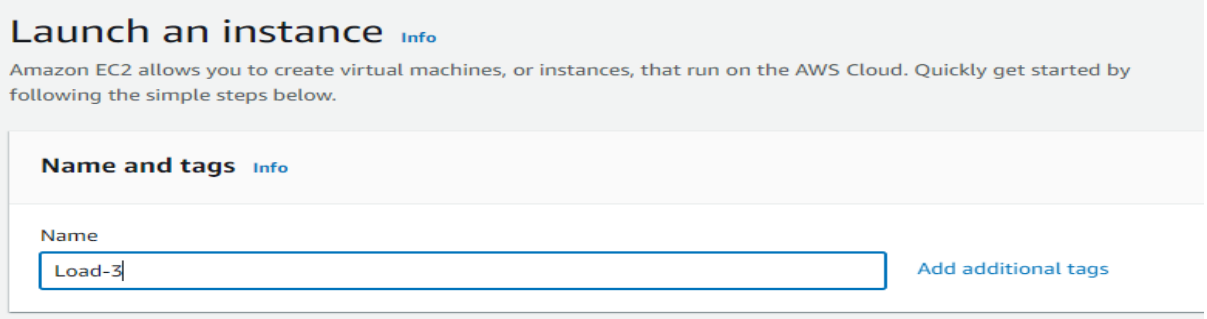
It shows as the instance initiated.



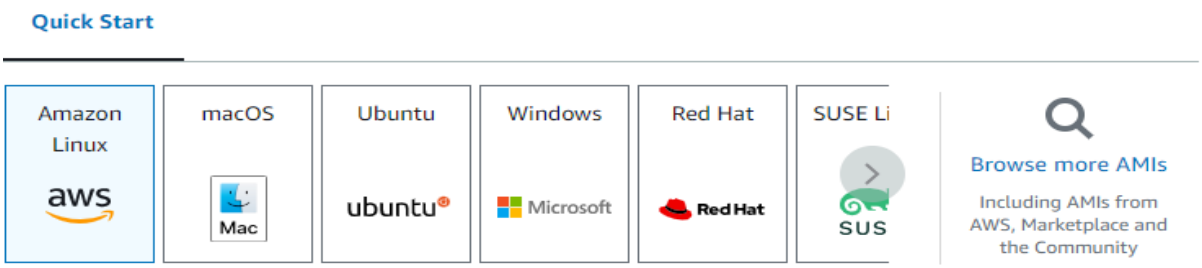
Now for creating the 3rd instance click on launch instances



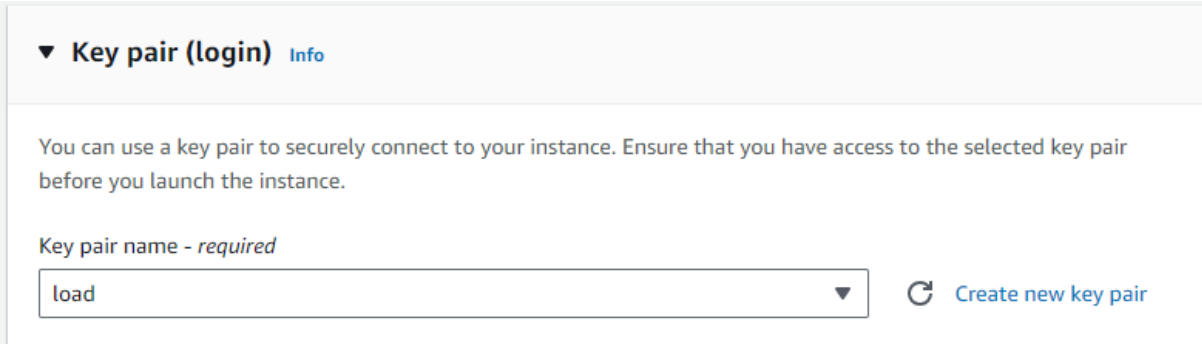
Give name for instance



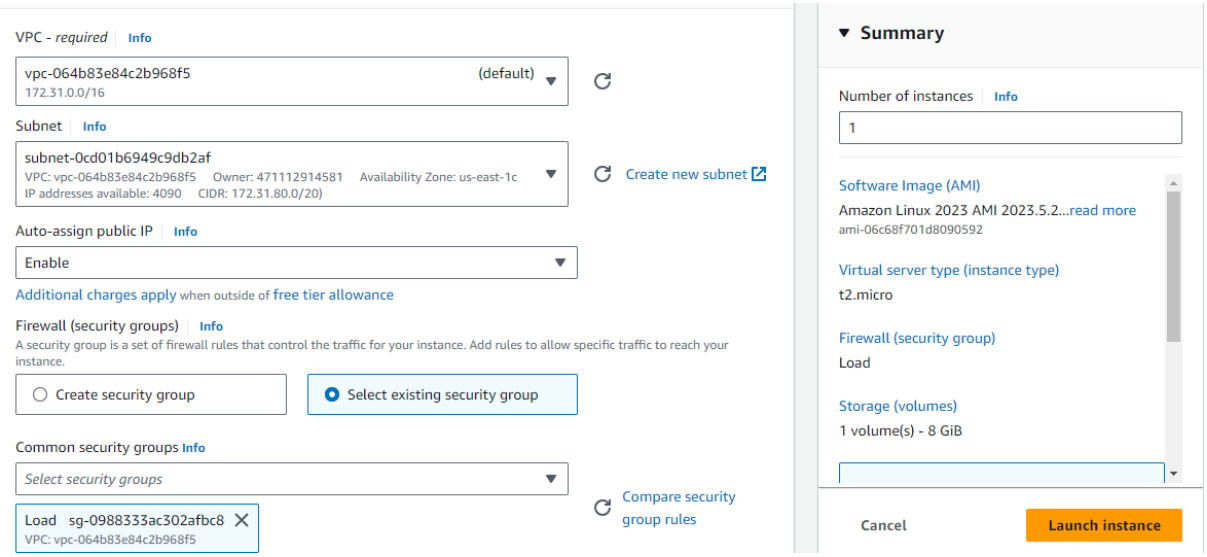
select amazon linux



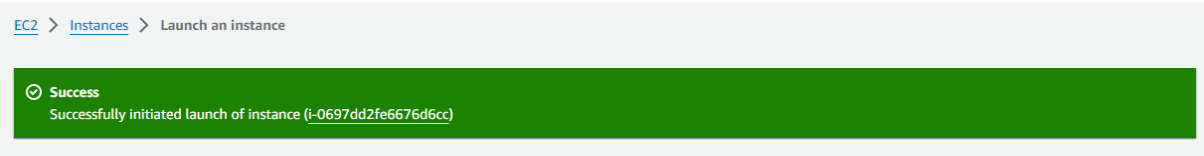
Give the key pair as same as the key pair given for instance 1 and 2



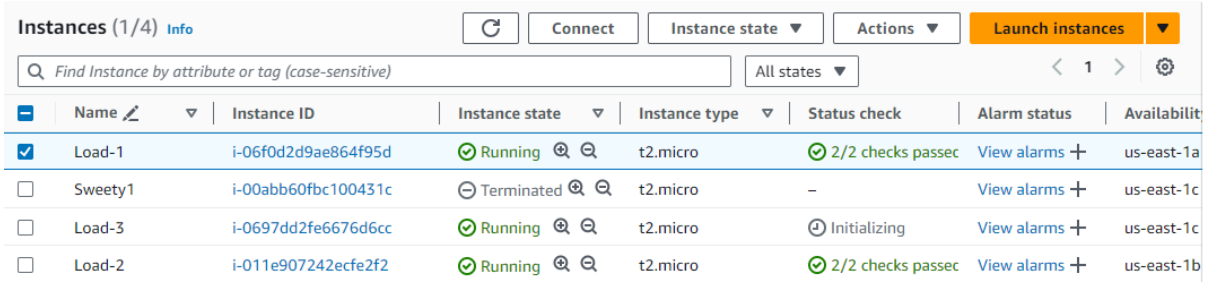
Edit network settings as same as the before instances but create the availability zone in us-east-1c and then click on launch instance.



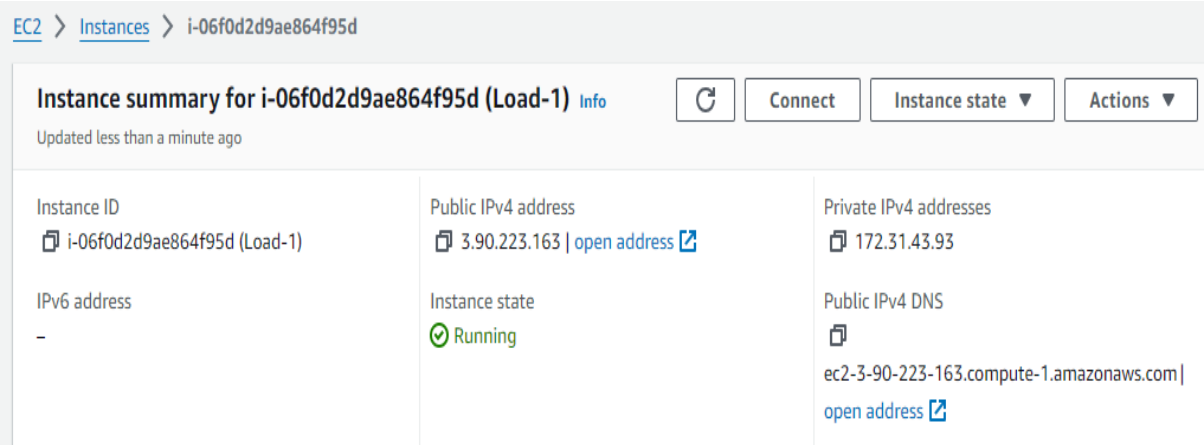
The instance is initiated.



