

Load balancer

Load balancing in the context of AWS involves distributing incoming application traffic across multiple resources to enhance availability and prevent overloading. AWS provides various types of load balancers, each serving specific use cases:

Application Load Balancer (ALB): Operates at the application layer and is designed for routing HTTP/HTTPS traffic. It is suitable for modern application architectures, providing advanced routing capabilities and supporting content-based routing.

Network Load Balancer (NLB): Works at the transport layer and is optimized for handling TCP, UDP, and TLS traffic. NLB provides high-throughput and low-latency performance, making it ideal for scenarios that require scalability and efficiency.

Classic Load Balancer (CLB): The legacy load balancer that distributes traffic across EC2 instances based on either the application or network layer. While still available, it is recommended to use ALB or NLB for more advanced features.

Gateway Load Balancer (GWLB): Provides private connectivity between virtual appliances in a service provider VPC and application resources. It facilitates secure communication and is used in scenarios where advanced networking and security services are required.

1. Create **multiple instances** for distributing the traffic.

Home page instance = 4

Laptop page instance = 3

Mobile page instances = 2

| Instances (9) Info | | | | | | | | | Refresh Connect | | Instance state ▼ | Actions ▼ | Launch instances ▼ |
|---|-------------------|---------------------|----------------|---|---|---------------|----------------|-------------------------------|---|--|------------------|-----------|------------------------------------|
| <input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> | | | | | | | | | Any state ▼ | | < 1 > ⚙ | | |
| <input type="checkbox"/> | Name ↗ | Instance ID | Instance state | | | Instance type | Status check | Alarm status | Availability Zone | | | | |
| <input type="checkbox"/> | home_instance 1 | i-07f82997a3bab7dbd | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | home_instance 2 | i-0103cbc57b6f14096 | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | home_instance 3 | i-04648eaa741291950 | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | home_instance 4 | i-0e6e72146ab734b5b | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | mobile_instance 1 | i-0a0c390f07ccaf5c0 | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | mobile_instance 2 | i-0e7f587eaa112f71a | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | laptop_instance 1 | i-0a1a8f981eb9ed36d | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | laptop_instance 2 | i-0955537be443c2b34 | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |
| <input type="checkbox"/> | laptop_instance 3 | i-0c7d27b9b0fd02a6b | Running | 🔍 | 🔍 | t2.micro | 🕒 Initializing | View alarms + | us-east-1a | | | | |

2. Add user data as per your requirement while creating the instance.....

#For Home page

```
#!/bin/bash

yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "this is home page $HOSTNAME" > /var/www/html/index.html
```

#For laptop page

```
#!/bin/bash

yum install httpd -y
systemctl start httpd
systemctl enable httpd
mkdir /var/www/html/laptop
echo "this is laptop page $HOSTNAME" > /var/www/html/laptop/index.html
```

#For mobile page

```
#!/bin/bash

yum install httpd -y
systemctl start httpd
systemctl enable httpd
mkdir /var/www/html/mobile
echo "this is mobile page $HOSTNAME" > /var/www/html/mobile/index.html
```

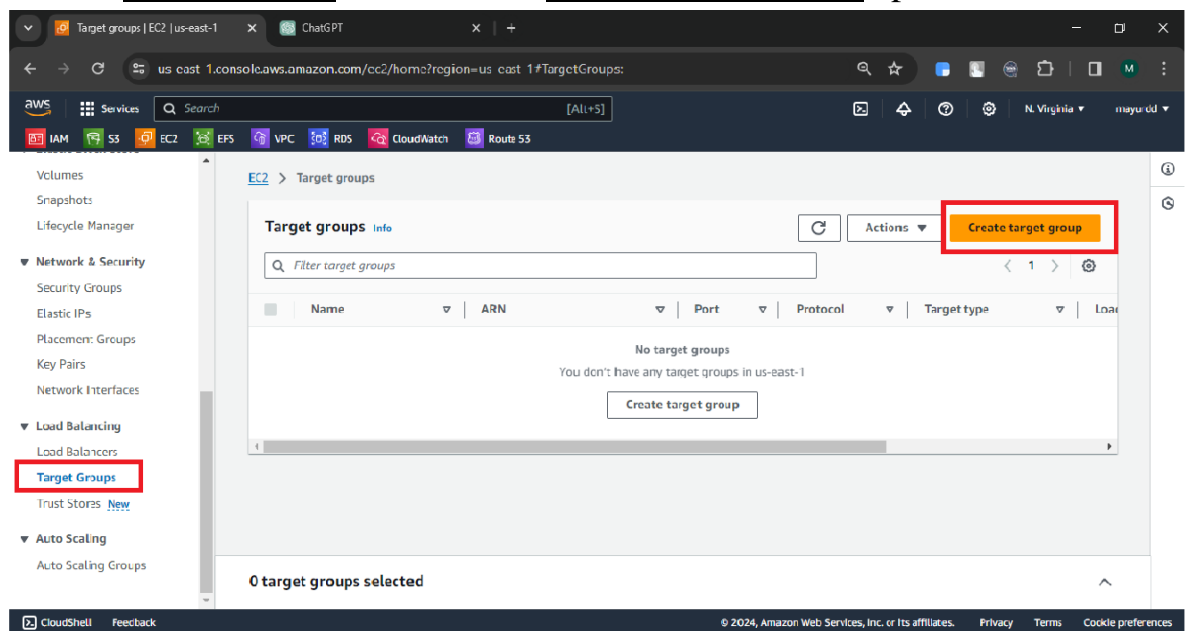
3. After successfully creating multiple instances add this instances **in target group**.

