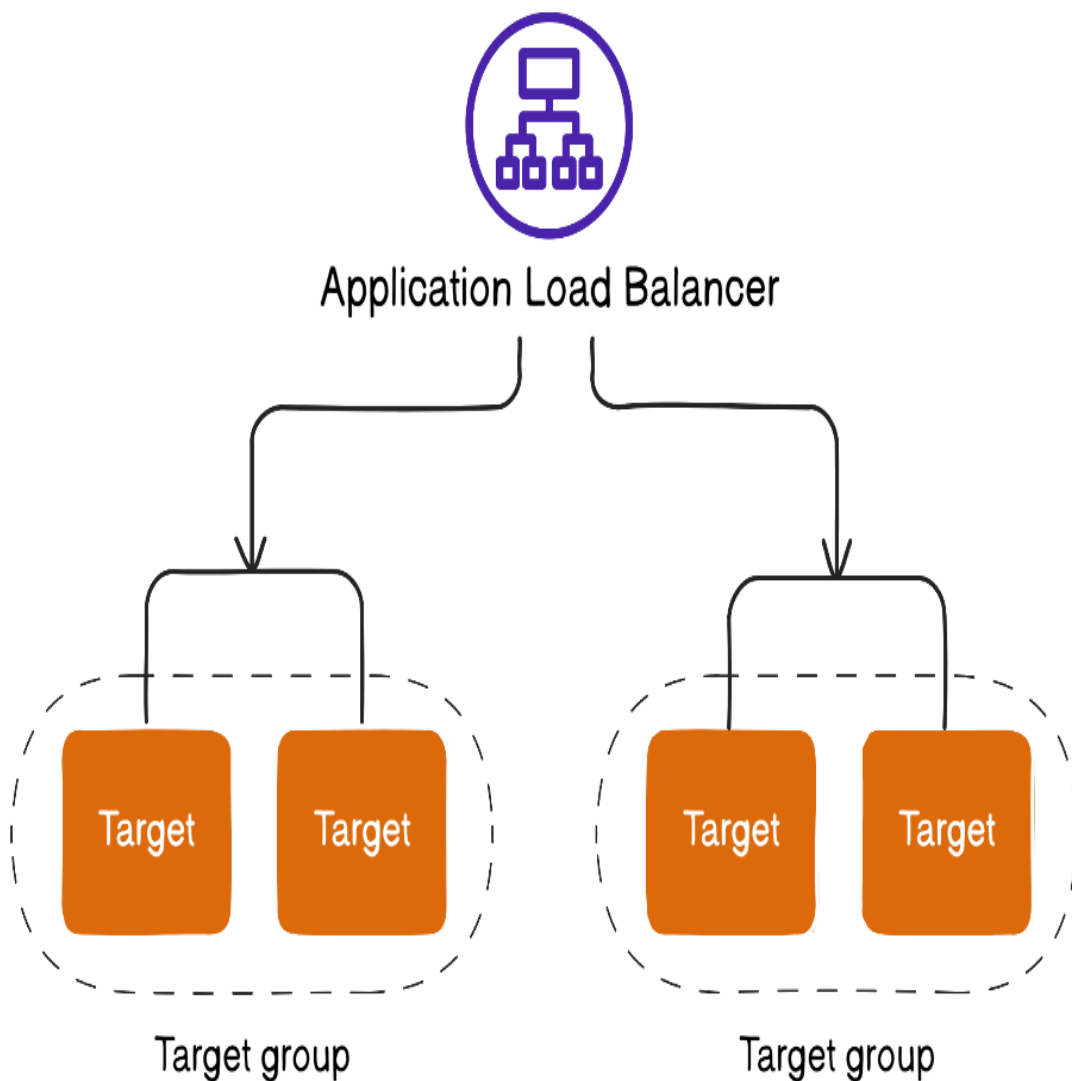


# Guided Lab: Creating Your First Application Load Balancer

## Description

Application Load Balancer (ALB) operates at the application layer (Layer 7 of the OSI model) and is designed to route HTTP/HTTPS traffic. But why use an ALB? While one can host an application on a single EC2 instance or vertically scale an instance for more resources, there are limits.



