

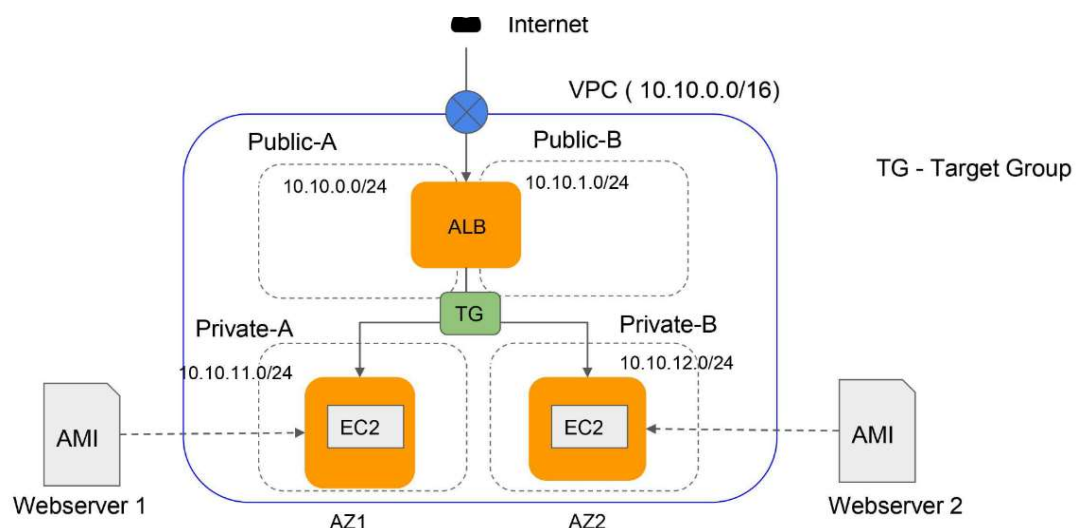
Application Load Balancer

The load balancer distributes incoming application traffic across multiple target groups, such as EC2 instances, in multiple Availability Zones. This increases the scalability and availability of the application.

By distributing network traffic and information flows across multiple servers, a load balancer ensures no single server bears too much demand. This improves application responsiveness and availability, enhances user experiences, and can protect from distributed denial-of-service ([DDoS](#)) attacks.

ALBs have three components – listeners, load balancer, and the target group. After receiving a request, the **load balancer** evaluates the **listener rules** in priority order (to choose which rule to execute). It then selects a target from the **target group** for the rule action.

Based on the below diagram , we are implementing the Application Load Balancer.



implementation

- 1.Create VPC , 2 public subnets and 2 private subnets
- 2.Launch 2 EC2 instances in 2 private subnets and select the pre-configured AMIs
- 3.Create EC2 target group and attach the above 2 ec2 instances into the target group.

4. Create Application load balancer , use 2 public subnets , configure the listener at port 80, create a security group and allow traffic from port 80 anywhere , attach to target group.
5. Take the load balance DNS name and access over the browser and refresh the page and see the traffic is passing to each servers simultaneously
6. Go to target groups and add stickiness to implement the traffic passage to one server only in the specified time.

Creating VPC

Goto AWS console search option and type VPC.

Then click on create VPC. Fill the below details and click on create VPC.

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources:

☒ VPC only

☐ VPC and more

Name tag - optional

Creates a tag with a key of 'Name' and a value that you specify.

ALB-VPC

IPv4 CIDR block [Info](#)

☒ IPv4 CIDR manual input

☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.10.0.0/16

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ IPAM-allocated IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

☐ IPv6 CIDR owned by me

Tenancy [Info](#)

Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Q Name



Value - optional

Q ALB-VPC



Remove tag

Add tag

You can add 49 more tags

Cancel

Create VPC

Create Internet Gateway

Goto Internet Gateway from LHS menu and click on create internet gateway

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="albigw"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

Then go to Actions from Internet Gateway home page and click on Attach VPC
Select the VPC created and click on Attach internet gateway above as below

VPC > Internet gateways > Attach to VPC (igw-07bece89db44b1b6b)

Attach to VPC (igw-07bece89db44b1b6b) [Info](#)

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

► AWS Command Line Interface command

Go to subnets from LHS , as per the diagram we need 4 subnets.

