

## Lesson 03 Demo 06

### Using a Classic Load Balancer to Distribute Traffic

**Objective:** To demonstrate the process of creating a Classic Load Balancer in AWS EC2 and deploying it to multiple instances in different availability zones

**Tools required:** AWS Management Console, AWS EC2, and web browser

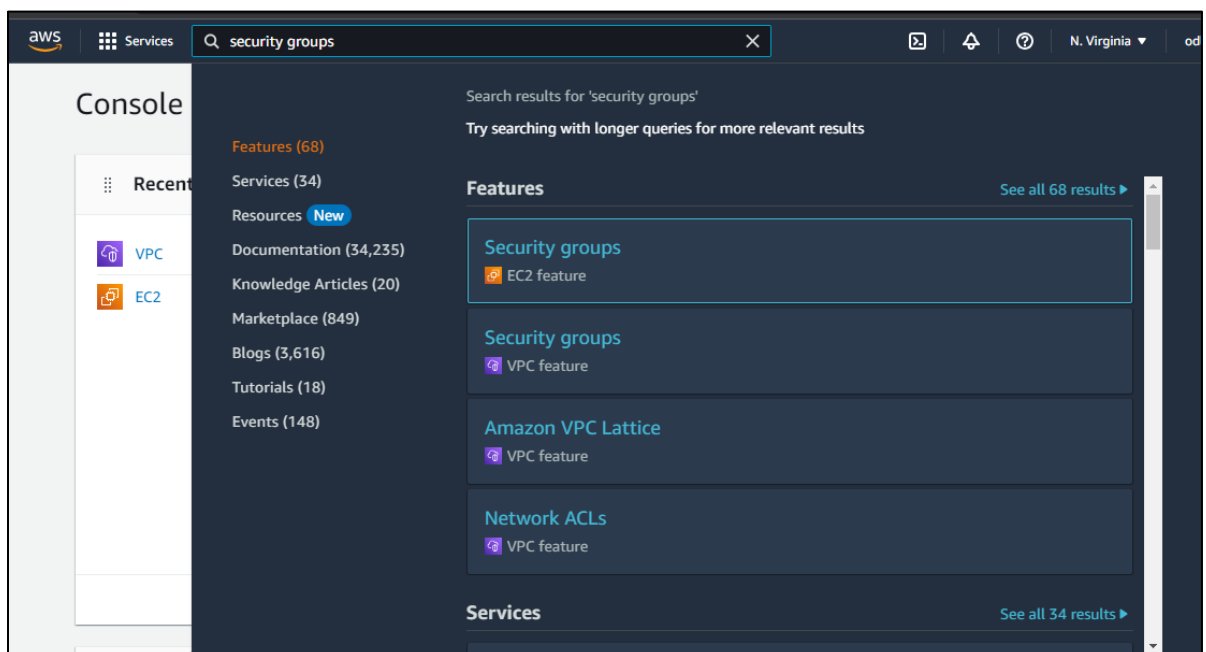
**Prerequisites:** None

Steps to be followed:

1. Create a security group
2. Launch instances with different availability zones
3. Create the Classic Load Balancer
4. Deploy the Classic Load Balancer to an EC2 instance

#### Step 1: Create a security group

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



1.2 Create a security group by filling in the basic details as shown in the screenshot:

- Security group name: **MyHttpServer**
- Description: **Allow SSH and Http traffic**

aws Services Search [Alt+S] N. Virginia odl\_user\_1036229 @ 0259-9511-4484

EC2 > Security Groups > Create security group

### Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name [Info](#)

MyHttpServer

Name cannot be edited after creation.

Description [Info](#)

Allow SSH and Http traffic

VPC [Info](#)

vpc-034cc6a8ed7395da5

1.3 Set the Inbound rules type as **SSH** and **HTTP** with source **Anywhere IPv4**

**Inbound rules** [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
SSH	TCP	22	Anywhere... 0.0.0.0/0		Delete
HTTP	TCP	80	Anywhere... 0.0.0.0/0		Delete

