AWS Mini Project: Load Balanced Auto-Healing Apache App using EC2 + ELB + ASG

👤 **Author**: Aryan Kaushik

📅 **Date**: 27/06/2025

🎯 Objective

This project demonstrates deploying a scalable and auto-healing Apache web server using:

- EC2 instance with User Data
- Application Load Balancer (ALB)
- Auto Scaling Group (ASG)

🧰 Tools and Services Used

- Amazon EC2 (Ubuntu 24.04 LTS)
- Application Load Balancer (ALB)
- Auto Scaling Group (ASG)
- Launch Template
- Apache2 (via User Data)
- Target Group
- Security Groups
- AWS Management Console

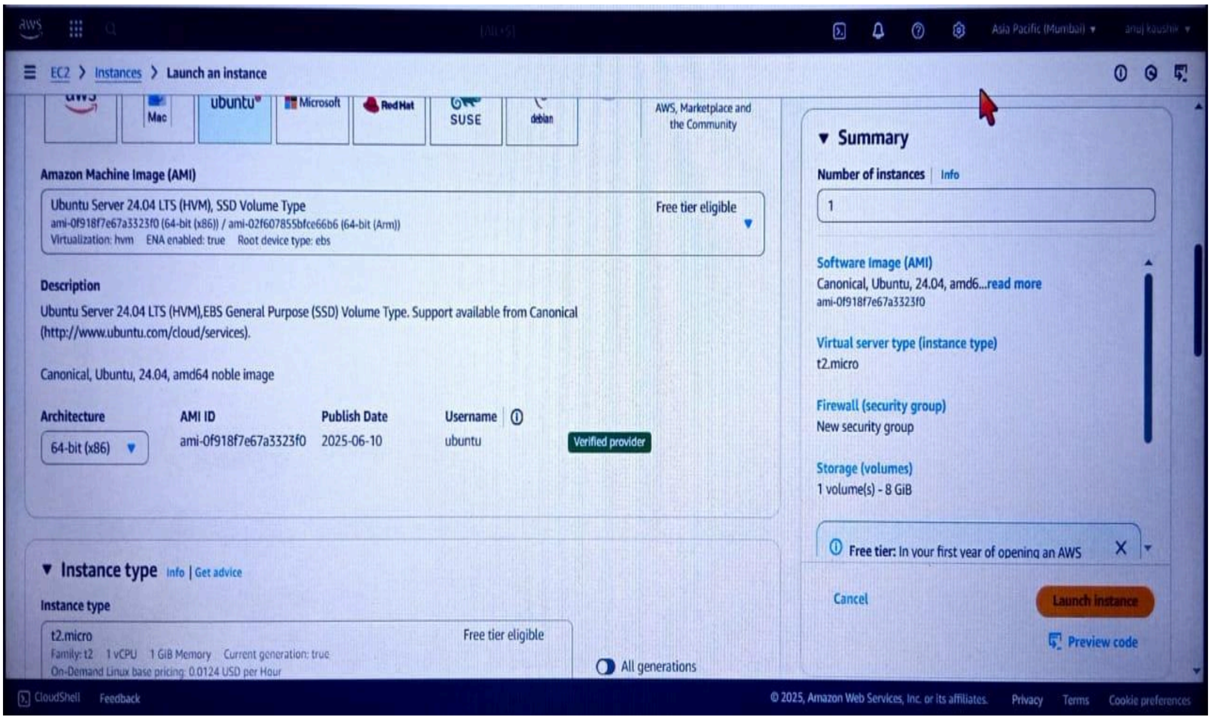
🔧 Step-by-Step Implementation

1️⃣ Launch EC2 with Apache via User Data

- Go to EC2 → Launch Instance
- Choose Ubuntu 24.04 LTS AMI
- Instance type: t3.micro (Free Tier)
- Add User Data

```
#!/bin/bash
apt update -y
apt install apache2 -y
echo "<h1>Hello from $(hostname)</h1>" > /var/www/html/index.html
systemctl enable apache2
systemctl start apache2
```

- Allow HTTP (80) and SSH (22) in Security Group



Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0f918f7e67a3323f0 (64-bit (x86)) / ami-02f607855bfce66b6 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username
64-bit (x86) ▾	ami-0f918f7e67a3323f0	2025-06-10	ubuntu