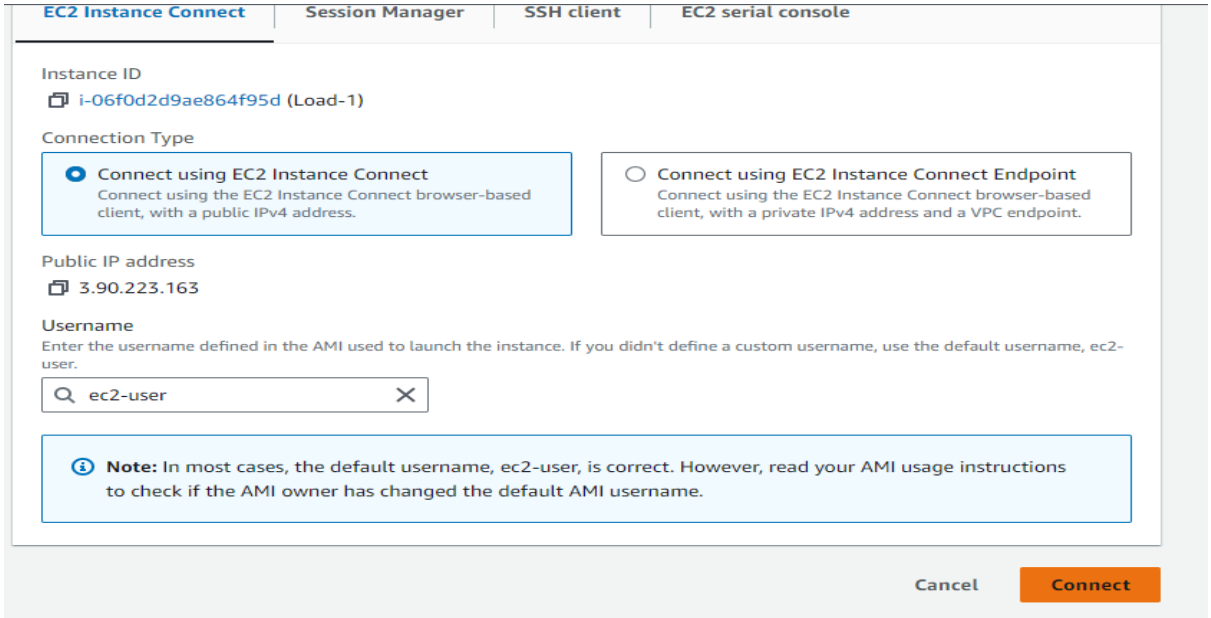
Go to instances > select on the first created instance > click on instance ID



Click on connect

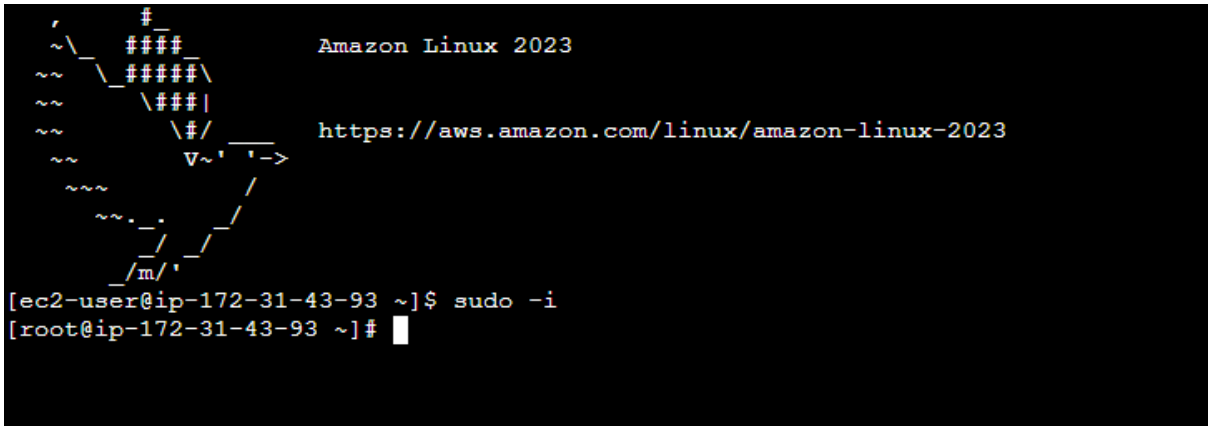


Click on EC2 instance connect and then click on connect



Instance will be connected to amazon linux.

Use command sudo -i for connecting it to the root user.



Update and install nginx by using the commands-

yum update -y

yum install nginx -y

```
[ec2-user@ip-172-31-43-93 ~]$ sudo -i
[root@ip-172-31-43-93 ~]# yum update -y
Last metadata expiration check: 0:08:30 ago on Wed Jul  3 17:10:39 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-43-93 ~]# yum install nginx -y
```

nginx will start to install now

```
 Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    1/7
 Installing       : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    1/7
 Installing       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                       2/7
 Installing       : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                              3/7
 Installing       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                        4/7
 Installing       : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64                                          5/7
 Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                  6/7
 Installing       : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               7/7
 Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               7/7
 Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                  1/7
 Verifying        : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                        2/7
 Verifying        : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                              3/7
 Verifying        : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               4/7
 Verifying        : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64                                          5/7
 Verifying        : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    6/7
 Verifying        : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                       7/7

nstalled:
 generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 libunwind-1.4.0-5.amzn2023.0.2.x86_64
 nginx-1:1.24.0-1.amzn2023.0.2.x86_64              nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64   nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
 nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

omplete!
root@ip-172-31-43-93 ~]#
```

search for the path of nginx by-

cd /usr/share/nginx/html

```
[root@ip-172-31-43-93 ~]# cd /usr/share/nginx/html
[root@ip-172-31-43-93 html]#
```

remove the file index.html by-

rm -rf index.html

and then create file by-

vi index.html

```
[root@ip-172-31-43-93 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-43-93 html]# rm -rf index.html
[root@ip-172-31-43-93 html]# vi index.html
```

Insert and then add data as follows-

```
"index.html" [New]                                                    0,0-1         All
```

```
hii this is load 1
:wq
```

save the file →esc + shift + : wq

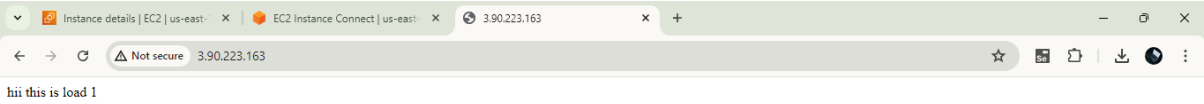
Now restart nginx by using-

systemctl restart nginx

```
[root@ip-172-31-43-93 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-43-93 html]# rm -rf index.html
[root@ip-172-31-43-93 html]# vi index.html
[root@ip-172-31-43-93 html]# systemetl restart nginx
-bash: systemetl: command not found
[root@ip-172-31-43-93 html]# systemetl restart nginx
-bash: systemetl: command not found
[root@ip-172-31-43-93 html]# systemet1 restart nginx
-bash: systemet1: command not found
[root@ip-172-31-43-93 html]# systemctl restart nginx
[root@ip-172-31-43-93 html]#
```

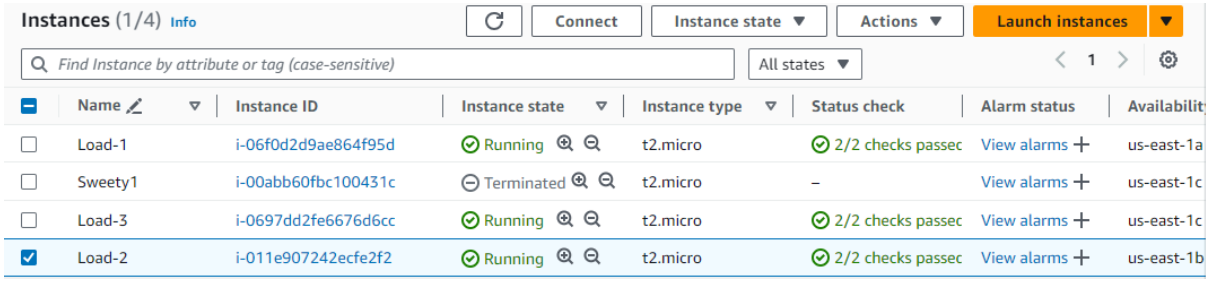
go back to instance ID and copy the public IPv4 address

