

# GLA UNIVERSITY



Accredited with **A+** Grade by **NAAC**

12-B Star Accredited

## Computer Science and Engineering Department.

### ❖ Project Report

- Course – JOVAC AWS

Supervisor – Raushan Kumar Singh

Ruby Panwar

### Submitted by

Kuldeep Sharma (2215000956)

Anamika Singh (2215000221)

Tanish Gupta (2215001829)

Vishnu Upadhyay (2215001991)

Sandeep Kumar (2215001573)



# Topics of Project

## Project Title

**Create an autoscaling Group and put a load balancer and test it.**

## Objective/Problem statement

**how to make cloud environment scalable in any situation.**

---

## **STEPS FOR CREATING PROJECT TITLE ON AWS.**

- Create two server/instances.
- Create target group and connect server.
- Create Load balancer.
- Create Launch template.
- Create Auto scaling group.

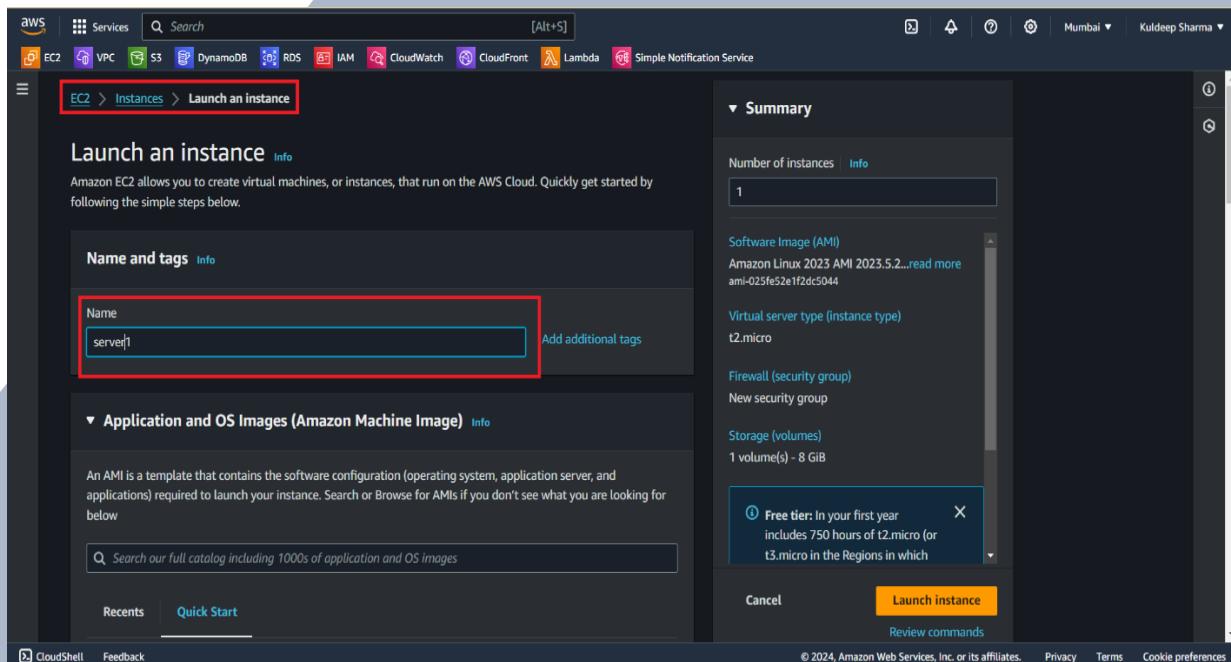
## **TECHNOLOGY USED:**

- AWS EC2
- Load balancer / Auto Scaling group

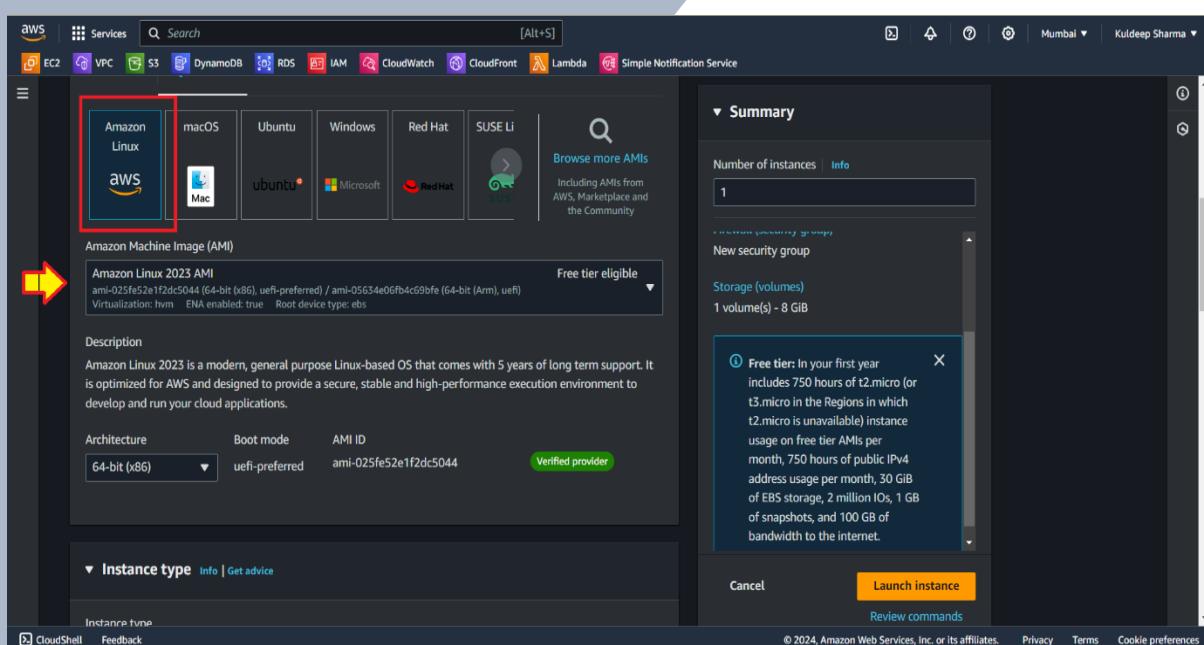
# ➤ Steps to create instances.

---

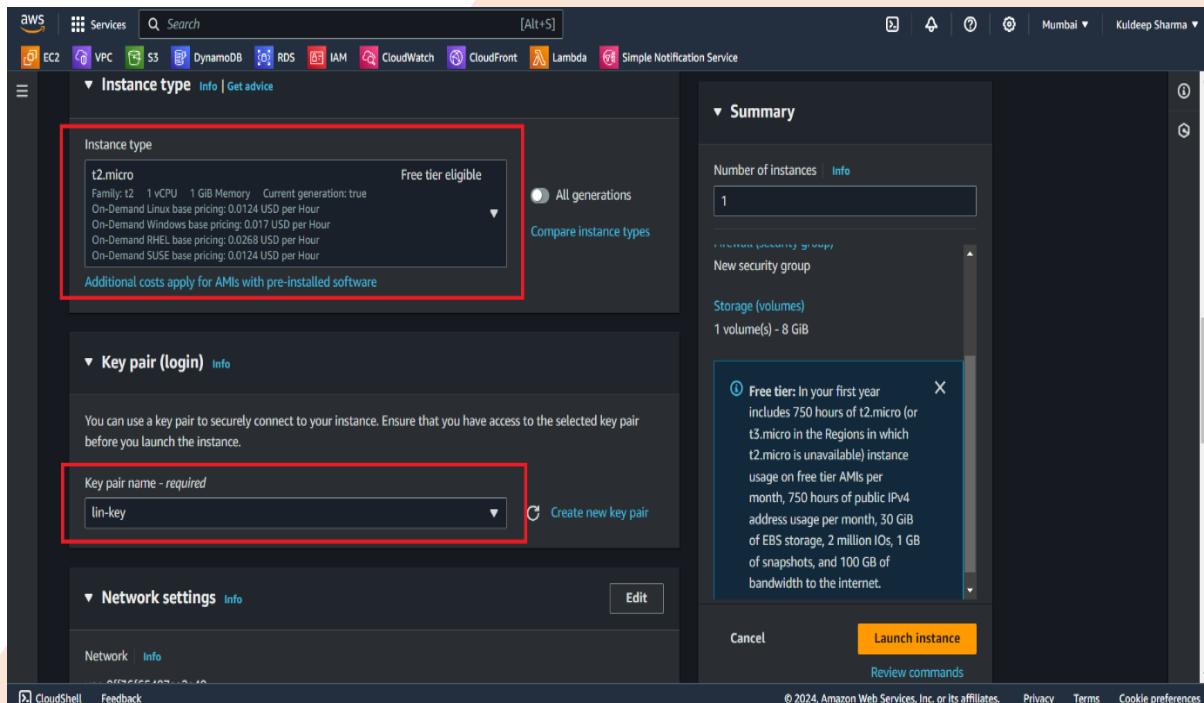
- Goto EC2 select instances and click on launch instances Enter server name.



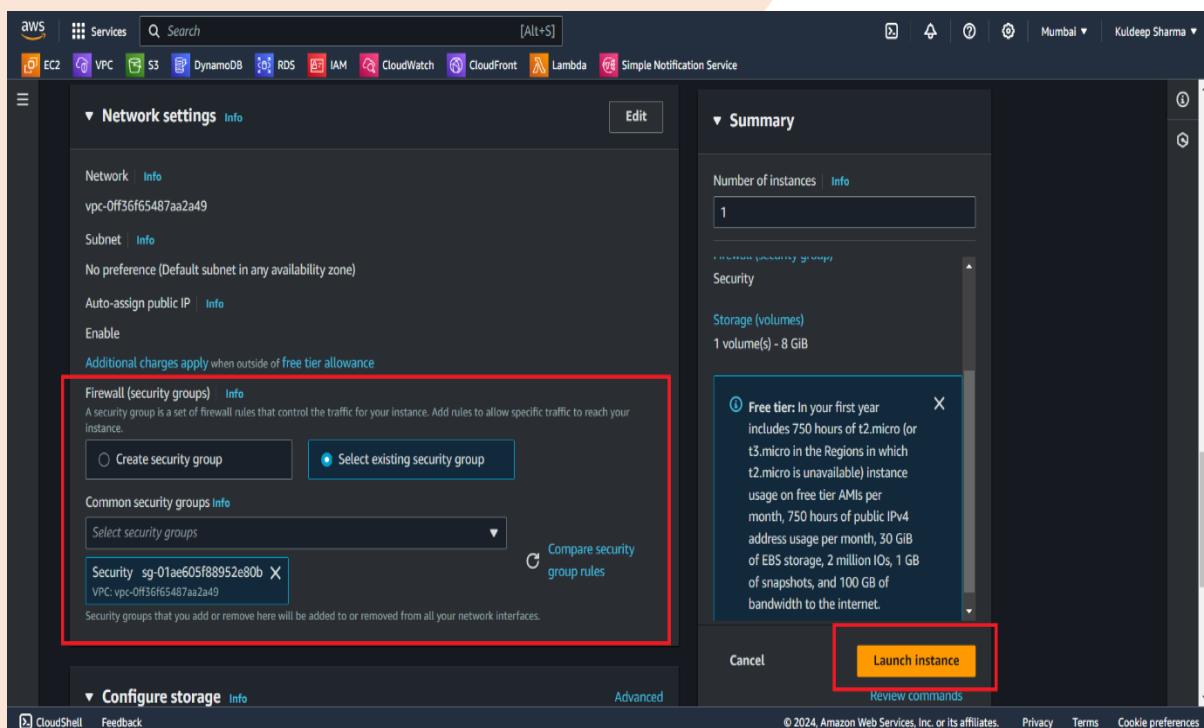
- Select Amazon Linux and AMI.



