

Task 2- Create 3 instances , install nginx and apply ALB

ALB – ALB abbreviates as Application load balancer.

Load balancer is used to manage the incoming traffic on servers.

For creating Application load balancer the following steps need to be followed:-

- First, we need to create security group
- Then we need to create 3 instances with 3 different availability zones
- Install nginx in 3 instances
- Create target groups and link it with the 3 created instances
- Attach the created target group with the load balancer
- This is done because of balancing the traffic or data between 3 instances
- Let us do this in detail

Select security groups under network and security.

▼ Network & Security

Security Groups

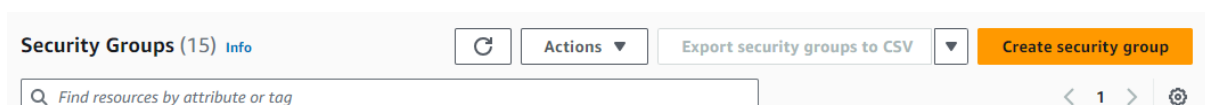
Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Click on create security group.



Give security name and give description as allow.

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](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Now click on add rule in inbound rules.

Inbound rules

Info

This security group has no inbound rules.

Add rule

Type – HTTP, Source- 0.0.0.0/0

Inbound rules

Info

Type	Protocol	Port range	Source	Description - optional	
HTTP	TCP	80	Any...	0.0.0.0/0	Delete
0.0.0.0/0					

Add rule

Add rule

Type- all traffic, source – 0.0.0.0/0

Inbound rules

Info

Type	Protocol	Port range	Source	Description - optional	
HTTP	TCP	80	Any...	0.0.0.0/0	Delete
0.0.0.0/0					
All traffic	All	All	Any...	0.0.0.0/0	Delete
0.0.0.0/0					

Add rule

Click on create security group.

You can add up to 50 more tags

CancelCreate security group

The security group is created now and it shows as follows:-

Security group (sg-0988333ac302afbc8 | Load) was created successfully

Details

EC2 > Security Groups > sg-0988333ac302afbc8 - Load

sg-0988333ac302afbc8 - Load

Actions

Details

Security group name	Security group ID	Description	VPC ID
Load	sg-0988333ac302afbc8	allow	vpc-064b83e84c2b968f5
Owner	Inbound rules count	Outbound rules count	
471112914581	2 Permission entries	1 Permission entry	

Inbound rules

Outbound rules

Tags

After creating the security group we need to create 3 instances with 3 different availability zones.

Click on ec2 dashboard and then click on instances.

The screenshot shows the AWS EC2 Dashboard. On the left is a navigation menu with options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Console-to-Code', and 'Instances'. The 'Instances' section is expanded. The main area is titled 'Resources' and shows a table of EC2 resources in the 'US East (N. Virginia) Region'. The table lists various resources and their counts.

Resources	
Instances (running)	1
Dedicated Hosts	0
Instances	1
Load balancers	0
Security groups	15
Volumes	1
Auto Scaling Groups	0
Elastic IPs	0
Key pairs	8
Placement groups	0
Snapshots	0

Click on launch instances.

The screenshot shows the 'Instances (1)' header in the AWS console. It includes a search bar, a 'Connect' button, a dropdown for 'Instance state', and a dropdown for 'Actions'. A prominent orange 'Launch instances' button is visible on the right.

Give instance name

The screenshot shows the 'Launch an instance' page. It has a breadcrumb trail 'EC2 > Instances > Launch an instance'. Below the title, there's a description of Amazon EC2. The 'Name and tags' section is active, showing a text input field with 'Load-1' and a button to 'Add additional tags'.

Select amazon linux

The screenshot shows the 'Quick Start' section for selecting an AMI. It features a row of six cards for different operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. The 'Amazon Linux' card is highlighted with a blue border. To the right of these cards is a search icon and a link to 'Browse more AMIs'.

Click on create new key pair

▼ **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*

Select



[Create new key pair](#)

Give the key pair name and then click on create new pair

Key pair name

Key pairs allow you to connect to your instance securely.

load

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type



RSA

RSA encrypted private and public key pair



ED25519

ED25519 encrypted private and public key pair

Private key file format



.pem

For use with OpenSSH



.ppk

For use with PuTTY



When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Cancel

Create key pair

Edit network settings

▼ Network settings

Info

Edit

Network | Info

vpc-064b83e84c2b968f5

Subnet | Info

No preference (Default subnet in any availability zone)

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

We'll create a new security group called 'launch-wizard-8' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Choose subnet in the availability zone in us-east-1a > enable > click on select existing security group(select the security which we have created) > select security group.

VPC - required | Info

vpc-064b83e84c2b968f5
172.31.0.0/16

(default) ▼



Subnet | Info

subnet-0eb78bf9a19cf3c68

VPC: vpc-064b83e84c2b968f5 Owner: 471112914581

Availability Zone: us-east-1a IP addresses available: 4090 CIDR: 172.31.32.0/20



Create new subnet [↗](#)

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 ▼

Load sg-0988333ac302afbc8 ✕
VPC: vpc-064b83e84c2b968f5



Compare security
group rules

Click on launch instance.

⌚ Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Edit

▶ **Advanced details** Info

firewall (security group)
Load

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

Cancel

Launch instance

Review commands

The instance is created successfully.

EC2 > Instances > Launch an instance

🟢 Success
Successfully initiated launch of instance (i-06f0d2d9ae864f95d)

▶ Launch log

Now again launch new instance

Instances (2) Info

🔄 Connect Instance state ▾ Actions ▾ Launch instances ▾

🔍 Find Instance by attribute or tag (case-sensitive) All states ▾ < 1 > ⚙️

<input type="checkbox"/>	Name ↗	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability
<input type="checkbox"/>	Load-1	i-06f0d2d9ae864f95d	🟢 Running 🔍 🔍	t2.micro	🕒 Initializing	View alarms +	us-east-1a

Give name to instance

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

Load-2

Add additional tags

Select amazon linux

Quick Start

Amazon Linux
aws

macOS
Mac

Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Li
SUS

🔍

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Give key pair name as same as the key pair name for the first instance .

