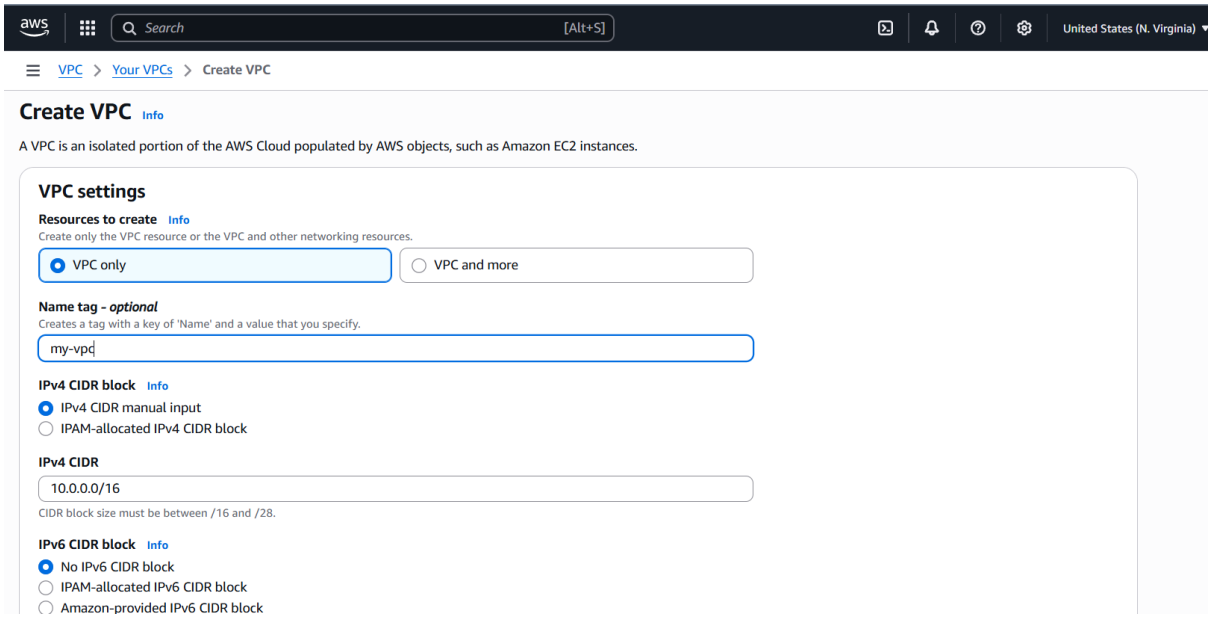
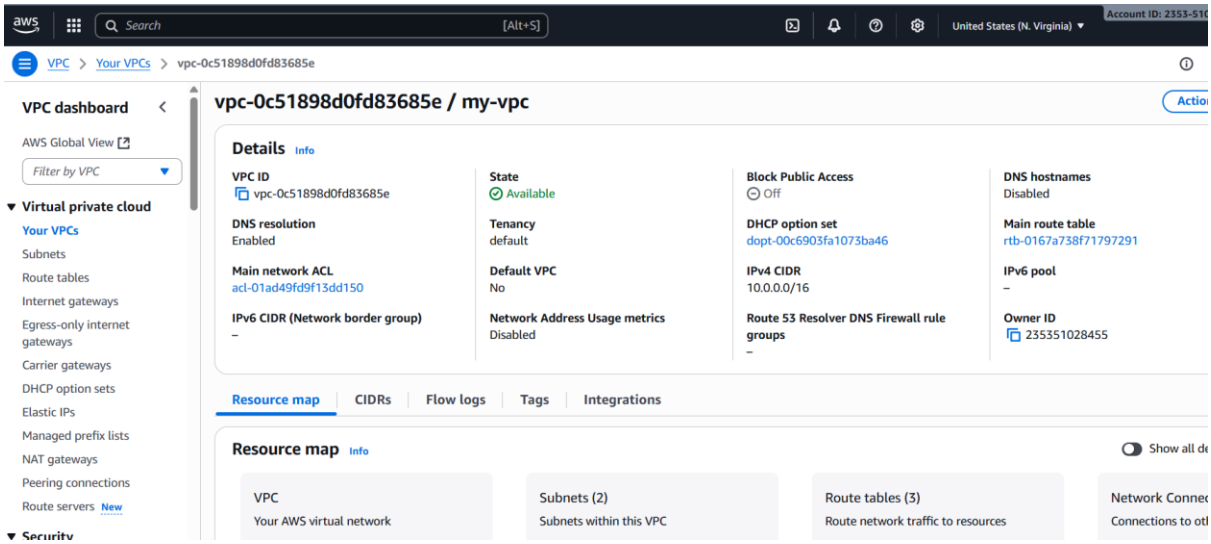
1. Create one VPC in N. Virginia region.

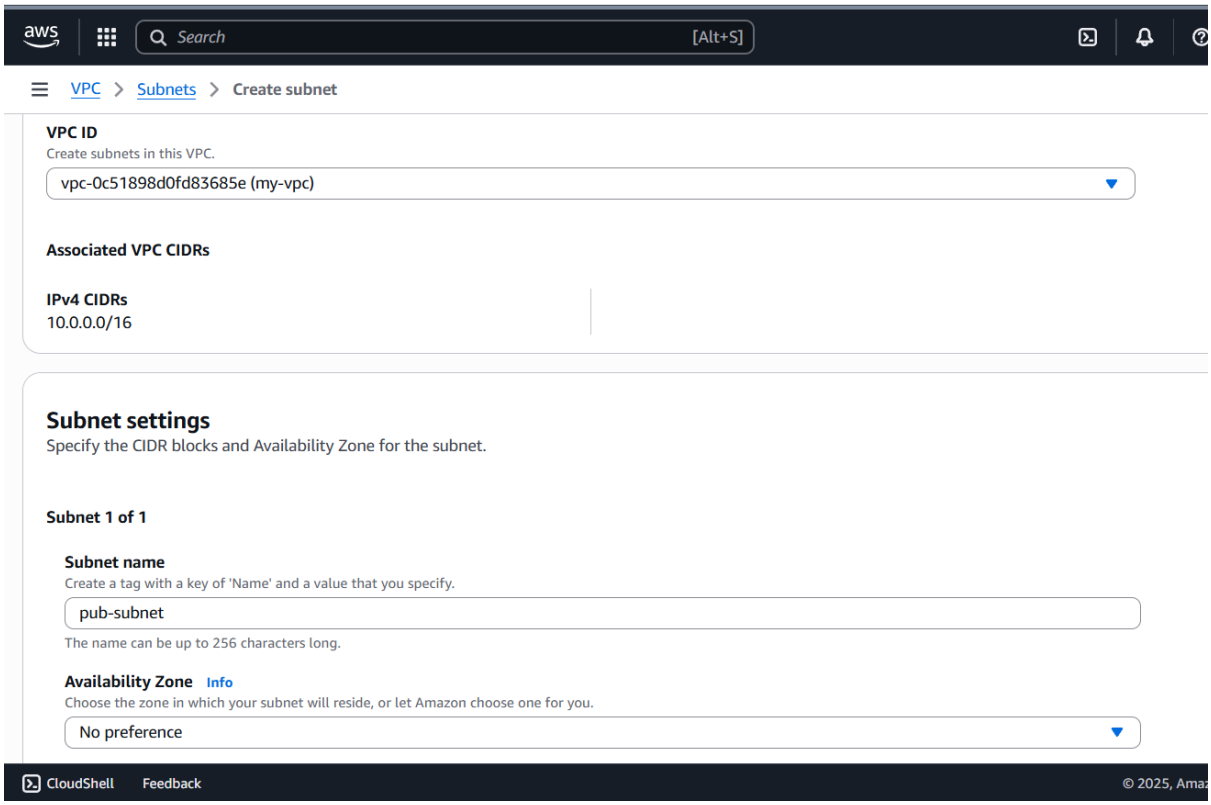
Go to n virginia region and select vpc create vpc.



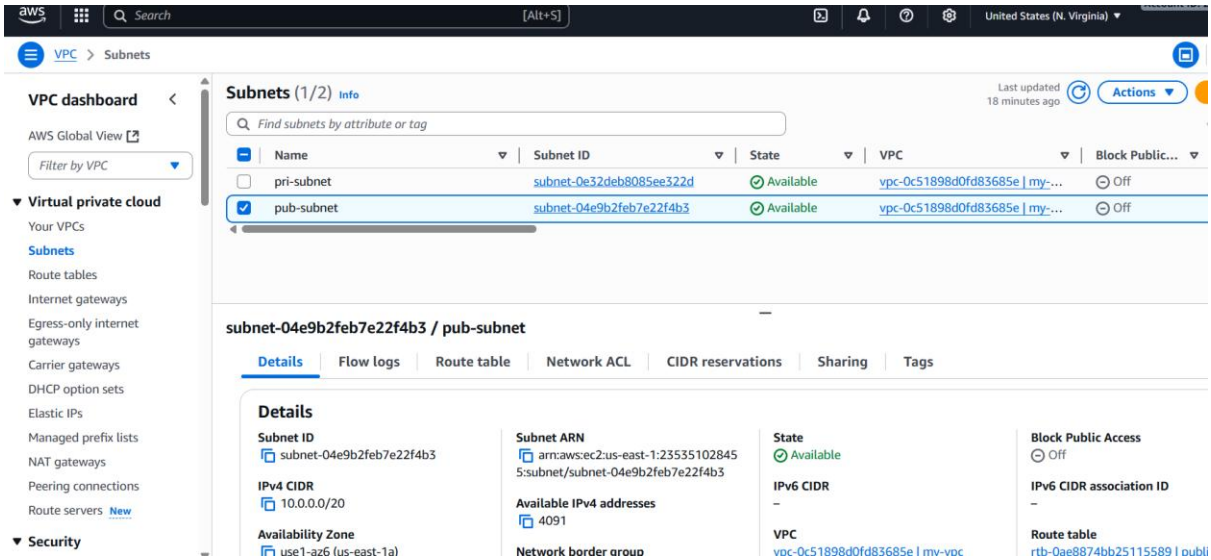


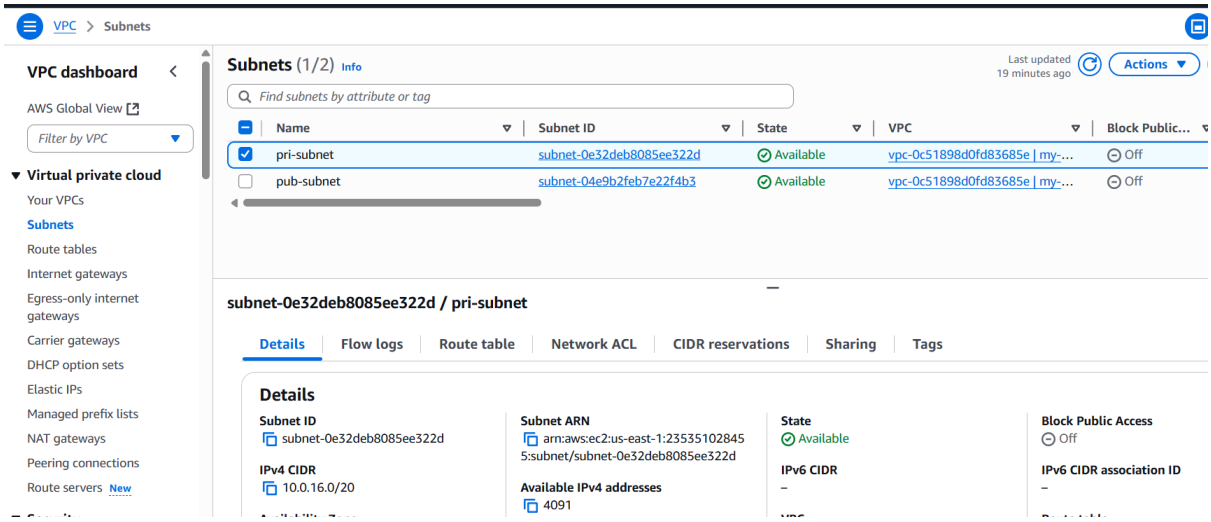
2.Create two subnets: one public subnet and one private subnet.

Go to subnets and create subnet and select vpc and give the cidr range for subnet.



