Project Details:

Objective: Create a web application and deploy it on an EC2 instance using different Linux distributions.

The web application must include at least 3 HTML pages.

Load Balancing must be implemented for the deployed application.

The deployment process should involve different AWS IAM users—each user should deploy a separate instance of the web application.

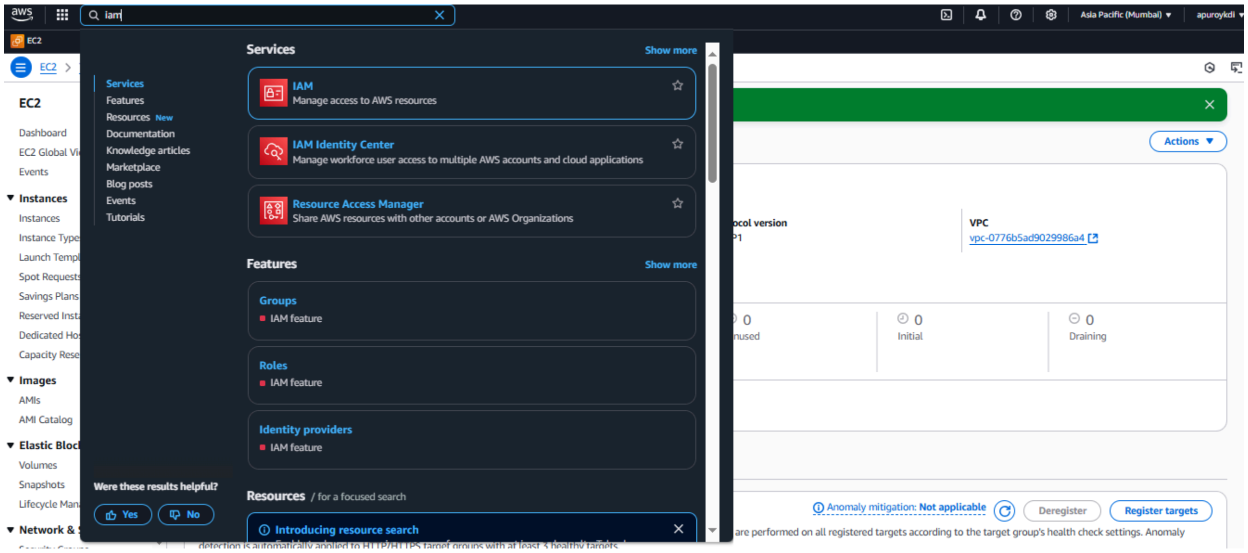
The website should be set up and managed by these different IAM users accordingly.

—---

Deployment:

To do this project we need to login into **AWS Console** with the **Root Id**.Then need to go to The **IAM Dashboard** to create the Users and the groups. As bellow:

IAM Dashboard:



IAM
Manage access to AWS resources

IAM Identity Center
Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager
Share AWS resources with other accounts or AWS Organizations

Groups
IAM feature

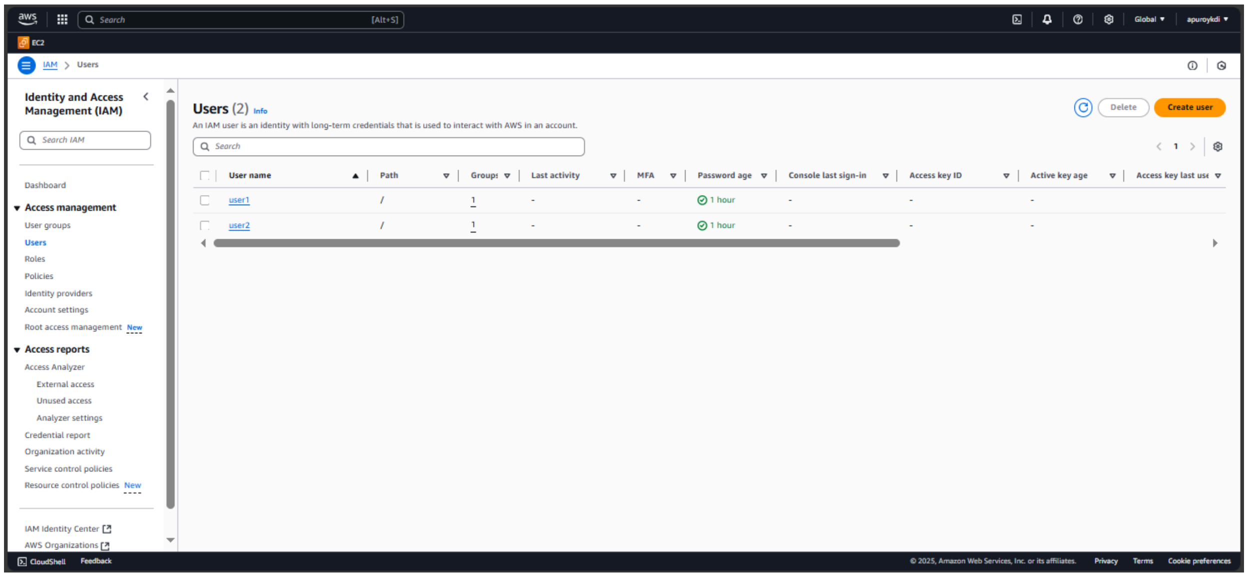
Roles
IAM feature

Identity providers
IAM feature

Resources / for a focused search

Introducing resource search

Users list:



Users (2)

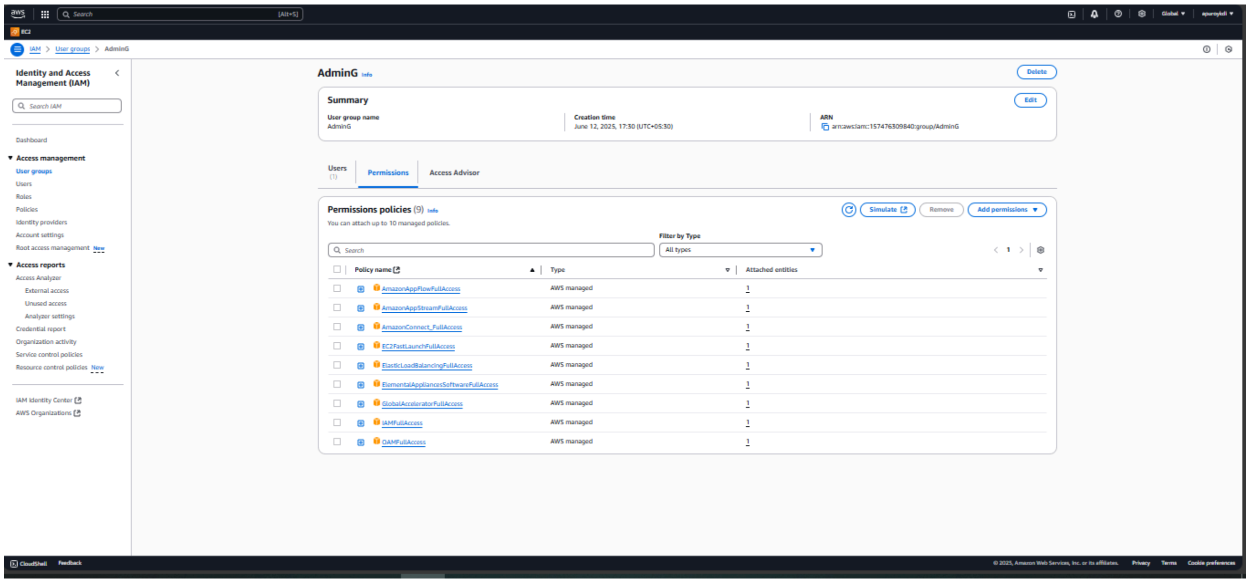
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Access key last use
user1	/	1	-	-	✓ 1 hour	-	-	-	
user2	/	1	-	-	✓ 1 hour	-	-	-	

We created two users mentioned above. And these users are added with groups for their specified permissions..

