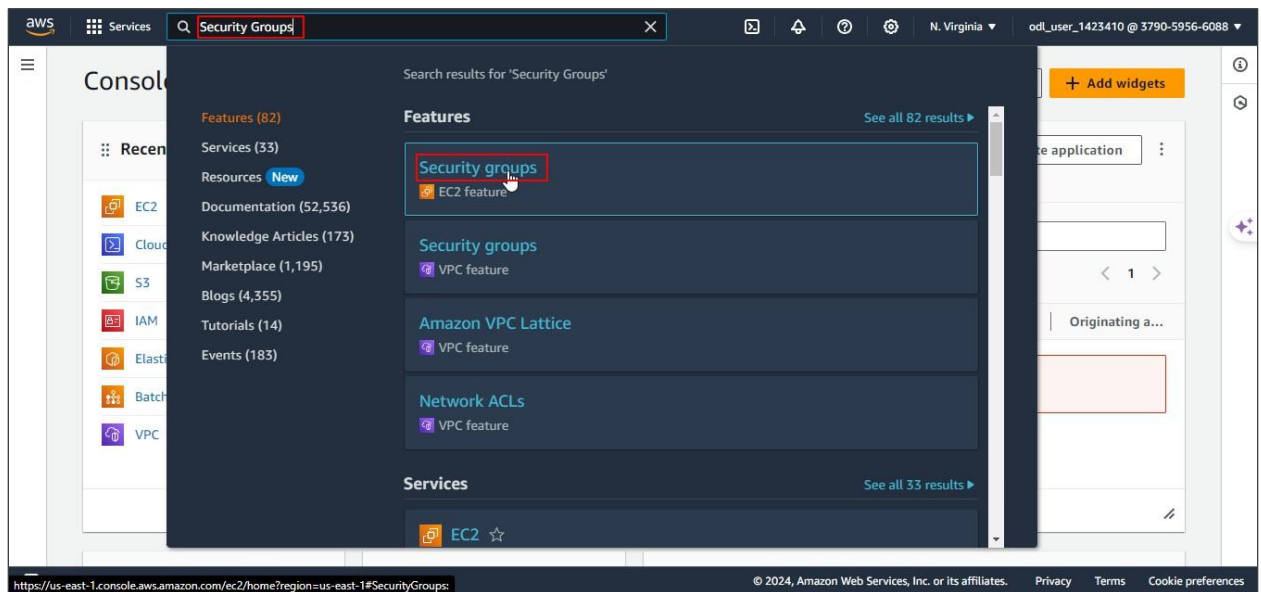


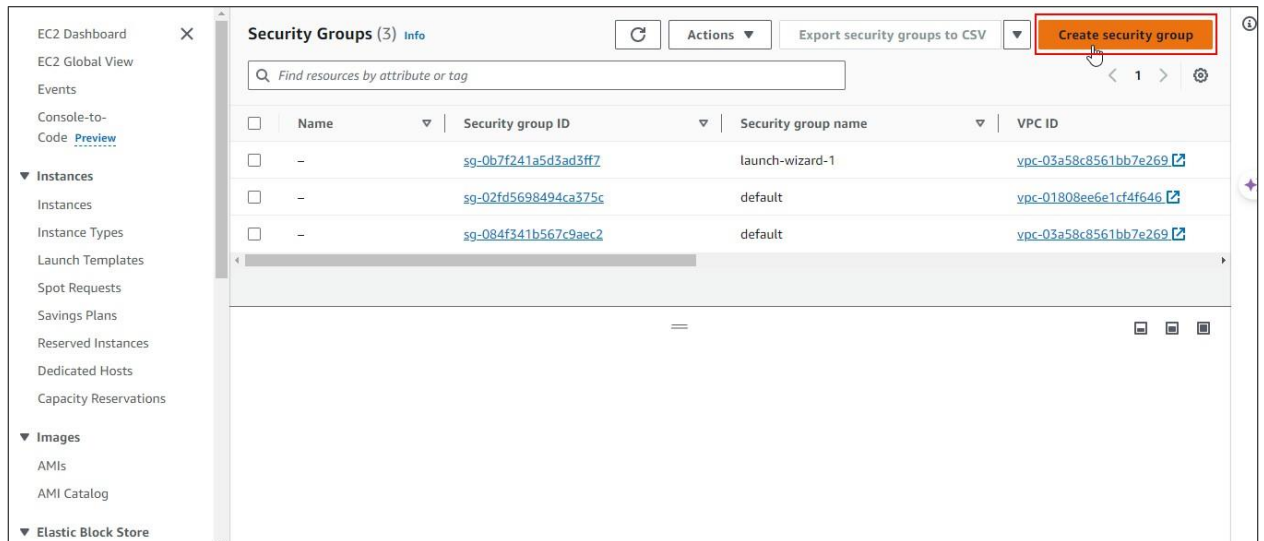
# Using a Classic Load Balancer to Distribute Traffic

## Step 1: Create a security group

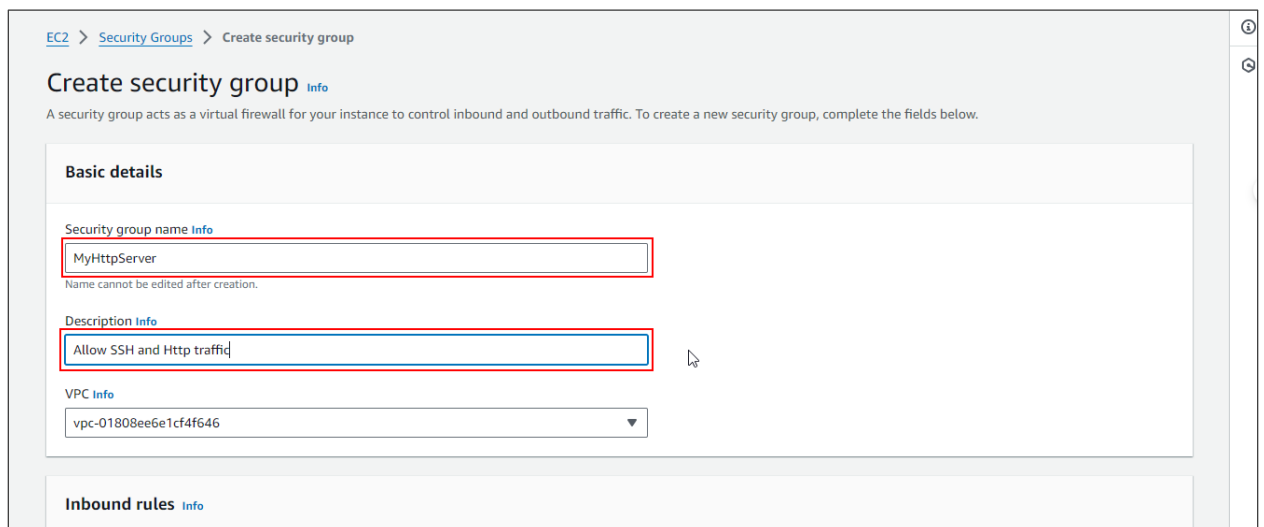
1.1 Navigate to the AWS Management Console home page, search for and click on **Security Groups**



## 1.2 Click on **Create security group**



## 1.3 In the **Create security group** section, add **MyHttpServer** for the Security group name and **Allow SSH and Http traffic** for the Description



#### 1.4 Set the Inbound rules type to **SSH** and **HTTP** with source set to **Anywhere IPv4**

The screenshot shows the 'Inbound rules' section of the AWS Security Groups console. The VPC is set to 'vpc-01808ee6e1cf4f646'. There are two inbound rules defined:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Any...	
HTTP	TCP	80	Any...	

Below the rules, there is a warning message: 'Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

#### 1.5 Click on **Create security group**

The screenshot shows the 'Create security group' page in the AWS console. At the bottom right, the 'Create security group' button is highlighted with a red box. Above it, there is a warning message: 'Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.'