▾ Instance type

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour

▾ Summary

Number of instances | Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0f918f7e67a3323f0

Virtual server type (instance type)
t2.micro

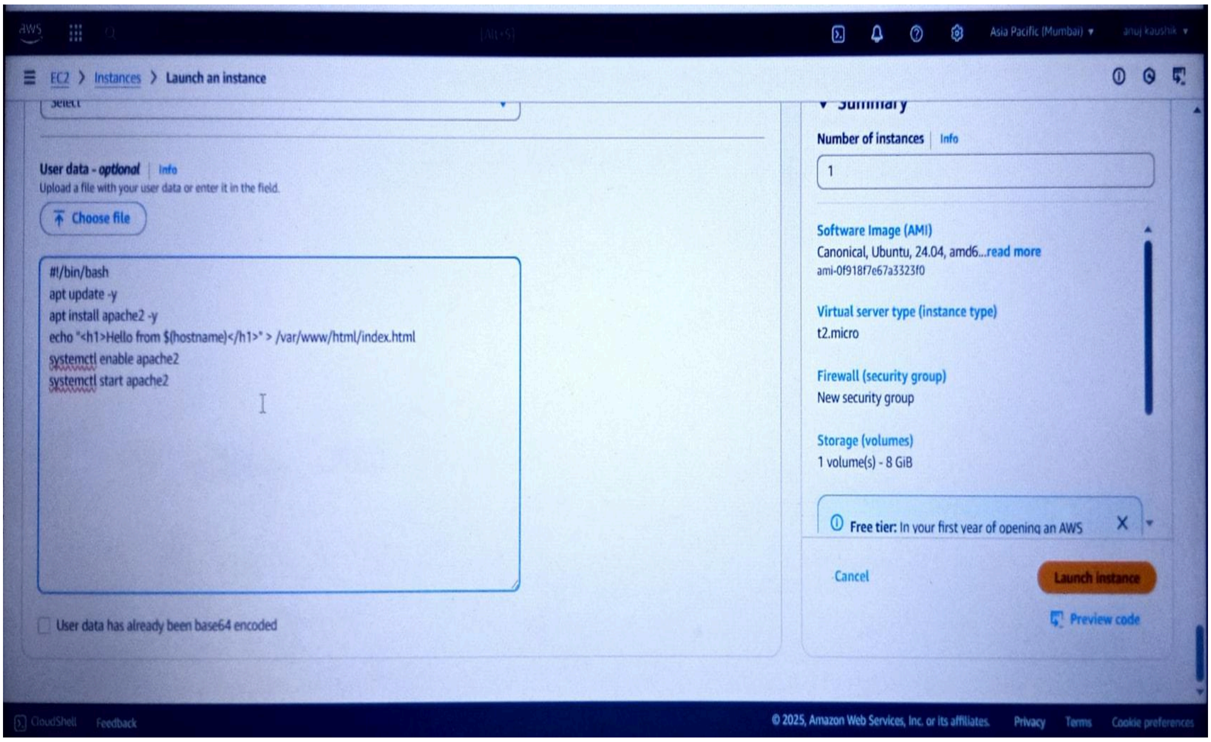
Firewall (security group)
New security group

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

Free tier: In your first year of opening an AWS

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

Figure 1: EC2 Launch Configuration (Ubuntu + t3.micro)


EC2 > Instances > Launch an instance
User data - optional
Info
Upload a file with your user data or enter it in the field.
Choose file

```
#!/bin/bash
apt update -y
apt install apache2 -y
echo "<h1>Hello from $(hostname)</h1>" > /var/www/html/index.html
systemctl enable apache2
systemctl start apache2
```