Group Name and Permissions With user name:

AdminG group:



AdminG

Summary

User group name: AdminG

Creation time: June 12, 2025, 17:30 (UTC+05:30)

ARN: arn:aws:iam::157476309840:group/AdminG

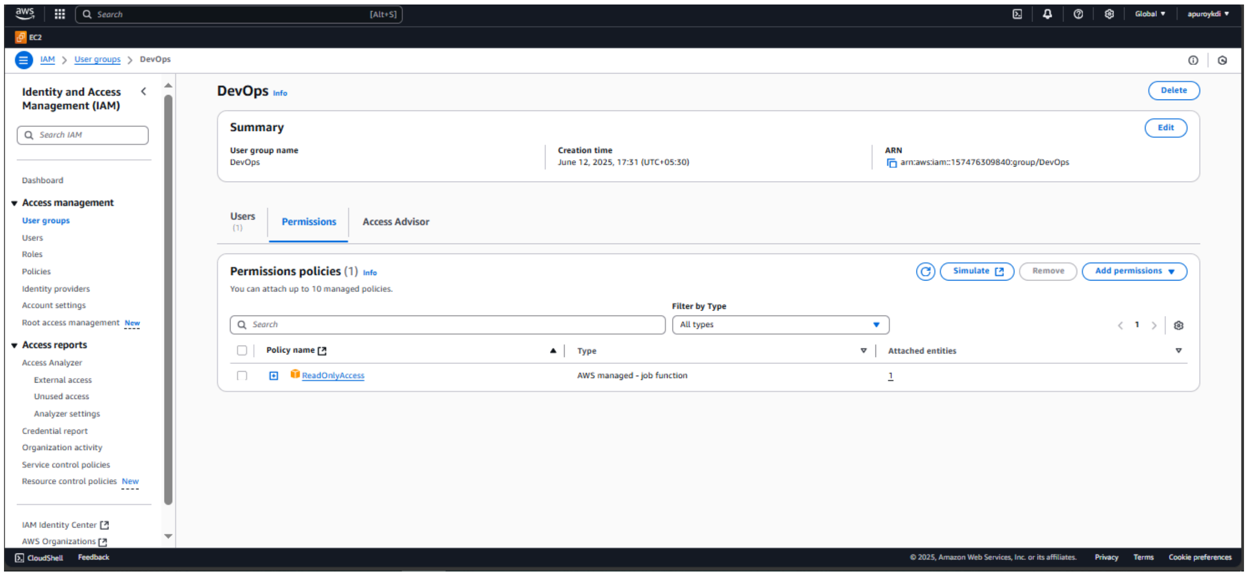
Permissions policies (9)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonAppFlowFullAccess	AWS managed	1
AmazonAppStreamFullAccess	AWS managed	1
AmazonConnect_FullAccess	AWS managed	1
EC2FastLaunchFullAccess	AWS managed	1
ElasticLoadBalancingFullAccess	AWS managed	1
ElementalAppliancesSoftwareFullAccess	AWS managed	1
GlobalAcceleratorFullAccess	AWS managed	1
IAMFullAccess	AWS managed	1
OAMFullAccess	AWS managed	1

This is the **Admin group** named **AdminG** and **user1** is added to this group who has the full permission to maintain all the services as EC2 Instance, IAM …..

DevOps group:


DevOps Info

Summary

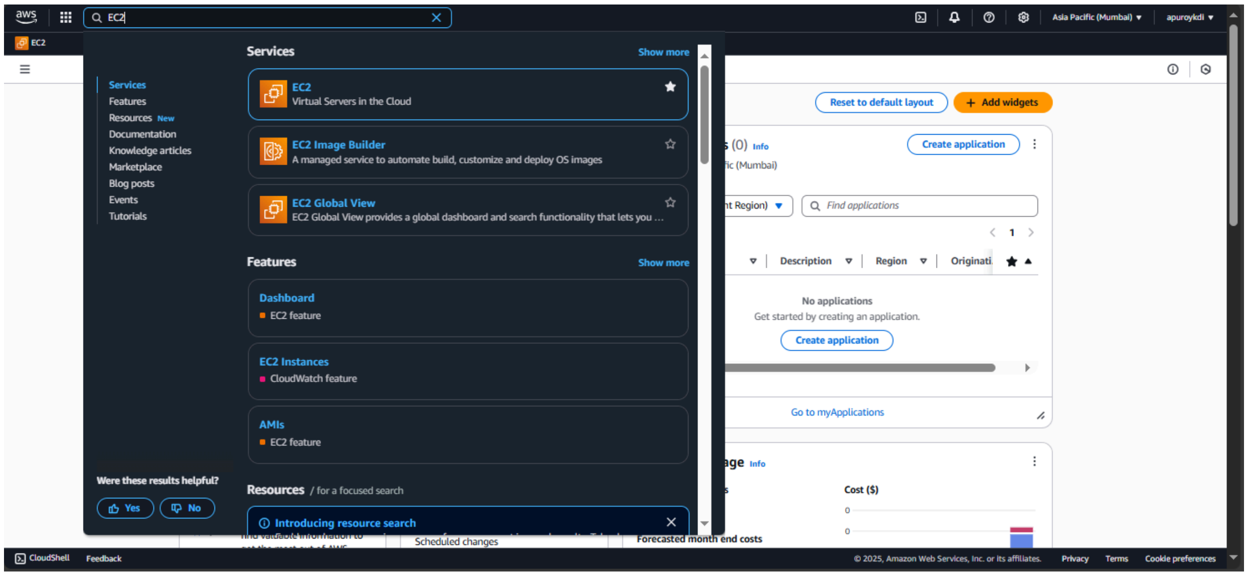
User group name	Creation time	ARN
DevOps	June 12, 2025, 17:31 (UTC+05:30)	arn:aws:iam::157476309840:group/DevOps

Permissions policies (1)
You can attach up to 10 managed policies.

Policy name	Type	Attached entities
ReadOnlyAccess	AWS managed - job function	1

This is the **Normal group** named **DevOps** and **user2** is added to this group who has only the **read-only** permission.

For the rest of the process we need to go into **EC2 Service** by searching in the search bar**.** As bellow:


Services

EC2
Virtual Servers in the Cloud

EC2 Image Builder
A managed service to automate build, customize and deploy OS images

EC2 Global View
EC2 Global View provides a global dashboard and search functionality that lets you ...

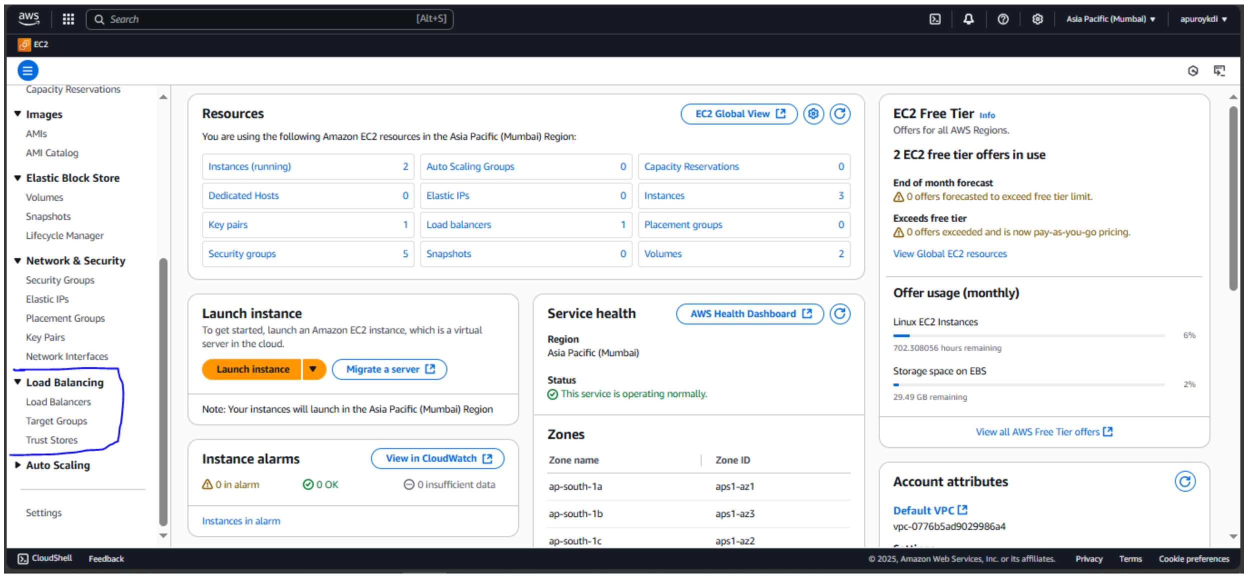
Features

Dashboard
EC2 feature

EC2 Instances
CloudWatch feature

AMIs
EC2 feature

Now at first we have to configure the **Load Balancers and the Target Groups** from the **Load Balancing** option which is in the left side and in the third position from bottom of EC2 Options menu. As bellow:



Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	2	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	3
Key pairs	1	Load balancers	1	Placement groups	0
Security groups	5	Snapshots	0	Volumes	2

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Instance alarms

0 in alarm · 0 OK · 0 insufficient data

Instances in alarm

Service health

Region
Asia Pacific (Mumbai)

Status
This service is operating normally.

