<u>PROJECT</u>

Project Title : Two Tier Architecture on AWS using Network Load Balancer

Date : 20/10/2025

Name of the person : Gadi Thirupathi

Name of the project : Two Tier Architecture on AWS using Network Load Balancer

Objective of the project :

- A fully configured VPC
- Ec2 Instances
- public network load balancerand target group with TCP   80 port
- private network load balancer and target group with TCP 8080 port

Steps done in the project :

step 1 :

- create a VPC with the cidr 10.0.0.0/16 give the name tag "my-vpc"
- create a internet gateway and attach to the VPC
- create the subnets in the VPC
- Edit the route table traffic
- Edit the inbound rules in security group

step 2 :

- create four ec2 instances
- 1.web-server01 install the nginx server
- 2.web-server02 install the nginx server
- 3.Apllication-server01 install the tomcat app
- 4.Application-server02 install the tomcat app

Step 3 :

- create the public network load balancer give the name tag "p-nlb",select internet facing select the vpc,subnets.
- create the target group give the name tag "p-tg",give the TCP port 80, select the web-servers include as pending and create tg.
- select the created target group give the port 80 and create load balancer.

step 4 :

- create the private network load balancer give the name tag "pvt-nlb",select internally select the vpc,subnets.
- create the target group give name tag "pvt-tg" give the port 8080 select the application servers include as pending and create tg.
- select the ctrated target group give the tcp 8080 and create load balancer

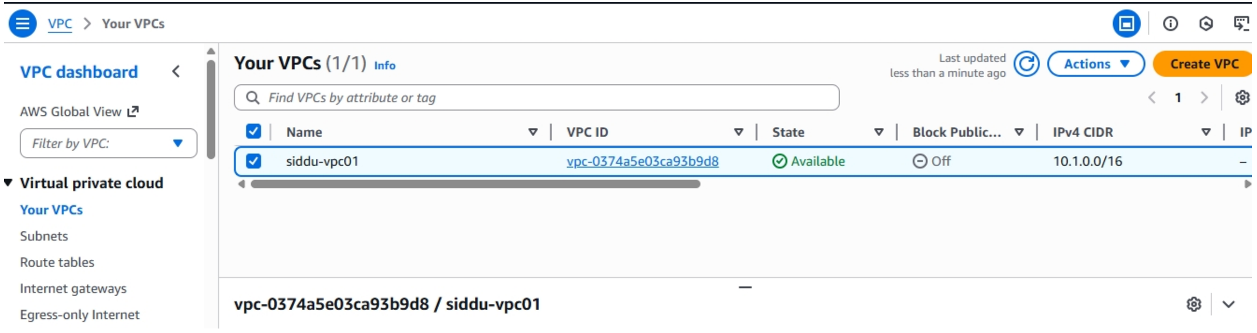
Step 5 :

- Here now we have to connect from webserver01 to App-server01 by the cmd # telnet<pvt>8080
  ===> we can see that our web-server is connected to App-server.

- Here now we have to connect with private network load balancer DNS by the cmd # telnet <pvtlbdns> 8080
  ===> we can see the ouput connected to the app-server

Output screenshots :

