

Load Balancers-Task

1. Configure Classic Load balancer.

- Classic Load Balancer is an AWS service that distributes incoming traffic across multiple EC2 instances to improve:

- Availability
- Fault tolerance
- Scalability

It works at Layer 4 (TCP) and Layer 7 (HTTP/HTTPS).

➤ Key Features

- Supports **HTTP, HTTPS, TCP, SSL**
- Performs **health checks** on instances
- Routes traffic only to **healthy instances**
- Works with **EC2 Auto Scaling**
- Provides **DNS name** for access
- Can be **internet-facing** or **internal**

➤ CLB Workflow

1. User sends request to CLB DNS.
2. CLB receives request.
3. Health check verifies instances.
4. CLB forwards traffic to **healthy EC2**.
5. If one fails, traffic is sent to others.

➤ Classic Load Balancer – Configuration Steps

- **Step 1: Open CLB**
- AWS Console → **EC2** → **Load Balancers** → **Create Load Balancer**
Choose **Classic Load Balance**.

Load Balancers-Task

The screenshot shows the 'Create Classic Load Balancer' configuration summary page in the AWS Management Console. The page is titled 'Summary' and contains several sections for reviewing and editing configurations:

- Basic configuration:** Name: Classic-LBNeelima, Scheme: Internet-facing.
- Network mapping:** VPC: vpc-0ac34f90e35f5b6d, Availability Zones and subnets: us-east-1a (subnet-00f0a7d1fb68fc5a1, public-Neelima), us-east-1b (subnet-07133be4bd6d0803c, pub-12).
- Security groups:** default (sg-06146d7aa2627d729).
- Listeners and routing:** HTTP:80.
- Health checks:** HTTP:80/index.html, Timeout: 2 seconds, Interval: 5 seconds, Unhealthy threshold: 2, Healthy threshold: 10.
- Instances:** No instances added yet.
- Attributes:** Cross-zone load balancing: On, Connection draining: On, Connection draining timeout: 300 seconds.
- Tags:** -

The bottom of the console shows the navigation bar with the AWS logo, search bar, and various utility links like CloudShell, Feedback, and Console Mobile App.

- **Add EC2 Instances**
- Select your running EC2
 - Click **Add to Registered**

The screenshot shows the 'Manage instances' page for a Classic Load Balancer named 'CLB-Neelima'. The page displays a table of instances and a 'Review selected instances' section.

Registration status	Instance ID	Name	State	Health
<input checked="" type="checkbox"/> Not registered	i-0fa9741c0d9c020b0	CLB-Neelima	Running	default

Review selected instances (1/1)

The instances being registered, or remaining registered. Remove instances by selecting them, then choosing Deselect.

☒ Registration status | Instance ID | Name | State | Health

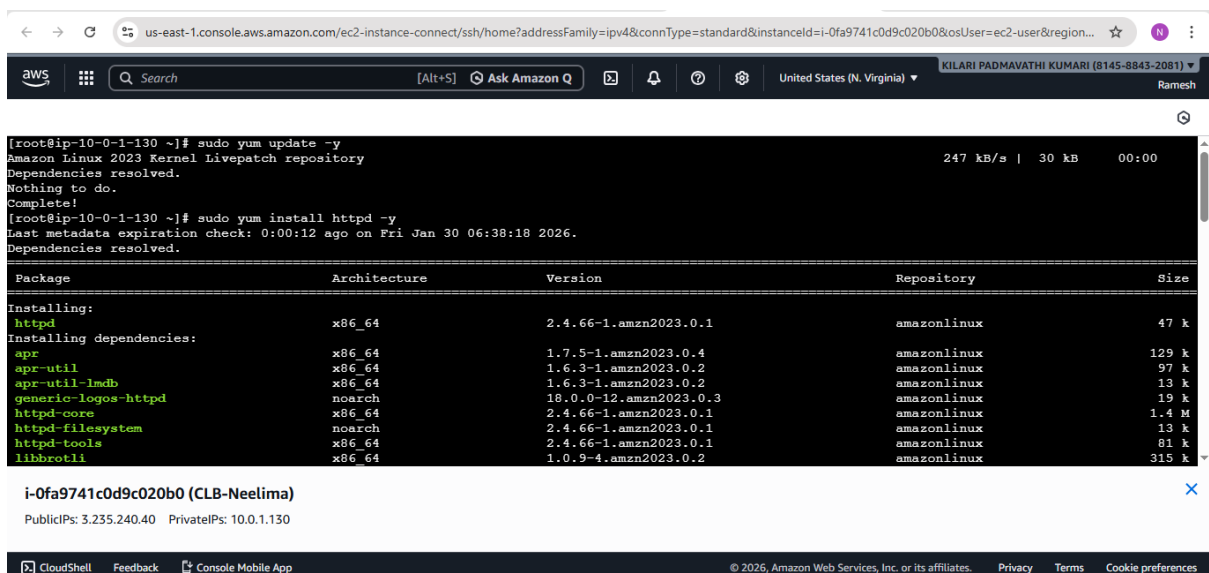
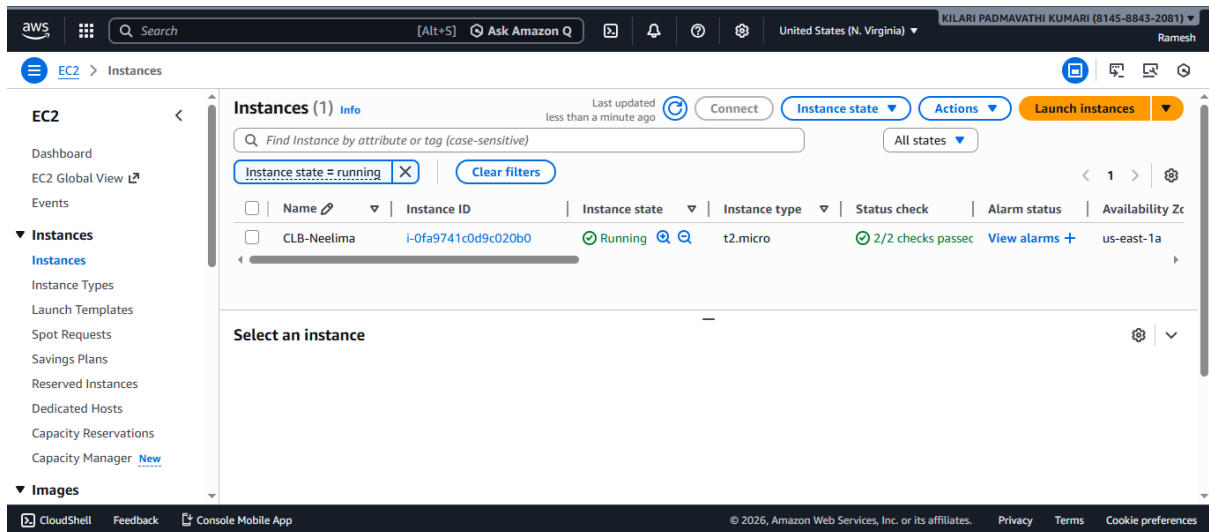
☒ Not registered | i-0fa9741c0d9c020b0 | CLB-Neelima | Running | -

Summary of changes
1 instance(s) will be registered.

[Cancel](#) [Save changes](#)

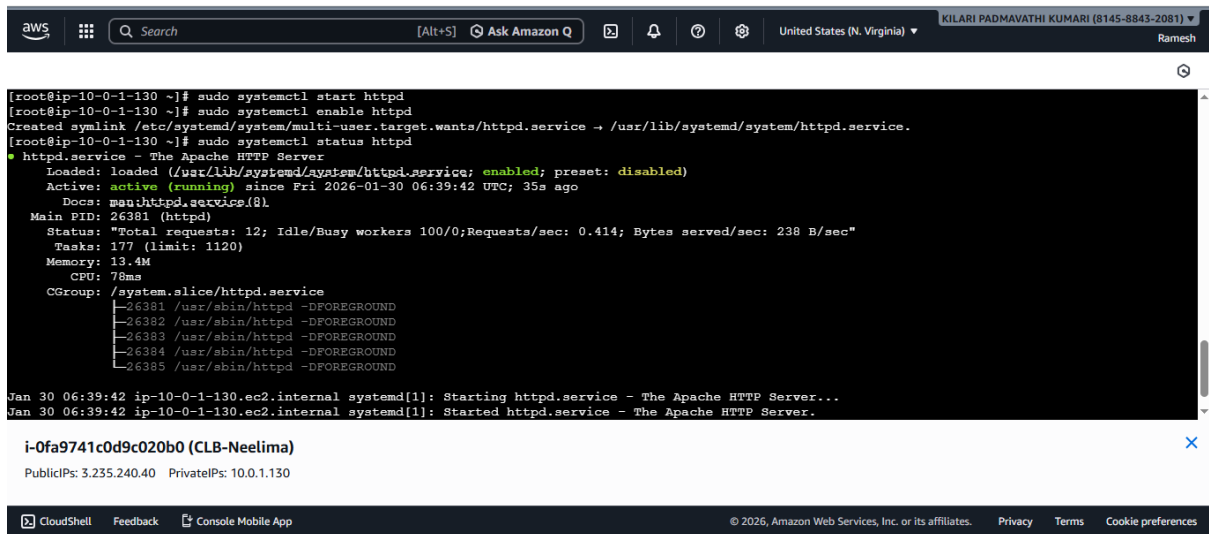
Load Balancers-Task

- Go to AWS console
- Create a EC2 instance



- Connect instance through ssh
- Switch to root user for more privileges
- Install httpd service in EC2
- Commands are as follows:
- Sudo yum update -y
- Sudo yum install httpd -y

Load Balancers-Task



```
[root@ip-10-0-1-130 ~]# sudo systemctl start httpd
[root@ip-10-0-1-130 ~]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-1-130 ~]# 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 Fri 2026-01-30 06:39:42 UTC; 35s ago
     Docs: man:httpd.service(8)
   Main PID: 26381 (httpd)
   Status: "Total requests: 12; Idle/Busy workers 100/0; Requests/sec: 0.414; Bytes served/sec: 238 B/sec"
   Tasks: 177 (limit: 1120)
  Memory: 13.4M
    CPU: 78ms
   CGroup: /system.slice/httpd.service
           └─26381 /usr/sbin/httpd -DFOREGROUND
             └─26382 /usr/sbin/httpd -DFOREGROUND
               └─26383 /usr/sbin/httpd -DFOREGROUND
                 └─26384 /usr/sbin/httpd -DFOREGROUND
                   └─26385 /usr/sbin/httpd -DFOREGROUND

Jan 30 06:39:42 ip-10-0-1-130.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Jan 30 06:39:42 ip-10-0-1-130.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
```

i-0fa9741c0d9c020b0 (CLB-Neelima)