▼

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*

load ▼

[Create new key pair](#)

Edit network settings as the instance 1 but create the availability zone in us-east-1b

VPC - *required*

Info

vpc-064b83e84c2b968f5

(default) ▼

Subnet

Info

subnet-099c01316e14fc969

VPC: vpc-064b83e84c2b968f5

Owner: 471112914581

Availability Zone: us-east-1b

IP addresses available: 4090

CIDR: 172.31.0.0/20

▼

[Create new subnet](#)

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 ▼

Load

sg-0988333ac302afbc8 ✕

VPC: vpc-064b83e84c2b968f5

[Compare security group rules](#)

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

And then click on launch instance

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Edit

►

Advanced details

Info

Firewall (security group)

Load

Storage (volumes)

1 volume(s) - 8 GiB

▼


Cancel

Launch instance

Review commands

It shows as the instance initiated.

EC2 > Instances > Launch an instance

 **Success**
Successfully initiated launch of instance (i-011e907242ecfe2f2)

Now for creating the 3rd instance click on launch instances

Instances (3) Info							
<input type="button" value="Refresh"/> <input type="button" value="Connect"/> <input type="button" value="Instance state"/> <input type="button" value="Actions"/>				<input type="button" value="Launch instances"/>			
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>				<input type="button" value="All states"/> < 1 >			
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	Load-1	i-06f0d2d9ae864f95d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a
<input type="checkbox"/>	Sweety1	i-00abb60fbc100431c	Terminated	t2.micro	-	View alarms +	us-east-1c
<input type="checkbox"/>	Load-2	i-011e907242ecfe2f2	Running	t2.micro	Initializing	View alarms +	us-east-1b

Give name for instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.


Name and tags [Info](#)


Name

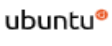
[Add additional tags](#)


select amazon linux


Quick Start



Amazon Linux



macOS


Ubuntu


Windows


Red Hat


SUSE Linux


[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Give the key pair as same as the key pair given for instance 1 and 2

▼ 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*

[Create new key pair](#)

Edit network settings as same as the before instances but create the availability zone in us-east-1c and then click on launch instance.

VPC - required [Info](#)

vpc-064b83e84c2b968f5 (default) ↕

Subnet [Info](#)

subnet-0cd01b6949c9db2af
VPC: vpc-064b83e84c2b968f5 Owner: 471112914581 Availability Zone: us-east-1c IP addresses available: 4090 CIDR: 172.31.80.0/20

Create new subnet ↗

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

Load sg-0988333ac302afbc8 ✕
VPC: vpc-064b83e84c2b968f5

Compare security group rules ↕

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...[read more](#)
ami-06c68f701d8090592

Virtual server type (instance type)
t2.micro

Firewall (security group)
Load

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

Cancel

Launch instance

The instance is initiated.



Go to instances > select on the first created instance > click on instance ID

Instances (1/4) [Info](#)

↕
Connect
Instance state ▼
Actions ▼
Launch instances ▼

Find Instance by attribute or tag (case-sensitive)
All states ▼

	Name ↗	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/>	Load-1	i-06f0d2d9ae864f95d	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
<input type="checkbox"/>	Sweety1	i-00abb60fbc100431c	Terminated	t2.micro	-	View alarms	us-east-1c
<input type="checkbox"/>	Load-3	i-0697dd2fe6676d6cc	Running	t2.micro	Initializing	View alarms	us-east-1c
<input type="checkbox"/>	Load-2	i-011e907242ecfe2f2	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b

Click on connect

[EC2](#) > [Instances](#) > i-06f0d2d9ae864f95d




Instance summary for i-06f0d2d9ae864f95d (Load-1) [Info](#)

↕
Connect
Instance state ▼
Actions ▼

Updated less than a minute ago

Instance ID i-06f0d2d9ae864f95d (Load-1)	Public IPv4 address 3.90.223.163 open address	Private IPv4 addresses 172.31.43.93
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-90-223-163.compute-1.amazonaws.com open address

Click on EC2 instance connect and then click on connect

EC2 Instance Connect	Session Manager	SSH client	EC2 serial console
<p>Instance ID  i-06f0d2d9ae864f95d (Load-1)</p> <p>Connection Type</p> <div> <div> <input checked="" type="radio"/> Connect using EC2 Instance Connect Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address. </div> <div> <input type="radio"/> Connect using EC2 Instance Connect Endpoint Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint. </div> </div> <p>Public IP address  3.90.223.163</p> <p>Username Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.</p> <div> <input type="text" value="ec2-user"/>  </div> <div> <p>Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.</p> </div>			
<p>Cancel</p>		<p>Connect</p>	

Instance will be connected to amazon linux.

```

aws  Services  Search  [Alt+S]  N. Virginia  VadiLakshmiSweth
#
#####
~\  #####
~~  \###|
~~  \#/
~~  V~' '->
~~
~~~
~~~.  _
~~~  _/m/'
ec2-user@ip-172-31-43-93 ~]$

```

Use command `sudo -i` for connecting it to the root user.