The data will be shown which is inserted in the file index.html

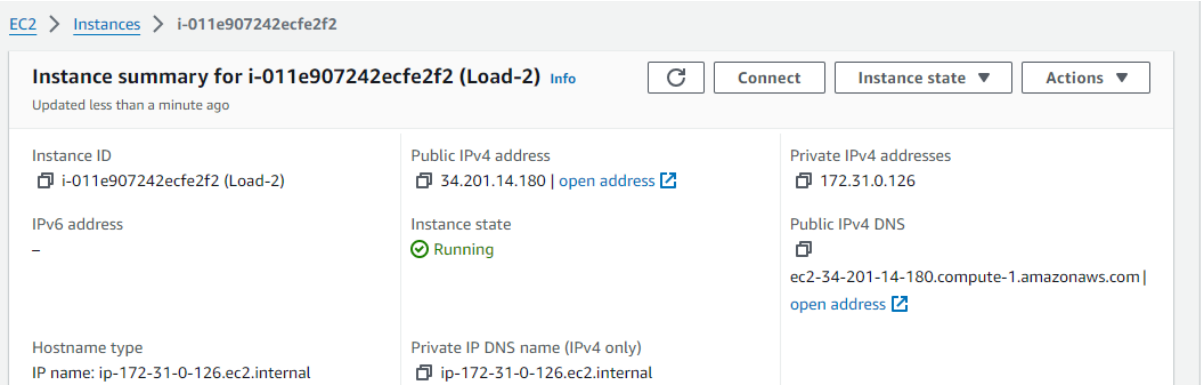


Do the same for instance 2 and instance 3

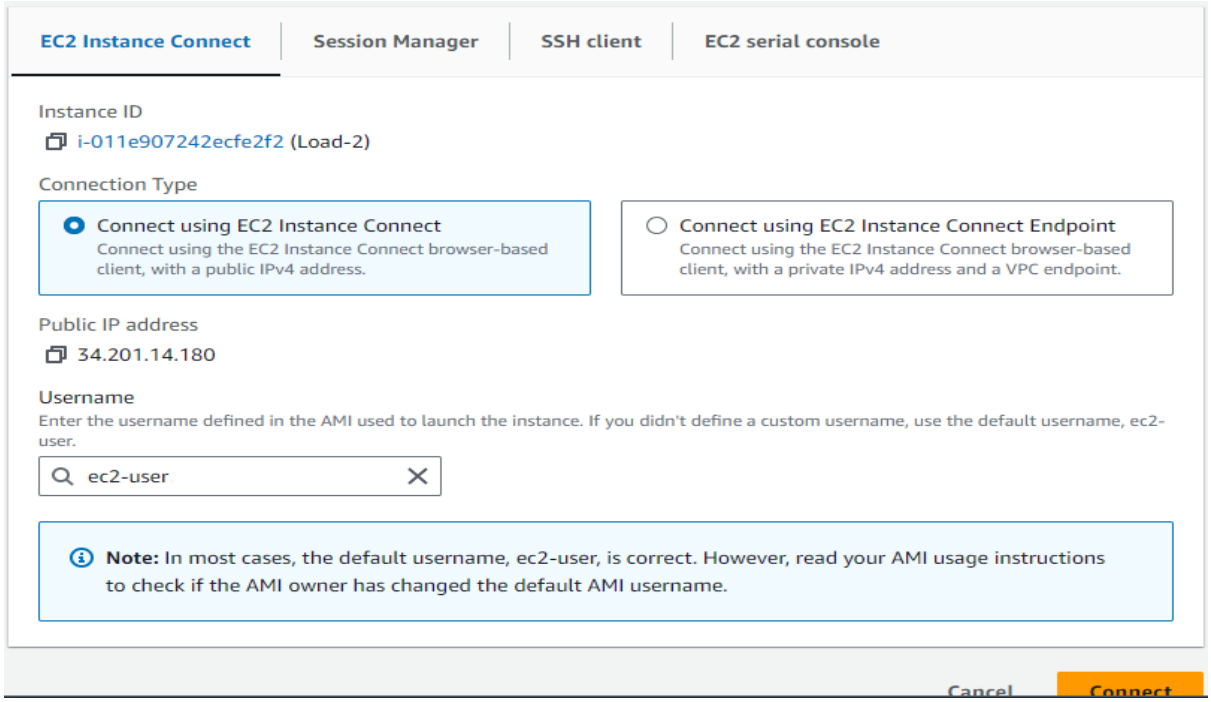
Click on the instance ID of 2nd instance



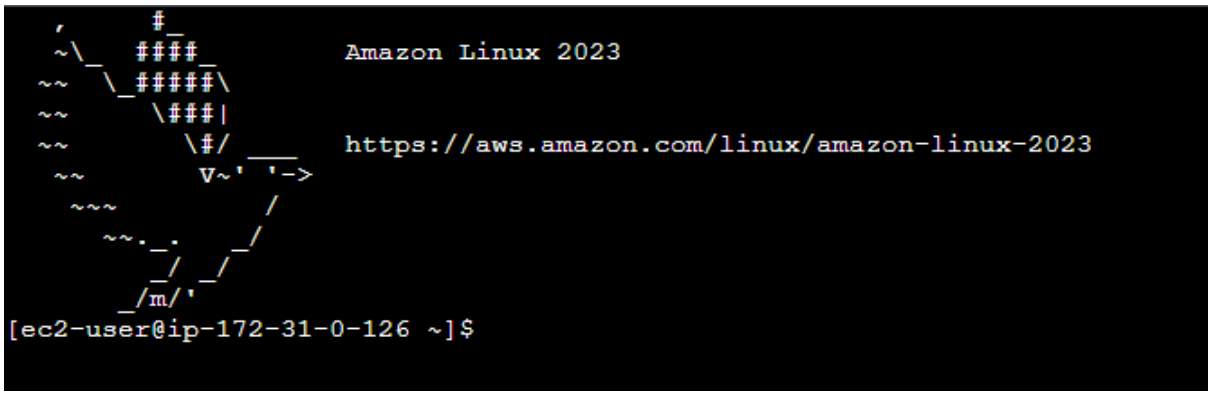
Click on connect



Click on EC2 instance connect and click on connect



It will take to aws linux.



For connecting to root user use command sudo -i

Install nginx

```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Jul  3 17:27:31 2024 from 18.206.107.28
[ec2-user@ip-172-31-0-126 ~]$ sudo -i
[root@ip-172-31-0-126 ~]# yum update -y
Last metadata expiration check: 0:14:20 ago on Wed Jul  3 17:14:19 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-0-126 ~]# yum install nginx -y
```

It will start installing nginx

```
  Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    1/1
  Installing       : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    1/7
  Installing       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                       2/7
  Installing       : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                              3/7
  Installing       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                        4/7
  Installing       : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64                                          5/7
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                  6/7
  Installing       : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               7/7
  Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               7/7
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                  1/7
  Verifying        : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                        2/7
  Verifying        : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                              3/7
  Verifying        : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               4/7
  Verifying        : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64                                          5/7
  Verifying        : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    6/7
  Verifying        : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                       7/7

Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 libunwind-1.4.0-5.amzn2023.0.2.x86_64
  nginx-1:1.24.0-1.amzn2023.0.2.x86_64              nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64   nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-172-31-0-126 ~]#
```

Check for nginx path

Remove index.html file

```
[root@ip-172-31-0-126 ~]# cd /usr/share/nginx/html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]# rm -rf index.html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]#
```

Create a file named with index.html

```
[root@ip-172-31-0-126 html]# vi index.html
```

Insert data and save the file →esc + shift + : wq

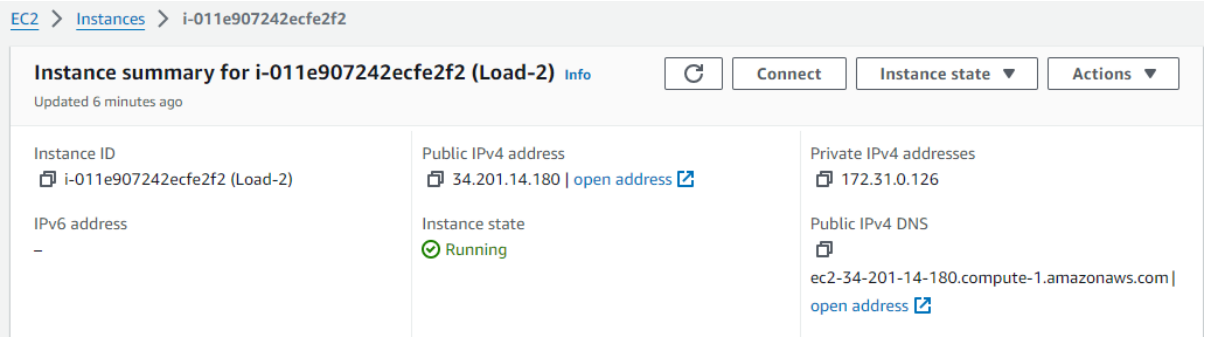
```
"index.html" [New]                                                    0,0-1         All
```

```
hii hi this load 2
:wq
```

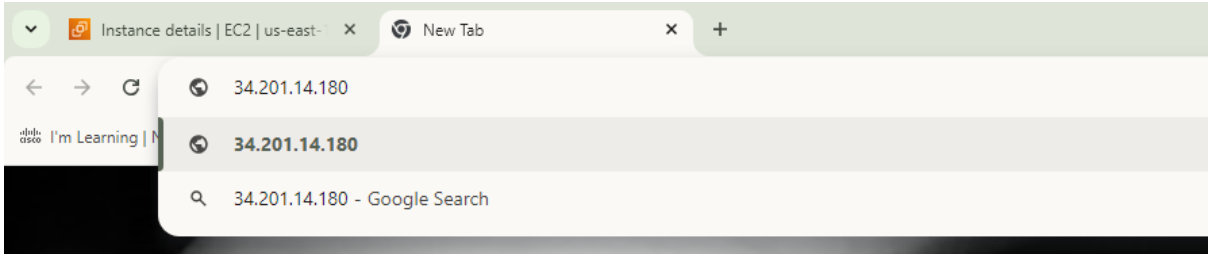
Restart nginx now

```
[root@ip-172-31-0-126 ~]# cd /usr/share/nginx/html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]# rm -rf index.html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]# vi index.html
[root@ip-172-31-0-126 html]# systemctl restart nginx
[root@ip-172-31-0-126 html]#
```

