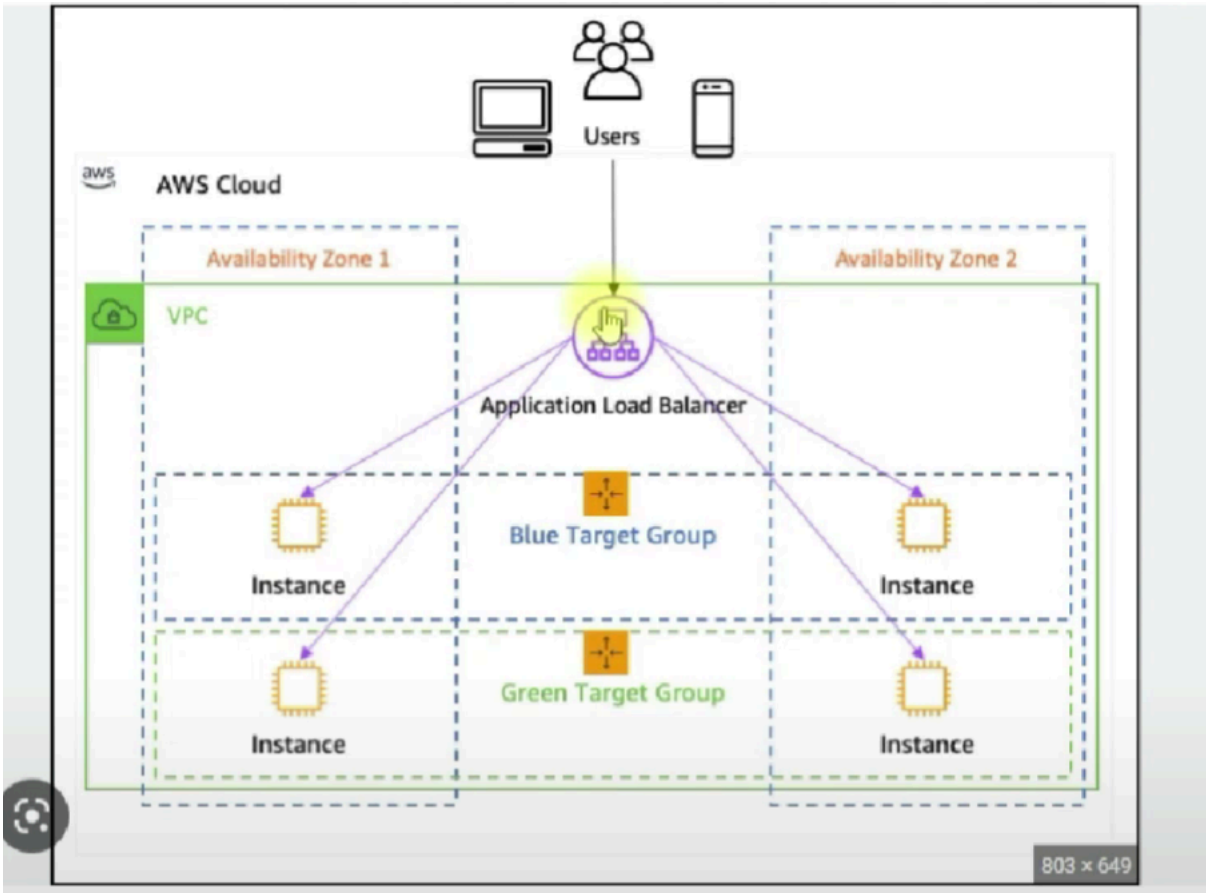


Load Balancer



Instances (1/3) [Info](#) Last updated less than a minute ago [Refresh](#) [Connect](#) [Instance state](#) [Actions](#)

[All states](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	instance3	i-06d0dea0e7ae4a2bf	Running	t3.micro	3/3 checks passed	View alarms	ap-south-1c
<input type="checkbox"/>	instance1	i-0ca15d1984d143c5b	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a
<input checked="" type="checkbox"/>	instance2	i-0f067560cfad83e5e	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b

3 Instance Created .

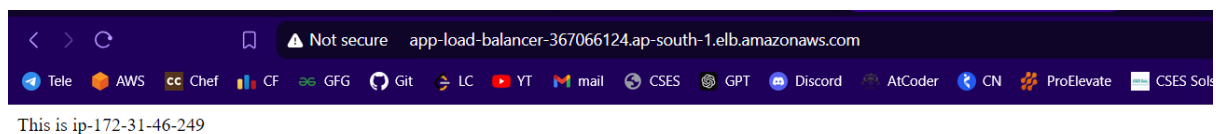
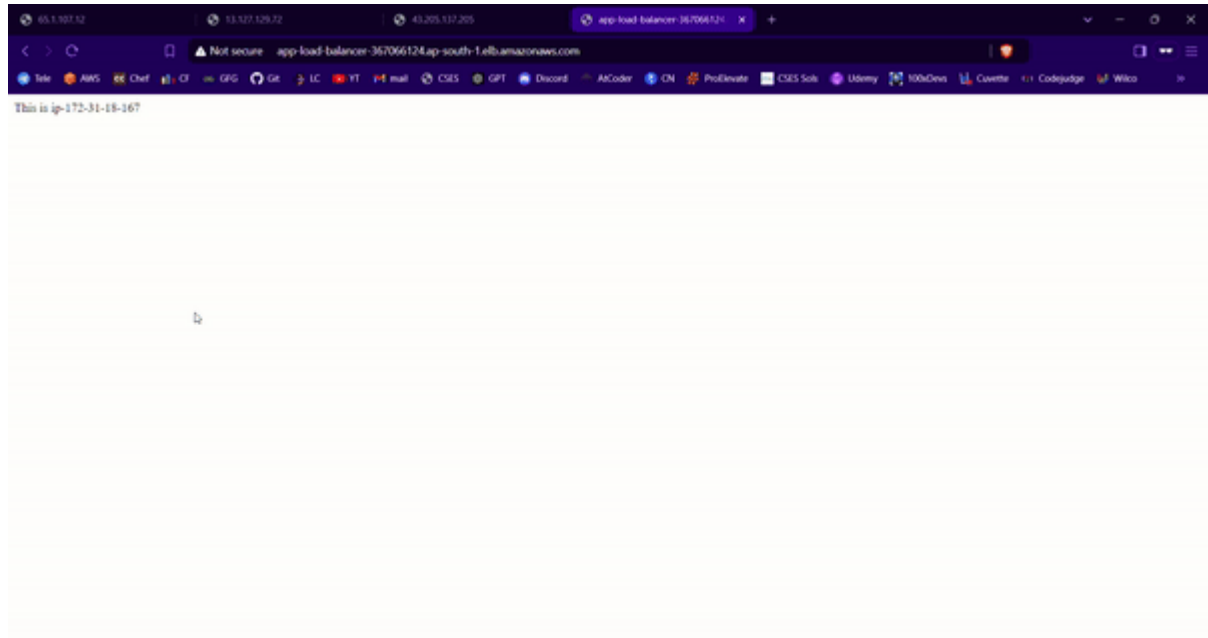
Creating a Security Group.