```

#
#####
~\  #####
~~  \###|
~~  \#/
~~  V~' '->
~~
~~~
~~~.  _
~~~  _/m/'
[ec2-user@ip-172-31-43-93 ~]$ sudo -i
[root@ip-172-31-43-93 ~]#

```

Update and install nginx by using the commands-

`yum update -y`

`yum install nginx -y`

```
[ec2-user@ip-172-31-43-93 ~]$ sudo -i
[root@ip-172-31-43-93 ~]# yum update -y
Last metadata expiration check: 0:08:30 ago on Wed Jul 3 17:10:39 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-43-93 ~]# yum install nginx -y
```

nginx will start to install now

```
Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/7
Installing      : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/7
Installing      : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 2/7
Installing      : libunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Installing      : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 4/7
Installing      : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Installing      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 6/7
Installing      : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Verifying      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 1/7
Verifying      : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 2/7
Verifying      : libunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Verifying      : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 4/7
Verifying      : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Verifying      : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 6/7
Verifying      : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 7/7

Installed:
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 libunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1:1.24.0-1.amzn2023.0.2.x86_64      nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64  nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

complete!
[root@ip-172-31-43-93 ~]#
```

search for the path of nginx by-

cd /usr/share/nginx/html

```
[root@ip-172-31-43-93 ~]# cd /usr/share/nginx/html
[root@ip-172-31-43-93 html]#
```

remove the file index.html by-

rm -rf index.html

and then create file by-

vi index.html

```
[root@ip-172-31-43-93 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-43-93 html]# rm -rf index.html
[root@ip-172-31-43-93 html]# vi index.html
```

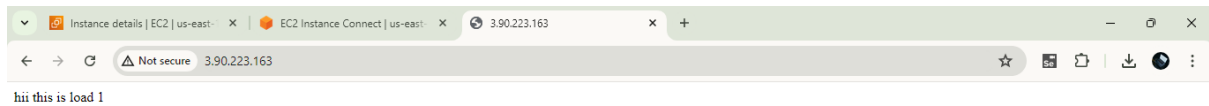
Insert and then add data as follows-



```
systemctl restart nginx
```

go back to instance ID and copy the public IPv4 address

The data will be shown which is inserted in the file index.html



Do the same for instance 2 and instance 3

Click on the instance ID of 2nd instance

Instances (1/4) Info								Refresh Connect Instance state ▼ Actions ▼ Launch instances ▼	
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>								All states ▼ < 1 > Settings	
<input type="checkbox"/>	Name ↗	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability		
<input type="checkbox"/>	Load-1	i-06f0d2d9ae864f95d	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	us-east-1a		
<input type="checkbox"/>	Sweety1	i-00abb60fbc100431c	Terminated 🔍 🔍	t2.micro	-	View alarms +	us-east-1c		
<input type="checkbox"/>	Load-3	i-0697dd2fe6676d6cc	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	us-east-1c		
<input checked="" type="checkbox"/>	Load-2	i-011e907242ecfe2f2	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	us-east-1b		


Click on connect

[EC2](#) > [Instances](#) > i-011e907242ecfe2f2

Instance summary for i-011e907242ecfe2f2 (Load-2) [Info](#)

Updated less than a minute ago

Instance ID

 i-011e907242ecfe2f2 (Load-2)



IPv6 address

-


Hostname type

IP name: ip-172-31-0-126.ec2.internal


Public IPv4 address

 34.201.14.180 | [open address](#) 


Instance state

 **Running**



Private IP DNS name (IPv4 only)

 ip-172-31-0-126.ec2.internal

Private IPv4 addresses

 172.31.0.126

Public IPv4 DNS

 ec2-34-201-14-180.compute-1.amazonaws.com | [open address](#) 

Click on EC2 instance connect and click on connect


EC2 Instance Connect

Session Manager

SSH client

EC2 serial console

Instance ID


 i-011e907242ecfe2f2 (Load-2)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.


☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.


Public IP address


 34.201.14.180

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

 ec2-user



 **Note:** In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

It will take to aws linux.

```

#_
,' \_ ##### Amazon Linux 2023
~~~ \_ ##### \
~~~ \_ ##### |
~~~ \_ # / https://aws.amazon.com/linux/amazon-linux-2023
~~~ v ~ ' ' ->
~~~~
~~~~ . _ . / /
~~~~ _ / _ / /
~~~~ _ / m / ' _ /
[ec2-user@ip-172-31-0-126 ~]$
```

For connecting to root user use command `sudo -i`

Install nginx

```
#  
~\_#### Amazon Linux 2023  
~~\_#####\  
~~\_###|  
~~\_#/ https://aws.amazon.com/linux/amazon-linux-2023  
~~V~' '->  
~~~~  
~~.-.  
/_m/' -
```

Last login: Wed Jul 3 17:27:31 2024 from 18.206.107.28
[ec2-user@ip-172-31-0-126 ~]\$ sudo -i
[root@ip-172-31-0-126 ~]# yum update -y
Last metadata expiration check: 0:14:20 ago on Wed Jul 3 17:14:19 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-0-126 ~]# yum install nginx -y

It will start installing nginx

```
Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/7
Installing      : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/7
Installing      : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 2/7
Installing      : libunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Installing      : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 4/7
Installing      : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Installing      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 6/7
Installing      : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Verifying       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 1/7
Verifying       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 2/7
Verifying       : libunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Verifying       : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 4/7
Verifying       : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 5/7
Verifying       : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 6/7
Verifying       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 7/7

Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 libunwind-1.4.0-5.amzn2023.0.2.x86_64
  nginx-1:1.24.0-1.amzn2023.0.2.x86_64          nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64  nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-172-31-0-126 ~]#
```

Check for nginx path

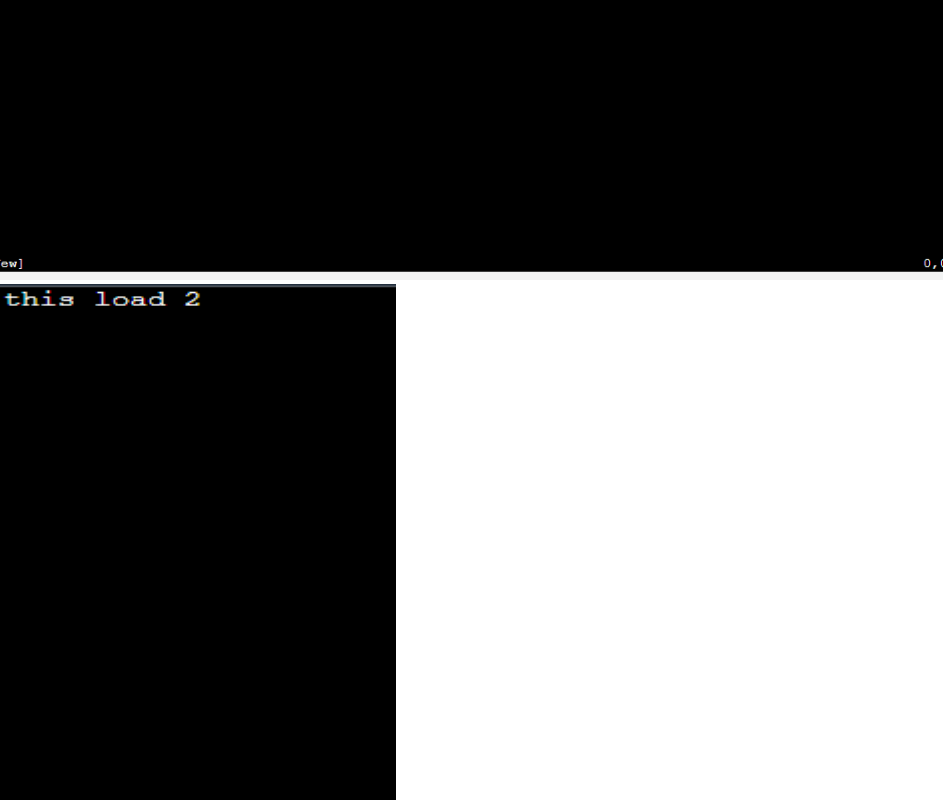
Remove index.html file

```
[root@ip-172-31-0-126 ~]# cd /usr/share/nginx/html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]# rm -rf index.html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]#
```

Create a file named with index.html

```
[root@ip-172-31-0-126 html]# vi index.html
```

Insert data and save the file → esc + shift + : wq



The screenshot shows a terminal window with a black background. At the top, there is a status bar with the text "index.html" [New] on the left, "0,0-1" in the center, and "All" on the right. Below the status bar, the command prompt "hii hi this load 2" is visible. The prompt is followed by a series of tilde characters (~) on subsequent lines, indicating a multi-line command or a list of options. The terminal window is titled "index.html" [New] in the top left corner.

Restart nginx now

```
[root@ip-172-31-0-126 ~]# cd /usr/share/nginx/html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]# rm -rf index.html
[root@ip-172-31-0-126 html]# ls
404.html  50x.html  icons  nginx-logo.png  poweredby.png
[root@ip-172-31-0-126 html]# vi index.html
[root@ip-172-31-0-126 html]# systemctl restart nginx
[root@ip-172-31-0-126 html]#
```


Go to instance ID and copy the Public IPv4 address

EC2 > Instances > i-011e907242ecfe2f2

Instance summary for i-011e907242ecfe2f2 (Load-2) Info Refresh Connect Instance state ▼ Actions ▼

Updated 6 minutes ago

Instance ID i-011e907242ecfe2f2 (Load-2)	Public IPv4 address 34.201.14.180 open address	Private IPv4 addresses 172.31.0.126
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-34-201-14-180.compute-1.amazonaws.com open address

Go to chrome and paste the IPv4 address

Instance details | EC2 | us-east-1 x New Tab x +

34.201.14.180

34.201.14.180

34.201.14.180 - Google Search

The data will be shown which is inserted in the file index.html

Instance details | EC2 | us-east-1 x 34.201.14.180 x +

Not secure 34.201.14.180

hii hi this load 2

Click on the instance ID of 3rd instance

Instances (1/4) Info Refresh Connect Instance state ▼ Actions ▼ Launch instances ▼

Find Instance by attribute or tag (case-sensitive) All states < 1 > ⚙

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	Load-1	i-06f0d2d9ae864f95d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a
<input type="checkbox"/>	Sweet1	i-00abb60fbc100431c	Terminated	t2.micro	-	View alarms +	us-east-1c
<input checked="" type="checkbox"/>	Load-3	i-0697dd2fe6676d6cc	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c
<input type="checkbox"/>	Load-2	i-011e907242ecfe2f2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b

Click on connect



EC2 > Instances > i-0697dd2fe6676d6cc

Instance summary for i-0697dd2fe6676d6cc (Load-3) Info Refresh Connect Instance state ▼ Actions ▼

Updated less than a minute ago

Instance ID i-0697dd2fe6676d6cc (Load-3)	Public IPv4 address 44.212.15.84 open address	Private IPv4 addresses 172.31.81.174
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-44-212-15-84.compute-1.amazonaws.com open address

Click on EC2 instance connect and click on connect

EC2 Instance Connect	Session Manager	SSH client	EC2 serial console
<p>Instance ID</p> <p> i-0697dd2fe6676d6cc (Load-3)</p> <p>Connection Type</p> <div><p><input checked="" type="radio"/> Connect using EC2 Instance Connect</p><p>Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.</p></div> <div><p><input type="radio"/> Connect using EC2 Instance Connect Endpoint</p><p>Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.</p></div> <p>Public IP address</p> <p> 44.212.15.84</p> <p>Username</p> <p>Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.</p> <div><input type="text" value="ec2-user"/></div> <div><p>Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.</p></div> <div><div>Cancel</div><div>Connect</div></div>			

It will take to aws linux

```
#_
~\##### Amazon Linux 2023
~~\#####\
~~\####|
~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~V~'-'>
~~~~
~~.-.-
~/m/'-
```

[ec2-user@ip-172-31-81-174 ~]\$

```
#  
~\### Amazon Linux 2023  
~~\#####  
~~\###|  
~~\#/ https://aws.amazon.com/linux/amazon-linux-2023  
~~V~' '~>  
~~~~  
~~.-.  
~-/_/  
~/m/'
```