Use of target Group: - A target group in an EC2 Load Balancer is a logical grouping of instances that receive incoming traffic. It plays a crucial role in distributing traffic effectively across multiple instances to ensure optimal performance and availability.

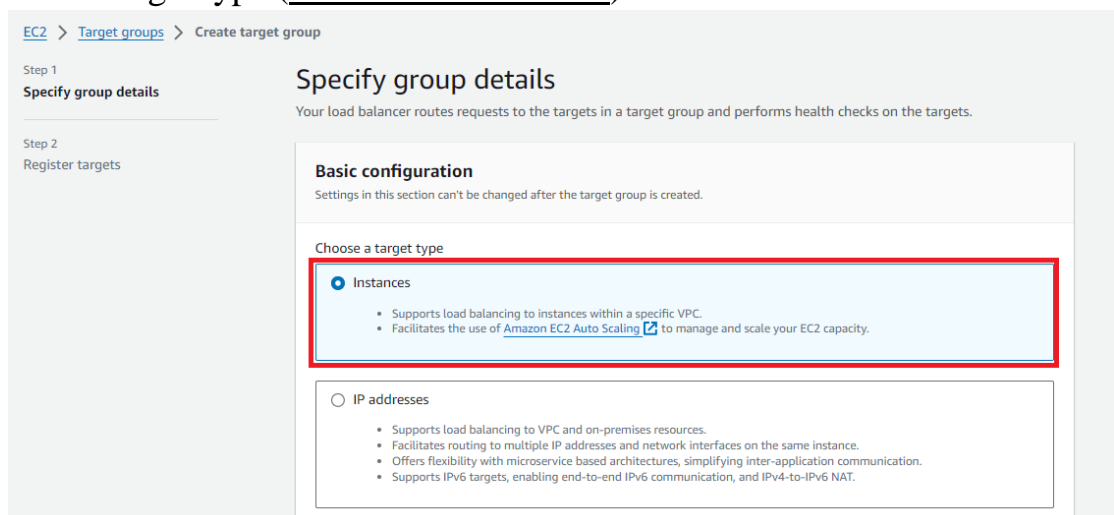
Here's how target groups are utilized.

- **Routing Traffic:** The target group determines which instances should receive traffic from the Load Balancer. It defines the set of registered instances that can handle requests.
- **Health Checks:** Target groups enable health checks on instances, monitoring their status. Instances failing health checks are automatically removed from the load-balanced pool, ensuring that traffic is directed only to healthy instances.
- **Auto Scaling Integration:** When used in conjunction with Auto Scaling groups, target groups facilitate the dynamic addition or removal of instances based on demand. Auto Scaling responds to changes in demand by adjusting the number of instances in the target group.
- **Load Balancing Algorithms:** Target groups allow you to define load balancing algorithms to distribute traffic evenly, enhancing the overall performance and resource utilization.
- **Security Groups Integration:** The Network Load Balancer manages traffic from the security groups associated with instances in the target group, ensuring a secure flow of data.