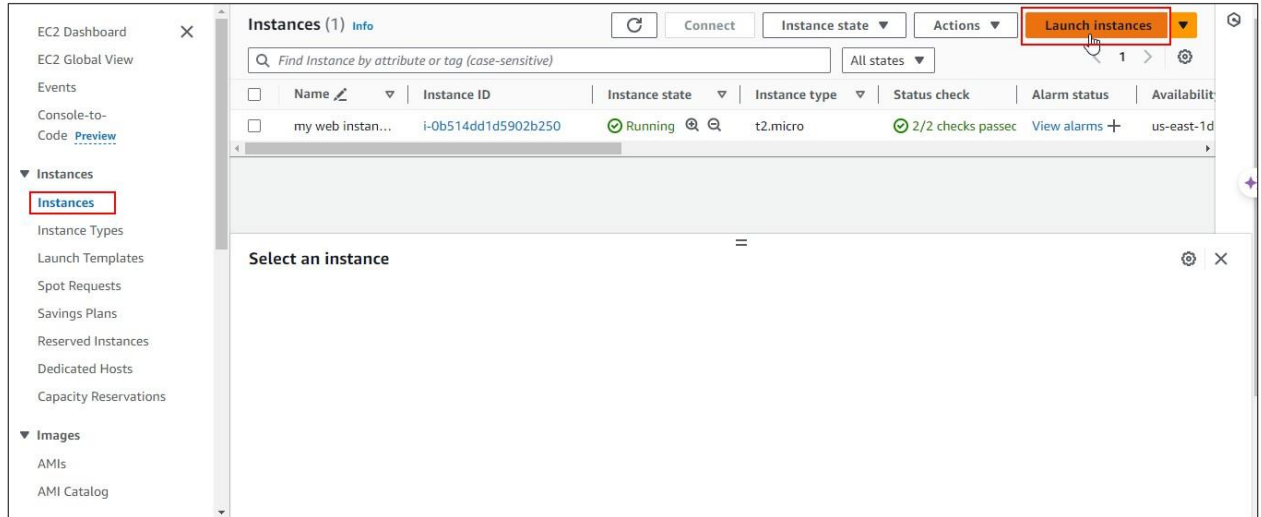
The screenshot shows the 'Details' page of the newly created security group 'sg-0da5d191c9a4676ea - MyHttpServer'. The page displays the following details:

Details			
Security group name	Security group ID	Description	VPC ID
MyHttpServer	sg-0da5d191c9a4676ea	Allow SSH and Http traffic	vpc-01808ee6e1cf4f646
Owner	Inbound rules count	Outbound rules count	
379059566088	2 Permission entries	1 Permission entry	

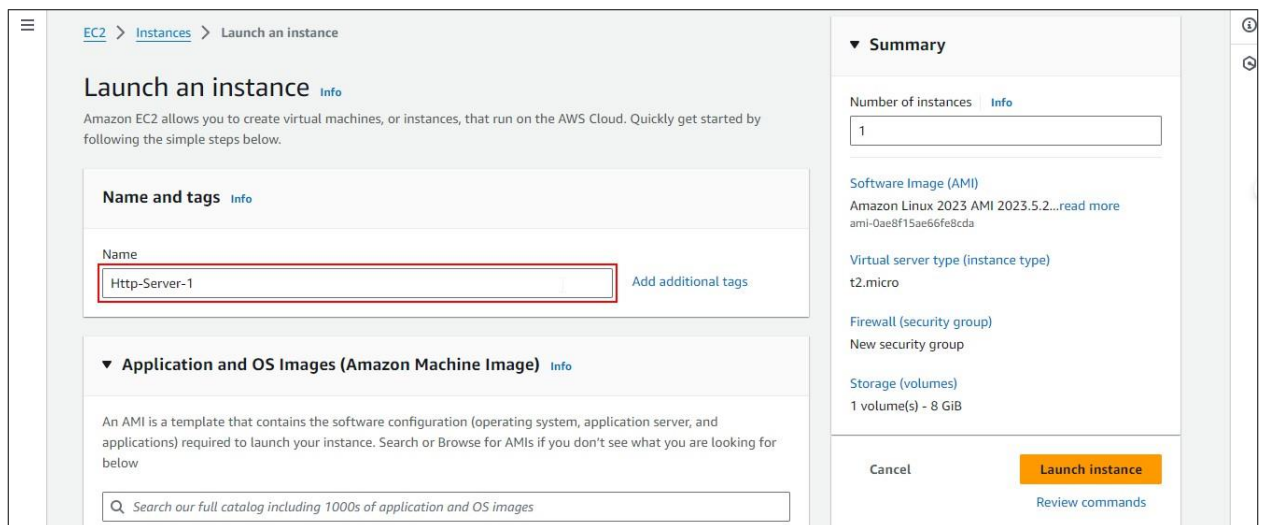
The security group has been created successfully.

## Step 2: Launch instances with different availability zones

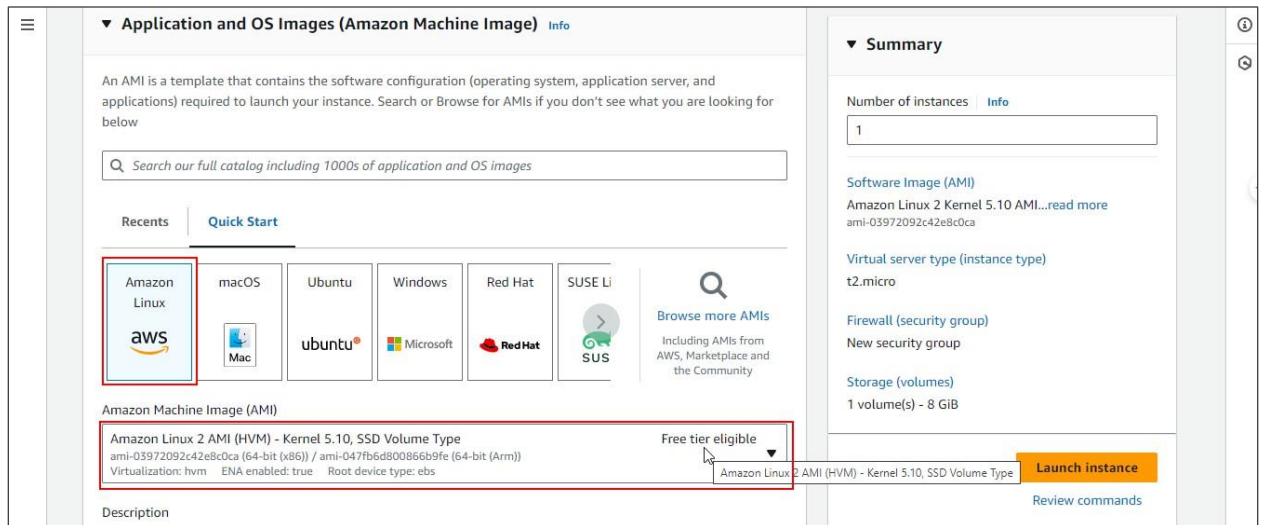
### 2.1 Navigate to **Instances** and click on **Launch instances**



