

Static website hosting => userdata
Loadbalancer, Monolith, Microservices

Website: Collection of web pages (html pages)
Static website (gives same response to every user) and Dynamic website (gives response based on user activity)

Webserver -> used to host/run our website

For Static websites we have httpd, apache2
For Dynamic websites: tomcat, IIS

Hosting website using httpd:

```
[ec2-user@ip-172-31-13-80 ~]$ sudo systemctl start httpd
```



It works!

```
[ec2-user@ip-172-31-13-80 ~]$ cd /var/www/html
```

```
[ec2-user@ip-172-31-13-80 html]$ sudo vi index.html
```



Welcome to AWS custom message!!

```
sudo yum update -y
sudo yum install httpd
sudo systemctl start httpd
```

Note: Enable HTTP: 80 in security group inbound rules

Access our website using EC2 VM public IP

To modify the content we can navigate:
`cd /var/www/html`
`sudo vi index.html`

Access again our website using EC2 instance public IP
User-data in EC2 VM:
--> used to execute script while launching machine
--> user data will execute only once

Create a new EC2 VM with below user data

```
#!/bin/bash
```

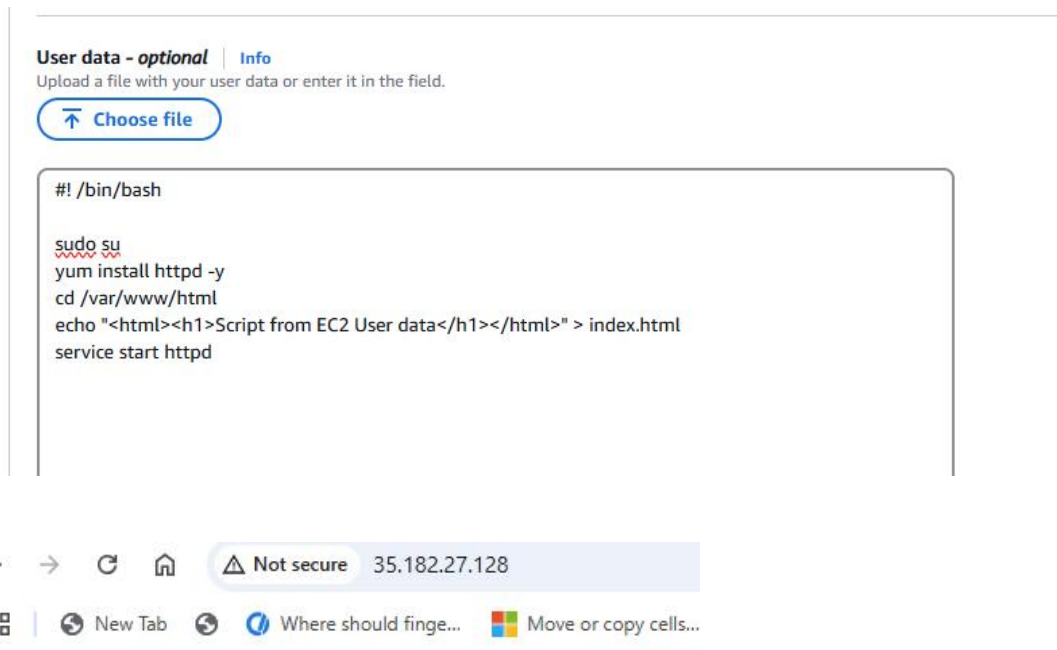
```
sudo su
```

```
yum install httpd -y
```

```
cd /var/www/html
```

```
echo "<html><h1>Script from EC2 User data</h1></html>" > index.html
```

```
systemctl start httpd
```



Script from EC2 User data

Disadvantages of having one server for our application:

One server must handle all the incoming requests

High burden on server, which might result in delay in responses

Can lead to server crash (single point of failure)

All these problems will lead to business loss

Deploying same application into multiple servers

App is now deployed in say 3 servers

How can we divert the traffic to multiple servers?