□ User data has already been base64 encoded

Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0f918f7e67a3323f0

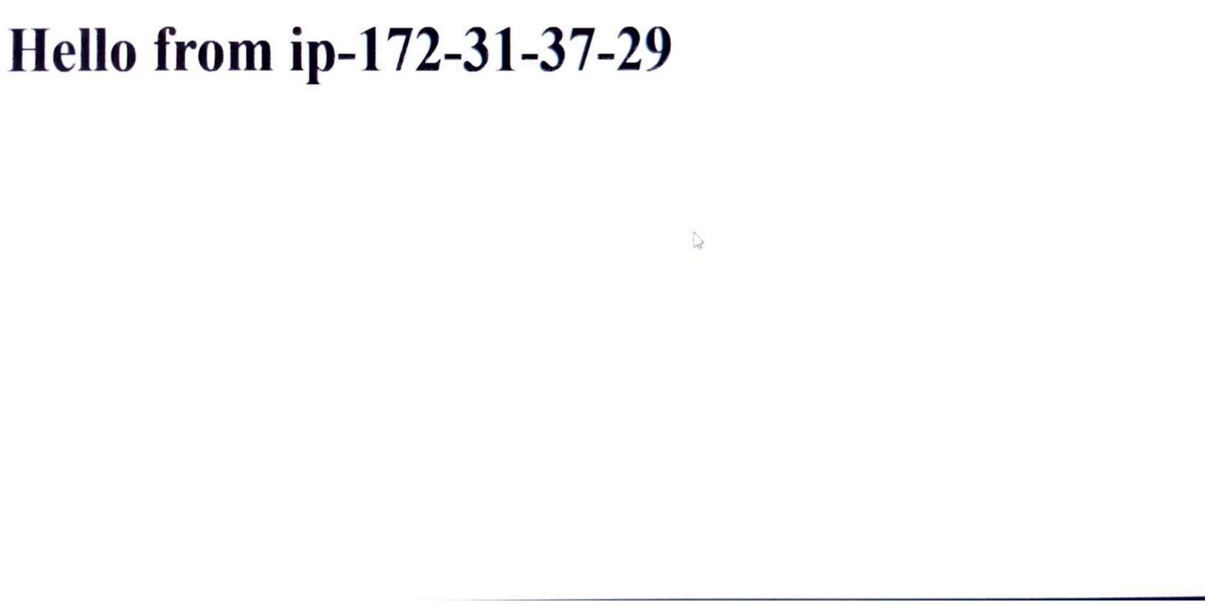
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year of opening an AWS

Figure 2: User Data Script Section (Apache install + hostname)