ALB is usually used when:

1. High availability is required: ALB can route traffic across multiple Availability Zones (AZs). If an instance in one AZ fails, ALB redirects traffic to healthy instances in other AZs.
2. Using Auto Scaling Groups (ASG) for dynamic scaling: ALB integrates seamlessly with EC2 Auto Scaling. Together, they help you create a multi-AZ environment that lets you scale out when the need arises. Instead of scaling a single instance vertically, you can simply add more instances when you need them, avoiding the fuss of resizing an EC2 instance.
3. HTTP-based routing is desired: ALB supports various request routing based on HTTP parameters. This enables use cases like:
  1. **Path-based routing:** Directs client requests to specific services based on the URL path. For instance, /images could route to an image server while /api goes to an API server.
  2. **Host-based routing:** Routes requests based on the domain name. Useful for hosting multiple domains on a single load balancer.
  3. **HTTP header routing:** Routes traffic based on headers, query parameters, or HTTP methods.

## Prerequisite

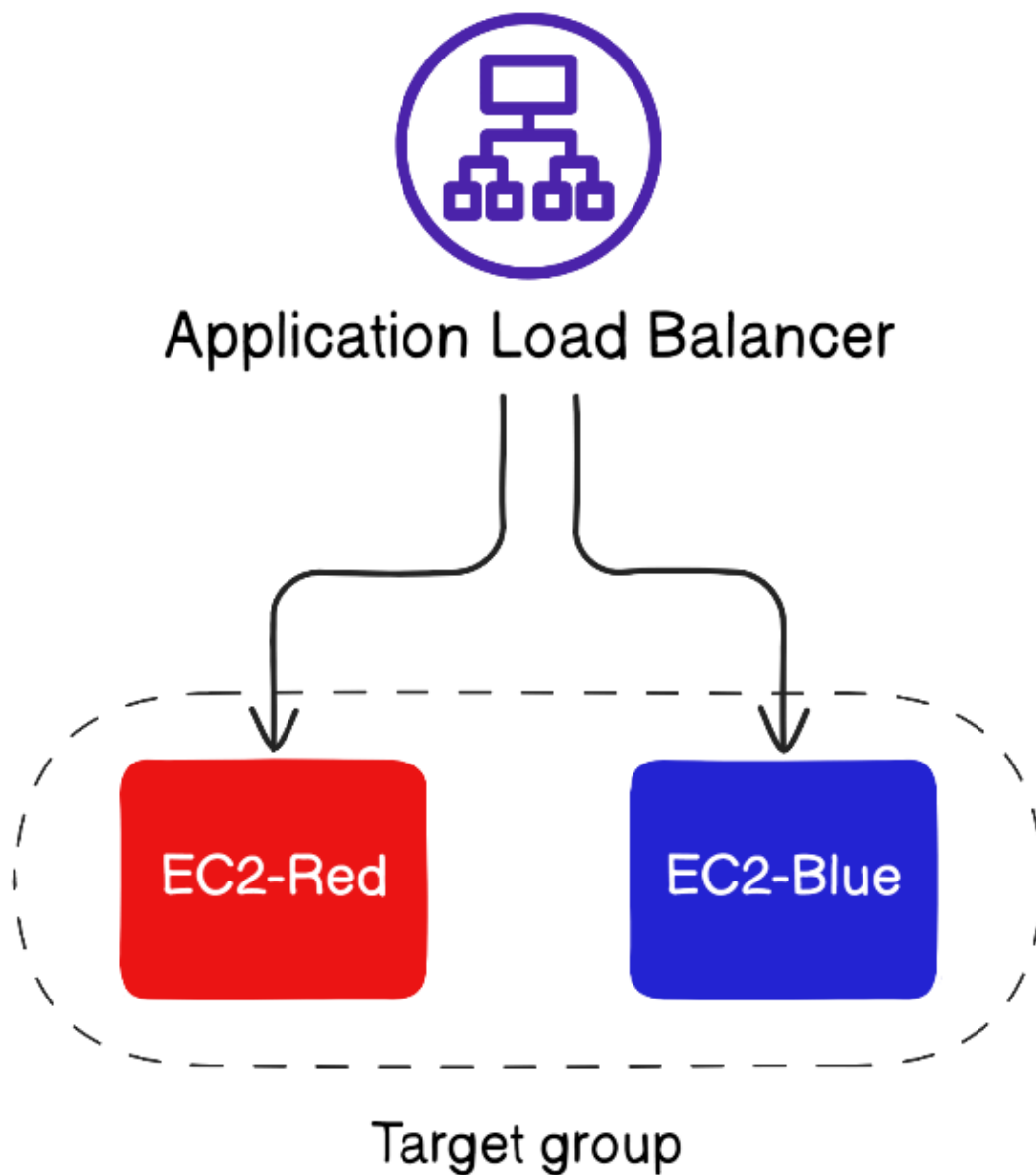
This lab assumes you have experience creating EC2 instances and are familiar with its basic components.