Load balancer: All requests to application will be diverted to multiple servers (Round robin)

Application will run on multiple servers

-> load will be distributed

-> Fast performance

-> High availability

Load balancer: it is used to distribute incoming load / requests to multiple servers in round robin technique

There are different types of Load balancers in AWS:

1. Application load balancer (http & https)

2. Network load balancer

3. Gateway load balancer

Classic Load Balancer (outdated / old gen)

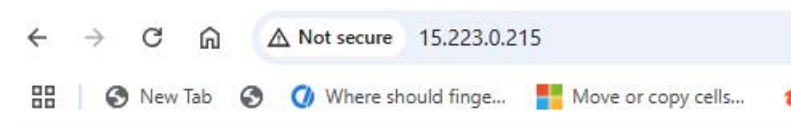
Practical task on Load balancer:

User data - optional | [Info](#)
Upload a file with your user data or enter it in the field.

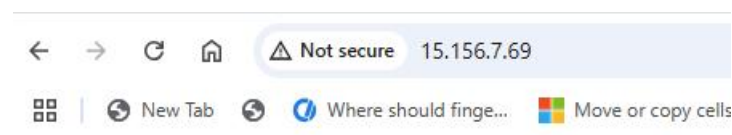
[Choose file](#)

```
#!/bin/bash

sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Banking Application Server 1</h1></html>" > index.html
systemctl start httpd
```



Banking Application Server 1



Banking Application Server 2

Target group: List of servers running our application

These two servers together is called as a Target group

Create two VMs and deploy Banking Application

Add these instances to one target group -> List of servers running our application

Target groups [info](#)

Q

Filter target groups

Name

ARN

Port

Protocol

Target type

No target groups

You don't have any target groups in ca-central-1

Create target group

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

☒ Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

☐ IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

☐ Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Target group name

BankingApplication

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 created. This choice cannot be changed after creation

HTTP

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

☒ IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

☐ IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this

-
vpc-0a752647f0a021f2e
IPv4 VPC CIDR: 172.31.0.0/16

Protocol version

☒ HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

Register targets

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

Available instances (2/3)

<input type="checkbox"/>	Instance ID	Name	State
<input checked="" type="checkbox"/>	i-0e52e3dbd40588c34	BankingServer2	Running
<input checked="" type="checkbox"/>	i-06556d6d83bafdaaa	BankingServer1	Running
<input type="checkbox"/>	i-0f87d8181852ad376	DevOpsCoursePractice	Running

Click ->Include as pending below -> Create Target Group

ly mitigation: **Not applicable**

ling to the target group's health check settings. Anomaly detection is

< 1 >

Administrative o...	Override details	Launch...	Anomaly
-		March 9, ...	Normal
-		March 9, ...	Normal

Register Targets

Go to Load Balancers

Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

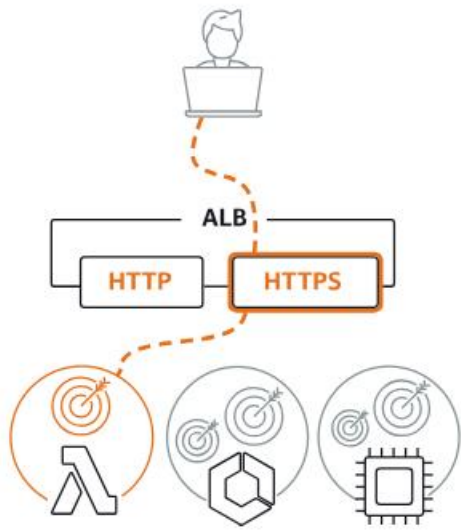
<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones
No load balancers					
You don't have any load balancers in ca-central-					
<input type="button" value="Create load balancer"/>					

Compare and select load balancer

A complete feature-by-feature comparison along with det

Load balancer types

Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic

N

CI

ne

at

Create Application Load Balancer
Internet facing and leave other options as it is

Basic configuration

Load balancer name

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

BankingLB

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 is publicly resolvable.
 - Requires a public subnet.