### 2.2 Add the Name as **Http-Server-1**

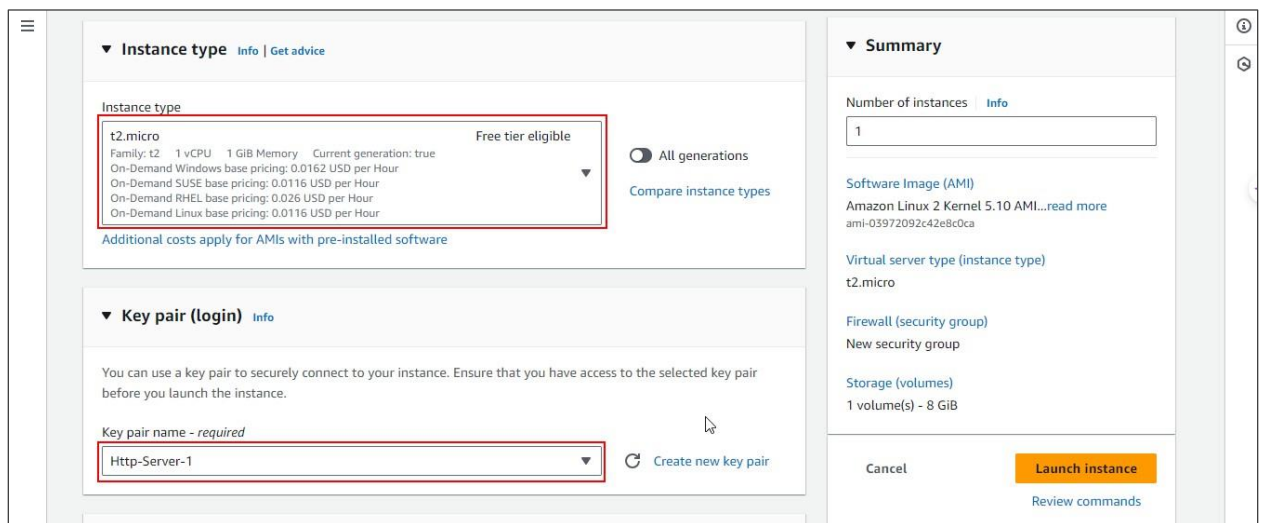


## 2.3 Select **Amazon Linux** as the OS and **Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type** as the AMI



Launch the first instance by assigning it a name and specifying the subnet information along with the availability zone

## 2.4 Select the **Instance type** as **t2.micro**, create a new key pair, and name it **Http-Server-1**



## 2.5 Enter the network settings details as shown:

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Http-Server-1 [Create new key pair](#)

▼ **Network settings** [Info](#)

VPC - *required* [Info](#)

vpc-01808ee6e1cf4f646 (default) [Create new VPC](#)

Subnet [Info](#)

subnet-0db63e80efd46fbb [Create new subnet](#)

VPC: vpc-01808ee6e1cf4f646 Owner: 379059566088 Availability Zone: us-east-1c  
Zone type: Availability Zone IP addresses available: 4091 CIDR: 172.31.0.0/20

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

▼ **Summary**

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-03972092c42e8c0ca

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel [Launch instance](#) [Review commands](#)

## 2.6 Click on **Select existing security group** and select **MyHttpServer**

vpc-01808ee6e1cf4f646 (default) [Create new VPC](#)

Subnet [Info](#)

subnet-0db63e80efd46fbb [Create new subnet](#)

VPC: vpc-01808ee6e1cf4f646 Owner: 379059566088 Availability Zone: us-east-1c  
Zone type: Availability Zone IP addresses available: 4091 CIDR: 172.31.0.0/20

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

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

MyHttpServer sg-0da5d191c9a4676ea [Compare security group rules](#)

VPC: vpc-01808ee6e1cf4f646

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

► Advanced network configuration

▼ **Summary**

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-03972092c42e8c0ca

Virtual server type (instance type)

t2.micro

Firewall (security group)

MyHttpServer

Storage (volumes)

1 volume(s) - 8 GiB

Cancel [Launch instance](#) [Review commands](#)

2.7 Provide the user data code under the **Advanced details** section to install and start the HTTP server, and click on **Launch instance**

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

```
# Use this for your user data (script from top to bottom)
```

```
# install httpd (Linux 2 version)
```

```
yum update -y
```

```
yum install -y httpd
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

```
echo "<h1>Welcome to AWSNetworks web-server $(hostname -f)</h1>" >
```

```
/var/www/html/index.html
```

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

[Choose file](#)

