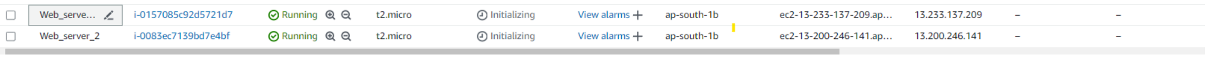
# Assignment 7 : Setting Up Two NGINX Web Servers with an Elastic Load Balancer (ELB) on AWS for Load Distribution

## 1. Launch Two EC2 Instances (Web Servers)

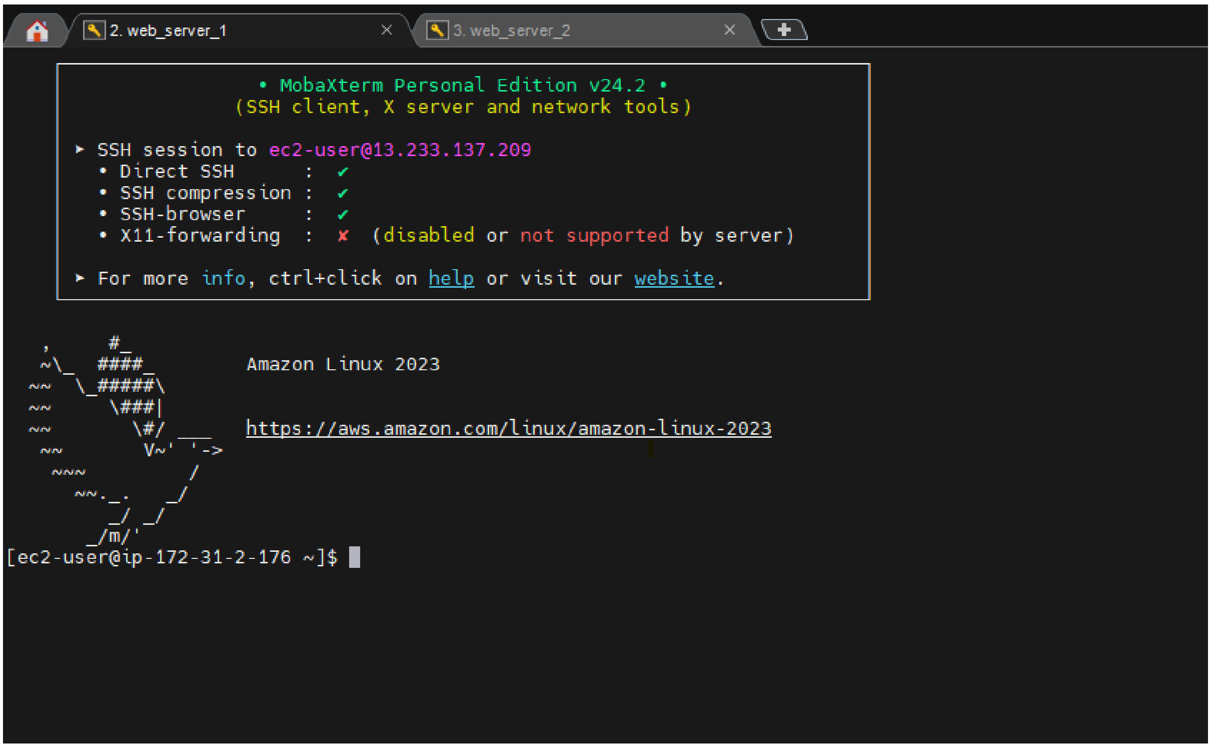
- Step 1: Go to the AWS Management Console, navigate to EC2 and click on Launch Instance.
- Step 2: Choose Amazon Linux 2 AMI as the base image.
- Step 3: Choose an instance type such as t2.micro (eligible for free tier).
- Step 4: Configure Security Groups:
  - Add a rule to allow HTTP (port 80) traffic from Anywhere (0.0.0.0/0).
  - Add a rule to allow SSH (port 22) traffic from Your IP.
- Step 5: Launch the instance and wait until it is running.
- Step 6: Repeat the process to create a second EC2 instance.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | Web_serve... | i-0157085c92d5721d7 | ✓ Running | t2.micro | Initializing | View alarms + | ap-south-1b | ec2-13-233-137-209.ap... | 13.233.137.209 | – | – |
| □ | Web_server_2 | i-0083ec7139bd7e4bf | ✓ Running | t2.micro | Initializing | View alarms + | ap-south-1b | ec2-13-200-246-141.ap... | 13.200.246.141 | – | – |

## 2. Configure NGINX on Both EC2 Instances

Login to EC2 instances via SSH:



```
                  • MobaXterm Personal Edition v24.2 •
                (SSH client, X server and network tools)

  ➤ SSH session to ec2-user@13.233.137.209
    • Direct SSH      :  ✔
    • SSH compression :  ✔
    • SSH-browser     :  ✔
    • X11-forwarding  :  ✘  (disabled or not supported by server)

  ➤ For more info, ctrl+click on help or visit our website.

   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-2-176 ~]$ 
```

