

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 page. It includes sections for Basic configuration, Network mapping, Security groups, and Listeners and routing. The 'Basic configuration' section shows a VPC (vpc-0ac34f9fe35f5b6d) and an Internet-facing scheme. The 'Network mapping' section shows availability zones and subnets (us-east-1a and us-east-1b). The 'Security groups' section lists a default security group (sg-06146d7aa2627d729). The 'Listeners and routing' section shows an HTTP listener on port 80. The 'Health checks' section defines a check for port 80 at index.html with specific timeout and interval settings. The 'Instances' section indicates no instances have been added yet. The 'Attributes' section shows cross-zone load balancing is enabled. The 'Tags' section has no tags defined.

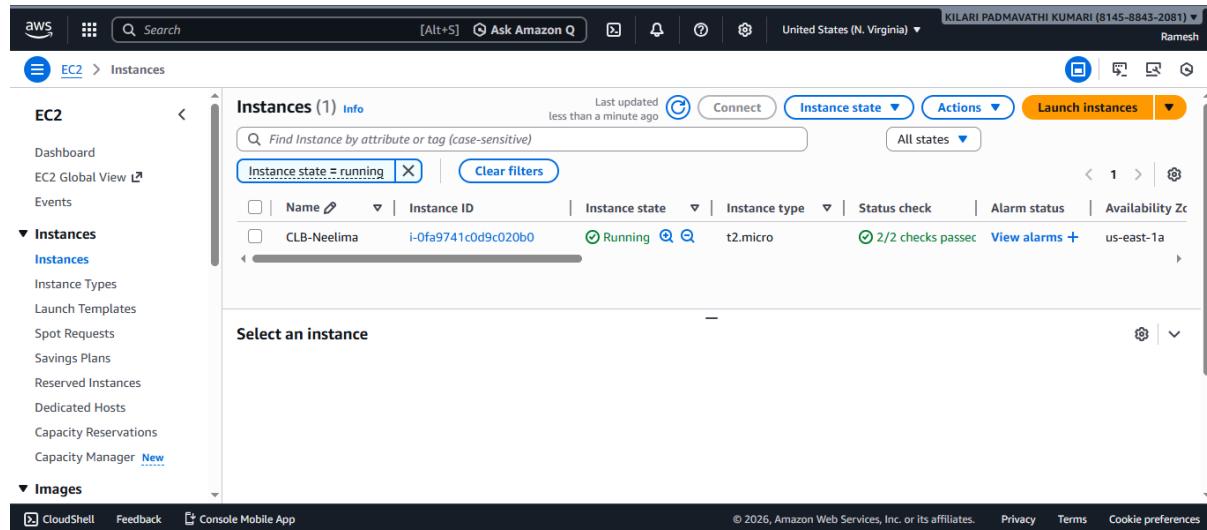
➤ Add EC2 Instances

- Select your running EC2
- Click Add to Registered

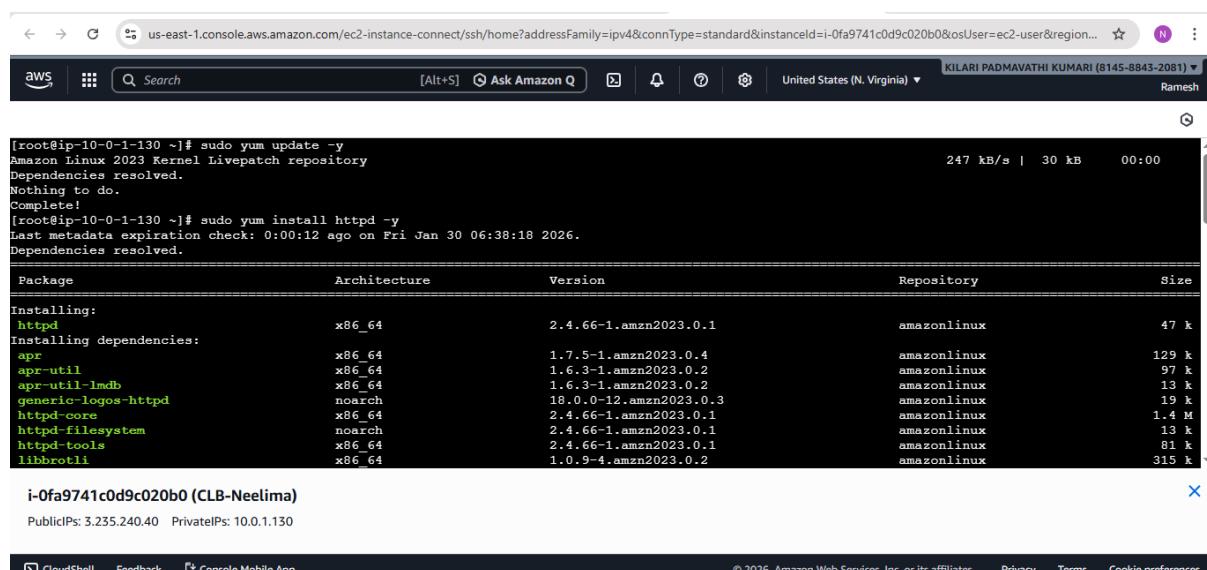
The screenshot shows the 'Manage instances' page for a load balancer named 'Classic-LBNeelima'. A sidebar on the left lists various EC2 management options like Volumes, Snapshots, Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. The main area displays a table of registered instances, showing one instance (i-0fa9741c0d9c020b0) which is currently not registered. A modal window titled 'Review selected instances (1/1)' allows the user to confirm the registration of this instance. The modal includes a summary message stating '1 instance(s) will be registered.' and two buttons: 'Cancel' and 'Save changes'.