4. Click on **Target group** and click on **Create target group** option



5. Select target type (in this case **instances**)



6. Specify name and the port

Target group name

homegroup

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

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP 80

IP address type

7. Select the Health check configuration....
(health check means Aws check the instances are running or not In background, in not then it will not forward the traffic to that instance. Also check the Advance health option.....)

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

8. Click on next option

Attributes

Information icon Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► **Tags - optional**

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel **Next**

9. Select the instance which you want to add in first target group.

Register targets

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 (4/9)

Filter instances

| | Instance ID | Name | State | Security groups | Zone |
|-------------------------------------|---------------------|-------------------|---------|------------------|------|
| <input checked="" type="checkbox"/> | i-07f82397a3bab7dbd | home_instance 1 | Running | launch-wizard-15 | us- |
| <input checked="" type="checkbox"/> | i-0103cbc57b6f14096 | home_instance 2 | Running | launch-wizard-15 | us- |
| <input checked="" type="checkbox"/> | i-04648eaa741291950 | home_instance 3 | Running | launch-wizard-15 | us- |
| <input checked="" type="checkbox"/> | i-0e6e72146ab734b5b | home_instance 4 | Running | launch-wizard-15 | us- |
| <input type="checkbox"/> | i-0a0c390f07ccaf5c0 | mobile_instance 1 | Running | launch-wizard-20 | us- |
| <input type="checkbox"/> | i-0e7f537eaa112f71a | mobile_instance 2 | Running | launch-wizard-20 | us- |
| <input type="checkbox"/> | i-0a1a8f981eb9ed36d | laptop_instance 1 | Running | launch-wizard-21 | us- |

10. Click on **include as pending below** option.

4 selected

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

1-65535 (separate multiple ports with commas)

include as pending below

11. Click on **create group target** option

Review targets

Targets (4) Remove all pending

 Show only pending < 1 > ⚙

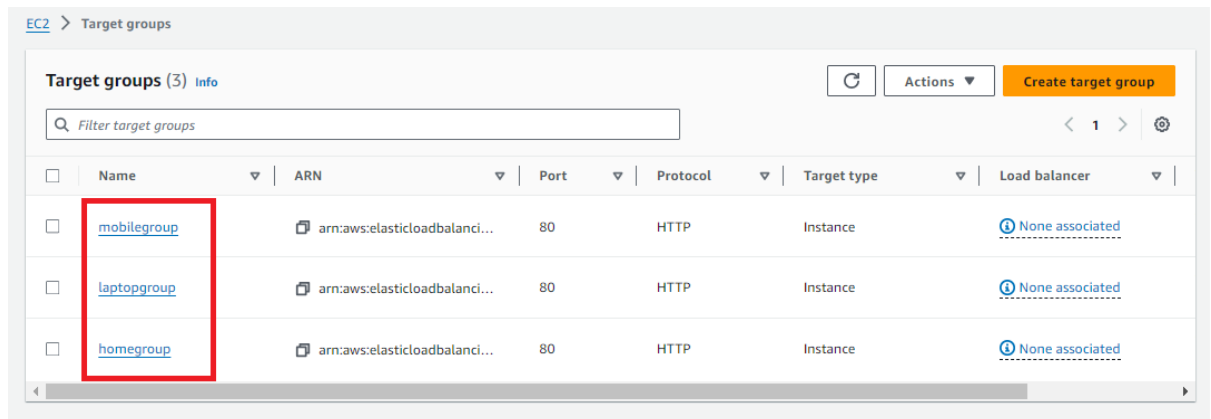
12. Target group created successfully....