- Select instance type t2micro and key pair ppk format.



- Select Security Group and click on launch instances.



## ○ Server is running

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists navigation options: EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups). The main content area displays a table titled 'Instances (1/2) Info' with two rows. A red box highlights the table header and the first row. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. The first row (server1) has no checked checkbox. The second row (server2) has a checked checkbox. The instance details for server2 are shown in a modal window below:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
server1	i-0c14401437c5fa2d1	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-13-23-
server2	i-09ffa1e2275115342	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-3-110-

**Modal Window (i-09ffa1e2275115342 (server2)):**

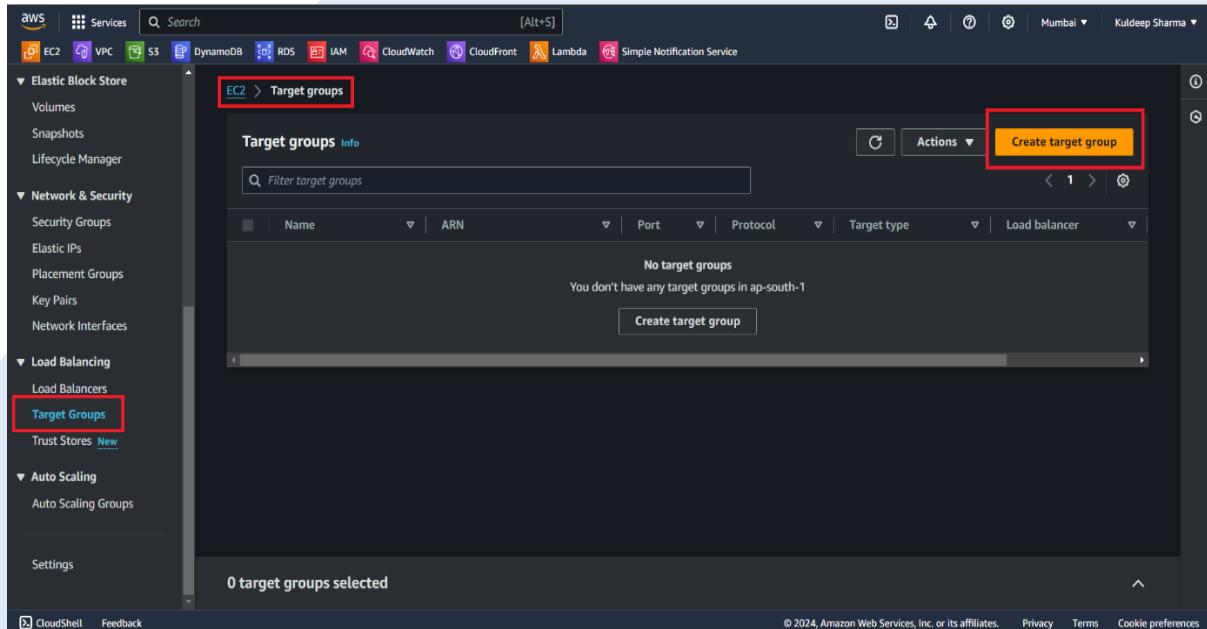
Details			Status and alarms	Monitoring	Security	Networking	Storage	Tags	
<b>Instance summary</b>									
Instance ID	i-09ffa1e2275115342 (server2)		Public IPv4 address	Private IPv4 addresses					
IPv6 address	-		3.110.190.156   <a href="#">open address</a>	172.31.38.29					
Hostname type			Instance state	Public IPv4 DNS					
			Running	ec2-3-110-190-156.ap-south-1.compute.amazonaws.com   <a href="#">open address</a>					
			Private IP DNS name (IPv4 only)						

At the bottom of the page, there are links for CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

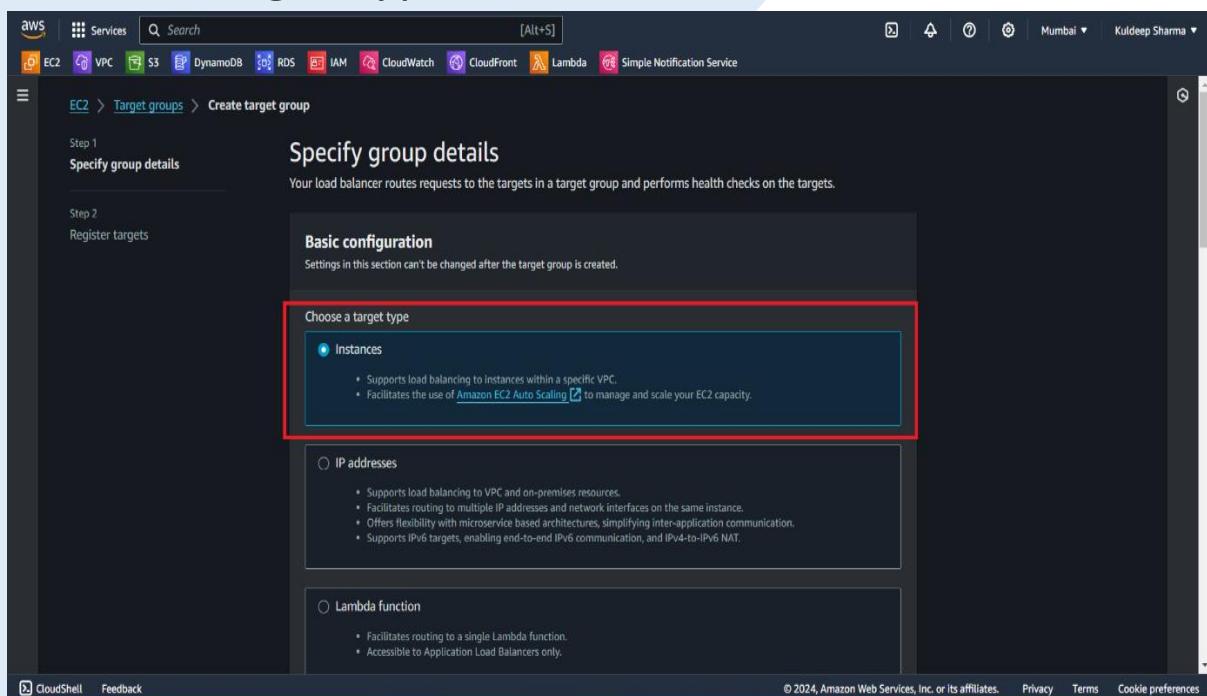
## ➤ Steps to create Target group.

---

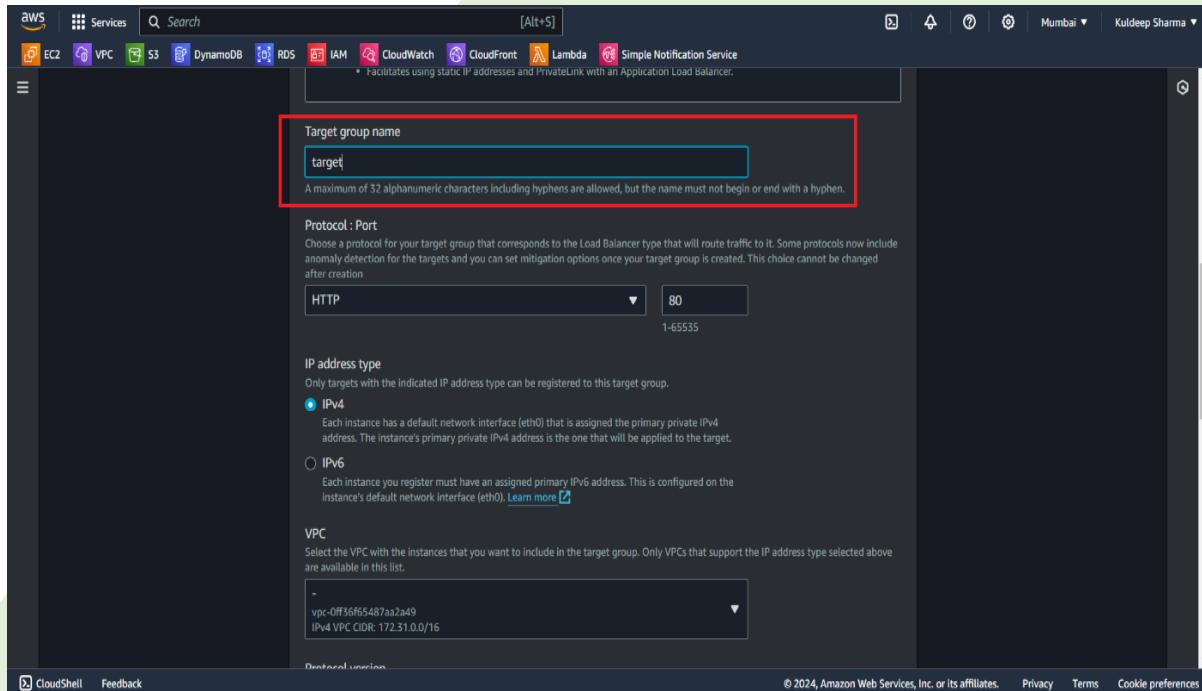
- Goto EC2 select target group and click create target group.



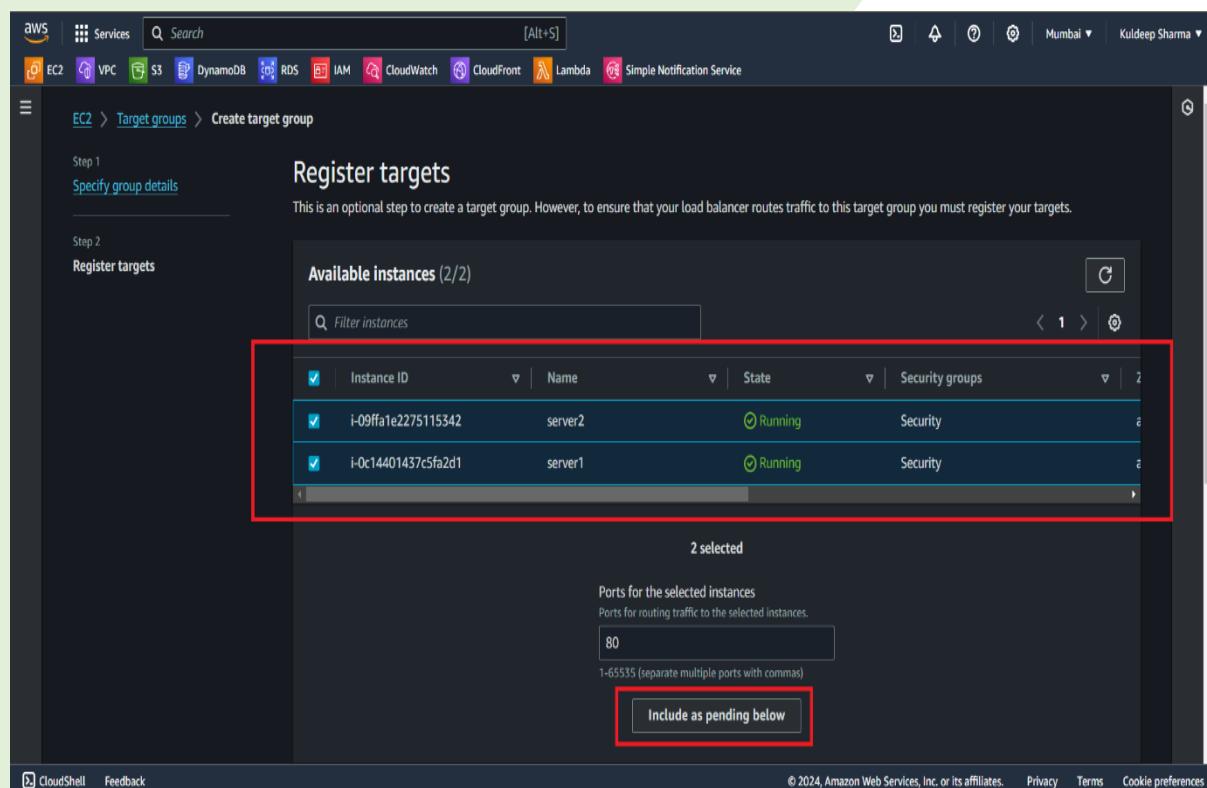
- Choose target type instances.



