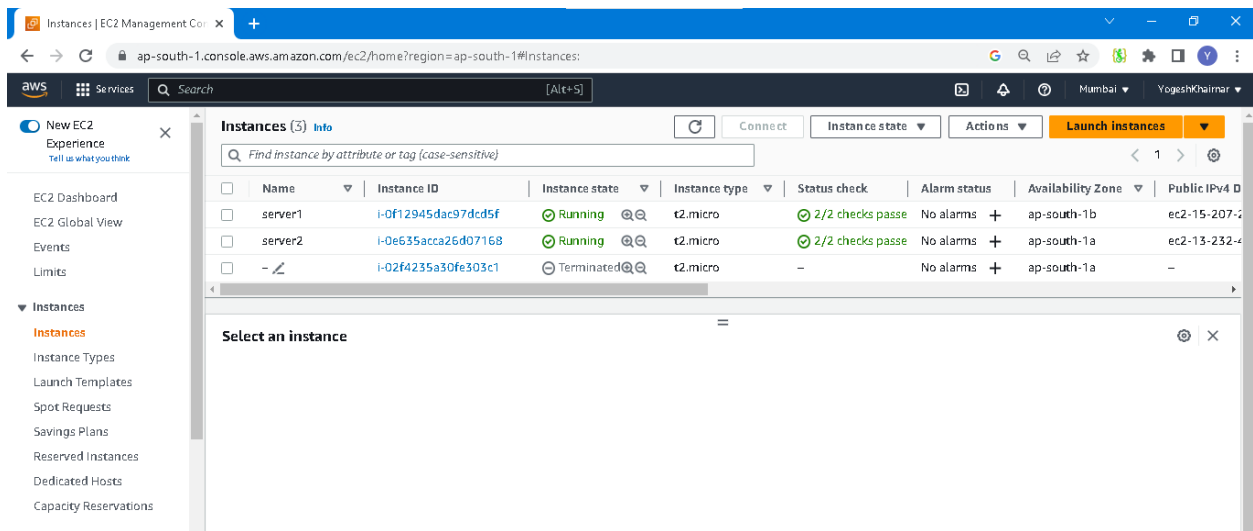


Load Balancer Practical.

Aim:-Host a website where 2 server will run simultaneously on one server.

Step 1:-Login to your aws account.

Step 2:Launch 2 instances.