- ☐ **Internal**
- Serves traffic from within your VPC.
 - Has private IP addresses.
 - DNS name is private.
 - Compatible with Amazon EC2 instances.

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 support the selected IP address type.

- ☒ **IPv4**
Includes only IPv4 addresses.
- ☐ **Dualstack**
Includes IPv4 and IPv6 addresses.
- ☐ **Dualstack without public IPv4**
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers.

Network mapping [Info](#)

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

VPC [Info](#)

The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must reside. [create a VPC](#).

vpc-0a752647f0a021f2e
ID: vpc-0a752647f0a021f2e

Select Availability Zones

The IPAM pool you choose will be the preferred source of public IPv4 addresses:

Availability Zones and subnets [Info](#)

Select at least two Availability Zones and a subnet for each zone. A load balancer n

☒ ca-central-1a (cac1-az1)

Subnet

Only CIDR blocks corresponding to the load balancer IP address type are used. /

subnet-02a70284a8b5c8bb9

IPv4 subnet CIDR: 172.31.16.0/20

☒ ca-central-1b (cac1-az2)

Subnet

Only CIDR blocks corresponding to the load balancer IP address type are used. /

subnet-038c457fd7226e0ec

IPv4 subnet CIDR: 172.31.0.0/20

☒ ca-central-1d (cac1-az4)

Subnet

Only CIDR blocks corresponding to the load balancer IP address type are used. /

subnet-05422c9c80857b14b

IPv4 subnet CIDR: 172.31.32.0/20

Select Security Groups, Target 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

DevOps-sg

sg-031a081efd38c0e3a VPC: vpc-0a752647f0a021f2e

default

sg-0483bbb02e36e7efe VPC: vpc-0a752647f0a021f2e

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 regist

▼ Listener HTTP:80

Protocol

HTTP

Port

80

1-65535

Default action

Forward to

BankingApplication

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.

subnet-038c457fd7226e0ec • ca-central-1d subnet-05422c9c80857b14b	Tags Edit
---	---------------------------

ill (WAF): -

your load balancer. You can view and edit them after creating the load balancer.

ps, all server-side tasks and their statuses become available for monitoring.

[Cancel](#)
[Create load balancer](#)

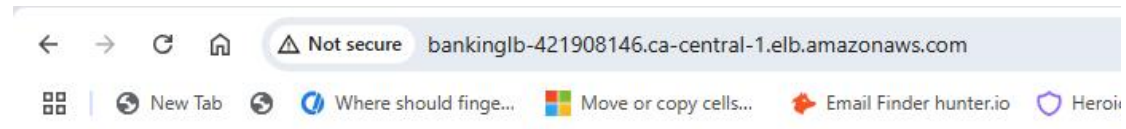
Scroll down then Create Load Balancer

It gives public DNS

BankingLB

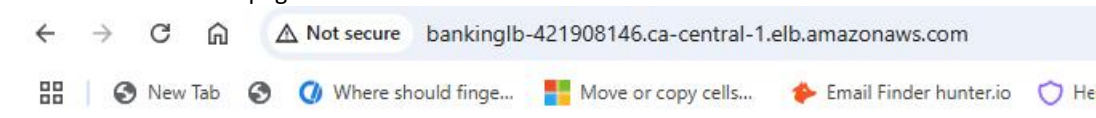
▼ Details			
Load balancer type Application	Status Active	VPC vpc-0a752647f0a021f2e	Load bala IPv4
Scheme Internet-facing	Hosted zone ZQSVJUPUGJ1EY	Availability Zones subnet-02a70284a8b5c8bb9 ca-central-1a (cac1-az1) subnet-05422c9c80857b14b ca-central-1d (cac1-az4) subnet-038c457fd7226e0ec ca-central-1b (cac1-az2)	Date creat March 9, 2
Load balancer ARN arn:aws:elasticloadbalancing:ca-central-1:577638386543:loadbalancer/app/BankingLB/1f13af3eb6e6576a		DNS name Info BankingLB-421908146.ca-central-1.elb.amazonaws.com (A Record)	