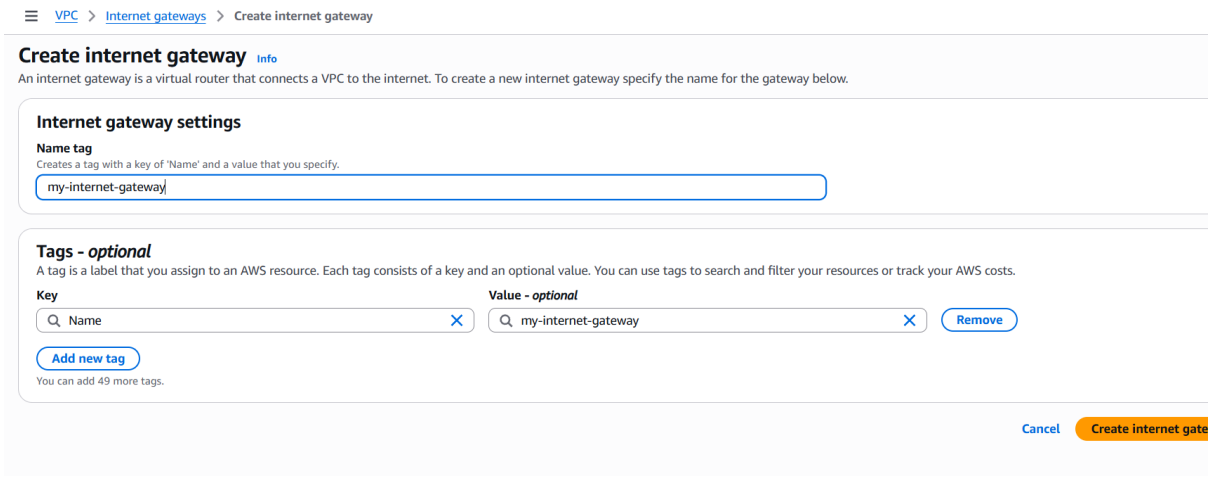
Create public subnet and select created vpc and give subnet name and cidr range .

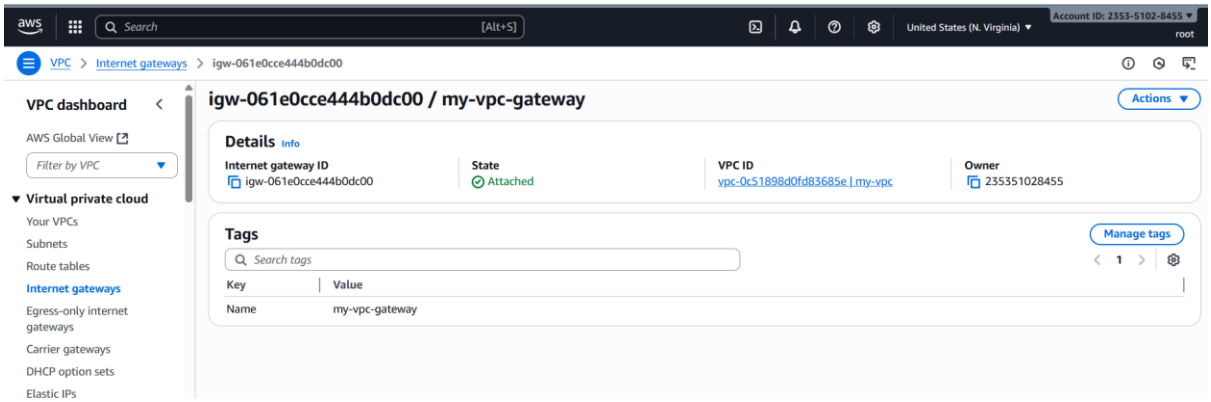




3. Attach an IGW to the VPC.

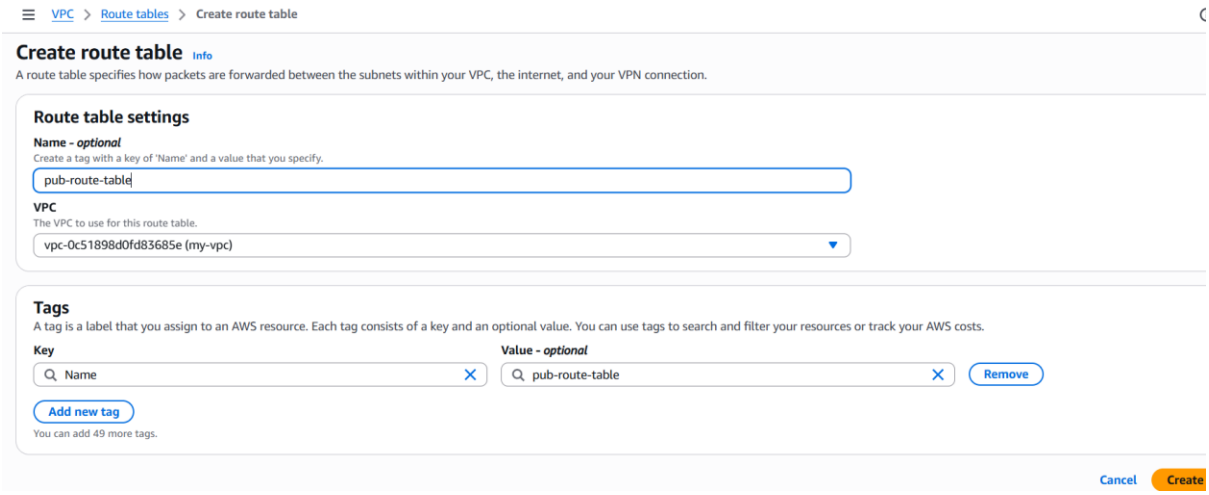
Go to internet gateways and create internet gateway and attach to vpc.

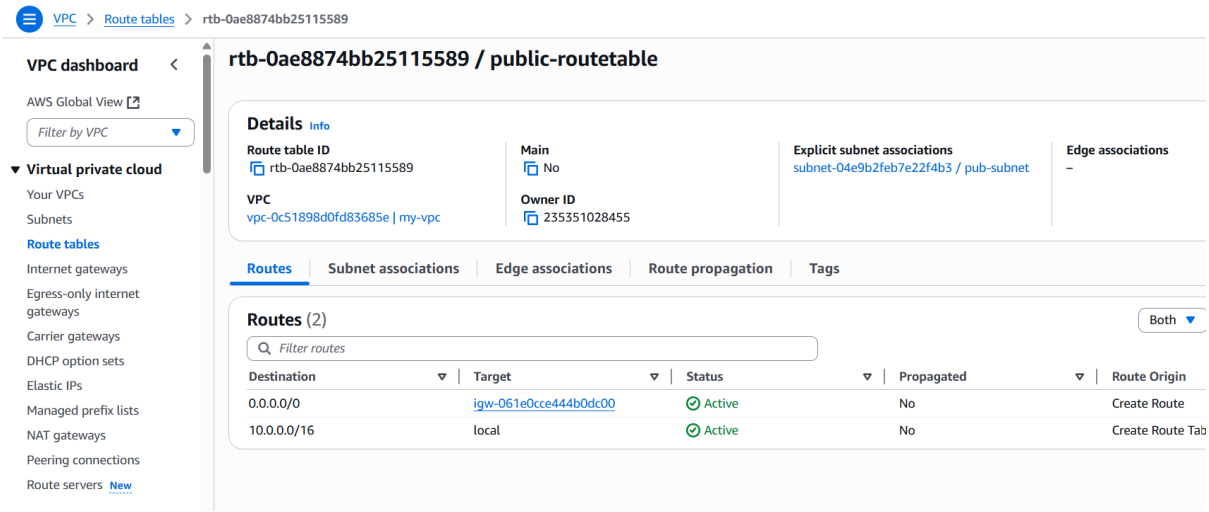




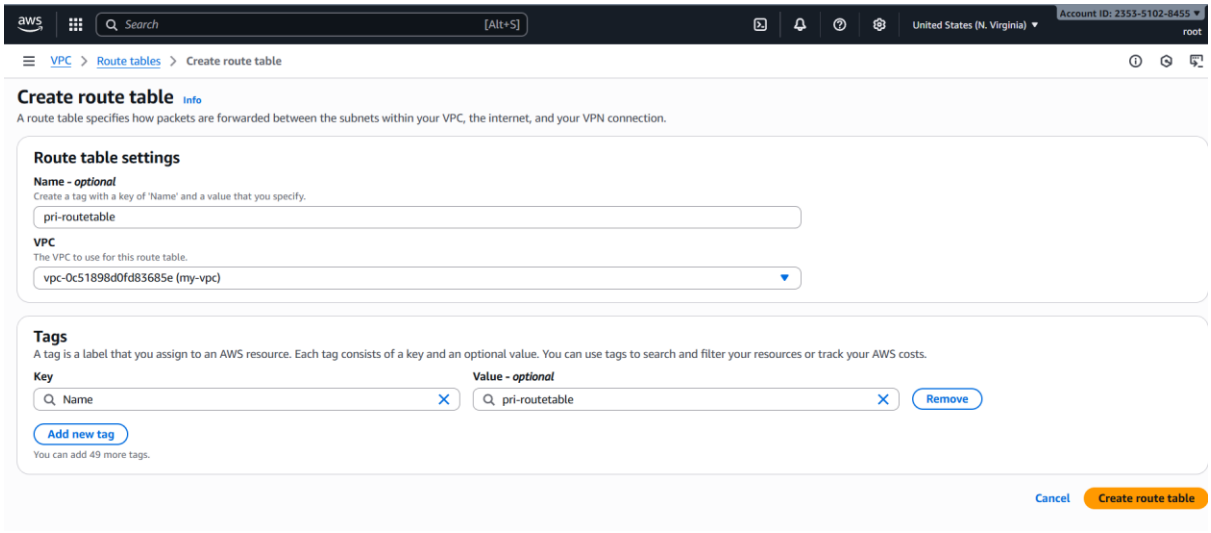
4. Create one public route table (RT) and one private route table.

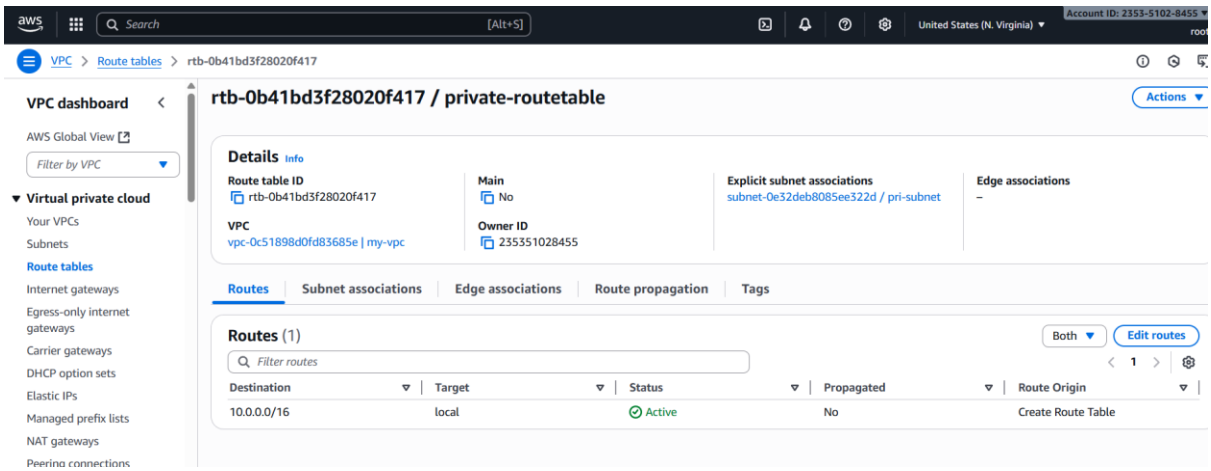
Go to route table and create route table and give name and select vpc.