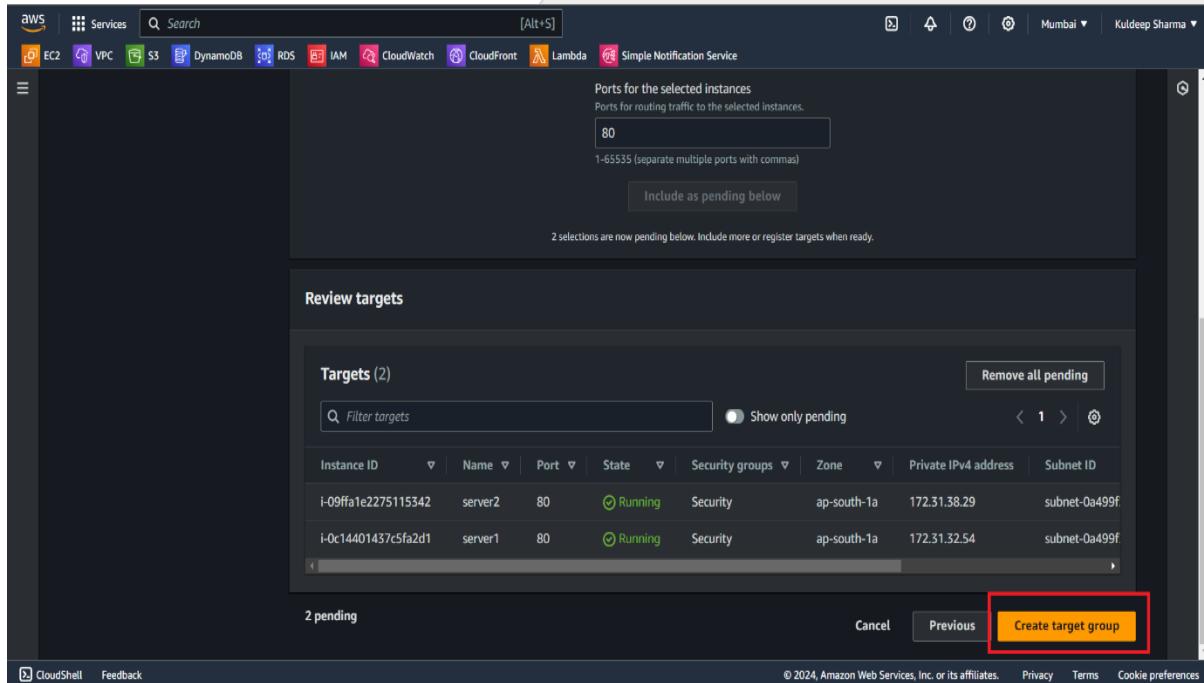
- Enter target group name target.



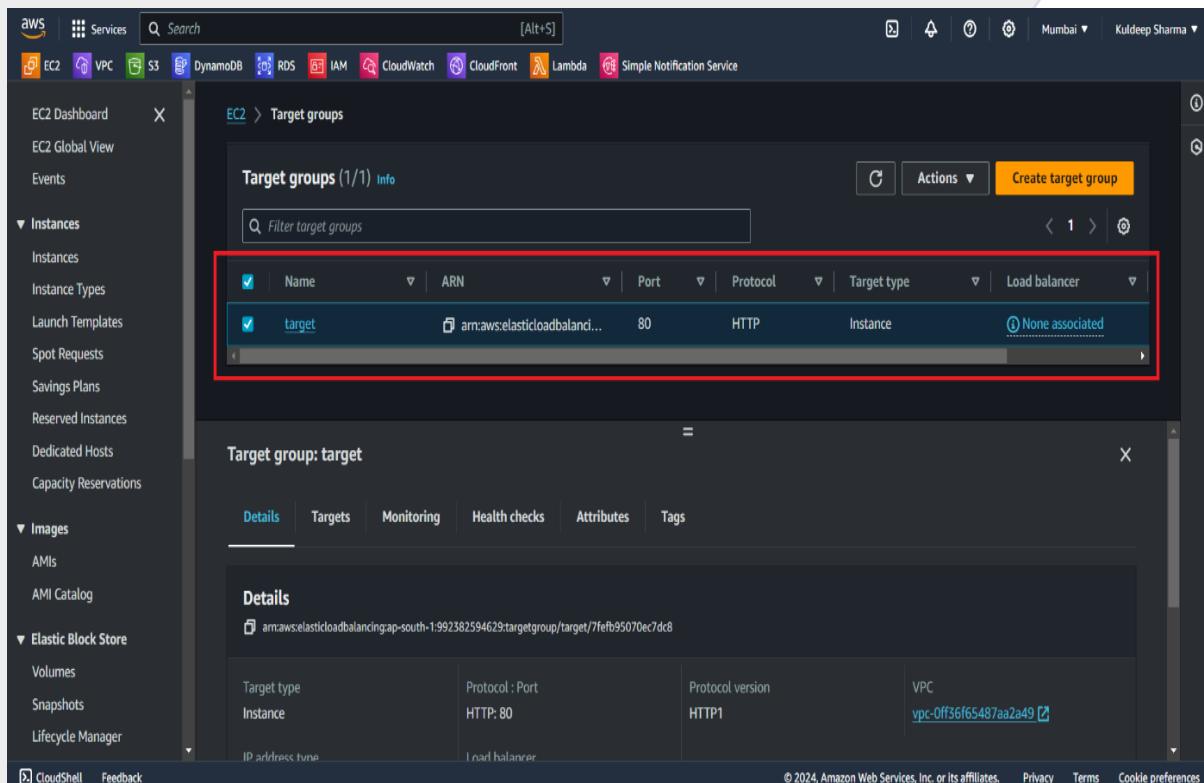
- Select all the instances click on include as pending below



- Click on create Target group



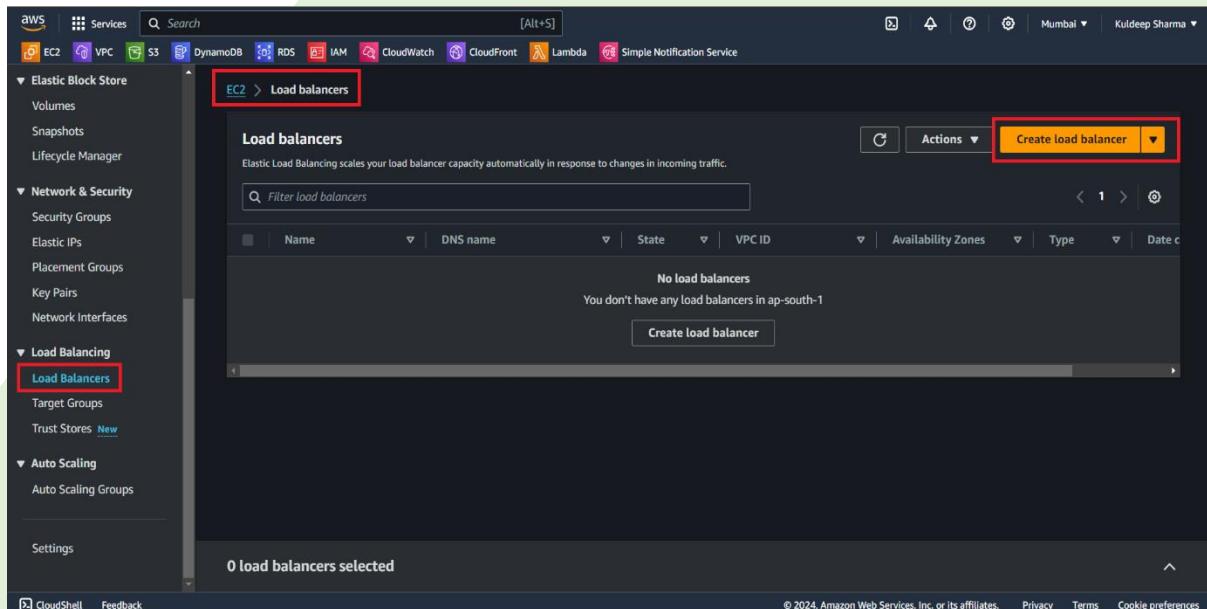
- Target group successfully created.



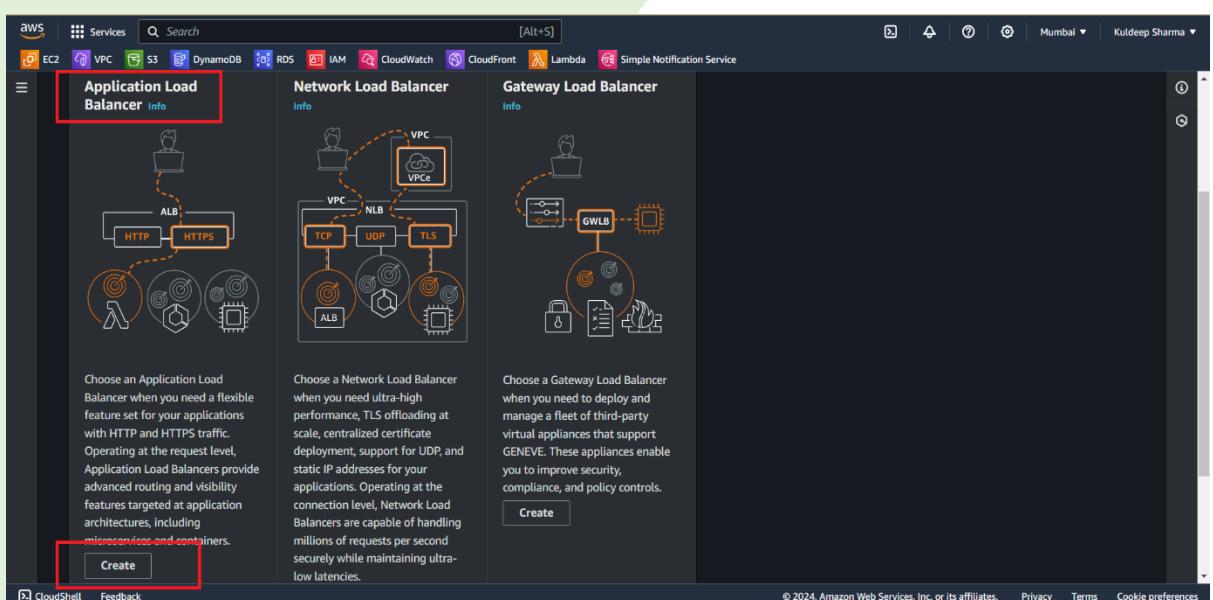
## ➤ Steps to create Load balancer.