```
[ec2-user@ip-172-31-81-174 ~]$ sudo -i  
[root@ip-172-31-81-174 ~]# yum update -y  
Last metadata expiration check: 0:20:44 ago on Wed Jul 3 17:17:01 2024.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[root@ip-172-31-81-174 ~]# yum install nginx -y
```

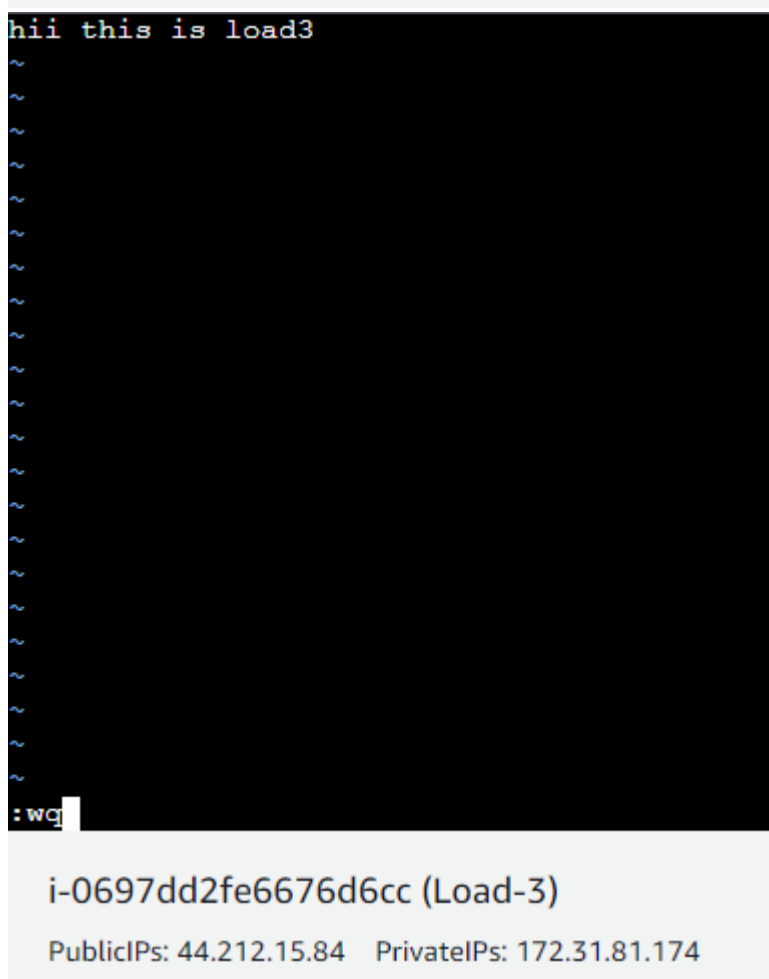
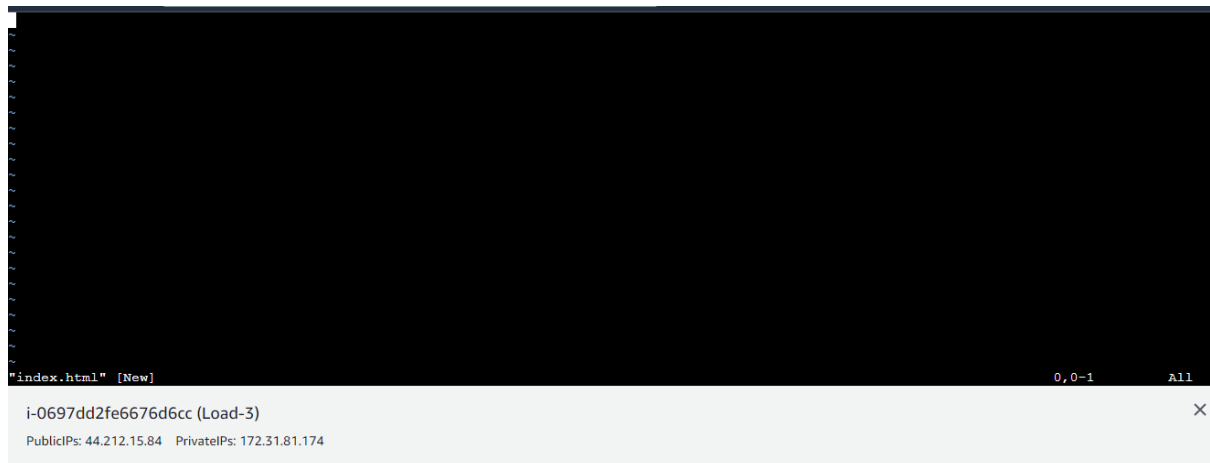
It will start installing nginx

Check for nginx path

Create a file named with index.html