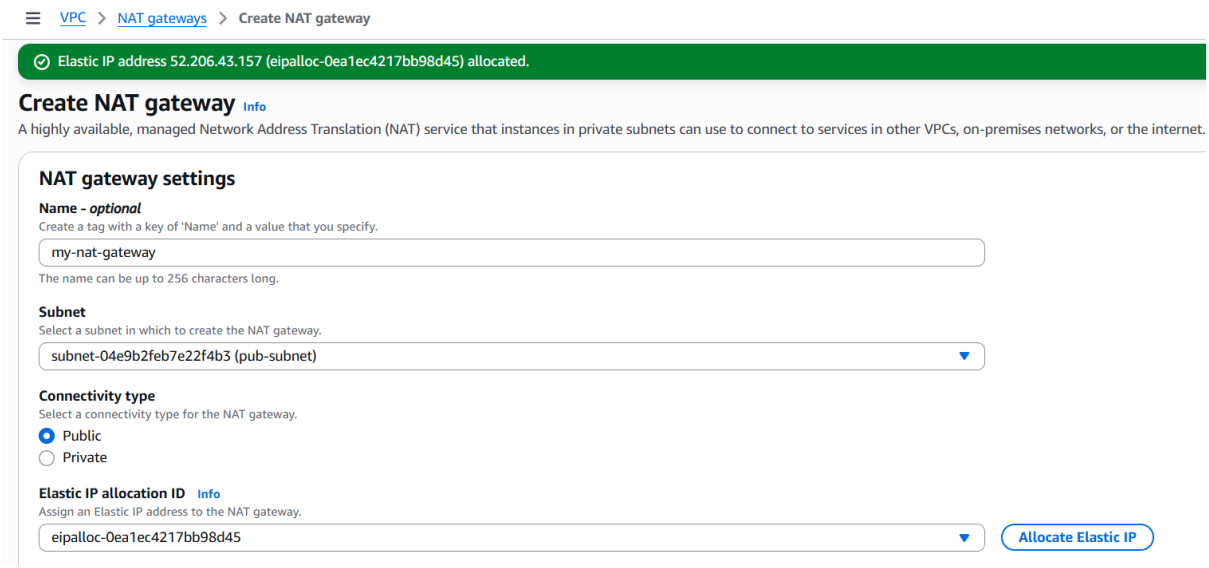
Create private route table and give name and select vpc.



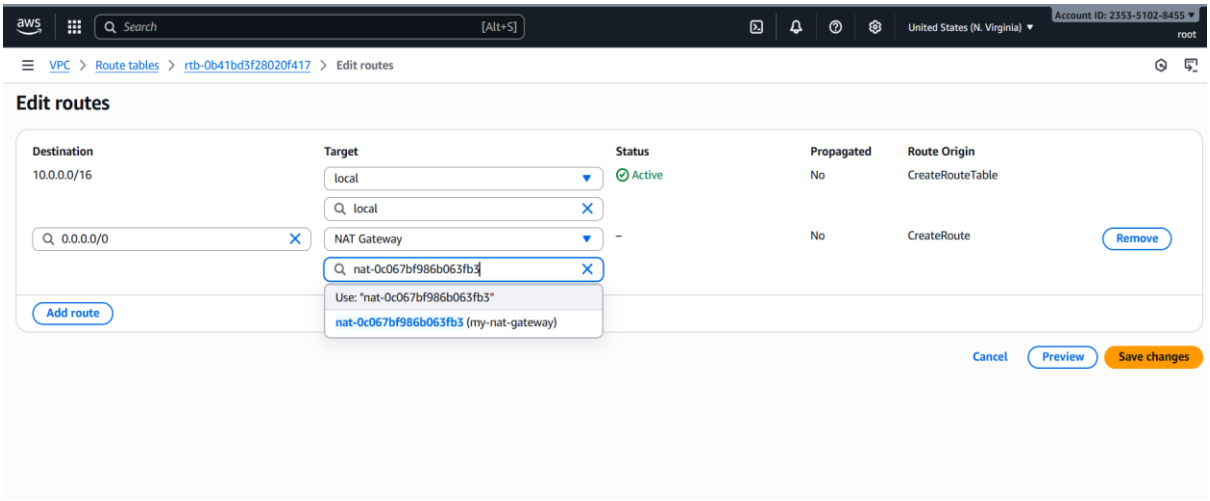


5.Deploy a NAT gateway in the public subnet and attach the NAT gateway to the private subnet.

Go to nat gateways and create nat gateway and select public subnet and allocate elastic ip.

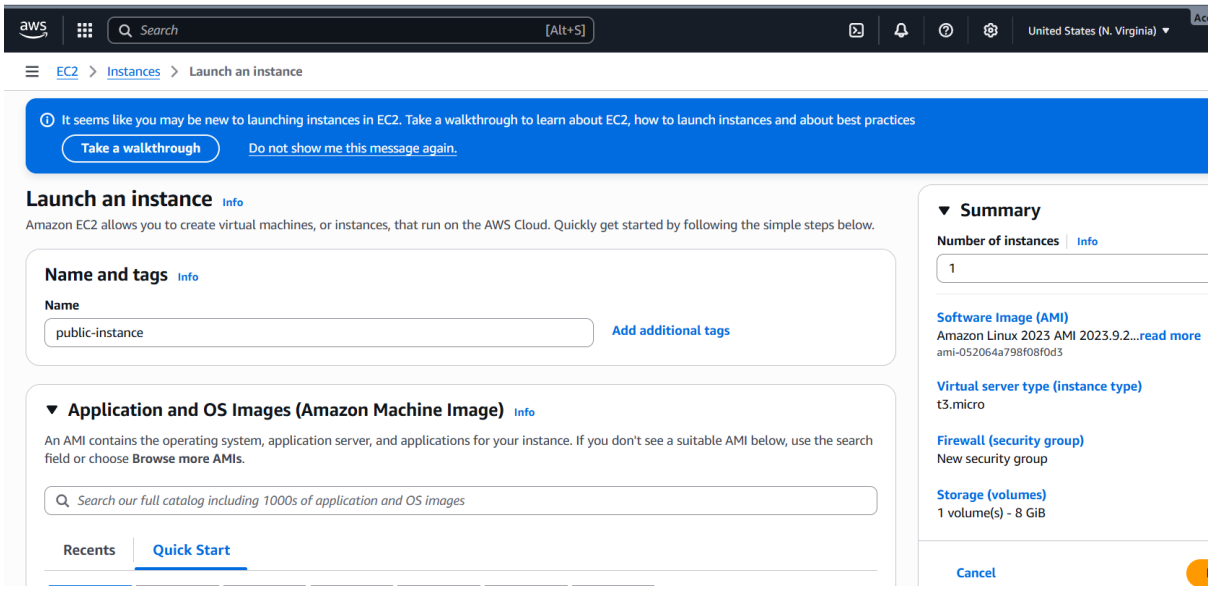


Go to route tables and go to private route table and edit routes and add nat gateway.

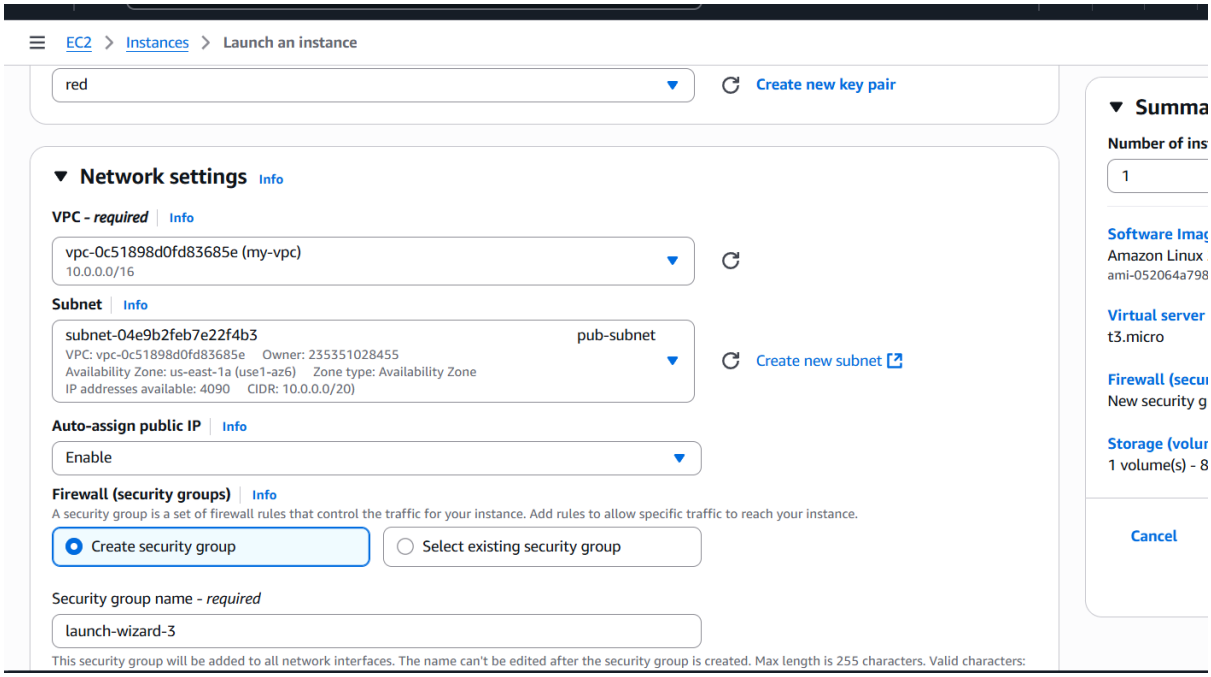


6.Create two instances, one in the public subnet and one in the private subnet.

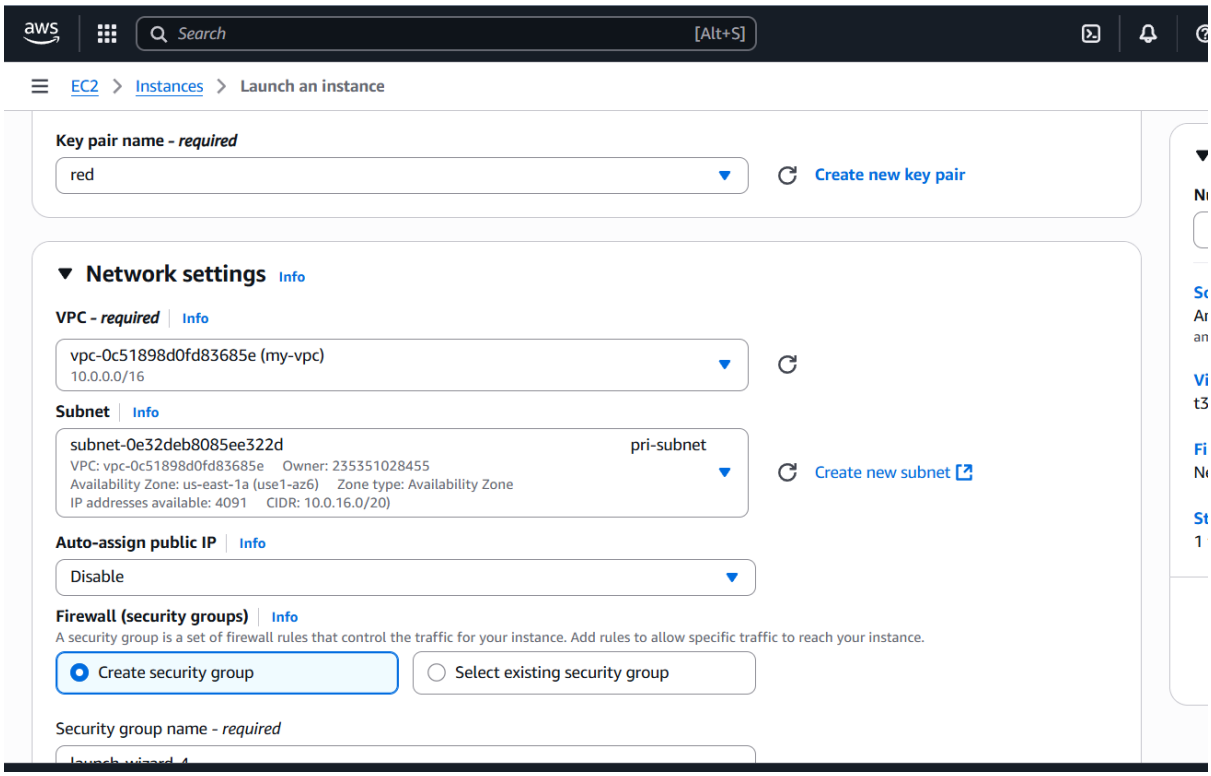
Go to instance and create instance give instance name and give the key pair

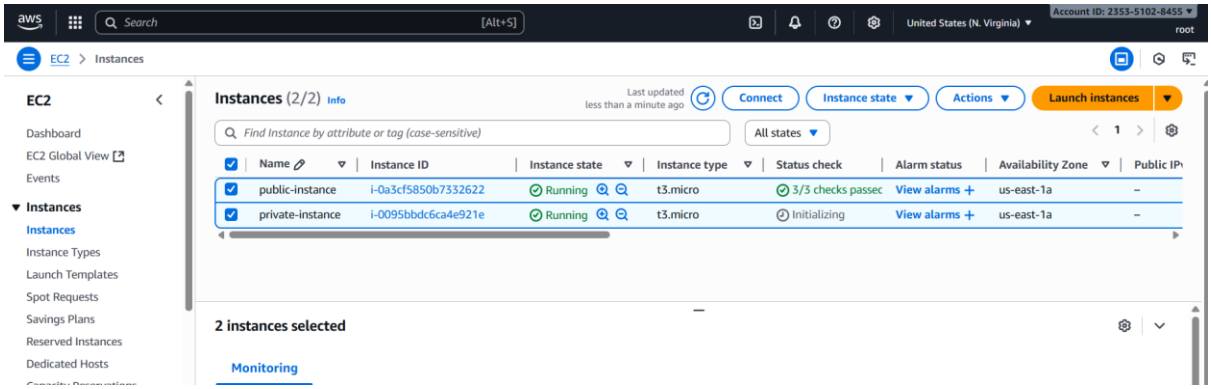


In network settings select your vpc and select public subnet and select existing security group or create new security group.