PublicIPs: 3.235.240.40 PrivateIPs: 10.0.1.130

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Sudo systemctl start httpd
- Sduo systemctl enable httpd
- Sudo systemctl status httpd (check it is running or not)

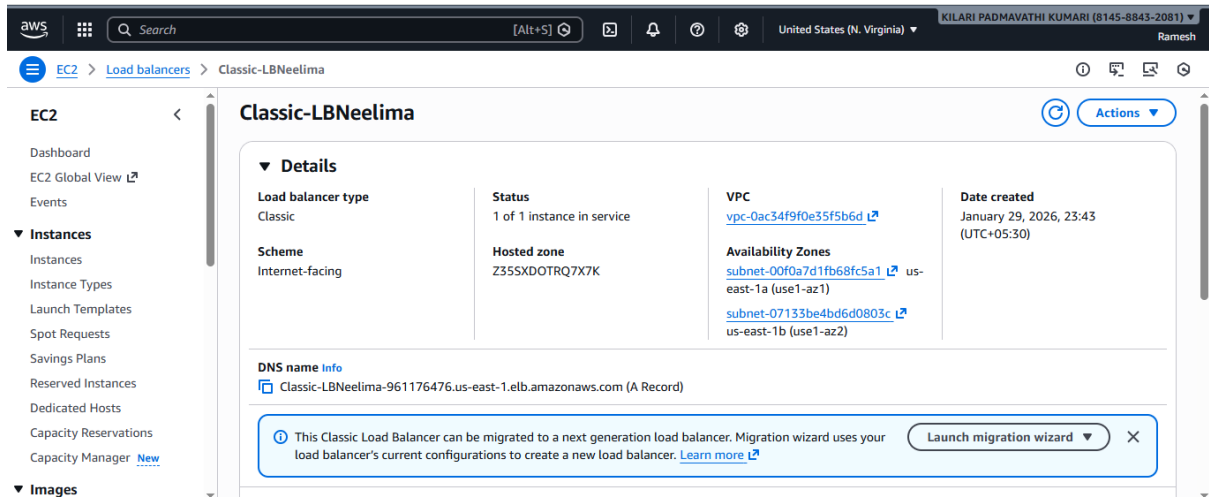
```
[root@ip-10-0-1-130 ~]# echo "<h1>CLB Working - Neelima</h1>" | sudo tee /var/www/html/index.html
<h1>CLB Working - Neelima</h1>
[root@ip-10-0-1-130 ~]# echo "<h1>CLB Working - Neelima</h1>" | sudo tee /var/www/html/index.html
<h1>CLB Working - Neelima</h1>
```

- Create a static webpage in ec2 command as follows
- echo "<h1>CLB working – Neelima</h1> |sudo tee /var/www/html/index.html
- check whether it is running in local machine with public ip
- Locally it is working

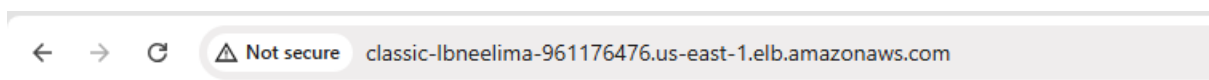


CLB Working - Neelima

Load Balancers-Task



- Go ec2 → Load balancers
- Click on created load balancer
- Copy DNS name paste link in browser and check for output.



CLB Working - Neelima

- “The Classic Load Balancer DNS acts as a single public endpoint. It receives client requests, performs health checks, and forwards traffic to healthy EC2 instances based on the configured listener rules.”

Load Balancers-Task

2. Configure Application Load balancer.

An ALB is a type of AWS Load Balancer that operates at the Layer 7 (Application Layer) of the OSI model. This means it can route traffic based on content of the request, such as:

- Hostname (Host header)
- URL path (/images, /api)
- HTTP headers, query strings, or source IP

Key features:

- Supports HTTP and HTTPS (web traffic).
- Content-based routing: Sends requests to different target groups based on URL or hostname.
- Works with EC2, ECS, Lambda, IP addresses as target

When to use ALB:

- You have multiple web applications on the same domain or server.
- You want to route /api requests to one service and /web requests to another.
- You need SSL termination and security policies at the load balancer level.

Key Components of ALB

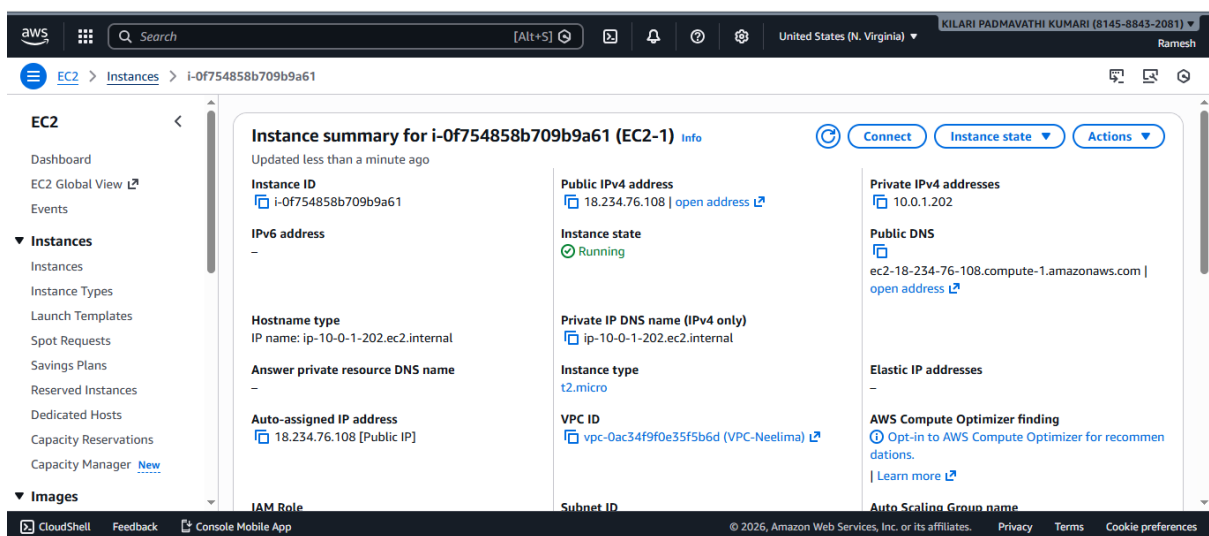
1. **Load Balancer** – The front-end endpoint that client's access.
2. **Listeners** – Check for incoming traffic on a port (usually 80 for HTTP or 443 for HTTPS).
3. **Target Groups** – Logical group of backend servers (EC2 instances, IPs, Lambda).
4. **Rules** – Define how requests are routed to different target groups.
5. **Health Checks** – Ensure traffic is sent only to healthy targets.

Load Balancers-Task

Configuration Steps in AWS Console

Step 1: Launch EC2 Instances

- Launch **2 or more EC2 instances** with a web server (Apache/Nginx).
- Make sure they are in **different subnets** (preferably in different AZs for high availability).
- Go to AWS search Ec2 instance
- Launch instance with public ip



- Connect to instance through ssh connection
- Install apache in EC2-1 instance commands are as follows
 - Sudo yum update -y
 - sudo yum install httpd -y



Load Balancers-Task

```
Complete!
[ec2-user@ip-10-0-1-202 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-1-202 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-10-0-1-202 ~]$ 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 Fri 2026-01-30 10:35:34 UTC; 1min 15s ago
     Docs: man:httpd.service(8)
   Main PID: 26935 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
     Tasks: 177 (limit: 1120)
    Memory: 12.9M
       CPU: 102ms
    CGroup: /system.slice/httpd.service
            └─26935 /usr/sbin/httpd -DFOREGROUND
              └─26957 /usr/sbin/httpd -DFOREGROUND
                └─26959 /usr/sbin/httpd -DFOREGROUND
```

i-Of754858b709b9a61 (EC2-1)

PublicIPs: 18.234.76.108 PrivateIPs: 10.0.1.202

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- sudo systemctl start httpd
- sudo systemctl enable httpd
- sudo systemctl status httpd

```
[ec2-user@ip-10-0-1-202 ~]$ echo "<h1>Welcome to Apache_Neelima</h1>" |sudo tee /var/www/html/index.html
<h1>Welcome to Apache_Neelima</h1>
[ec2-user@ip-10-0-1-202 ~]$
```

i-Of754858b709b9a61 (EC2-1)

PublicIPs: 18.234.76.108 PrivateIPs: 10.0.1.202

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Create webpage in Instance using html
- echo “<h1> message </h1> |sudo tee /var/www/html/index.html



- After webpage creation check locally using public ip

Load Balancers-Task

- Go to AWS search Ec2 instance
- Launch instance(EC2-2) with public ip

Instance summary for i-0ab987486937cdcb8 (Ec2-2) Info

Updated less than a minute ago

Instance ID
i-0ab987486937cdcb8

Public IPv4 address
100.53.141.240 | [open address](#)

Private IPv4 addresses
10.0.3.129

Instance state
Running

Public DNS
ec2-100-53-141-240.compute-1.amazonaws.com | [open address](#)

IPv6 address
-

Private IP DNS name (IPv4 only)
ip-10-0-3-129.ec2.internal

Instance type
t2.micro

Hostname type
IP name: ip-10-0-3-129.ec2.internal

Answer private resource DNS name
-

VPC ID
vpc-0ac34f9f0e35f5b6d (VPC-Neelima)

Auto-assigned IP address
100.53.141.240 [Public IP]

Elastic IP addresses
-

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)

IAM Role
-

Subnet ID
-

Auto Scaling Group name
-

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-3-129 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-3-129 ~]$ sudo yum install nginx -y
Last metadata expiration check: 0:00:52 ago on Fri Jan 30 10:20:49 2026.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
nginx	x86_64	1:1.24.0-1.el9	Amazon Linux 2023	1.2 MB

i-0ab987486937cdcb8 (Ec2-2)