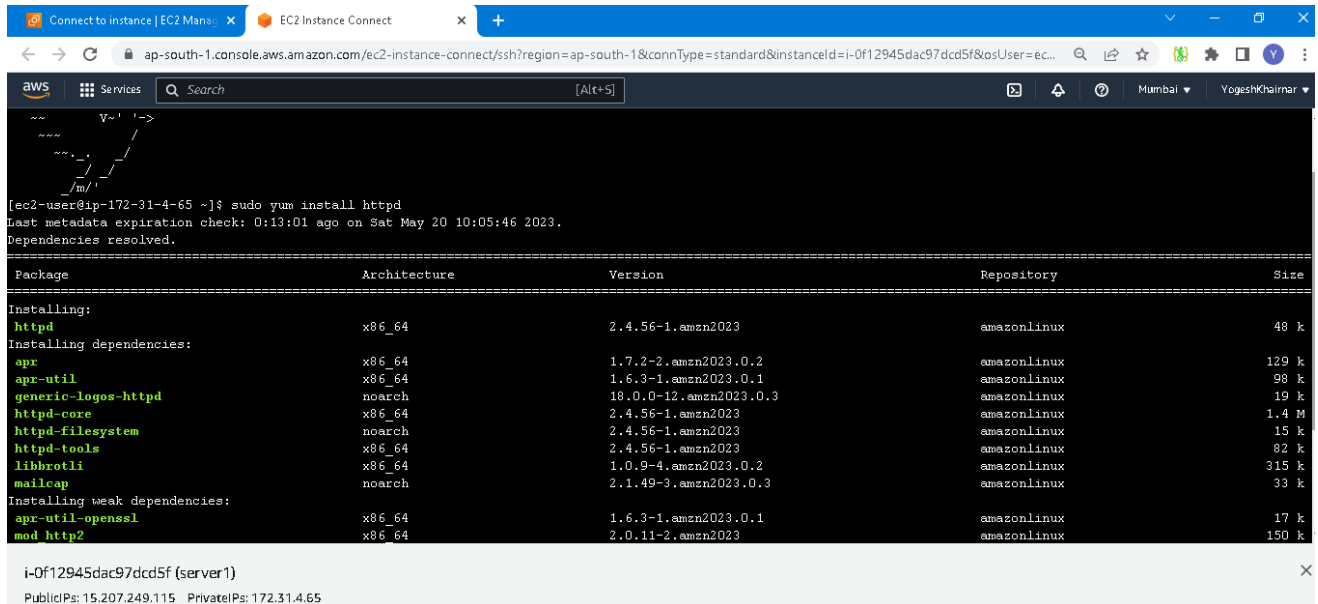


Wait until the status check gets 2/2 check pass or turns green.

(I have used AMI as Amazon Linux.)

Step 3:-After status check turn green >connect 1st instance>CLI will launch>Write following command for hosting website.

sudo yum install httpd

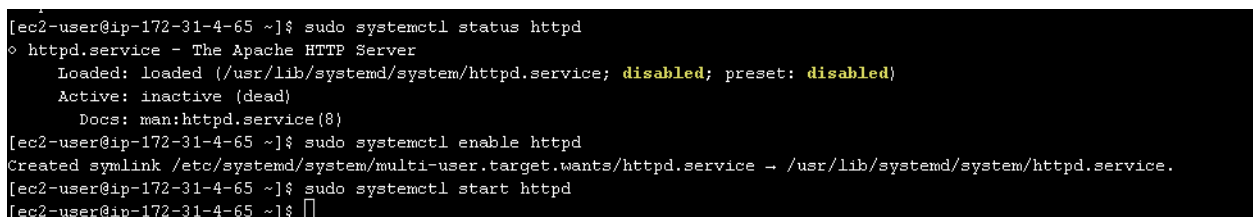


```
[ec2-user@ip-172-31-4-65 ~]$ sudo yum install httpd
Last metadata expiration check: 0:13:01 ago on Sat May 20 10:05:46 2023.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
httpd                                  x86_64            2.4.56-1.amzn2023  amazonlinux         48 k
Installing dependencies:
apr                                    x86_64            1.7.2-2.amzn2023.0.2  amazonlinux         129 k
apr-util                              x86_64            1.6.3-1.amzn2023.0.1  amazonlinux         98 k
generic-logos-httpd                  noarch            18.0.0-12.amzn2023.0.3  amazonlinux         19 k
httpd-core                            x86_64            2.4.56-1.amzn2023    amazonlinux         1.4 M
httpd-filesystem                     noarch            2.4.56-1.amzn2023    amazonlinux         15 k
httpd-tools                           x86_64            2.4.56-1.amzn2023    amazonlinux         82 k
libbrotli                             x86_64            1.0.9-4.amzn2023.0.2  amazonlinux        315 k
mailcap                               noarch            2.1.49-3.amzn2023.0.3  amazonlinux         33 k
Installing weak dependencies:
apr-util-openssl                     x86_64            1.6.3-1.amzn2023.0.1  amazonlinux         17 k
mod_http2                             x86_64            2.0.11-2.amzn2023    amazonlinux         150 k
=====
```

sudo systemctl status httpd

sudo systemctl enable httpd

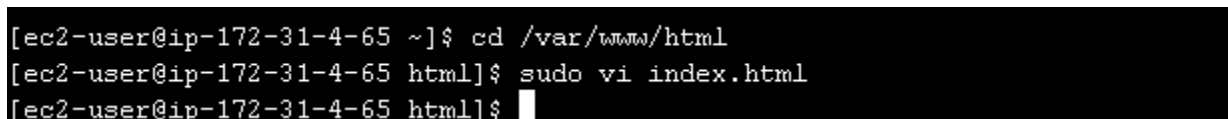
sudo systemctl start httpd



```
[ec2-user@ip-172-31-4-65 ~]$ sudo systemctl status httpd
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[ec2-user@ip-172-31-4-65 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-4-65 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-4-65 ~]$
```