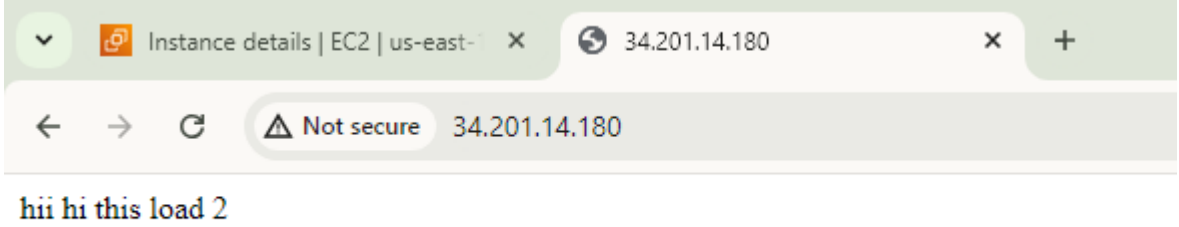
Go to instance ID and copy the Public IPv4 address



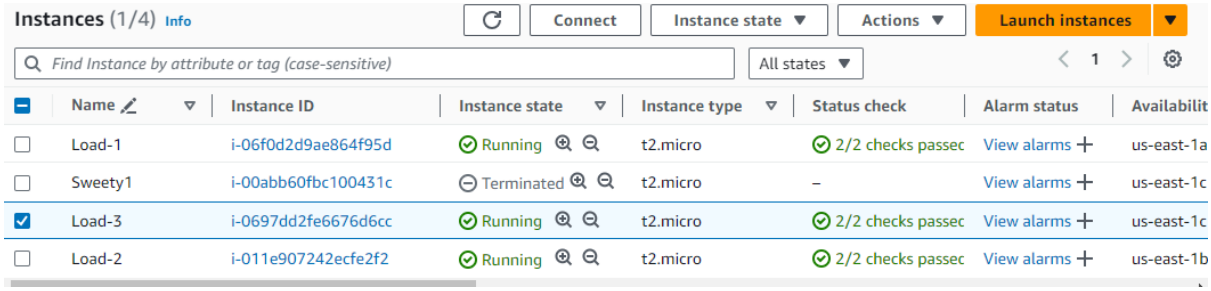
Go to chrome and paste the IPv4 address



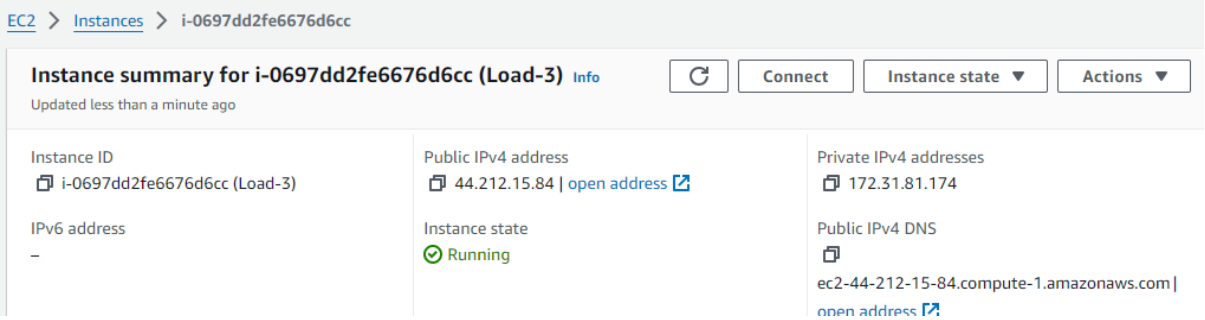
The data will be shown which is inserted in the file index.html



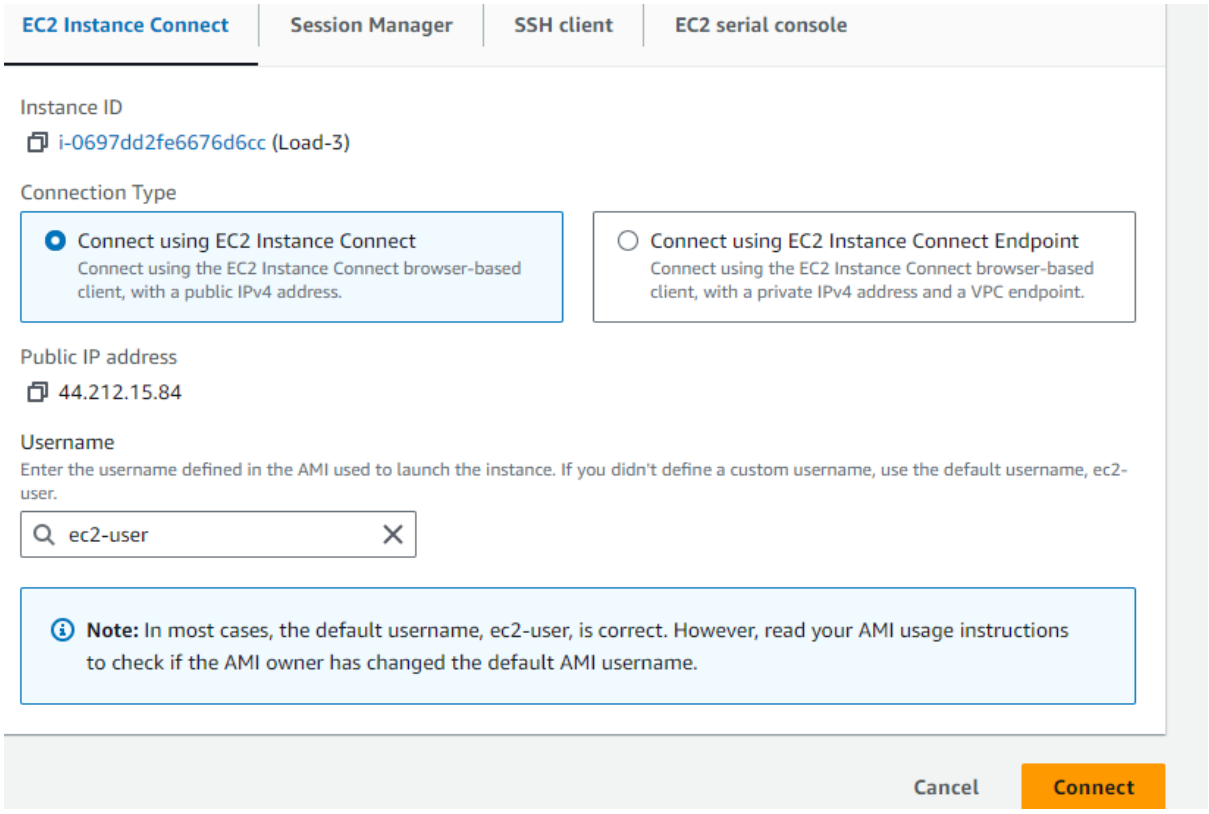
Click on the instance ID of 3rd instance



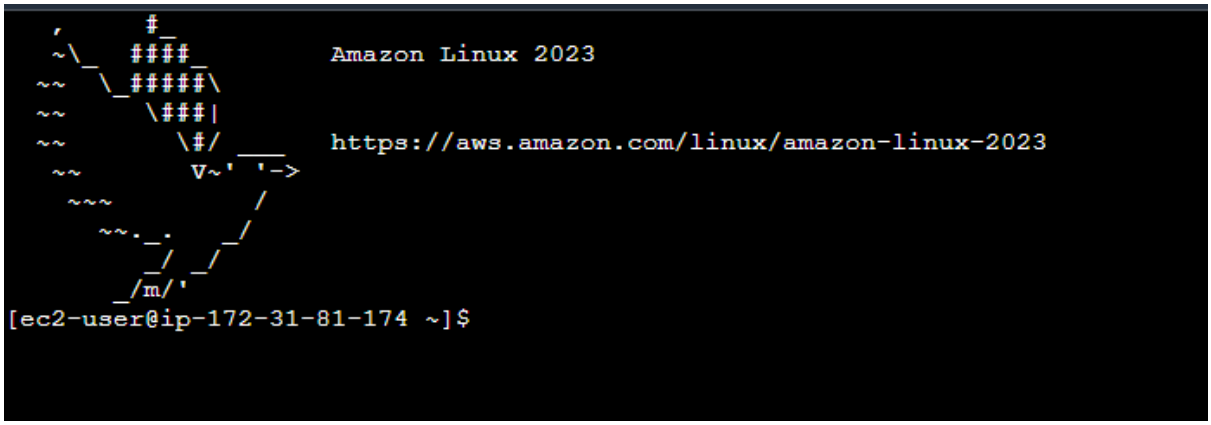
Click on connect



Click on EC2 instance connect and click on connect



It will take to aws linux



For connecting to root user use command sudo -i

```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-81-174 ~]$ sudo -i
[root@ip-172-31-81-174 ~]# yum update -y
Last metadata expiration check: 0:20:44 ago on Wed Jul  3 17:17:01 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-81-174 ~]# yum install nginx -y
```

Install nginx

It will start installing nginx

```
  Installing       : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    1/7
  Installing       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                       2/7
  Installing       : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                              3/7
  Installing       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                        4/7
  Installing       : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64                                          5/7
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                  6/7
  Installing       : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               7/7
  Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               7/7
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                  1/7
  Verifying        : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                        2/7
  Verifying        : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                              3/7
  Verifying        : nginx-1:1.24.0-1.amzn2023.0.2.x86_64                                               4/7
  Verifying        : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64                                          5/7
  Verifying        : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch                                    6/7
  Verifying        : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                       7/7

Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 libunwind-1.4.0-5.amzn2023.0.2.x86_64
  nginx-1:1.24.0-1.amzn2023.0.2.x86_64              nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64   nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-172-31-81-174 ~]#
```

i-0697dd2fe6676d6cc (Load-3)