DNS



BankingApplication Server 2

When we refresh the page:



Banking Application Server 1

If you decide to add a new Server say Server 3, we add to the Target group that's it Load balancer will be updated as well

Architectural design patterns:
Monolithic Vs Microservices

Monolithic -> developing all functionalities in single application

Deploying one application into multiple servers, it is Monolithic

For Monolithic app usually we need one target group

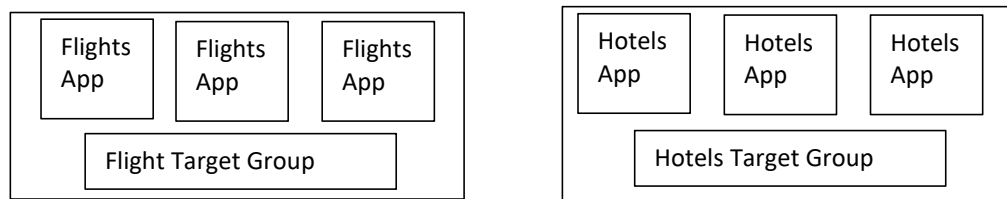
For Microservices: dividing functionalities into multiple APIs . One application is divided into micro applications.

BookingApplication -> features like hotels, flights, car rentals everything in one application. This is monolithic

(Monolithic) App -> UI <- end user -> UI -> Hotels App, Car rentals App, Flights App (Microservices)

So many requests are coming to Application -> we have a Load balancer in between. In Target group, we have 3 applications -> this is Monolithic

In Microservices, I have 3 car rental applications deployed on multiple servers, 3 hotel applications deployed on multiple servers, 3 flight applications deployed on multiple servers



So it is challenging for Load balancer. that's why we need to configure "Routing"

```
#!/bin/bash
sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Flights Service 1</h1></html>" > index.html
systemctl start httpd
```

```
#!/bin/bash
sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Flights Service 2</h1></html>" > index.html
systemctl start httpd
```



```
#!/bin/bash
sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Hotels Service 1</h1></html>" > index.html
systemctl start httpd
```

```
#!/bin/bash
sudo su
yum install httpd -y
```

```
cd /var/www/html
echo "<html><h1>Hotels Service 2</h1></html>" > index.html
systemctl start httpd
```

We create 2 servers for each service then we register into Target groups

Target groups (2) [Info](#)

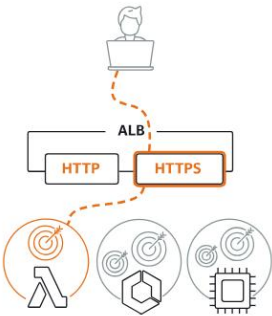
<input type="checkbox"/>	Name	ARN	Port	Protocol
<input type="checkbox"/>	Hotels-TG	 <code>arn:aws:elasticloadbalancin...</code>	80	HTTP
<input type="checkbox"/>	Flights-TG	 <code>arn:aws:elasticloadbalancin...</code>	80	HTTP

For a normal HTTP application, Application Load Balancer is recommended

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

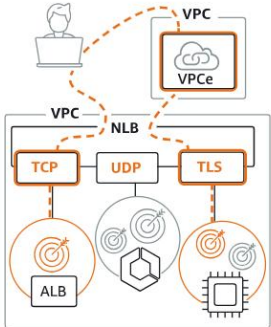
Load balancer types

Application Load Balancer [Info](#)




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

Network Load Balancer [Info](#)



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

Gateway Load Balancer [Info](#)



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.