Load Balancers-Task

- Go to AWS console
- Create a EC2 instance



The screenshot shows the AWS EC2 Instances page. A single instance, "CLB-Neelima" (i-0fa9741c0d9c020b0), is listed as "Running". The instance type is t2.micro, and it has 2/2 checks passed. The page includes a sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Capacity Manager.



The screenshot shows the AWS CloudShell terminal. The user runs the command `sudo yum update -y`, which updates the Amazon Linux 2023 Kernel Livepatch repository and installs the `httpd` service. The user then runs `sudo yum install httpd -y`, which installs the `httpd` package along with its dependencies. The terminal output shows the packages being installed from the `amazonlinux` repository.

Package	Architecture	Version	Repository	Size
<code>httpd</code>	x86_64	2.4.66-1.amzn2023.0.1	amazonlinux	47 k
<code>apr</code>	x86_64	1.7.5-1.amzn2023.0.4	amazonlinux	129 k
<code>apr-util</code>	x86_64	1.6.3-1.amzn2023.0.2	amazonlinux	97 k
<code>apr-util-lmdb</code>	x86_64	1.6.3-1.amzn2023.0.2	amazonlinux	13 k
<code>generic-logos-httpd</code>	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
<code>httpd-core</code>	x86_64	2.4.66-1.amzn2023.0.1	amazonlinux	1.4 M
<code>httpd-filesystem</code>	noarch	2.4.66-1.amzn2023.0.1	amazonlinux	13 k
<code>httpd-tools</code>	x86_64	2.4.66-1.amzn2023.0.1	amazonlinux	81 k
<code>libbrotli</code>	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k

- 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

The screenshot shows a terminal session in AWS CloudShell. The user runs three commands: `sudo systemctl start httpd`, `sudo systemctl enable httpd`, and `sudo systemctl status httpd`. The output indicates that the service is active and running, with a main PID of 26381. The terminal also shows log entries from January 30 at 06:39:42 indicating the start of the service.

```
[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:htpd.service(8)
   Main PID: 26381 (httpd)
      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)
Public IPs: 3.235.240.40 Private IPs: 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
- Sudo 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

A screenshot of a web browser window. The address bar shows the URL 3.235.240.40. The page content is "CLB Working - Neelima".

← → ⌛ Not secure 3.235.240.40

CLB Working - Neelima

Load Balancers-Task

EC2 > Load balancers > Classic-LBNeelima

Classic-LBNeelima

Details

Load balancer type	Status	VPC	Date created
Classic	1 of 1 instance in service	vpc-0ac34f9f0e35f5b6d	January 29, 2026, 23:43 (UTC+05:30)
Scheme	Hosted zone	Availability Zones	
Internet-facing	Z35SXDOTRQ7X7K	subnet-00fa0ad1fb68fc5a1 us-east-1a (use1-az1) subnet-07133be4bd6d0803c us-east-1b (use1-az2)	
DNS name	classic-lbneelima-961176476.us-east-1.elb.amazonaws.com (A Record)	ⓘ This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. Learn more	

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

← → ⌂ Not secure classic-lbneelima-961176476.us-east-1.elb.amazonaws.com

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

Instance summary for i-0f754858b709b9a61 (EC2-1) [Info](#)

Updated less than a minute ago

Attribute	Value
Instance ID	i-0f754858b709b9a61
IPv6 address	-
Hostname type	IP name: ip-10-0-1-202.ec2.internal
Answer private resource DNS name	-
Auto-assigned IP address	18.234.76.108 [Public IP]
Public IPv4 address	18.234.76.108 open address
Instance state	Running
Private IP DNS name (IPv4 only)	ip-10-0-1-202.ec2.internal
Instance type	t2.micro
VPC ID	vpc-0ac34f9f0e35f5b6d (VPC-Neelima)
Subnet ID	-
Private IPv4 addresses	10.0.1.202
Public DNS	ec2-18-234-76-108.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 to instance through ssh connection
- Install apache in EC2-1 instance commands are as follows
 - Sudo yum update -y
 - sudo yum install httpd -y

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

[ec2-user@ip-10-0-1-202 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Fri Jan 30 10:34:30 2026.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-202 ~]$ sudo yum install httpd -y
Last metadata expiration check: 0:00:13 ago on Fri Jan 30 10:34:30 2026.
Dependencies resolved.

Package          Architecture      Version       Repository      Size
i-0f754858b709b9a61 (EC2-1)
PublicIPs: 18.234.76.108  PrivateIPs: 10.0.1.202
```

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:htpd.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-0f754858b709b9a61 (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-0f754858b709b9a61 (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

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Images, CloudShell, Feedback, and Console Mobile App. The main area displays the instance summary for 'i-0ab987486937cdcb8 (Ec2-2)'. Key details include:

- Instance ID: i-0ab987486937cdcb8
- IPv6 address: -
- Hostname type: IP name: ip-10-0-3-129.ec2.internal
- Answer private resource DNS name: -
- Auto-assigned IP address: 100.53.141.240 [Public IP]
- Public IPv4 address: 100.53.141.240 [open address]
- Instance state: Running
- Private IP DNS name (IPv4 only): ip-10-0-3-129.ec2.internal
- Instance type: t2.micro
- VPC ID: vpc-0ac34f9f0e35f5b6d (VPC-Neelima)
- Subnet ID: -
- Private IPv4 addresses: 10.0.3.129
- Public DNS: ec2-100-53-141-240.compute-1.amazonaws.com [open address]
- Elastic IP addresses: -
- AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.
- Auto Scaling Group name: -