EC2 > Target groups

Target groups (1) Info Refresh Actions Create target group

 < 1 > ⚙

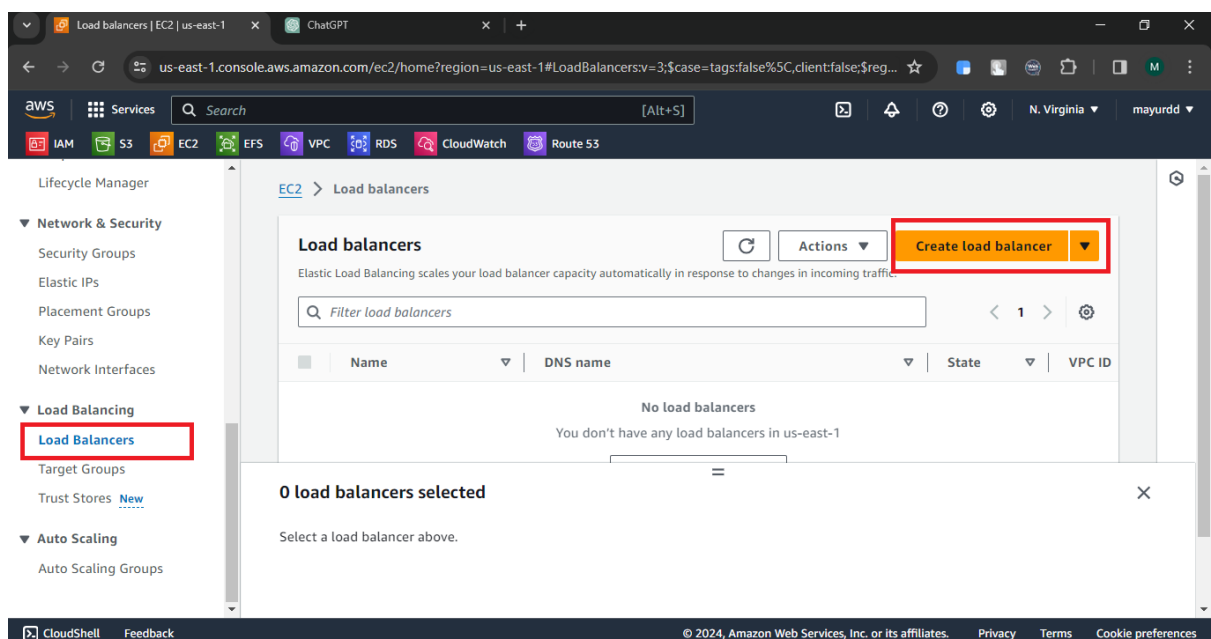
13. Perform same process and create another **three** target groups.



14. After creating Target group we need to create load balancer....

- Acts as the entry point for incoming traffic.
- Distributes traffic across multiple instances or targets, enhancing availability and fault tolerance.
- Manages routing decisions based on configured algorithms.
- Operates at the application, network, or transport layer, depending on the type of load balancer (Application Load Balancer, Network Load Balancer).

15. Click on **Load Balancers** and **Create load balancer** option




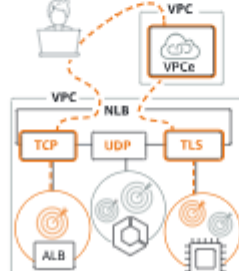

16. Click on Application load balancer

EC2 > Load balancers > Compare and select load balancer type

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

| Application Load Balancer | Network Load Balancer | Gateway Load Balancer |
|--|---|---|
|  <p>Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.</p> <p>Create</p> |  <p>Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.</p> <p>Create</p> |  <p>Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.</p> <p>Create</p> |

► Classic Load Balancer - previous generation

17. Basic configuration....

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
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type | Info
Select the type of IP addresses that your subnets use.

☒ IPv4
Recommended for internal load balancers.

☐ Dualstack
Includes IPv4 and IPv6 addresses.

18. Click on all zones (because we are not created all instances in one zone)

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

vpc-01a4a30d4e2c789b7
IPv4: 172.31.0.0/16

Mappings [Info](#)

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

- ☐ us-east-1a (use1-az4)
- ☐ us-east-1b (use1-az6)
- ☐ us-east-1c (use1-az1)
- ☐ us-east-1d (use1-az2)
- ☐ us-east-1e (use1-az3)
- ☐ us-east-1f (use1-az5)

19. Select security group (note port 80 should be enabled in selected security group)

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

launch-wizard-22
sg-0aea512113571f55e VPC: vpc-01a4a30d4e2c789b7

20. Select the target group, port and protocol

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 [Remove](#)

Protocol: HTTP Port: 80 1-65535

Default action [Info](#)

Forward to: homegroup
Target type: Instance, IPv4 HTTP [Create target group](#)

Listener tags - *optional*

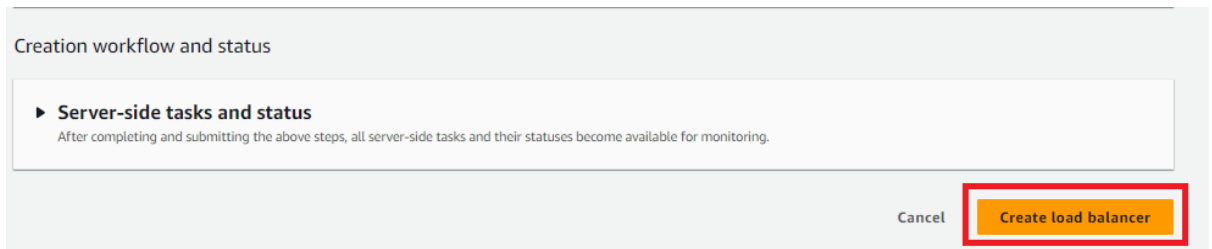
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)