cd /var/www/html

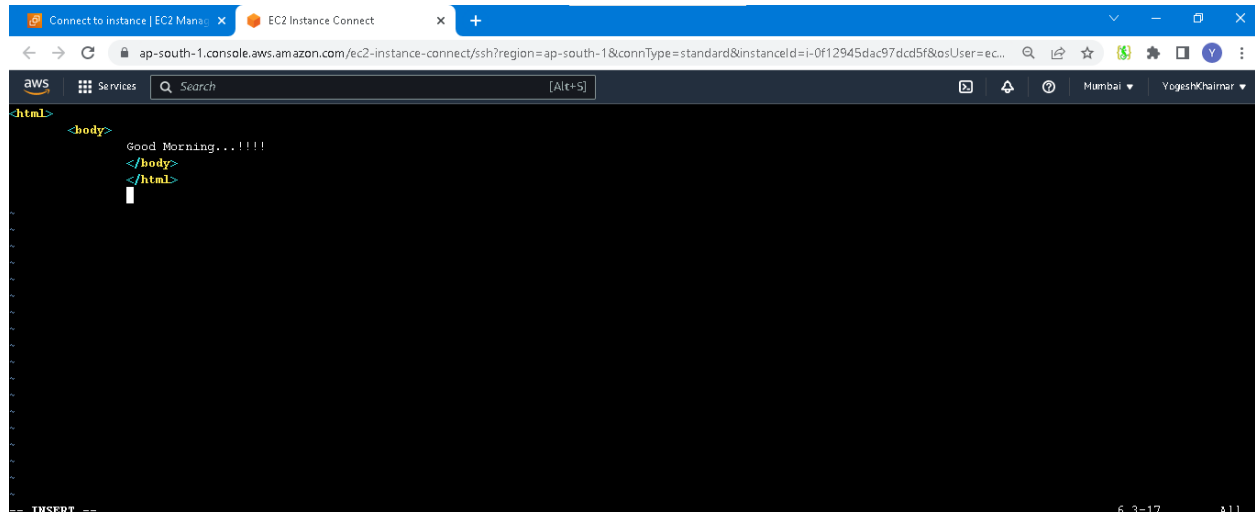
sudo vi index.html



```
[ec2-user@ip-172-31-4-65 ~]$ cd /var/www/html
[ec2-user@ip-172-31-4-65 html]$ sudo vi index.html
[ec2-user@ip-172-31-4-65 html]$
```

Editor page will open ,press i and write the following html code and save it by pressing Esc→shift+:→wq.

```
<html>
<body>
Good Morning ...!!!
</body>
</html>
```

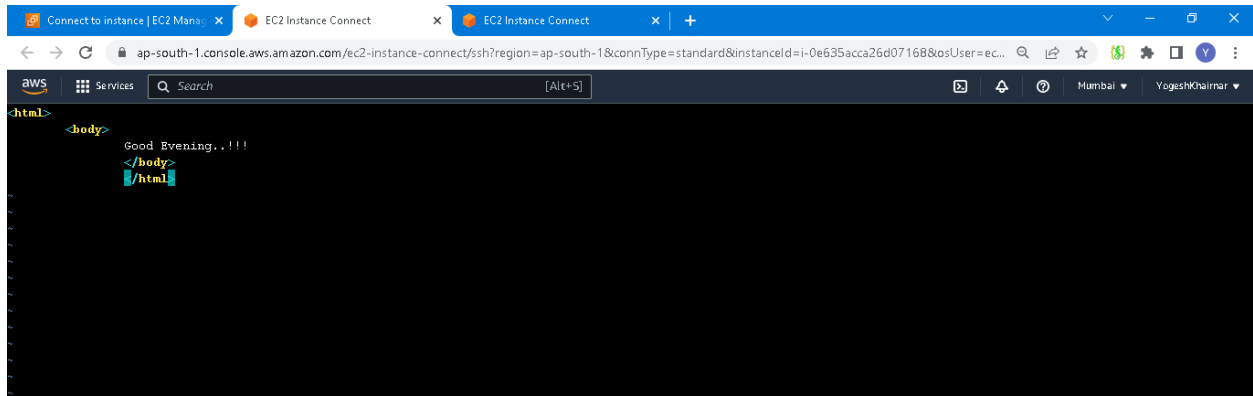


Now go back to instance >connect the 2nd instance>Follow the same above steps just change the index.html code.

```
Complete!
[ec2-user@ip-172-31-34-134 ~]$ sudo systemctl status httpd
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[ec2-user@ip-172-31-34-134 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-34-134 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-34-134 ~]$ cd /var/www/html
[ec2-user@ip-172-31-34-134 html]$ sudo vi index.html
[ec2-user@ip-172-31-34-134 html]$
```