**Once connected, follow these commands on both EC2 instances:**

**# Switch to the root user**

```
sudo -i
```

**# Update the instance**

```
yum update -y
```

**# Install NGINX**

```
yum install nginx -y
```

**# Start NGINX**

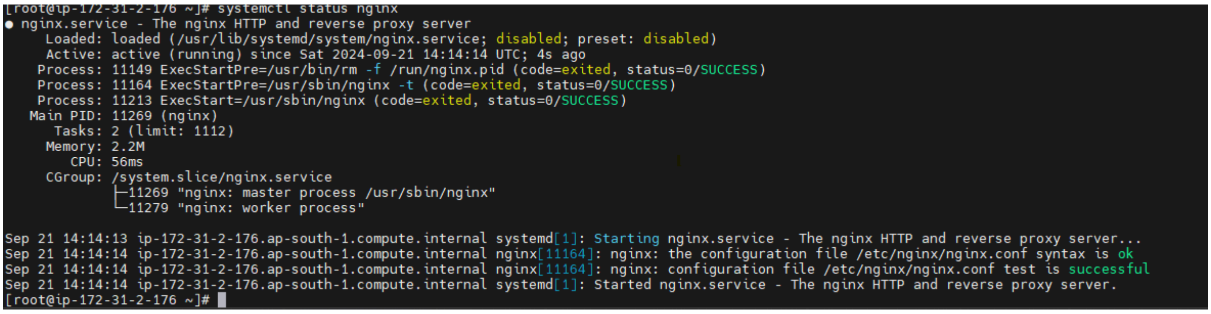
```
systemctl start nginx
```

**# Check NGINX status**

```
systemctl status nginx
```

**# Enable NGINX to start on boot**

```
systemctl enable nginx
```



```
[root@ip-172-31-2-176 ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: active (running) since Sat 2024-09-21 14:14:14 UTC; 4s ago
    Process: 11149 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 11164 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 11213 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 11269 (nginx)
      Tasks: 2 (limit: 1112)
     Memory: 2.2M
        CPU: 56ms
     CGroup: /system.slice/nginx.service
             ├─11269 "nginx: master process /usr/sbin/nginx"
             └─11279 "nginx: worker process"

Sep 21 14:14:13 ip-172-31-2-176.ap-south-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Sep 21 14:14:14 ip-172-31-2-176.ap-south-1.compute.internal nginx[11164]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 21 14:14:14 ip-172-31-2-176.ap-south-1.compute.internal nginx[11164]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 21 14:14:14 ip-172-31-2-176.ap-south-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-172-31-2-176 ~]#
```

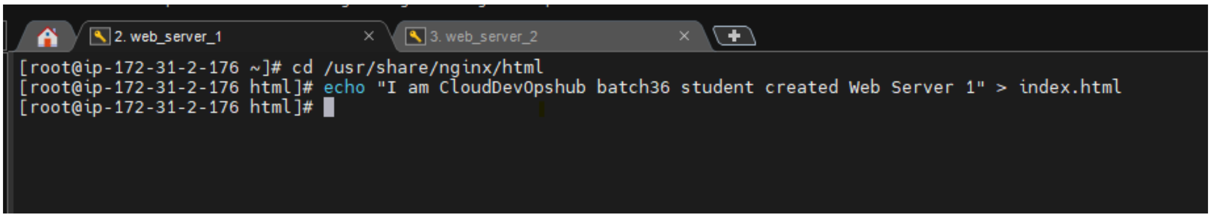
## 3. Configure Web Content for Each Server

Now, add unique content to each web server to differentiate between them.

**On Web Server 1:**

```
# Go to the NGINX HTML directory
cd /usr/share/nginx/html

# Replace the default index.html file with custom content
echo "I am CloudDevOpshub batch36 student created Web Server 1" > index.html
```

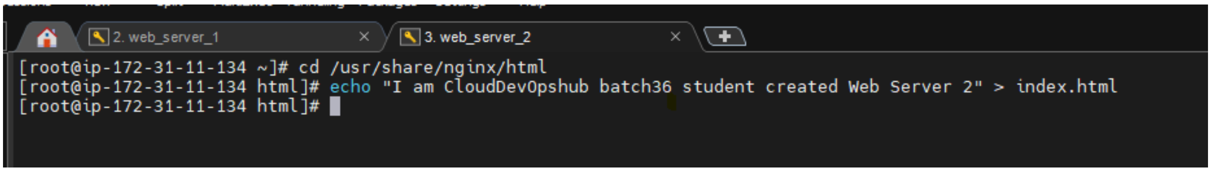


```
[root@ip-172-31-2-176 ~]# cd /usr/share/nginx/html
[root@ip-172-31-2-176 html]# echo "I am CloudDevOpshub batch36 student created Web Server 1" > index.html
[root@ip-172-31-2-176 html]#
```