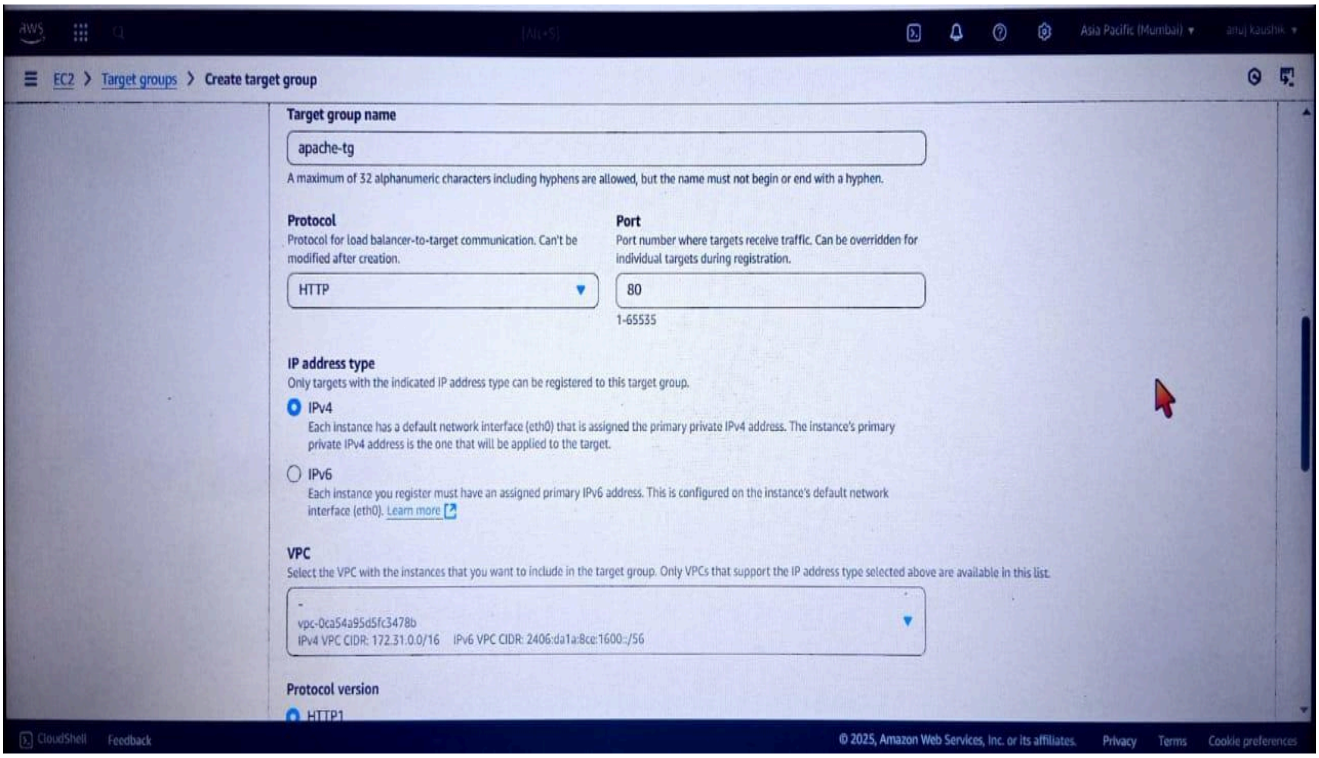


Hello from ip-172-31-37-29

Figure 3: Apache Default Page in Browser

2️⃣Create Target Group

- Go to EC2 → Target Groups → Create
- Name: apache-tg, Type: Instances
- Protocol: HTTP, Port: 80
- Health Check Path: /



EC2 > Target groups > Create target group

Target group name

apache-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Protocol for load balancer-to-target communication. Can't be modified after creation.

HTTP ▾

Port

Port number where targets receive traffic. Can be overridden for individual targets during registration.

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

● IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

○ IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

VPC

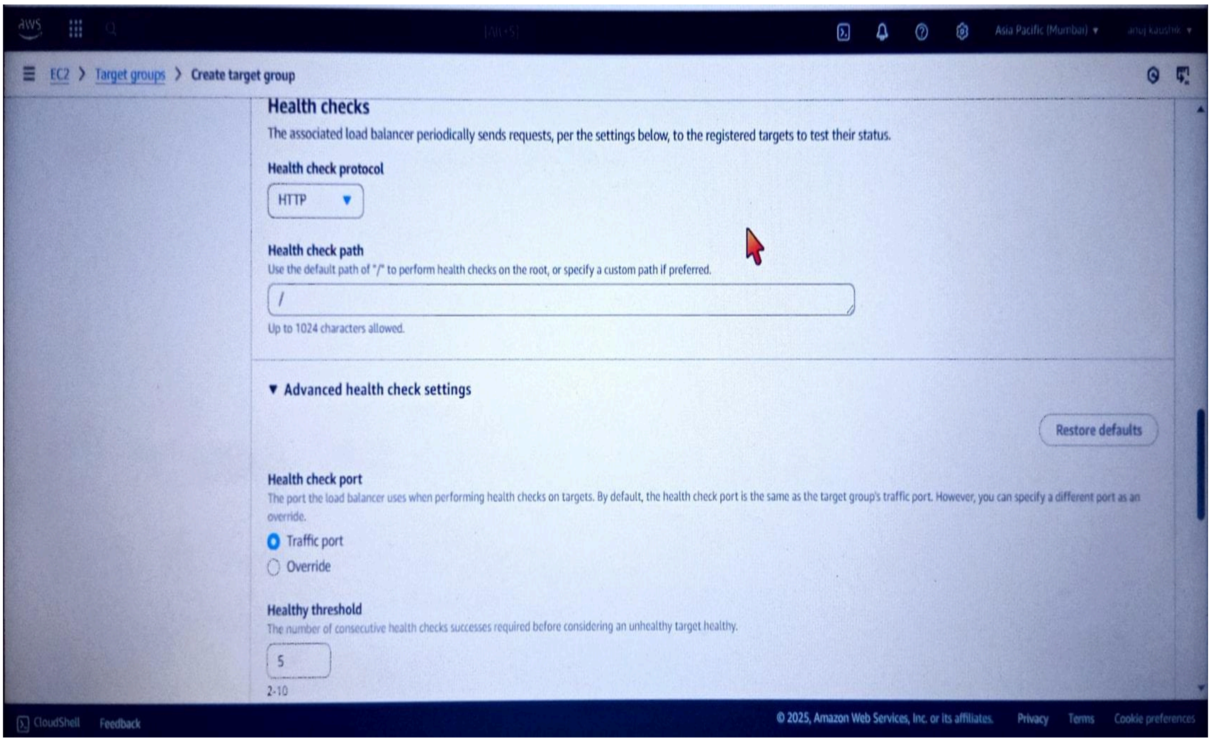
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