```
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Welcome to AWSNetworks web-server $(hostname -f)</h1>" >
/var/www/html/index.html
```

☐ User data has already been base64 encoded

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-03972092c42e8c0ca

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
MyHttpServer

Storage (volumes)  
1 volume(s) - 8 GiB

Cancel **Launch instance**  
[Review console](#)

2.8 Repeat the steps to launch the second instance with a different availability zone

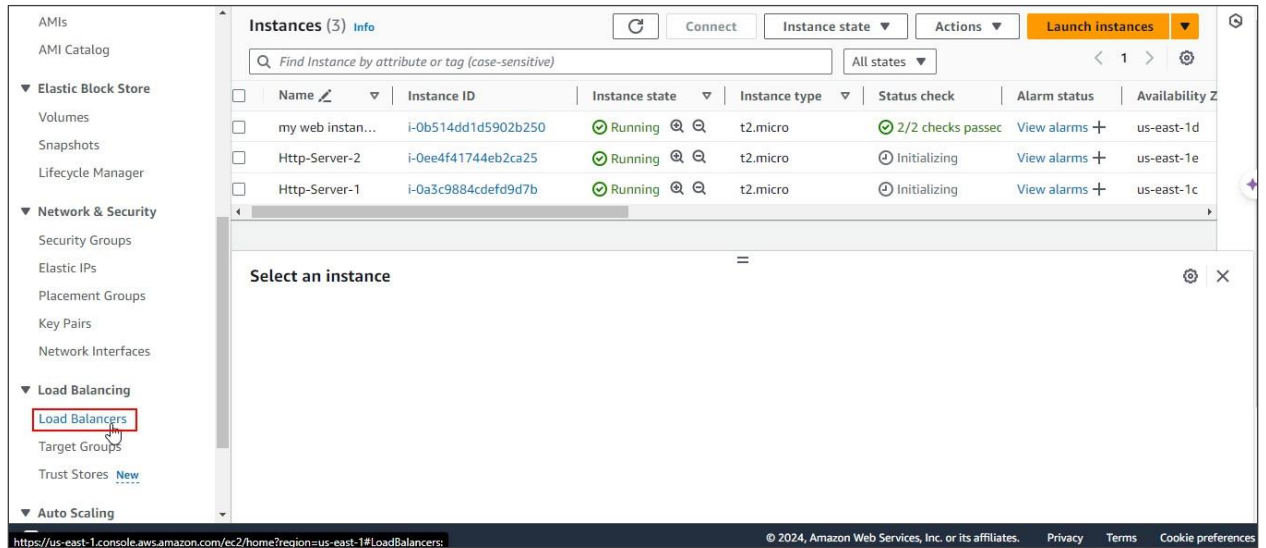
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	my web instan...	i-0b514dd1d5902b250	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1d
<input type="checkbox"/>	Http-Server-2	i-0ee4f41744eb2ca25	Initializing	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1e
<input type="checkbox"/>	Http-Server-1	i-0a3c9884cdefd9d7b	Initializing	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c

Select an instance

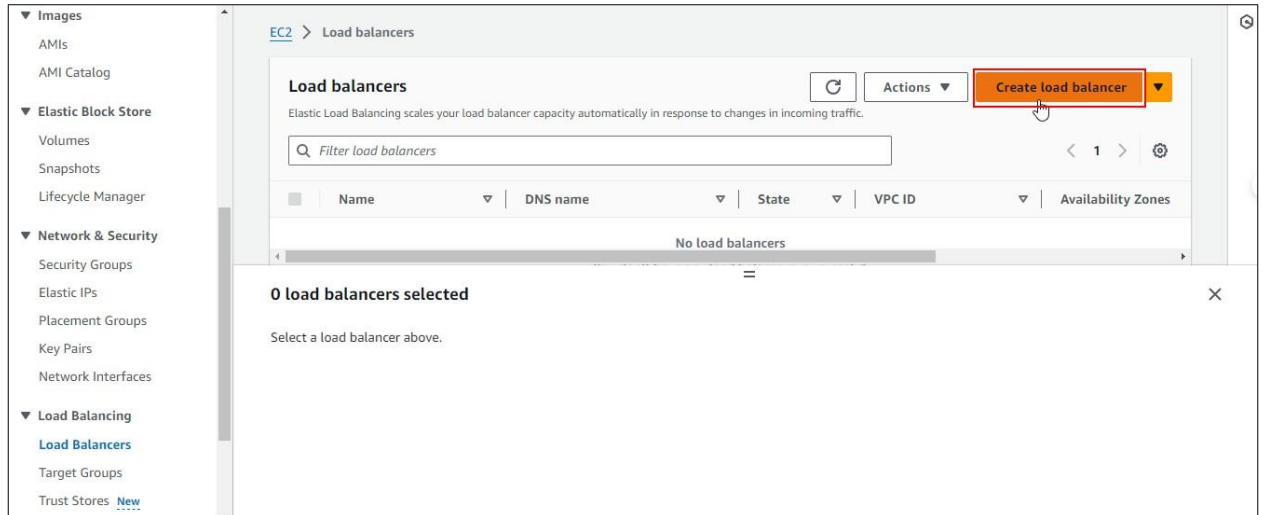
The instances with different availability zones have been launched successfully.

## Step 3: Create the Classic Load Balancer

### 3.1 Navigate to **Load Balancers** on the left pane and click on it

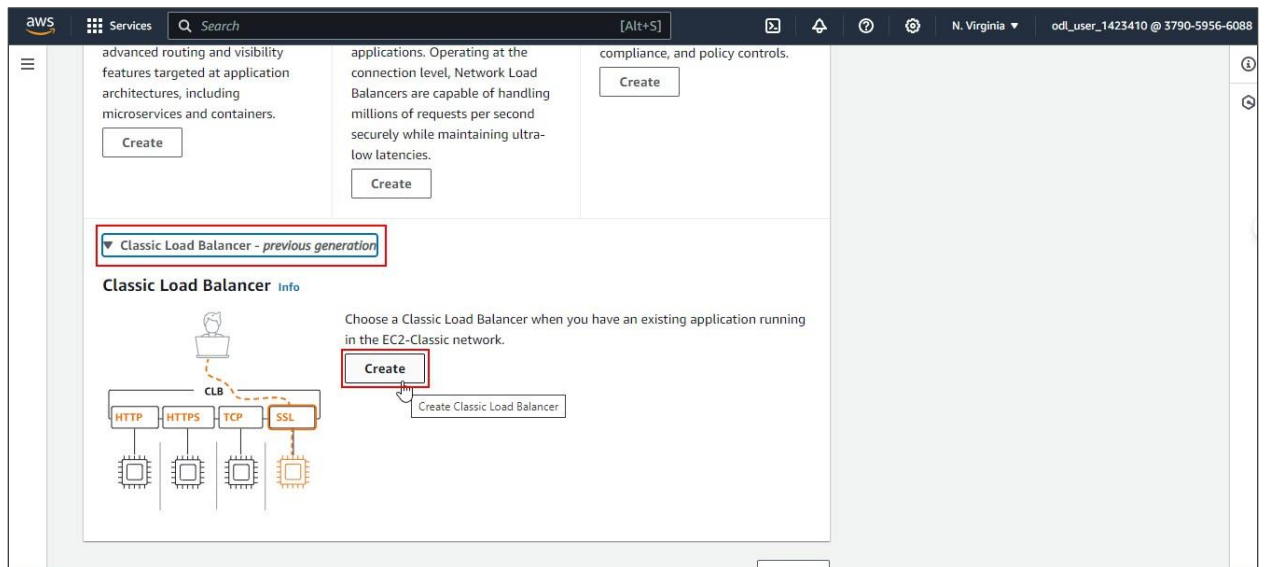


### 3.2 Click on the **Create Load Balancer** button

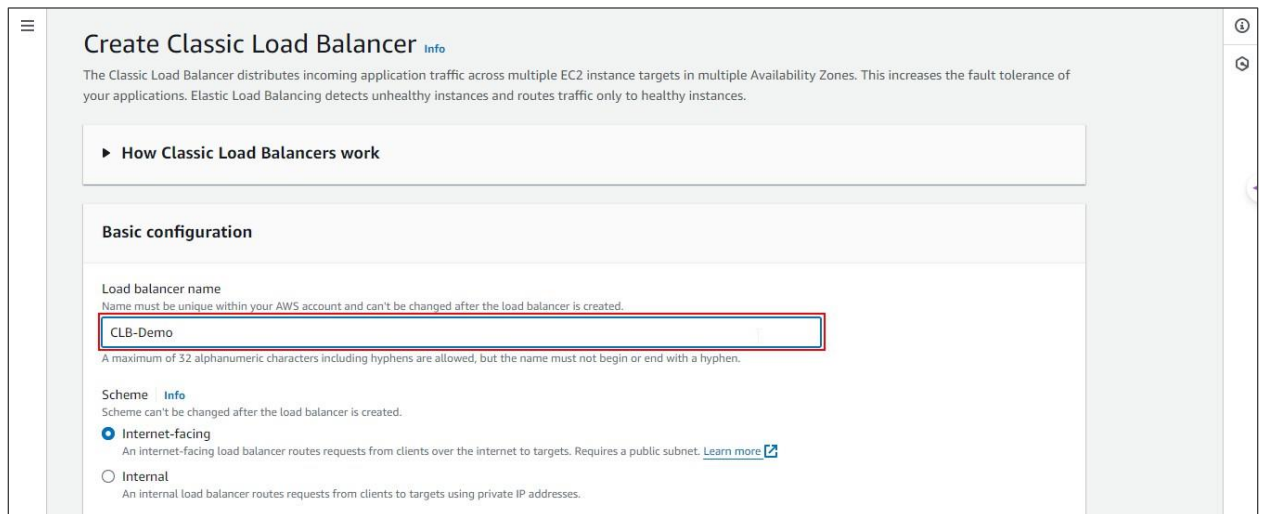




### 3.3 Select **Classic Load Balancer** and click on **Create**



### 3.4 Enter **CLB-Demo** as the **Load Balancer Name**



### 3.5 Select **us-east-1c** and **us-east-1e** as the **Availability Zones** in the **Mappings** section

**Mappings**

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

**Availability Zones**

☐ us-east-1a (use1-az4)

☐ us-east-1b (use1-az6)

☒ **us-east-1c (use1-az1)**

**Subnet**

subnet-0db63e80efd46fbb  
IPv4 subnet CIDR: 172.31.0.0/20

IPv4 address

Assigned by AWS

☐ us-east-1d (use1-az2)

☒ **us-east-1e (use1-az3)**

**Subnet**

subnet-0338f45f2f5d78423  
IPv4 subnet CIDR: 172.31.0.0/20

IPv4 address

Assigned by AWS

### 3.6 Select the existing security groups **MyHttpServer** and **default**

**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

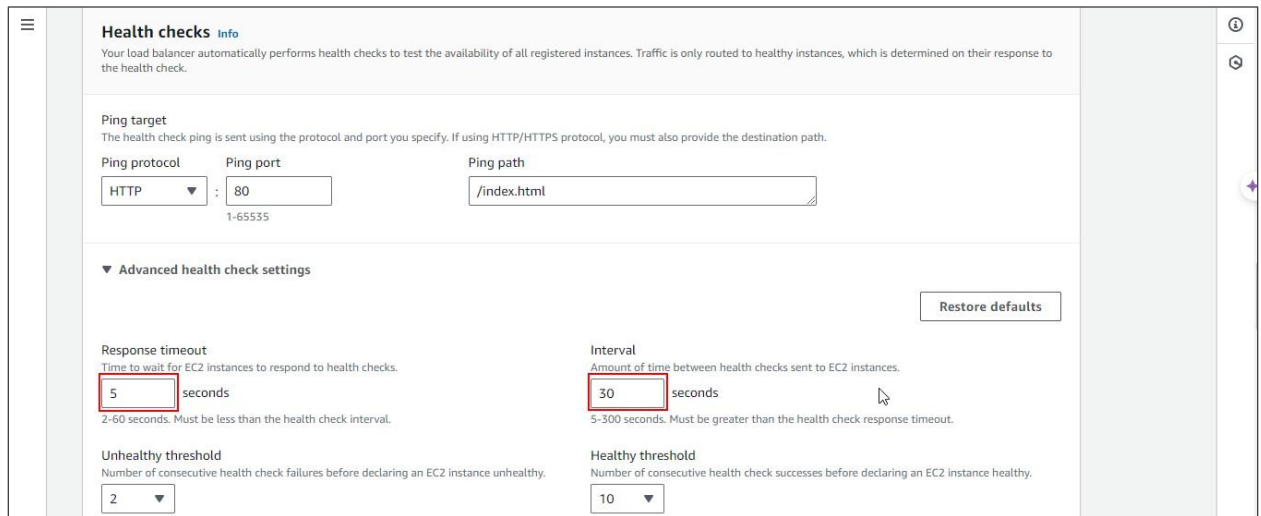
**default**  **MyHttpServer**

sg-02fd5698494ca375c VPC: vpc-01808ee6e1cf4f646 sg-0da5d191c9a4676ea VPC: vpc-01808ee6e1cf4f646

**Listeners and routing** [Info](#)

A listener is a process that checks for connection requests using the protocol and port you configure. The settings you define for a listener determine how the load balancer routes requests to its registered targets.