PublicIPs: 44.212.15.84 PrivateIPs: 172.31.81.174

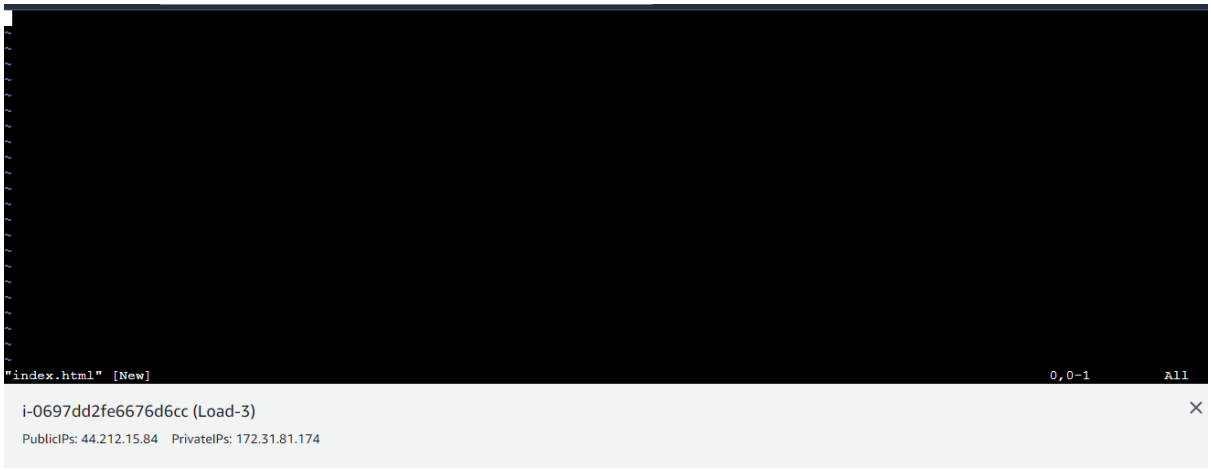
Check for nginx path

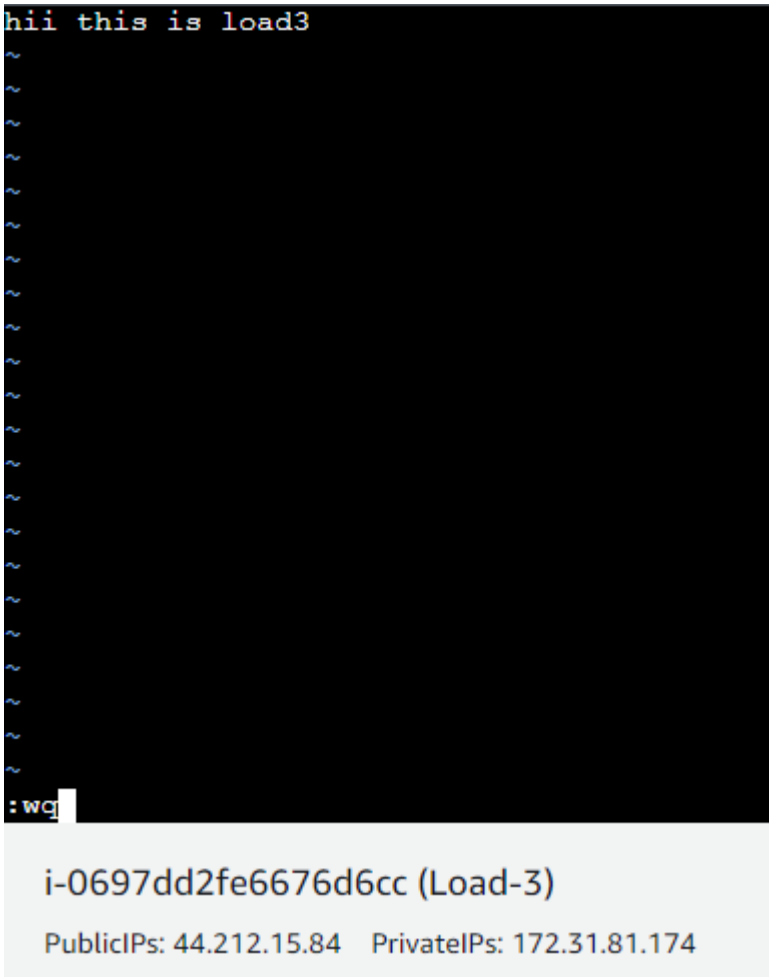
Remove index.html file

Create a file named with index.html

```
[root@ip-172-31-81-174 ~]# cd /usr/share/nginx/html
[root@ip-172-31-81-174 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-81-174 html]# rm -rf index.html
[root@ip-172-31-81-174 html]# ls
404.html  50x.html  icons  nginx-logo.png  poweredby.png
[root@ip-172-31-81-174 html]# vi index.html
```

Insert data and save the file →esc + shift + : wq

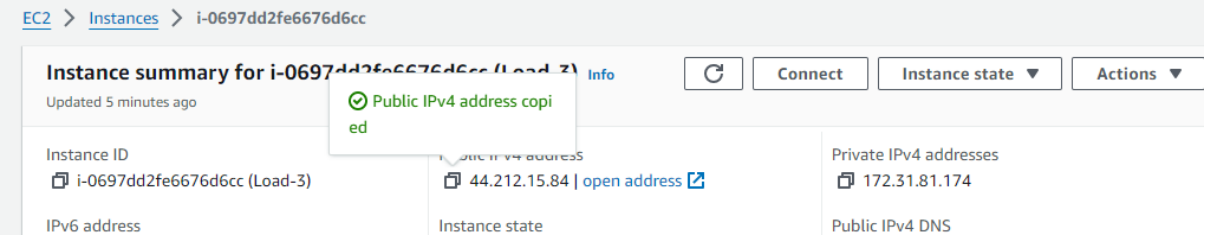




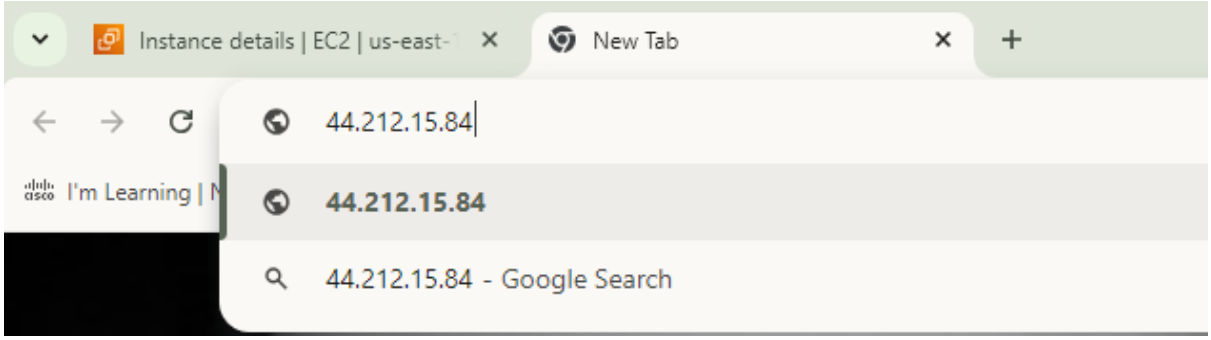
Restart nginx now

```
[root@ip-172-31-81-174 ~]# cd /usr/share/nginx/html
[root@ip-172-31-81-174 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-81-174 html]# rm -rf index.html
[root@ip-172-31-81-174 html]# ls
404.html  50x.html  icons  nginx-logo.png  poweredby.png
[root@ip-172-31-81-174 html]# vi index.html
[root@ip-172-31-81-174 html]# systemctl restart nginx
[root@ip-172-31-81-174 html]#
```

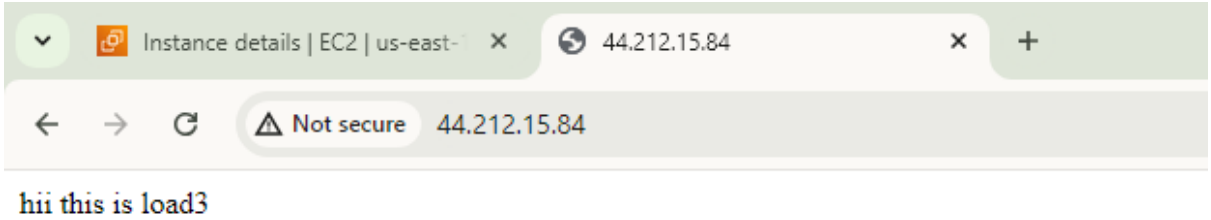
Go to instance ID and copy the Public IPv4 address