PublicIPs: 100.53.141.240 PrivateIPs: 10.0.3.129

- Connect to instance using ssh connection
- Install Nginx in ec2 instance and commands are as follows
 - sudo yum update -y
 - sudo yum install nginx -y

Load Balancers-Task

```
Complete!
[ec2-user@ip-10-0-3-129 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-3-129 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-3-129 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Fri 2026-01-30 10:25:46 UTC; 36s ago
     Main PID: 26598 (nginx)
       Tasks: 2 (limit: 1120)
      Memory: 2.5M
         CPU: 39ms
    CGroup: /system.slice/nginx.service
            └─26598 "nginx: master process /usr/sbin/nginx"
              └─26600 "nginx: worker process"
```

i-0ab987486937cdcb8 (Ec2-2)

PublicIPs: 100.53.141.240 PrivateIPs: 10.0.3.129

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms

- sudo systemctl start nginx
- sudo systemctl enable nginx
- sudo systemctl status nginx

```
[ec2-user@ip-10-0-3-129 ~]$ echo "<h1>welcome to Nginx_app-LB-Neelima</h1>" | sudo tee /usr/share/nginx/html/index.html
<h1>welcome to Nginx_app-LB-Neelima</h1>
[ec2-user@ip-10-0-3-129 ~]$
```

- Create a webpage in ece instance using html
- echo "<h1>Message</h1>" | sudo tee /usr/share/nginx/html/index.html

← → ↻ ⚠ Not secure 100.53.141.240

welcome to Nginx_app-LB-Neelima

- Test webpage using public ip in local machine.

Load Balancers-Task

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTargetGroup:

aws Search [Alt+S] United States (N. Virginia) KILARI PADMAVATHI KUMARI (8145-8843-2081) Ramesh

EC2 > Target groups > Create target group

Indicate what resource type you want to target. Only the selected resource type can be registered to this target group.

- ☒ **Instances**
Supports load balancing to instances in a VPC. Integrate with Auto Scaling Groups or ECS services for automatic management.
Suitable for: ALB NLB GWLB
- ☐ **IP addresses**
Supports load balancing to VPC and on-premises resources. Facilitates routing to IP addresses and network interfaces on the same instance. Supports IPv6 targets.
Suitable for: ALB NLB GWLB
- ☐ **Lambda function**
Supports load balancing to a single Lambda function. ALB required as traffic source.
Suitable for: ALB
- ☐ **Application Load Balancer**
Allows use of static IP addresses and PrivateLink with an Application Load Balancer. NLB required as traffic source.
Suitable for: NLB

Target group name
Name must be unique per Region per AWS account.
Test-ALB
Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 8/32

Protocol
Protocol for communication between the load balancer and targets.
HTTP

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Now Go to EC2 → Target Groups → create target groups
- Select instances
- Enter Target group Name and enter details and
- click on Next

aws Search [Alt+S] United States (N. Virginia) KILARI PADMAVATHI KUMARI (8145-8843-2081) Ramesh

EC2 > Target groups > Create target group

● Create target group
Step 2 - recommended
Register targets
Step 3
○ Review and create

Register targets - recommended
This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter instances

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups
<input checked="" type="checkbox"/>	i-0ab987486937cdcb8	Ec2-2	Running	default
<input checked="" type="checkbox"/>	i-0f754858b709b9a61	EC2-1	Running	default

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- In Register targets Add Instances we required.
- Ports for the selected instances click on included as

Load Balancers-Task

Health check details

Health check protocol HTTP	Health check path /	Health check port traffic-port	Interval 30 seconds
Timeout 5 seconds	Healthy threshold 5	Unhealthy threshold 2	Success codes 200

Step 2: Register targets [Edit](#)

Targets (2)

Instance ID	Name	Port	Zone
i-0ab987486937cdcb8	Ec2-2	80	us-east-1b
i-0f754858b709b9a61	EC2-1	80	us-east-1a

[Cancel](#) [Previous](#) [Create target group](#)

- Click on Create target group

Test-ALB [Actions](#)

Details

[arn:aws:elasticloadbalancing:us-east-1:814588432081:targetgroup/Test-ALB/1142ba9826245ad4](#)

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0ac34f9f0e35f5b6d
IP address type IPv4	Load balancer None associated		

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

0 Anomalous

- The above image shows Target groups created successfully and we can information in details.

Load Balancers-Task

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► **How Application Load Balancers work**

Basic configuration

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

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**

- Serves Internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.

☐ **Internal**

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.

- Now, Create Application Load Balancer
- EC2 → Load Balancers → create Application Load Balancer
- In Basic Configuration → Enter Load Balancer Name
- Scheme → Internet-facing
- Add VPC and Subnets, Security groups, listeners and routing

Summary

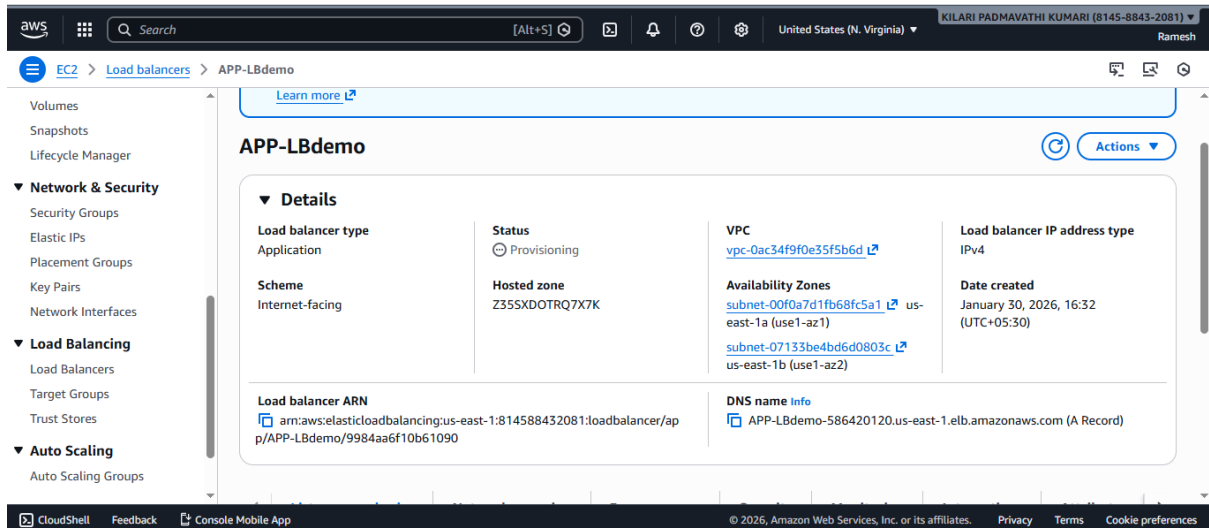
Review and confirm your configurations. [Estimate cost](#)

Basic configuration Edit Name: APP-LBdemo Scheme: Internet-facing IP address type: IPv4	Network mapping Edit VPC: vpc-0ac34f90e35f5b6d Public IPv4 IPAM pool: - Availability Zones and subnets: <ul style="list-style-type: none">us-east-1a subnet-00f0a7d1fb68fc5a1 public-Neelimaus-east-1b subnet-07133be4bd6d0803c pub-12	Security groups Edit default sg-06146d7aa2627d729	Listeners and routing Edit HTTP:80 Forward to 1 target group
Service integrations Edit Amazon CloudFront + AWS Web Application Firewall (WAF): - AWS WAF: - AWS Global Accelerator: -		Tags Edit -	

Attributes

- In summary we see complete configurations.
- Click on create.

Load Balancers-Task



- Click on App-LBdemo create before.
- In details we have DNS name,copy it
- Paste DNS name in Browser and check output.



Conclusion

An Application Load Balancer (ALB) allows us to distribute incoming web traffic to multiple backend servers. In this setup, we successfully configured:

- One EC2 instance running Apache other EC2 instance running Nginx
- Both accessed through a single Application Load Balancer

Load Balancers-Task

- The ALB forwards each request to **only one server at a time**, using a round-robin method. That is why refreshing the ALB URL shows either the Apache page or the Nginx page.
- This proves that the **Application Load Balancer efficiently manages and routes traffic**, improves availability, and allows multiple applications to run behind a single endpoint.

Load Balancers-Task

3. Configure Network Load balancer.

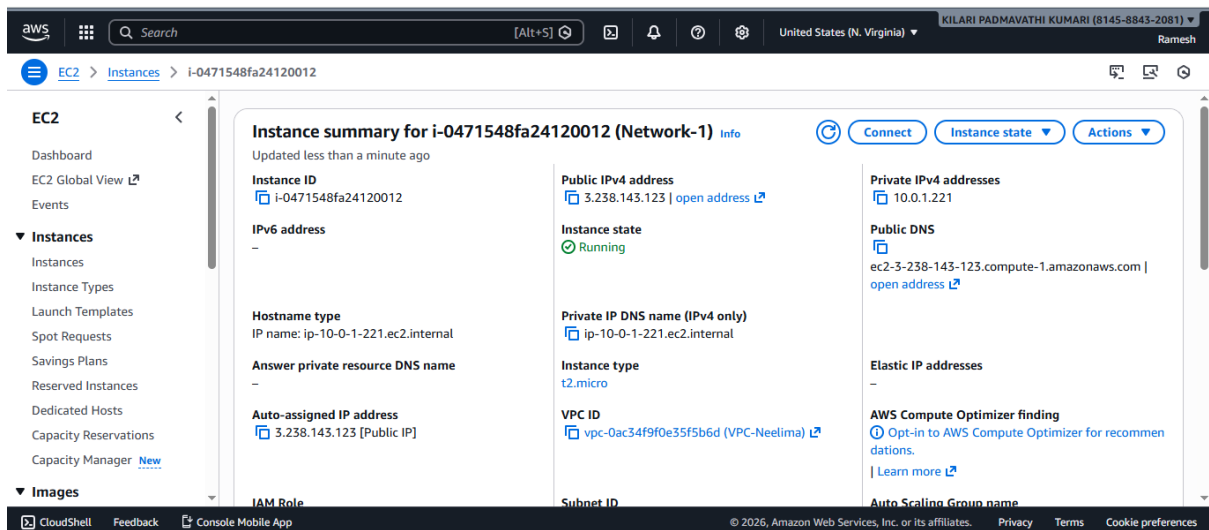
A Network Load Balancer (NLB) operates at Layer 4 (TCP/UDP) of the OSI model. It forwards traffic based on IP address and port, not on HTTP content.