Repeat the same process for private instance. In network settings give the private subnet.





7. Deploy Apache server on both EC2 instances with a sample index.html file.

Connect to public instance.

```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i red.pem ec2-user@54.165.102.79
The authenticity of host '54.165.102.79 (54.165.102.79)' can't be established.
ED25519 key fingerprint is SHA256:NODh1opfGgj8tAgOiSUgknZpoGvXX92qeIcTIuNlo4U.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.165.102.79' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-7-172 ~]$ sudo su -
[root@ip-10-0-7-172 ~]#
```

Sudo yum update -y

Sudo yum install httpd -y

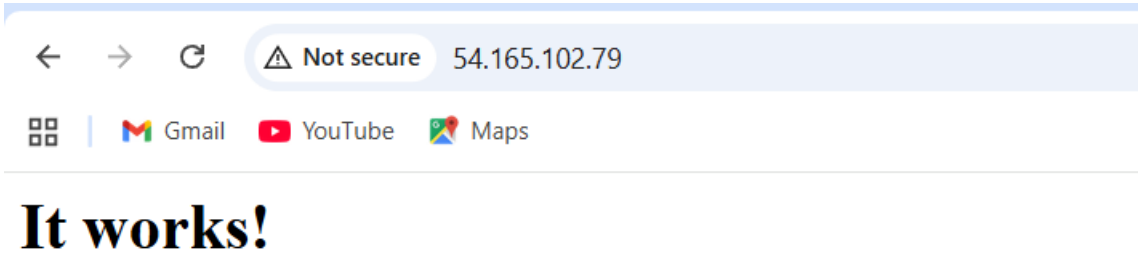
```
[root@ip-10-0-7-172 ~]# sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-7-172 ~]# yum install httpd -y
Last metadata expiration check: 0:00:29 ago on Wed Oct  8 15:14:19 2025.
Dependencies resolved.
================================================================================
 Package                                Architecture              Version
================================================================================
Installing:
 httpd                                  x86_64                    2.4.65-1.amzn2023.
Installing dependencies:
 apr                                    x86_64                    1.7.5-1.amzn2023.0
 apr-util                               x86_64                    1.6.3-1.amzn2023.0
 generic-logos-httpd                    noarch                    18.0.0-12.amzn2023
```

Sudo systemctl start httpd

Sudo systemctl start httpd

```
[root@ip-10-0-7-172 ~]# sudo systemctl start httpd
[root@ip-10-0-7-172 ~]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Wed 2025-10-08 15:15:29 UTC; 58s ago
       Docs: man:httpd.service(8)
   Main PID: 27057 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/s
      Tasks: 177 (limit: 1053)
     Memory: 13.3M
        CPU: 106ms
     CGroup: /system.slice/httpd.service
             ├─27057 /usr/sbin/httpd -DFOREGROUND
             ├─27084 /usr/sbin/httpd -DFOREGROUND
             ├─27088 /usr/sbin/httpd -DFOREGROUND
             ├─27089 /usr/sbin/httpd -DFOREGROUND
             └─27091 /usr/sbin/httpd -DFOREGROUND
```

Open with public ip in your browser.



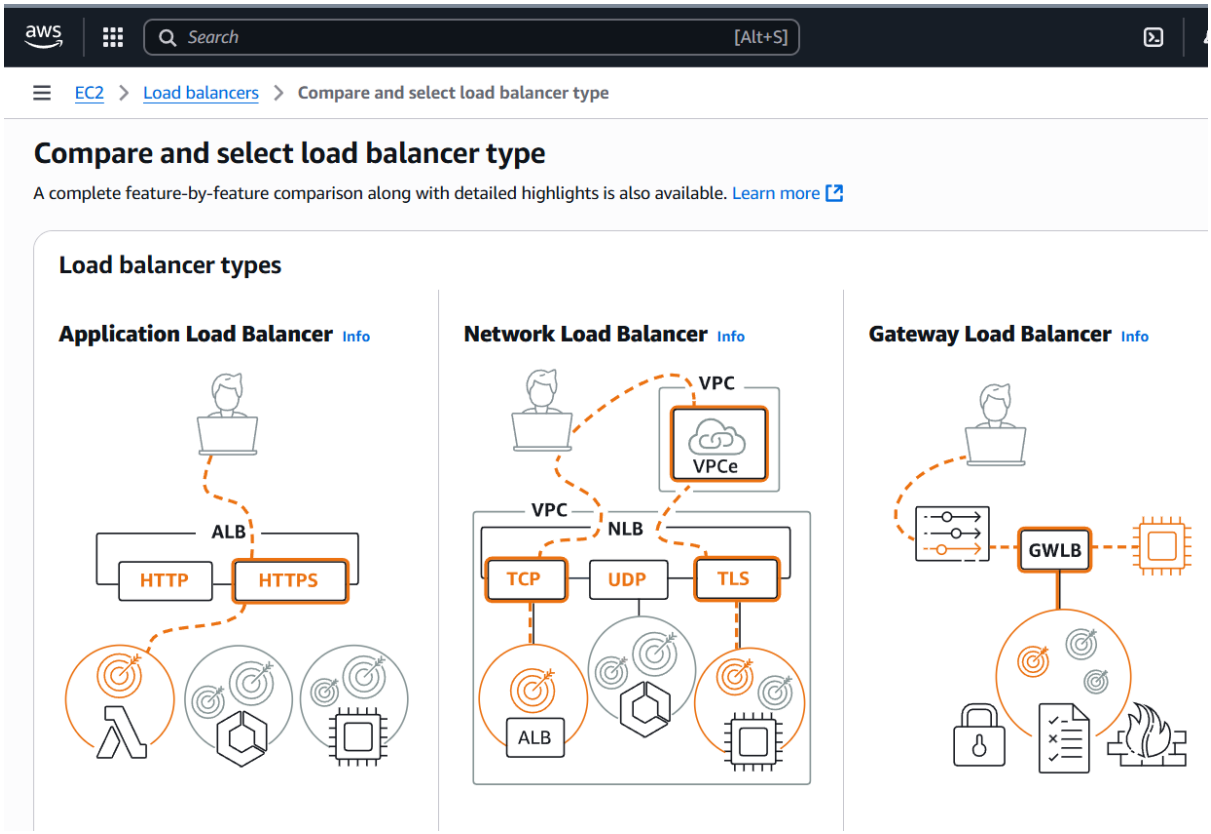
Login with the public ip from that login with the private instance for that there is no public ip so,we use public instance ip as bastion server.

```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i red.pem ec2-user@54.165.102.79
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Oct  8 15:13:39 2025 from 103.143.169.218
[ec2-user@ip-10-0-7-172 ~]$ sudo su -
Last login: Wed Oct  8 15:13:48 UTC 2025 on pts/1
[root@ip-10-0-7-172 ~]# ssh -i red.pem ec-2 user@10.0.20.86
Warning: Identity file red.pem not accessible: No such file or directory.
ssh: Could not resolve hostname ec-2: Name or service not known
[root@ip-10-0-7-172 ~]# ssh -i red.pem ec2-user@10.0.20.86
Warning: Identity file red.pem not accessible: No such file or directory.
The authenticity of host '10.0.20.86 (10.0.20.86)' can't be established.
ED25519 key fingerprint is SHA256:eKoLd6V1vzbWyE3O6wscXsvO27Af9DSFaOZGy9SZC1E.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.20.86' (ED25519) to the list of known hosts.
ec2-user@10.0.20.86: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[root@ip-10-0-7-172 ~]#
```

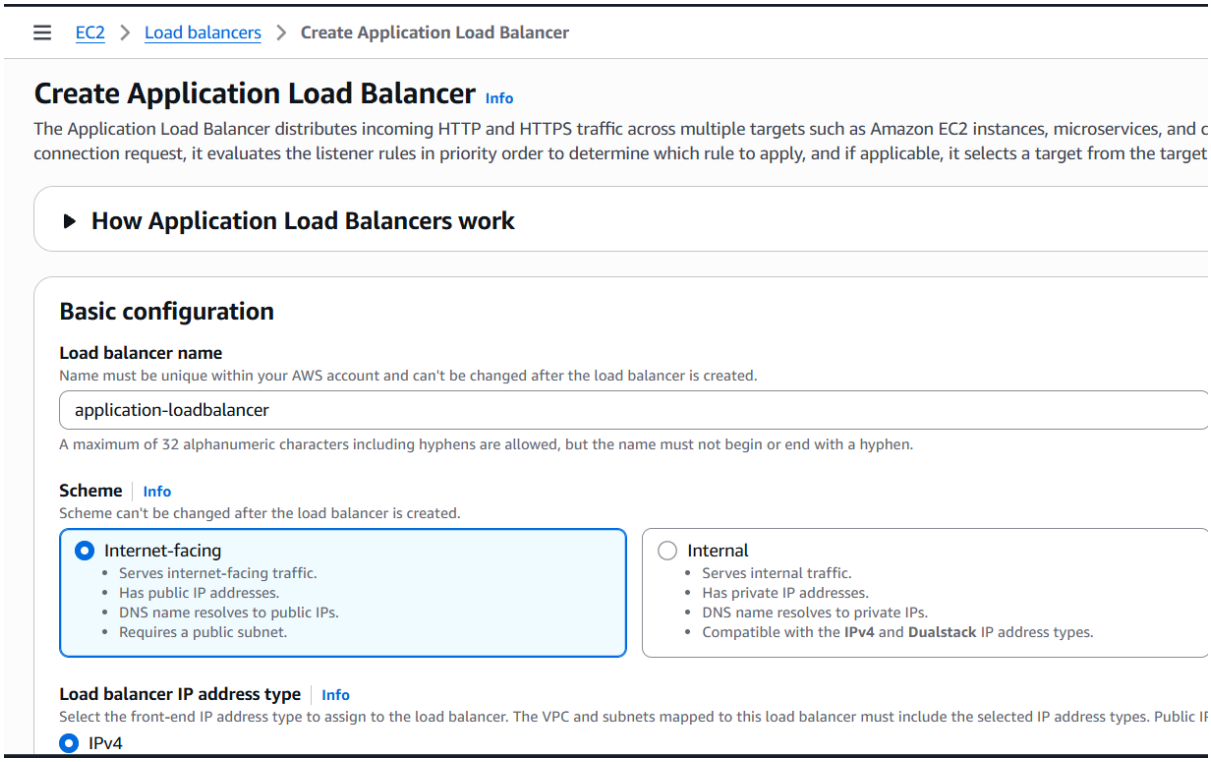
```
[root@ip-10-0-7-172 ~]# yum install httpd -y
Last metadata expiration check: 0:22:01 ago on Wed Oct  8 15:14:19 2025.
Package httpd-2.4.65-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-7-172 ~]# sudo systemctl start httpd
[root@ip-10-0-7-172 ~]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Wed 2025-10-08 15:15:29 UTC; 21min ago
       Docs: man:httpd.service(8)
   Main PID: 27057 (httpd)
     Status: "Total requests: 2; Idle/Busy workers 100/0;Requests/sec: 0.00156; Bytes served/sec:   1 B/sec"
      Tasks: 230 (limit: 1053)
     Memory: 16.6M
        CPU: 1.567s
     CGroup: /system.slice/httpd.service
             ├─27057 /usr/sbin/httpd -DFOREGROUND
             ├─27084 /usr/sbin/httpd -DFOREGROUND
             ├─27088 /usr/sbin/httpd -DFOREGROUND
             ├─27089 /usr/sbin/httpd -DFOREGROUND
             ├─27091 /usr/sbin/httpd -DFOREGROUND
             └─27507 /usr/sbin/httpd -DFOREGROUND

Oct 08 15:15:29 ip-10-0-7-172.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 08 15:15:29 ip-10-0-7-172.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 08 15:15:29 ip-10-0-7-172.ec2.internal httpd[27057]: Server configured, listening on: port 80
[root@ip-10-0-7-172 ~]#
```

8.Create one application load balancer and attach it to both EC2 instances.