[Create](#)

First I add only Flights-TG

IPv4 subnet CIDR: 172.31.0.0/20

☒ **ca-central-1d (cac1-az4)**
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-05422c9c80857b14b
IPv4 subnet CIDR: 172.31.32.0/20

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

DevOps-sg
sg-031a081efd38c0e3a VPC: vpc-0a752647f0a021f2e

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

▼ Listener **HTTP:80**

Protocol: HTTP Port: 80

Default action: [Info](#)
Forward to: Flights-TG
Target type: Instance, IPv4
HTTP

[Create target group](#)

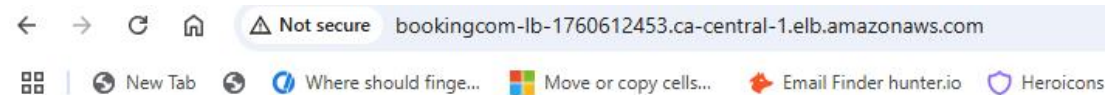
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

<input type="checkbox"/>	Name	State	VPC ID	Availability zones
<input type="checkbox"/>	BookingCom-LB	Provisioning...	vpc-0a752647f0a021f2e	3 Availability zones



Flights Service 1



Flights Service 2

Click on Add Listener

Load balancer: **BookingCom-LB**

Details | **Listeners and rules** | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

Listeners and rules (1) [Info](#) [Manage rules](#) [Manage listener](#) [Add listener](#)

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Filter listeners

<input type="checkbox"/>	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
<input type="checkbox"/>	HTTP:80	Forward to target group • Flights-TG 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applicable

Hotels added

Default actions | [Info](#)
 The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Routing actions

☒ Forward to target groups
 ☐ Redirect to URL

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

Target group

Hotels-TG
 Target type: Instance, IPv4

HTTP

[Add target group](#)

You can add up to 4 more target groups.

Target group stickiness | [Info](#)
 Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on

☐ Turn on target group stickiness

[Default attributes are applied when adding a listener. You can edit them after adding the listener.](#)

Two listeners

Listeners and rules						
Listeners and rules (2) Info						
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action.						
<input type="text" value="Filter listeners"/>						
<input type="checkbox"/>	Protocol:Port		Default action	Rules	ARN	Security
<input type="checkbox"/>	HTTP:81	⚠ Not reachable	Forward to target group <ul style="list-style-type: none"> Hotels-TG: 1 (100%) Target group stickiness: Off 	1 rule	ARN	Not app
<input type="checkbox"/>	HTTP:80		Forward to target group <ul style="list-style-type: none"> Flights-TG: 1 (100%) Target group stickiness: Off 	1 rule	ARN	Not app

81 is not reachable so we are going to edit the existing listener

Edit existing target group and add new Target group

Listener configuration

The listener will be identified by the protocol and port.

Protocol

Used for connections from clients to the load balancer.

HTTP

Port

The port on which the load balancer is listening

80

1-65535

Default actions [Info](#)

The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Routing actions

☒ Forward to target groups

☐ Redirect to URL

Forward to target group [Info](#)

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

Target group

Flights-TG

Target type: Instance, IPv4

HTTP

Hotels-TG

Target type: Instance, IPv4

HTTP

[Add target group](#)

You can add up to 3 more target groups.

d port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its regis

Load balancer

[BookingCom-LB](#)

Default actions

Forward to target group

- [Flights-TG](#): 1 (50%)
- [Hotels-TG](#): 1 (50%)
- Target group stickiness: Off

er/app/BookingCom-LB/15287b603181b510/7b559285eeaaad3d

It shows both Hotel and Flight services but we don't want this random decision

Add rule

Load balancer ARN

[arn:aws:elasticloadbalancing:ca-central-1:577638386543:loadbalancer/app/BookingCom-LB/15287b603181b510](#)

DNS name [Info](#)

[BookingCom-LB-1760612453.ca-central-1.elb.amazonaws.com](#) (A Rec

Listeners and rules

Network mapping

Resource map

Security

Monitoring

Integrations

Attributes

Capacity

Tags

Listeners and rules (1/1) [Info](#)

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.



Protocol:Port



Default action



Rules



ARN



Security policy



Def



[HTTP:80](#)

Forward to target group

- [Flights-TG](#): 1 (50%)
- [Hotels-TG](#): 1 (50%)
- Target group stickiness: Off

[1 rule](#)

ARN

Not applicable

Not applicable

Select Query String

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.

Query string

▼

Query string

Route requests based on key:value pairs or values in the query string. Not case sensitive.