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information: © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

The screenshot shows the AWS CloudShell terminal. It displays the following command output:

```
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.

Transaction Summary
  Package          Architecture Version      Repository      Size
Installing:
  i-0ab987486937cdcb8 (Ec2-2)

  PublicIPs: 100.53.141.240 PrivateIPs: 10.0.3.129
```

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information: © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

- 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 ~]$
```

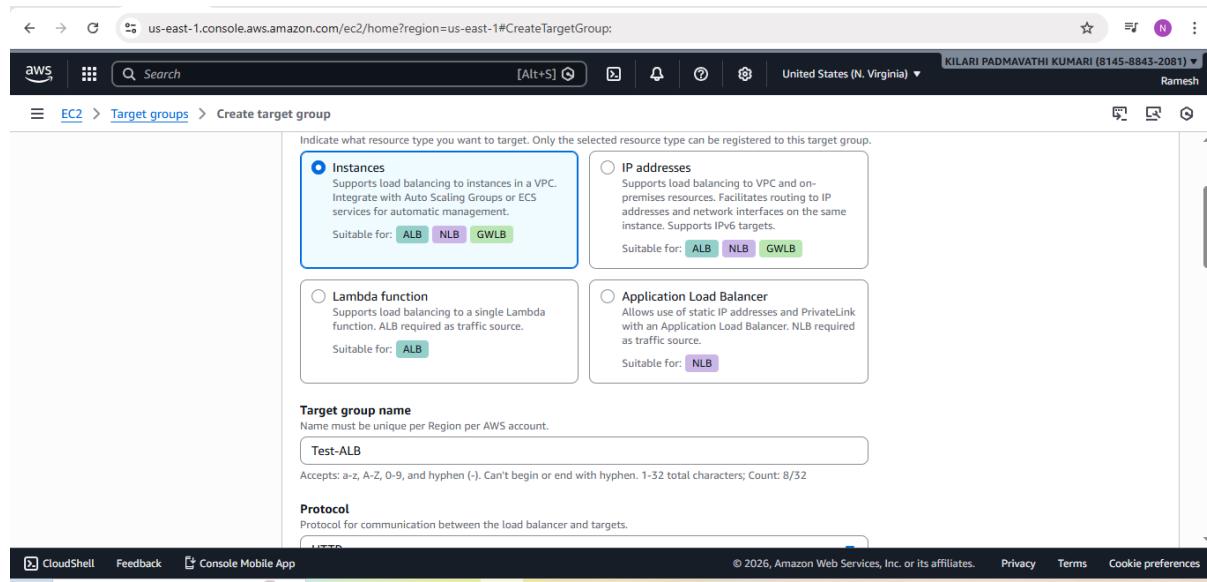
- Create a webpage in ece instance using html
- echo "<h1>Message</h1>" | sudo tee /usr/share/nginx/html/index.html

← → ⌂ Not secure 100.53.141.240

welcome to Nginx_ app-LB-Neelima

- Test webpage using public ip in local machine.

Load Balancers-Task



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

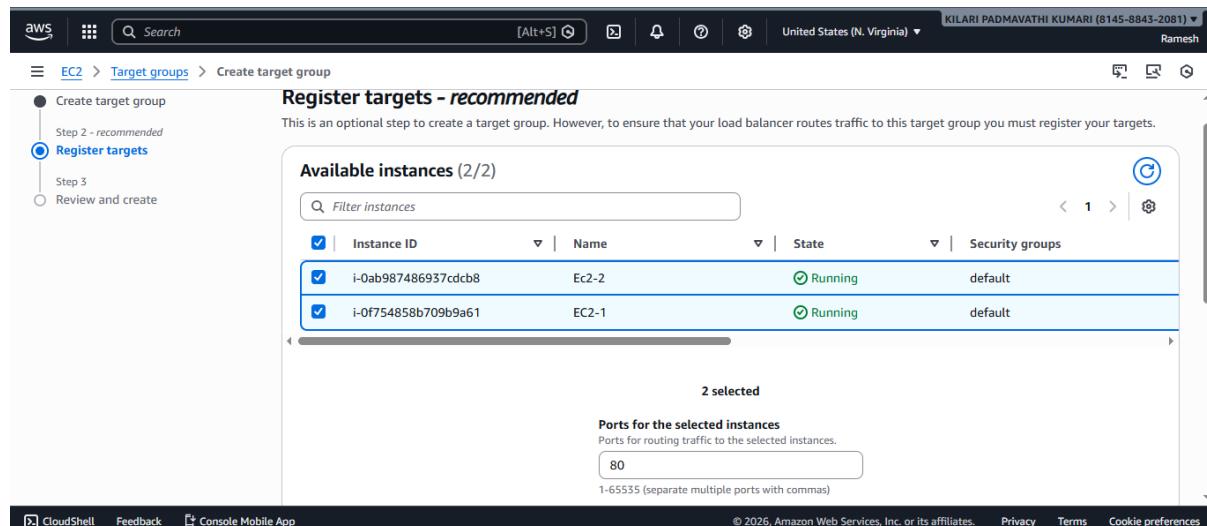
Target group name
Name must be unique per Region per AWS account.

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.

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



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.

Register targets

Create target group
Step 2 - recommended

Step 3
Review and create

Available instances (2/2)

Instance ID	Name	State	Security groups
i-0ab987486937cdcb8	Ec2-2	Running	default
i-0f754858b709b9a61	EC2-1	Running	default

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

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

The screenshot shows the 'Create target group' wizard. Step 1: 'Health check details' is completed with the following configuration:

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

Step 2: 'Register targets' shows two targets registered:

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

Buttons at the bottom include 'Cancel', 'Previous', and a prominent orange 'Create target group' button.

- Click on Create target group

The screenshot shows the 'Test-ALB' target group details. A success message is displayed: "Successfully created the target group: Test-ALB. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." The target group details are as follows:

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

Summary statistics:

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

Actions dropdown is available in the top right corner.

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

Load Balancers-Task

The screenshot shows the AWS CloudFront console with the following details:

- Region:** United States (N. Virginia)
- User:** KILARI PADMAVATHI KUMARI (8145-8843-2081) - Ramesh
- Breadcrumbs:** EC2 > Load balancers > Create Application Load Balancer
- Title:** Create Application Load Balancer Info
- Description:** 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.
- Section:** How Application Load Balancers work
- Basic configuration:**
 - Load balancer name:** APP-LBdemo (highlighted in blue)
 - Scheme:** Internet-facing (selected)
 - Internal:** (radio button not selected)
- Buttons:** CloudShell, Feedback, Console Mobile App