So attach the VPC into the subnet

VPC > Subnets > Create subnet

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.
vpc-0a9e21ad923b3743e (ALB-VPC) ▼

Associated VPC CIDRs

IPv4 CIDRs
10.10.0.0/16

And create 4 subnets as per below images: Give subnet name, select an AZ and give the IP , then click on create subnet.

Subnet 1 of 4

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

▼ Tags - optional

Key

Value - optional

You can add 49 more tags.

Subnet 2 of 4

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

▼ Tags - optional

Key

Value - optional

You can add 49 more tags.

Subnet 3 of 4

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

▼ Tags - optional

Key

Value - optional



Remove

Add new tag

You can add 49 more tags.

Remove

Subnet 4 of 4

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

▼ Tags - optional

Key

Value - optional



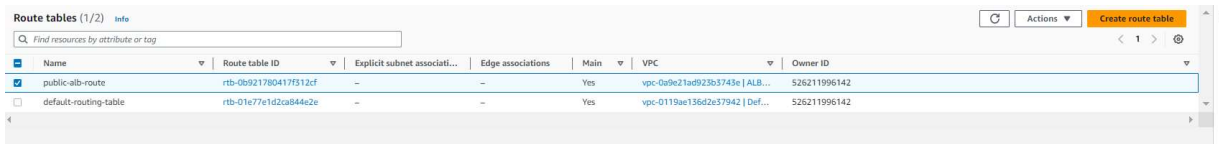
Remove

Add new tag

You can add 49 more tags.

Remove

Goto Routing Tables from LHS and we can see a default and one unnamed routing table. The unnamed routing table we can use either for public or private. So here I'm going to name the unnamed routing table as public.