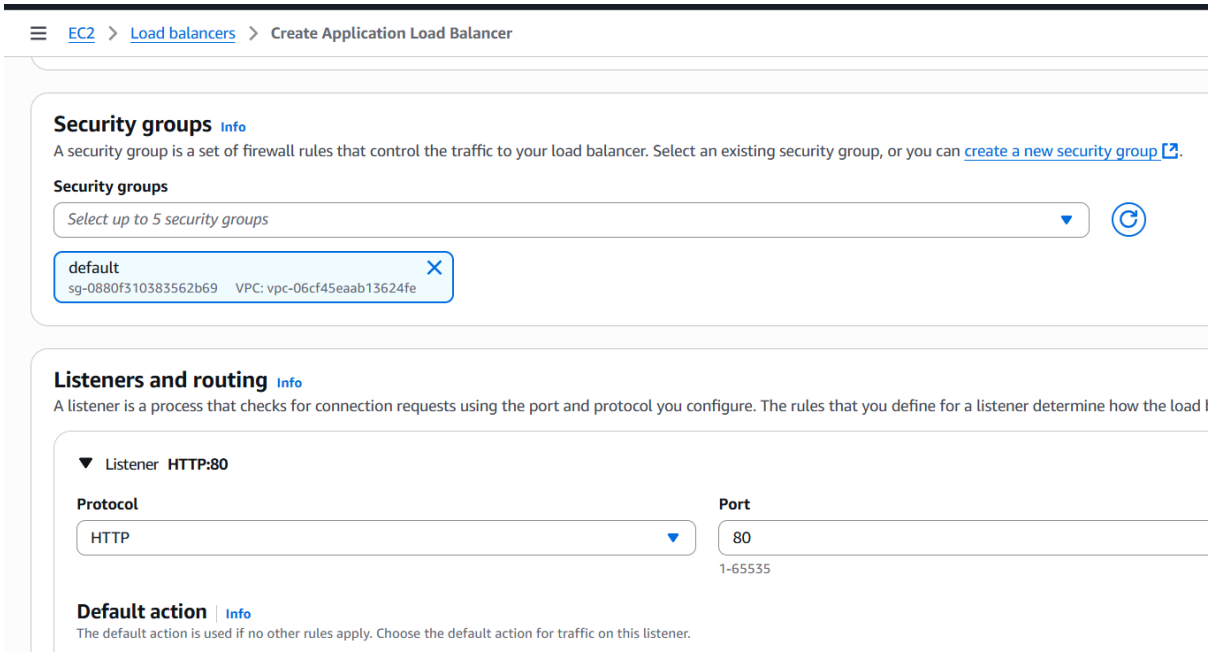
Go to load balancers and select application load balancer and create load balancer.



Give the name and select vpc,select internet facing and select ipv4.

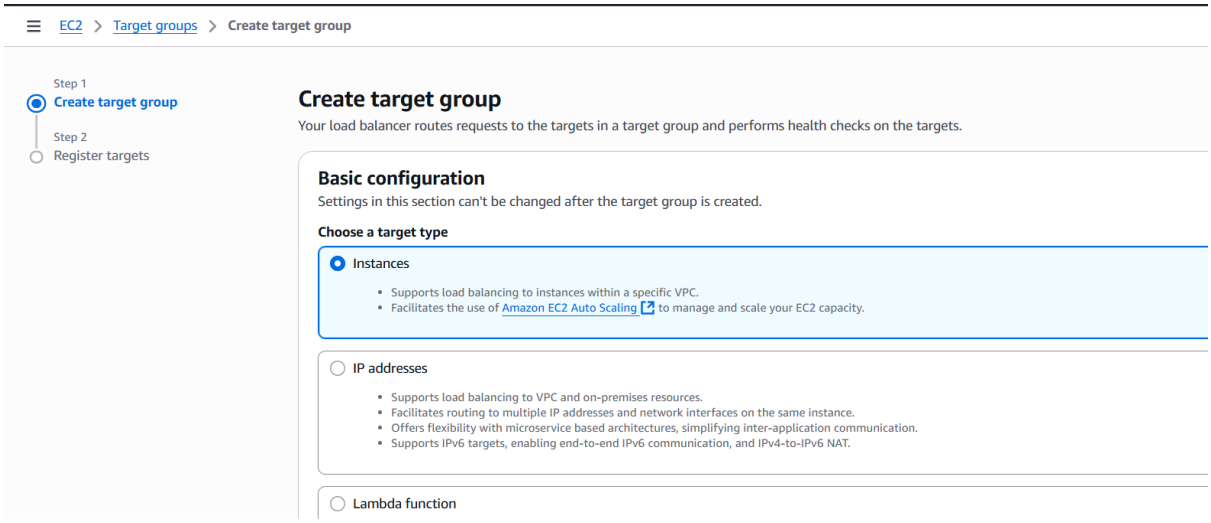


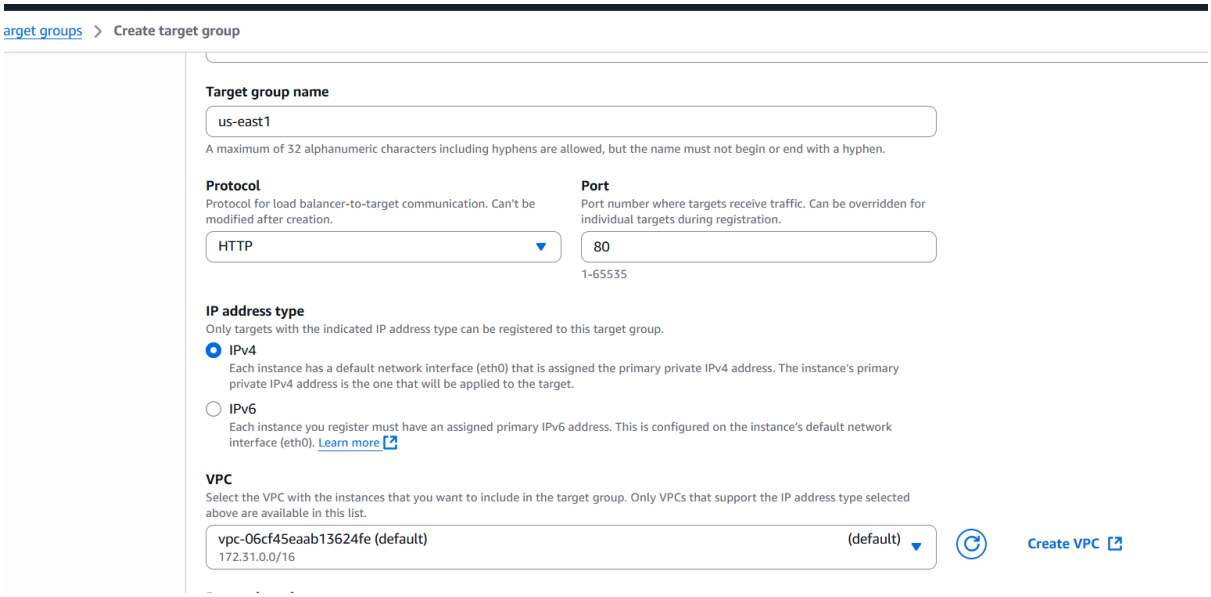
Select security groups and listners port 80.

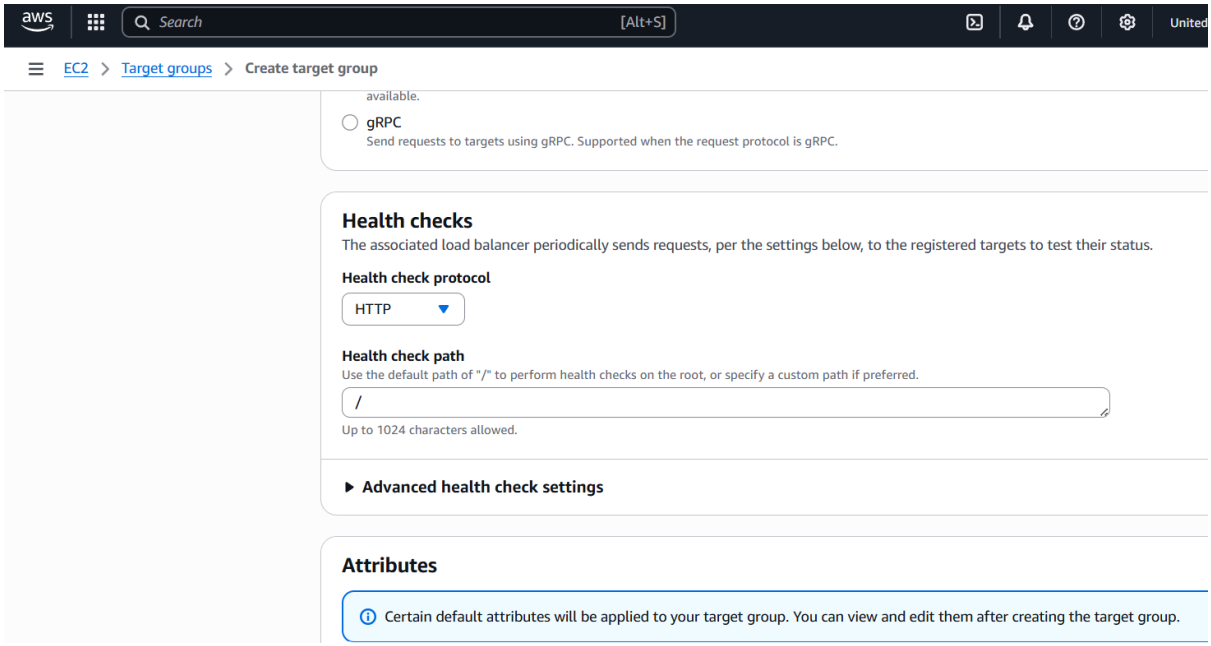


Create target group to redirect to target group which we placed.

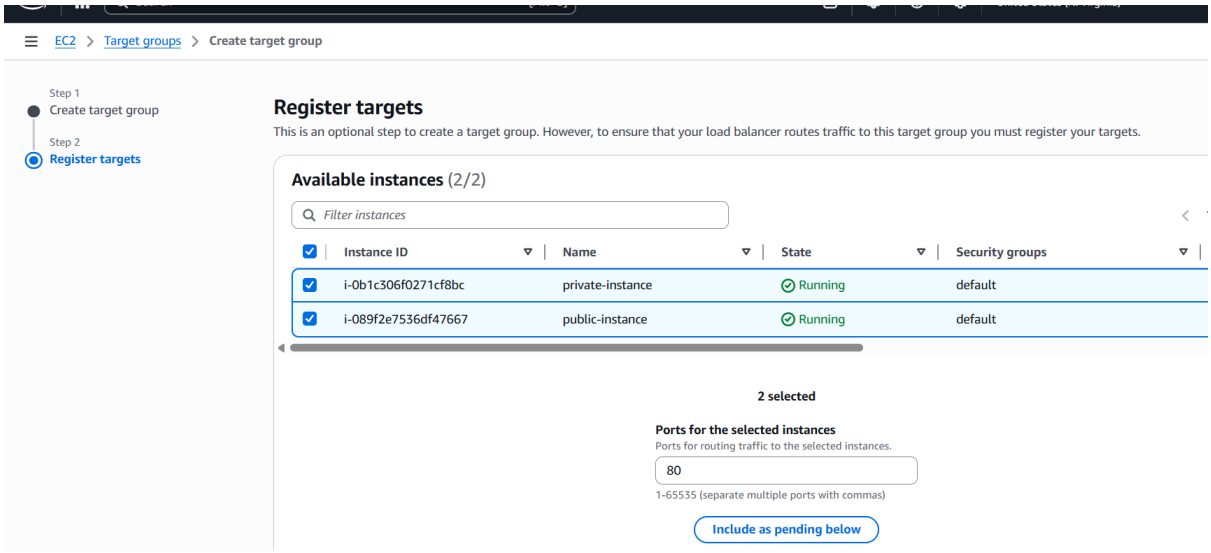
Go to ec2 and select target groups and create.