---

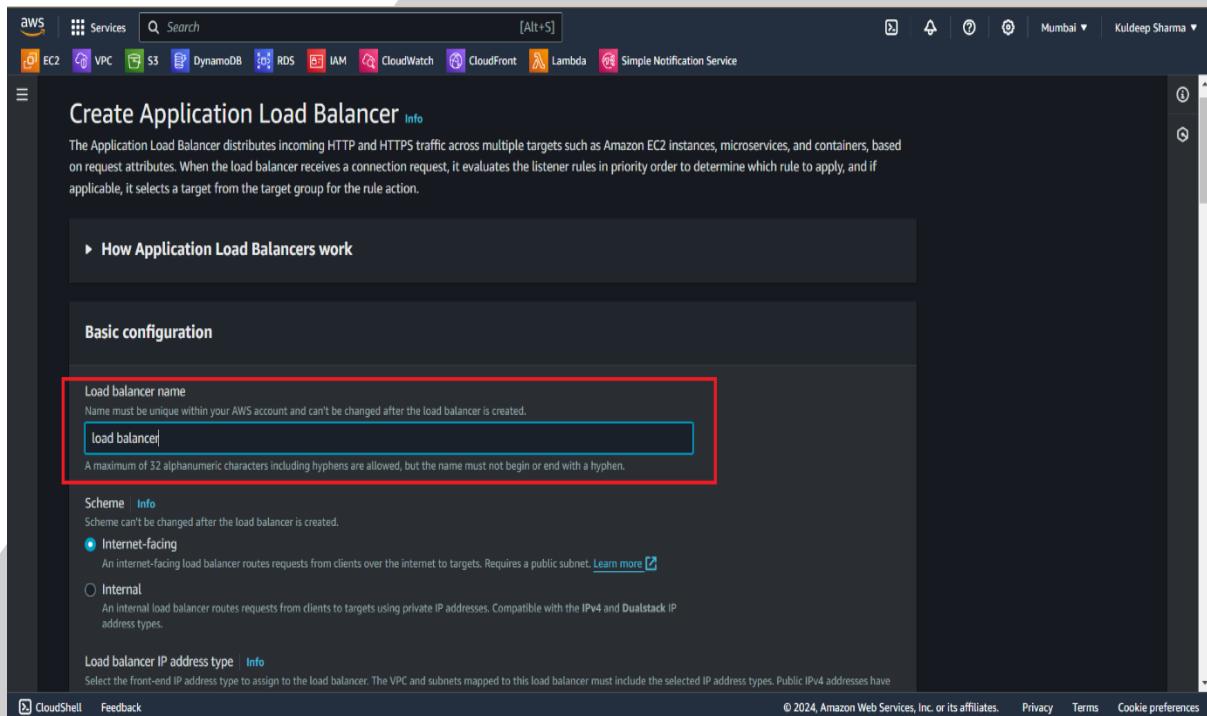
- Goto EC2 select load balancer and click on create load balancer.



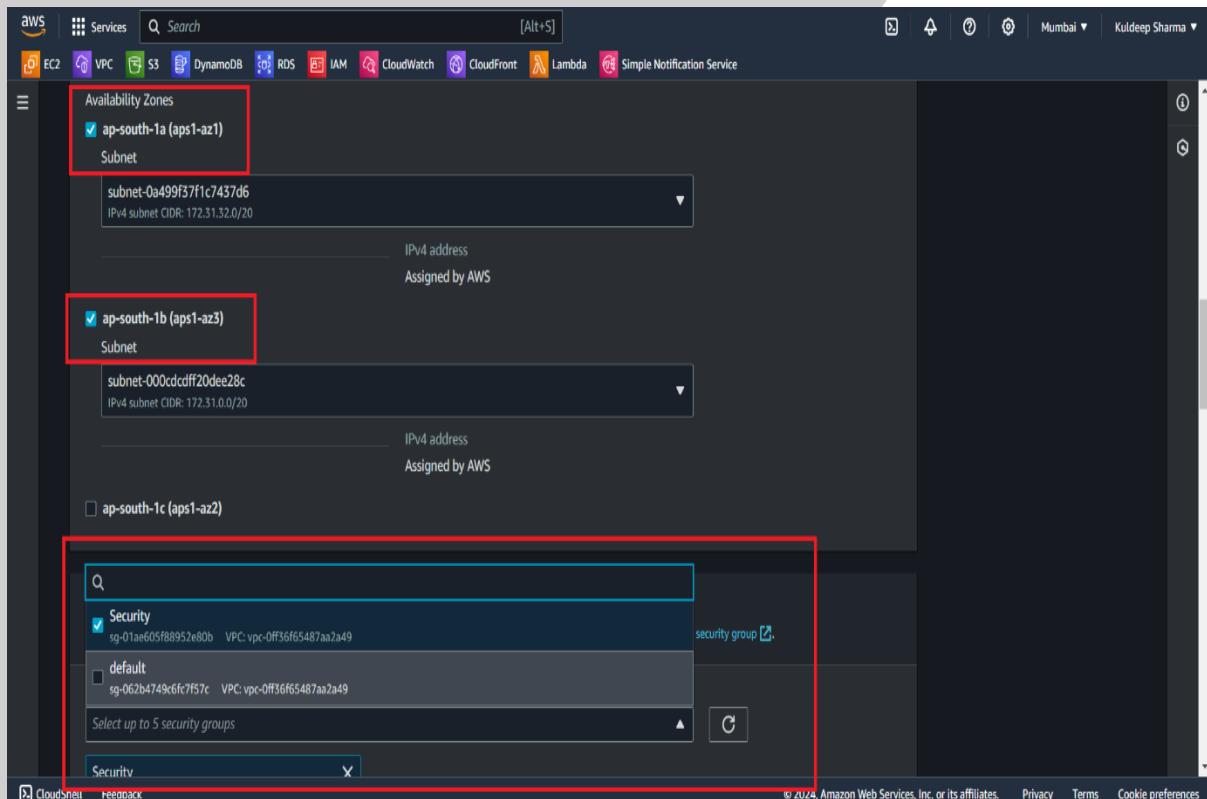
- Select Application Load Balancer and click on create



- Enter the name of load balancer.



- Select availability zones and security group.



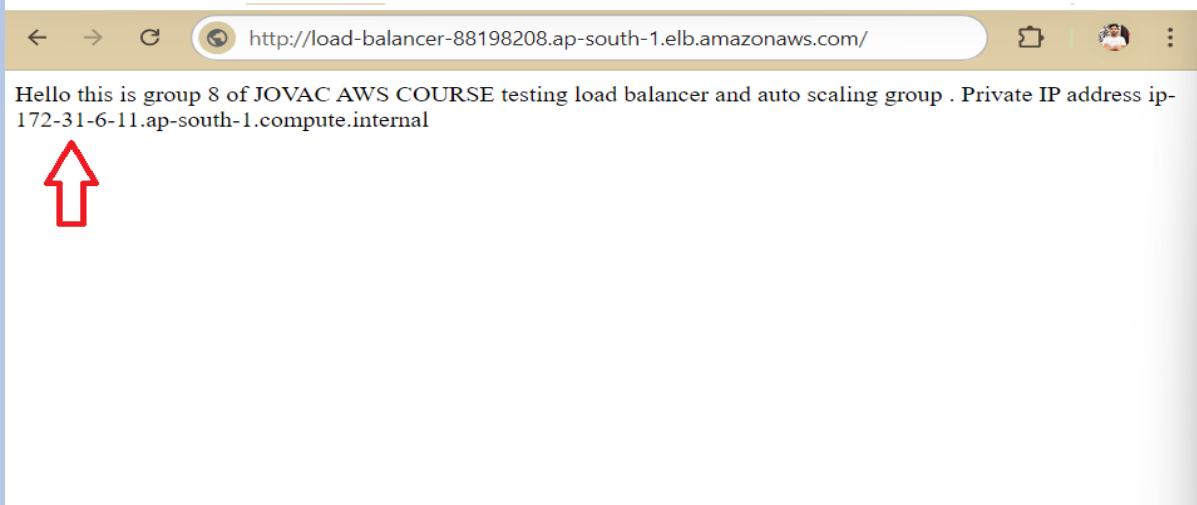
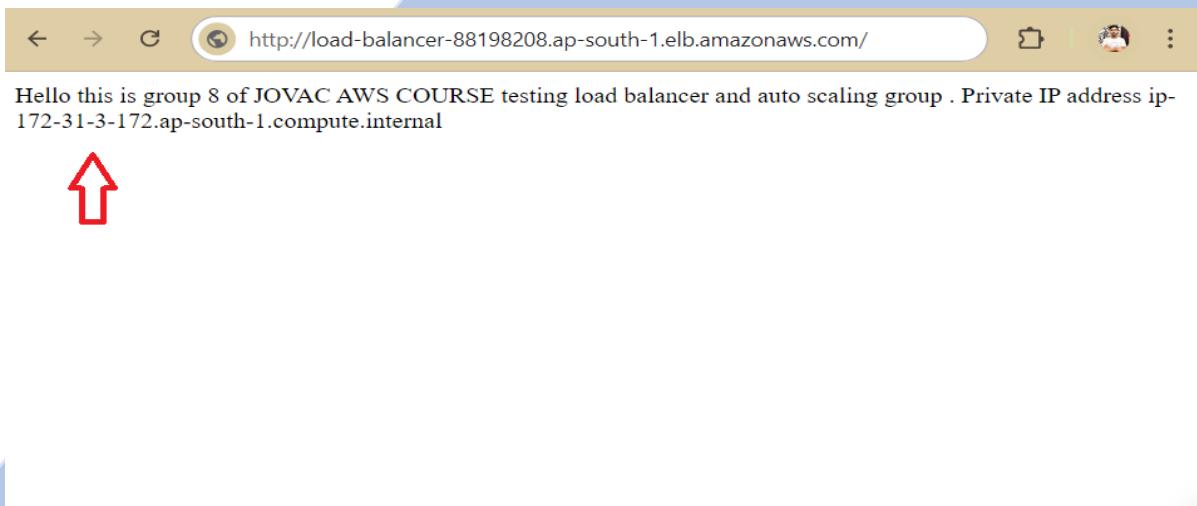
- Select target group.

A screenshot of the AWS CloudFront console. The top navigation bar shows services like EC2, VPC, S3, DynamoDB, RDS, IAM, CloudWatch, CloudFront, Lambda, and Simple Notification Service. The main area is titled 'Listeners and routing' under 'Info'. A red box highlights the 'Forward to' dropdown menu, which contains 'Select a target group' and 'Create target'. Below this, there's a section for 'Listener tags - optional' and another for 'Load balancer tags - optional'. At the bottom, there are buttons for 'Add listener tag' and 'Add listener'.

- Load balancer is created and DNS link is active.