==> vpc

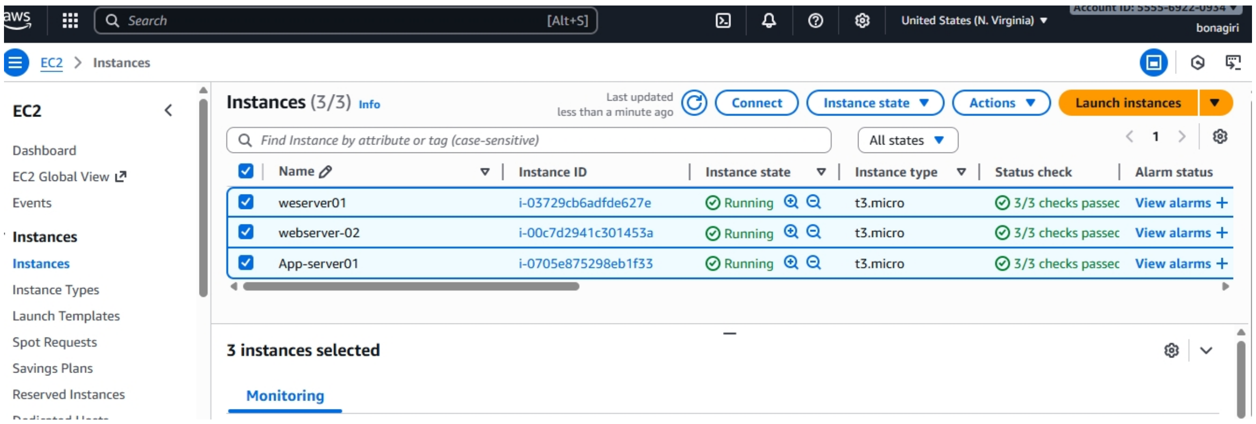


Your VPCs (1/1) Info

Name	VPC ID	State	Block Public...	IPv4 CIDR	IP
siddu-vpc01	vpc-0374a5e03ca93b9d8	Available	Off	10.1.0.0/16	–

vpc-0374a5e03ca93b9d8 / siddu-vpc01

==> Ec2 Instances :



EC2 > Instances

Instances (3/3) Info — Last updated less than a minute ago — Connect | Instance state | Actions | Launch instances

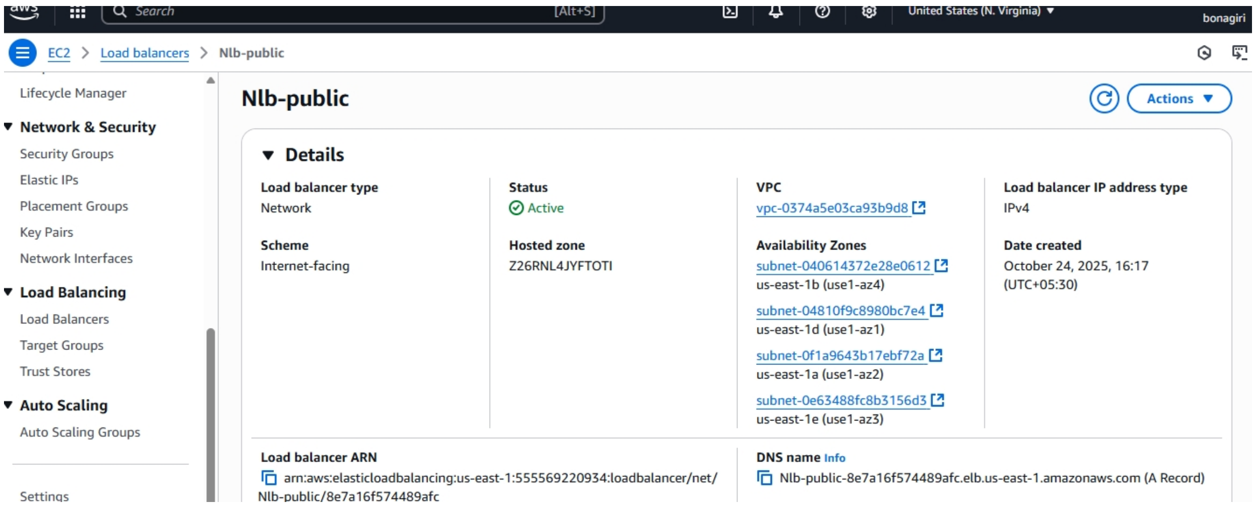
Find Instance by attribute or tag (case-sensitive) — All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
weserver01	i-03729cb6adfde627e	Running	t3.micro	3/3 checks passec	View alarms +
webserver-02	i-00c7d2941c301453a	Running	t3.micro	3/3 checks passec	View alarms +
App-server01	i-0705e875298eb1f33	Running	t3.micro	3/3 checks passec	View alarms +