-
vpc-0ca54a95d5fc3478b
IPv4 VPC CIDR: 172.31.0.0/16   IPv6 VPC CIDR: 2406:da1a:8ce:1600::/56 ▾

Protocol version

● HTTP1

Figure 4: Target Group Creation Settings



Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP ▼

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

▼ **Advanced health check settings**

Health check port

The port the load balancer uses when performing health checks on targets. By default, the health check port is the same as the target group's traffic port. However, you can specify a different port as an override.

● Traffic port

○ Override

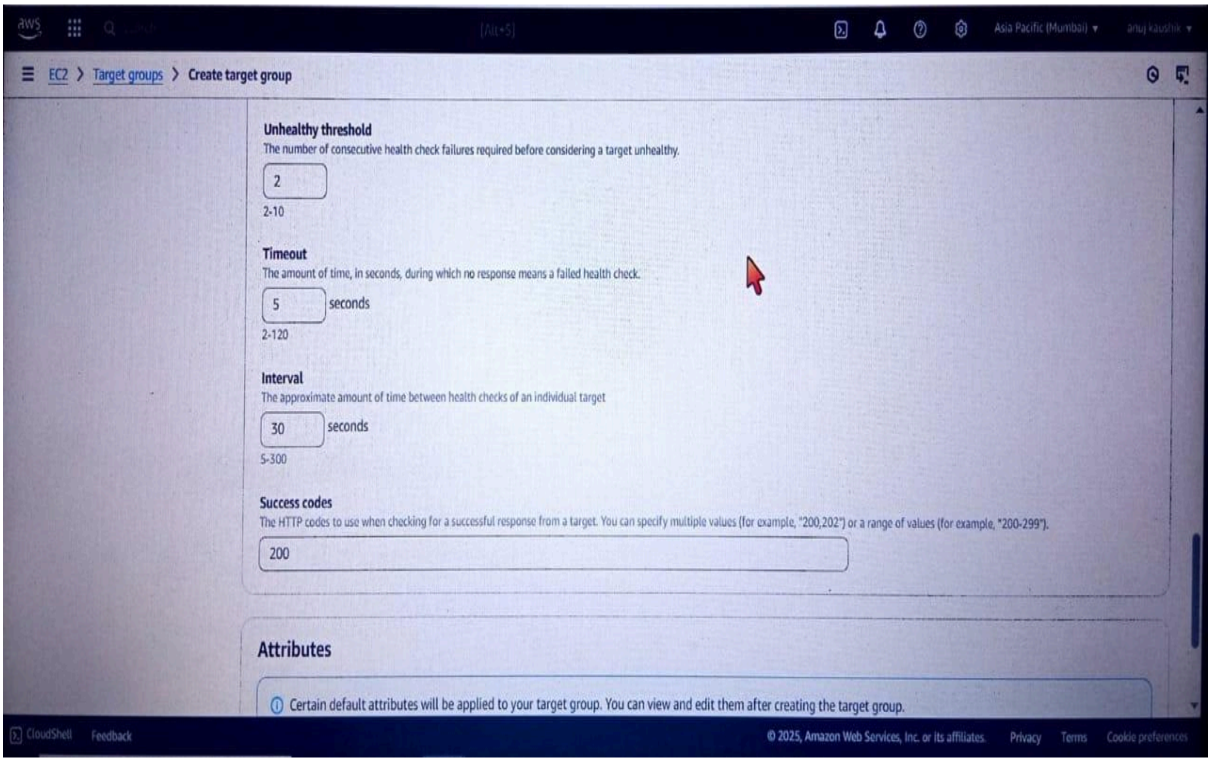
Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

5

2-10

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



EC2 > Target groups > Create target group

Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

Timeout

The amount of time, in seconds, during which no response means a failed health check.