Add rule

1.4 Click on **Create security group**

**Tags - optional**

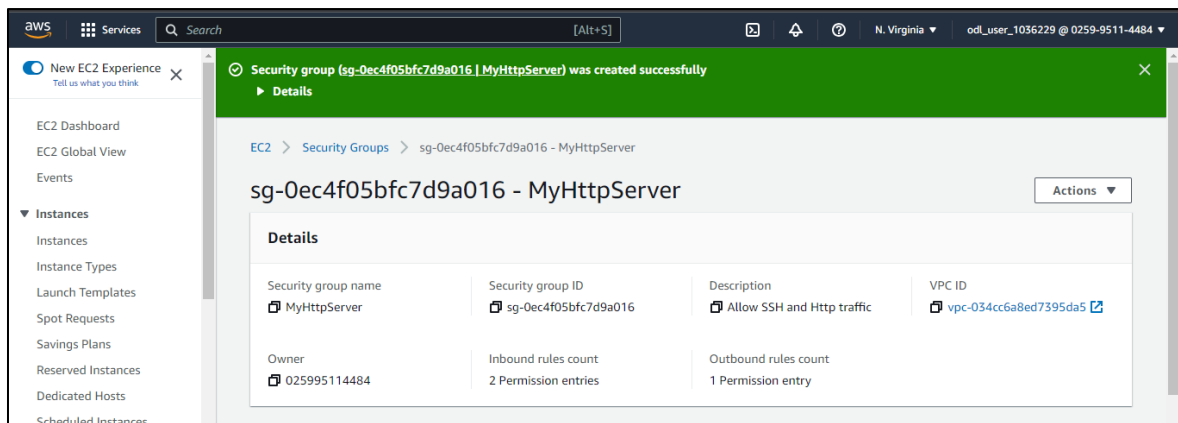
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

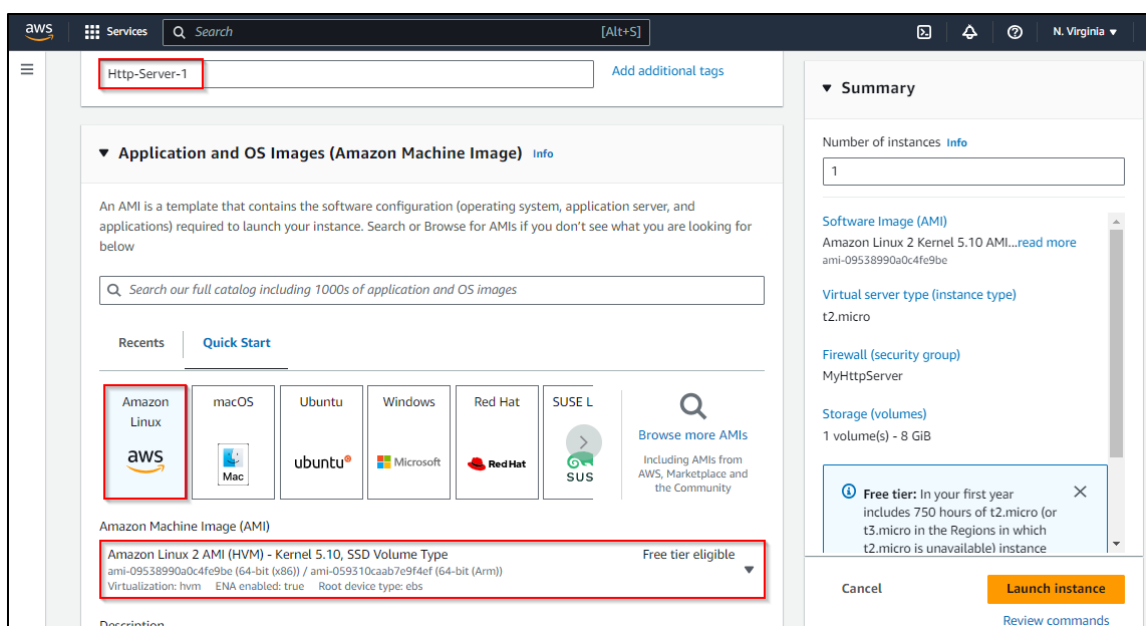
You can add up to 50 more tags.

Cancel Create security group



## Step 2: Launch instances with different availability zones

### 2.1 Enter the Name and tags as **Http-Server-1**, and select **Amazon Linux > Amazon Linux 2 AMI – kernel 5.10, SSD Volume Type**



Launch the first instance by giving it a name and providing the subnet information with a specific availability zone.