Key - optional

Value

is

type

flights

🗑

or

type

hotels

🗑

Valid characters are a-z, A-Z, 0-9 and special characters.
Query parameter must be 1-128 characters.

Valid characters are a-z, A-Z, 0-9 and special characters.
Query parameter must be 1-128 characters.

Add new value

You can add up to 3 more condition values for this rule.

Cancel

Confirm

We have edited the Query String

Edit 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.

Query string

▼

Query string

Route requests based on key:value pairs or values in the query string. Not case sensitive.

Key - optional

Value

is

type

flights

🗑

Valid characters are a-z, A-Z, 0-9 and special characters.
Query parameter must be 1-128 characters.

Valid characters are a-z, A-Z, 0-9 and special characters.
Query parameter must be 1-128 characters.

Add new value

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

Cancel

Confirm

Set Priority to 1

Set rule priority [Info](#)

Each rule has a priority. 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

Listener rules (2) [Info](#)

[Rule limits](#) [Reset priorities](#) [Add gap between priorities ▼](#)

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.

	Name tag	Priority ↗	Conditions (If)	Actions (Then)	ARN	Tags
⋮	mybooking	<input type="text" value="1"/> <small>Priority value must be 1-50,000.</small>	Query String is type=flights	Forward to target group <ul style="list-style-type: none">Flights-TG ↗: 1 (100%)Target group stickiness: Off	Pending	1 tag
	Default	Last (default)	If no other rule applies	Forward to target group <ul style="list-style-type: none">Flights-TG ↗: 1 (100%)Target group stickiness: Off	ARN	0 tags

[Cancel](#) [Previous](#) [Next](#)

Listener rules created

[Rules](#) | [Attributes](#) | [Tags](#)

Listener rules (2) [Info](#)

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

<input type="checkbox"/>	Name tag	Priority ▲	Conditions (If)	Actions (Then)
<input type="checkbox"/>	mybooking	1	Query String is type=flights	Forward to target group <ul style="list-style-type: none">Flights-TG ↗: 1 (100%)Target group stickiness: Off

Forward to target group

When I pass the Query parameter I get only Flights in this case

<http://bookingcom-lb-1760612453.ca-central-1.elb.amazonaws.com/?type=flights>

[←](#) [→](#) [🔄](#) [🏠](#) [⚠ Not secure](#) [bookingcom-lb-1760612453.ca-central-1.elb.amazonaws.com/?type=flights](#)

Flights Service 1

Again add the rule

Add rule [Info](#)

Define the rule and then review it in the context of the other rules on this listener.

► Listener details: HTTP:80

Name and tags [Info](#)

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

Name [Add additional tags](#)

[Cancel](#) [Next](#)

Add rule [info](#)

Define the rule and then review it in the context of the other rules on this listener.

► Listener details: HTTP:80

Name and tags [info](#)

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

Name

hotels

[Add additional tags](#)

[Cancel](#)

[Next](#)

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.

Query string ▼

Query string

Route requests based on key:value pairs or values in the query string. Not case sensitive.

Key - *optional*

Value

is type

hotels



Valid characters are a-z, A-Z, 0-9 and [special characters](#).
Query parameter must be 1-128 characters.

Valid characters are a-z, A-Z, 0-9 and [special characters](#).
Query parameter must be 1-128 characters.

[Add new value](#)

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

[Cancel](#)

[Confirm](#)

Define rule actions [Info](#)

These actions will be applied to requests matching the rule conditions.

► Listener details: HTTP:80

Actions

Action types

Routing actions

☒ Forward to target groups

☐ Redirect to URL

Forward to target group [Info](#)

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

Target group

Hotels-TG

Target type: Instance, IPv4

HTTP ▼

[Add target group](#)

You can add up to 4 more target groups.