sg-05e358d4c2e79d279 - load-balancer-security [Actions](#)

Details

Security group name load-balancer-security	Security group ID sg-05e358d4c2e79d279	Description http access	VPC ID vpc-07edff8b878be24f8b
Owner 590183860624	Inbound rules count 0 Permission entries	Outbound rules count 1 Permission entry	

Create a Target Group. Add all the 3 instance.
Create Application Load Balancer.
Select Target Group created Earlier.



Path Base Routing in Application Load Balancer

```
apt-get update
apt-get install nginx -y

mkdir -p /var/www/html/test

echo "this is test server and hostname is $(hostname)" > /var/www/html/test/index.html
```

Path-based routing in an **Application Load Balancer (ALB)** allows you to route traffic to different targets based on the

URL path of the request. This is useful when you want to send traffic to different backend services (target groups) depending on the path in the URL.

Use Cases

- **Microservices:** Route traffic to different services based on the request path (e.g., `/api` goes to one service, `/static` goes to another).
- **Multi-Tenant Applications:** Serve different applications or versions based on the URL path (e.g., `/app1` vs `/app2`).
- **Web Applications:** Send API requests and static content requests to different target groups (e.g., `/api/` to backend servers and `/images/` to a CDN or separate service).






Creating another load balancer ..which on hitting `/api` goes to another instance.

For that lets build two test instances ..which will be connected to another Load balancer.

Created a 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 (2/5)						
<input type="text" value="Filter instances"/>						< 1 > 
	Instance ID	Name	State	Security groups	Zone	
<input checked="" type="checkbox"/>	i-01f1e3fcb1df4bc50	test-instance2	 Running	load-balancer-security	ap-south-1b	
<input checked="" type="checkbox"/>	i-0a56af6526c39420e	test-instance1	 Running	load-balancer-security	ap-south-1b	

Select the target Group.

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

test-for-api

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

Listeners and rules (1/1)

Info

⌂

Manage rules

Add rule

Edit rules

Reprioritize rules

Filter listeners

<input checked="" type="checkbox"/>	Protocol:Port	Default action	Rules	ARN	Security policy
<input checked="" type="checkbox"/>	HTTP:80	<div>Forward to target group<ul style="list-style-type: none">test-for-api: 1 (100%)Target group stickiness: Off</div>	1 rule	ARN	Not applicable

404 Not Found

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

< 1 > ⚙

<input type="checkbox"/>	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate
<input type="checkbox"/>	HTTP:80	<div>Forward to target group<ul style="list-style-type: none">target-group: 1 (100%)Target group stickiness: Off</div>	1 rule	<div>ARN</div>	Not applicable	Not applicable

Rules

Tags

Listener rules (1)

Info

Rule limits

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.

Filter rules

⚙

<input type="checkbox"/>	Name tag	Priority	Conditions (If)	Actions (Then)	ARN	Tags
<input type="checkbox"/>	Default	Last (default)	If no other rule applies	<div>Forward to target group<ul style="list-style-type: none">target-group: 1 (100%)Target group stickiness: Off</div>	<div>ARN</div>	0 tags

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

testapi-rule

Add additional tags

Cancel

Next

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

target-group-api
Target type: Instance, IPv4

HTTP ▼



We

1

0-9

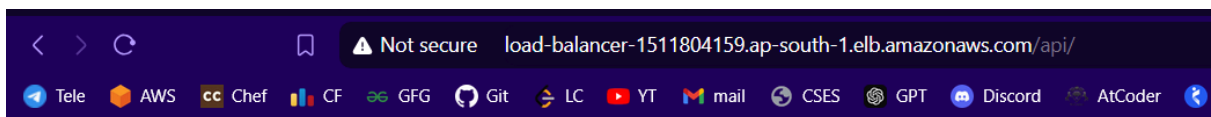
Add target group

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 cl bind a user's session to a specific target, turn on the Target Group attribute Stickiness.

☐ Turn on target group stickiness



this is test server and hostname is ip-172-31-1-177

```
apt-get update  
apt-get install nginx -y
```

```
mkdir -p /var/www/html/test  
echo "this is test server and hostname is  
$(hostname)" > /var/www/html/test/index.html
```

Test locally- curl localhost/test