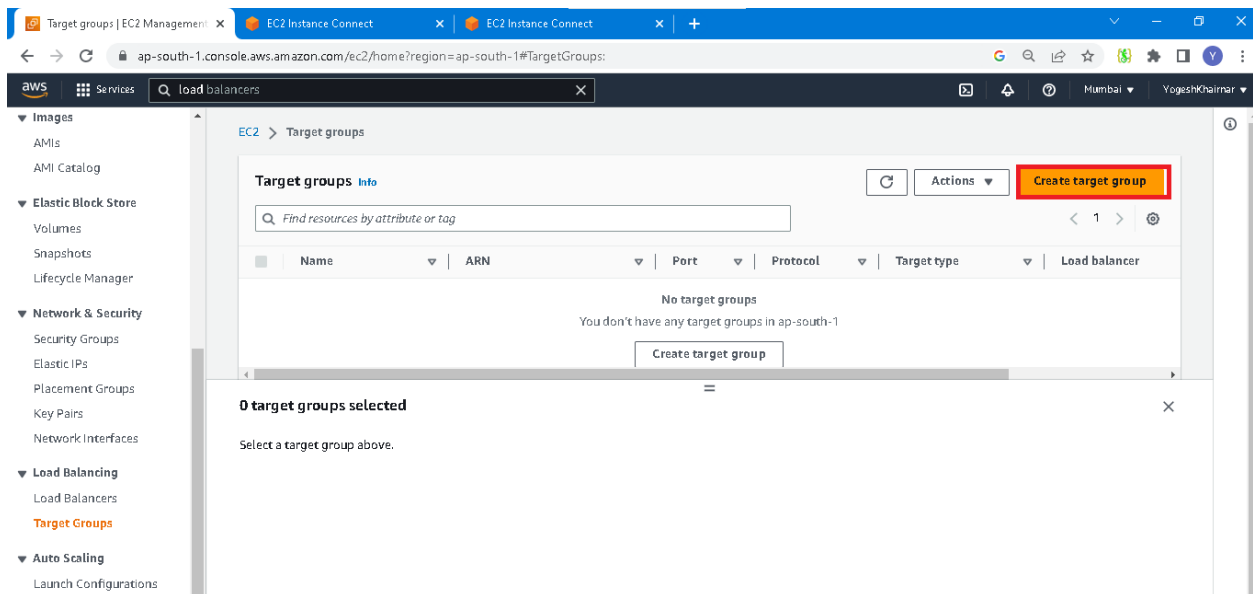
Write the following code in the editor and save it by pressing
Esc→shift+:→wq.

```
<html>
<body>
Good Evening ...!!!
</body>
</html>
```



Step 4:-Target Group.

Now go to EC2>Load balancing>Target group>click on create target group.



Specify group details > Basic configuration > Choose target type as *Instances*.

The screenshot shows the 'Specify group details' page in the AWS Management Console. The page is titled 'Specify group details' and includes a sub-header 'Your load balancer routes requests to the targets in a target group and performs health checks on the targets.' The page is divided into two main sections: 'Basic configuration' and 'Choose a target type'. Under 'Basic configuration', there is a note: 'Settings in this section can't be changed after the target group is created.' The 'Choose a target type' section has three radio buttons: 'Instances' (selected), 'IP addresses', and 'Lambda function'. The 'Instances' option is highlighted with a blue border and includes the following bullet points: 'Supports load balancing to instances within a specific VPC.', 'Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.'

Give the target name > keep the rest settings as default no changes needed > then click on Next.

The screenshot shows the 'Specify group details' page in the AWS Management Console, specifically the 'Basic configuration' section. The 'Target group name' field is filled with 'server3'. Below it, a note states: 'A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.' The 'Protocol' dropdown is set to 'HTTP' and the 'Port' field is filled with '80'. The 'VPC' dropdown is set to 'vpc-055fb55d54078257f' with the IP address '172.31.0.0/16' displayed below it. The 'Protocol version' section has three radio buttons: 'HTTP1' (selected), 'HTTP2', and 'gRPC'. The 'Health checks' section has a note: 'The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.' The 'Health check protocol' dropdown is set to 'HTTP'.