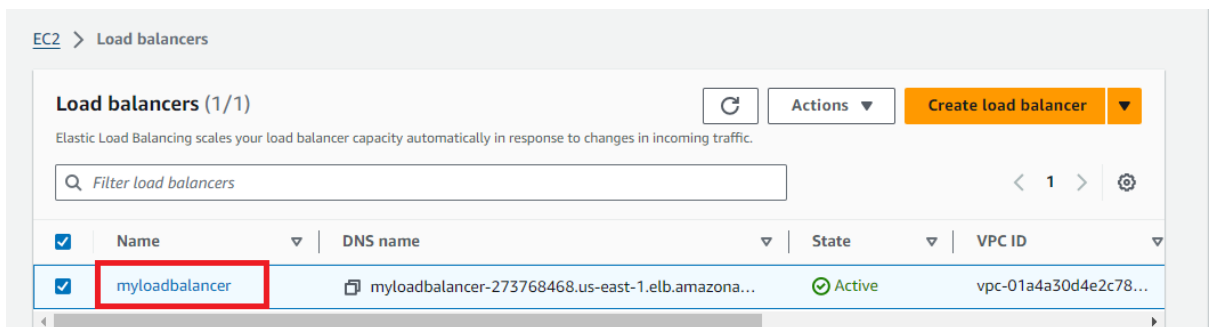
You can add up to 50 more tags.

[Add listener](#)

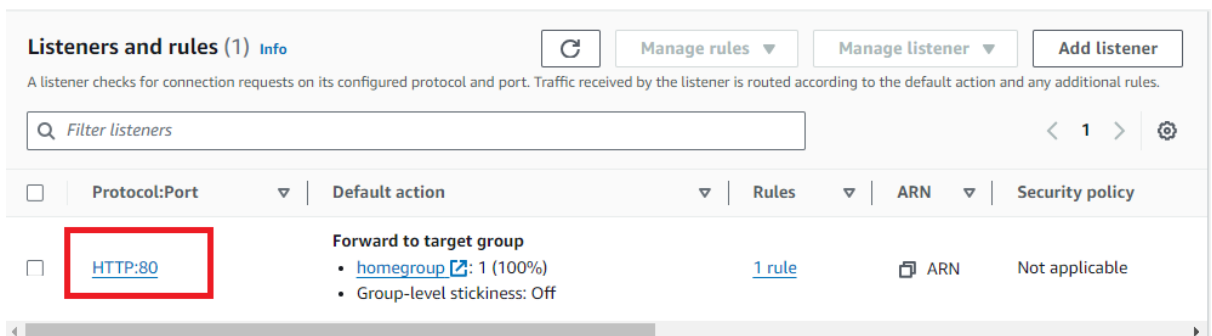
21. Click on create load balancer



22. Load balancer Created successfully....



23. Select the Created load balancer and click on (HTTP:80) option



24. The Default Rule is already created. Click on **Add rule** option.

The screenshot shows the 'Listener rules' page in the AWS Management Console. At the top, there are tabs for 'Rules' and 'Tags'. Below the tabs, the page title is 'Listener rules (1)' with an 'Info' link. To the right of the title are links for 'Rule limits', a refresh icon, an 'Actions' dropdown menu, and a red-bordered 'Add rule' button. Below this is a search bar labeled 'Filter rules' and a settings gear icon. A table lists the existing rules. The first rule is 'Default', which is highlighted with a black box. It has a priority of 'Last (default)' and the condition 'If no other rule applies'. The actions are 'Forward to target group', which includes 'homegroup' (100%) and 'Group-level stickiness: Off'.

| <input type="checkbox"/> | Name tag | Priority | Conditions (If) | Actions (Then) |
|--------------------------|----------|----------------|--------------------------|--|
| <input type="checkbox"/> | Default | Last (default) | If no other rule applies | Forward to target group <ul style="list-style-type: none">homegroup: 1 (100%)Group-level stickiness: Off |