Zones

Zone name	Zone ID
ap-south-1a	aps1-az1
ap-south-1b	aps1-az3
ap-south-1c	aps1-az2

EC2 Free Tier Info
Offers for all AWS Regions.

2 EC2 free tier offers in use

End of month forecast
0 offers forecasted to exceed free tier limit.

Exceeds free tier
0 offers exceeded and is now pay-as-you-go pricing.

View Global EC2 resources

Offer usage (monthly)

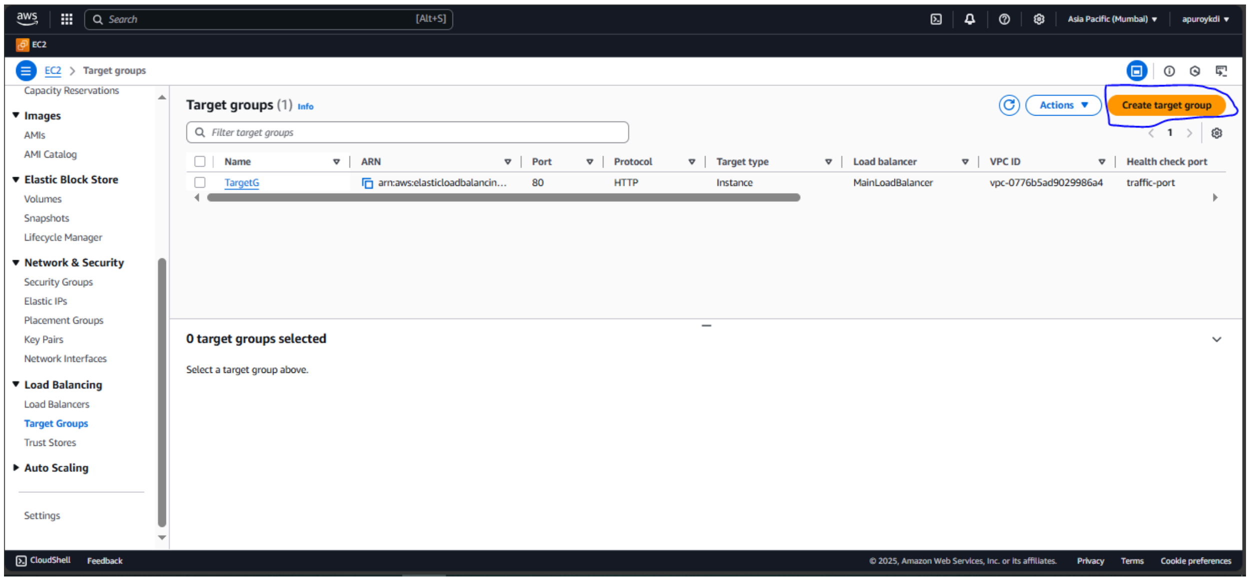
Linux EC2 Instances 6%
702.308056 hours remaining

Storage space on EBS 2%
29.49 GB remaining

Account attributes

Default VPC
vpc-0776b5ad9029986a4

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Target groups (1) Info

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID	Health check port
TargetG	arn:aws:elasticloadbalancin...	80	HTTP	Instance	MainLoadBalancer	vpc-0776b5ad9029986a4	traffic-port

0 target groups selected

Select a target group above.

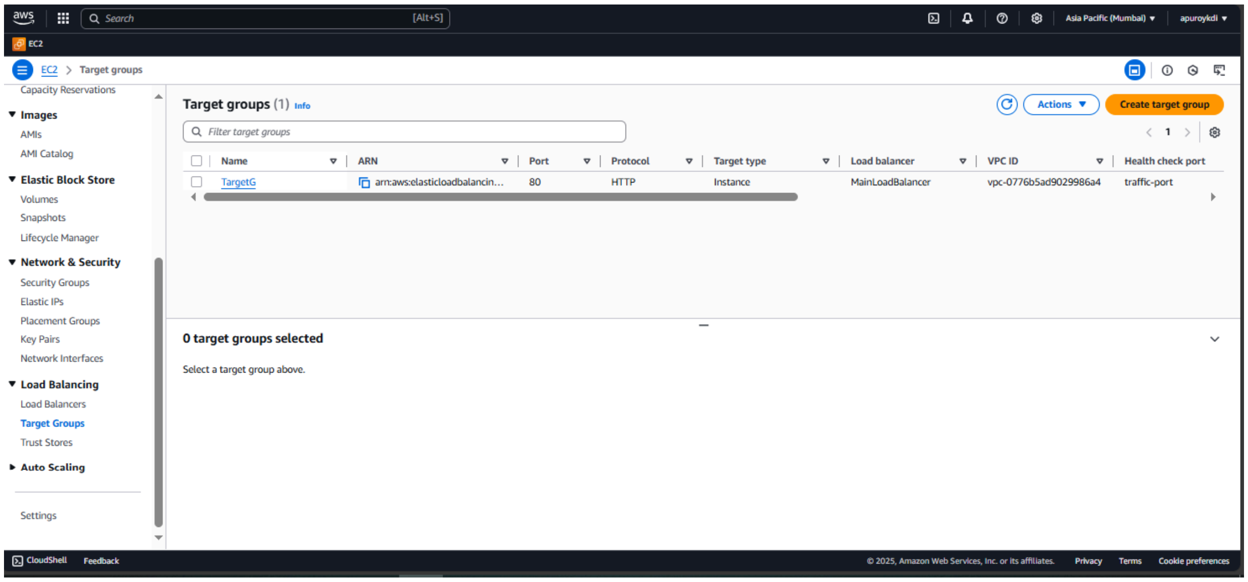
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Here we can see the three options as listed on the above picture:

1. **Load Balancers**
2. **Target Groups**
3. **Trust Stores**

Before configuring the **Load Balancers** , we need to configure the **Target Groups** first. So we click the 2nd option for **Target Groups .**

Target Group Dashboard:



Target groups (1) Info

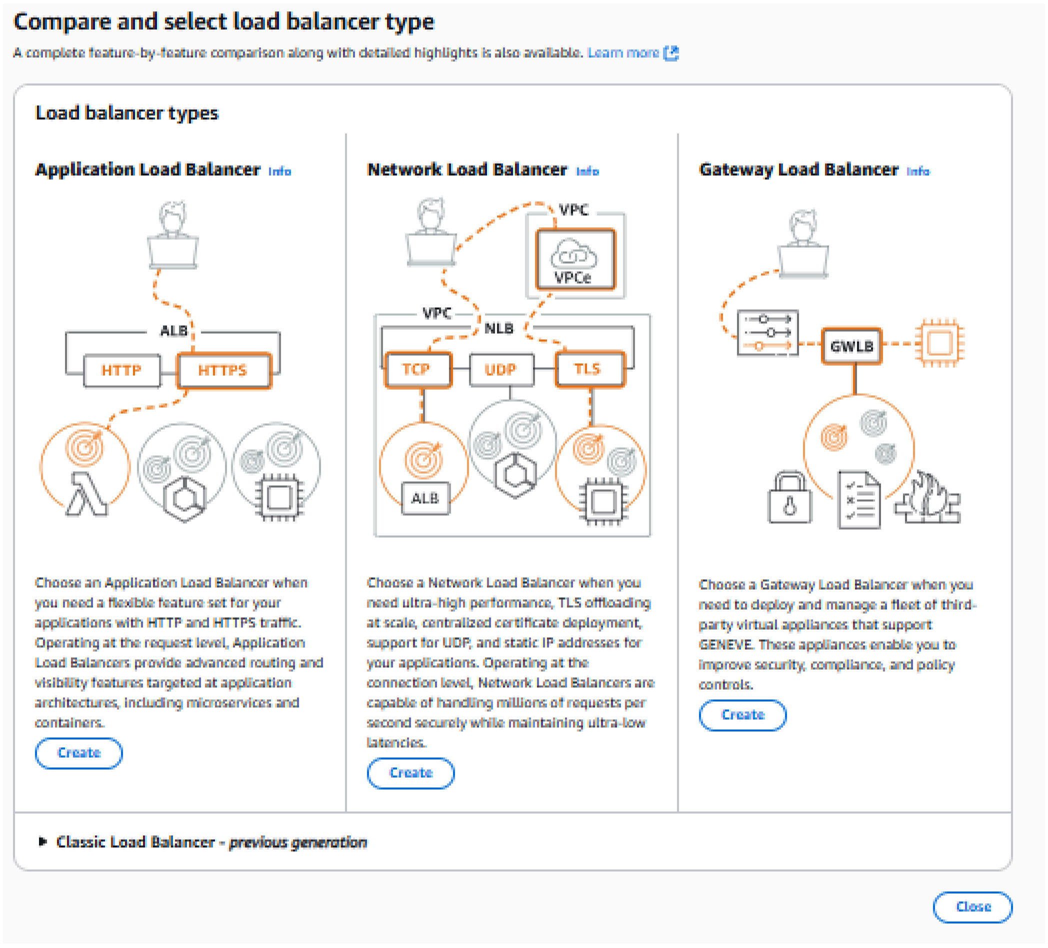
Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID	Health check port
TargetG	arn:aws:elasticloadbalancin...	80	HTTP	Instance	MainLoadBalancer	vpc-0776b5ad9029986a4	traffic-port

0 target groups selected

Select a target group above.