- It is used for:
- Very high traffic
 - Low latency
 - TCP, UDP, TLS based applications
 - When you need static IP addresses

Configuration Steps

Step 1 – Launch EC2 Instances

- Create 2 EC2 instances
- Install any server (Apache / Nginx / Tomcat)
- Make sure both are in the same VPC



Load Balancers-Task

[illegible]

aws

Search

[Alt+S]

United States (N. Virginia)

KILARI PADMAVATHI KUMARI (\$145-8843-2081)

Ramesh

```
[root@ip-10-0-1-221 ~]# sudo systemctl start httpd
[root@ip-10-0-1-221 ~]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-1-221 ~]# 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 Fri 2026-01-30 13:57:51 UTC; 28s ago
     Docs: man:httpd.service(8).
 Main PID: 26466 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
       Tasks: 177 (limit: 1120)
      Memory: 12.9M
         CPU: 73ms
    CGroup: /system.slice/httpd.service
            └─26466 /usr/sbin/httpd -DFOREGROUND
              └─26467 /usr/sbin/httpd -DFOREGROUND
                └─26468 /usr/sbin/httpd -DFOREGROUND
                  └─26469 /usr/sbin/httpd -DFOREGROUND
                    └─26470 /usr/sbin/httpd -DFOREGROUND

Jan 30 13:57:51 ip-10-0-1-221.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Jan 30 13:57:51 ip-10-0-1-221.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
```

i-0471548fa24120012 (Network-1)

Public IPs: 3.238.143.123 Private IPs: 10.0.1.221

```
[root@ip-10-0-1-221 ~]# echo "<h1>Welcome to NLB-Neelima</h1>" |sudo tee /var/www/html/index.html
<h1>Welcome to NLB-Neelima</h1>
[root@ip-10-0-1-221 ~]#
```

i-0471548fa24120012 (Network-1)

PublicIPs: 3.238.143.123 PrivateIPs: 10.0.1.221

CloudShell Feedback Console Mobile App © 2026, Amazon Web S

← → ↻ ⚠ Not secure 3.238.143.123

Welcome to NLB-Neelima

Load Balancers-Task

EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Capacity Manager

Images

Instance summary for i-0fddb8d38ff422021 (Network-2)

Updated less than a minute ago

Instance ID

i-0fddb8d38ff422021

IPv6 address

-

Hostname type

IP name: ip-10-0-3-247.ec2.internal

Answer private resource DNS name

-

Auto-assigned IP address

54.91.182.172 [Public IP]

IAM Role

-

Public IPv4 address

54.91.182.172 | open address

Instance state

Running

Private IP DNS name (IPv4 only)

ip-10-0-3-247.ec2.internal

Instance type

t2.micro

VPC ID

vpc-0ac34f90e35f5b6d (VPC-Neelima)

Subnet ID

-

Private IPv4 addresses

10.0.3.247

Public DNS

ec2-54-91-182-172.compute-1.amazonaws.com | open address

Elastic IP addresses

-

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name

-

Connect

Instance state

Actions

Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

```
[ec2-user@ip-10-0-3-247 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-3-247 ~]$ sudo yum install httpd -y
Last metadata expiration check: 0:00:18 ago on Fri Jan 30 14:07:02 2026.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				

i-0fddb8d38ff422021 (Network-2)

PublicIPs: 54.91.182.172 PrivateIPs: 10.0.3.247

```
[ec2-user@ip-10-0-3-247 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-3-247 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-10-0-3-247 ~]$ 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 Fri 2026-01-30 14:08:32 UTC; 36s ago
     Docs: man:httpd.service(8)
  Main PID: 26597 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
   Tasks: 177 (limit: 1120)
  Memory: 12.9M
    CPU: 71ms
   CGroup: /system.slice/httpd.service
           └─26597 /usr/sbin/httpd -DFOREGROUND
           └─26598 /usr/sbin/httpd -DFOREGROUND
           └─26599 /usr/sbin/httpd -DFOREGROUND
           └─26600 /usr/sbin/httpd -DFOREGROUND
```

i-0fddb8d38ff422021 (Network-2)

PublicIPs: 54.91.182.172 PrivateIPs: 10.0.3.247

```
[ec2-user@ip-10-0-3-247 ~]$ echo "<h1>Welcome to NLB-Neelima</h1>" | sudo tee /var/www/html/index.html
<h1>Welcome to NLB-Neelima</h1>
[ec2-user@ip-10-0-3-247 ~]$
```

i-0fddb8d38ff422021 (Network-2)

PublicIPs: 54.91.182.172 PrivateIPs: 10.0.3.247

Load Balancers-Task

←

→

↺

⚠ Not secure

54.91.182.172

Welcome to NLB-Neelima

aws

Search

[Alt+S]

United States (N. Virginia)

KILARI PADMAVATHI KUMARI (8145-8843-2081)

Ramesh

EC2 > Target groups > Create target group

Step 2 - recommended

Register targets

Step 3

Review and create

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter instances

< 1 >

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups
<input checked="" type="checkbox"/>	i-0fddb8d38ff422021	Network-2	Running	default
<input checked="" type="checkbox"/>	i-0471548fa24120012	Network-1	Running	default

2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include as pending below

aws

Search

[Alt+S]

United States (N. Virginia)

KILARI PADMAVATHI KUMARI (8145-8843-2081)

Ramesh

EC2 > Target groups > NLBdemo

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Load Balancers

Target Groups

Trust Stores

Auto Scaling

Auto Scaling Groups

Settings

Successfully created the target group: NLBdemo.

Give feedback

Actions

Details

am:aws:elasticloadbalancing:us-east-1:814588432081:targetgroup/NLBdemo/be7227890a136819

Target type

Instance

Protocol : Port

TCP: 80

VPC

ypc-0ac34f9f0e35f5b6d

IP address type

IPv4

Load balancer

None associated

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

Load Balancers-Task

aws

Search

[Alt+S]

United States (N. Virginia)

KILARI PADMAVATHI KUMARI (8145-8843-2081)

Ramesh

EC2

Load balancers

Create Network Load Balancer

Basic configuration

Load balancer name

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

NLB-Neelima

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

Scheme

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

☒ Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

☐ Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.

Load balancer IP address type

[Info](#)

Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types.

☒ IPv4

- Includes only IPv4 addresses.

☐ Dualstack

- Includes IPv4 and IPv6 addresses.

Search

[Alt+S]

United States (N. Virginia)

KILARI PADMAVATHI KUMARI (8145-8843-2081)

Ramesh

EC2

Load balancers

Create Network Load Balancer

Review

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose Create load balancer.

Summary

Review and confirm your configurations. [Estimate cost](#)

Basic configuration [Edit](#)

Name: NLB-Neelima

Scheme: Internet-facing

IP address type: IPv4

Network mapping [Edit](#)

VPC: [vpc-0ac34f9f0e35f5b6d](#)

Availability Zones and subnets:

- us-east-1a
[subnet-00f0a7d1fb68fc5a1](#)
public-Neelima
- us-east-1b
[subnet-07133be4bd6d0803c](#)
pub-12

Security groups [Edit](#)

default

[sg-06146d7aa2627d729](#)

Listeners and routing [Edit](#)

TCP:80 | Forward to 1 target group

Service integrations [Edit](#)

AWS Global Accelerator: -

Tags [Edit](#)

-

Attributes

CloudShell

Feedback

Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates.

Privacy

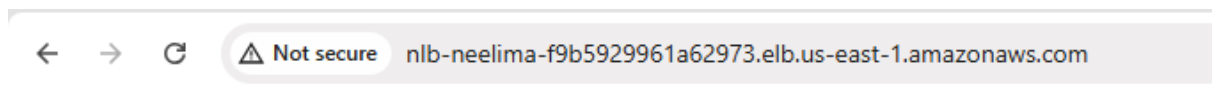
Terms

Cookie preferences

The screenshot displays the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, search bar, user profile (Ramesh), and location (United States (N. Virginia)). Below this, the breadcrumb trail shows 'EC2 > Load balancers > NLB-Neelima'. The left-hand navigation pane lists categories like Network & Security, Load Balancing, and Auto Scaling. The main content area is titled 'NLB-Neelima' and contains a 'Details' section. This section provides information about the load balancer type (Network), status (Active), VPC ID, availability zones, hosted zone, DNS name info, and date created.

Details	
Load balancer type Network	Status ✔ Active
Scheme Internet-facing	Hosted zone Z26RNLA4JYFTOTI
VPC vpc-0ac34f9f0e35f5b6d	Availability Zones subnet-00fa7d1fb68fc5a1 us-east-1a (use1-az1) subnet-07133be4bd6d0803c us-east-1b (use1-az2)
Load balancer IP address type IPv4	Date created January 30, 2026, 19:55 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:814588432081:loadbalancer/net/NLB-Neelima/f9b5929961a62973	
DNS name Info NLB-Neelima-f9b5929961a62973.elb.us-east-1.amazonaws.com (A Record)	

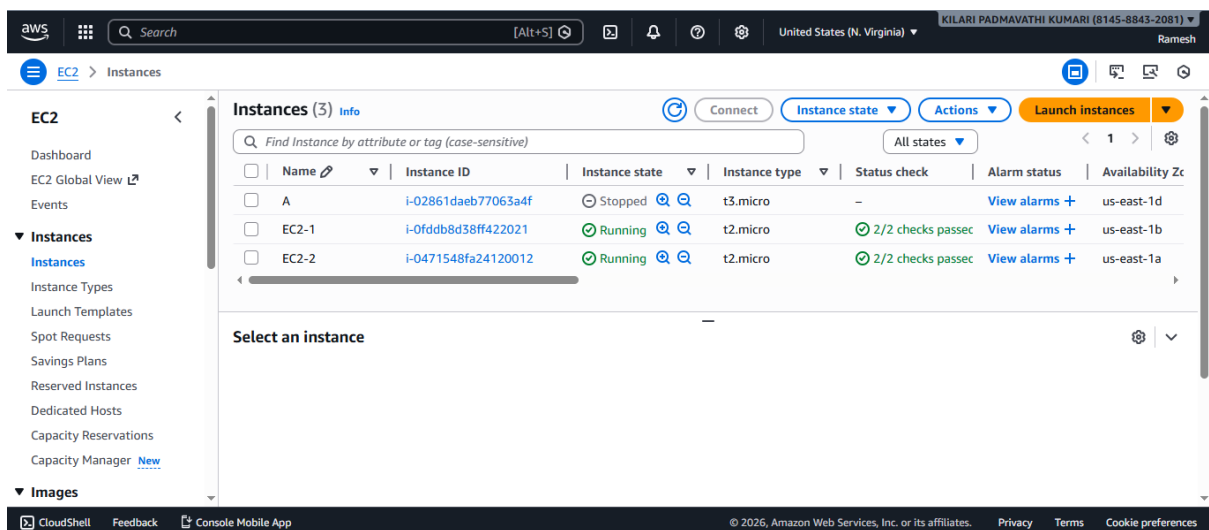
Load Balancers-Task



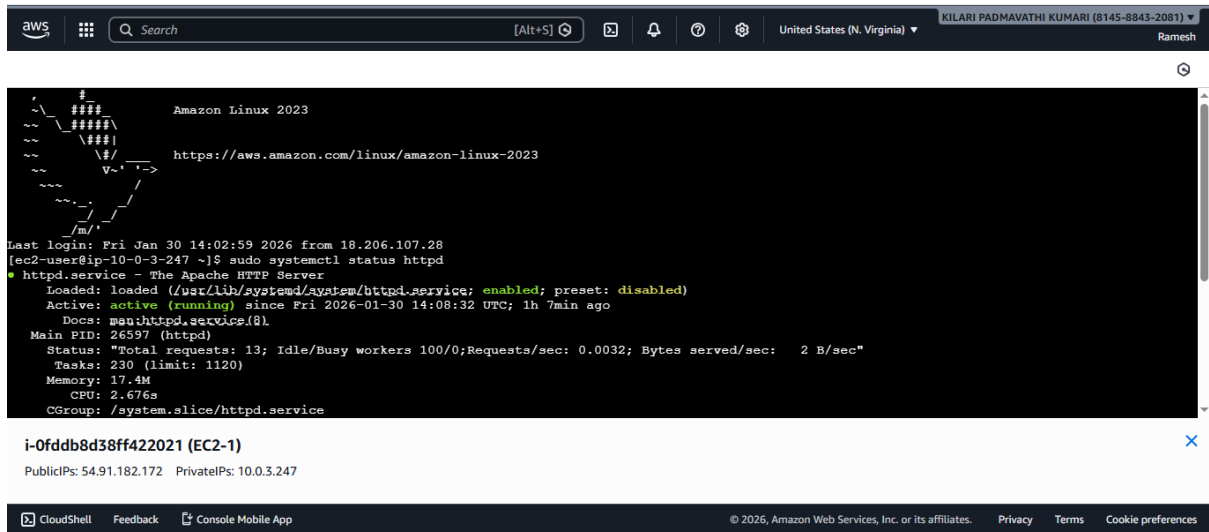
Welcome to NLB-Neelima

4. Attach SSL for application load balancer.

- Create Instances with public ip
- Make sure instances are running or not



Load Balancers-Task



The screenshot shows the AWS CloudShell interface. At the top, there's a search bar and navigation icons. The main terminal window displays the following output:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

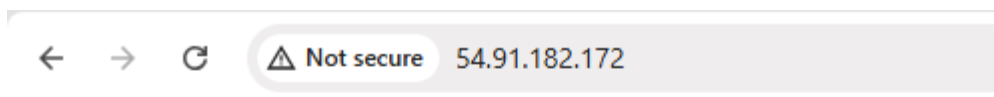
Last login: Fri Jan 30 14:02:59 2026 from 18.206.107.28
[ec2-user@ip-10-0-3-247 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Fri 2026-01-30 14:08:32 UTC; 1h 7min ago
     Docs: man:httpd.service(8)
    Main PID: 26597 (httpd)
   Status: "Total requests: 13; Idle/Busy workers 100/0; Requests/sec: 0.0032; Bytes served/sec: 2 B/sec"
     Tasks: 230 (limit: 1120)
    Memory: 17.4M
       CPU: 2.676s
    CGroup: /system.slice/httpd.service
```