## 2.2 Create a new key pair, and name it as **Http-Server-1**

**t2.micro** Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
 On-Demand Windows pricing: 0.0162 USD per Hour  
 On-Demand SUSE pricing: 0.0116 USD per Hour  
 On-Demand RHEL pricing: 0.0716 USD per Hour  
 On-Demand Linux pricing: 0.0116 USD per Hour

**▼ Key pair (login)** [Info](#)

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](#)

## 2.3 Enter the following details as shown in the screenshot for network settings:

Key pair name - *required*

Select ↕ [Create new key pair](#)

**▼ Network settings** [Info](#)

**VPC - *required*** [Info](#)

**vpc-034cc6a8ed7395da5** (default) ↕

**Subnet** [Info](#)

**subnet-09badd7ff8b3a1232** ↕ [Create new subnet](#)

VPC: vpc-034cc6a8ed7395da5 Owner: 025995114484 Availability Zone: us-east-1c  
 IP addresses available: 4091 CIDR: 172.31.16.0/20

**Auto-assign public IP** [Info](#)

Enable ↕

**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.

**▼ Summary**

Number of instances [Info](#)

1

**Software Image (AMI)**

Amazon Linux 2023 AMI 2023.1.2...[read more](#)  
 ami-0f34c5ae932e6f0e4

**Virtual server type (instance type)**

t2.micro

**Firewall (security group)**

MyHttpServer

**Storage (volumes)**

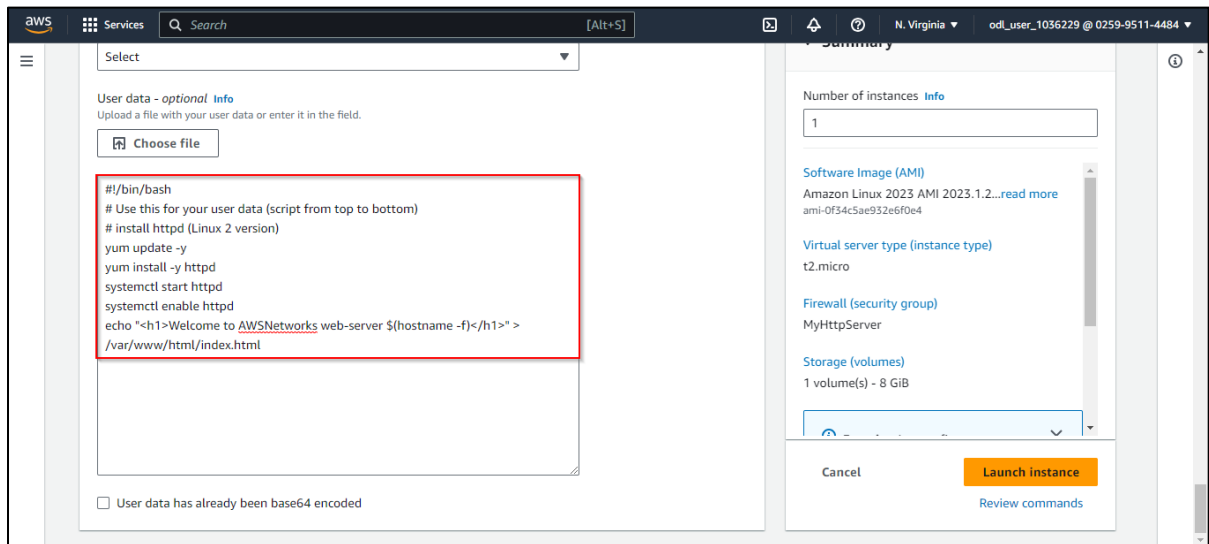
1 volume(s) - 8 GiB

## 2.4 Select the existing security group, and click on the **MyHttpServer**

The screenshot shows the AWS Management Console interface for configuring an EC2 instance. The 'Firewall (security groups)' section is active, showing options to 'Create security group' or 'Select existing security group'. The 'Select existing security group' option is highlighted with a red box. Below this, a list of 'Common security groups' is displayed, with 'MyHttpServer' (VPC: vpc-034cc6a8ed7395da5, sg-0ec4f05bfc7d9a016) selected. The 'default' security group is also listed. The 'Configure storage' section shows a root volume of 8 GiB, gp3 type, and not encrypted. The 'Summary' section on the right shows the instance configuration, including the software image (Amazon Linux 2023 AMI), virtual server type (t2.micro), and storage (1 volume of 8 GiB). The 'Launch instance' button is visible in the bottom right corner.