**On Web Server 2:**

# Go to the NGINX HTML directory
cd /usr/share/nginx/html

# Replace the default index.html file with custom content
echo "I am CloudDevOpshub batch36 student created Web Server 2" > index.html

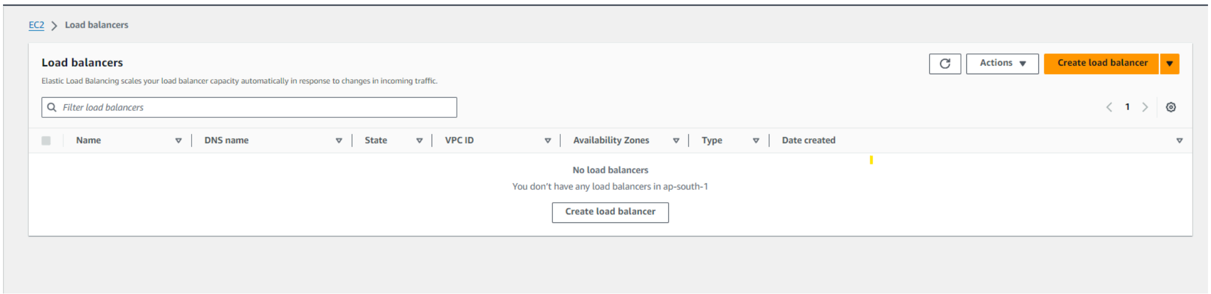


```
[root@ip-172-31-11-134 ~]# cd /usr/share/nginx/html
[root@ip-172-31-11-134 html]# echo "I am CloudDevOpshub batch36 student created Web Server 2" > index.html
[root@ip-172-31-11-134 html]#
```

## 4. Create an Elastic Load Balancer (ELB)

Now, create an Elastic Load Balancer (ELB) that will distribute traffic between these two web servers.

**Step 1:** In the AWS Management Console, go to EC2 > Load Balancers and click Create Load Balancer.



Load balancers

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

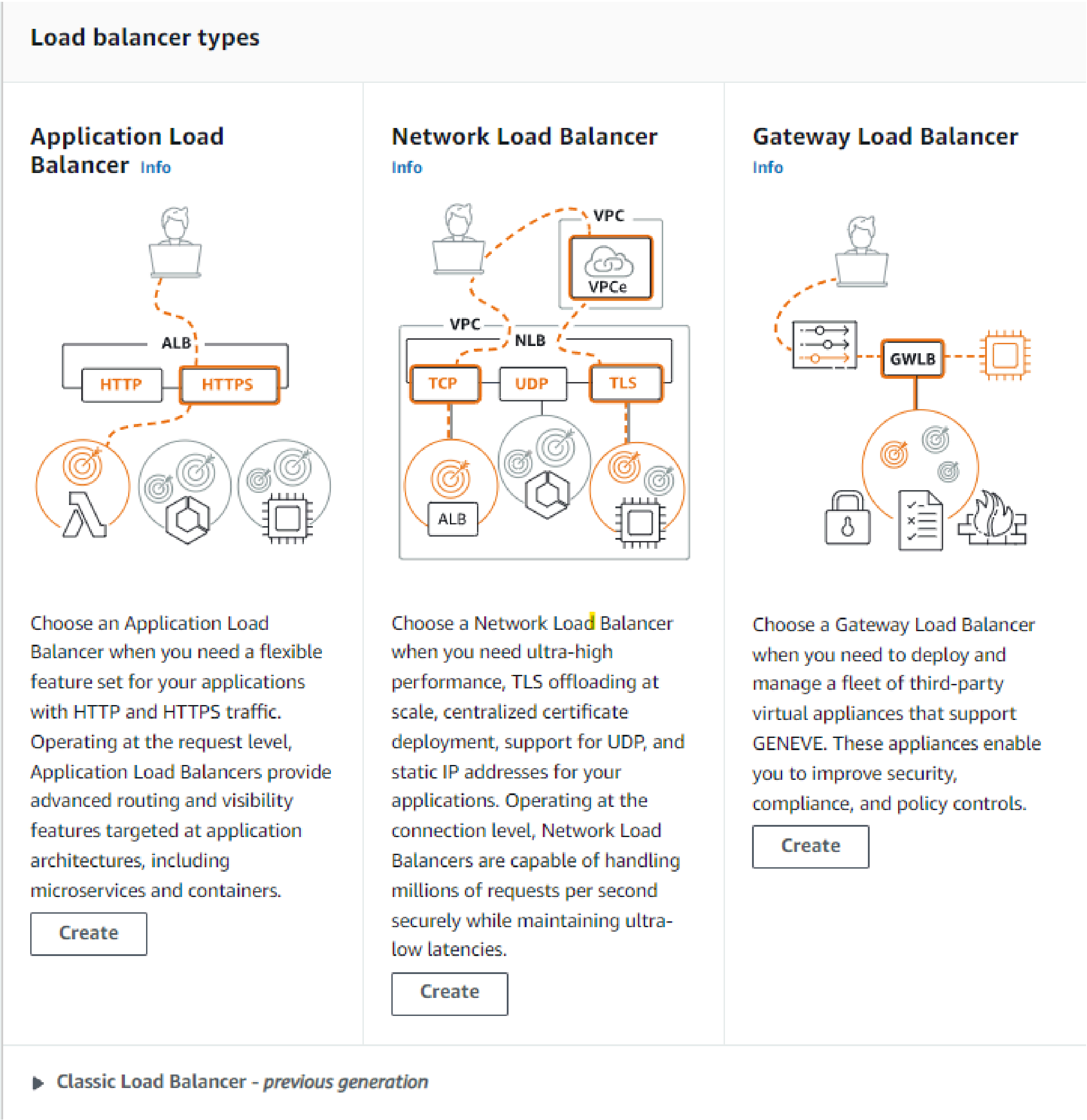
| Name | DNS name | State | VPC ID | Availability Zones | Type | Date created |
|---|---|---|---|---|---|---|