5 seconds

2-120

Interval

The approximate amount of time between health checks of an individual target

30 seconds

5-300

Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

200

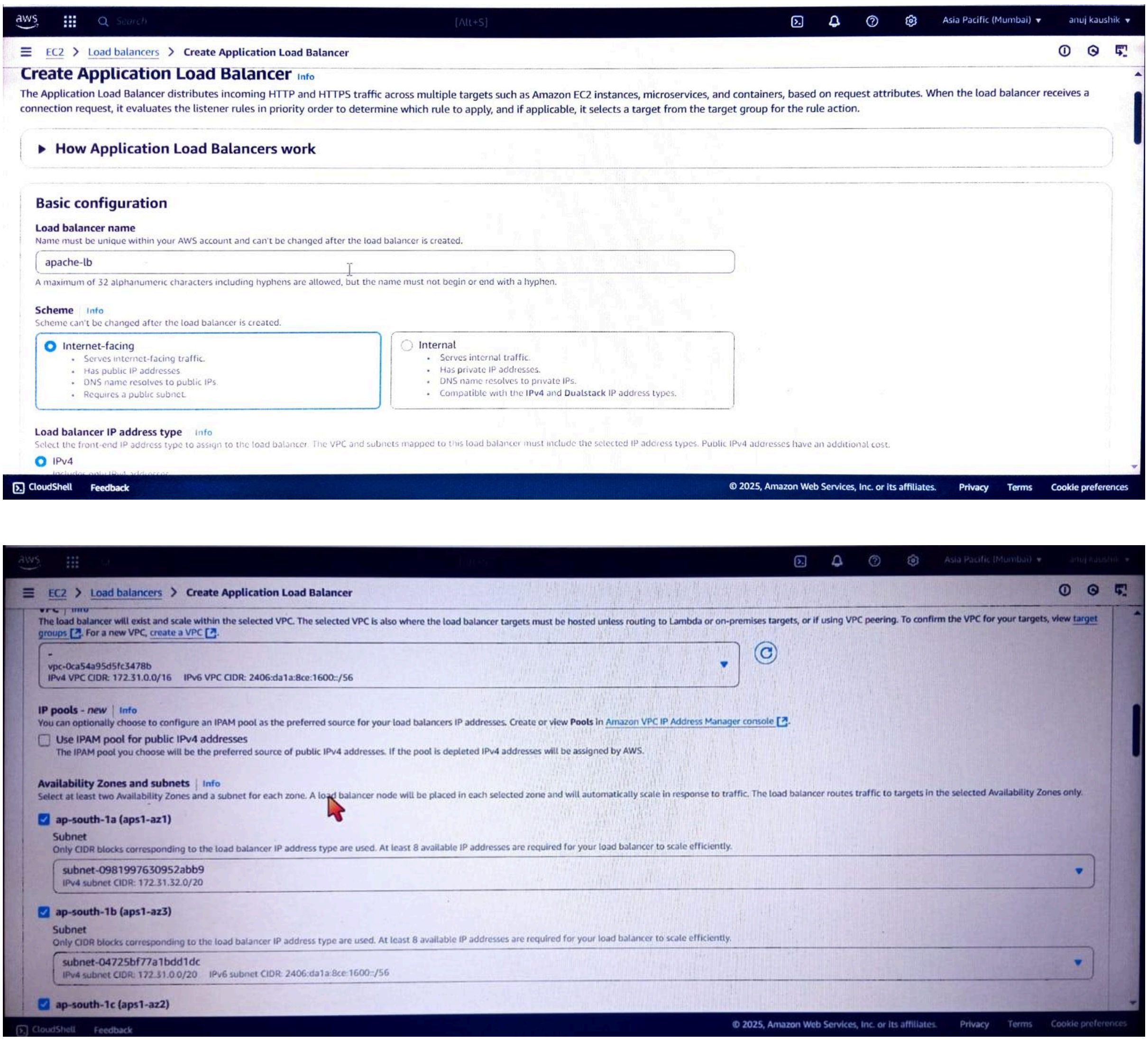
Attributes

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

Figure 5: Health Check Configuration (Path /)

3️⃣Create Application Load Balancer (ALB)

- Go to EC2 → Load Balancers → Create
- Name: apache-lb, Internet-facing
- Select 3 Availability Zones
- Listener: HTTP, Port 80
- Attach target group: apache-tg
- Create new SG allowing HTTP



EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▸ How Application Load Balancers work

Basic configuration

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

apache-lb

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
- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

○ Internal
- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the **IPv4** and **Dualstack** IP address types.