3 instances selected

Monitoring

==>Public load balancer



EC2 > Load balancers > Nlb-public

Nlb-public

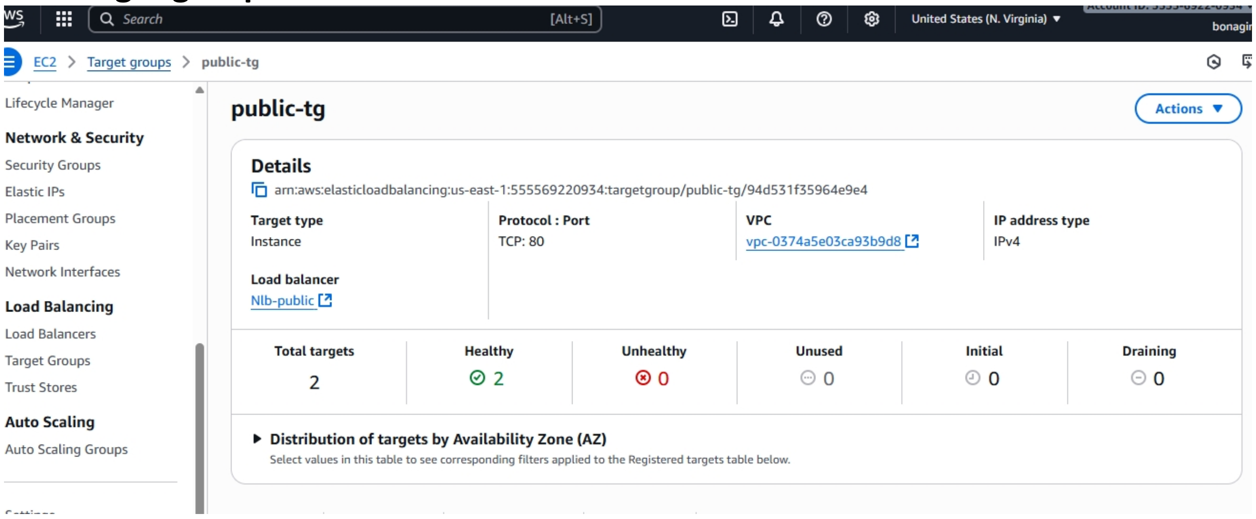
Details

Load balancer type Network	**Status** Active	**VPC** vpc-0374a5e03ca93b9d8	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** Z26RNL4JYFTOTI	**Availability Zones** subnet-040614372e28e0612 us-east-1b (use1-az4); subnet-04810f9c8980bc7e4 us-east-1d (use1-az1); subnet-0f1a9643b17ebf72a us-east-1a (use1-az2); subnet-0e63488fc8b3156d3 us-east-1e (use1-az3)	**Date created** October 24, 2025, 16:17 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:555569220934:loadbalancer/net/Nlb-public/8e7a16f574489afc

DNS name Info
Nlb-public-8e7a16f574489afc.elb.us-east-1.amazonaws.com (A Record)

==> Target group



EC2 > Target groups > public-tg

public-tg

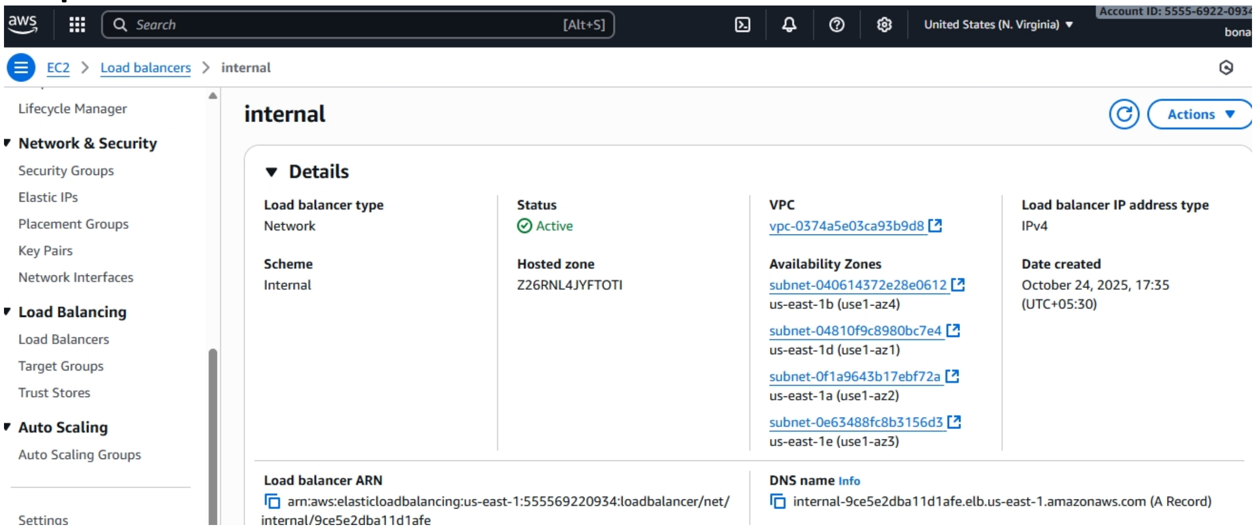
Details
arn:aws:elasticloadbalancing:us-east-1:555569220934:targetgroup/public-tg/94d531f35964e9e4