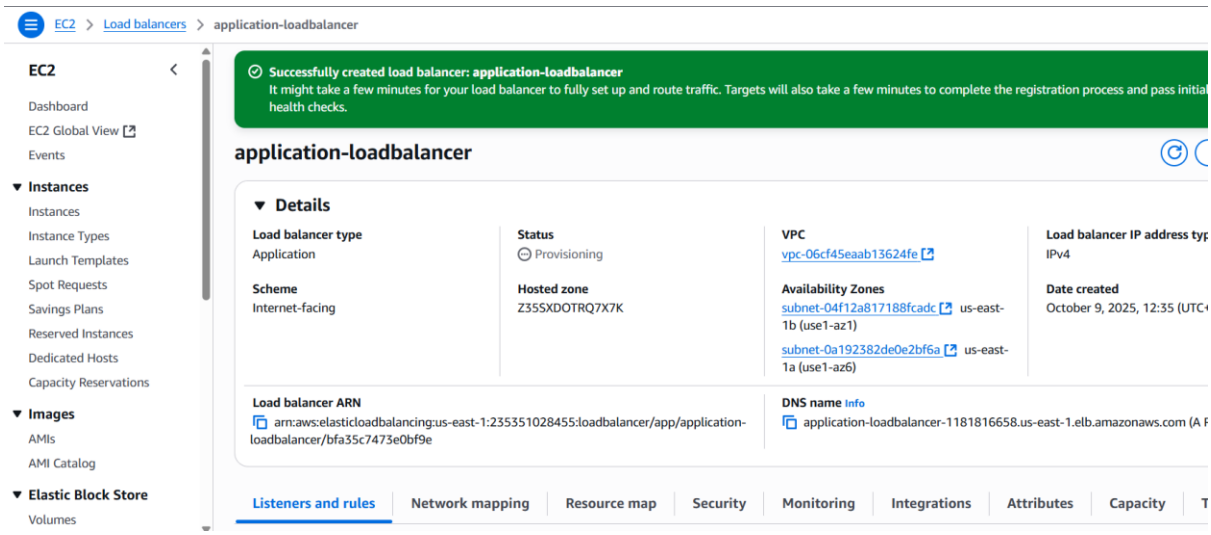






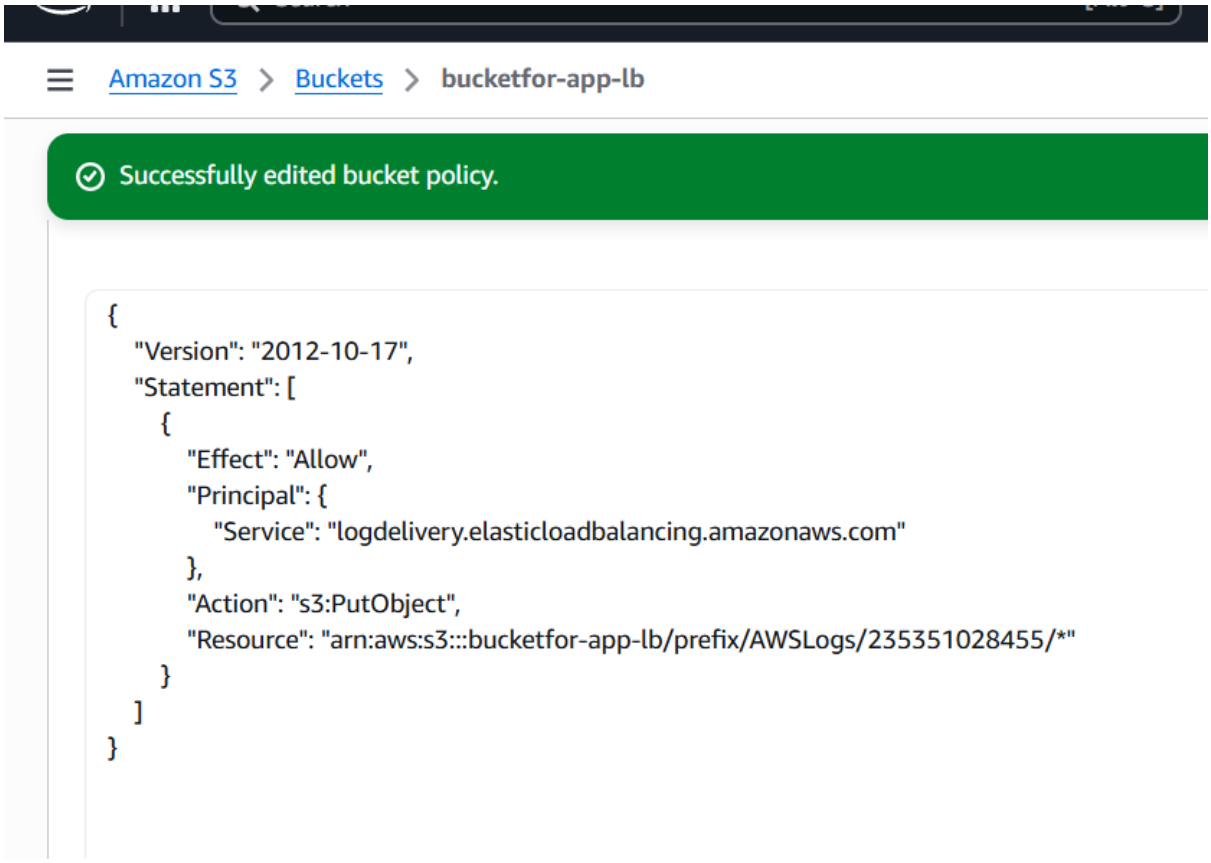
Select the both instances and click create.





9.Store application load balancer logs in S3.

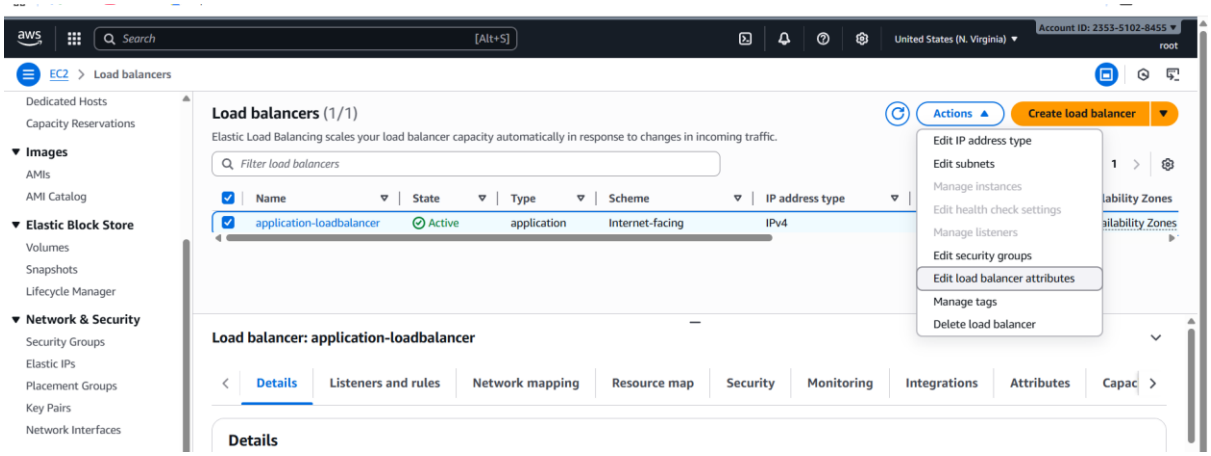
Go to s3 bucket and create a bucket and go to that bucket and add permissions add the poicy.



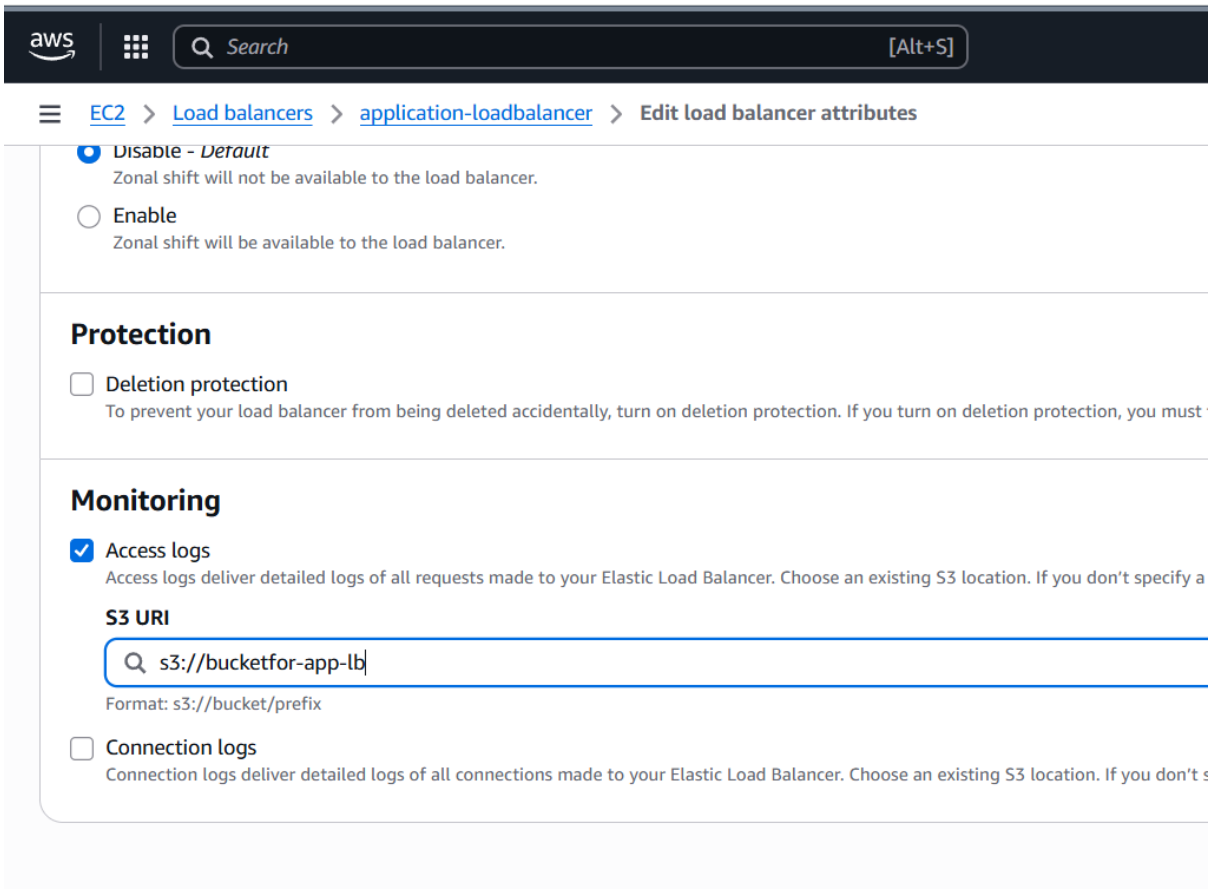

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Principal": {
    "Service":
"logdelivery.elasticloadbalancing.amazonaws.com"
```