If we don't have any created group, then we need to create groups by clicking the **Create target group** (as shown in the above picture) . But we have created a group named **TargetG.**

As we have created the **Target Group**. Now I need to configure the **Load Balancers.** After clicking the Load Balancers we would see the manu as **Create load balancer .** By clicking that menu there will be displayed three new menus to create the load balancer as **1.HTTP, HTTPS, 2.TCP, UDP, TLS, 3. GWLB** respectively . But we need the 1st one for this project .



Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. Learn more

Load balancer types

Application Load Balancer Info

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

Create

Network Load Balancer Info

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer Info

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

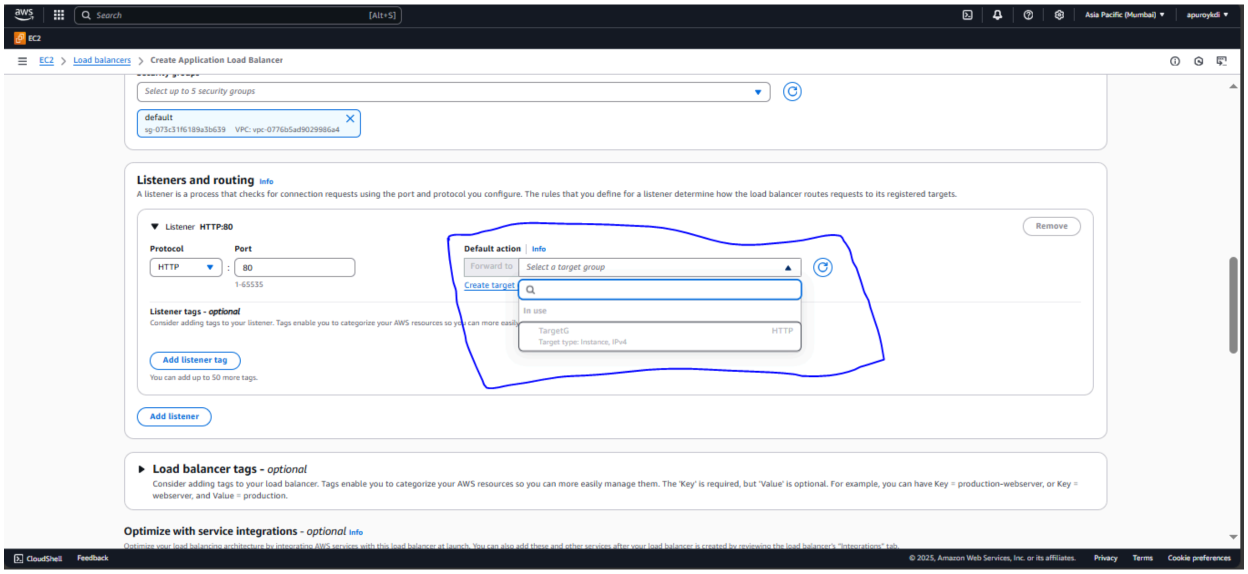
Create

▸ Classic Load Balancer - *previous generation*

Close

After this, there will be some queries that we need to fill properly to create a loader.

Select the Target Group:



EC2 > Load balancers > Create Application Load Balancer

Select up to 5 security groups

default
sg-073c31f6189a3b639 VPC: vpc-0776b5ad9029986a4

Listeners and routing Info

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

▾ Listener HTTP:80 — Remove

Protocol: HTTP
Port: 80
1-65535

Default action Info
Forward to: Select a target group
Create target
In use
TargetG — HTTP
Target type: Instance, IPv4

Listener tags - *optional*
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily

Add listener tag
You can add up to 50 more tags.

Add listener

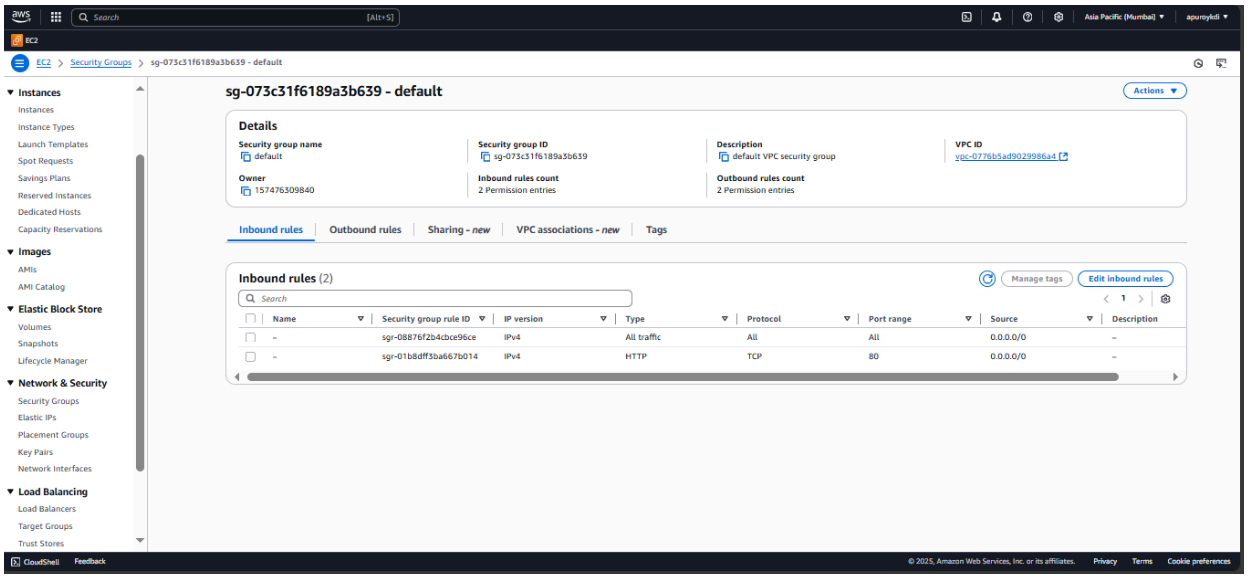
▸ **Load balancer tags** - *optional*
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value =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

Optimize with service integrations - *optional* Info
Optimize your load balancing architecture by integrating AWS services with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the load balancer's "Integrations" tab.

This is why we need to create the **Target Group** first. Here we have to select or create the **Target Group** to fulfill the criteria to create the **Load Balancer .**

Now we need to configure the **Security Groups for Inbound rules Outbound rules.** Only allow the **HTTP** connections for the Port 80. As below:

Inbound Rules:



sg-073c31f6189a3b639 - default

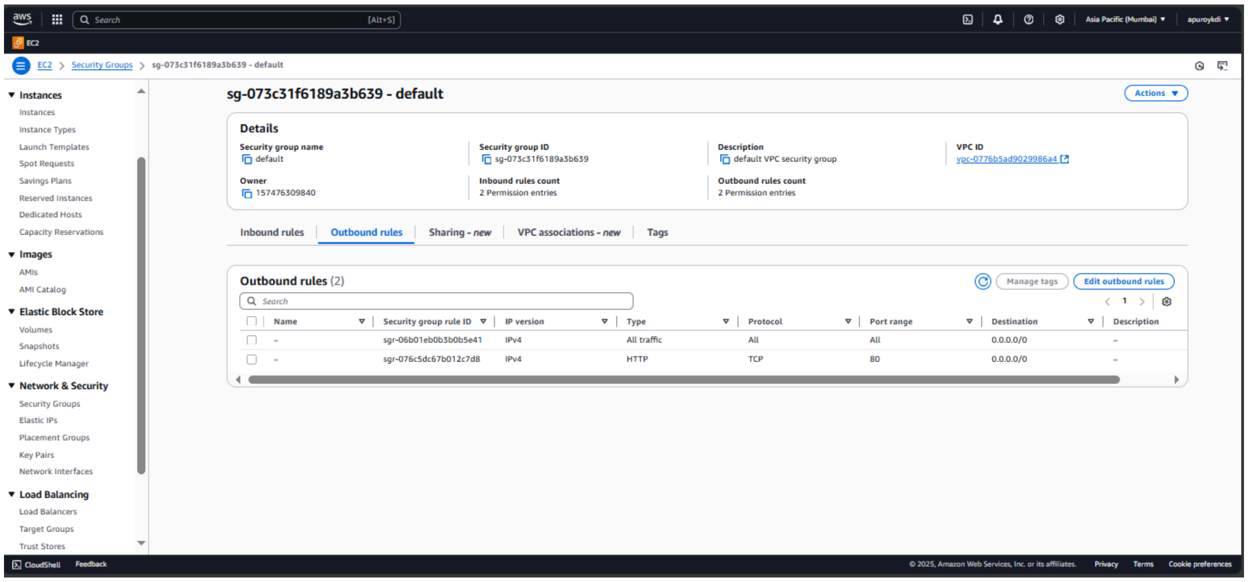
Details

Security group name: default
Security group ID: sg-073c31f6189a3b639
Description: default VPC security group
VPC ID: vpc-0776b5ad9029986a4
Owner: 157476309840
Inbound rules count: 2 Permission entries
Outbound rules count: 2 Permission entries

Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-08876f2b4cbce96ce	IPv4	All traffic	All	All	0.0.0.0/0	-
-	sgr-01b8dff3ba667b014	IPv4	HTTP	TCP	80	0.0.0.0/0	-

Outbound Rules:



sg-073c31f6189a3b639 - default

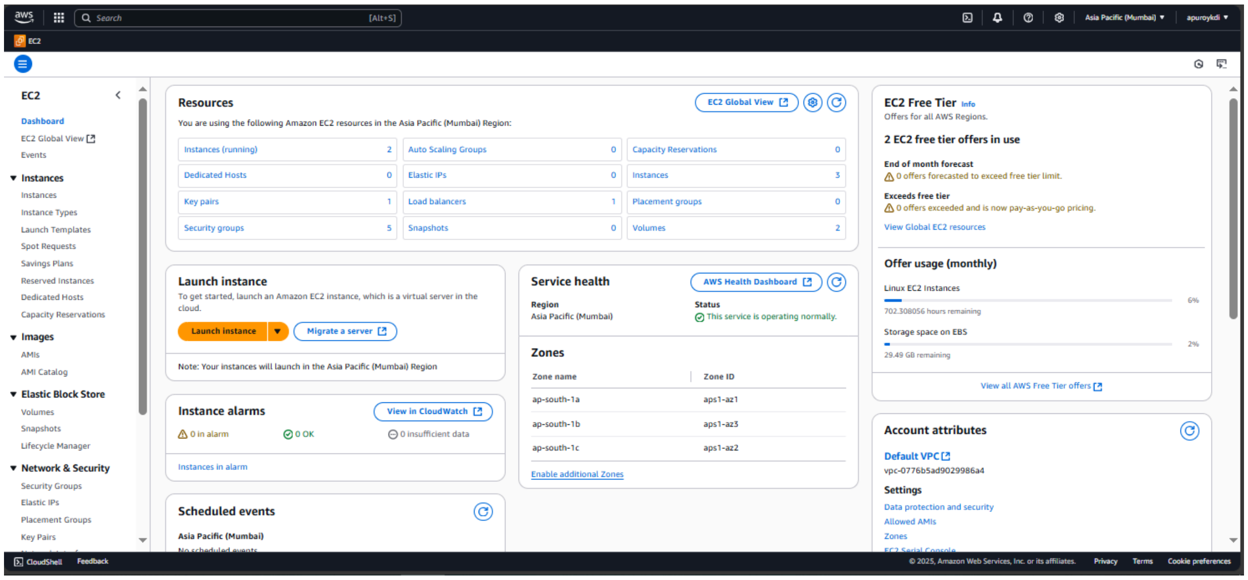
Details

Security group name: default
Security group ID: sg-073c31f6189a3b639
Description: default VPC security group
VPC ID: vpc-0776b5ad9029986a4
Owner: 157476309840
Inbound rules count: 2 Permission entries
Outbound rules count: 2 Permission entries

Outbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-06b01eb0b3b0b5e41	IPv4	All traffic	All	All	0.0.0.0/0	-
-	sgr-076c5dc67b012c7d8	IPv4	HTTP	TCP	80	0.0.0.0/0	-

Now we will go for creating the EC2 Instances and click the **Launch instance** as showing below:



Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	2	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	3
Key pairs	1	Load balancers	1	Placement groups	0
Security groups	5	Snapshots	0	Volumes	2

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Service health

Region: Asia Pacific (Mumbai)

Status: This service is operating normally.

Zones

Zone name	Zone ID
ap-south-1a	aps1-az1
ap-south-1b	aps1-az3
ap-south-1c	aps1-az2

Instance alarms

0 in alarm   0 OK   0 insufficient data

Scheduled events

Asia Pacific (Mumbai)

No scheduled events

EC2 Free Tier

Offers for all AWS Regions.

2 EC2 free tier offers in use

End of month forecast
0 offers forecasted to exceed free tier limit.

Exceeds free tier
0 offers exceeded and is now pay-as-you-go pricing.

Offer usage (monthly)

Linux EC2 Instances 6%
702.308056 hours remaining

Storage space on EBS 2%
29.49 GB remaining

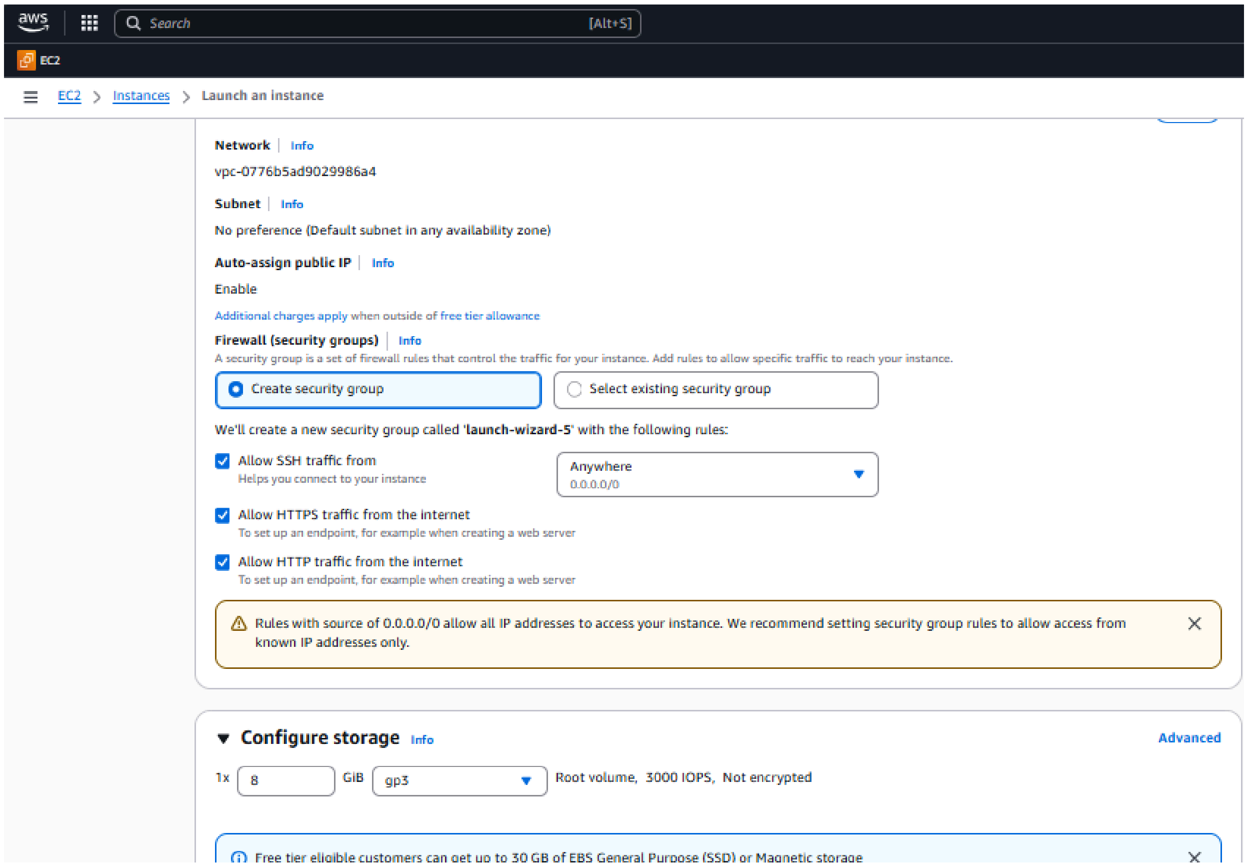
Account attributes

Default VPC
vpc-0776b5ad9029986a4

Settings
Data protection and security
Allowed AMIs
Zones
EC2 Serial Console

© 2025, Amazon Web Services, Inc. or its affiliates.