Register targets>Select that 2 instances which we have created>then click on *Include as pending below*>click on create target group.

Target groups | EC2 Management | EC2 Instance Connect | EC2 Instance Connect

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup:

Services load balancers

Step 1
Specify group details

Step 2
Register targets

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 resources by property or value

<input checked="" type="checkbox"/>	Instance ID	Name	Status	Security groups
<input checked="" type="checkbox"/>	i-0f12945dac97dcd5f	server1	Running	launch-wizard-23
<input checked="" type="checkbox"/>	i-0e635acca26d07168	server2	Running	launch-wizard-24

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

Target groups | EC2 Management | EC2 Instance Connect | EC2 Instance Connect

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup:

Services load balancers

Step 1
Specify group details

Step 2
Register targets

Review targets

Targets (2)

Remove all pending

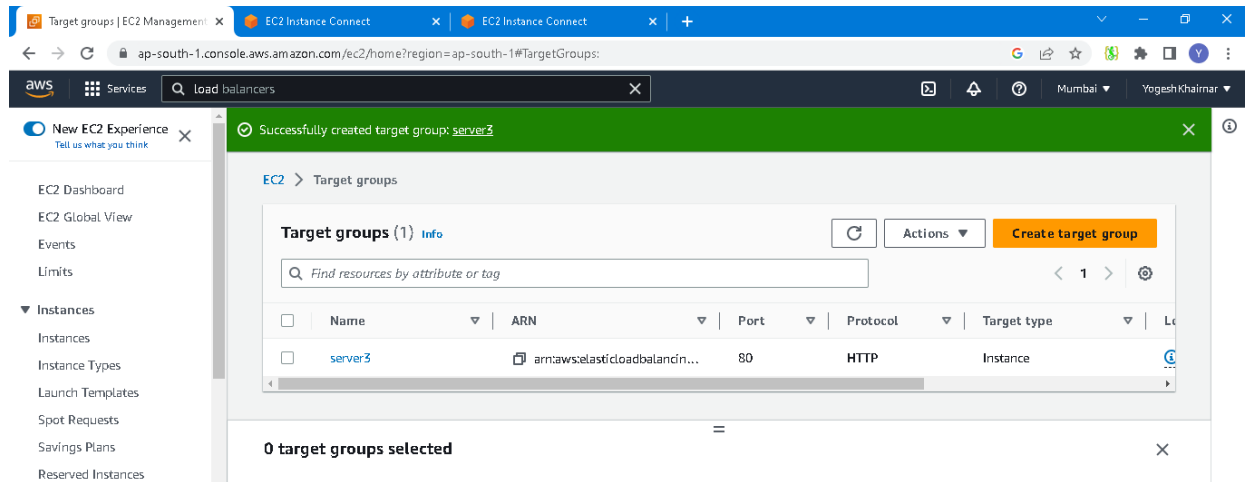
All Filter resources by property or value

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone
X	Pending	i-0f12945dac97dcd5f	server1	80	Running	launch-wizard-23	ap-s
X	Pending	i-0e635acca26d07168	server2	80	Running	launch-wizard-24	ap-s