- Footer:** © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- 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

The screenshot shows the AWS CloudFront console with the following details:

- Region:** United States (N. Virginia)
- User:** KILARI PADMAVATHI KUMARI (8145-8843-2081) - Ramesh
- Breadcrumbs:** EC2 > Load balancers > Create Application Load Balancer
- Title:** Summary
- Review and confirm your configurations:**
 - Basic configuration:** Edit
Name: APP-LBdemo
Scheme: Internet-facing
IP address type: IPv4
 - Network mapping:** Edit
VPC: vpc-0ac34f9f0e35f5b6d
Public IPv4 IPAM pool: -
Availability Zones and subnets:
 - us-east-1a
subnet-00fa7d1fb68fc5a1
public-Neelima
 - us-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: -
- Attributes:** Edit
Tags: Edit
-

Buttons: CloudShell, Feedback, Console Mobile App

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

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

Load Balancers-Task

The screenshot shows the AWS EC2 Load Balancers console. On the left, there's a navigation sidebar with options like Volumes, Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main area displays the details of a load balancer named 'APP-LBdemo'. The 'Details' section includes fields for Load balancer type (Application), Status (Provisioning), VPC (vpc-0ac34f9f0e35f5b6d), Hosted zone (Z35SXDOTRQ7X7K), Availability Zones (subnet-00f0a7d1fb68fc5a1, subnet-07133be4bd6d0803c), Load balancer ARN (arn:aws:elasticloadbalancing:us-east-1:814588432081:loadbalancer/app/APP-LBdemo/9984aa6f10b61090), and DNS name (app-lbdemo-586420120.us-east-1.elb.amazonaws.com). The DNS name is also shown as an A Record.

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

Two browser windows are shown. The top window has a title bar 'app-lbdemo-586420120.us-eas' and displays the text 'welcome to Nginx_ app-LB-Neelima'. The bottom window has a title bar 'app-lbdemo-586420120.us-eas' and displays the text 'Welcome to Apache_Neelima'. Both windows show a warning 'Not secure' and the URL 'app-lbdemo-586420120.us-east-1.elb.amazonaws.com'.

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

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Capacity Manager), and Images. The main content area displays the 'Instance summary for i-0471548fa24120012 (Network-1)' for an 'i-0471548fa24120012' instance. The summary includes fields for Instance ID, IPv6 address, Hostname type (IP name: ip-10-0-1-221.ec2.internal), Answer private resource DNS name, Auto-assigned IP address (3.238.143.123 [Public IP]), Public IPv4 address (3.238.143.123), Instance state (Running), Private IP DNS name (IPv4 only) (ip-10-0-1-221.ec2.internal), Instance type (t2.micro), VPC ID (vpc-0ac34f9f0e35f5b6d (VPC-Neelima)), Subnet ID, and other details like Private IPv4 addresses (10.0.1.221), Public DNS (ec2-3-238-143-123.compute-1.amazonaws.com), and Elastic IP addresses. There are also sections for AWS Compute Optimizer finding and Auto Scaling Group name. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and links for Privacy, Terms, and Cookie preferences.

Load Balancers-Task

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