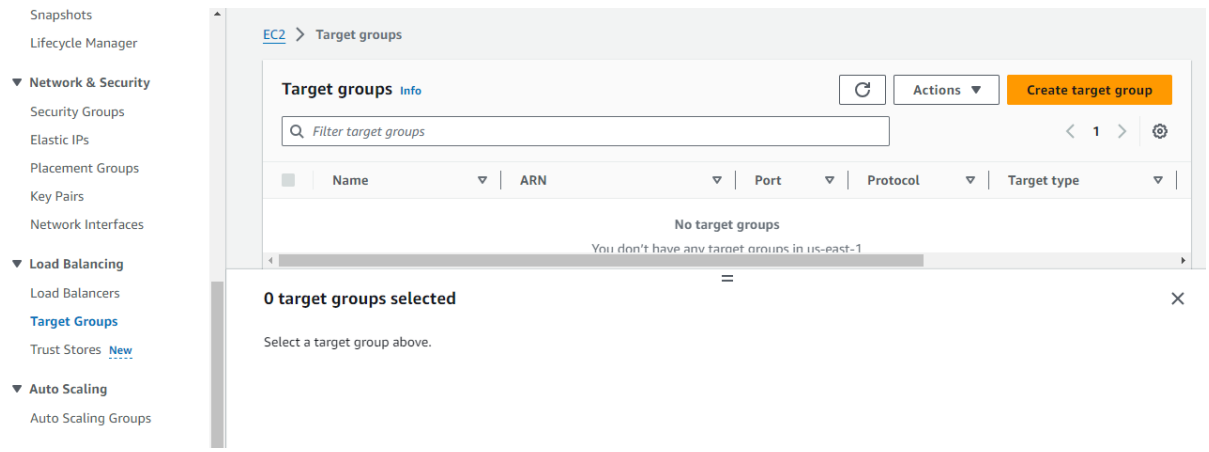
Go to chrome and paste the IPv4 address



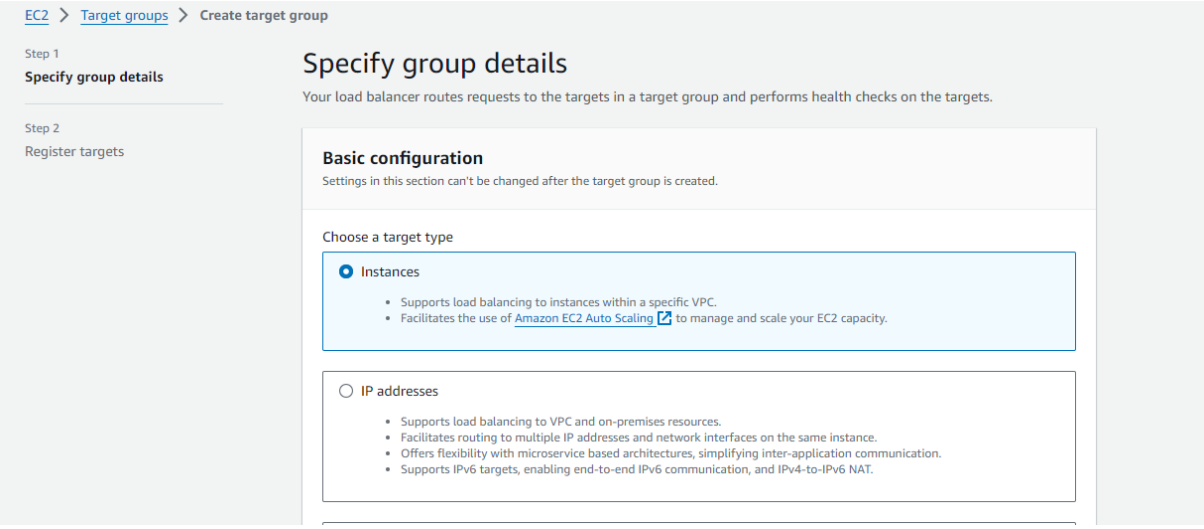
The data will be shown which is inserted in the file index.html



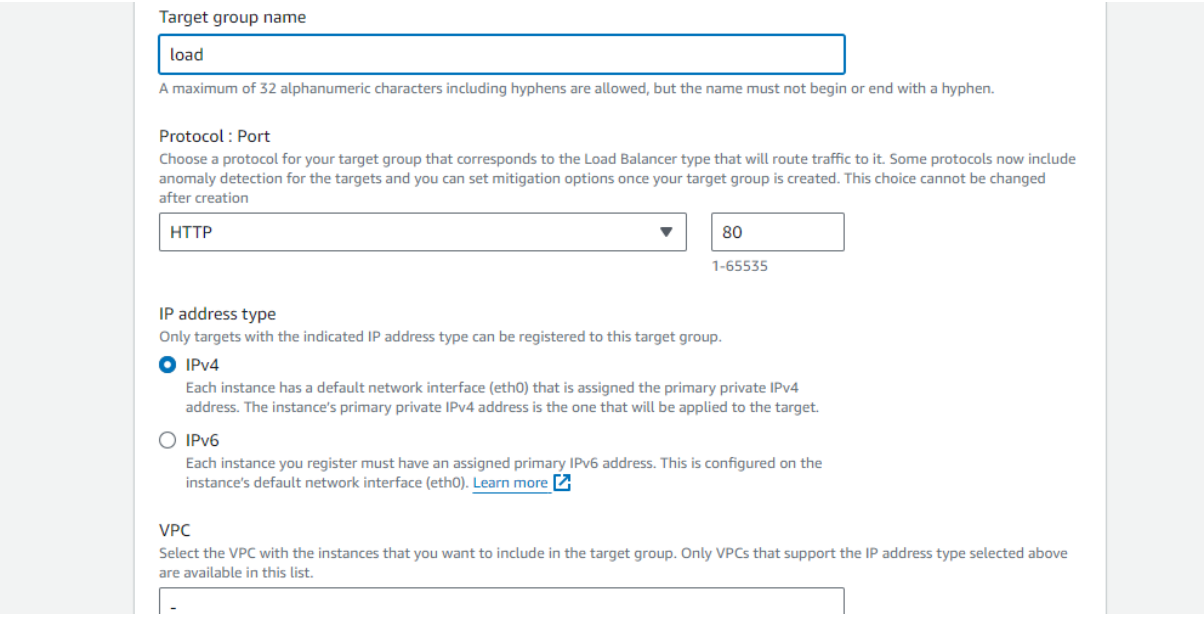
Click on target groups which is under load balancing option and then click on create target group.



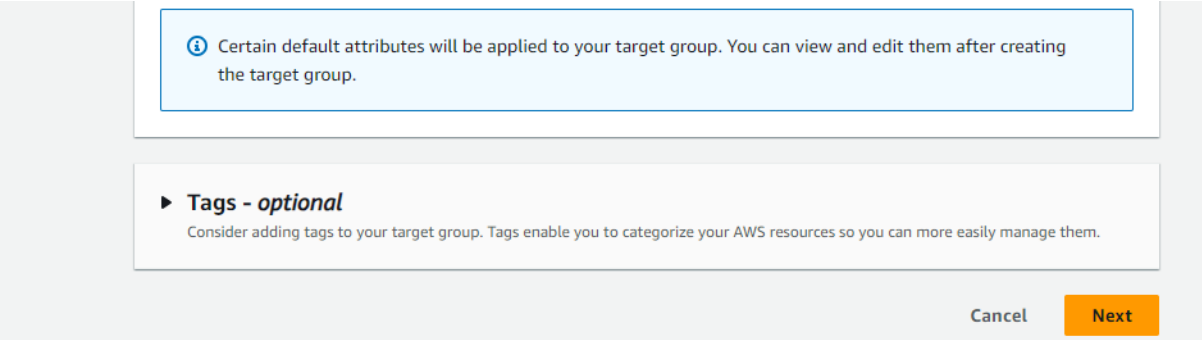
Target type will be chosen as instances by default