If you find any gaps in your knowledge, consider taking the following labs:

- Creating an Amazon EC2 instance (Linux)
- Setting up a Web server on an EC2 instance
- Launching an EC2 Instance with User Data

## Objectives

In this lab, we'll set up two EC2 instances to demonstrate how an ALB distributes traffic visually. One will display a red web page, and the other a blue one. As you access the ALB, the page colors will switch, representing the load distribution between instances.

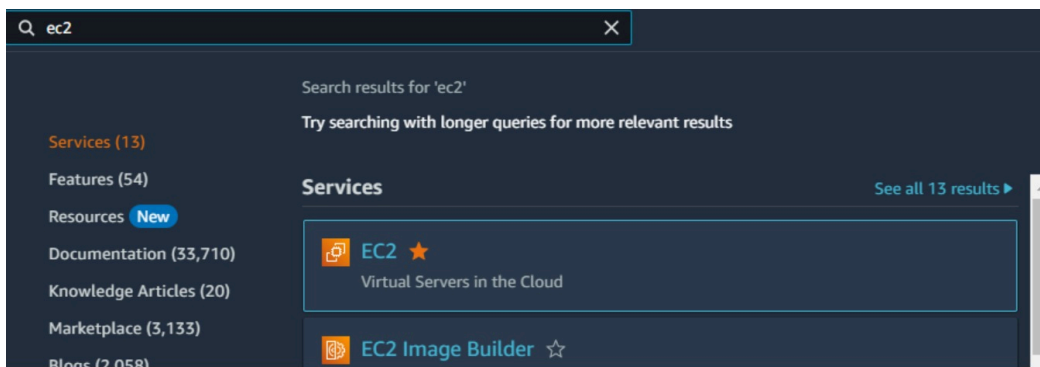


Subscribe to access AWS  
PlayCloud Labs

## Lab Steps

### Creating the EC2 instances

1. Search 'ec2' in the AWS Management Console search bar. Click **EC2** on the search results.



2. On the left window, under **Network & Security**, select **Security Groups**, then click **Create security group**. Create a security group named 'ALB-SG' for the Application Load Balancer. Configure it with an inbound rule allowing HTTP traffic from 0.0.0.0/0.

3. Create another security group named 'SERVER-SG' for the EC2 instances. Configure it with an inbound rule allowing HTTP traffic from the Application Load Balancer's security group (ALB-SG).

By setting up these security groups, we make sure that the application cannot be accessed directly using the EC2 instances' public IP addresses. All incoming traffic is routed exclusively through the ALB, ensuring users interact with our application via the ALB's endpoint only.

3. Launch two EC2 instances with the following configurations.

## EC2-RED

- Name: EC2-RED
- Instance type: t2.micro
- AMI: Amazon Linux
- Key pair: We're not going to SSH into any instances in this lab, so just select the '**Proceed without key pair**' option).
- Security Group: SERVER-SG
- User data:

```
#!/bin/bash
sudo yum update -y
sudo yum install nginx -y
sudo service nginx start
echo '<html><body style="background-color:red;"><h1>EC2
RED server</h1></body></html>' | sudo tee
/usr/share/nginx/html/index.html > /dev/null
sudo service nginx reload
```

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

 Choose file

```
#!/bin/bash
sudo yum update -y
sudo yum install nginx -y
sudo service nginx start
echo '<html><body style="background-color:red;"><h1>EC2 RED server</h1>
</body></html>' | sudo tee /usr/share/nginx/html/index.html > /dev/null
sudo service nginx reload
```

☐ User data has already been base64 encoded

## EC2-BLUE

- Name: EC2-BLUE
- Instance type: t2.micro
- AMI: Amazon Linux
- Key pair: We're not going to SSH into any instances in this lab, so just select the '**Proceed without key pair**' option).
- Security Group: SERVER-SG
- User data:

```
#!/bin/bash
sudo yum update -y
sudo yum install nginx -y
sudo service nginx start
echo '<html><body style="background-color:blue;">
<h1>EC2 BLUE server</h1></body></html>' | sudo tee
/usr/share/nginx/html/index.html > /dev/null
sudo service nginx reload
```