Target type Instance	**Protocol : Port** TCP: 80	**VPC** vpc-0374a5e03ca93b9d8	**IP address type** IPv4
Load balancer Nlb-public			

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

==>private load baancer



EC2 > Load balancers > internal

internal

Details

Load balancer type
Network

Status
Active

VPC
vpc-0374a5e03ca93b9d8

Load balancer IP address type
IPv4

Scheme
Internal

Hosted zone
Z26RNL4JYFTOTI

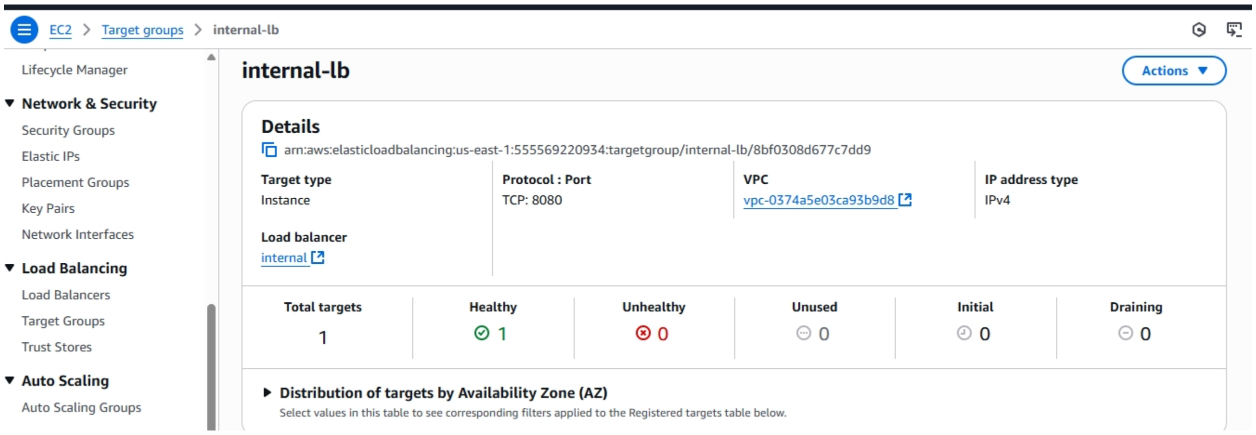
Availability Zones
subnet-040614372e28e0612
us-east-1b (use1-az4)
subnet-04810f9c8980bc7e4
us-east-1d (use1-az1)
subnet-0f1a9643b17ebf72a
us-east-1a (use1-az2)
subnet-0e63488fc8b3156d3
us-east-1e (use1-az3)

Date created
October 24, 2025, 17:35 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:555569220934:loadbalancer/net/internal/9ce5e2dba11d1afe

DNS name Info
internal-9ce5e2dba11d1afe.elb.us-east-1.amazonaws.com (A Record)

==>Target group



EC2 > Target groups > internal-lb

internal-lb

Details
arn:aws:elasticloadbalancing:us-east-1:555569220934:targetgroup/internal-lb/8bf0308d677c7dd9