No load balancers

You don't have any load balancers in ap-south-1

Create load balancer

**Step 2:** Select Application Load Balancer.



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 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

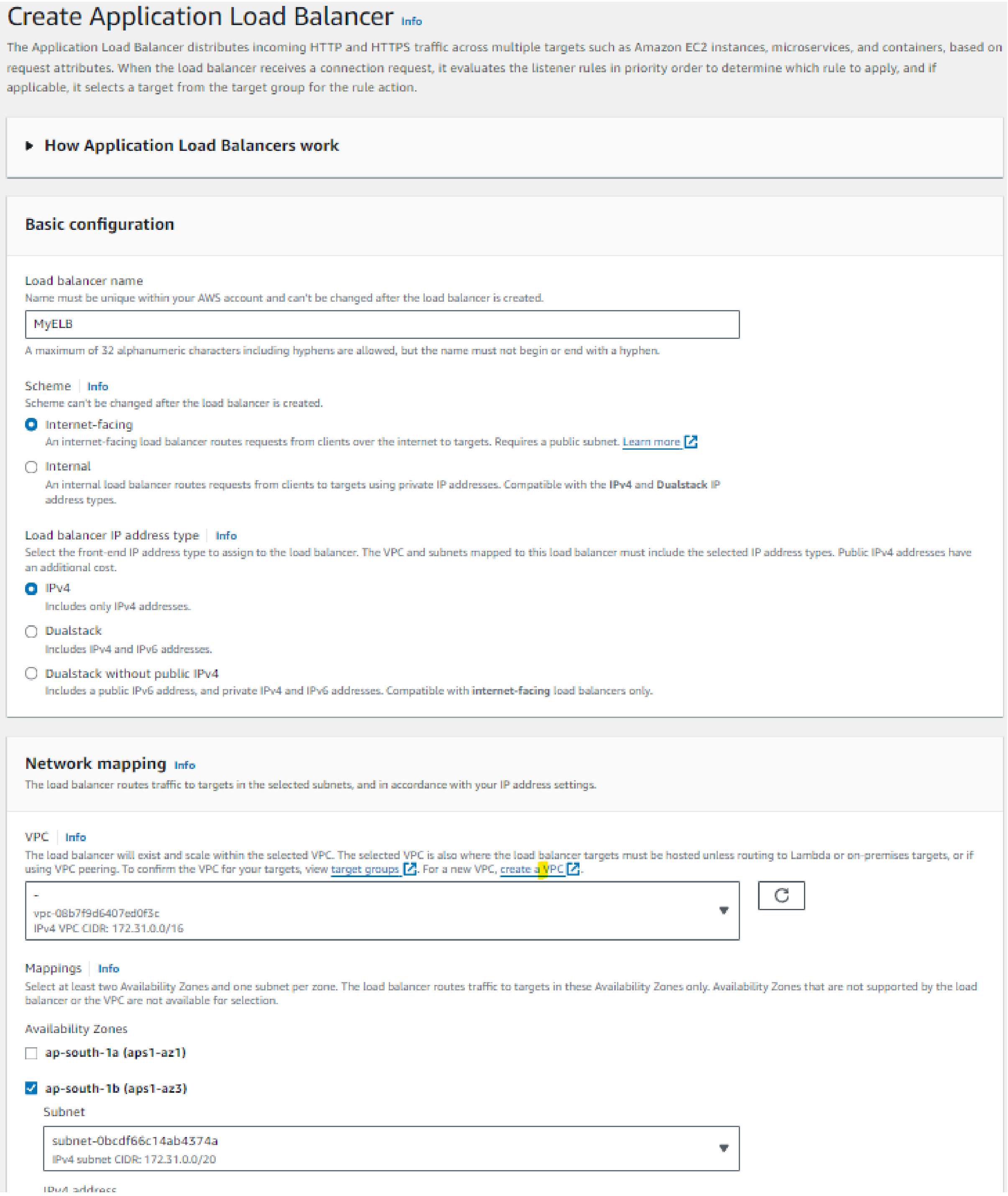
Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