A screenshot of the AWS EC2 Load Balancers console. The left sidebar shows navigation paths: Elastic Block Store > Volumes, Network & Security > Security Groups, Placement Groups, Key Pairs, Network Interfaces, Load Balancing > Load Balancers, Target Groups, Trust Stores, Auto Scaling > Auto Scaling Groups, and Settings. The main area shows a table of 'Load balancers (1/1)'. A red box highlights the row for 'load-balancer', which has a status of 'Active', a VPC ID of 'vpc-0ff36f65487aa2a49', and three availability zones. Below this, a detailed view for 'Load balancer: load-balancer' is shown, with tabs for Details, Listeners and rules, Network mapping, Resource map - new, Security, Monitoring, Integrations, Attributes, and Tags. The 'Details' tab is selected, displaying information such as Load balancer type (Application), Status (Active), VPC (vpc-0ff36f65487aa2a49), and Load balancer IP address type (IPv4).

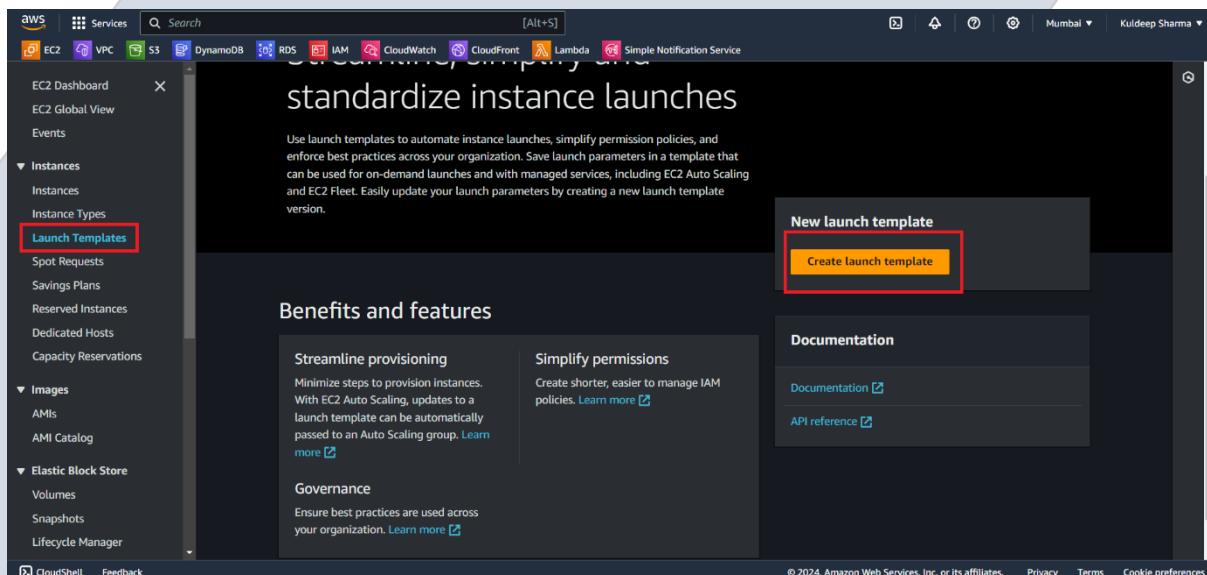
- After refreshing the DNS link server request is changing.



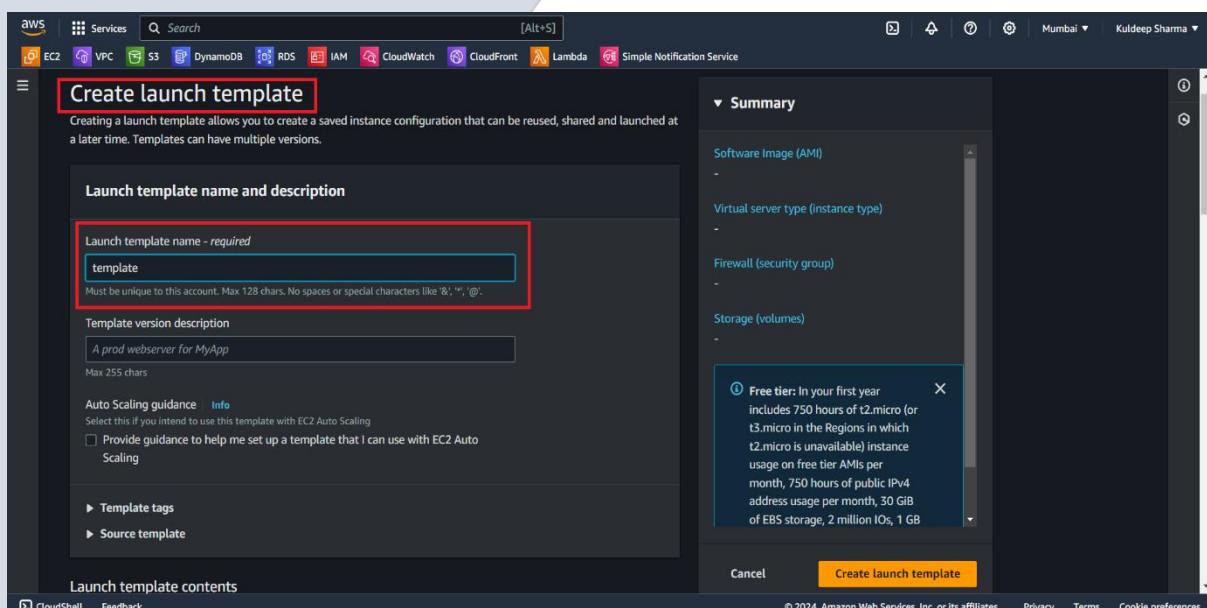
# ➤ Steps to create launch templates.

---

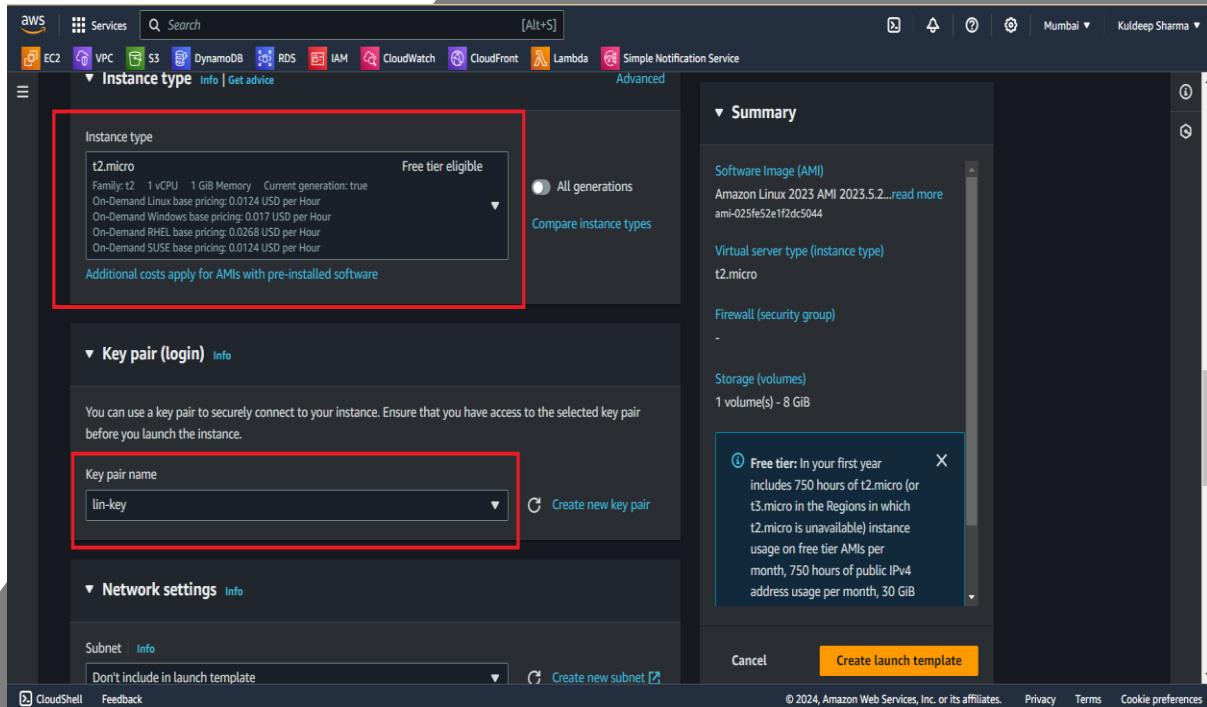
- Goto EC2 select launch template and click on create launch template.



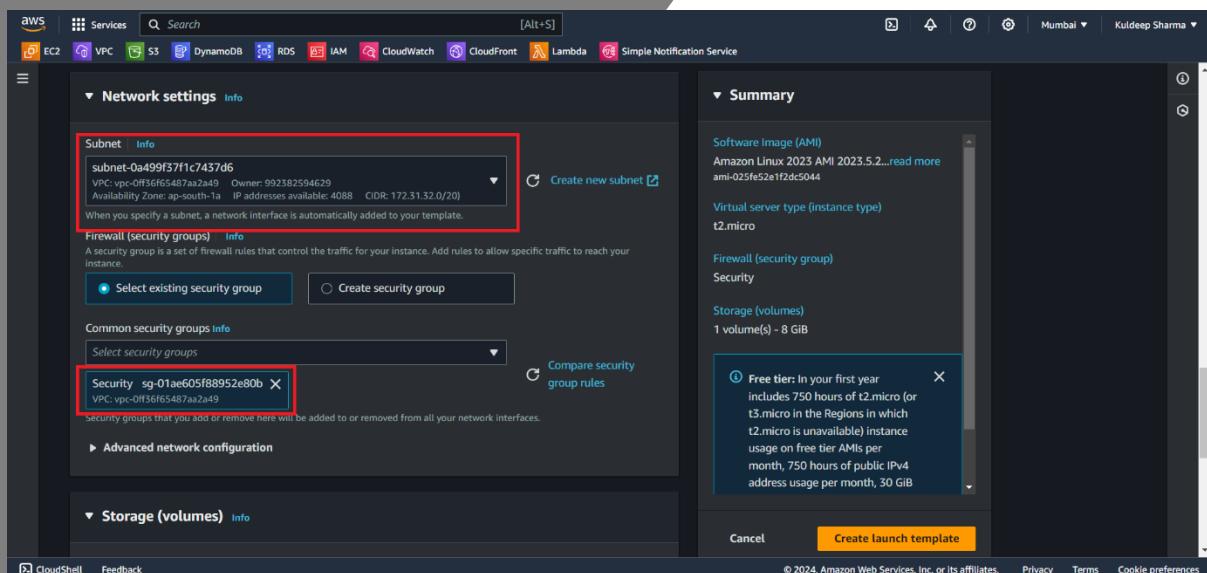
- Enter template name.



- Select instance type and Key pair.