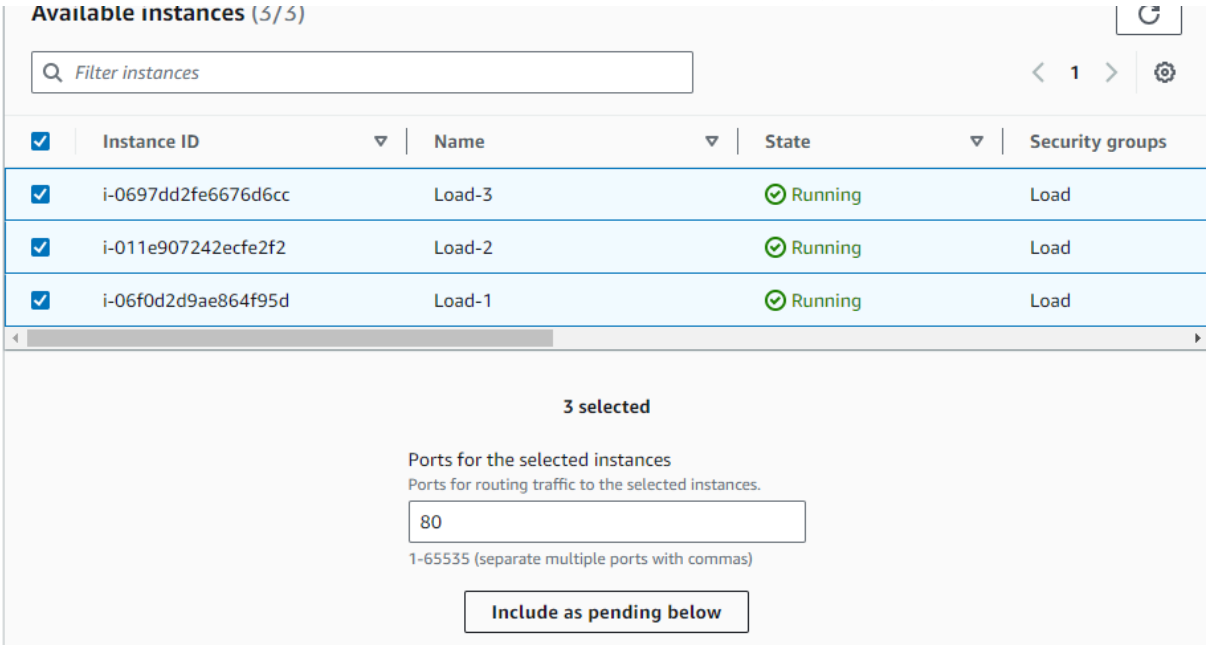
Give target group name



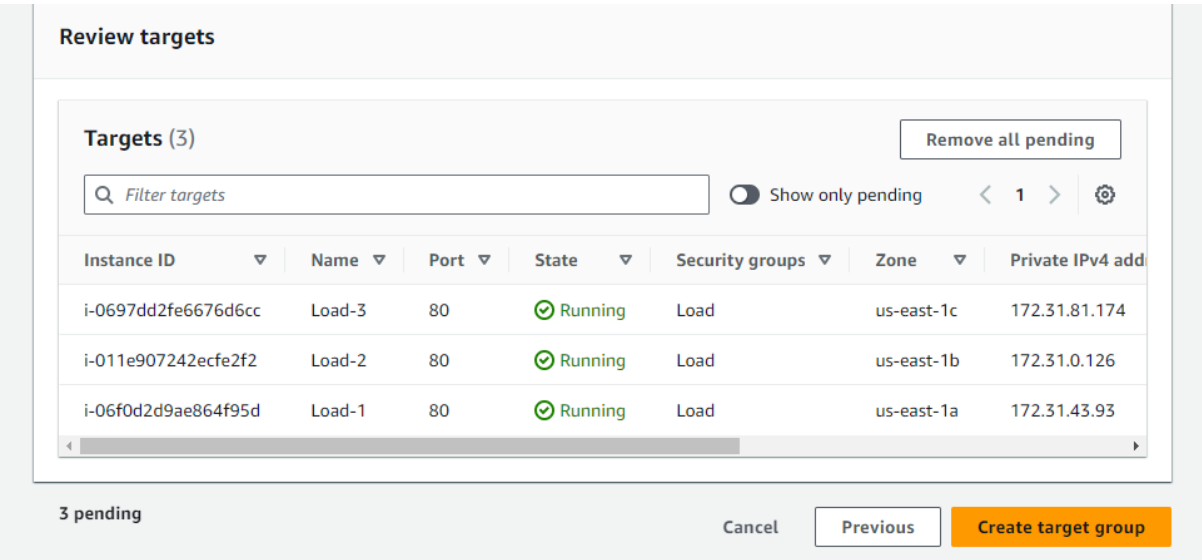
and then click on next



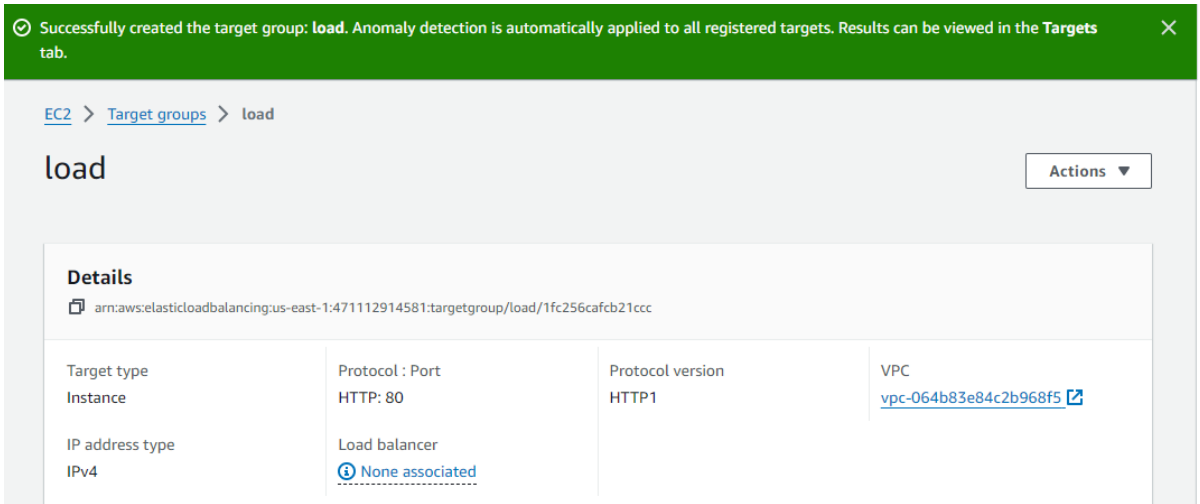
Select all the three 3 instances and click on include as pending below



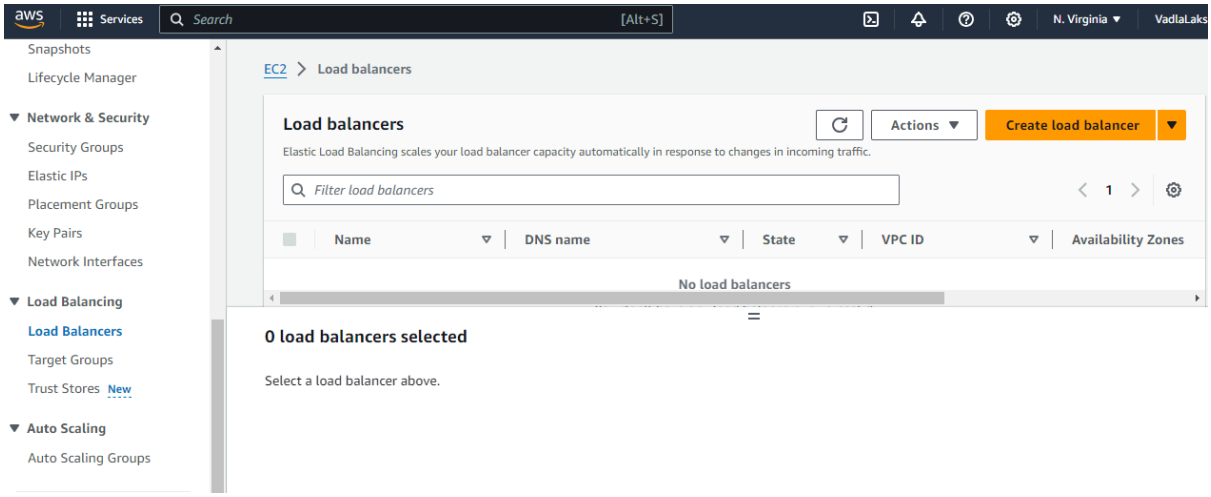
It will as follows and click on create target group



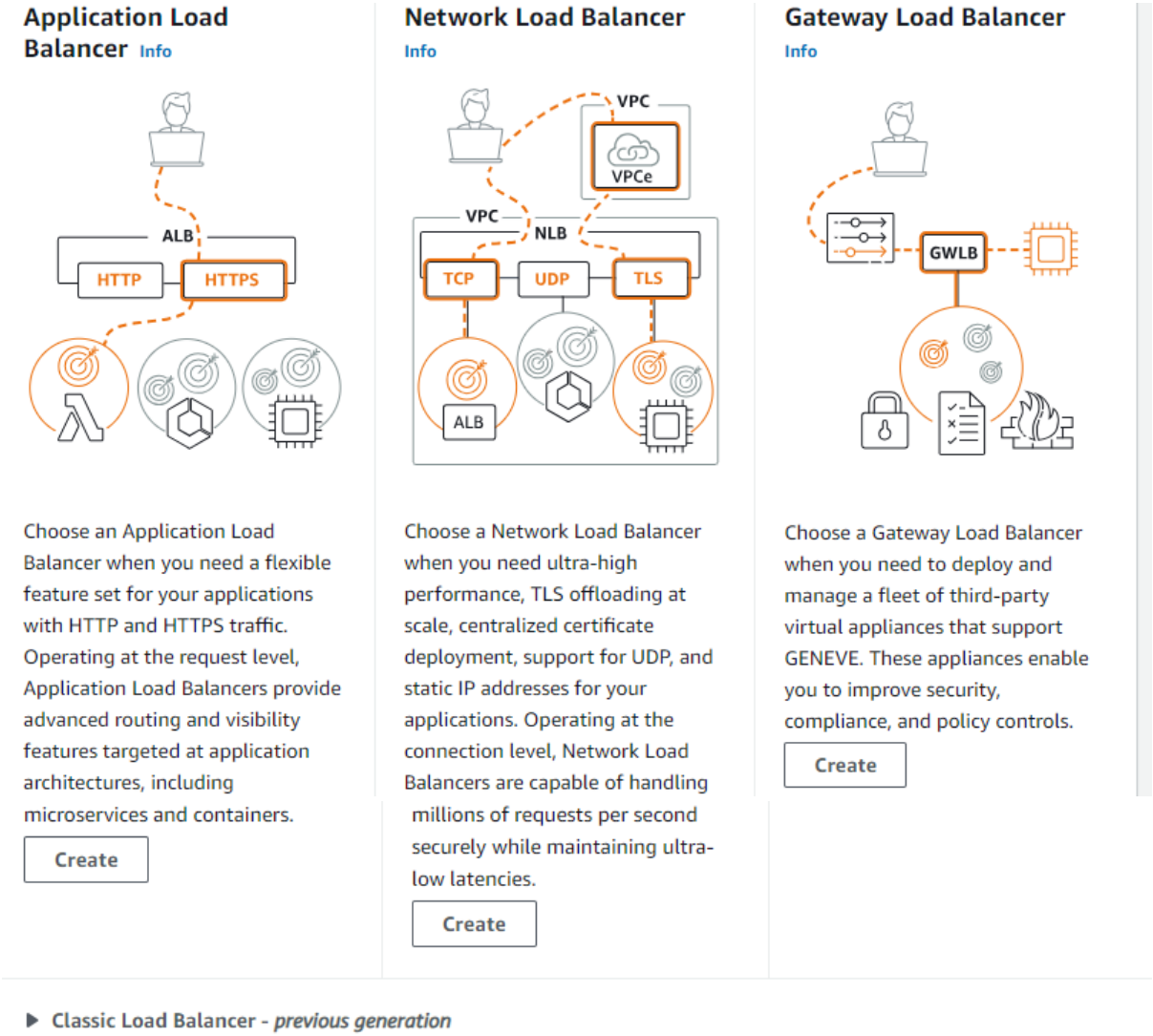
Now the target group is created.



Now go to load balancer which is under load balancing and then click on create load balancer.



Click on create option which is under application load balancer.



Give load balancer name

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

Load

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme | Info

Scheme can't be changed after the load balancer is created.