25. Give the name and click on next

The screenshot shows the 'Add rule' page in the AWS Management Console. The page has a breadcrumb trail: 'EC2 > Load balancers > myloadbalancer > HTTP:80 listener > Add rule'. On the left, there are five steps: 'Step 1: Add rule', 'Step 2: Define rule conditions', 'Step 3: Define rule actions', 'Step 4: Set rule priority', and 'Step 5: Review and create'. The main content area is titled 'Add rule' with an 'Info' link. Below the title is a description: 'Define the rule and then review it in the context of the other rules on this listener.' There are two main sections: 'Listener details: HTTP:80' and 'Name and tags'. The 'Name and tags' section has a 'Name' field with the value 'laptop-page' and an 'Add additional tags' link. At the bottom right, there are 'Cancel' and 'Next' buttons. The 'Next' button is highlighted with a red box.

26. Click on **Add condition** option

The screenshot shows the 'Conditions' page in the AWS Management Console. The page title is 'Conditions (0)' with a 'Rule limits' link. Below the title, there is a message: 'No conditions' and 'No conditions to display.' At the bottom, there is a red-bordered 'Add condition' button.

27. Select the Preferred options and click on confirm

Add condition Rule limits ×

Rule condition types
Route traffic based on the condition type of each request. Each rule can include one of each of the following conditions: host-header, path, http-request-method and source-ip. Each rule can include one or more of each of the following conditions: http-header and query-string.


Path ▼

Path
Define the path. For example: /item/*. Case sensitive.
is /laptop/ 🗑️
Maximum 128 characters. Allowed characters are `[a-z]`, `[A-Z]`, `[0-9]`; the following special characters: `[-.$/~"*@:~]`; `&` (using `&`); and wildcards (`*` and `?`).

Add new value

You can add up to 4 more condition values for this rule.

Cancel



Confirm

28. Click on next option

Define rule conditions [Info](#)
Requests reaching this rule must match all specified conditions for the rule to apply. At least 1 condition is required.

► **Listener details:** HTTP:80

Conditions (1) Rule limits


Edit Delete Add condition

Path (1) [Info](#) 🗑️
If
Path
is
/laptop/

AND

Cancel

Previous



Next

29. Select the target group and click on next

Actions

Action types

Routing actions

☒ Forward to target groups ☐ Redirect to URL ☐ Return fixed response

Forward to target group [Info](#)
Choose a target group and specify routing weight or [Create target group](#).

Target group

laptopgroup HTTP Weight **1** Percent **100%**
Target type: Instance, IPv4 0-999

You can add up to 4 more target groups.

Group-level stickiness [Info](#)
If a target group is sticky, requests routed to it remain in that target group for the duration of the session. Individual target stickiness is a configuration of the target group.

☐ Turn on group-level stickiness

30. Set the priority is 1

EC2 > Load balancers > myloadbalancer > HTTP:80 listener > Add rule

Step 1
[Add rule](#)

Step 2
[Define rule conditions](#)

Step 3
[Define rule actions](#)

Step 4
Set rule priority

Step 5
[Review and create](#)

Set rule priority [Info](#)

Each rule has a priority. Rules are evaluated in priority order from the lowest value to the highest value. The default rule is evaluated last. You can change the priority of a non-default rule at any time. You can't change the priority of the default rule.

► **Listener details:** HTTP:80

Rule: laptop-page

Priority
Rule priority controls the evaluation order of a rule within the listener's set of rules. You can leave gaps in priority numbers.

1
1 - 50000

31. Click on create option

Step 2
[Define rule conditions](#)

Step 3
[Define rule actions](#)

Step 4
[Set rule priority](#)

Step 5
Review and create

Listener details: HTTP:80

Rule details: laptop-page Edit ▼

| | | |
|---------------|---|--|
| Priority 1 | Conditions (If) If request matches all: Path Pattern is /laptop/ | Actions (Then) Forward to target group <ul style="list-style-type: none">laptopgroup ↗: 1 (100%)Group-level stickiness: Off |
|---------------|---|--|

Rule ARN
Pending

Rule tags (1) Edit

Tags can help you manage, identify, organize, search for and filter resources.

| Key | Value |
|------|-------------|
| Name | laptop-page |

Cancel Previous **Create**

32. New rule added successfully....

Listener rules (2) [Info](#) [Rule limits](#) [Refresh](#) [Actions ▼](#) [Add rule](#)