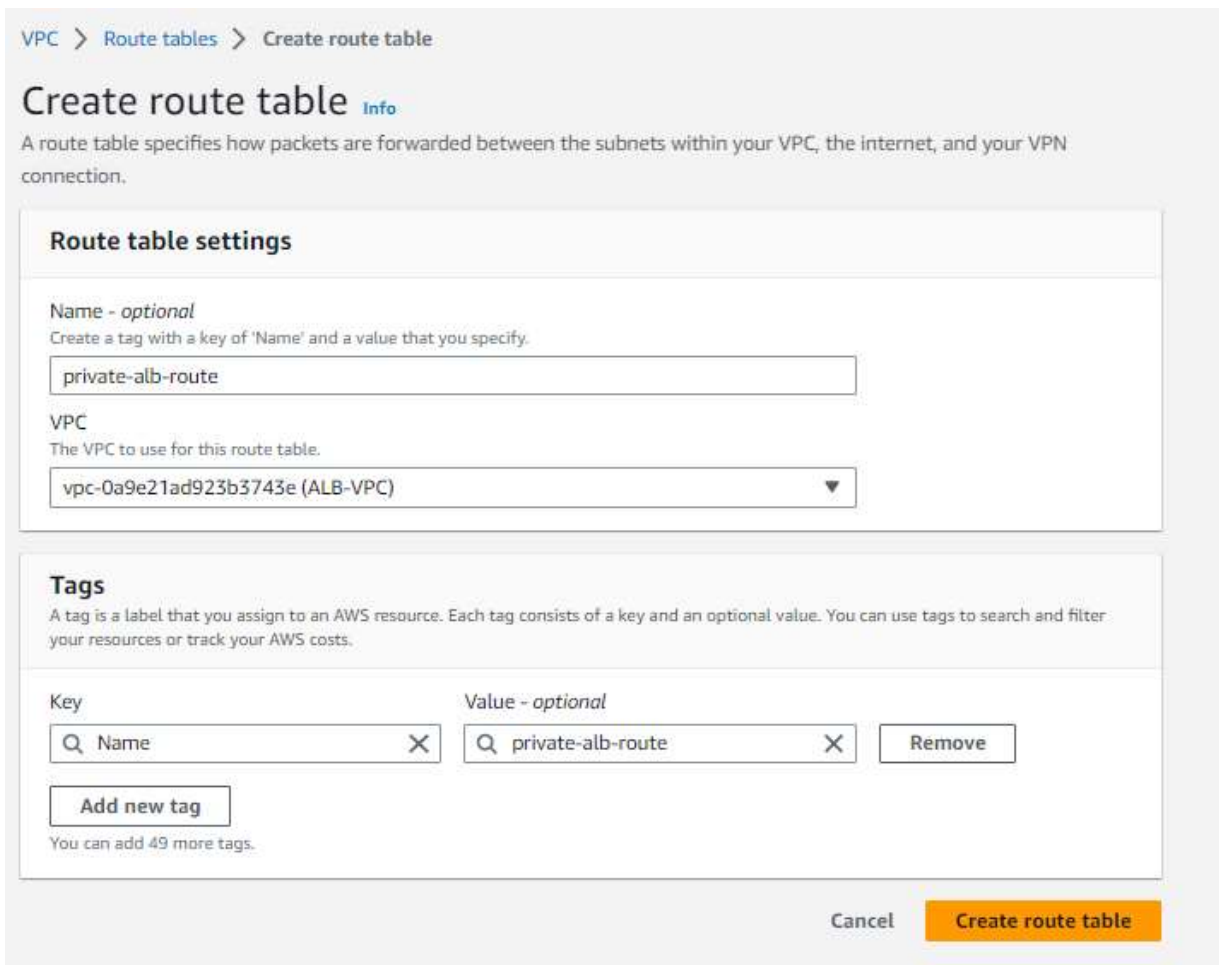


Route tables (1/2) Info

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC	Owner ID
public-alb-route	rtb-0b921780417f512cf	-	-	Yes	vpc-0a9e21ad923b3743e ALB...	526211996142
default-routing-table	rtb-01e77e1d2ca8462e	-	-	Yes	vpc-0119ae136d2e37942 Def...	526211996142

Now go to create route table and give inputs as below



VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

private-alb-route

VPC
The VPC to use for this route table.

vpc-0a9e21ad923b3743e (ALB-VPC)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

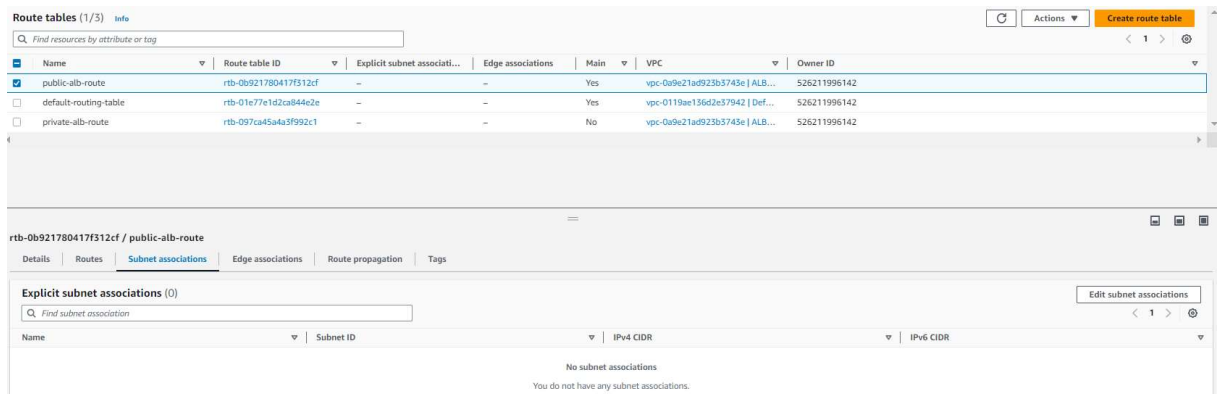
Key	Value - optional	
Name	private-alb-route	Remove