- (●) Internet-facing
  An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
- ( ) Internal
  An internal load balancer routes requests from clients to targets using private IP addresses. Compatible with the **IPv4** and **Dualstack** IP address types.

Load balancer IP address type | Info

Select the type of IP addresses that your subnets use. Public IPv4 addresses have an additional cost.

- (●) IPv4
  Includes only IPv4 addresses.
- ( ) Dualstack
  Includes IPv4 and IPv6 addresses.
- ( ) Dualstack without public IPv4
  Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

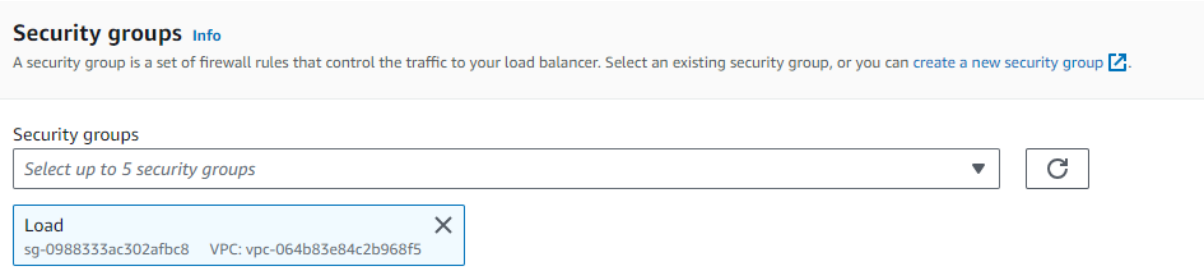
Select the three availability zones where the 3 instances are created

Mappings | Info

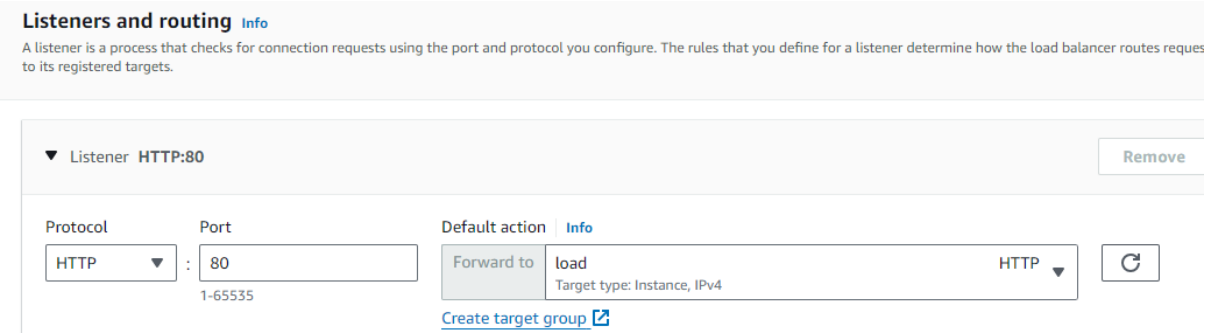
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

- [x] **us-east-1a (use1-az6)**
  Subnet
  subnet-0eb78bf9a19cf3c68
  IPv4 address
  Assigned by AWS
- [x] **us-east-1b (use1-az1)**
  Subnet
  subnet-099c01316e14fc969
  IPv4 address
  Assigned by AWS
- [x] **us-east-1c (use1-az2)**
  Subnet
  subnet-0cd01b6949c9db2af
  IPv4 address
  Assigned by AWS

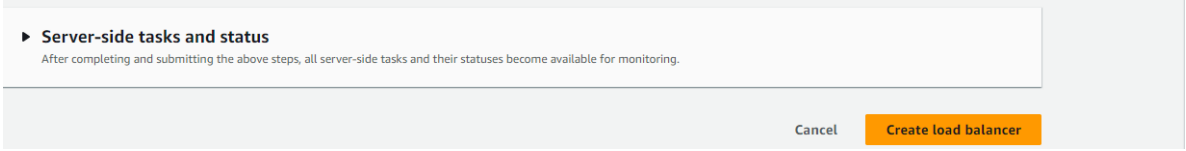
Remove the default security group and select the security group which we created.



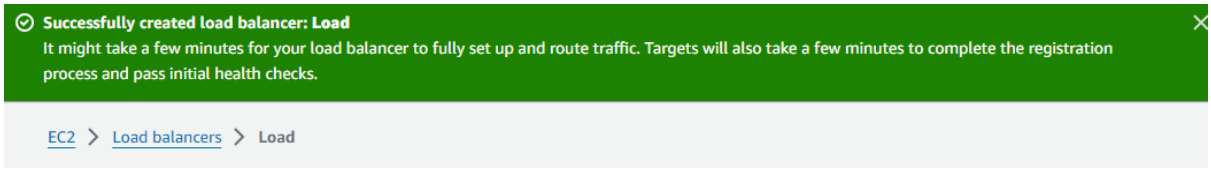
Attach the target group which we created.



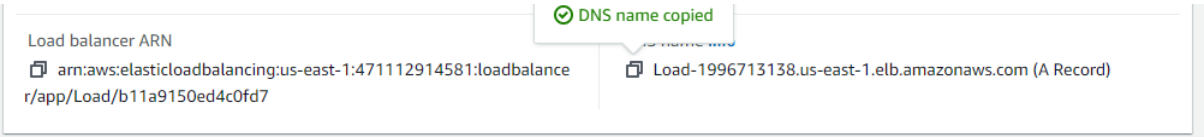
Click on create load balancer.



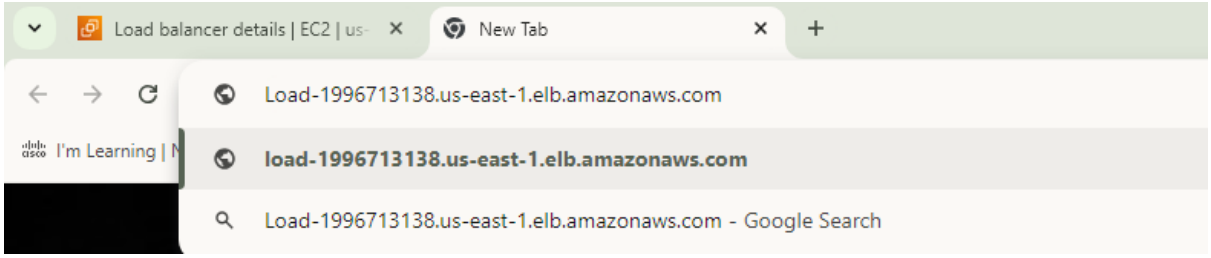
The target is attached to load balancer and the load balancer is created.



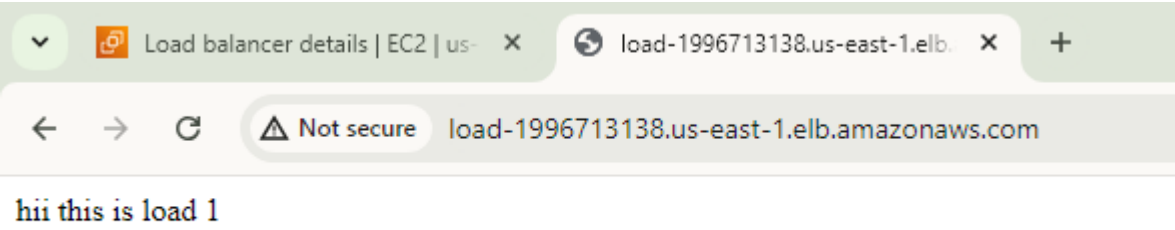
Copy the DNS name of the load balancer



Paste it in chrome

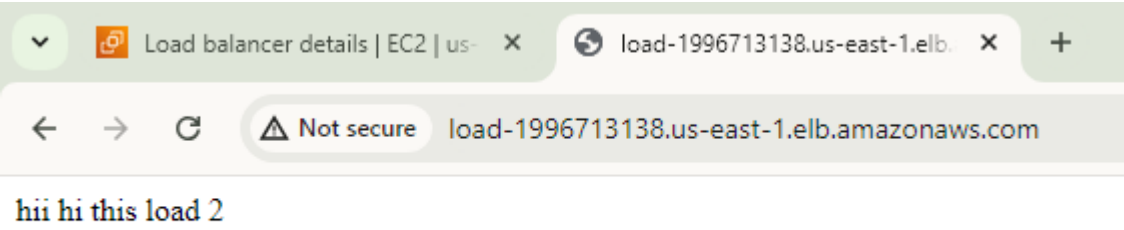


the data which we inserted in first instance will be appeared.



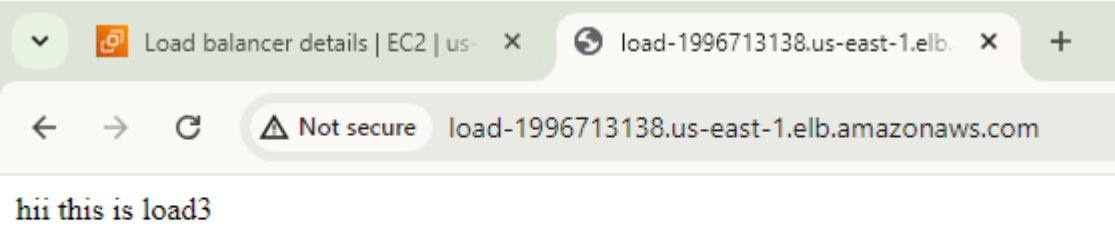
Do Refresh

Then the data which we inserted in second instance will be appeared.



Do Refresh

Then the data which we inserted in third instance will be appeared.



THIS IS HOW THE LOAD BALANCER WILL WORKS.