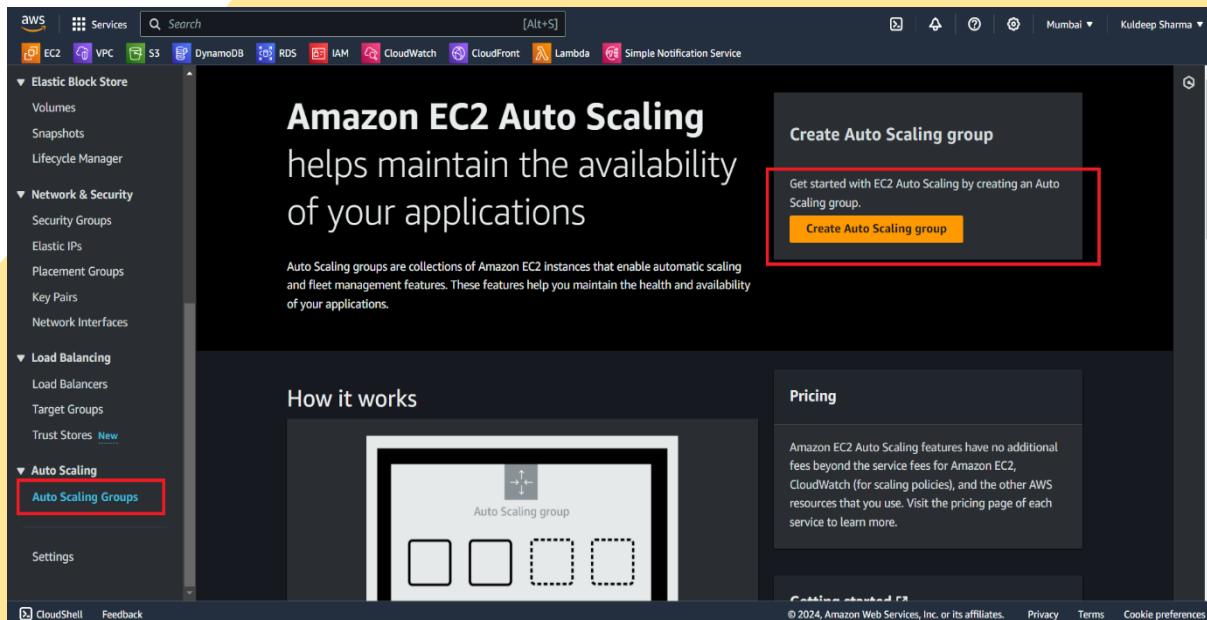
- Select subnet and security group and click on create launch template.



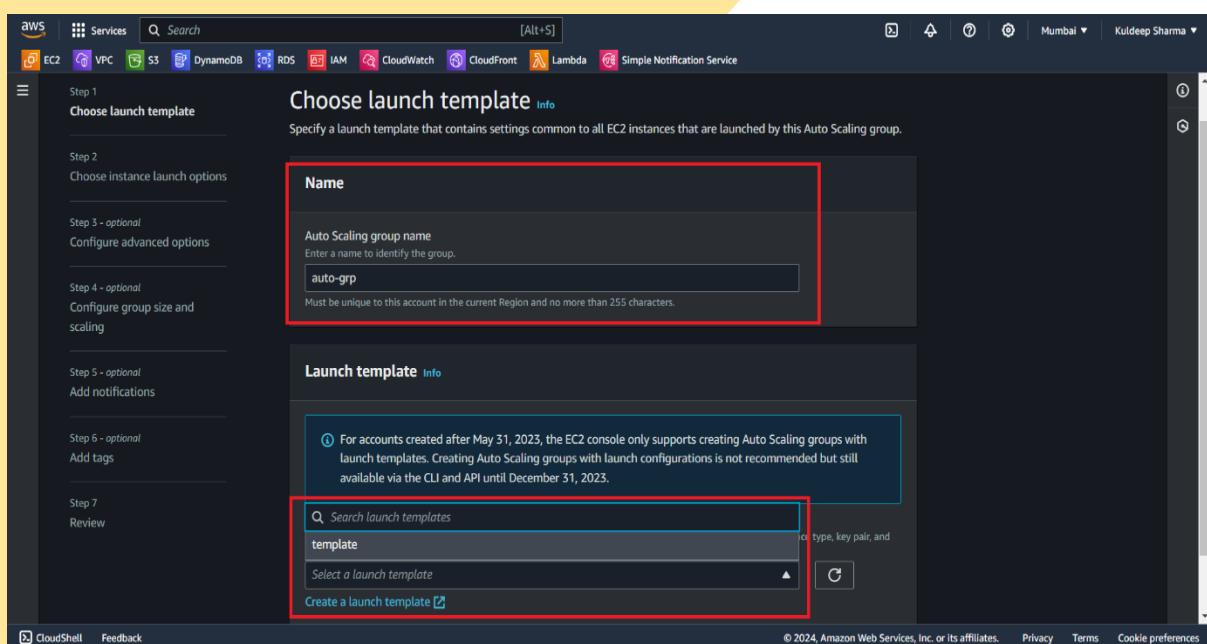
# ➤ Steps to create Auto scaling group.

---

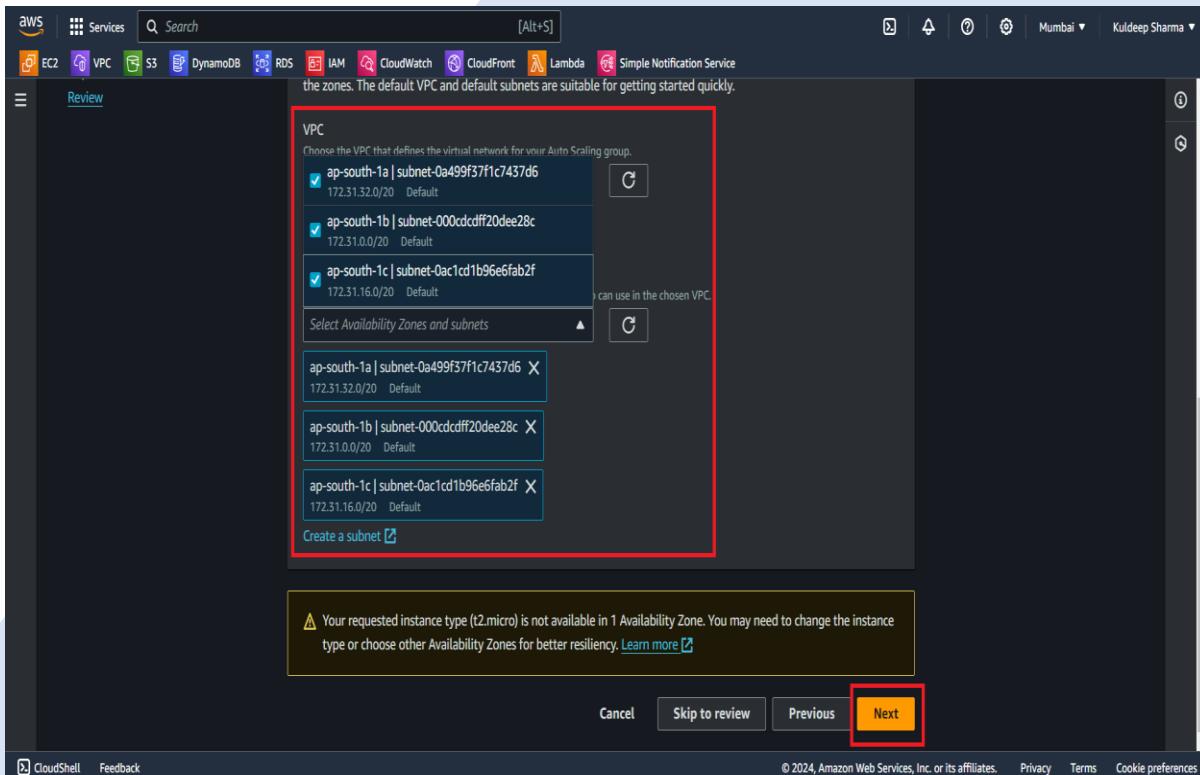
- Goto EC2 select autoscaling and click on create.



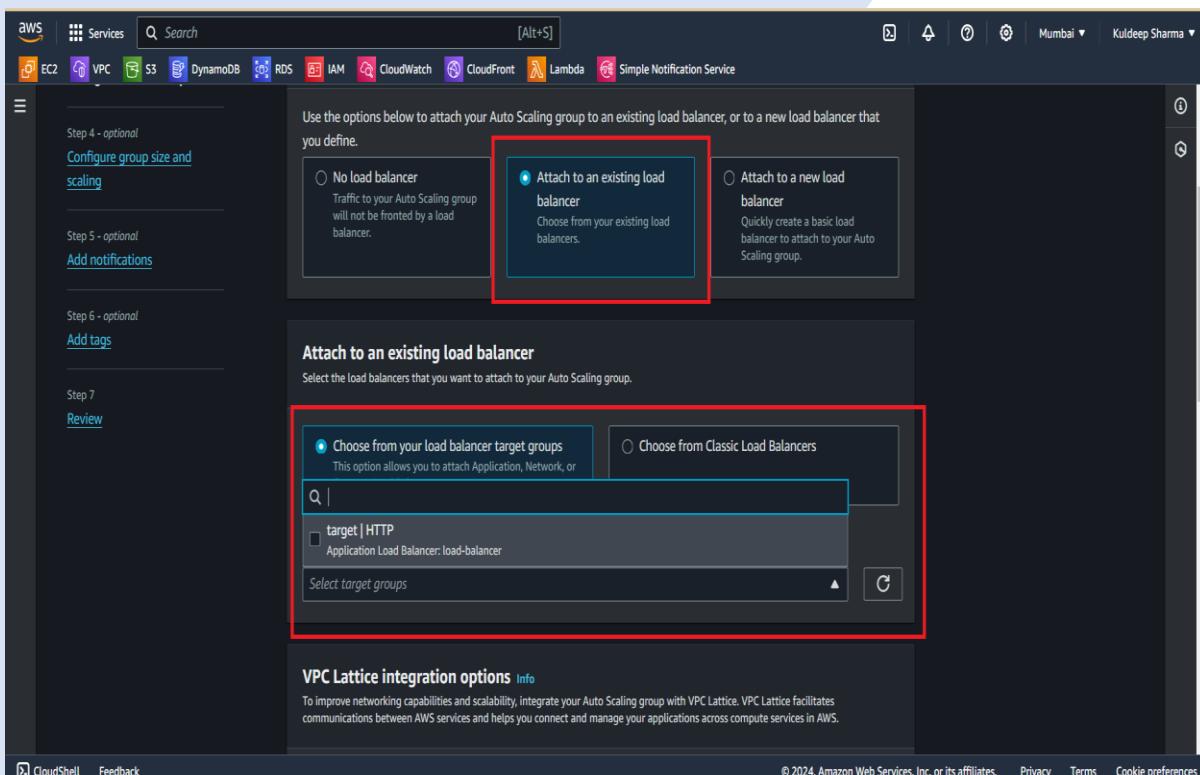
- Enter name and select launch template.