▶ Classic Load Balancer - *previous generation*

**Step 3**: Configure the Load Balancer:

- Name: MyELB
- Scheme: Internet-facing
- Listeners: HTTP on port 80
- Availability Zones: Select the same region and Availability Zone where your EC2 instances are located.



# Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▸ **How Application Load Balancers work**

## Basic configuration

Load balancer name

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

MyELB

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. Compatible with the **IPv4** and **Dualstack** IP address types.

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 include the selected IP address types. Public IPv4 addresses have an additional cost.

● 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 only.

## Network mapping Info

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

VPC | Info

The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups. For a new VPC, create a VPC.

-
vpc-08b7f9d6407ed0f3c
IPv4 VPC CIDR: 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.

Availability Zones

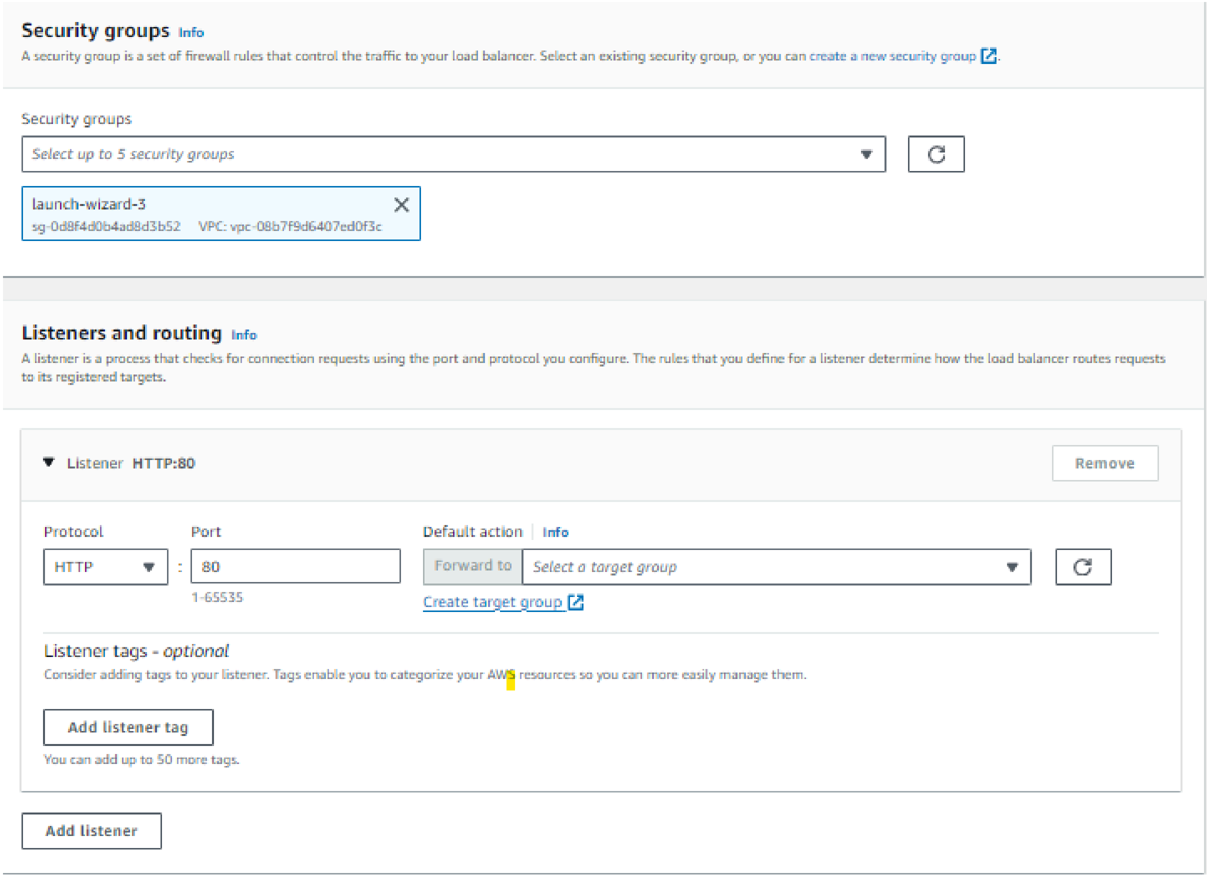
☐ **ap-south-1a (aps1-az1)**

☑ **ap-south-1b (aps1-az3)**

Subnet

subnet-0bcdf66c14ab4374a
IPv4 subnet CIDR: 172.31.0.0/20

IPv4 address



**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-3
sg-0d8f4d0b4ad8d3b52 VPC: vpc-08b7f9d6407ed0f3c

**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: Select a target group
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.