Target type
Instance

Protocol : Port
TCP: 8080

VPC
vpc-0374a5e03ca93b9d8

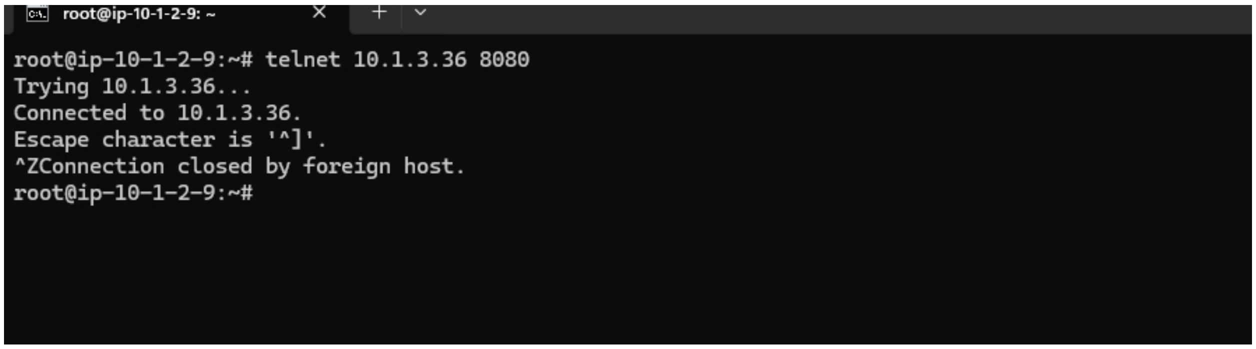
IP address type
IPv4

Load balancer
internal

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	1	0	0	0	0

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

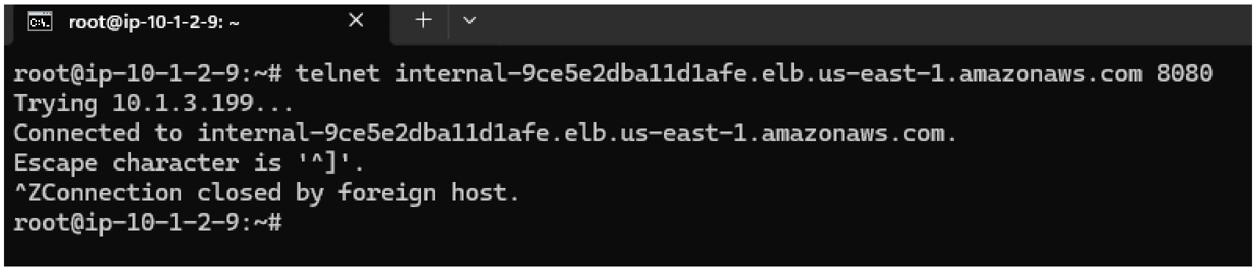
==> connect to the ssh and test connect of web-server to app-server with pvt-ip:



```
root@ip-10-1-2-9: ~

root@ip-10-1-2-9:~# telnet 10.1.3.36 8080
Trying 10.1.3.36...
Connected to 10.1.3.36.
Escape character is '^]'.
^ZConnection closed by foreign host.
root@ip-10-1-2-9:~#
```

==> connect internally by NLB private DNS to App-server



```
root@ip-10-1-2-9: ~

root@ip-10-1-2-9:~# telnet internal-9ce5e2dba11d1afe.elb.us-east-1.amazonaws.com 8080
Trying 10.1.3.199...
Connected to internal-9ce5e2dba11d1afe.elb.us-east-1.amazonaws.com.
Escape character is '^]'.
^ZConnection closed by foreign host.
root@ip-10-1-2-9:~#
```

connected internally sucessfully by private network load balancer DNS.

Tools ans services used:

- **AWS VPC**
- **AWS EC2**
- **AWS LOAD BALANCER**
- **CMD**

Conclusion : In this project succesfully implemented a Two-Tier Architecture using Aws services.consisting of a web tier and application tier. By integrating Load Balancer ensured high availabilty and efficient traffic distribution between servers and having high security through internall connetion from web to app servers.