```
[root@ip-172-31-81-174 ~]# cd /usr/share/nginx/html
[root@ip-172-31-81-174 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-172-31-81-174 html]# rm -rf index.html
[root@ip-172-31-81-174 html]# ls
404.html  50x.html  icons  nginx-logo.png  poweredby.png
[root@ip-172-31-81-174 html]# vi index.html
```

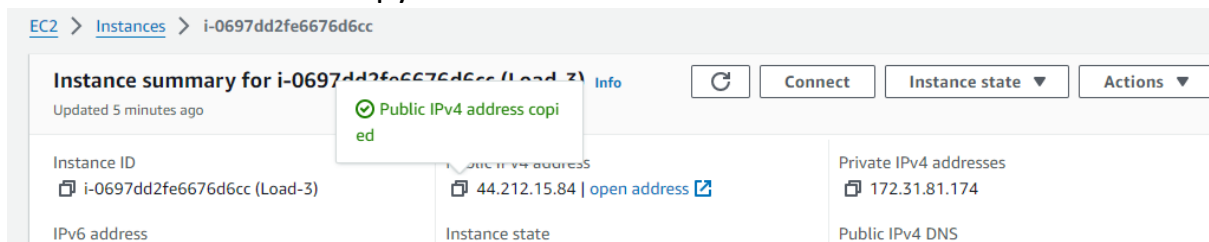
Insert data and save the file → esc + shift + : wq



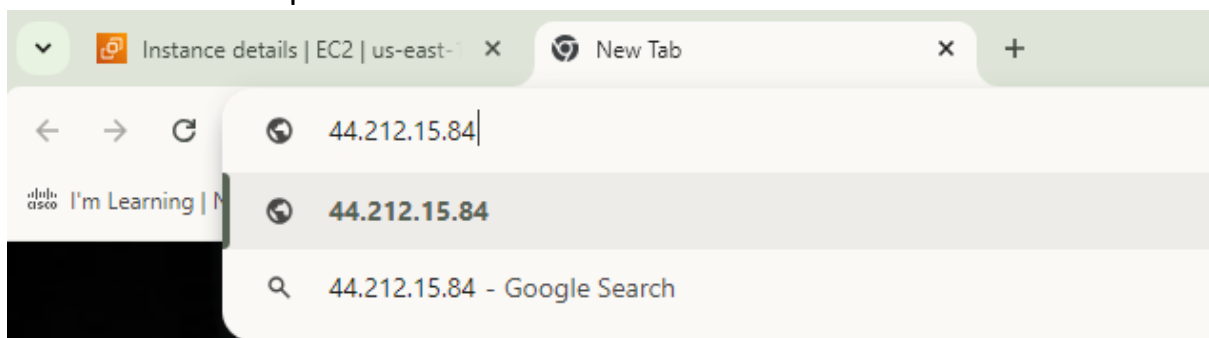
Restart nginx now