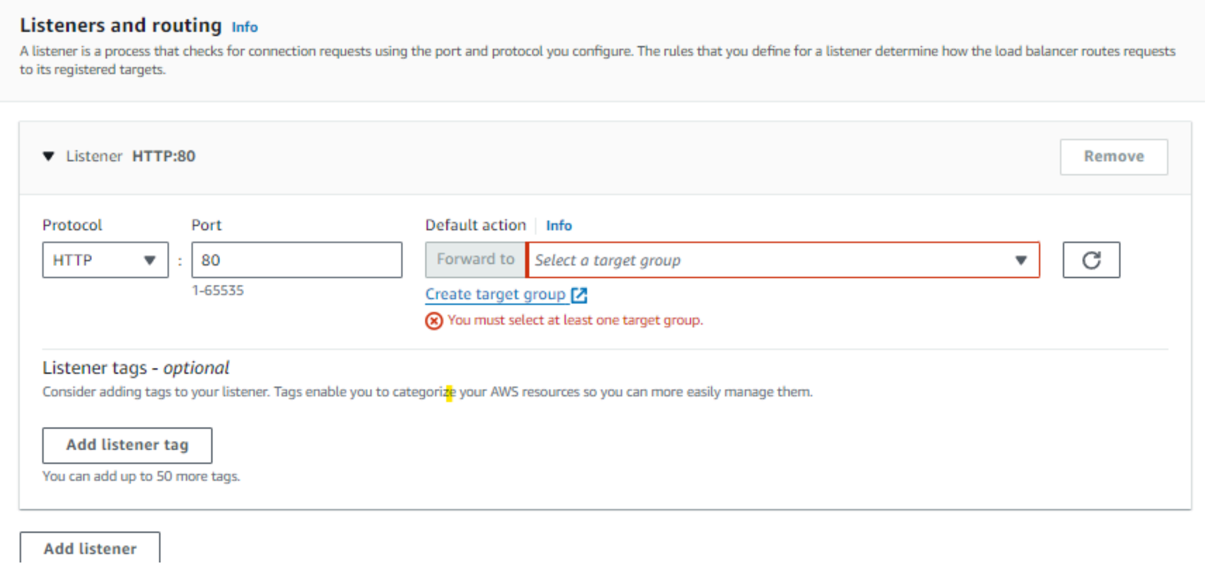
Add listener tag

You can add up to 50 more tags.

Add listener

**Step 4**: Configure the target group:

- Target Type: Instance
- Protocol: HTTP
- Port: 80
- Health Checks: HTTP, path /



**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: Select a target group
Create target group
ⓧ You must select at least one target group.

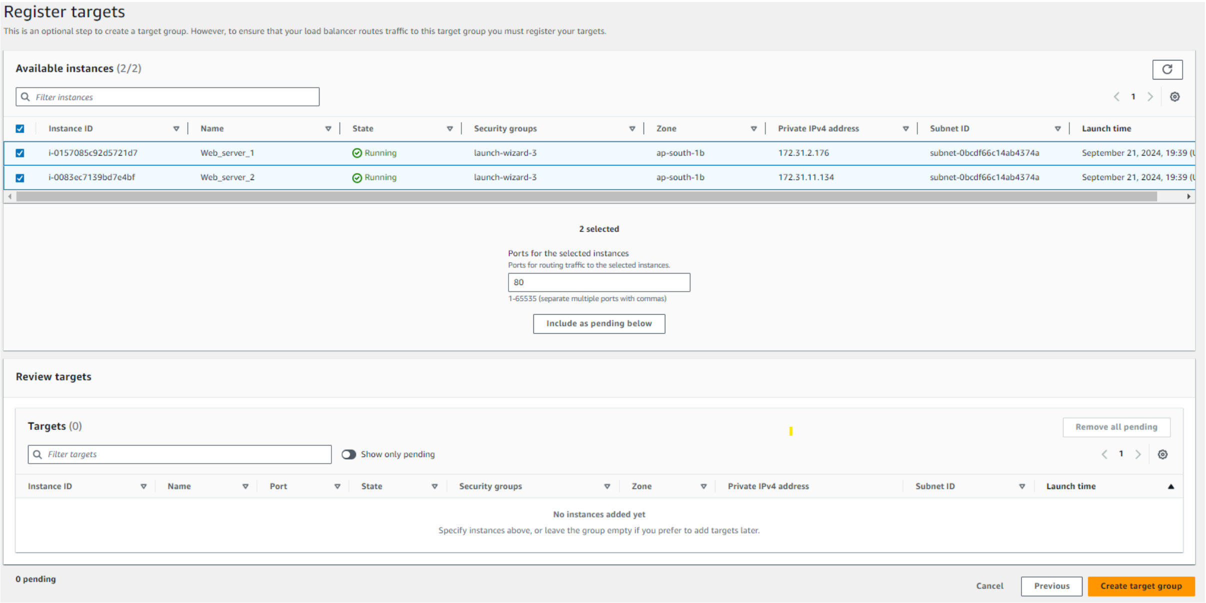
Listener tags - *optional*

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



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** (2/2)