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

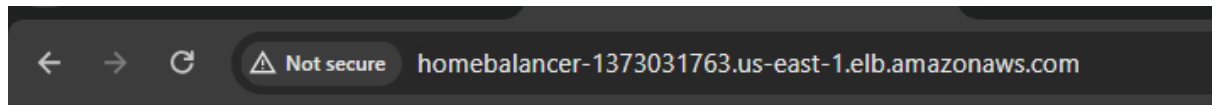
| <input type="checkbox"/> | Name tag | Priority | Conditions (If) | Actions (Then) | ARN |
|--------------------------|--------------------|----------------|---------------------------------|--|-----|
| <input type="checkbox"/> | laptop-page | 1 | Path Pattern is /laptop/ | Forward to target group <ul style="list-style-type: none">laptopgroup ↗: 1 (100%)Group-level stickiness: Off | ARN |
| <input type="checkbox"/> | <i>Default</i> | Last (default) | <i>If no other rule applies</i> | Forward to target group <ul style="list-style-type: none">homegroup ↗: 1 (100%)Group-level stickiness: Off | ARN |

33. Follow same steps and create rule for **mobile** page. New rule is added

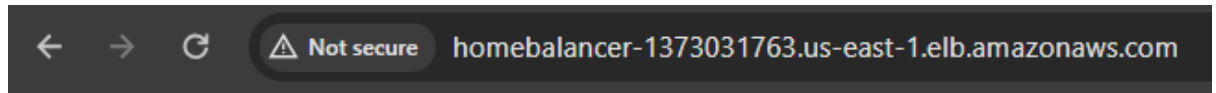
| | | | | | |
|--------------------------|--------------------|----------------|--------------------------|---|-----|
| <input type="checkbox"/> | laptop-page | 1 | Path Pattern is /laptop/ | Forward to target group <ul style="list-style-type: none">laptopgroup ↗: 1 (100%)Group-level stickiness: Off | ARN |
| <input type="checkbox"/> | mobile-page | 2 | Path Pattern is /mobile/ | Forward to target group <ul style="list-style-type: none">mobilegroup ↗: 1 (100%)Group-level stickiness: Off | ARN |
| <input type="checkbox"/> | Default | Last (default) | If no other rule applies | Forward to target group <ul style="list-style-type: none">homegroup ↗: 1 (100%)Group-level stickiness: Off | ARN |

34.Results

#Home page using load balancer

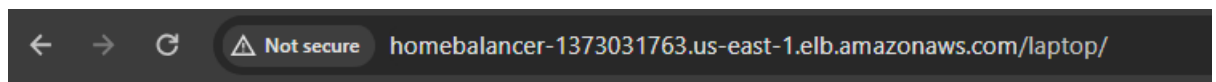


this is main homepage ip-172-31-25-183.ec2.internal

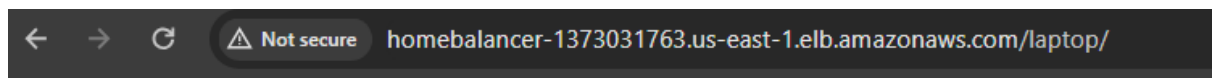


this is main homepage ip-172-31-26-192.ec2.internal

#Laptop page using load balancer

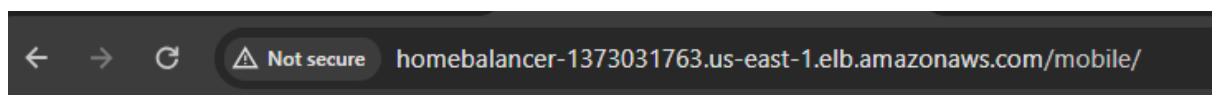


this is laptop page ip-172-31-29-60.ec2.internal

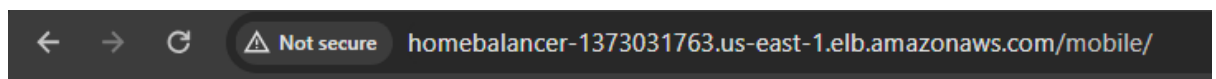


this is laptop page ip-172-31-27-97.ec2.internal

#Mobile page using load balancer



this is mobile page ip-172-31-23-126.ec2.internal



this is mobile page ip-172-31-24-42.ec2.internal