```
'`#` Amazon Linux 2023
`~\`####` https://aws.amazon.com/linux/amazon-linux-2023
`~` \#/
`~` V~` ->
`~` /
`~` /`/
`~` /m`/
[ec2-user@ip-10-0-1-221 ~]$ sudo -i
[root@ip-10-0-1-221 ~]# sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-1-221 ~]# sudo yum install httpd -y
Last metadata expiration check: 0:00:21 ago on Fri Jan 30 13:55:10 2026.
Dependencies resolved.

Package Architecture Version Repository Size
i-0471548fa24120012 (Network-1)
```

PublicIPs: 3.238.143.123 PrivateIPs: 10.0.1.221

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

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

PublicIPs: 3.238.143.123 PrivateIPs: 10.0.1.221

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

```
[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



Welcome to NLB-Neelima

Load Balancers-Task

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

EC2 Instances > i-0fdbb8d38ff422021

Instance summary for i-0fdbb8d38ff422021 (Network-2)

Updated less than a minute ago

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0fdbb8d38ff422021	54.91.182.172 open address	10.0.3.247
IPv6 address	Instance state	Public DNS
-	Running	ec2-54-91-182-172.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-10-0-3-247.ec2.internal	ip-10-0-3-247.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
-	t2.micro	Opt-in to AWS Compute Optimizer for recommendations.
Auto-assigned IP address	VPC ID	Learn more
54.91.182.172 [Public IP]	vpc-0ac34f9f0e35f5b6d (VPC-Neelima)	
IAM Role	Subnet ID	Auto Scaling Group name
-	-	-

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

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

```
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-0fdbb8d38ff422021 (Network-2)
PublicIPs: 54.91.182.172 PrivateIPs: 10.0.3.247
```

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

```
[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: manhttpd.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-0fdbb8d38ff422021 (Network-2)
PublicIPs: 54.91.182.172 PrivateIPs: 10.0.3.247

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

```
NL-B-Neelima$ echo "<h1>Welcome to NLB-Neelima</h1>" |sudo tee /var/www/html/index.html
NL-B-Neelima$
```

i-0fdbb8d38ff422021 (Network-2)
PublicIPs: 54.91.182.172 PrivateIPs: 10.0.3.247

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

Load Balancers-Task



Welcome to NLB-Neelima

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)

Instance ID	Name	State	Security groups
i-0fdbb8d38ff422021	Network-2	Running	default
i-0471548fa24120012	Network-1	Running	default

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

Successfully created the target group: NLBdemo.

NLBdemo

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

The screenshot shows the 'Basic configuration' step of the CloudFront distribution creation wizard. It includes fields for Name (NLB-Neelima), Scheme (Internet-facing selected), Load balancer IP address type (IPv4 selected), and a summary of chosen settings.

The screenshot shows the 'Review' step of the CloudFront distribution creation wizard. It displays the configuration details: Name (NLB-Neelima), Scheme (Internet-facing), IP address type (IPv4), Network mapping (VPC: vpc-0ac34f9f0e35f5b6d, Availability Zones: us-east-1a, us-east-1b), Security groups (default sg-06146d7aa2627d729), Listeners and routing (TCP:80 | Forward to 1 target group), and Tags (None).

The screenshot shows the details page for the NLB-Neelima distribution. It lists the following information:

- 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.
- Details:** Load balancer type (Network), Status (Active), VPC (vpc-0ac34f9f0e35f5b6d), Load balancer IP address type (IPv4), Scheme (Internet-facing), Hosted zone (Z26RNL4JYFTOTI), Availability Zones (us-east-1a, us-east-1b), DNS name (NLB-Neelima-f9b5929961a62973.elb.us-east-1.amazonaws.com).
- Listeners:** The 'Listeners' tab is selected, showing the configuration for port 80.
- Network mapping:** Shows the mapping to the VPC and subnets.
- Resource map:** Shows the resources associated with the distribution.
- Monitoring:** Shows monitoring metrics.
- Integrations:** Shows integrations with other services.
- Attributes:** Shows distribution attributes.
- Capacity:** Shows distribution capacity.

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

The screenshot shows the AWS EC2 Instances page. The left sidebar has 'EC2' selected under 'Instances'. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Z
A	i-02861daeb77063a4f	Stopped	t3.micro	-	View alarms +	us-east-1d
EC2-1	i-0fddb8d38ff422021	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b
EC2-2	i-0471548fa24120012	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a

Below the table, there is a section titled 'Select an instance'.

Load Balancers-Task

```
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:htpd.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

[ec2-user@ip-10-0-3-247 ~]$ i-0fddb8d38ff422021 (EC2-1)
PublicIPs: 54.91.182.172 PrivateIPs: 10.0.3.247

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

- 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.

← → G ▲ Not secure 54.91.182.172

Welcome to App_LB-Neelima

Load Balancers-Task

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for EC2, Dashboard, EC2 Global View, Instances (selected), Images, IAM Role, CloudShell, Feedback, and Console Mobile App. The main content area displays the 'Instance summary for i-0471548fa24120012 (EC2-2)'. It includes fields for Instance ID (i-0471548fa24120012), IPv6 address (empty), Hostname type (IP name: ip-10-0-1-221.ec2.internal), Answer private resource DNS name (empty), Auto-assigned IP address (empty), Public IPv4 address (3.238.143.123), Instance state (Running), Private IP DNS name (ip-10-0-1-221.ec2.internal), Instance type (t2.micro), VPC ID (vpc-0ac34f9f0e35f5b6d (VPC-Neelima)), Subnet ID (empty), Private IP4 addresses (10.0.1.221), Public DNS (ec2-3-238-143-123.compute-1.amazonaws.com), Elastic IP addresses (empty), AWS Compute Optimizer finding (Opt-in to AWS Compute Optimizer for recommendations), and Auto Scaling Group name (empty). The bottom right corner shows '© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

- Create other instance with public ip

The screenshot shows the AWS CloudShell terminal. The command history shows:
Last login: Fri Jan 30 13:54:35 2026 from 18.206.107.27
[ec2-user@ip-10-0-1-221 ~]\$ sudo -i
[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; 1h 22min ago
 Main PID: 26466 (httpd)
 Status: "Total requests: 15; Idle/Busy workers 100/0;Requests/sec: 0.00302; Bytes served/sec: 2 B/sec"
 Tasks: 230 (limit: 1120)
 Memory: 17.3M
 CPU: 4.002s
The terminal prompt is [root@ip-10-0-1-221 ~]#.