### 3.7 Change the Response timeout to 5 seconds and the Interval timeout to 30 seconds in the **Health checks** domain



**Health checks** [Info](#)

Your load balancer automatically performs health checks to test the availability of all registered instances. Traffic is only routed to healthy instances, which is determined on their response to the health check.

**Ping target**  
The health check ping is sent using the protocol and port you specify. If using HTTP/HTTPS protocol, you must also provide the destination path.

Ping protocol: HTTP : Ping port: 80 Ping path: /index.html

▼ **Advanced health check settings** Restore defaults

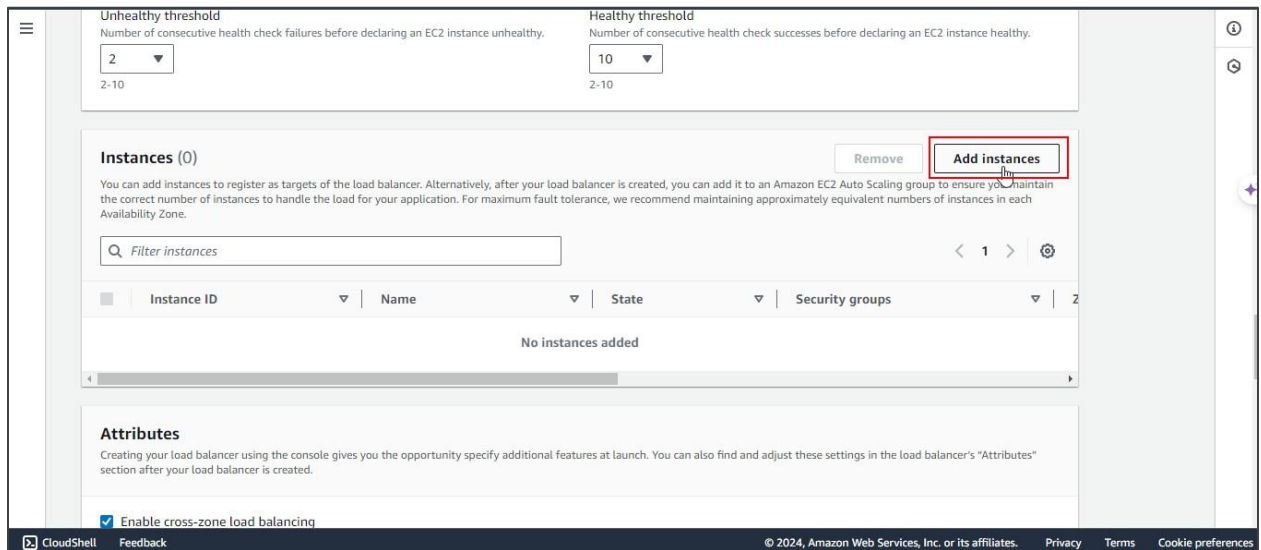
**Response timeout**  
Time to wait for EC2 instances to respond to health checks.  
**5** seconds  
2-60 seconds. Must be less than the health check interval.

**Interval**  
Amount of time between health checks sent to EC2 instances.  
**30** seconds  
5-300 seconds. Must be greater than the health check response timeout.

**Unhealthy threshold**  
Number of consecutive health check failures before declaring an EC2 instance unhealthy.  
2

**Healthy threshold**  
Number of consecutive health check successes before declaring an EC2 instance healthy.  
10

### 3.8 Click on **Add instances**



**Unhealthy threshold**  
Number of consecutive health check failures before declaring an EC2 instance unhealthy.  
2

**Healthy threshold**  
Number of consecutive health check successes before declaring an EC2 instance healthy.  
10

**Instances (0)** Remove **Add instances**

You can add instances to register as targets of the load balancer. Alternatively, after your load balancer is created, you can add it to an Amazon EC2 Auto Scaling group to ensure you maintain the correct number of instances to handle the load for your application. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone.

Filter instances

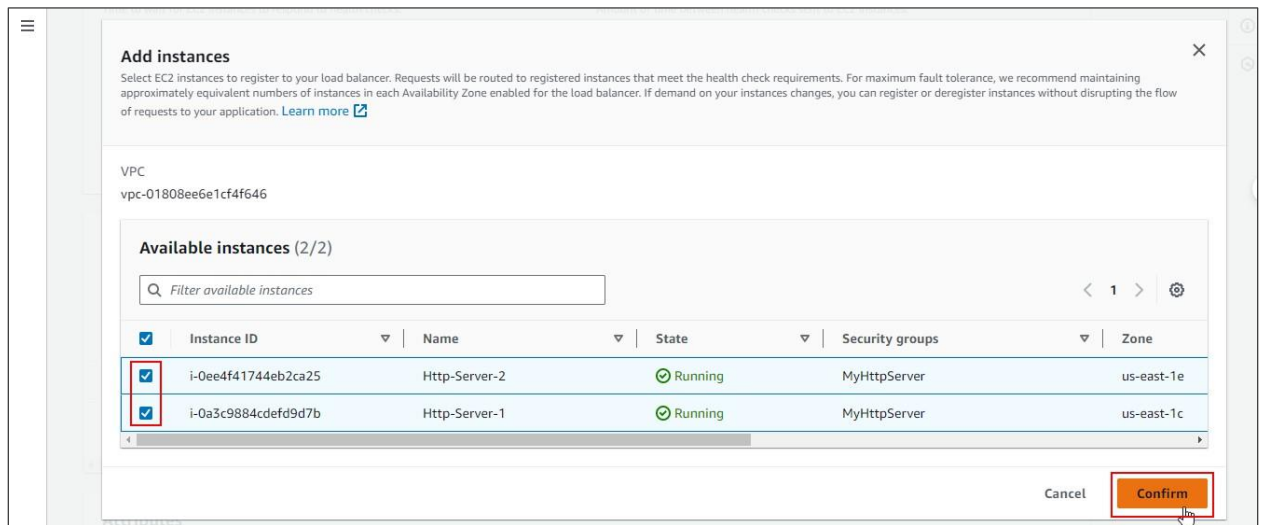
Instance ID	Name	State	Security groups
No instances added			

**Attributes**  
Creating your load balancer using the console gives you the opportunity specify additional features at launch. You can also find and adjust these settings in the load balancer's "Attributes" section after your load balancer is created.

☒ Enable cross-zone load balancing

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### 3.9 Select both instances and click on **Confirm**