```
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::bucketfor-app-lb/prefix/AWSLogs/235351028455/*"
  }
 ]
}
```

Go to load balancers and edit the load balancer attribute.



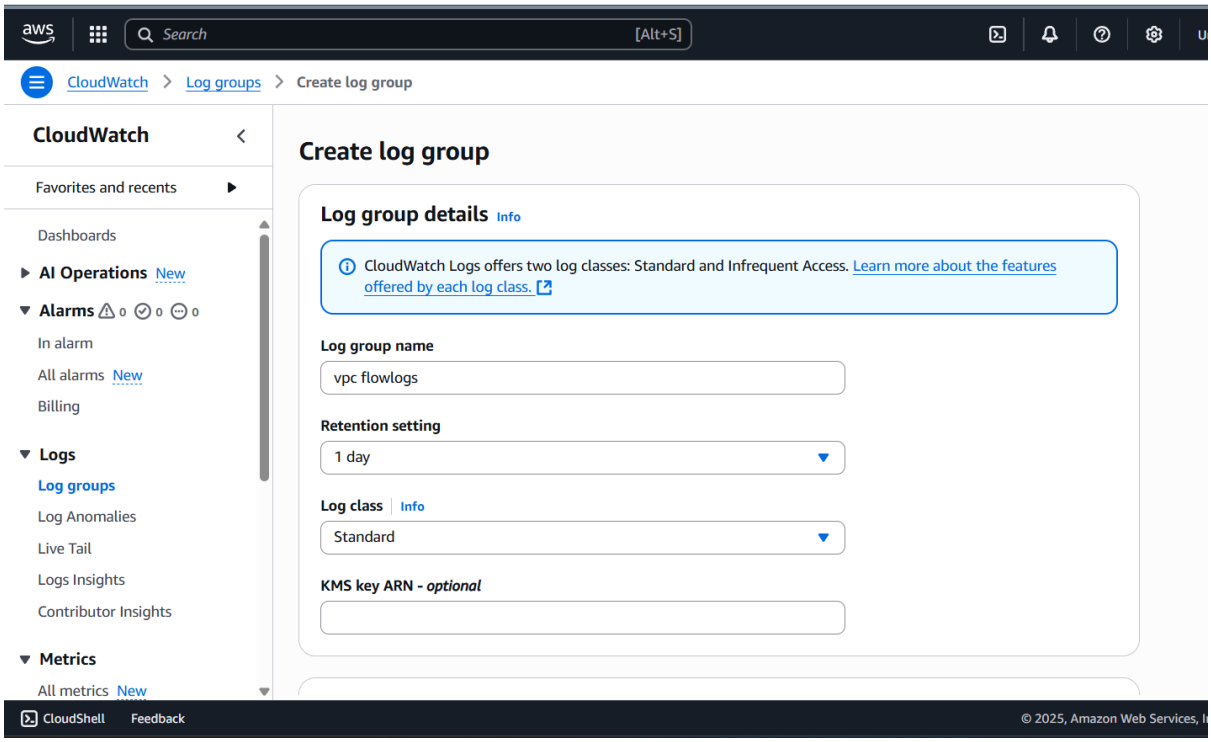
Go to monitoring and select access logs and choose your bucket.



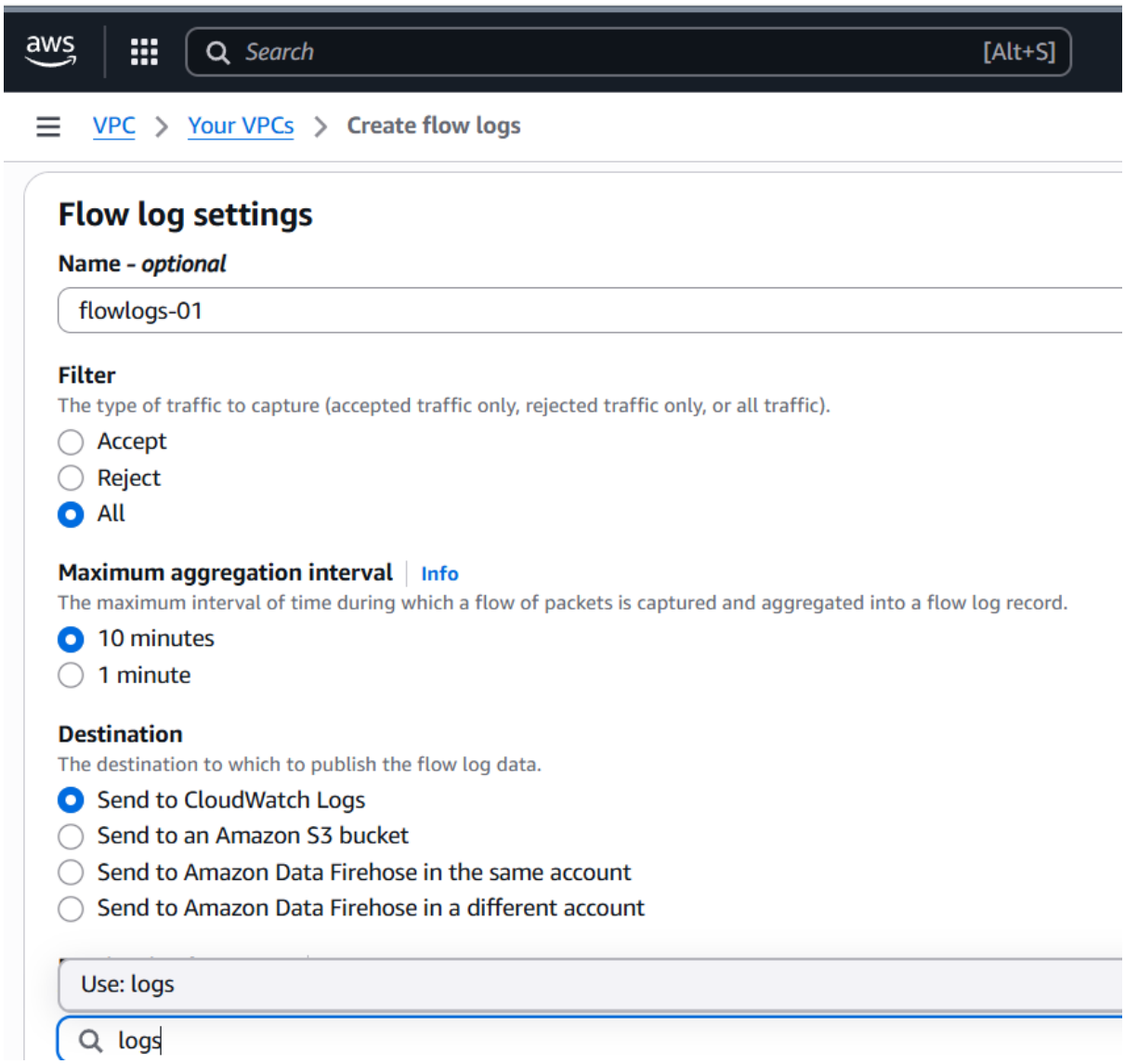
Then the flowlogs are stored in the bucket.

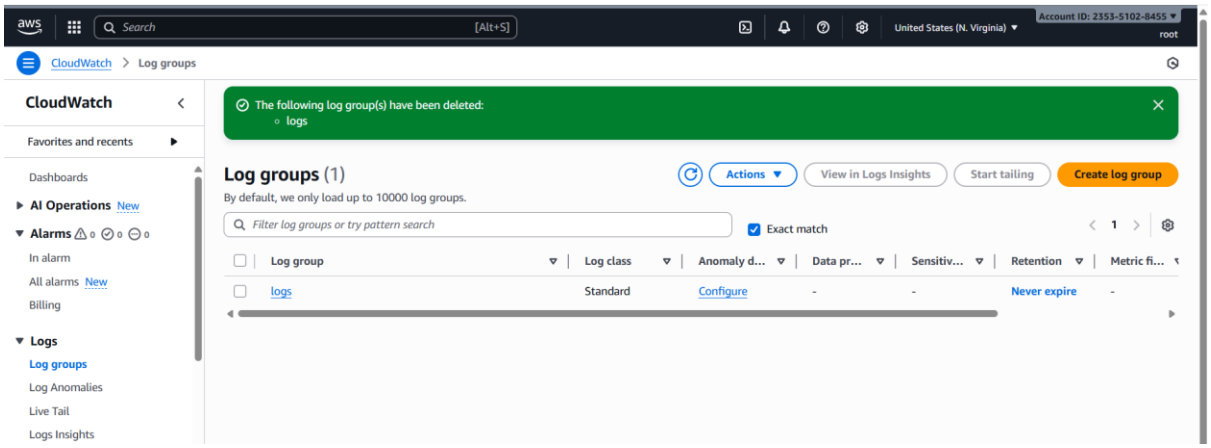
10.Store the VPC flow logs in a CloudWatch log group.

Go to cloud watch and select flowlog groups and create flowlogs



Go to vpc and select flowlogs and create flowlogs.


aws
Search
[Alt+S]
VPC > Your VPCs > Create flow logs
Flow log settings
Name - optional
flowlogs-01
Filter
The type of traffic to capture (accepted traffic only, rejected traffic only, or all traffic).
Accept
Reject
All
Maximum aggregation interval | Info
The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.
10 minutes
1 minute
Destination
The destination to which to publish the flow log data.
Send to CloudWatch Logs
Send to an Amazon S3 bucket
Send to Amazon Data Firehose in the same account
Send to Amazon Data Firehose in a different account
Use: logs
logs


aws
Search
[Alt+S]
United States (N. Virginia)
Account ID: 2353-5102-8455
root
CloudWatch > Log groups
CloudWatch
Favorites and recents
Dashboards
AI Operations New
Alarms
In alarm
All alarms New
Billing
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
The following log group(s) have been deleted:
logs
Log groups (1)
By default, we only load up to 10000 log groups.
Actions
View in Logs Insights
Start tailing
Create log group
Filter log groups or try pattern search
Exact match
Log group
Log class
Anomaly d...
Data pr...
Sensitiv...
Retention
Metric fi...
logs
Standard
Configure
Never expire

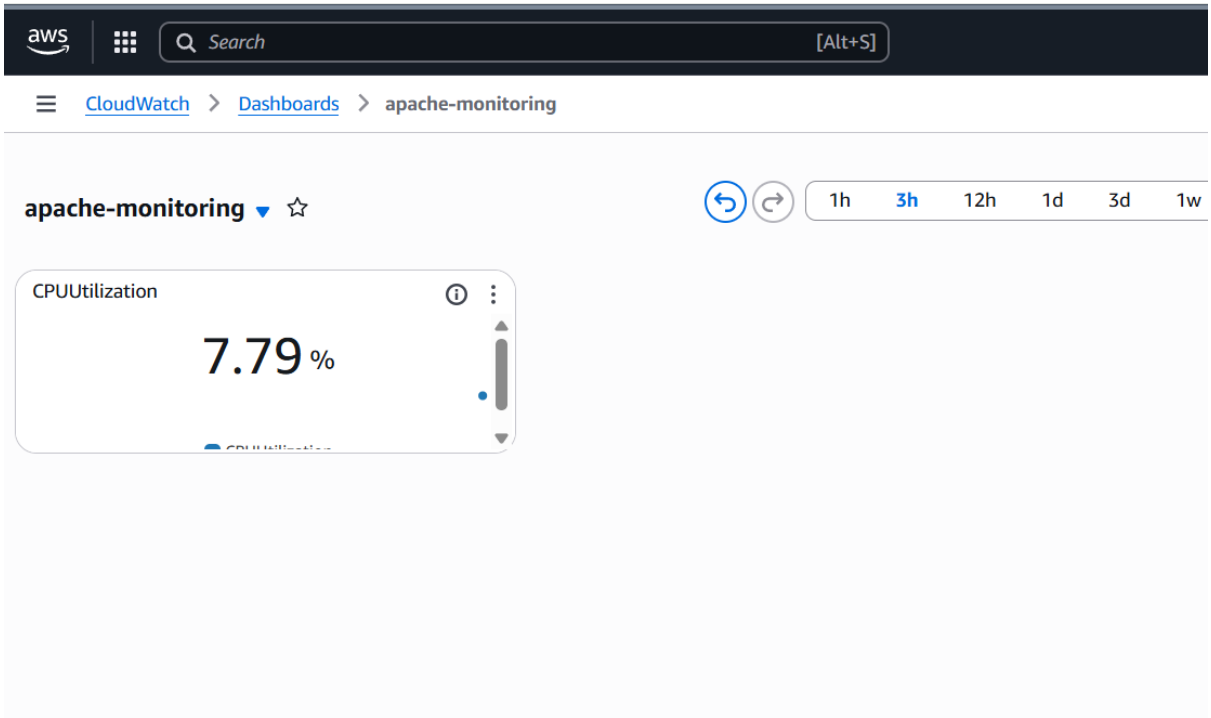
11.Create monitoring dashboards to monitor CPU utilization and to monitor the Apache service.

Launch a instance and install apache in that.