Add new tag

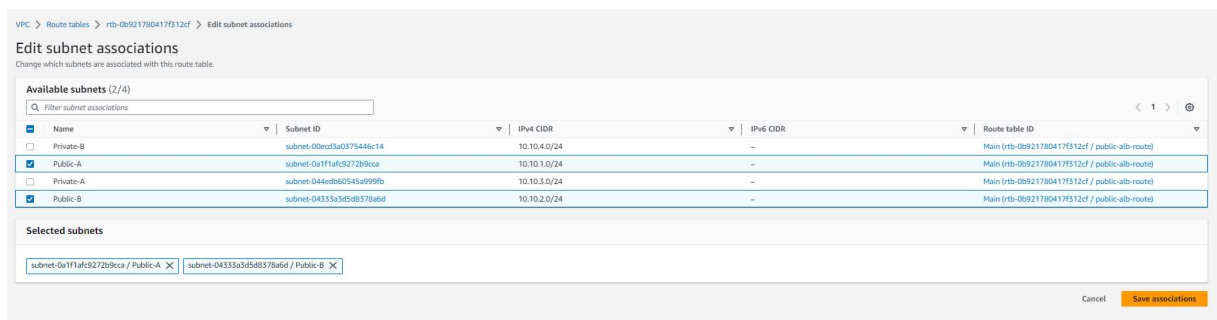
You can add 49 more tags.

Cancel Create route table

Go to subnet allocations and click on edit subnet allocations

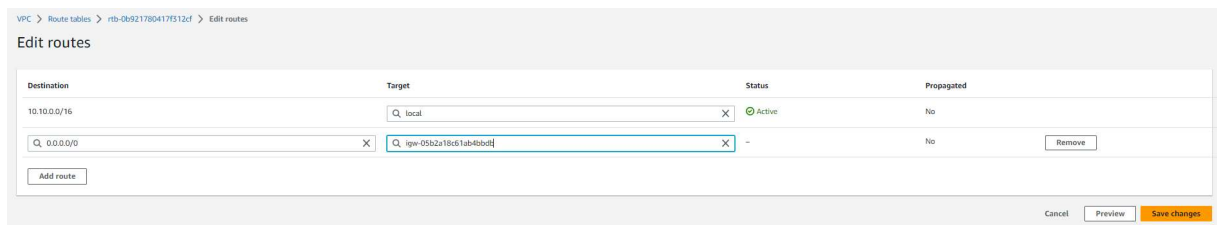


Select both public subnets and click on save associations



Sameway do for private-alb-route as well by selecting the private subnets.

Now click on public-alb-route and click on edit routes , add as below and click on save changes.



Now all done from VPC end.

Go to AWS EC2 console , we need to launch 2 instances as per the diagram.

Launch instance with all default settings except in Network Settings → Select existing security group which had port 22 and 80 open.

Please refer this link to know how to configure this EC2 - <https://Inkd.in/g-P4ePXT> and in Advanced details add the below commands and launch instance.

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

```
yum install httpd -y
```

```
service httpd start
```

```
chkconfig httpd on
```

```
echo " Ohio Server1 - Try fail smash it" > /var/www/html/index.html
```

Once the instance is 2/2 checks passed, then follow like below and click on create image

Instances (1/1) [Info](#)

Find instance by attribute or tag (case-sensitive)

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm
<input checked="" type="checkbox"/>	<div><div>-</div><div></div></div>		<div><div>Running</div></div>	<div><div>t2.micro</div></div>	<div><div>2/2 checks passed</div></div>	<div><div>No al</div></div>

Launch instances

Launch instance from template

Migrate a server

Connect

Stop instance

Start instance

Reboot instance

Hibernate instance

Terminate instance

Instance settings

Networking

Security

Image and templates

Monitor and troubleshoot

Create image

Create template from instance

Launch more like this

Give an image name like below

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
i-09870890fd1445a31

Image name
webserver1
Maximum 127 characters. Can't be modified after creation.