2 pending

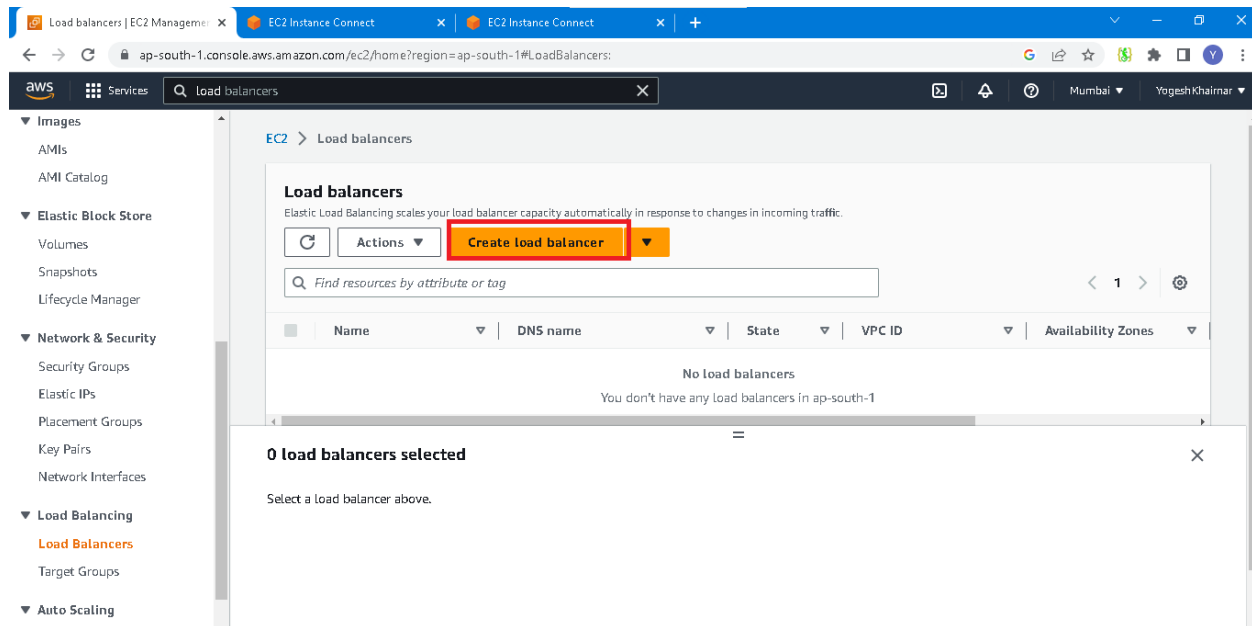
Cancel Previous **Create target group**

Target group will created.



Step 5:-Load Balancers.

Go to load balancers>Create load balancer.



Application Load Balancer>Create.

The screenshot shows the AWS Management Console's 'Load balancer types' page. It features three columns, each representing a different load balancer type:

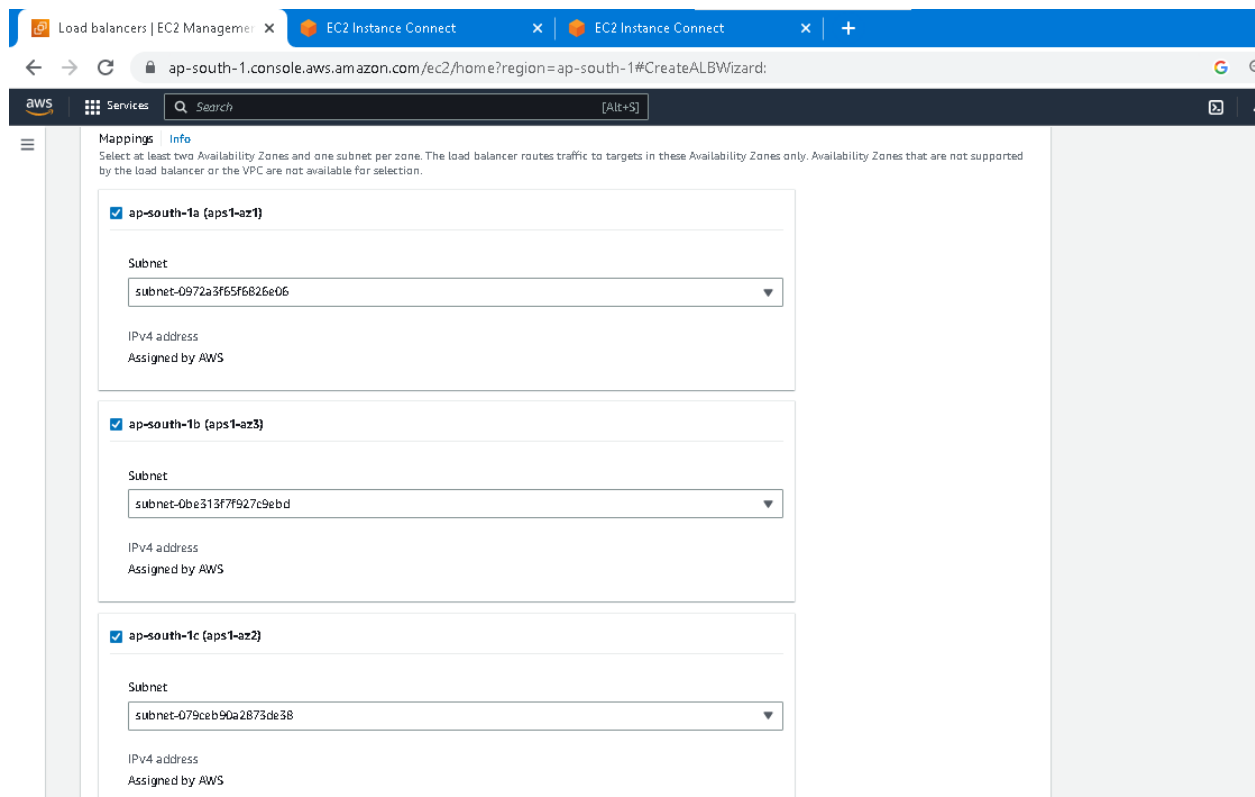
- Application Load Balancer (ALB):** The diagram shows a client connecting to an ALB, which then routes traffic to target instances. The description states: "Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers." The 'Create' button is highlighted with a red box.
- Network Load Balancer (NLB):** The diagram shows a client connecting to an NLB within a VPC, which routes traffic to target instances. The description states: "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."
- Gateway Load Balancer (GWLB):** The diagram shows a client connecting to a GWLB, which routes traffic to a fleet of third-party virtual appliances. The description states: "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 Application Load Balancer>Basic Configuration -Give load balancer name,Select scheme as Internet-facing.