**Add instances**

Select EC2 instances to register to your load balancer. Requests will be routed to registered instances that meet the health check requirements. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone enabled for the load balancer. If demand on your instances changes, you can register or deregister instances without disrupting the flow of requests to your application. [Learn more](#)

VPC  
vpc-01808ee6e1cf4f646

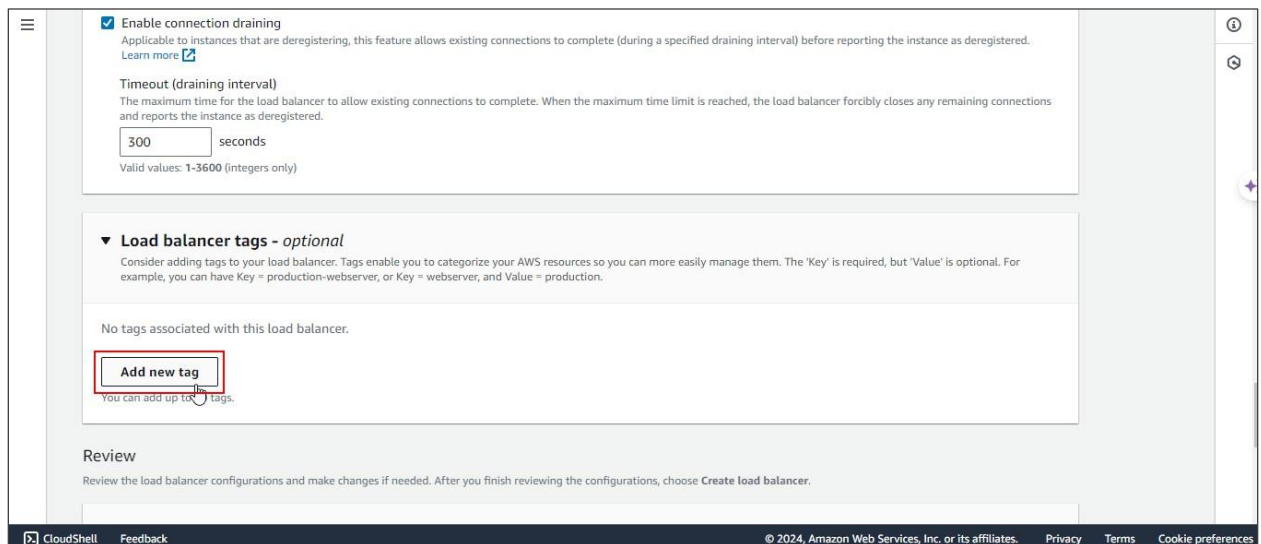
**Available instances (2/2)**

Filter available instances

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone
<input checked="" type="checkbox"/>	i-0ee4f41744eb2ca25	Http-Server-2	Running	MyHttpServer	us-east-1e
<input checked="" type="checkbox"/>	i-0a3c9884cdefd9d7b	Http-Server-1	Running	MyHttpServer	us-east-1c

Cancel **Confirm**

### 3.10 Click on **Add new tag** in the **Load balancer tags - optional** section



☒ **Enable connection draining**  
Applicable to instances that are deregistering, this feature allows existing connections to complete (during a specified draining interval) before reporting the instance as deregistered. [Learn more](#)

**Timeout (draining interval)**  
The maximum time for the load balancer to allow existing connections to complete. When the maximum time limit is reached, the load balancer forcibly closes any remaining connections and reports the instance as deregistered.

300 seconds  
Valid values: 1-3600 (integers only)

**Load balancer tags - optional**  
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.

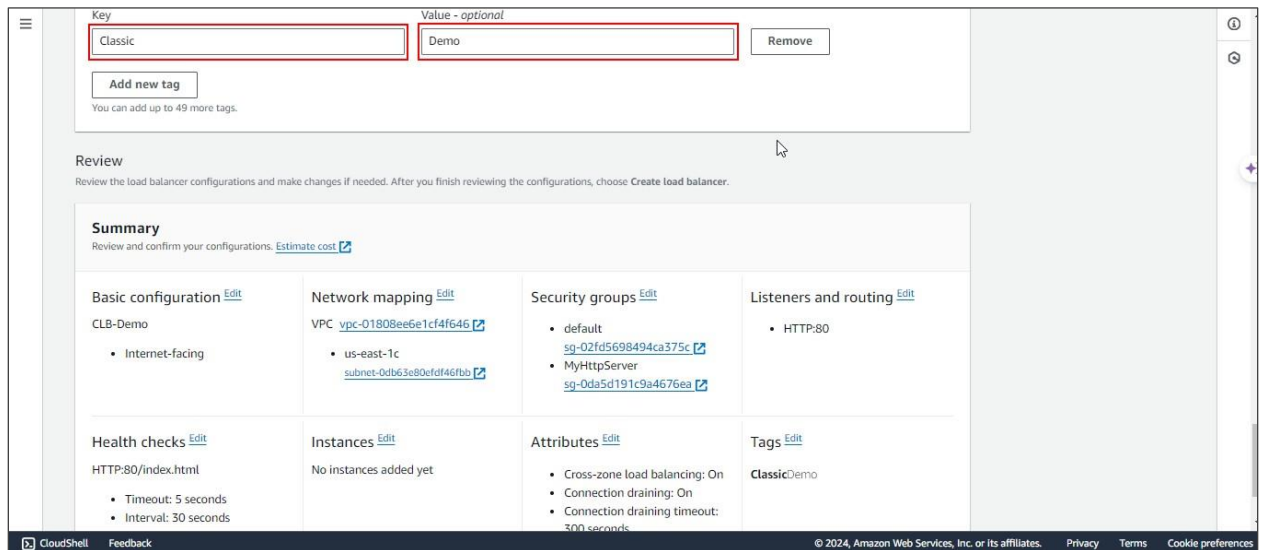
No tags associated with this load balancer.

**Add new tag**  
You can add up to 50 tags.

**Review**  
Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose **Create load balancer**.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### 3.11 Provide a **Key** and **Value** name for the tags, then verify the details



Key: Classic Value - optional: Demo Remove

Add new tag

You can add up to 49 more tags.

**Review**

Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose **Create load balancer**.

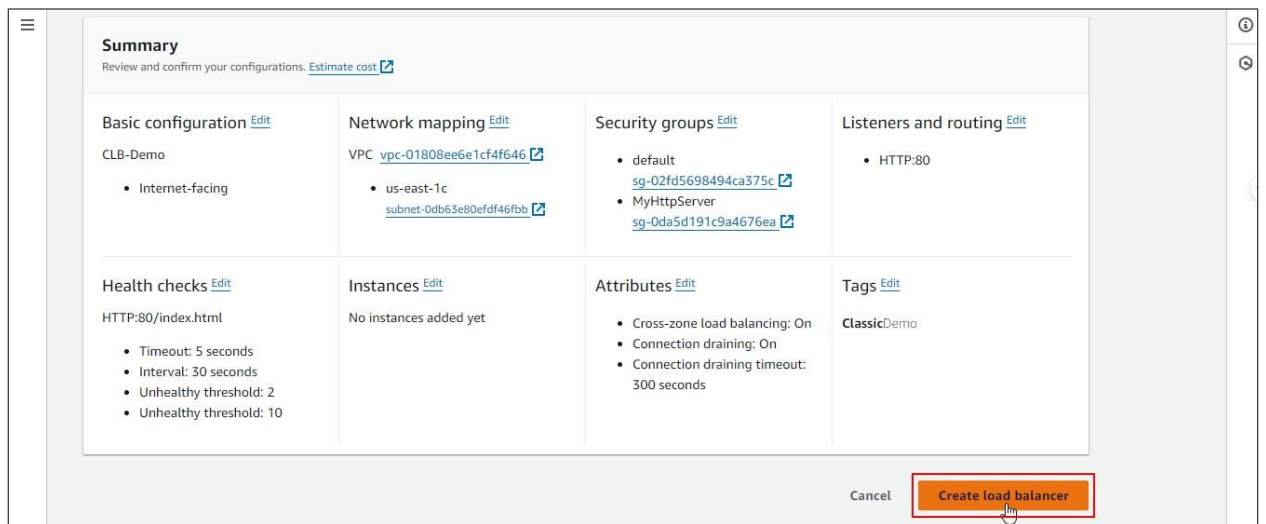
**Summary**

Review and confirm your configurations. [Estimate cost](#)

<b>Basic configuration</b> <a href="#">Edit</a> CLB-Demo <ul style="list-style-type: none"><li>Internet-facing</li></ul>	<b>Network mapping</b> <a href="#">Edit</a> VPC <a href="#">vpc-01808ee6e1cf4f646</a> <ul style="list-style-type: none"><li>us-east-1c<ul style="list-style-type: none"><li><a href="#">subnet-0db63e80efd46fbb</a></li></ul></li></ul>	<b>Security groups</b> <a href="#">Edit</a> <ul style="list-style-type: none"><li>default<ul style="list-style-type: none"><li><a href="#">sg-02fd5698494ca375c</a></li></ul></li><li>MyHttpServer<ul style="list-style-type: none"><li><a href="#">sg-0da5d191c9a4676ea</a></li></ul></li></ul>	<b>Listeners and routing</b> <a href="#">Edit</a> <ul style="list-style-type: none"><li>HTTP:80</li></ul>