#### User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

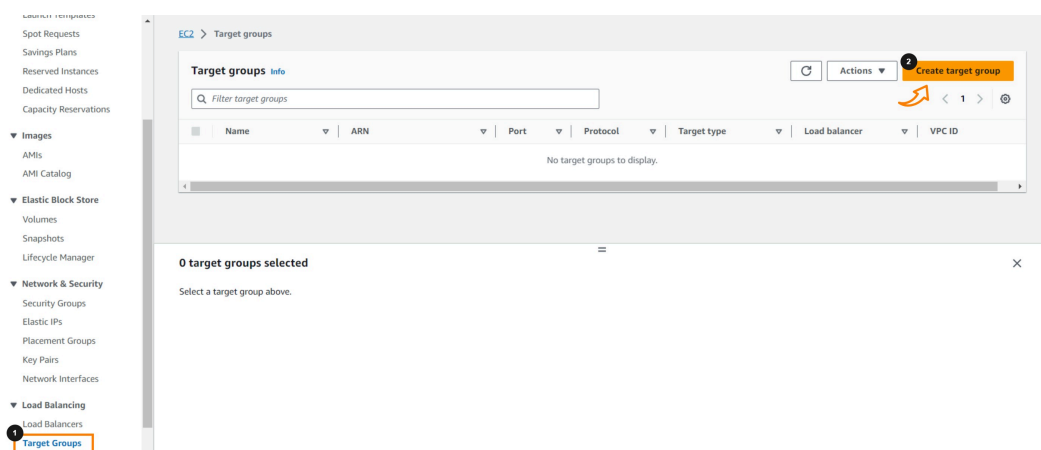
 Choose file

```
#!/bin/bash
sudo yum update -y
sudo yum install nginx -y
sudo service nginx start
echo '<html><body style="background-color:blue;"><h1>EC2 BLUE server</h1>
</body></html>' | sudo tee /usr/share/nginx/html/index.html > /dev/null
sudo service nginx reload
```

☐ User data has already been base64 encoded

## Creating the Target Group

4. On the left side of the EC2 Management Console, Under **Load Balancing**, select **Target Groups** then click **Create target group**



## 5. For **target type**, choose **Instances**.

### 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.

☐ **Application Load Balancer**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

## 6. Enter **my-alb-target-group** as the **Target group name**.

Target group name

my-alb-target-group

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

## 7. For **Protocol**, select **HTTP** and enter **80** for the **Port** number. Make sure the **Ipv4** option is selected **IP address type**.

Protocol : Port

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](#)

## 8. For **VPC**, select the default VPC. Select **HTTP1** for the **Protocol version**.

#### 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 list.

-

vpc-080e9d01d1e9f8245  
IPv4: 192.168.5.0/26

▼

#### 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.

☐ gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

9. In the **Health checks** section, choose HTTP and leave the default '/' path as the Health check path.

### 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.

► Advanced health check settings

Health checks allow the ALB to ping or check your servers to see if they're okay. Think of it like a regular wellness check. It does this by trying to access a specific path, like '/', on your server. If it gets a response, it knows the server is good. If not, it thinks the server is unhealthy and stops sending it traffic. Changing the health check path to something that doesn't respond correctly can mistakenly mark healthy targets as unhealthy.

10. At the bottom of the page, click **Next**.

► **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

11. Select the **EC2-RED** and **EC2-BLUE** instances as targets, then click the **Include as pending below** button.



**Available instances (2/2)**

Filter instances

Instance ID	Name	State	Security groups	Zone	Private IP address
i-0e77474c494a80224	EC2-BLUE	Running	SERVER-SG	us-east-1b	192.168.5.27
i-0ee9324631bf48a97	EC2-RED	Running	SERVER-SG	us-east-1b	192.168.5.22

2 selected

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

80

1-65535 (separate multiple ports with commas)

Include as pending below

12. Click the **Create target group** button.

**Review targets**

Targets (2)

Filter targets

Show only pending

Remove all pending

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID
X	Pending	i-0e77474c494a80224	EC2-BLUE	80	Running	SERVER-SG	us-east-1b	192.168.5.27	subnet-08
X	Pending	i-0ee9324631bf48a97	EC2-RED	80	Running	SERVER-SG	us-east-1b	192.168.5.22	subnet-08

2 pending

Cancel Previous **Create target group**

## Creating the Application Load Balancer

13. Under **Load Balancing**, select **Load Balancers**, then click **Create Load Balancer**.

Reserved Instances  
Dedicated Hosts  
Capacity Reservations

▼ Images  
AMIs  
AMI Catalog

▼ Elastic Block Store  
Volumes  
Snapshots  
Lifecycle Manager

▼ Network & Security  
Security Groups  
Elastic IPs  
Placement Groups  
Key Pairs  
Network Interfaces

▼ Load Balancing  
**Load Balancers**  
Target Groups

▼ Auto Scaling  
Auto Scaling Groups

EC2 > Load balancers

**Load balancers**

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

Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
No resources to display						

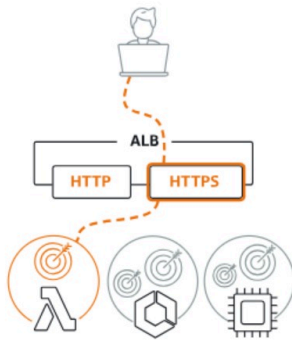
0 load balancers selected

Select a load balancer above.