i-0471548fa24120012 (EC2-2)
PublicIPs: 3.238.143.123 PrivateIPs: 10.0.1.221

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

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

The screenshot shows the AWS CloudShell terminal. The command history shows:
[root@ip-10-0-1-221 ~]# echo "<h1>Welcome to App_LB-Neelima</h1>" | sudo tee /var/www/html/index.html
<h1>Welcome to App_LB-Neelima</h1>
[root@ip-10-0-1-221 ~]#

i-0471548fa24120012 (EC2-2)
PublicIPs: 3.238.143.123 PrivateIPs: 10.0.1.221

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services

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



Welcome to App_LB-Neelima

Load Balancers-Task

The screenshot shows the AWS EC2 Target groups Instance page. A green success message at the top states: "Successfully created the target group: Instance. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." The main section is titled "Instance" and contains a "Details" table. The table includes fields for Target type (Instance), Protocol (HTTP: 80), Protocol version (HTTP1), IP address type (IPv4), and VPC (vpc-0ac34f9f0e35f5b6d). Below the table, a summary table shows the following metrics: Total targets (2), Healthy (0), Unhealthy (0), Unused (2), Initial (0), and Draining (0). The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.

- Now go to Ec2→Target groups
- Create Target group with instances

The screenshot shows the AWS EC2 Load balancers Application LB demo page. A green success message at the top states: "Successfully created the load balancer: AppLBdemo. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." The main section is titled "AppLBdemo" and contains a "Details" table. The table includes fields for Load balancer type (Application), Status (Active), VPC (vpc-0ac34f9f0e35f5b6d), and Load balancer IP address type (IPv4). Other details shown include Hosted zone (Z35SXDOTRQ7X7K), Availability Zones (subnet-00f0a7d1f1b68fc5a1 and subnet-07133be4bd6d0803c), Date created (January 30, 2026, 21:00 UTC+05:30), and DNS name (AppLBdemo-1164523870.us-east-1.elb.amazonaws.com). Below the table, tabs for Listeners and rules, Network mapping, Resource map, Security, Monitoring, Integrations, and Attributes are visible. The bottom of the page includes standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.

- 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

The screenshot shows the 'Request public certificate' step in the AWS Certificate Manager wizard. It includes fields for entering domain names, options for allowing export, and a validation method section.

Domain names
Provide one or more domain names for your certificate.
Fully qualified domain name | Info
neelimaranidevops.online

Add another name to this certificate
You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

Allow export | Info
 Disable export
Use this certificate only with integrated AWS services. The private key for this certificate will be disallowed for exporting from AWS.

Enable export
Export this certificate and private key for use with any TLS workflow. ACM will charge your account based on the requested domains when the certificate is issued for the first time and for each renewal.

Validation method | Info

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

Load Balancers-Task

The screenshot shows the 'Request public certificate' page in AWS Certificate Manager. Under 'Validation method', 'DNS validation - recommended' is selected. Under 'Key algorithm', 'RSA 2048' is selected. At the bottom, there are 'Tags' and navigation links for CloudShell, Feedback, and Console Mobile App.

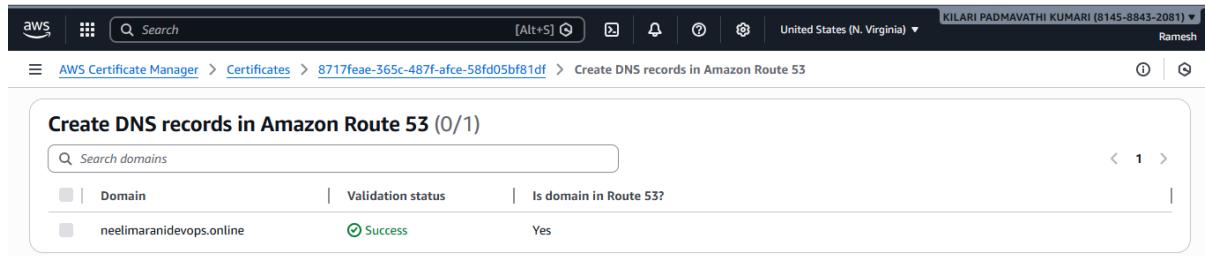
- Choose **DNS validation**
- Click **Request**