The screenshot shows the 'Create Application Load Balancer' wizard in the AWS Management Console. The 'Basic configuration' section is visible, containing the following fields:

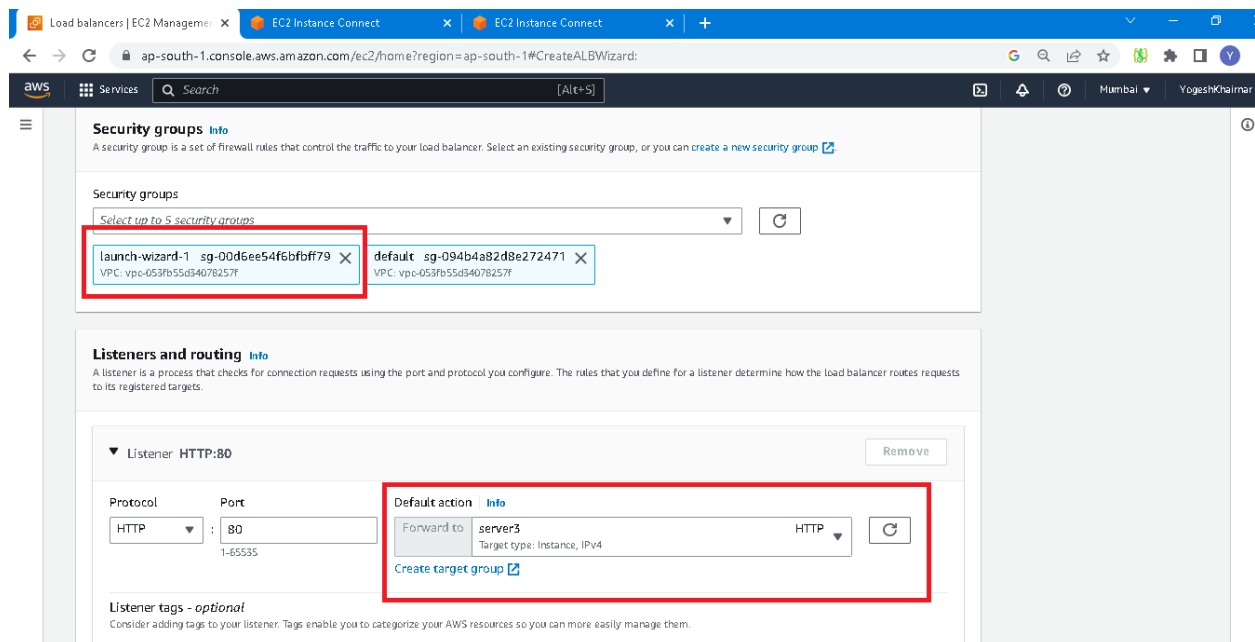
- Load balancer name:** A text input field with the value 'lb'. Below the field, it states: "Name must be unique within your AWS account and can't be changed after the load balancer is created. A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen."
- Scheme:** A dropdown menu with 'Internet-facing' selected. Below the dropdown, it states: "Scheme can't be changed after the load balancer is created." The 'Internet-facing' option is described as: "An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)". The 'Internal' option is described as: "An internal load balancer routes requests from clients to targets using private IP addresses."