```
[root@ip-172-31-83-40 ~]# yum install httpd
Last metadata expiration check: 0:00:17 ago on Thu Oct  9 09:43:54 2025.
Dependencies resolved.
================================================================================
 Package                         Architecture            Version
================================================================================
Installing:
 httpd                           x86_64                  2.4.65-1.amzn2023.0.1
Installing dependencies:
 apr                             x86_64                  1.7.5-1.amzn2023.0.4
 apr-util                        x86_64                  1.6.3-1.amzn2023.0.1
 generic-logos-httpd             noarch                  18.0.0-12.amzn2023.0.3
 httpd-core                      x86_64                  2.4.65-1.amzn2023.0.1
 httpd-filesystem                noarch                  2.4.65-1.amzn2023.0.1
 httpd-tools                     x86_64                  2.4.65-1.amzn2023.0.1
 libbrotli                       x86_64                  1.0.9-4.amzn2023.0.2
 mailcap                         noarch                  2.1.49-3.amzn2023.0.3
Installing weak dependencies:
 apr-util-openssl                x86_64                  1.6.3-1.amzn2023.0.1
 mod_http2                       x86_64                  2.0.27-1.amzn2023.0.3
 mod_lua                         x86_64                  2.4.65-1.amzn2023.0.1
```

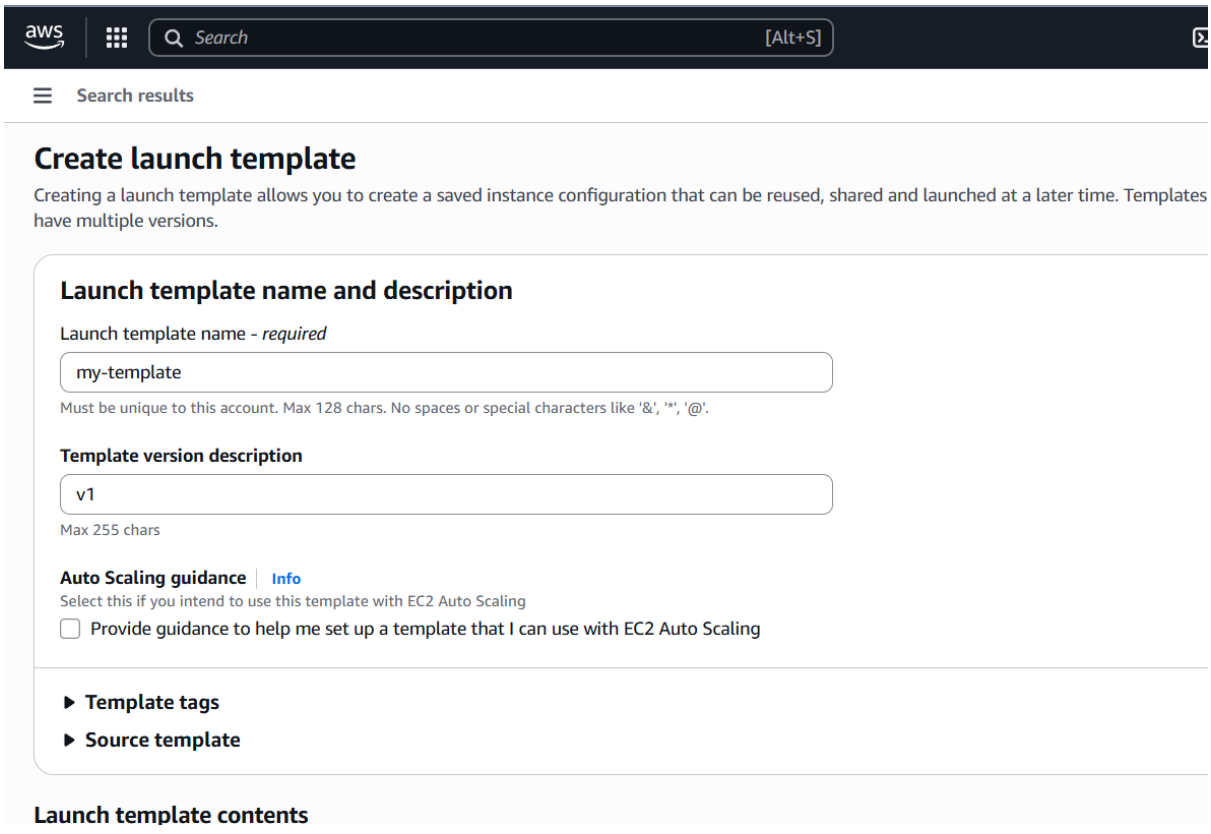
Go to cloudwatch and create a dashboard.

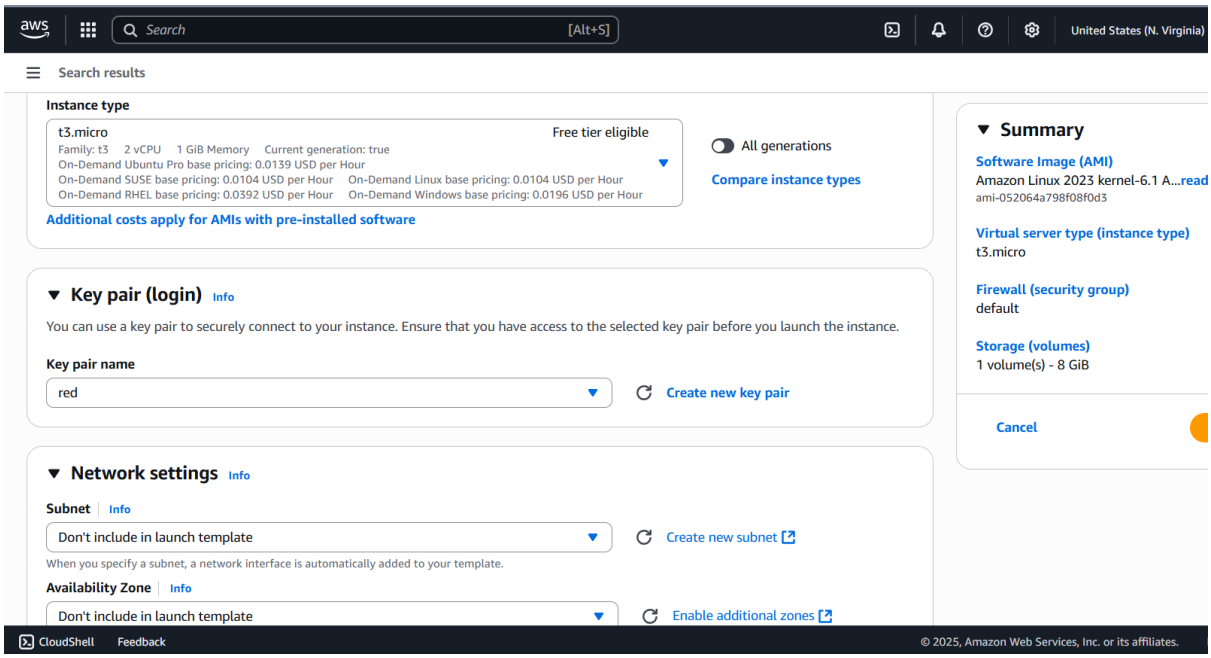
Select the metrics as number,ec2,copy the instance id and paste there and select apache service cpu utilization.

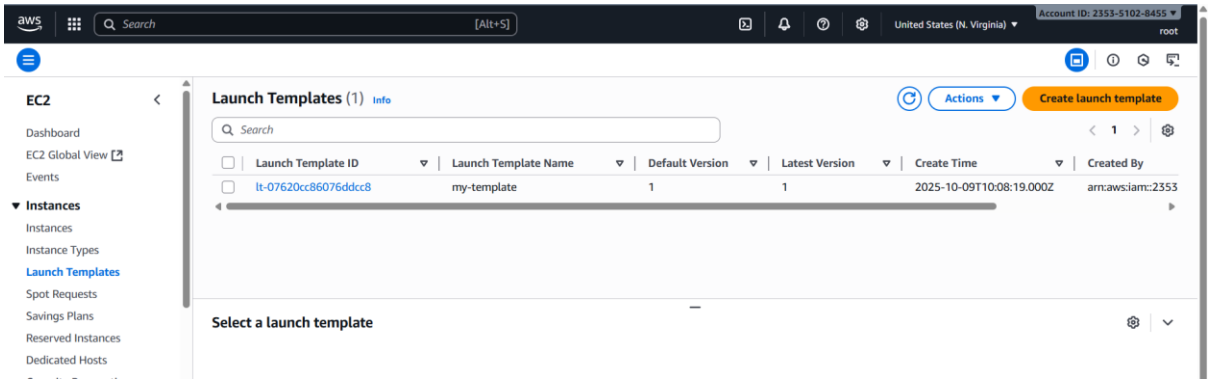


12.If CPU utilization is more than 70%, then it should trigger auto scaling and launch new instance.

Go to templates launch create template

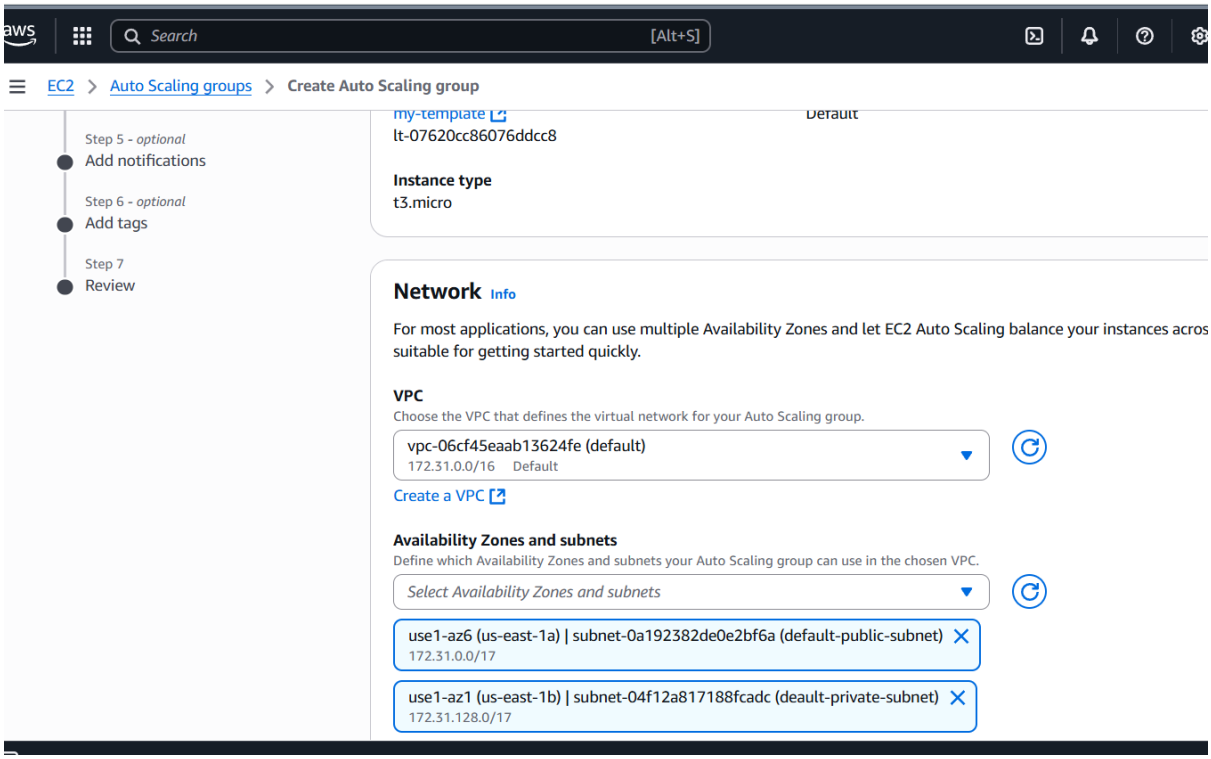


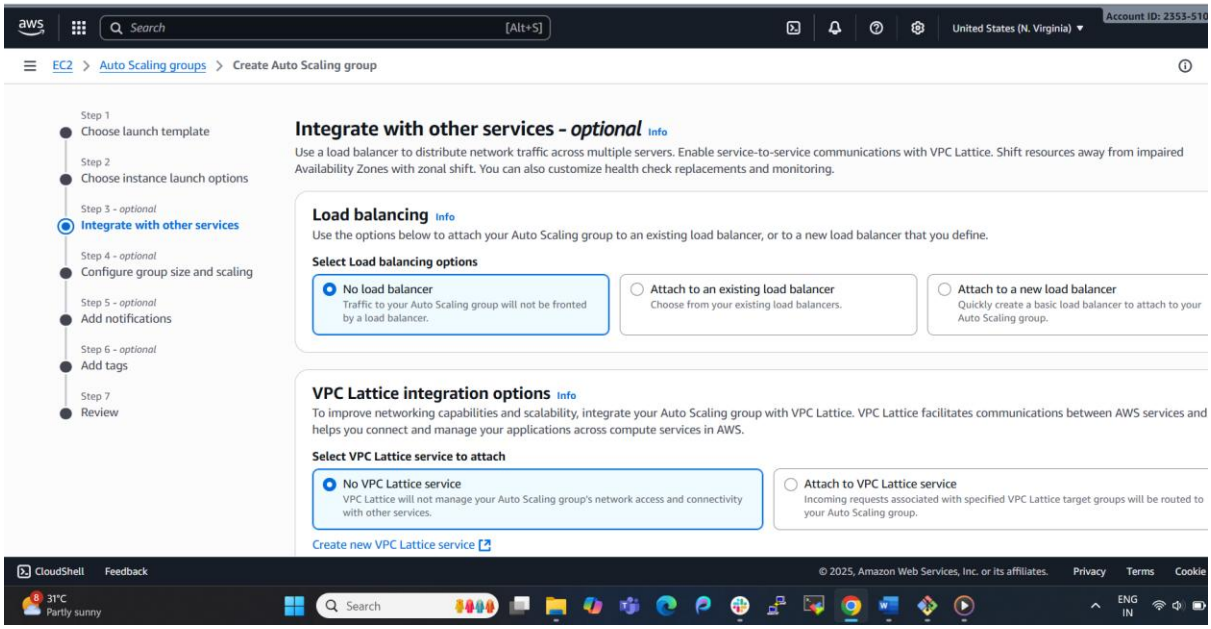




Go to auto scaling groups and create auto scaling groups.

Give the name and select subnets, and give the security groups.





Two instances were created.