```
[root@ip-172-31-81-174 ~]# cd /usr/share/nginx/html
[root@ip-172-31-81-174 html]# ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png
[root@ip-172-31-81-174 html]# rm -rf index.html
[root@ip-172-31-81-174 html]# ls
404.html 50x.html icons nginx-logo.png poweredby.png
[root@ip-172-31-81-174 html]# vi index.html
[root@ip-172-31-81-174 html]# systemctl restart nginx
[root@ip-172-31-81-174 html]#
```

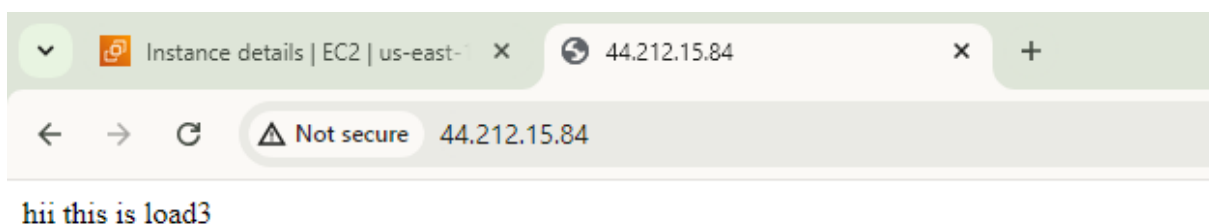
Go to instance ID and copy the Public IPv4 address



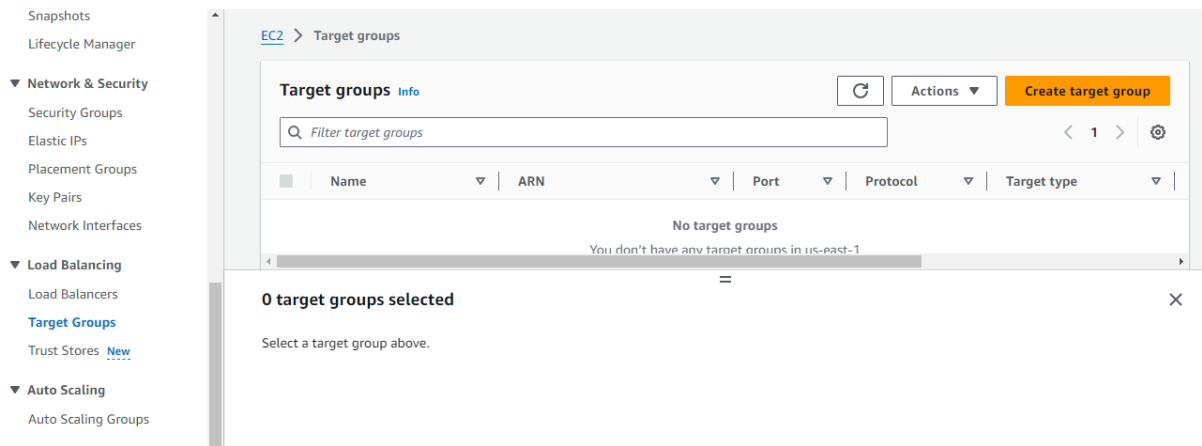
Go to chrome and paste the IPv4 address



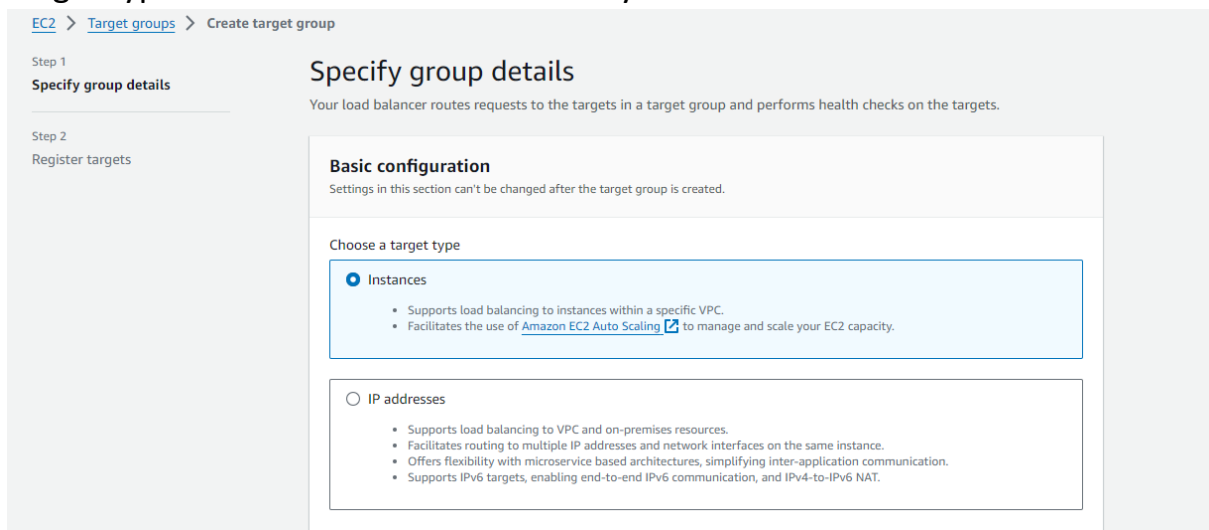
The data will be shown which is inserted in the file index.html



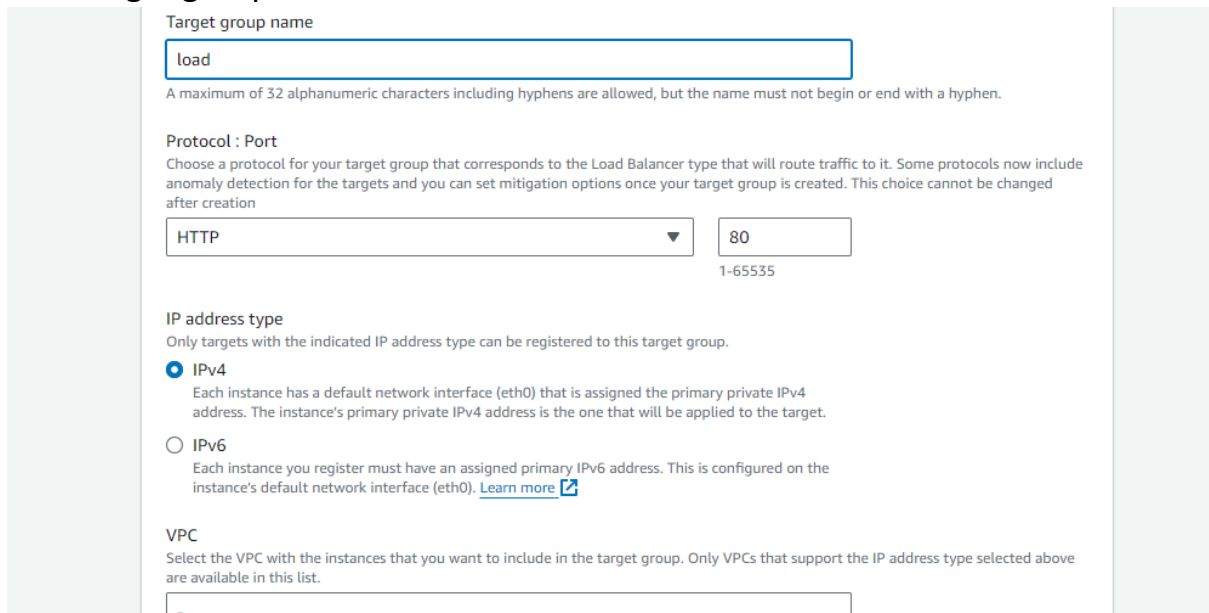
Click on target groups which is under load balancing option and then click on create target group.



Target type will be chosen as instances by default



Give target group name



and then click on next

ⓘ Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

► **Tags - optional**

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel Next

Select all the three 3 instances and click on include as pending below

Available instances (5/5)

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups
<input checked="" type="checkbox"/>	i-0697dd2fe6676d6cc	Load-3	Running	Load
<input checked="" type="checkbox"/>	i-011e907242ecfe2f2	Load-2	Running	Load
<input checked="" type="checkbox"/>	i-06f0d2d9ae864f95d	Load-1	Running	Load

3 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

1-65535 (separate multiple ports with commas)

Include as pending below

It will as follows and click on create target group

Review targets

Targets (3)

Remove all pending

☐ Show only pending

< 1 >


⚙

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 add
i-0697dd2fe6676d6cc	Load-3	80	Running	Load	us-east-1c	172.31.81.174
i-011e907242ecfe2f2	Load-2	80	Running	Load	us-east-1b	172.31.0.126
i-06f0d2d9ae864f95d	Load-1	80	Running	Load	us-east-1a	172.31.43.93

3 pending

Cancel Previous Create target group


Now the target group is created.



 Successfully created the target group: **load**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the **Targets** tab.

[EC2](#) > [Target groups](#) > load






load

Actions ▾

Details
 `arn:aws:elasticloadbalancing:us-east-1:471112914581:targetgroup/load/1fc256cafc21ccc`

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-064b83e84c2b968f5 
IP address type	Load balancer		
IPv4	 None associated		

Now go to load balancer which is under load balancing and then click on create load balancer.

 Services [Alt+S]     N. Virginia ▾ VadaLaks