In Mappings select all 3 availability zones.

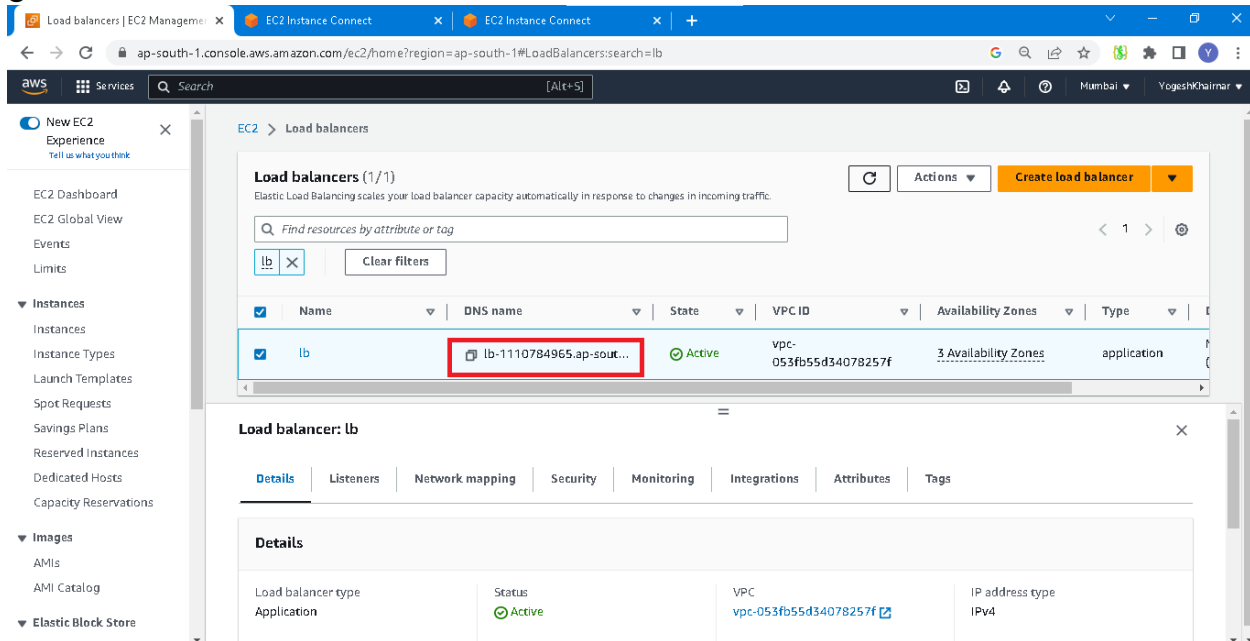


Security groups-Select vpc launch wizard-1,also keep the default vpc.

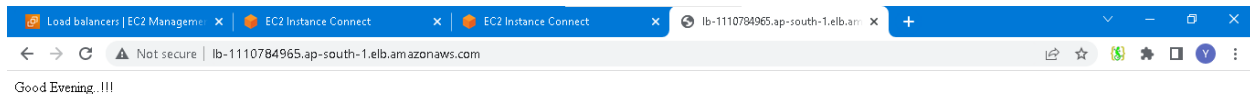
Listener Routing-Select the target group which we have created.>After this click on create load balancer.



Click on view load balancer>load balancer is created>Now wait until state gets *active*.

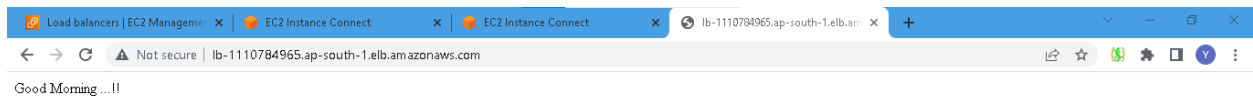


After state get active copy that DNS name and paste in the browser.



According to the traffic the instance page will launch it can be of 1st or 2nd instance.

Refresh the page



You can see that on one server simultaneously 2 server are launched.