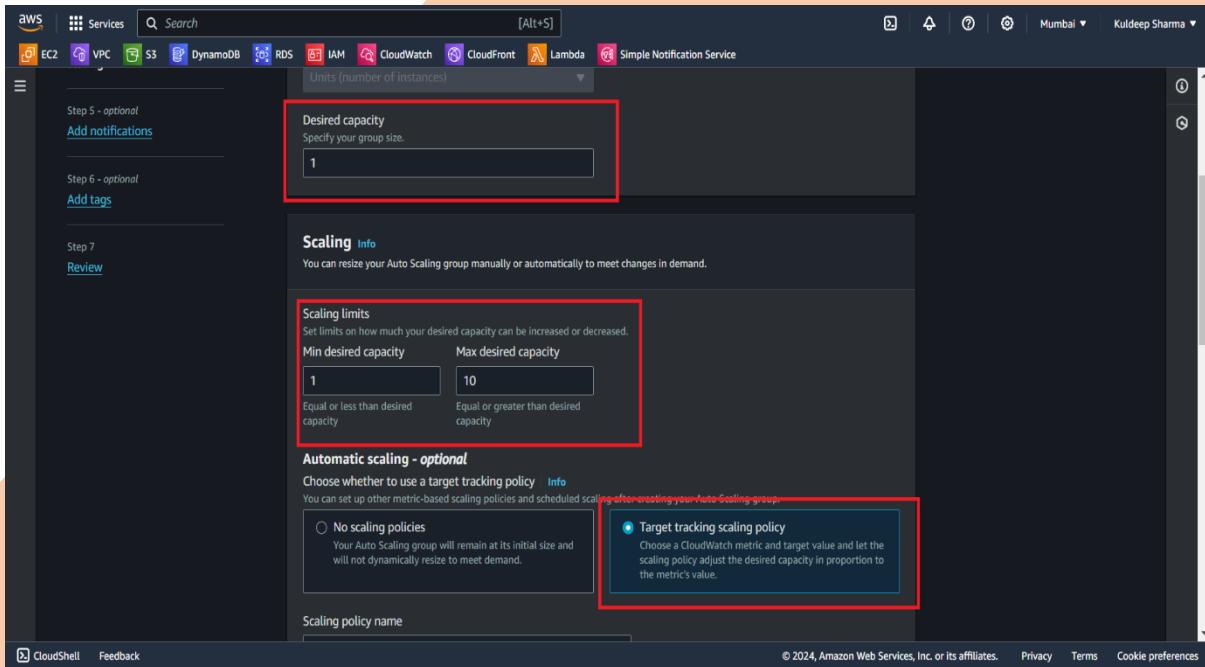
- Select availability zones and click on next.



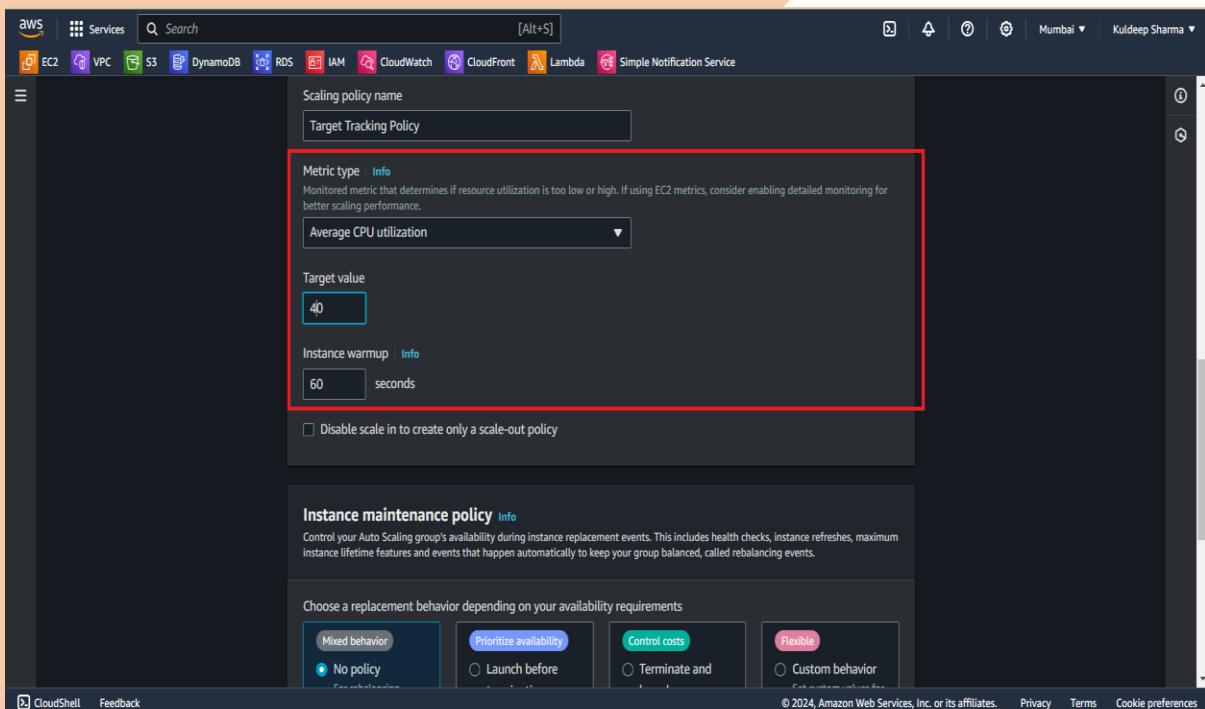
- Select load balancer and target group.



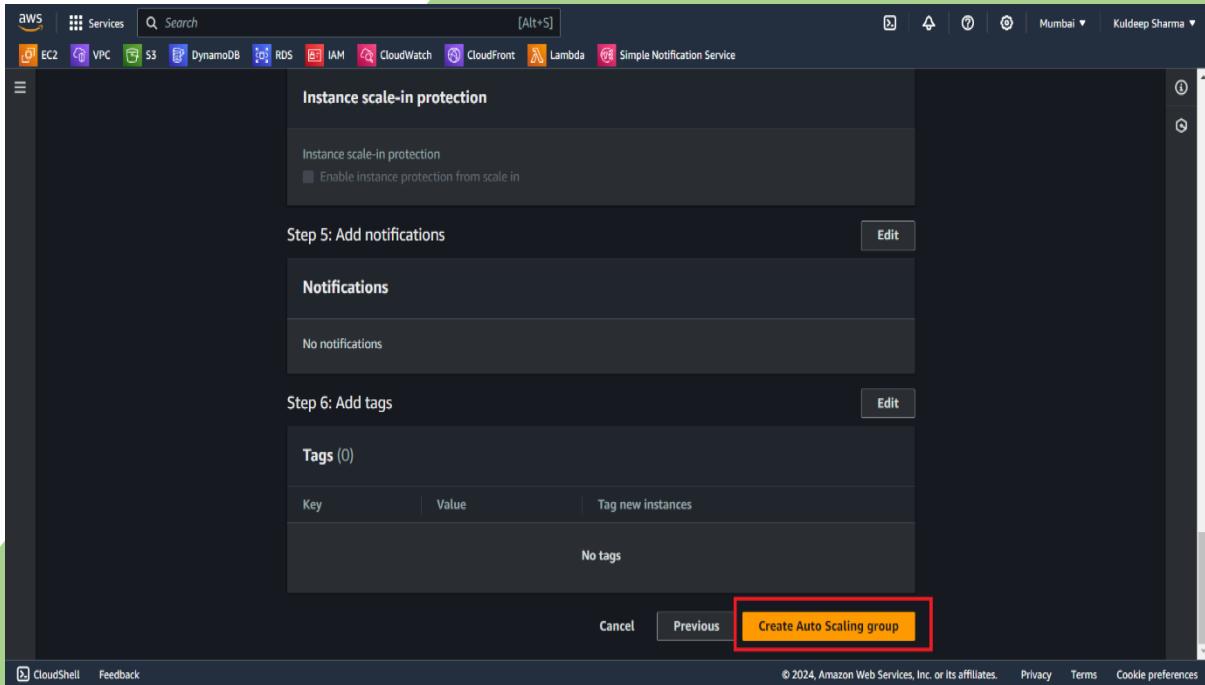
- Select target tracking scaling policy and desired , min and max no. of instances to be created.



- Select CPU utilization and target value and instance warmup.



- Click on create Auto scaling group.

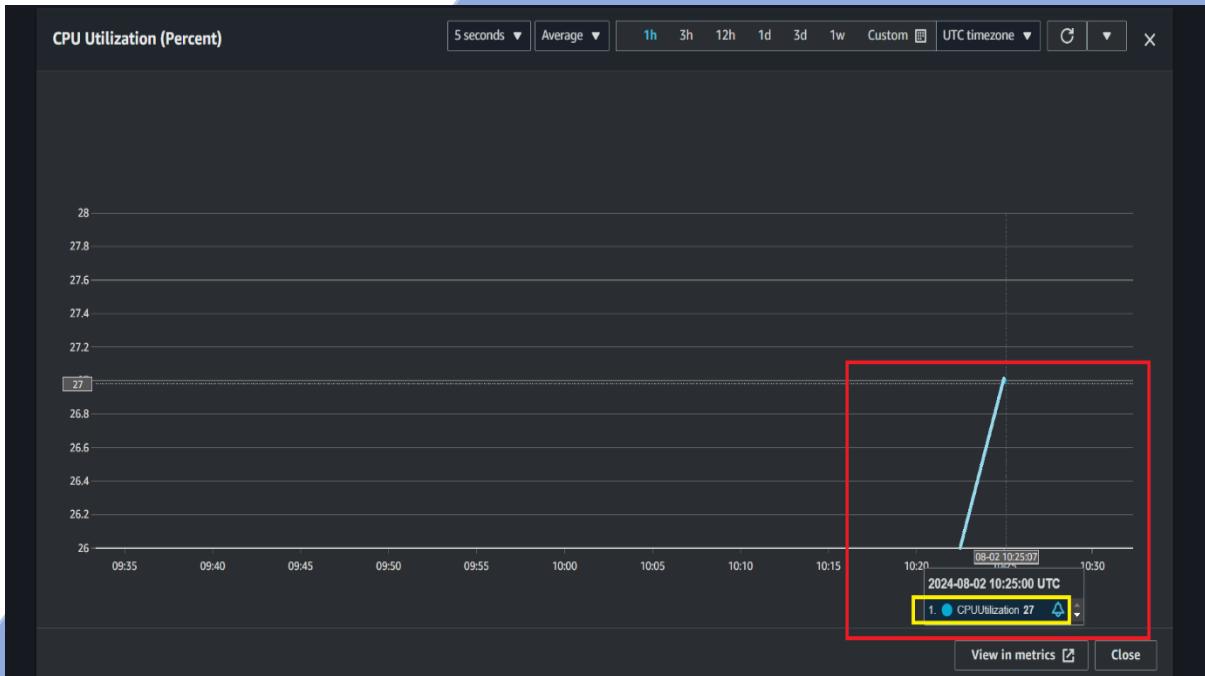


- Stress command running on Server.