Then we will type the instance or the server name as our choice . And also select the **Amazon Machine Image (AMI)** according to our needs **or** requirements . Need to remember that the **Firewall (Security Groups)** alway be the same as shown below in the picture for accessing multiple **Apache Servers**. Need to check all the boxes as it may be needed in the future (both HTTP and HTTPS) to secure our server or website by applying the **SSL Certificate**.



Network | Info
vpc-0776b5ad9029986a4

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-5**' with the following rules:

- [✓] Allow SSH traffic from — Anywhere 0.0.0.0/0
  Helps you connect to your instance
- [✓] Allow HTTPS traffic from the internet
  To set up an endpoint, for example when creating a web server
- [✓] Allow HTTP traffic from the internet
  To set up an endpoint, for example when creating a web server

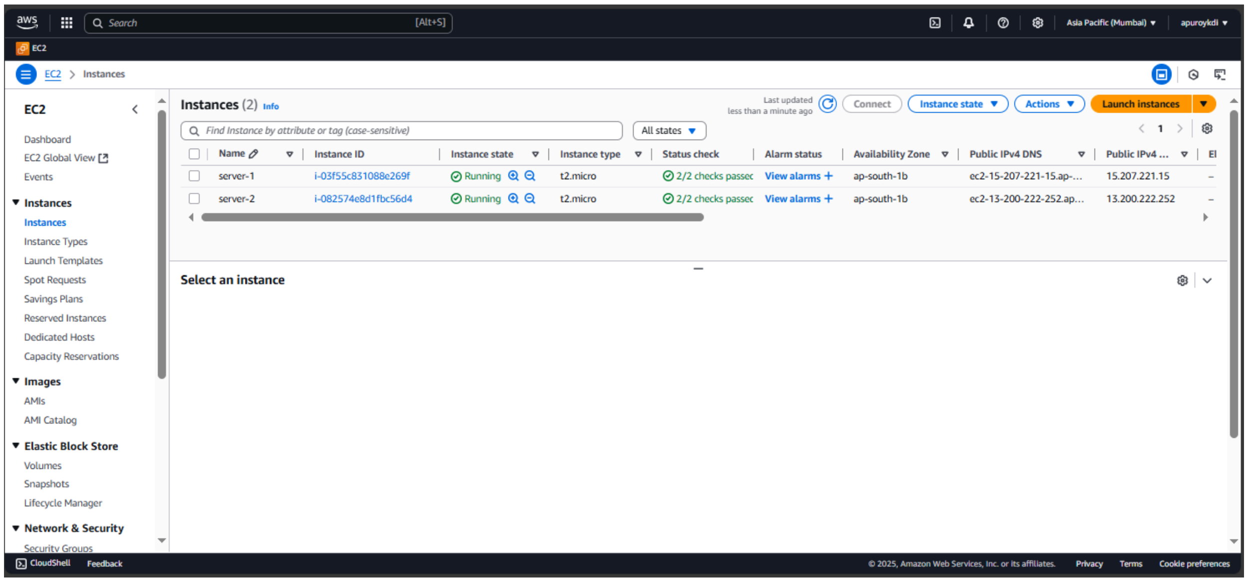
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info — Advanced

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

After creating the EC2 Instance need to wait a few seconds or minutes as it will take time to initialize the instance to use . Then login into the server by connecting.



Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
server-1	i-03f55c831088e269f	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-15-207-221-15.ap-...	15.207.221.15
server-2	i-082574e8d1fbc56d4	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-200-222-252.ap...	13.200.222.252

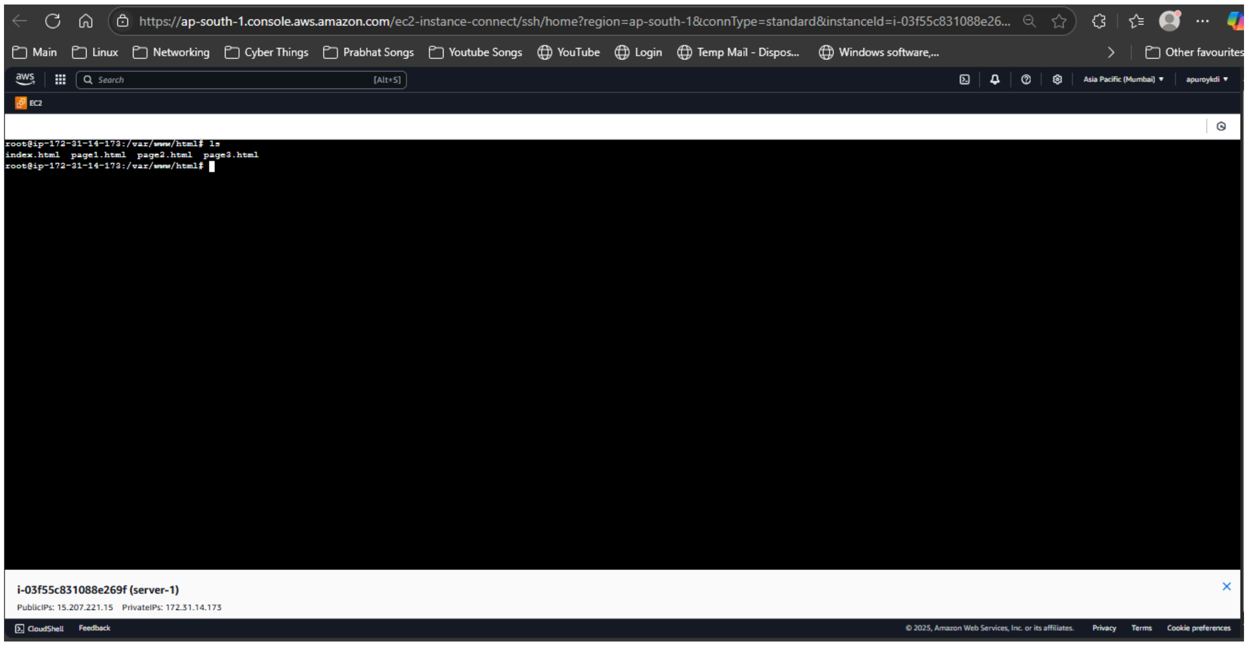
Select an instance

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

After successfully login into the server, need to configure the apache2 service by some commands . The commands given below:

Redhat based:
$sudo su —------ (For root access)
#yum update -y —-- (To update the system .)
#yum install httpd -y —--- (To install apache2 server into the system)
#systemctl status httpd —--(To check the status the service is running or not)
#systemctl start httpd —-- (To start the service if not started)
#cd /var/www/html —--- (The default location of apache server where we need to add the web pages)
#echo "<head>This is the page 1 from the server 1</head>" > page1.html —--- (First page with the extension of .html)
#echo "<head>This is the page 2 from the server 1</head>" > page2.html —--- (Second page with the extension of .html)
#echo "<head>This is the page 3 from the server 1</head>" > page3.html —--- (Third page with the extension of .html)
#systemctl restart apache2 —-- (To restart the service to the updation and execution of the pages into the web server)
#

1st Server is Amazon Linux:



```
root@ip-172-31-14-173:/var/www/html# ls
index.html  page1.html  page2.html  page3.html
root@ip-172-31-14-173:/var/www/html#
```

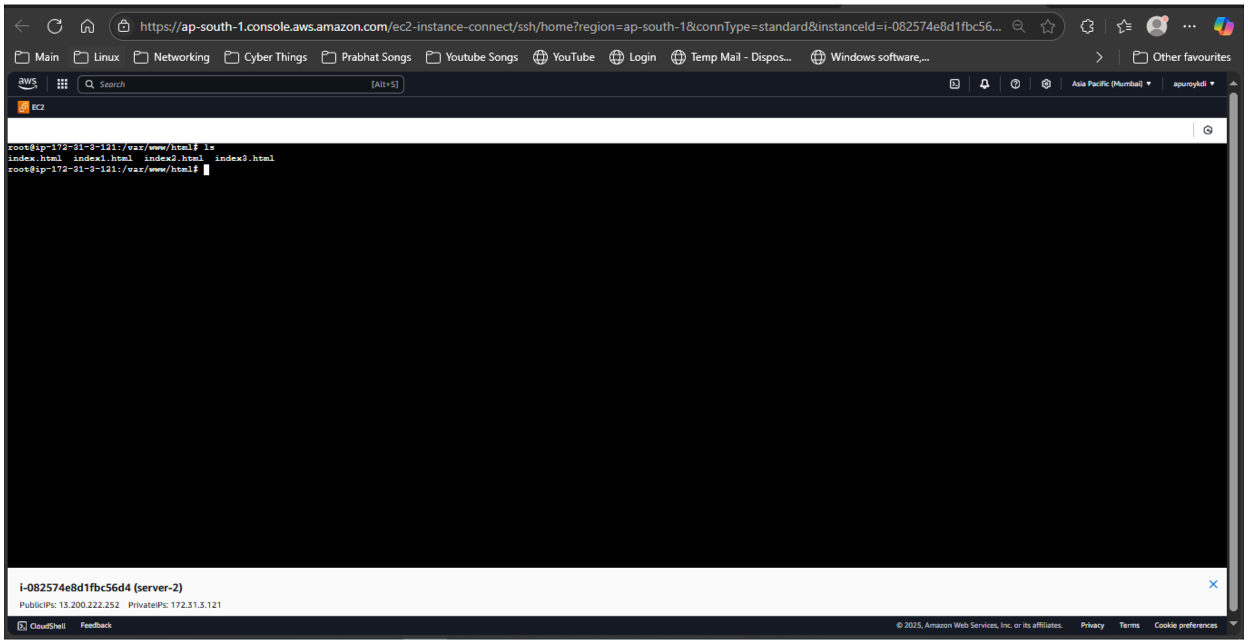
i-03f55c831088e269f (server-1)
PublicIPs: 15.207.221.15   PrivateIPs: 172.31.14.173