validate Domain

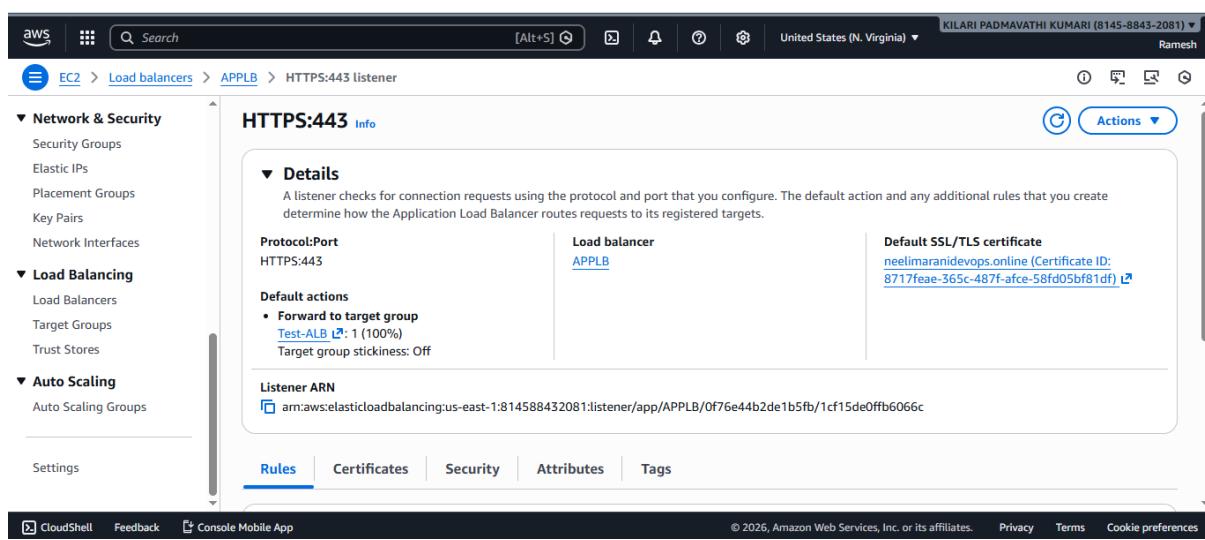
1. Open the certificate → click **Create record in Route53**
2. If domain is not in Route53 → copy CNAME and add in your DNS provider
3. Wait until **Status = Issued**

The screenshot shows the 'Certificate status' page for a certificate with ARN: arn:aws:acm:us-east-1:814588432081:certificate/8717feae-365c-487f-afce-58fd05bf81df. The status is 'Issued'. Below it, the 'Domains' section lists one domain: neelimaranidevops.online, which has a status of 'Success' and a CNAME type pointing to _695c1f0328ecb084aps.online.

Load Balancers-Task



The screenshot shows the 'Create DNS records in Amazon Route 53' page. It lists one domain: 'neelimaranidevops.online' with a validation status of 'Success' and 'Is domain in Route 53?' set to 'Yes'. There are 'Cancel' and 'Create records' buttons at the bottom.

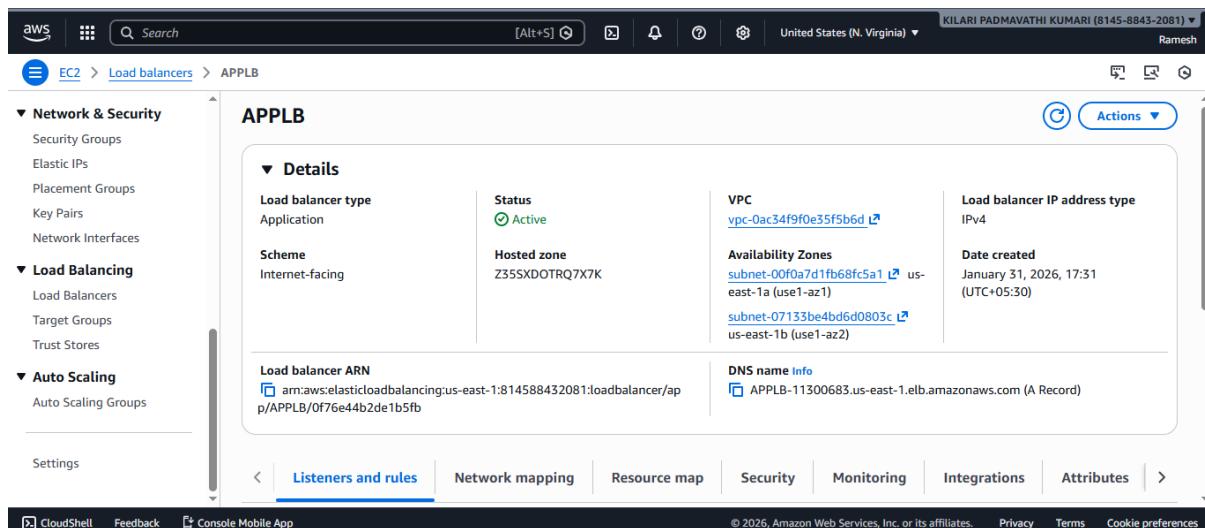


The screenshot shows the 'HTTPS:443' listener configuration for the Application Load Balancer 'APPLB'. It details the protocol as 'HTTPS:443', the load balancer as 'APPLB', and the default action as 'Forward to target group Test-ALB (100%)'. The ARN is listed as 'arn:aws:elasticloadbalancing:us-east-1:814588432081:listener/app/APPLB/0f76e44b2de1b5fb/1cf15de0ffb6066c'. The 'Rules' tab is selected.

5. Map Application load balancer to R53

Step 1: Go to your ALB

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



The screenshot shows the 'Details' section of the Application Load Balancer 'APPLB'. It includes fields for 'Load balancer type' (Application), 'Status' (Active), 'VPC' (vpc-0ac34f9f0e35f5b6d), 'Scheme' (Internet-facing), 'Hosted zone' (Z35SXDOTRQ7X7K), 'Availability Zones' (subnet-00fa07d1fb68fc5a1, subnet-07133be4bd6d0803c), 'Load balancer IP address type' (IPv4), 'Date created' (January 31, 2026, 17:31 (UTC+05:30)), and 'DNS name' (APPLB-11300683.us-east-1.elb.amazonaws.com (A Record)). The 'Listeners and rules' tab is selected.