Filter instances

| ☑ | Instance ID | Name | State | Security groups | Zone | Private IPv4 address | Subnet ID | Launch time |
|---|---|---|---|---|---|---|---|---|
| ☑ | i-0157085c92d5721d7 | Web_server_1 | ✓ Running | launch-wizard-3 | ap-south-1b | 172.31.2.176 | subnet-0bcdf66c14ab4374a | September 21, 2024, 19:39 (U |
| ☑ | i-0083ec7139bd7e4bf | Web_server_2 | ✓ Running | launch-wizard-3 | ap-south-1b | 172.31.11.134 | subnet-0bcdf66c14ab4374a | September 21, 2024, 19:39 (U |

**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

**Review targets**

**Targets** (0)

Filter targets

Show only pending

Remove all pending

| Instance ID | Name | Port | State | Security groups | Zone | Private IPv4 address | Subnet ID | Launch time |
|---|---|---|---|---|---|---|---|---|

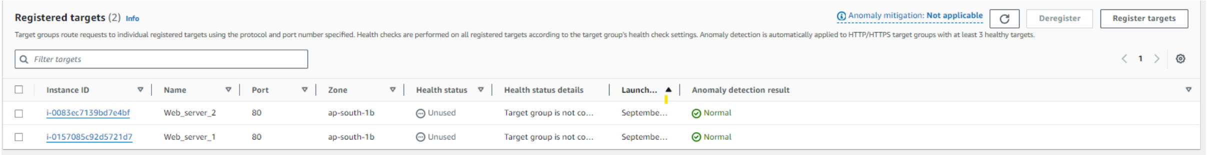
No instances added yet

Specify instances above, or leave the group empty if you prefer to add targets later.

0 pending

Cancel Previous Create target group

## **Step 5:** Register the EC2 instances (Web Server 1 and Web Server 2) to the target group.



**Registered targets** (2) Info

Anomaly mitigation: Not applicable

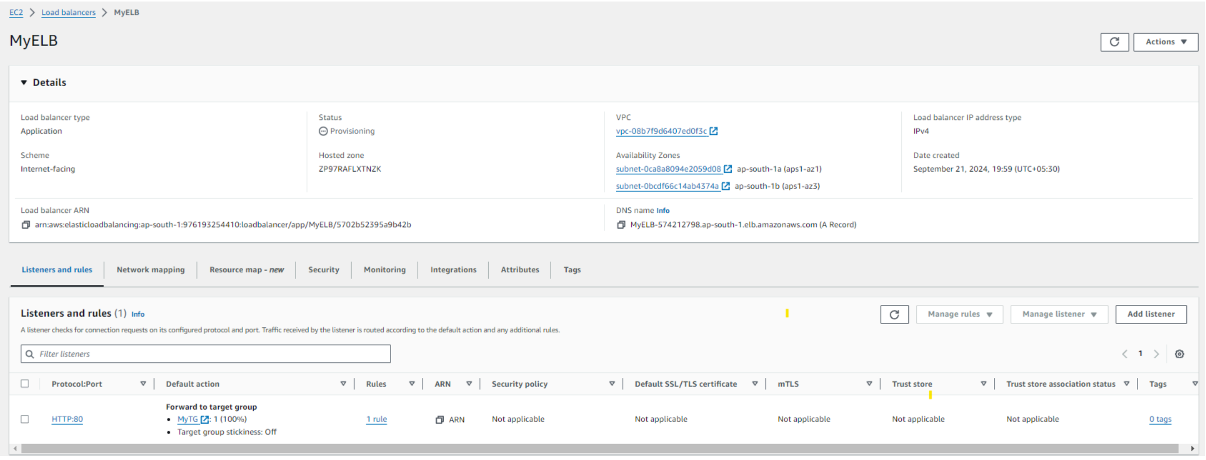
Deregister Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

| □ | Instance ID | Name | Port | Zone | Health status | Health status details | Launch... | Anomaly detection result |
|---|---|---|---|---|---|---|---|---|
| □ | i-0083ec7139bd7e4bf | Web_server_2 | 80 | ap-south-1b | Unused | Target group is not co... | Septembe... | ✓ Normal |
| □ | i-0157085c92d5721d7 | Web_server_1 | 80 | ap-south-1b | Unused | Target group is not co... | Septembe... | ✓ Normal |