Below the terminal output, the instance details are shown: i-Ofddb8d38ff422021 (EC2-1) with Public IPs: 54.91.182.172 and Private IPs: 10.0.3.247.

- Connect instance with ssh through public ip
- Instance must contain apache.
- Make apache is active and running.

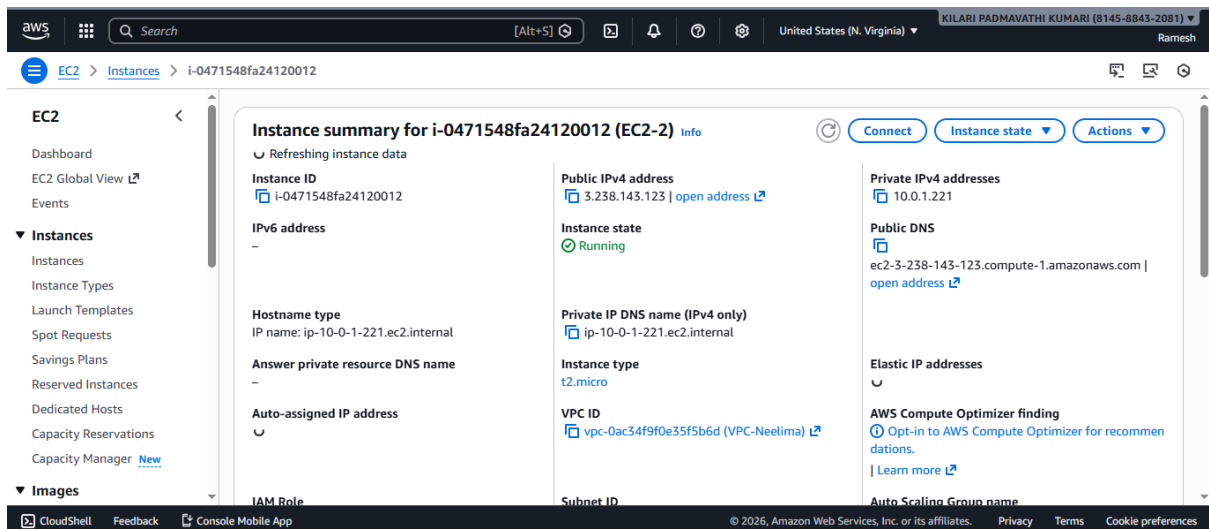
```
[ec2-user@ip-10-0-3-247 ~]$ echo "<h1>Welcome to App_LB-Neelima</h1>" |sudo tee /var/www/html/index.html
<h1>Welcome to App_LB-Neelima</h1>
[ec2-user@ip-10-0-3-247 ~]$
```

- Create a webpage
- Test it in local machine with public ip.

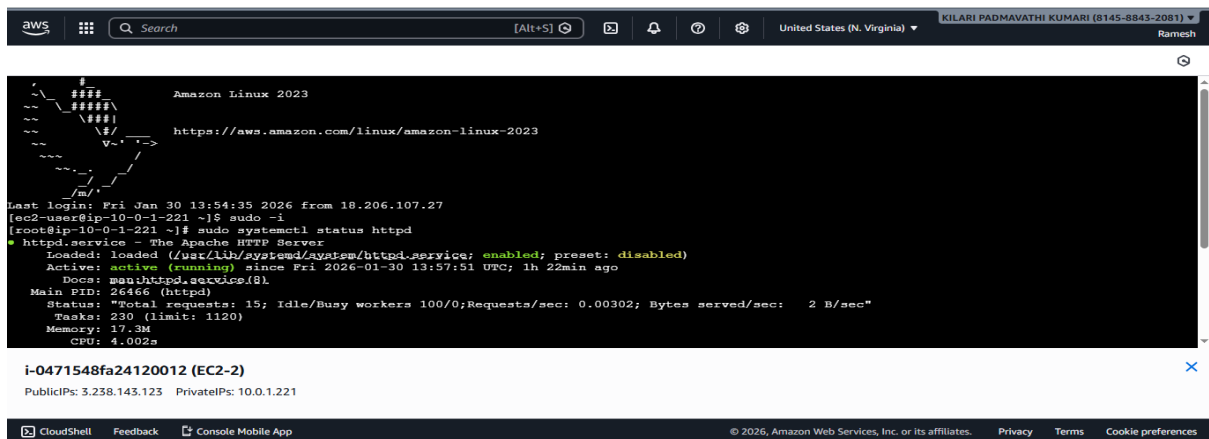


Welcome to App_LB-Neelima

Load Balancers-Task



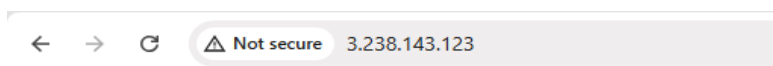
- Create other instance with public ip



- Connect instance with public ip through ssh
- Check whether apache is active and running

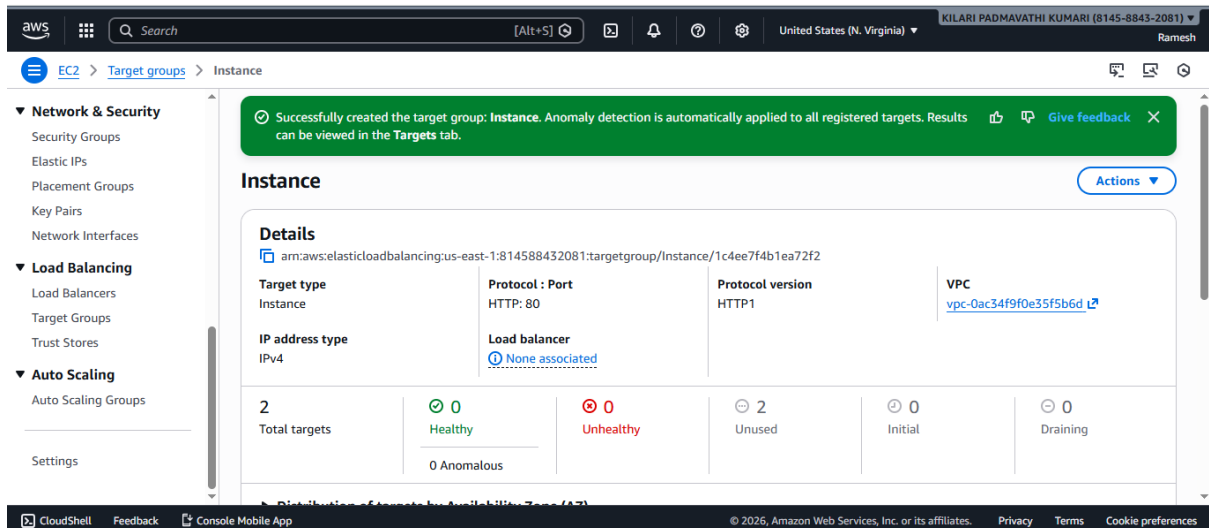


- Create a webpage within the instance
- Test it in browser within local machine.