2.5 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
```



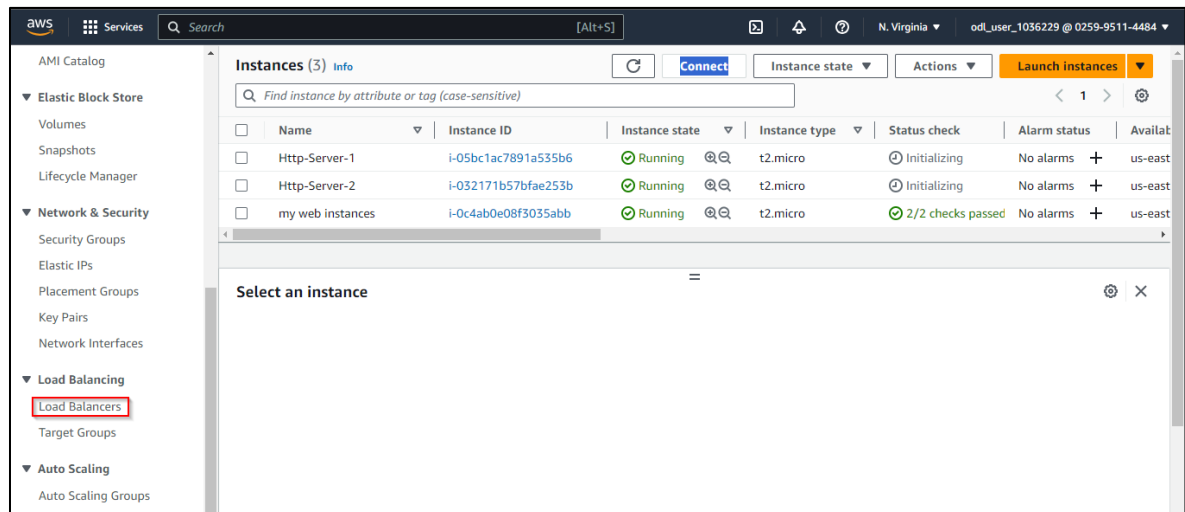
2.6 Repeat the steps to launch the second instance with a different availability zone

Instances (2) <a href="#">Info</a>							
<input type="text" value="Search"/> <span>Refresh</span> <span>Connect</span> <span>Instance state</span> <span>Actions</span> <span>Launch instances</span>							
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Http-Server-2	i-0ad7f5cb194caca4a	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	us-east-1b
<input type="checkbox"/>	Http-Server-1	i-04001409a53e2e490	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	us-east-1c

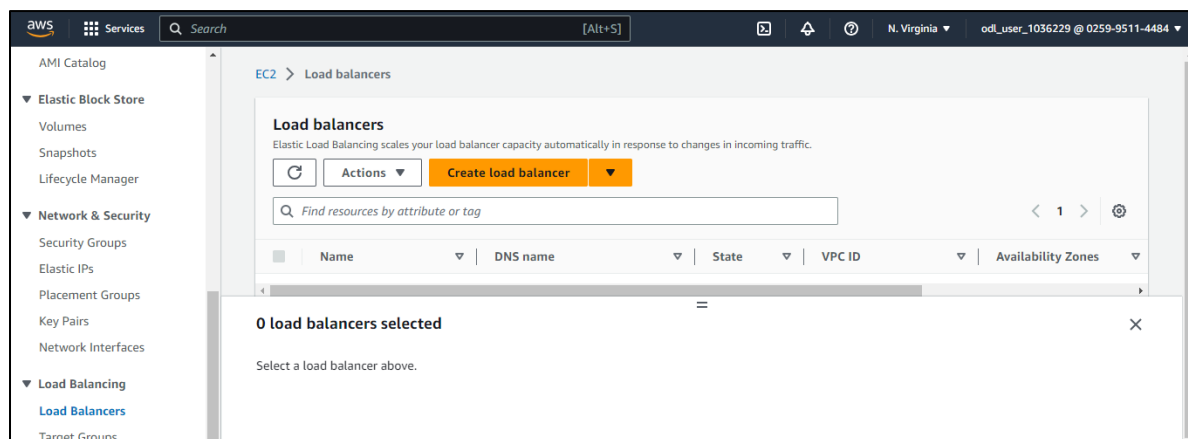
The instances with different availability zones have been launched successfully.

## Step 3: Create the Classic Load Balancer

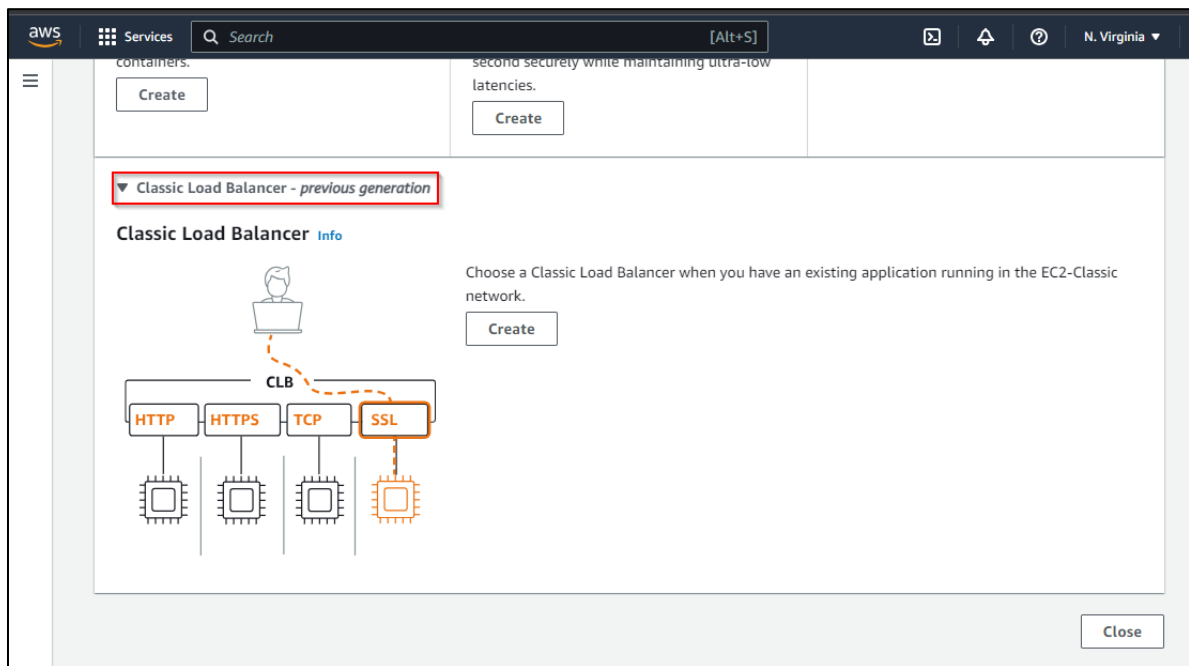
### 3.1 Navigate to the EC2 console, and click on the Load Balancers



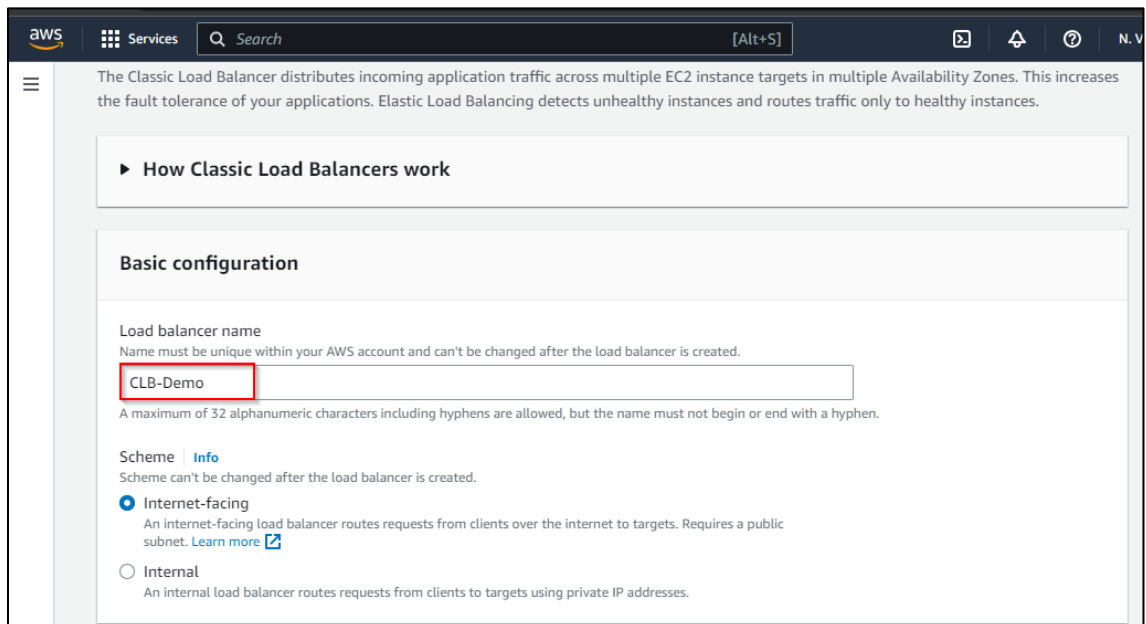