Target group stickiness [Info](#)

Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to:

☐ Turn on target group stickiness

► Listener details: HTTP:80

Listener rules (3) [Info](#)

[Rule limits](#)



[Reset priority](#)

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

Name tag	Priority ↗	Conditions (If)	Actions (Then)
mybooking	1	Query String is type=flights	Forward to target group <ul style="list-style-type: none">Flights-TG 1 (100%)Target group stickiness: Off
⋮ hotels	<input type="text" value="2"/> <small>Priority value must be 1-50,000.</small>	Query String is type=hotels	Forward to target group <ul style="list-style-type: none">Hotels-TG 1 (100%)Target group stickiness: Off

We see two rules

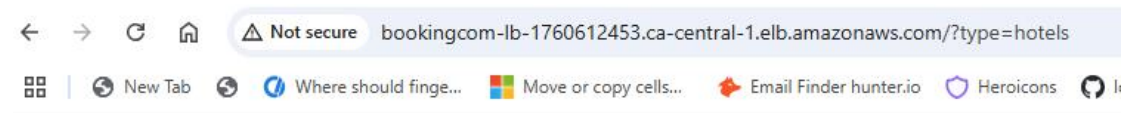
Listener rules (5) [Info](#)

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

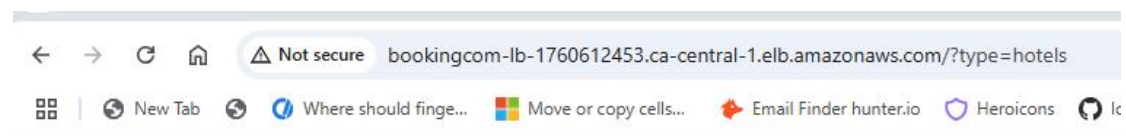
<input type="checkbox"/>	Name tag	Priority ▲	Conditions (If)	Actions (Then)
<input type="checkbox"/>	mybooking	1	Query String is type=flights	Forward to target group <ul style="list-style-type: none">Flights-TG 1 (100%)Target group stickiness: Off
<input type="checkbox"/>	hotels	2	Query String is type=hotels	Forward to target group <ul style="list-style-type: none">Hotels-TG 1 (100%)Target group stickiness: Off

Now we are getting only Hotels Service

<http://bookingcom-lb-1760612453.ca-central-1.elb.amazonaws.com/?type=hotels>



Hotels Service 1



Hotels Service 2

In what situation, we may have to create multiple target groups?

Microservices

Types of Load Balancers:

Application Load Balancers (ALB)

Network Load Balancers (NLB)

Gateway Load Balancers (GLB)

OSI model: Open Systems Interconnections

It contains 7 layers, it will represent how request will transfer from Client to Server

Client -----> Server

<-----

During this communication, there are 7 layers and they are called as OSI layers

Layer 7: Application layer (Application Load Balancer)

Layer 6: Presentation layer

Layer 5: Session layer

Layer 4: Transport layer (Network Load Balancer)

Layer 3: Network layer

Layer 2: Data link layer

Layer 1: Physical layer

Client's request first goes to Application layer (At this point, Application Load Balancer intercepts the request, then it determines where it has to go Hostname, entire path: like Flights or Hotels etc) -> Presentation layer -> Session layer -> Transport layer (say if it is a high performance application where we don't want to decide which Target group to go to etc, something like Gaming application where decision-making is not important then Network Load Balancer comes into picture) -> Network layer -> Data link layer -> Physical layer -> Server (App)

Decision-making -> Application Load Balancer

Non-decision-making something like Gaming then we go with Network Load Balancer

Application Load Balancer:

Operates at 7 (Application layer of OSI model)

It is designed to route HTTP and HTTPS traffic based on content (path-based routing & host-based routing with the help of HTTP headers)

Ideal for modern web applications, Microservices and (Container based application)

Request per minute -> maybe 10k / min -> lets say we have 3 servers currently

Say 100,000 RPM -> then we need 30 servers to manage 100,000 requests

If requests increase to 200,000 then we need to increase the servers accordingly

Auto-scaling concept is there to handle variations in servers based on RPM

Auto-scaling -> Fault tolerance (if some instance is unhealthy, it will automatically launch to replace it),
Cost management, Availability