Image description - optional
Image description
Maximum 255 characters

No reboot
☐ Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	8	EBS General Purpose S...	3000		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately
Tag the image and the snapshots with different tags.

No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Create image](#)


Terminate the instance created and go to LHS AMIs and find the AMI which we created just now. Same way create another AMI image also and named and webserver2.

Now we have our own AMI, create instance with the our own AMI and select the VPC which we have created and select the private-A subnet.
All other settings are default.

▼ **Network settings** [Info](#)

VPC - *required* [Info](#)


vpc-0a9e21ad923b3743e (ALB-VPC)
10.10.0.0/16



Subnet [Info](#)

subnet-044edb60545a999fb
Private-A

VPC: vpc-0a9e21ad923b3743e Owner: 526211996142
Availability Zone: us-east-2a IP addresses available: 251 CIDR: 10.10.3.0/24


[Create new subnet](#)

Auto-assign public IP [Info](#)

Disable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group
☐ Select existing security group

Security group name - *required*

alb-servers

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and `_-./()#/@[]+=&:{}!$*`

Description - *required* [Info](#)

alb-servers

And in Rules , add ssh and http with custom IP as 10.10.0.0/16 .All other settings are default.

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 10.10.0.0/16) Remove

Type Info	Protocol Info	Port range Info
ssh ▼	TCP	22
Source type Info	Source Info	Description - optional Info
Custom ▼	<input type="text" value="Add CIDR, prefix list or security"/> 10.10.0.0/16 ✕	e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 10.10.0.0/16) Remove

Type Info	Protocol Info	Port range Info
HTTP ▼	TCP	80
Source type Info	Source Info	Description - optional Info
Custom ▼	<input type="text" value="Add CIDR, prefix list or security"/> 10.10.0.0/16 ✕	e.g. SSH for admin desktop

Add security group rule

► Advanced network configuration

Create second instance as per the diagram and select the AMI name as Webserver2 ,select the VPC and subnet as private-B. Select the security group we have just created above alb-servers and under advance details paste the same command as below and launch instance with all default settings.

```
#!/bin/bash
yum install httpd -y
service httpd start
chkconfig httpd on
echo " Ohio Server2 - Try fail smash it" > /var/www/html/index.html
```

▼ Network settings Info

VPC - required Info

vpc-0a9e21ad923b3743e (ALB-VPC)
10.10.0.0/16

Subnet Info

subnet-00ecd3a0375446c14 Private-B
VPC: vpc-0a9e21ad923b3743e Owner: 526211996142
Availability Zone: us-east-2b IP addresses available: 251 CIDR: 10.10.4.0/24

Create new subnet

Auto-assign public IP Info

Disable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups Info

Select security groups

alb-servers sg-0acfc226c85b947aa X
VPC: vpc-0a9e21ad923b3743e

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

Creating Target Group

Go to LHS and select Target Groups and click on Create target group. Give a name and select the VPC which we have created above.

Create target group

×

Your load balancer routes requests to the targets in a target group using the target group settings that you specify, and performs health checks on the targets using the health check settings that you specify.

Target group name

alb-webserver

Target type

☒ Instance
 ☐ IP
 ☐ Lambda function

Protocol

HTTP

Port

80

VPC

vpc-0a9e21ad923b3743e (10.10.0.0/16) | A

Health check settings

Protocol

HTTP

Path

/

▶ Advanced health check settings

Cancel

Create

Once created → go to Targets , click on edit and select the instances and below and click on Add to registered and click on save.

Register and deregister targets

×

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Remove

<input type="checkbox"/>	Instance	Name	Port	State	Security groups	Zone
No instances available.						

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered

on port 80

Search Instances

×

<input checked="" type="checkbox"/>	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-08c9d179ef168b76c	webserver1	running	alb-servers	us-east-2a	subnet-044edb60545a999fb	10.10.3.0/24
<input checked="" type="checkbox"/>	i-08c96c78d9f3b9a28	webserver2	running	alb-servers	us-east-2b	subnet-00ecd3a0375446c14	10.10.4.0/24

Cancel

Save

Now the target group has the instances which we have created.