### 3.2 Click on the Create Load Balancer button



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



### 3.4 Provide a Load Balancer name as **CLB-Demo**





### 3.5 Click on the Mappings of **us-east 1c**

**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.

☐ **us-east-1a (use1-az1)**

☐ **us-east-1b (use1-az2)**

☒ **us-east-1c (use1-az4)**

Subnet

subnet-09badd7ff8b3a1232

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

MyHttpServer  
sg-0ec4f05bfc7d9a016    VPC: vpc-034cc6a8ed7395da5

default  
sg-09c5c272829d6bcfe    VPC: vpc-034cc6a8ed7395da5

### 3.7 Change the Response timeout to 5 seconds and the Internal timeout to 30 seconds

#### 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

1-65535

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 Select both instances, and click on **Confirm**

aws

Services

Search

[Alt+S]

N. Virginia

odi\_user\_1036229 @ 0259-9511-4484

#### 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-034cc6a8ed7395da5

Available instances (2/2)

Filter available instances by property or value

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone
<input checked="" type="checkbox"/>	i-05bc1ac7891a535b6	Http-Server-1	Running	MyHttpServer	us-east-1c
<input checked="" type="checkbox"/>	i-032171b57bfae253b	Http-Server-2	Running	MyHttpServer	us-east-1b

Cancel

Confirm

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

**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.

Key: classic Value - optional: Demo [Remove]

[Add new tag]  
You can add up to 49 more tags.

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

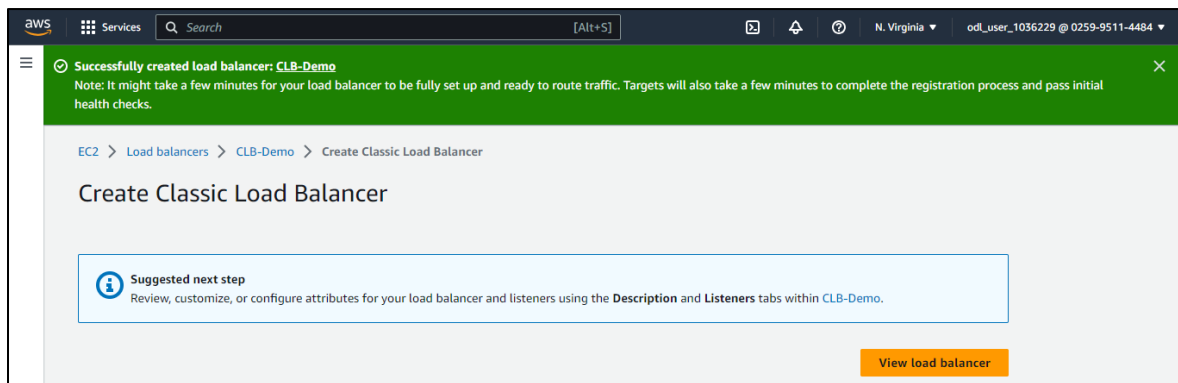
<b>Basic configuration</b> <a href="#">Edit</a> CLB-Demo <ul style="list-style-type: none"> <li>Internet-facing</li> </ul>	<b>Network mapping</b> <a href="#">Edit</a> VPC <a href="#">vpc-034cc6a8ed7395da5</a> Subnet not defined	<b>Security groups</b> <a href="#">Edit</a> <ul style="list-style-type: none"> <li>MyHttpServer <a href="#">sg-0ec4f05bfc7d9a016</a></li> <li>default <a href="#">sg-09c5c272829d6bcfe</a></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> 2 instances added <ul style="list-style-type: none"> <li>1 instance in us-east-1b</li> <li>1 instance in us-east-1c</li> </ul>	<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> classicDemo

### 3.10 Click on **Create load balancer**

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

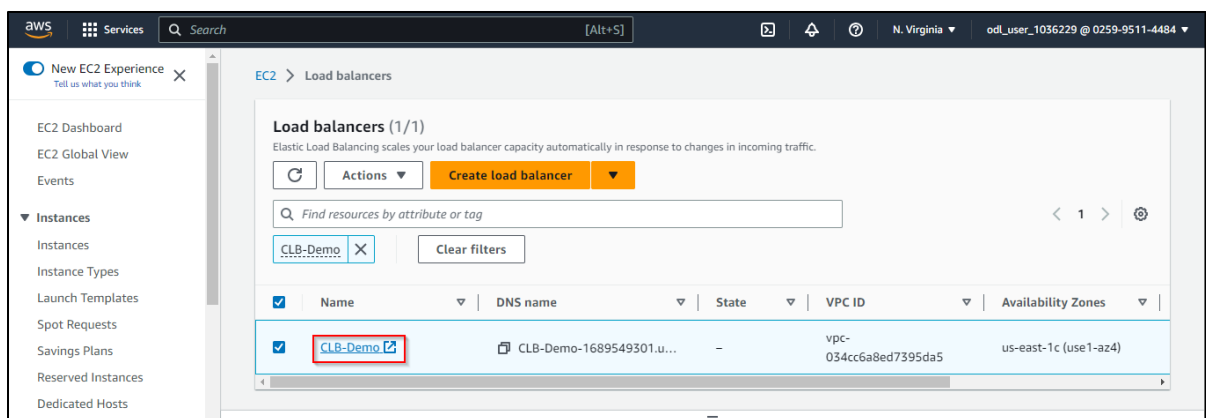
<b>Basic configuration</b> <a href="#">Edit</a> CLB-Demo <ul style="list-style-type: none"> <li>Internet-facing</li> </ul>	<b>Network mapping</b> <a href="#">Edit</a> VPC <a href="#">vpc-034cc6a8ed7395da5</a> Subnet not defined	<b>Security groups</b> <a href="#">Edit</a> <ul style="list-style-type: none"> <li>MyHttpServer <a href="#">sg-0ec4f05bfc7d9a016</a></li> <li>default <a href="#">sg-09c5c272829d6bcfe</a></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> 2 instances