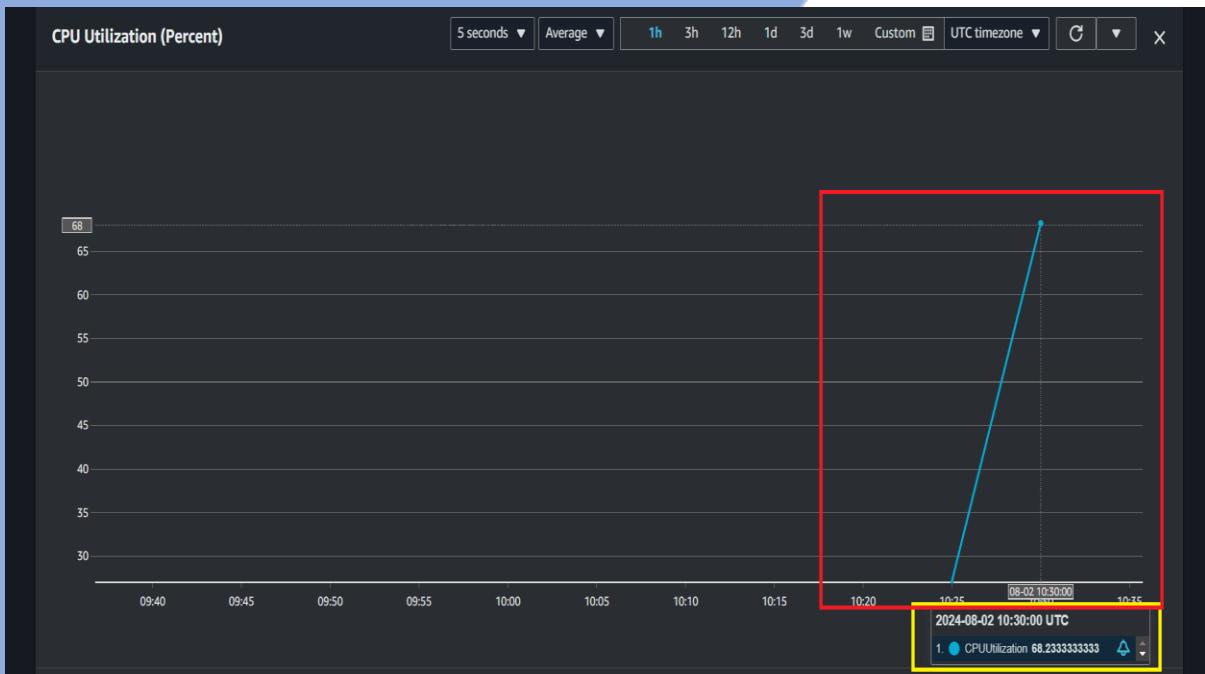
```
top - 10:31:46 up 3 min, 3 users, load average: 0.87, 0.45, 0.19
Tasks: 124 total, 9 running, 115 sleeping, 0 stopped, 0 zombie
%Cpu(s):100.0 us, 0.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 949.5 total, 503.7 free, 149.8 used, 296.0 buff/cache
MiB Swap : 0.0 total, 0.0 free, 0.0 used. 656.4 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
26269 root      20   0   3512   108    0 R 13.3  0.0 0:00.56 stress
26270 root      20   0   3512   108    0 R 13.3  0.0 0:00.55 stress
26271 root      20   0   3512   108    0 R 13.3  0.0 0:00.56 stress
26272 root      20   0   3512   108    0 R 13.3  0.0 0:00.56 stress
26273 root      20   0   3512   108    0 R 13.3  0.0 0:00.56 stress
26274 root      20   0   3512   108    0 R 13.3  0.0 0:00.56 stress
26276 root      20   0   3512   108    0 R 13.3  0.0 0:00.56 stress
26275 root      20   0   3512   108    0 R  6.7  0.0 0:00.55 stress
26277 root      20   0  223920  3224  2692 R  6.7  0.3 0:00.01 top
  1 root      20   0 105664 16800 10064 S  0.0  1.7 0:01.27 systemd
  2 root      20   0      0    0    0 S  0.0  0.0 0:00.00 kthreadd
  3 root      0 -20     0    0    0 I  0.0  0.0 0:00.00 rcu_gp
  4 root      0 -20     0    0    0 I  0.0  0.0 0:00.00 rcu_par_gp
  5 root      0 -20     0    0    0 I  0.0  0.0 0:00.00 slub_flushwq
  6 root      0 -20     0    0    0 I  0.0  0.0 0:00.00 netns
  7 root      20   0      0    0    0 I  0.0  0.0 0:00.00 kworker/0:0-events
  8 root      0 -20     0    0    0 I  0.0  0.0 0:00.00 kworker/0:0H-events_highpri
  9 root      20   0      0    0    0 I  0.0  0.0 0:00.14 kworker/u30:0-flush-202:0
 10 root      0 -20     0    0    0 I  0.0  0.0 0:00.00 mm_percpu_wq
 11 root      20   0      0    0    0 I  0.0  0.0 0:00.00 rcu_tasks_kthread
 12 root      20   0      0    0    0 I  0.0  0.0 0:00.00 rcu_tasks_rude_kthread
```

- Initially CPU utilization is 27 less than target value.



- After some time utilization (68%) is increased and greater than target value.



- Instances is creating to reduce utilization and stress.

The image consists of three vertically stacked screenshots of the AWS EC2 Instances page, each showing a list of running instances. A red box highlights the last instance in each list, which is in the 'Initializing' state. This visualizes the process of creating new instances.

**Screenshot 1:** Shows 4 instances. The last instance, 'i-0e12e567542ca76db', is in the 'Initializing' state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
server2	i-00cdf403ad319f26e	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-3-109-14
server1	i-04aeedb6d071d77bf	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-233-3
	i-0e12e567542ca76db	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-201-9

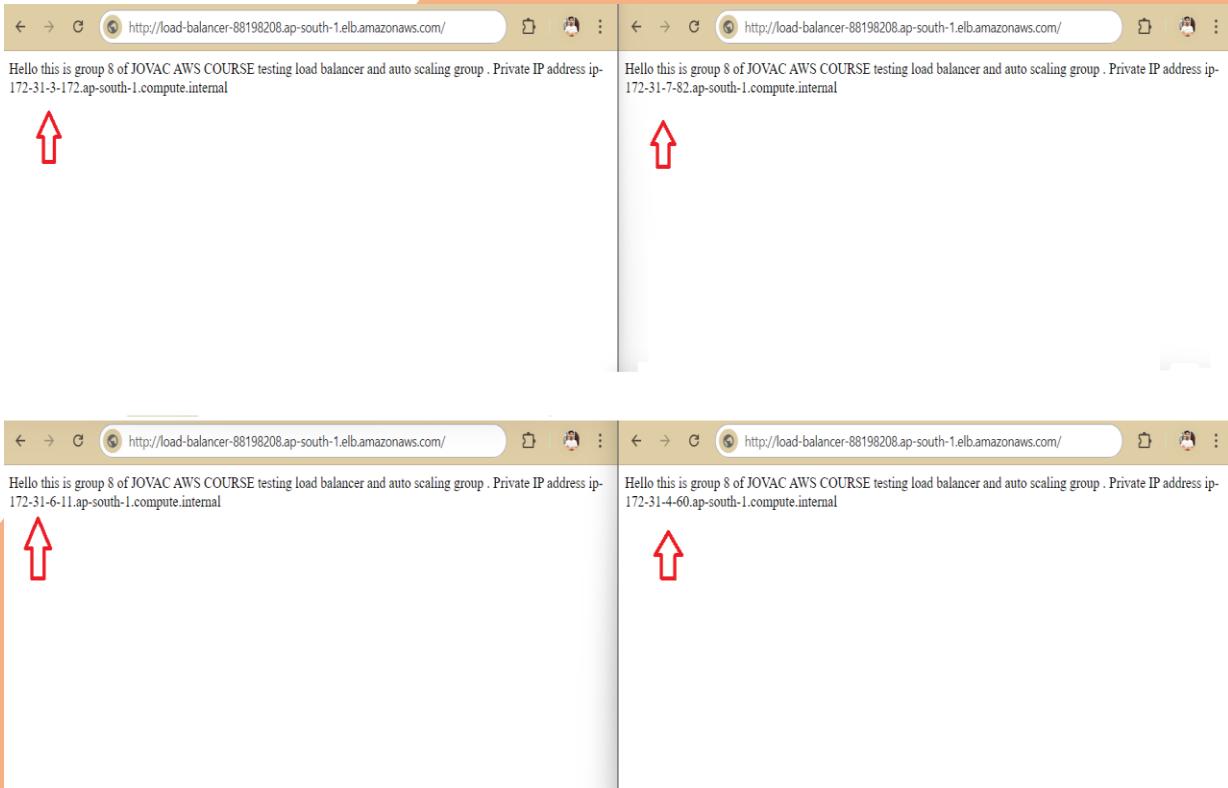
**Screenshot 2:** Shows 5 instances. The last instance, 'i-0891cd9462eaec0d7', is in the 'Initializing' state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
server2	i-00cdf403ad319f26e	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-3-109-14
server1	i-04aeedb6d071d77bf	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-233-3
	i-0891cd9462eaec0d7	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-127-7
	i-0e12e567542ca76db	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-201-9

**Screenshot 3:** Shows 6 instances. The last instance, 'i-067cba18b32c13496', is in the 'Initializing' state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
server2	i-00cdf403ad319f26e	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-3-109-14
server1	i-04aeedb6d071d77bf	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-233-3
	i-0891cd9462eaec0d7	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-127-7
	i-0e12e567542ca76db	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-201-9
	i-067cba18b32c13496	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-3-109-2C

- Server distribute requests.



- After creating instances stress and utilization decrease.



- CPU Utilization decrease then instances also shutting down and terminated.

The screenshot shows the AWS EC2 Instances page with a red box highlighting several instances in the list. The instances are:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
	i-085c2341c83de3205	Terminated	t2.micro	-	View alarms +	ap-south-1b	-
server2	i-06450d5b77c2acbbe	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-23-
	i-0131c52af938711cf	Terminated	t2.micro	-	View alarms +	ap-south-1b	-
server1	i-087556ab2b985a366	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-108-
	i-012aff7dadb312e65	Shutting-d...	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-110-
	i-077c4d1e92fff8ed5	Terminated	t2.micro	-	View alarms +	ap-south-1b	-
	i-0e0a22ead75e4caf	Terminated	t2.micro	-	View alarms +	ap-south-1b	-
	i-02e7030ca2d9b3240	Shutting-d...	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-35-15-
	i-0ed504abc69d89279	Shutting-d...	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-110-

Select an instance

## ➤ Objective/Problem statement.

---

- How to make cloud environment scalable in any situation.

- What Is Cloud Scalability?

- Cloud scalability refers to the ability of a cloud computing environment to easily expand or contract its resources and services according to demand. This flexibility allows for the accommodation of workload fluctuations smoothly, without compromising on performance or availability.
- It offers the advantage of easily increasing computing resources, such as storage, RAM, or processing power, to handle peaks in demand. Conversely, it also enables the reduction of resources during low-demand periods, ensuring that businesses only pay for what they use.

## • **Types of Cloud Scalability.**

1. Vertical Scaling
2. Horizontal Scaling
3. Diagonal Scaling

## • **How to Achieve Cloud Scalability?**

---

- There are several features and capabilities that enable cloud systems to be scaled.

### 1. Auto-Scaling

Cloud platform auto-scaling features automatically adjust the number of compute resources assigned to an application based on its needs. This mechanism ensures that the application has the resources it needs when demand spikes, and scales down resources during slower periods to save costs.

## 2. Load Balancing

Load balancers distribute incoming network traffic across multiple servers to ensure no single server becomes overwhelmed. This is crucial for maintaining online availability and performance. Efficiently spreading the load helps achieve scalability, accommodating more users without degradation in service quality.

## 3. Containerization

Containerization is a lightweight alternative to virtualization that involves encapsulating an application in a container with its own operating environment. Containers are inherently portable and can be easily created, replicated, deployed, and moved across cloud environments, making scaling up to meet demand spikes or scaling down during quieter periods straightforward and efficient.

## 4. Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is a practice where infrastructure provisioning and management are performed through code. This approach automates the setup and scaling of cloud environments, ensuring consistency and eliminating manual errors.



**THANK YOU**