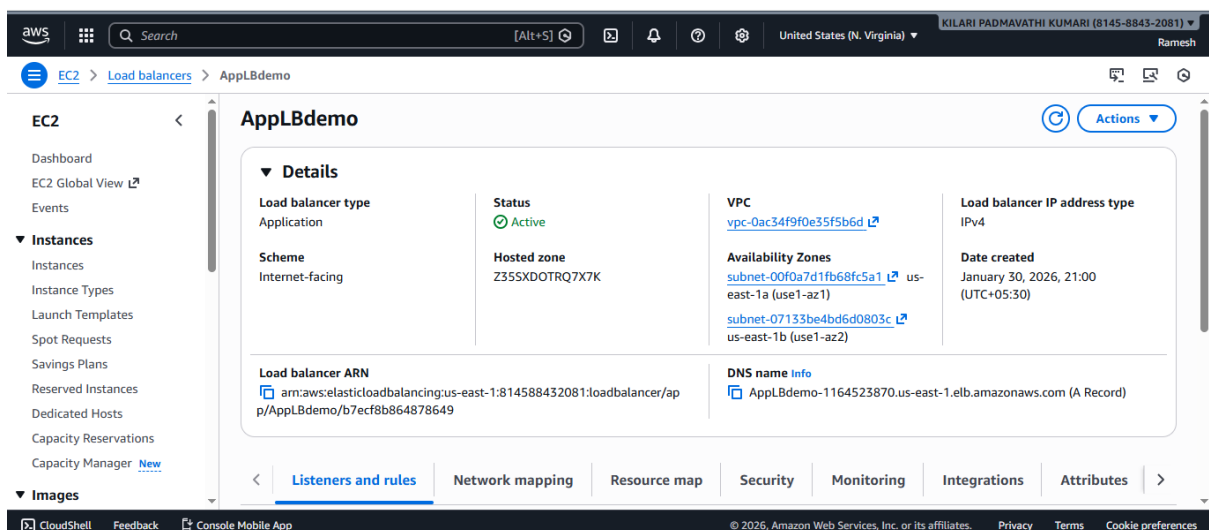


Welcome to App_LB-Neelima

Load Balancers-Task

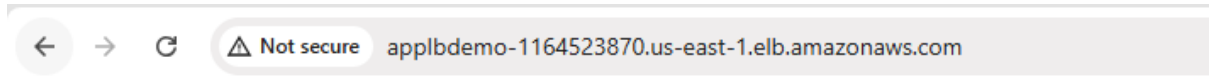


- Now go to EC2 → Target groups
- Create Target group with instances



- EC2 → load balancers → Application LB
- Create Application LB with VPC and subnet
- Internet facing
- Target groups.

Load Balancers-Task

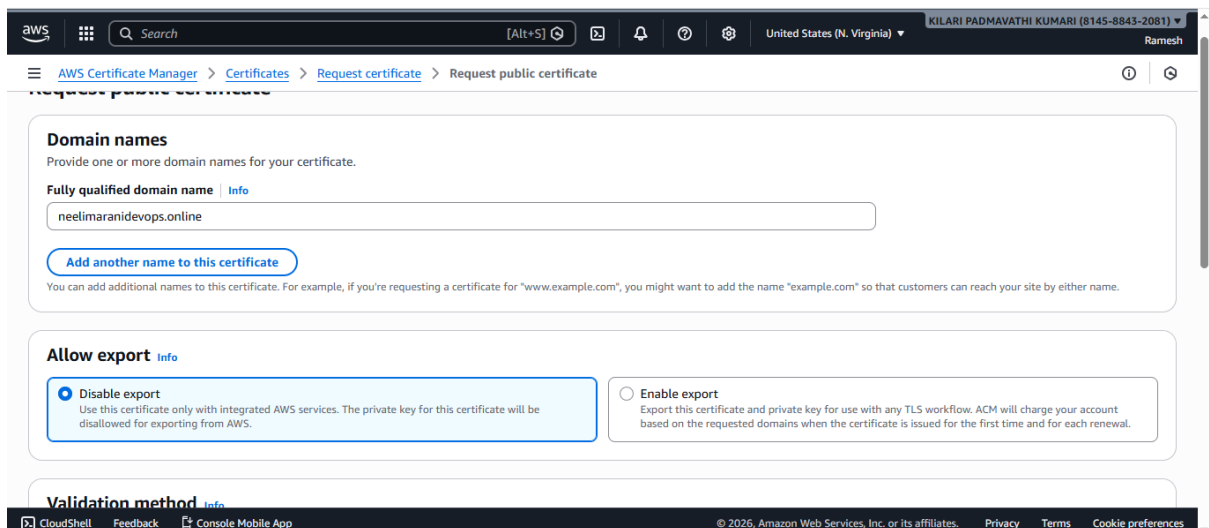


Welcome to App_LB-Neelima

- To verify Copy DNS name and browse it

Step 1: Request SSL Certificate (ACM)

1. Open AWS Console → Certificate Manager (ACM)
2. Click Request certificate
3. Select Public certificate → Next
4. Enter your domain name
Example:neelimaranidevops.online



Load Balancers-Task

Validation method [Info](#)

Select a method for validating domain ownership.

☒ **DNS validation - recommended**
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

☐ **Email validation**
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

Key algorithm [Info](#)

Select an encryption algorithm. Some algorithms may not be supported by all AWS services.

☒ **RSA 2048**
RSA is the most widely used key type.

☐ **ECDSA P 256**
Equivalent in cryptographic strength to RSA 3072.

☐ **ECDSA P 384**
Equivalent in cryptographic strength to RSA 7680.

- Choose **DNS validation**
- Click **Request**

validate Domain

1. Open the certificate → click **Create record in Route53**
2. If domain is not in Route53 → copy CNAME and add in your DNS provider
3. Wait until **Status = Issued**

Certificate status

Identifier: 8717feae-365c-487f-afce-58fd05bf81df **Status** Issued

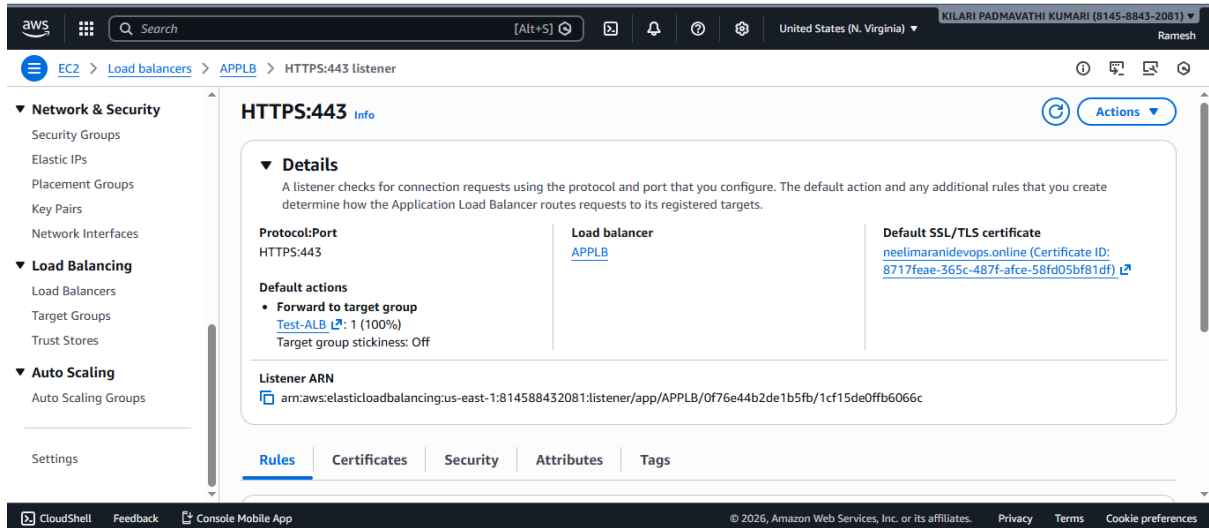
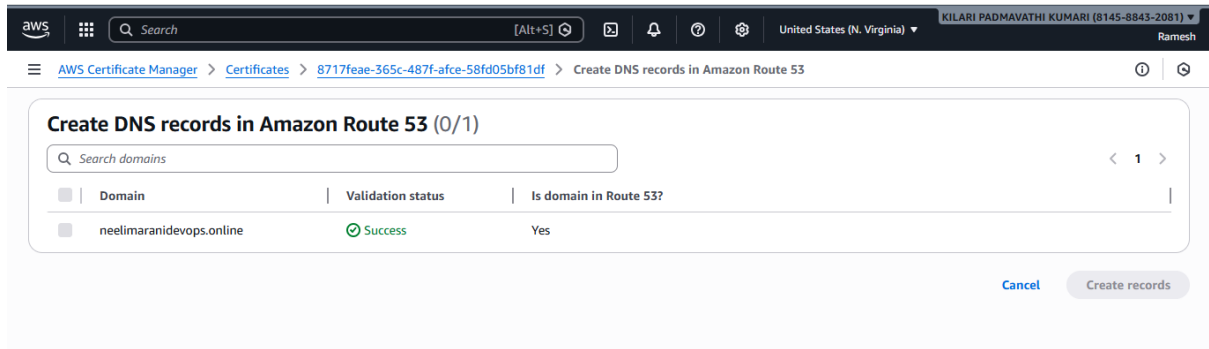
ARN: arn:aws:acm:us-east-1:814588432081:certificate/8717feae-365c-487f-afce-58fd05bf81df

Type: Amazon Issued

Domains (1) [Create records in Route 53](#) [Export to CSV](#)

Domain	Status	Renewal status	Type	CNAME name
neelimaranidevops.online	Success	-	CNAME	_695c1f0328ecb084a.ps.online.