## **Step 6:** Review and create the Load Balancer.



EC2 > Load balancers > MyELB

MyELB

Actions

**Details**

Load balancer type: Application

Scheme: Internet-facing

Status: Provisioning

Hosted zone: ZP97RAFLXTNZK

VPC: vpc-08b7f9d6407ed0f3c

Availability Zones: subnet-0ca8a8094e2059d08 ap-south-1a (aps1-az1)<br>subnet-0bcdf66c14ab4374a ap-south-1b (aps1-az3)

Load balancer IP address type: IPv4

Date created: September 21, 2024, 19:59 (UTC+05:30)

Load balancer ARN: arn:aws:elasticloadbalancing:ap-south-1:976193254410:loadbalancer/app/MyELB/5702b52395a9b42b

DNS name Info: MyELB-574212798.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules | Network mapping | Resource map - *new* | Security | Monitoring | Integrations | Attributes | Tags

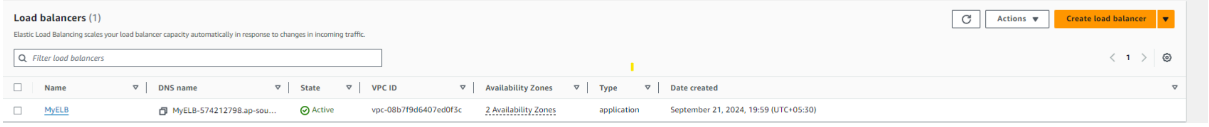
**Listeners and rules** (1) Info

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

| □ | Protocol:Port | Default action | Rules | ARN | Security policy | Default SSL/TLS certificate | mTLS | Trust store | Trust store association status | Tags |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | HTTP:80 | **Forward to target group**<br>• MyTG: 1 (100%)<br>• Target group stickiness: Off | 1 rule | ARN | Not applicable | Not applicable | Not applicable | Not applicable | Not applicable | 0 tags |



**Load balancers** (1)

Actions Create load balancer

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

Filter load balancers

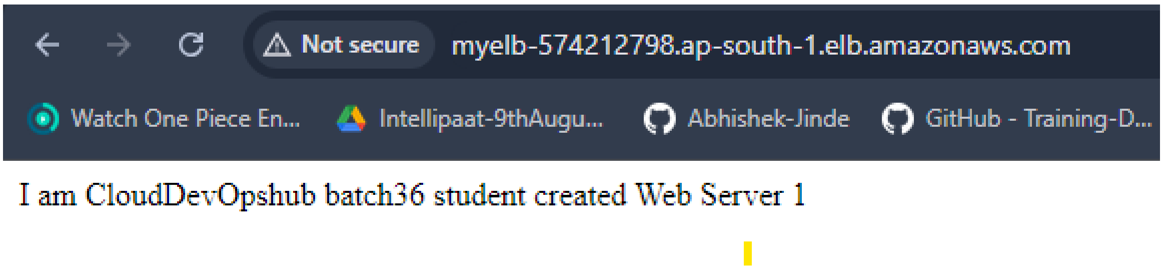
| □ | Name | DNS name | State | VPC ID | Availability Zones | Type | Date created |
|---|---|---|---|---|---|---|---|
| □ | MyELB | MyELB-574212798.ap-sou... | ✓ Active | vpc-08b7f9d6407ed0f3c | 2 Availability Zones | application | September 21, 2024, 19:59 (UTC+05:30) |

## 5. Test the Load Balancer

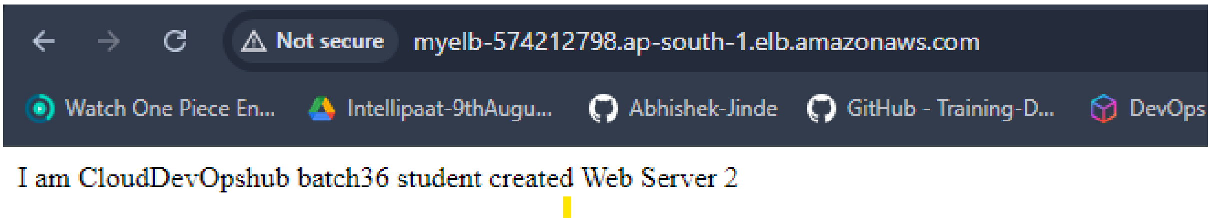
Once the Load Balancer is active, you can test it by visiting the Load Balancer's DNS name (found in the Load Balancer details) in your browser:

http://<ELB-DNS-Name>

You should see the content from either Web Server 1 or Web Server 2, and it will alternate as the load balancer distributes traffic between the two.



I am CloudDevOpshub batch36 student created Web Server 1



I am CloudDevOpshub batch36 student created Web Server 2