15.207.221.15 This is the Public address of server 1

Debian or Ubuntu based:
$sudo su —------ (For root access)

#apt update -y   —-- (To update the system .)
#apt install apache2 —--- (To install apache2 server into the systemc)
#stemctl status apache2 —--(To check the status the service is running or not)
#systemctl start apache2 —-- (To start the service if not started)
#cd /var/www/html —--- (The default location of apache server where we need to add the web pages)
#echo “<head>This is 1st page from the server 1</head>” > index1.html —--- (First page with the extension of .html)
#echo “<head>This is 2nd page from the server 1</head>” > index2.html —--- (Second page with the extension of .html)
#echo “<head>This is 3rd page from the server 1</head>” > index3.html —--- (Third page with the extension of .html)
#systemctl restart apache2 —-- (To restart the service to the updation and execution of the pages into the web server)
#

2nd Server is Ubuntu:

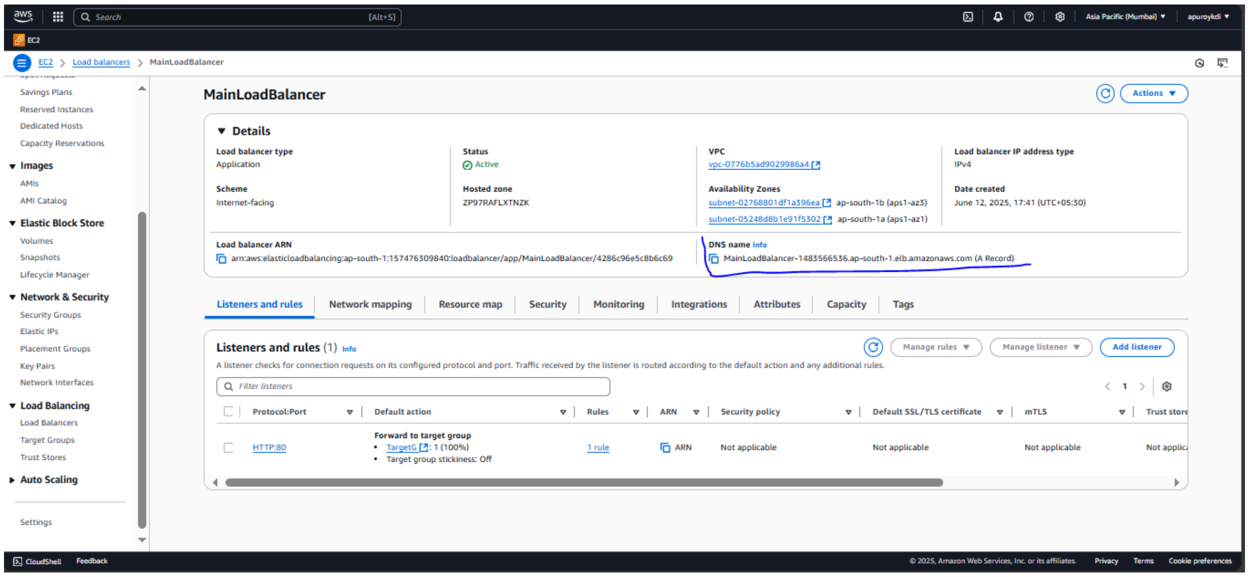


```
root@ip-172-31-3-121:/var/www/html# ls
index.html  index1.html  index2.html  index3.html
root@ip-172-31-3-121:/var/www/html# 
```

i-082574e8d1fbc56d4 (server-2)
PublicIPs: 13.200.222.252   PrivateIPs: 172.31.3.121

13.200.222.252 This is the Public address of Server 2

The Dashboard of Load Balancers:



MainLoadBalancer

Details

Load balancer type Application	**Status** Active	**VPC** vpc-0776b5ad9029986a4	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** ZP97RAFLXTNZK	**Availability Zones** subnet-02768801df1a396ea ap-south-1b (aps1-az3); subnet-05248d8b1e91f5302 ap-south-1a (aps1-az1)	**Date created** June 12, 2025, 17:41 (UTC+05:30)

Load balancer ARN arn:aws:elasticloadbalancing:ap-south-1:157476309840:loadbalancer/app/MainLoadBalancer/4286c96e5c8b6c69

DNS name MainLoadBalancer-1483566536.ap-south-1.elb.amazonaws.com (A Record)

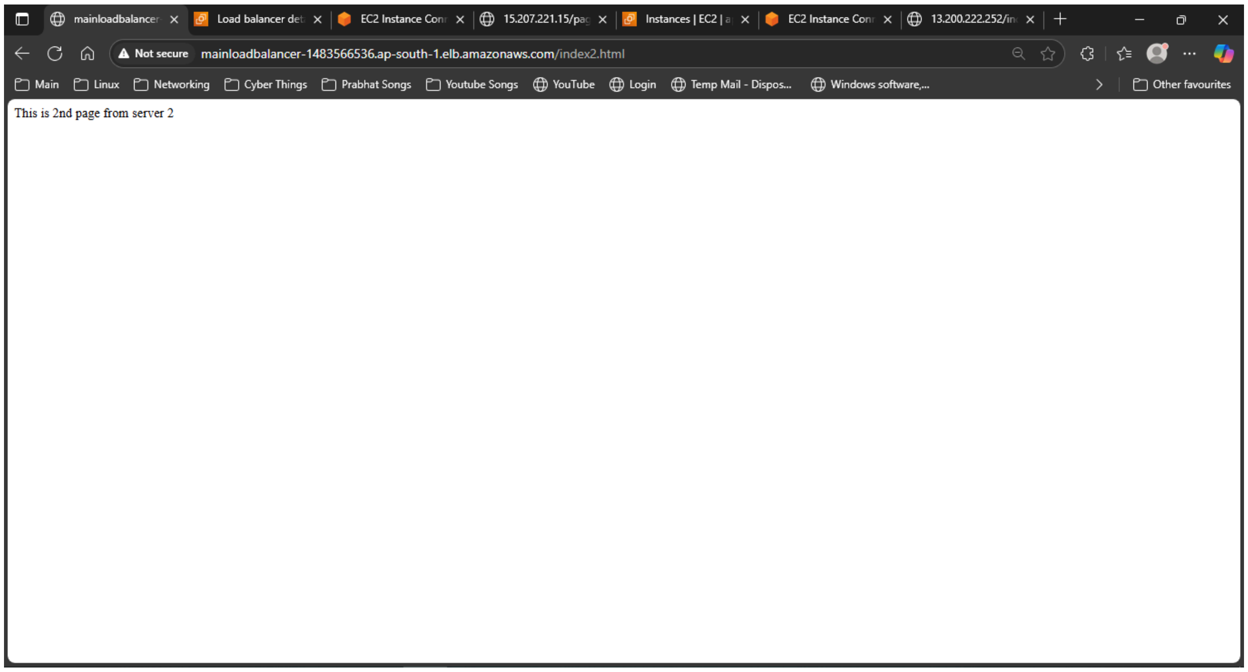
Listeners and rules (1)

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	Forward to target group • TargetG: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applic

Here in the above picture , we can see lots of Links , each link is used for individual purposes. So for our case , we just take the **DNS** for balancing the loader . MainLoadBalancer-1483566536.ap-south-1.elb.amazonaws.com By this link we can now access each page of that two servers's web pages . just suffixing with the name of the page we want to access . as shown below:

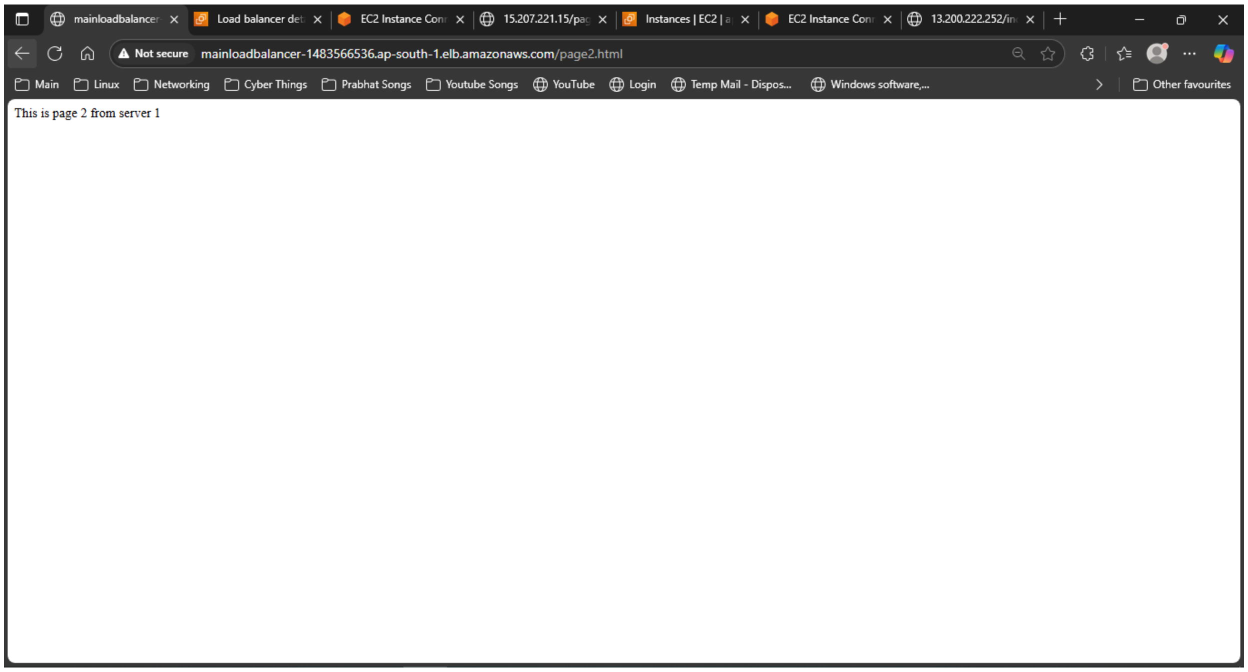
By suffixing with '/index2.html' we can access data from server-2.



mainloadbalancer-1483566536.ap-south-1.elb.amazonaws.com/index2.html

This is 2nd page from server 2

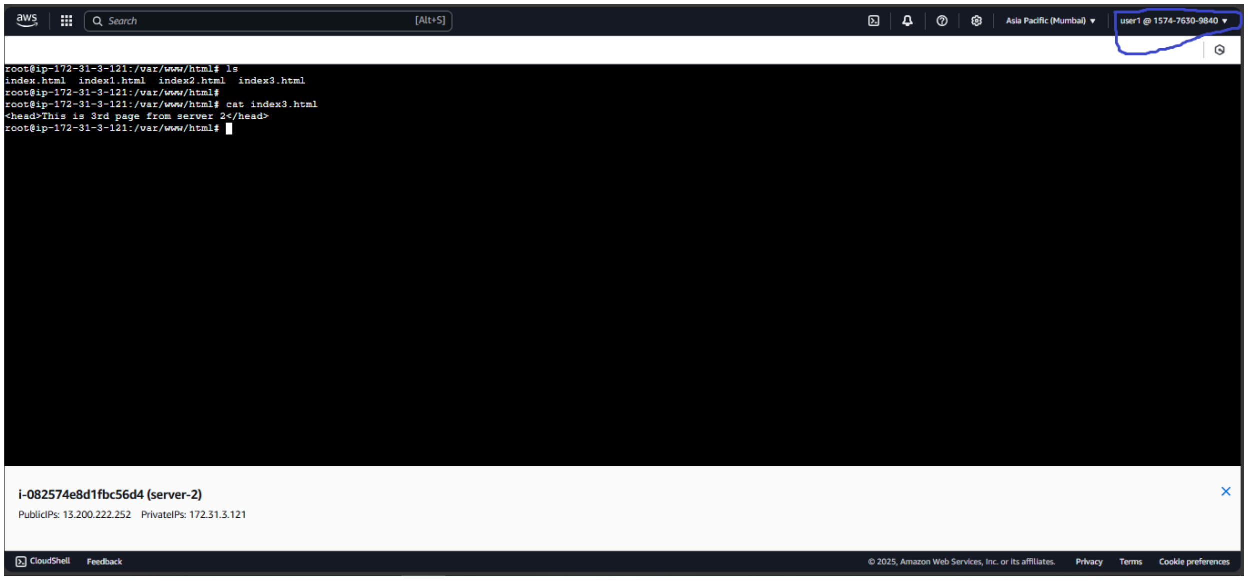
By suffixing with ‘/page2.html’ we can access the data from server-1


Not secure
mainloadbalancer-1483566536.ap-south-1.elb.amazonaws.com/page2.html

This is page 2 from server 1

Every configuration is completed now let's check the permission by the **user1** and the **user2.**

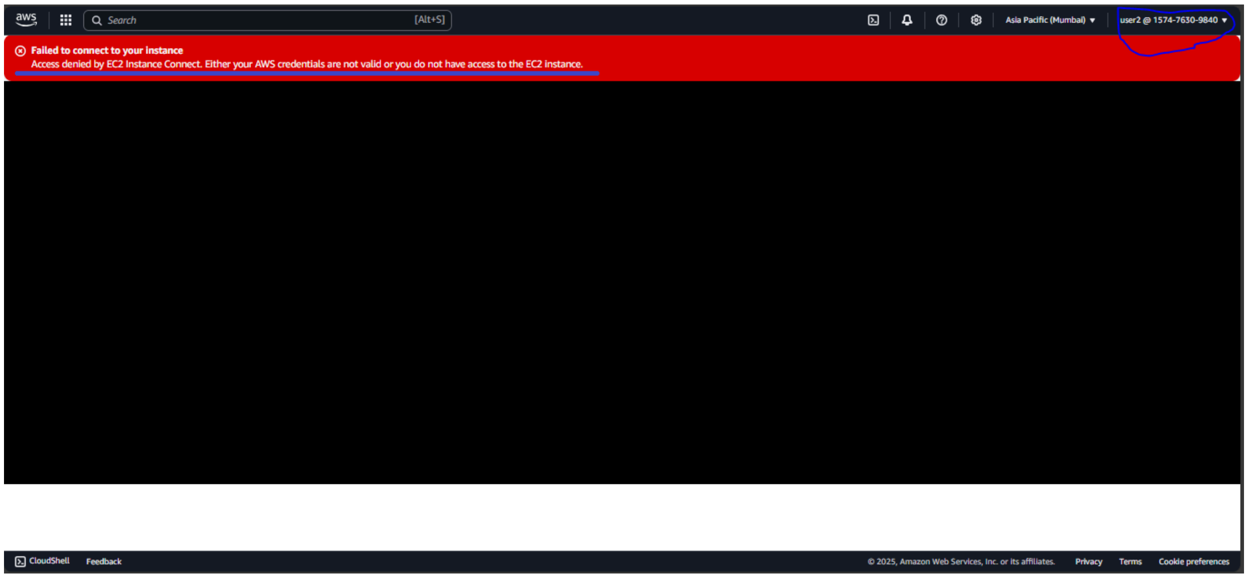
User1: user1 has successfully accessed the terminal of the server-1. user1 has the full permission as Admin.



```
root@ip-172-31-3-121:/var/www/html# ls
index.html  index1.html  index2.html  index3.html
root@ip-172-31-3-121:/var/www/html#
root@ip-172-31-3-121:/var/www/html# cat index3.html
<head>This is 3rd page from server 2</head>
root@ip-172-31-3-121:/var/www/html#
```

i-082574e8d1fbc56d4 (server-2)
PublicIPs: 13.200.222.252 PrivateIPs: 172.31.3.121

User2: user2 has no permission to access the terminal of any server, but has the read-only permission.



Failed to connect to your instance

Access denied by EC2 Instance Connect. Either your AWS credentials are not valid or you do not have access to the EC2 instance.

<u>**The End**</u>