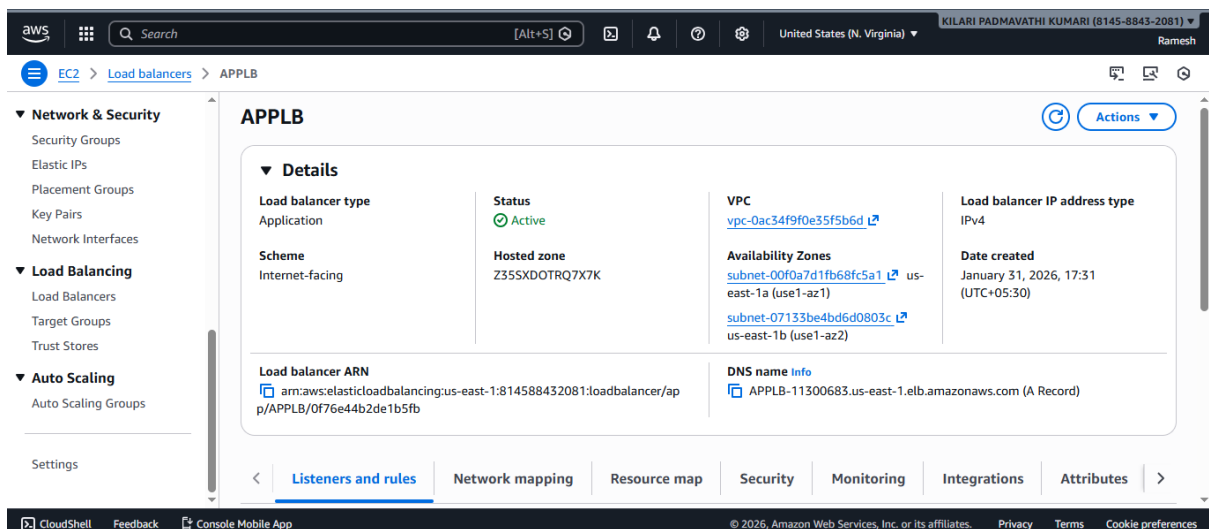
Load Balancers-Task



5. Map Application load balancer to R53

Step 1: Go to your ALB

1. Open EC2 → Load Balancers
2. Select your Application Load Balancer

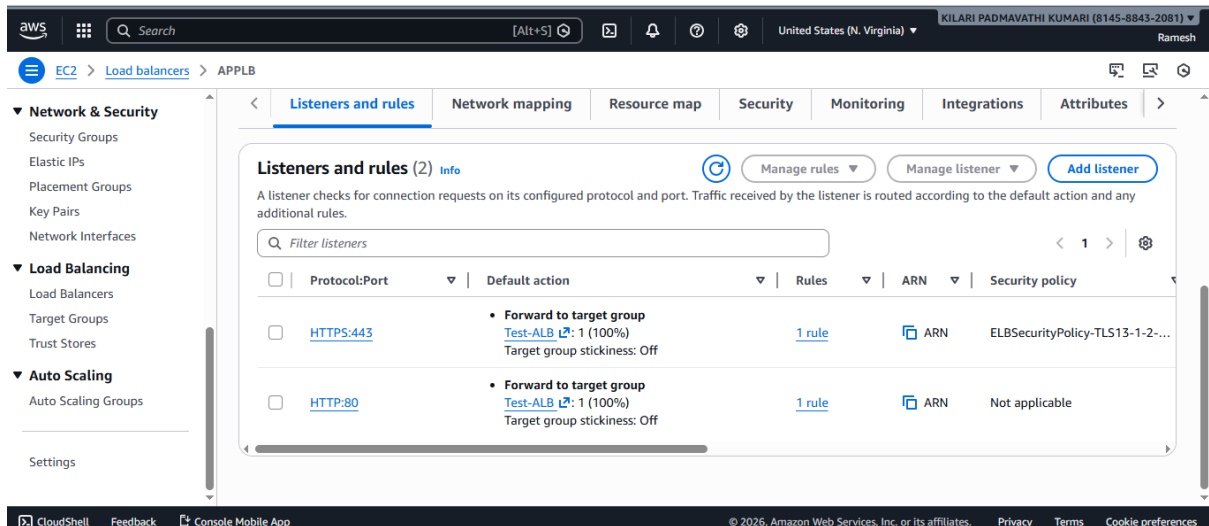


Load Balancers-Task

Step 2: Add HTTPS Listener

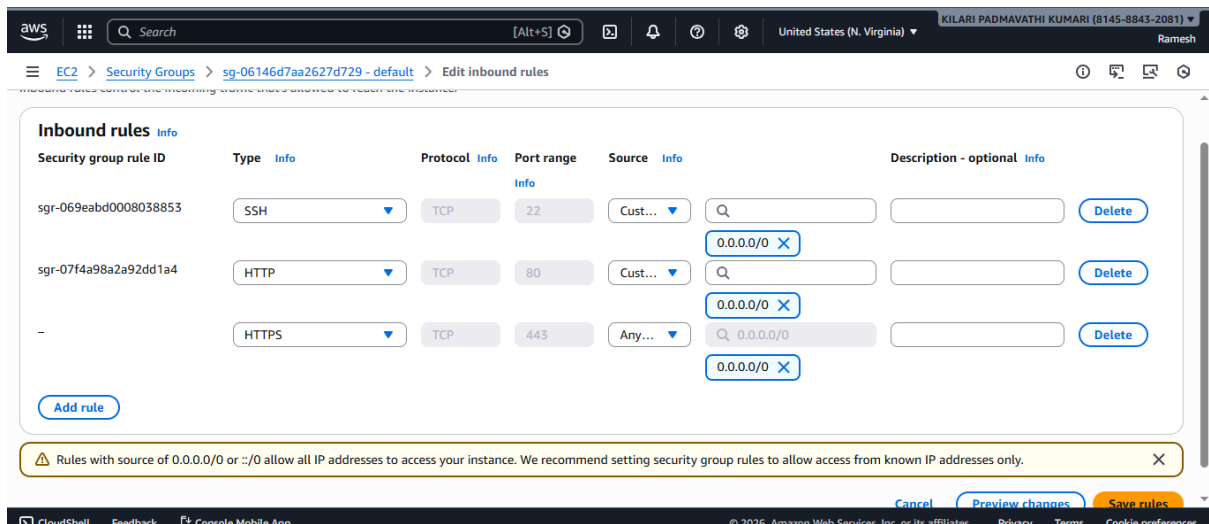
Go to **Listeners** → **Add listener**

- Protocol: **HTTPS**
- Port: **443**
- Default action: **Forward to your target group**

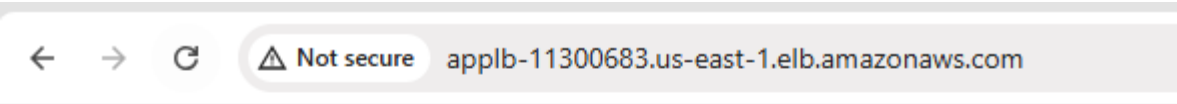


Step 4: Update Security Group

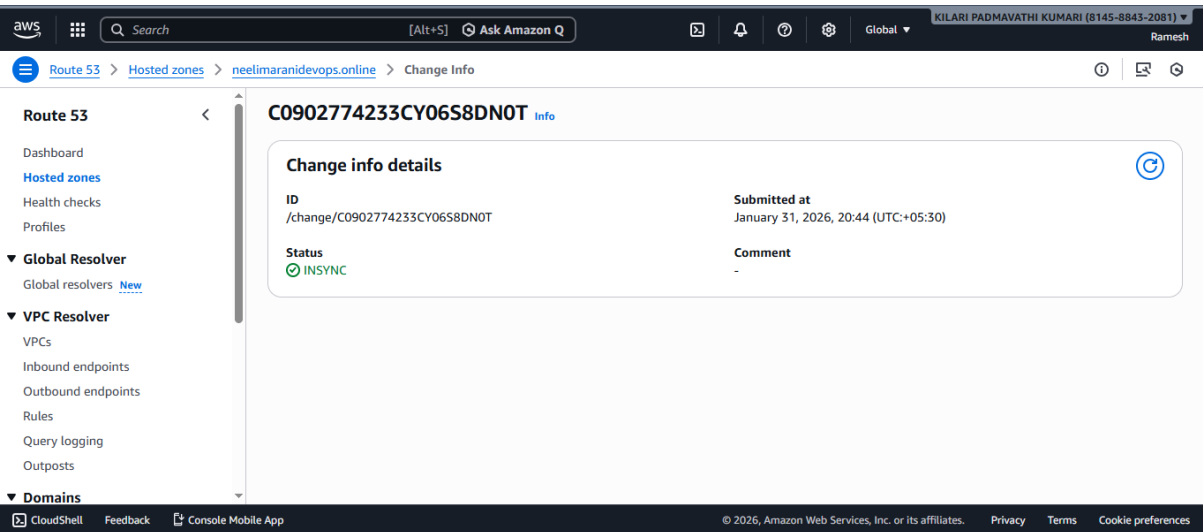
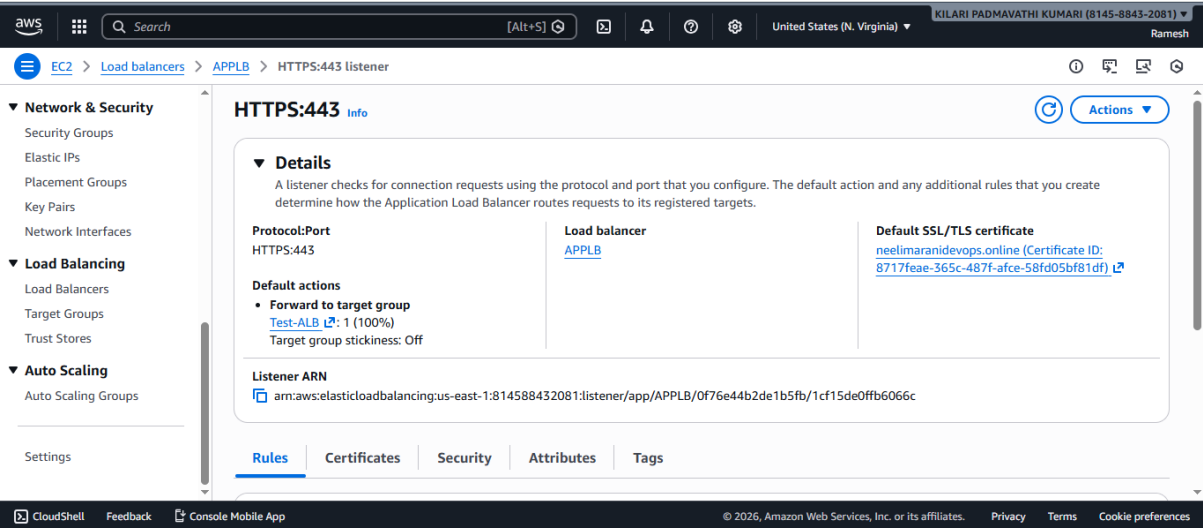
- Allow **Inbound port 443** for **0.0.0.0/0** in ALB security group