added <ul style="list-style-type: none"> <li>1 instance in us-east-1b</li> <li>1 instance in us-east-1c</li> </ul>	<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> classicDemo

Cancel **Create load balancer**

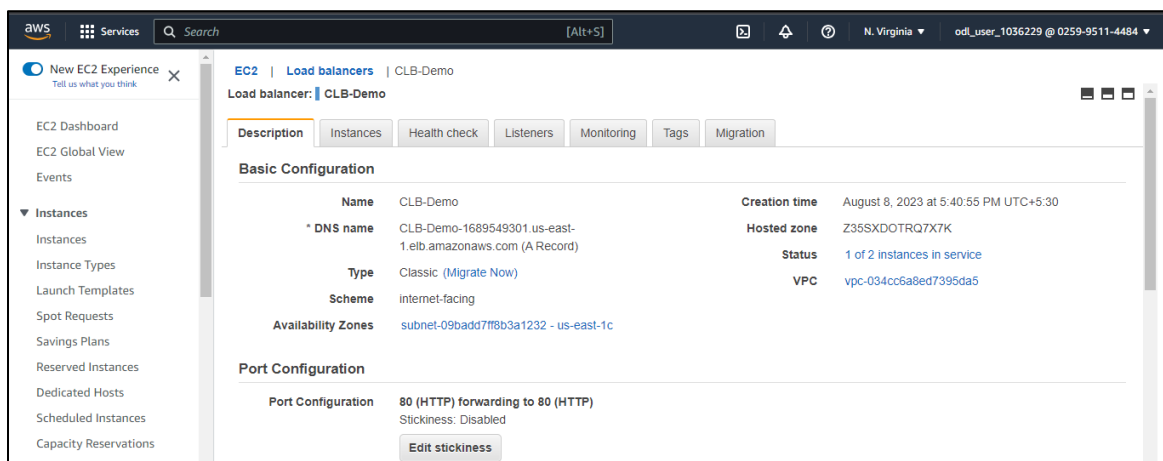


## Step 4: Deploy the Classic Load Balancer to an EC2 instance

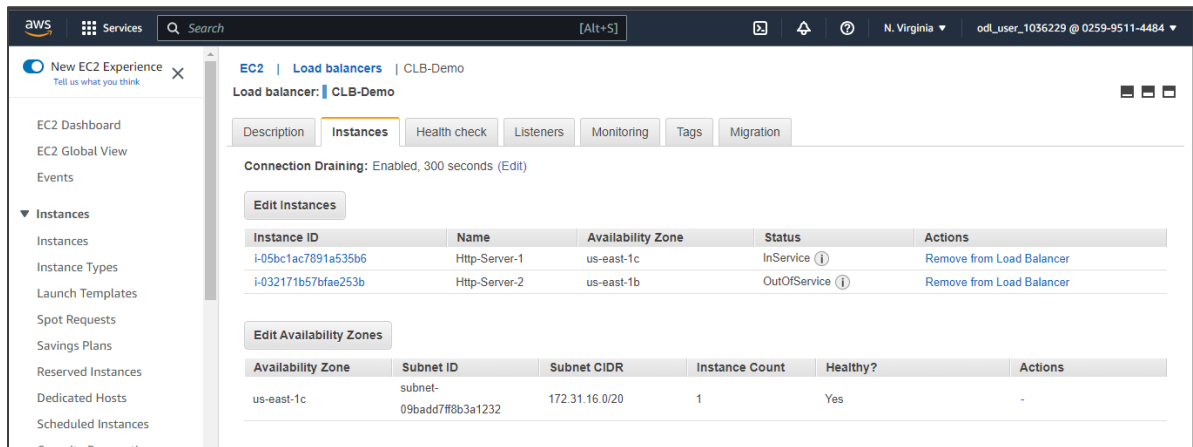
### 4.1 Click on the CLB-Demo load balancer



### 4.2 Navigate to the Load Balancer created in Step 3, and check the details under the Description tab

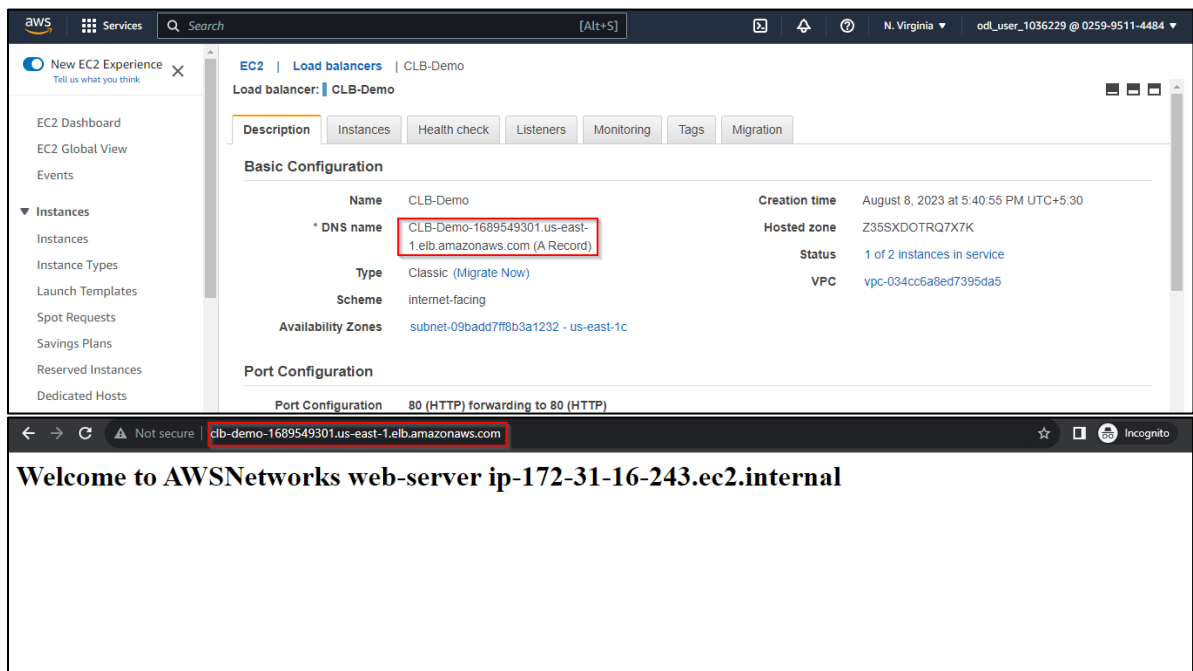


#### 4.3 Click on the **Instances** tab, and check the status of both instances



The status needs to be **InService** which means both the instances are running successfully.

#### 4.4 Copy the **DNS name** and paste it in browser to see the output



**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 will be able to successfully deploy the Classic Load Balancer to an EC2 instance while the user data script is running.