<b>Health checks</b> <a href="#">Edit</a> HTTP:80/index.html <ul style="list-style-type: none"><li>Timeout: 5 seconds</li><li>Interval: 30 seconds</li></ul>	<b>Instances</b> <a href="#">Edit</a> No instances added yet	<b>Attributes</b> <a href="#">Edit</a> <ul style="list-style-type: none"><li>Cross-zone load balancing: On</li><li>Connection draining: On</li><li>Connection draining timeout: 300 seconds</li></ul>	<b>Tags</b> <a href="#">Edit</a> <b>Classic</b> Demo

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### 3.12 Click on **Create load balancer**



**Summary**

Review and confirm your configurations. [Estimate cost](#)

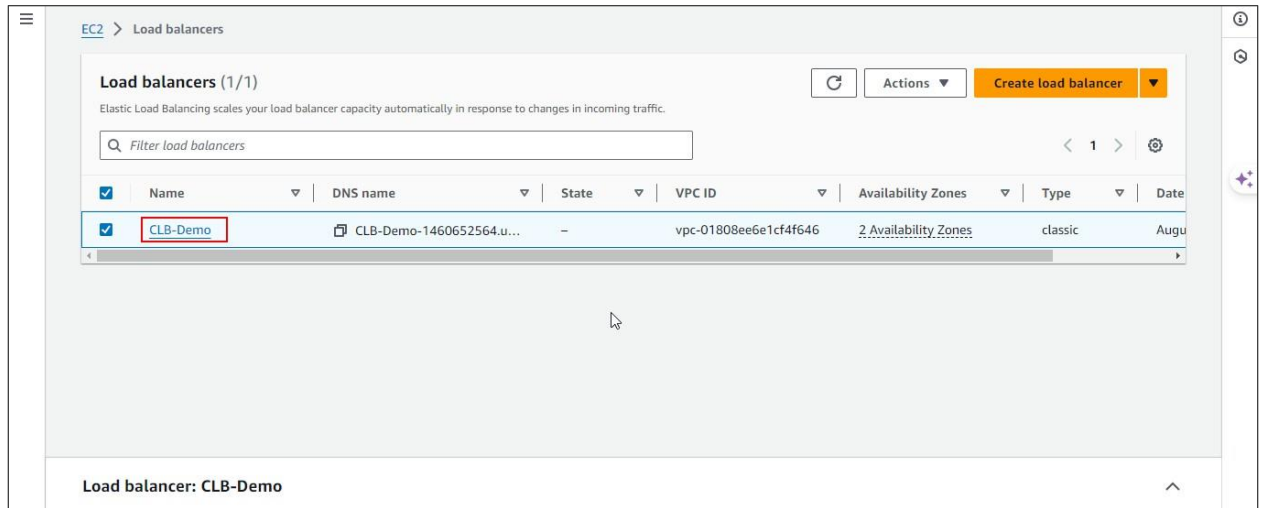
<b>Basic configuration</b> <a href="#">Edit</a> CLB-Demo <ul style="list-style-type: none"><li>Internet-facing</li></ul>	<b>Network mapping</b> <a href="#">Edit</a> VPC <a href="#">vpc-01808ee6e1cf4f646</a> <ul style="list-style-type: none"><li>us-east-1c<ul style="list-style-type: none"><li><a href="#">subnet-0db63e80efd46fbb</a></li></ul></li></ul>	<b>Security groups</b> <a href="#">Edit</a> <ul style="list-style-type: none"><li>default<ul style="list-style-type: none"><li><a href="#">sg-02fd5698494ca375c</a></li></ul></li><li>MyHttpServer<ul style="list-style-type: none"><li><a href="#">sg-0da5d191c9a4676ea</a></li></ul></li></ul>	<b>Listeners and routing</b> <a href="#">Edit</a> <ul style="list-style-type: none"><li>HTTP:80</li></ul>
<b>Health checks</b> <a href="#">Edit</a> HTTP:80/index.html <ul style="list-style-type: none"><li>Timeout: 5 seconds</li><li>Interval: 30 seconds</li><li>Unhealthy threshold: 2</li><li>Unhealthy threshold: 10</li></ul>	<b>Instances</b> <a href="#">Edit</a> No instances added yet	<b>Attributes</b> <a href="#">Edit</a> <ul style="list-style-type: none"><li>Cross-zone load balancing: On</li><li>Connection draining: On</li><li>Connection draining timeout: 300 seconds</li></ul>	<b>Tags</b> <a href="#">Edit</a> <b>Classic</b> Demo

Cancel **Create load balancer**

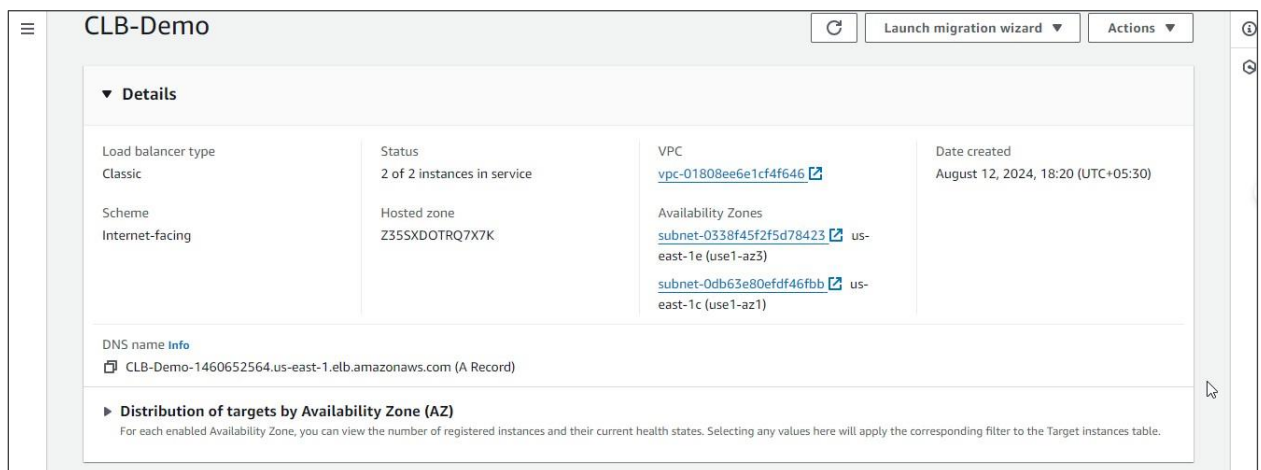
The load balancer has been created successfully.

## Step 4: Deploy the Classic Load Balancer to an EC2 instance

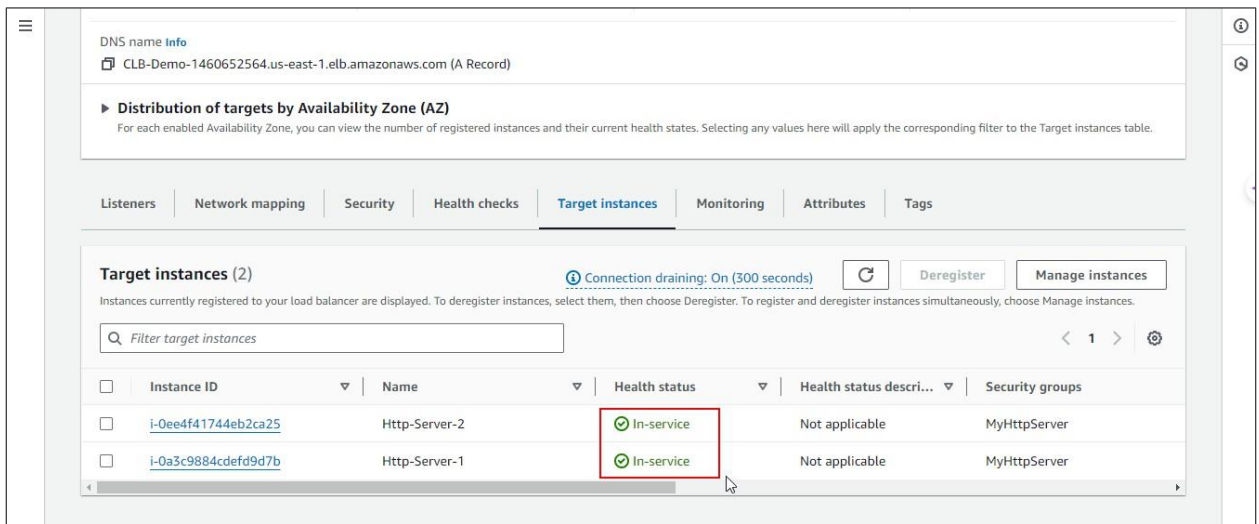
### 1.1 Click on the **CLB-Demo** load balancer



### 1.2 Verify the details



### 1.3 Click on the **Target instances** tab and check the status of both instances



DNS name [Info](#)  