Create load balancer

14. Click the **Create** button under **Application Load Balancer**.

### Application Load Balancer [Info](#)

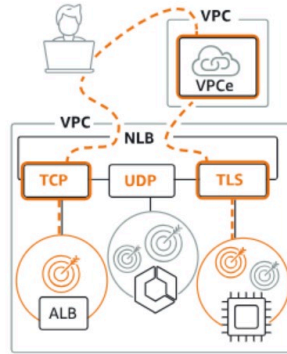


Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create



### 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 second securely while maintaining ultra-low latencies.

Create

### 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

## 15. On **Basic configuration**:

- Enter '*my-alb*' as the load balancer name.
- Select **Internet-facing** for the Scheme.
- Select **IPv4** as the IP address type.

### Basic configuration

#### Load balancer name

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

my-alb

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.

## 16. For **Network mapping**, select the default VPC. For **Mappings**, select all the checkboxes.

Each AZ you select is where the ALB will deploy a node to handle incoming traffic. These nodes in different AZs provide redundancy and ensure high availability for your application. Even though our

main goal in this lab isn't high availability and our EC2 instances won't be deployed across different AZs, please check all the available AZ checkboxes. Doing so mirrors a setup you'd use for a multi-AZ architecture.

#### 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-080e9d01d1e9f8245  
IPv4: 192.168.5.0/26

↻

#### 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-az2)

Subnet

subnet-0658d5a6c6531580b

IPv4 address

Assigned by AWS

##### ☒ us-east-1b (use1-az4)

Subnet

subnet-080e4a793c23366be

IPv4 address

Assigned by AWS

##### ☒ us-east-1c (use1-az6)

Subnet

subnet-06198423051f6c3b6

IPv4 address

Assigned by AWS

17. For **Security groups**, click the dropdown menu and select **ALB-SG**

#### 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

↻

ALB-SG  
sg-0efd90211c4070958 VPC: vpc-080e9d01d1e9f8245

✕

18. On **Listeners and routing**:

a. Select HTTP for Protocol

b. Set 80 for Port

c. Click the **Select a target group** dropdown menu, and click **my-alb-target-group**.

d. Scroll down to the bottom page and click **Create load balancer**.

### 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

Port

Default action

[Info](#)

HTTP

:

80

Forward to

my-alb-target-group

HTTP

1-65535

Target type: Instance, IPv4

↻

[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

19. After you've created the load balancer, a confirmation will appear at the top of the page. Click on **View Load Balancer** to navigate to your ALB.

On the ALB dashboard, you'll see the status of your newly created ALB. Ensure that its state changes to **Active** before proceeding.

Load balancers (1)						
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.						
<input type="text" value="Filter load balancers"/>						
<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	
<input type="checkbox"/>	<a href="#">my-alb</a>	my-alb-412856487.us-eas...	Active	vpc-080e9d01d1e9f82...	<a href="#">3 Availability Zones</a>	

20. Once active, copy your ALB's DNS Name. This is the endpoint through which the ALB will route traffic to your instances.

Load balancers (1)						
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.						
<input type="text" value="Filter load balancers"/>						
<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	
<input type="checkbox"/>	<a href="#">my-alb</a>	my-alb-412856487.us-eas...	Active	vpc-080e9d01d1e9f82...	<a href="#">3 Availability Zones</a>	

21. Enter your ALB's DNS Name into your browser. You should see a color-coded page (red or blue). Hit refresh several times to see the load balancing in action (switching between colors). This demonstrates the ALB distributing traffic across your instances. Note that due to browser caching and the nature of load balancing, the switch might not always be immediate.

## EC2 RED server

Notice how the responses consistently alternate between red and blue? This is because the default routing mechanism for ALB is 'round-robin'. When there are two servers, as we have now, each load balancer node receives approximately 50% of the traffic. The round-robin algorithm ensures incoming traffic is distributed evenly between the servers, cycling from one to the next in order.

You've successfully set up an Application Load Balancer and observed how it distributes incoming traffic across two instances. This foundational knowledge of ALB is crucial in scaling web applications. As you delve deeper into AWS and its services, you'll appreciate the flexibility and robustness that tools like the ALB offer. Well done on completing this lab!