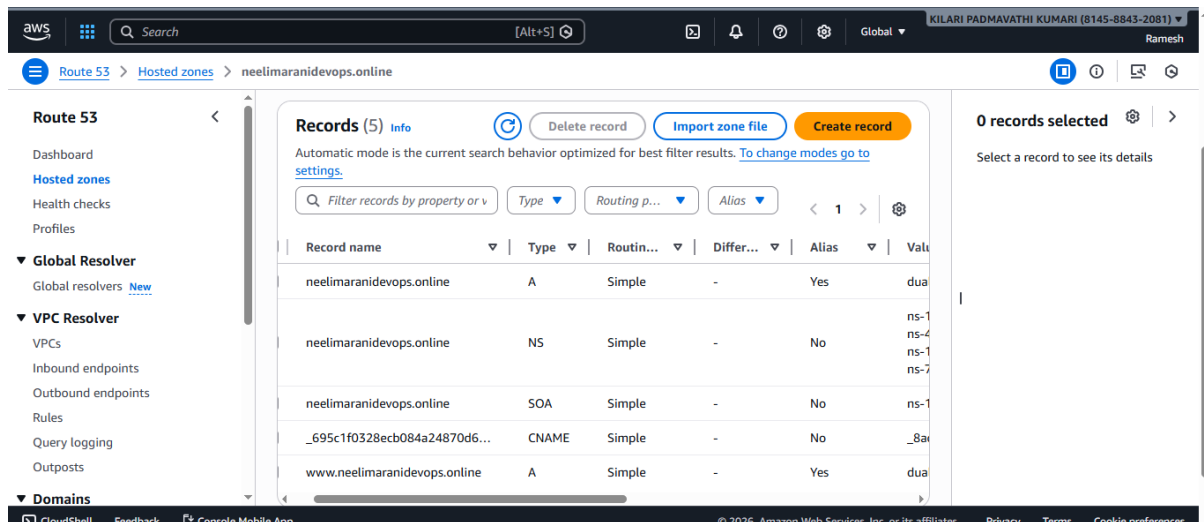
Load Balancers-Task



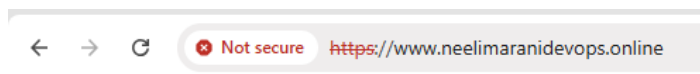
Welcome to App_LB-Neelima



Load Balancers-Task



- Add Records in Route53 as shown above.
- Browser with our domain name :neelimaranidevops.online



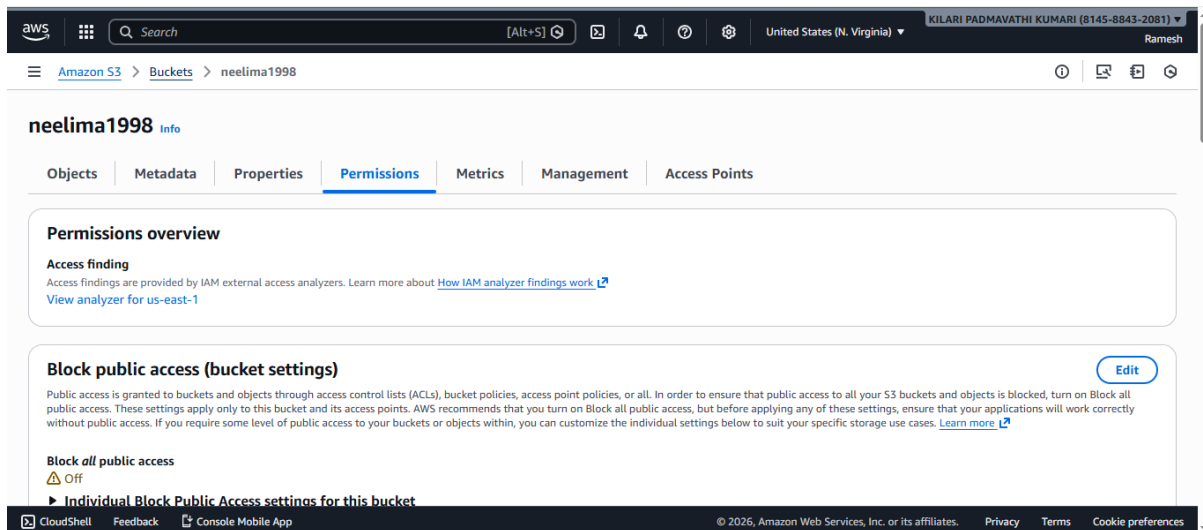
Welcome to App_LB-Neelima

Load Balancers-Task

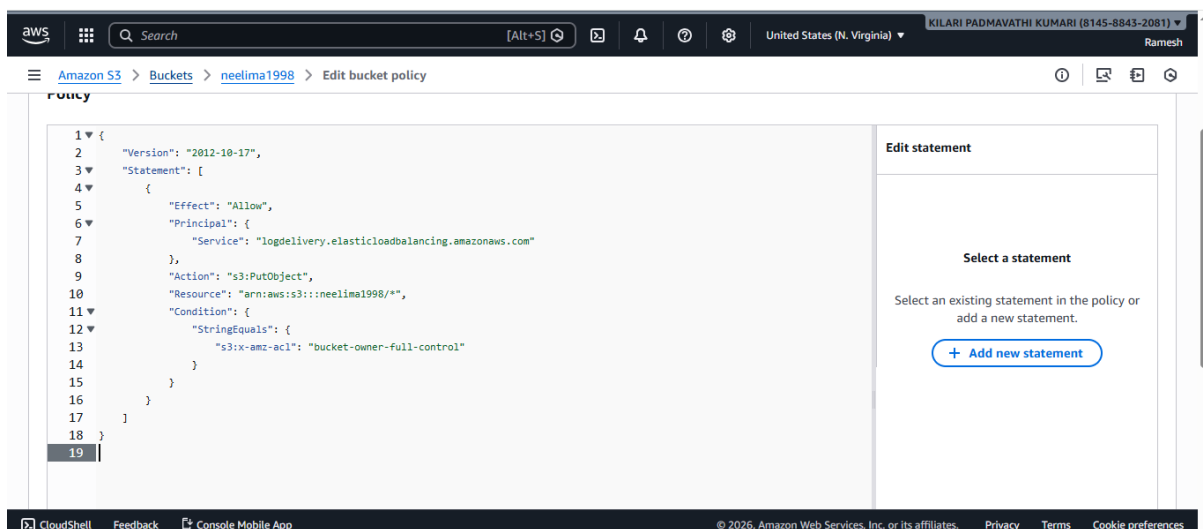
6. Push the application load balancer logs to S3.

step 1: Create an S3 bucket for ALB logs

1. Go to AWS S3 in the console.
2. Click Create bucket.
3. Give it a name like: neelima1998



- Go to permission tab
- Click on Bucket policy
- Edit Bucket policy
- Click on save



Load Balancers-Task

Monitoring

☒ **Access logs**
Access logs deliver detailed logs of all requests made to your Elastic Load Balancer. Choose an existing S3 location. If you don't specify a prefix, the logs are stored in the root of the bucket. Additional charges apply. [Learn more](#)

S3 URI
 [View](#) [Browse S3](#)
Format: s3://bucket/prefix

☐ **Connection logs**
Connection logs deliver detailed logs of all connections made to your Elastic Load Balancer. Choose an existing S3 location. If you don't specify a prefix, the logs are stored in the root of the bucket. Additional charges apply. [Learn more](#)

☐ **Health check logs - new**
Health check logs deliver detailed logs of health checks for all targets in your Elastic Load Balancer's target groups. Choose an existing S3 location. If you don't specify a prefix, the logs are stored in the root of the bucket. Additional charges apply. [Learn more](#)

[Cancel](#) [Save changes](#)

- Go to Ec2 → Load Balancers
- Select Application load Balancers
- Edit Load balancers Attributes
- Monitoring → enable Access logs
- Browser s3 bucket where we want to store logs.
- Click on save changes.

Amazon S3

Buckets

General purpose buckets

Directory buckets

Table buckets

Vector buckets

Access management and security

Access Points

Access Points for FSx

Access Grants

IAM Access Analyzer

Storage management and insights

Storage Lens

Batch Operations

814588432081_elasticloadbalancing_us-east-1_app.APPLB.0f76e44b2de1b5fb_20260131T1330Z_3.214.118.42_3pbl45vg.log.gz

[Copy S3 URI](#) [Download](#) [Open](#) [Object actions](#)

Properties **Permissions** **Versions**

Object overview

Owner
39626811894f9034cd2e0d9b0b263126dbabcf2be650440e357a30a29220670

AWS Region
US East (N. Virginia) us-east-1

Last modified
January 31, 2026, 19:00:03 (UTC+05:30)

S3 URI
[s3://neelima1998/AWSLogs/814588432081/elasticloadbalancing/us-east-1/2026/01/31/814588432081_elasticloadbalancing_us-east-1_app.APPLB.0f76e44b2de1b5fb_20260131T1330Z_3.214.118.42_3pbl45vg.log.gz](#)

Amazon Resource Name (ARN)
[arn:aws:s3::neelima1998/AWSLogs/814588432081/elasticloadbalancing/us-east-1/2026/01/31/814588432081_elasticloadbalancing_us-east-1_app.APPLB.0f76e44b2de1b5fb_20260131T1330Z_3.214.118.42_3pbl45vg.log.gz](#)

- To verify Go to s3 bucket → AWSLogs .