CLB-Demo-1460652564.us-east-1.elb.amazonaws.com (A Record)

► **Distribution of targets by Availability Zone (AZ)**  
For each enabled Availability Zone, you can view the number of registered instances and their current health states. Selecting any values here will apply the corresponding filter to the Target instances table.

Listeners | Network mapping | Security | Health checks | **Target instances** | Monitoring | Attributes | Tags

**Target instances (2)** Connection draining: On (300 seconds) Deregister Manage instances

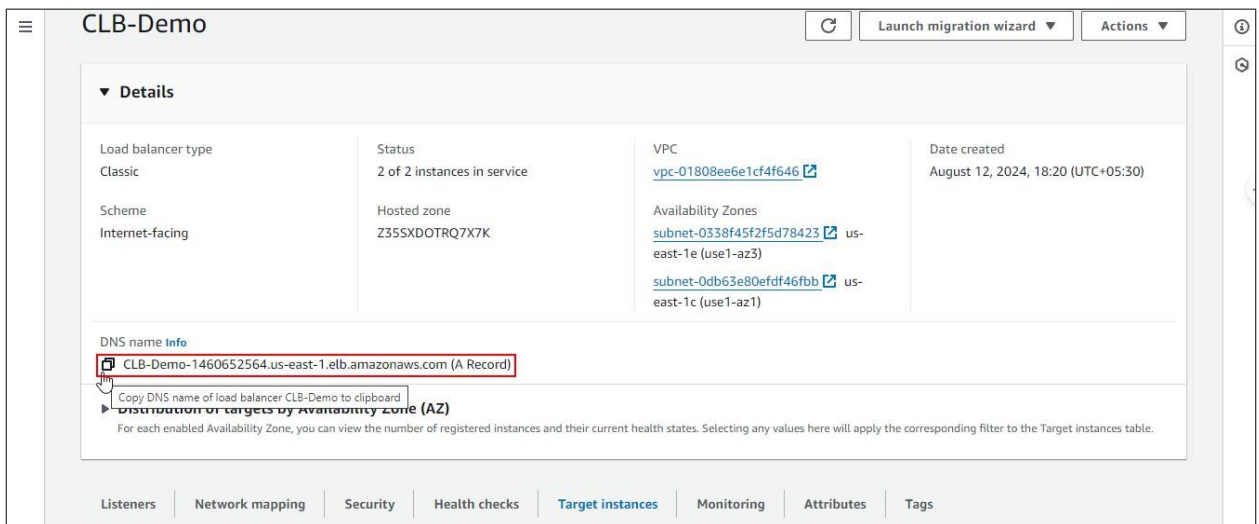
Instances currently registered to your load balancer are displayed. To deregister instances, select them, then choose Deregister. To register and deregister instances simultaneously, choose Manage instances.

Filter target instances

Instance ID	Name	Health status	Health status descri...	Security groups
<a href="#">i-0ee4f41744eb2ca25</a>	Http-Server-2	In-service	Not applicable	MyHttpServer
<a href="#">i-0a3c9884cdefd9d7b</a>	Http-Server-1	In-service	Not applicable	MyHttpServer

The status needs to be **In-service**, which means that both instances are running successfully.

### 1.4 Copy the **DNS name** and paste it into the browser to view the output



CLB-Demo Launch migration wizard Actions

▼ **Details**

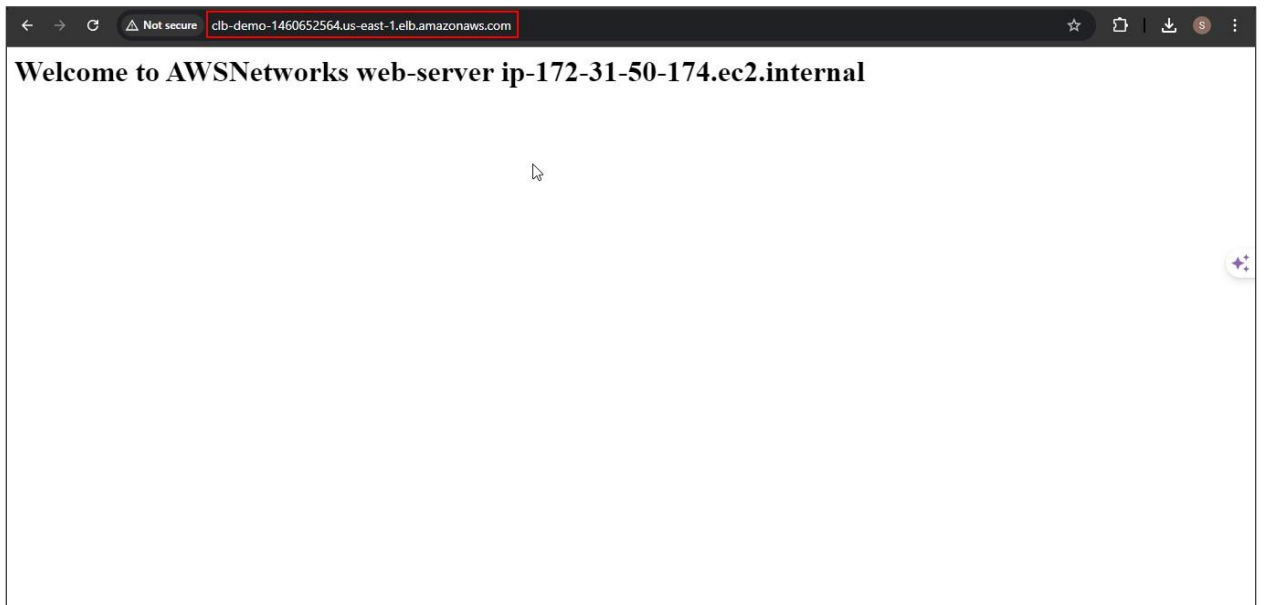
Load balancer type Classic	Status 2 of 2 instances in service	VPC <a href="#">vpc-01808ee6e1cf4f646</a>	Date created August 12, 2024, 18:20 (UTC+05:30)
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones <a href="#">subnet-0338f45f2f5d78423</a> us-east-1e (use1-az3) <a href="#">subnet-0db63e80efdf46fbb</a> us-east-1c (use1-az1)	

DNS name [Info](#)  
[CLB-Demo-1460652564.us-east-1.elb.amazonaws.com \(A Record\)](#)

Copy DNS name of load balancer CLB-Demo to clipboard

► **Distribution of targets by Availability Zone (AZ)**  
For each enabled Availability Zone, you can view the number of registered instances and their current health states. Selecting any values here will apply the corresponding filter to the Target instances table.

Listeners | Network mapping | Security | Health checks | **Target instances** | Monitoring | Attributes | Tags



**Note:** The user data script running on the instances will display a welcome message when accessing the Load Balancer's DNS name in the browser.

By following these steps, you have successfully deployed a Classic Load Balancer and distributed traffic across EC2 instances.