Load Balancers-Task

Step 2: Add HTTPS Listener

Go to **Listeners → Add listener**

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

The screenshot shows the AWS Load Balancers console with the 'Listeners and rules' tab selected. There are two listeners listed:

- HTTPS:443**: Forwarded to target group **Test-ALB** (1 rule, ARN: ELBSecurityPolicy-TLS13-1-2-...)
- HTTP:80**: Forwarded to target group **Test-ALB** (1 rule, ARN: Not applicable)

Step 4: Update Security Group

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

The screenshot shows the AWS Security Groups console with the 'Edit inbound rules' page for a specific security group. It lists three rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-069eabd0008038853	SSH	TCP	22	Cust... (0.0.0.0/0)	
sgr-07f4a98a2a92dd1a4	HTTP	TCP	80	Cust... (0.0.0.0/0)	
-	HTTPS	TCP	443	Any... (0.0.0.0/0)	

A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."

Load Balancers-Task



Welcome to App_LB-Neelima

The screenshot shows the AWS EC2 Load Balancers console. The left sidebar navigation includes Network & Security, Load Balancing, Auto Scaling, and Settings. The main content area displays the details of the HTTPS:443 listener for the ALB APPLB. Key details shown include:

- Protocol:Port**: HTTPS:443
- Load balancer**: APPLB
- Default SSL/TLS certificate**: neelimaraindevops.online (Certificate ID: 8717feae-365c-487f-acfe-58fd05bf81df)
- Default actions**: Forward to target group Test-ALB (100%)
- Listener ARN**: arn:aws:elasticloadbalancing:us-east-1:814588432081:listener/app/APPLB/0f7e44b2de1b5fb/1cf15de0ff6066c

The bottom navigation bar includes CloudShell, Feedback, and Console Mobile App.

The screenshot shows the AWS Route 53 console. The left sidebar navigation includes Hosted zones, Global Resolver, VPC Resolver, and Domains. The main content area displays the change info details for a specific hosted zone entry. Key details shown include:

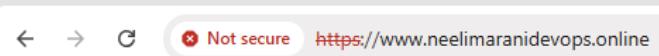
- ID**: /change/C0902774233CY06S8DN0T
- Submitted at**: January 31, 2026, 20:44 (UTC+05:30)
- Status**: INSYNC
- Comment**: -

The bottom navigation bar includes CloudShell, Feedback, and Console Mobile App.

Load Balancers-Task

Record name	Type	Routing policy	Differentially tracked	Alias	Value
neelimaranidevops.online	A	Simple	-	Yes	dua...
neelimaranidevops.online	NS	Simple	-	No	ns-1 ns-4 ns-1 ns-7
neelimaranidevops.online	SOA	Simple	-	No	ns-1
_695c1f0328ecb084a24870d6...	CNAME	Simple	-	No	_8a...
www.neelimaranidevops.online	A	Simple	-	Yes	dua...

- 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

The screenshot shows the AWS S3 Bucket Permissions Overview page for a bucket named 'neelima1998'. The 'Permissions' tab is selected. Under 'Access finding', it says 'Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#)'. Below that, under 'Block public access (bucket settings)', it says 'Block all public access' is off. There is a link to 'Individual Block Public Access settings for this bucket'. The bottom navigation bar includes CloudShell, Feedback, Console Mobile App, Privacy, Terms, and Cookie preferences.

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

The screenshot shows the AWS S3 Bucket Policy Editor page for the 'neelima1998' bucket. The 'Policy' tab is selected. On the left, there is a JSON editor showing a policy document with 19 lines. On the right, there is a sidebar with a 'Select a statement' section and a button '+ Add new statement'. The bottom navigation bar includes CloudShell, Feedback, Console Mobile App, Privacy, Terms, and Cookie preferences.

```
1 Version: "2012-10-17",
2 Statement: [
3   {
4     Effect: "Allow",
5     Principal: {
6       Service: "logdelivery.elasticloadbalancing.amazonaws.com"
7     },
8     Action: "s3:PutObject",
9     Resource: "arn:aws:s3:::neelima1998/*",
10    Condition: {
11      StringEquals: {
12        "s3:x-amz-acl": "bucket-owner-full-control"
13      }
14    }
15  }
16 ]
17 ]
18 }
19 ||
```

Load Balancers-Task

The screenshot shows the 'Edit load balancer attributes' page for an Application Load Balancer named 'APPLB'. Under the 'Monitoring' section, the 'Access logs' checkbox is selected, and the S3 URI is set to 's3://neelima1998'. There are other options like 'Connection logs' and 'Health check logs' which are not selected. At the bottom right, there are 'Cancel' and 'Save changes' buttons.

- 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.

The screenshot shows the 'Object overview' for a log file named '814588432081_elasticloadbalancing_us-east-1_app.APPLB.0f76e44b2de1b5fb_20260131T1330Z_3.214.118.42_3pb145vg.log.gz'. The object was created on January 31, 2026, at 19:00:03 UTC+05:30. The ARN is '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_3pb145vg.log.gz'. The S3 URI is 's3://neelima1998/AWSLogs/814588432081/elasticloadbalancing/us-east-1/2026/01/31/814588432081_elasticloadbalancing_us-east-1_app.APPLB.0f76e44b2de1b5fb_20260131T1330Z_3.214.118.42_3pb145vg.log.gz'.

- To verify Go to s3 bucket →AWSlogs .