Go to Load balancer → Create load balancer → Application load balancer

Give name to the load balancer and under Availability Zones

Select the newly created VPC and select the 2 AZs and select the subnet as PublicA and PublicB as the traffic first needs to go to public IPs and then private IPs as per the diagram.

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name ⁽ⁱ⁾

Scheme ⁽ⁱ⁾ ☒ internet-facing
☐ internal

IP address type ⁽ⁱ⁾

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
<input type="text" value="HTTP"/>	<input type="text" value="80"/>

[Add listener](#)

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC ⁽ⁱ⁾

Availability Zones

- ☒ us-east-2a IPv4 address ⁽ⁱ⁾ Assigned by AWS
- ☒ us-east-2b IPv4 address ⁽ⁱ⁾ Assigned by AWS

Add-on services

Additional AWS services can be integrated with this load balancer at launch when you enable them below. You can also add these and other services after your load balancer is created by reviewing the "Integrated Services" tab for the selected load balancer.

AWS Global Accelerator ☐ Create an accelerator to get static IP addresses and improve the performance and availability of your application. [Learn more](#)
Additional resources: [aws.amazon.com/globalaccelerator](#)

[Cancel](#) [Next: Configure Security Settings](#)

Create new load balancer with below functionalities

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group ☒ Create a new security group
☐ Select an existing security group

Security group name

Description

Type ⁽ⁱ⁾	Protocol ⁽ⁱ⁾	Port Range ⁽ⁱ⁾	Source ⁽ⁱ⁾
<input type="text" value="SSH"/>	<input type="text" value="TCP"/>	<input type="text" value="22"/>	<input type="text" value="Custom 10.10.0.0/16"/>
<input type="text" value="HTTP"/>	<input type="text" value="TCP"/>	<input type="text" value="80"/>	<input type="text" value="Custom 10.10.0.0/16"/>

[Add Rule](#)

Select the existing target group and create the load balancer

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks on the targets using these settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer. You can edit or add listeners after the load balancer is created.

Target group

Target group ⁽ⁱ⁾

Name ⁽ⁱ⁾

Target type

- ☒ Instance
- ☐ IP
- ☐ Lambda function

Protocol ⁽ⁱ⁾

Port ⁽ⁱ⁾

Protocol version ⁽ⁱ⁾ ☒ 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.

Health checks

Protocol ⁽ⁱ⁾

Path ⁽ⁱ⁾

[Advanced health check settings](#)

Now select the alb-lb1 load balancer and under Security Groups click on the link of the security group which will open a new window.

Go to inbound rules and click on edit inbound rules and remove the rules already added and select anywhere IPv4 from source and click on save rules.

EC2 > Security Groups > sg-0561d1515f21630x0 - alb-lb-SG > Edit inbound rules

Edit inbound rules info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>info</small>	Protocol <small>info</small>	Port range <small>info</small>	Source <small>info</small>	Description - optional <small>info</small>	
sg-003d40372cefd326f	SSH	TCP	22	Anywhere-IPv4		Delete
sg-01f078041a217de51	HTTP	TCP	80	Anywhere-IPv4		Delete

[Add rule](#)

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

Take the DNS name from the load balancer and try to access it in a window. So we will get the output as below



Ohio Server1 - Try fail smash it

If we refresh the browser , we will get the output of 2 instances.

But what if we want to pass the traffic only to webserver1

Then go to target groups and under Attributes , click on Edit attributes

Select it Enable and give 5 seconds as below

Edit attributes ✕

Deregistration delay info seconds
Specify a value from 0-3600.

Slow start duration info seconds
Specify a value from 30-900 or 0 to disable.

Load balancing algorithm info ☒ Round robin ☐ Least outstanding requests

Stickiness info ☒ Enable

Stickiness duration seconds Specify a value between 1 second and 7 days.

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

Now go to the browser and refresh the page, we will get the server1 page for 5-6 seconds , we will get the server2 page once the cookie expires.