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

Load Balancers

Target Groups


Trust Stores [New](#)

▼ Auto Scaling

Auto Scaling Groups

[EC2](#) > Load balancers

Load balancers

 Actions ▾ [Create load balancer](#) ▾

Name ▾

DNS name ▾

State ▾

VPC ID ▾

Availability Zones

No load balancers

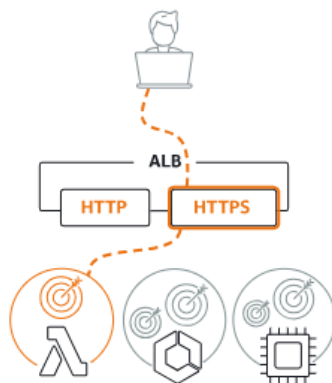
0 load balancers selected

Select a load balancer above.

Click on create option which is under application load balancer.

Application Load Balancer

[Info](#)

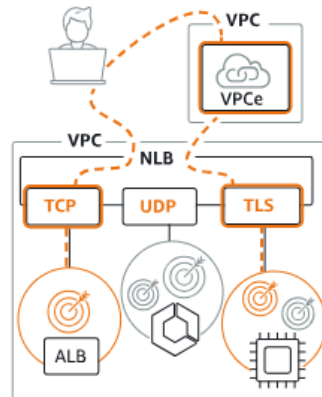


Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

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**

Give load balancer name

Basic configuration

Load balancer name

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

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 type of IP addresses that your subnets use. 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.

Select the three availability zones where the 3 instances are created

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.

☒ **us-east-1a (use1-az6)**

Subnet

IPv4 address

Assigned by AWS

☒ **us-east-1b (use1-az1)**

Subnet

IPv4 address

Assigned by AWS

☒ **us-east-1c (use1-az2)**

Subnet

IPv4 address

Assigned by AWS

Remove the default security group and select the security group which we created.

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

Load

sg-0988333ac302afbc8 VPC: vpc-064b83e84c2b968f5

Attach the target group which we created.

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

load

HTTP

Target type: Instance, IPv4

[Create target group](#)

Click on create load balancer.

► Server-side tasks and status

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Cancel

Create load balancer

The target is attached to load balancer and the load balancer is created.

✔ Successfully created load balancer: Load

It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

[EC2](#) > [Load balancers](#) > Load

Copy the DNS name of the load balancer

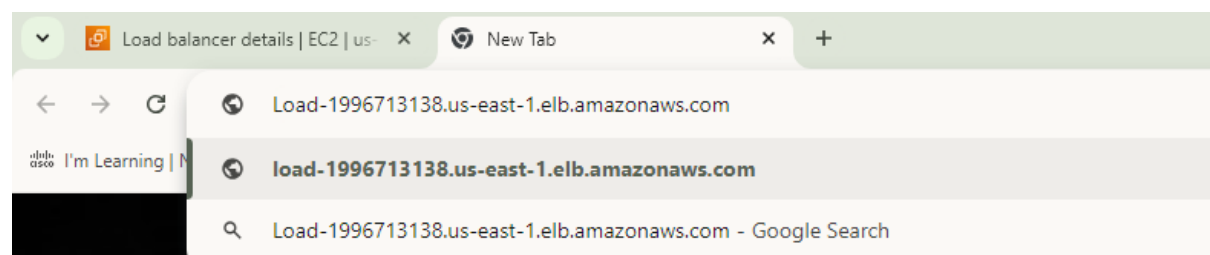
Load balancer ARN

arn:aws:elasticloadbalancing:us-east-1:471112914581:loadbalancer/app/Load/b11a9150ed4c0fd7

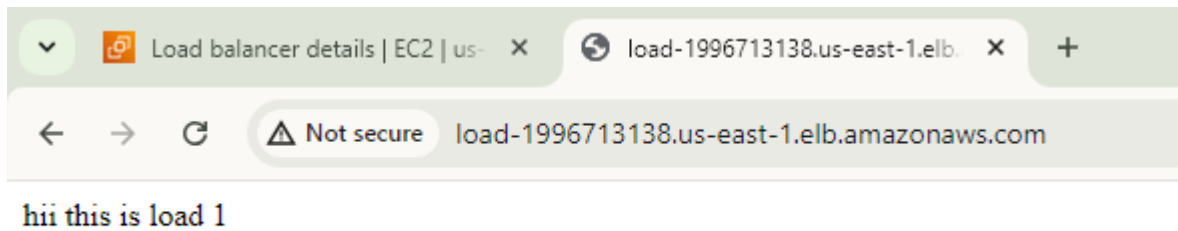
✔ DNS name copied

Load-1996713138.us-east-1.elb.amazonaws.com (A Record)

Paste it in chrome

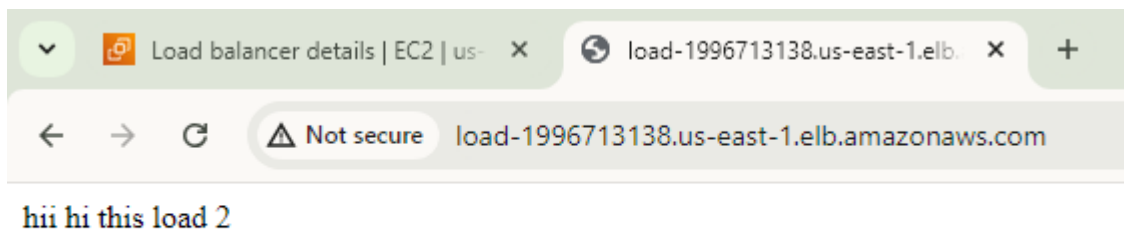


the data which we inserted in first instance will be appeared.



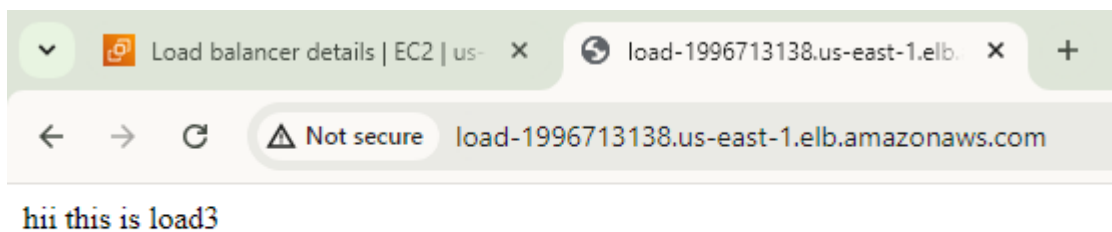
Do Refresh

Then the data which we inserted in second instance will be appeared.



Do Refresh

Then the data which we inserted in third instance will be appeared.



THIS IS HOW THE LOAD BALANCER WILL WORKS.