Load balancer IP address type Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP adress types. Public IPv4 addresses have an additional cost.

● IPv4

EC2 > Load balancers > Create Application Load Balancer

The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view target groups. For a new VPC, create a VPC.

-
vpc-0ca54a95d5fc3478b
IPv4 VPC CIDR: 172.31.0.0/16 IPv6 VPC CIDR: 2406:da1a:8ce:1600::/56

IP pools - *new* | Info
You can optionally choose to configure an IPAM pool as the preferred source for your load balancers IP addresses. Create or view **Pools** in Amazon VPC IP Address Manager console.

☐ Use IPAM pool for public IPv4 addresses
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

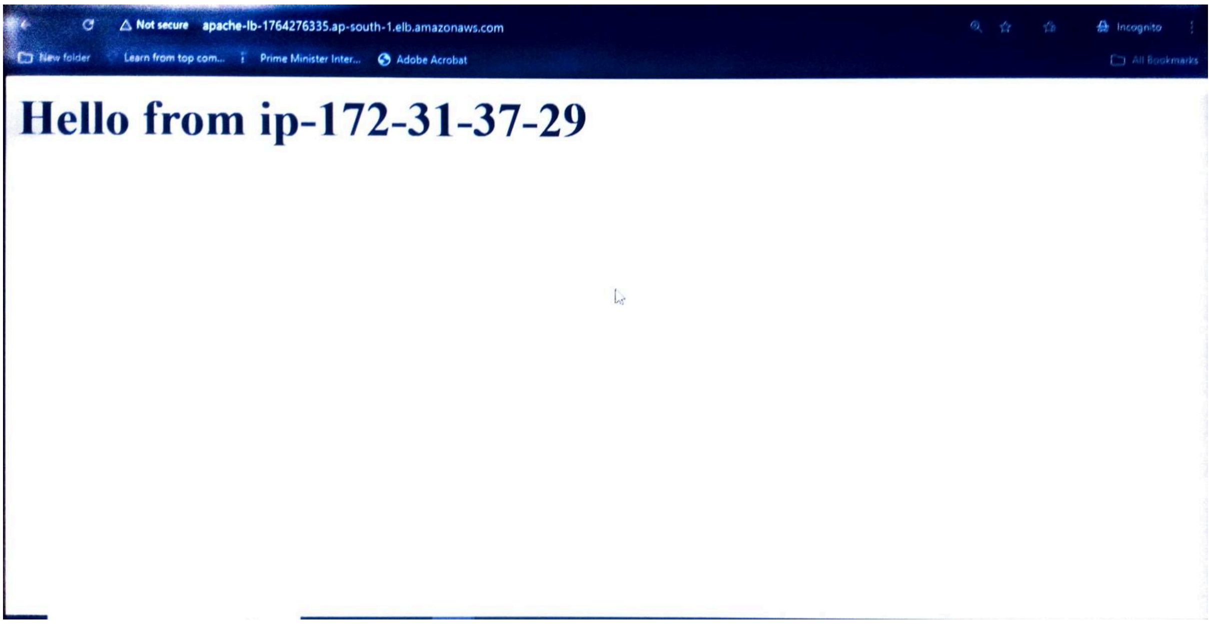
Availability Zones and subnets | Info
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

☑ **ap-south-1a (aps1-az1)**
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-0981997630952abb9
IPv4 subnet CIDR: 172.31.32.0/20

☑ **ap-south-1b (aps1-az3)**
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-04725bf77a1bdd1dc
IPv4 subnet CIDR: 172.31.0.0/20 IPv6 subnet CIDR: 2406:da1a:8ce:1600::/56

☑ **ap-south-1c (aps1-az2)**

Figure 6: Application Load Balancer Configuration (3 AZs)

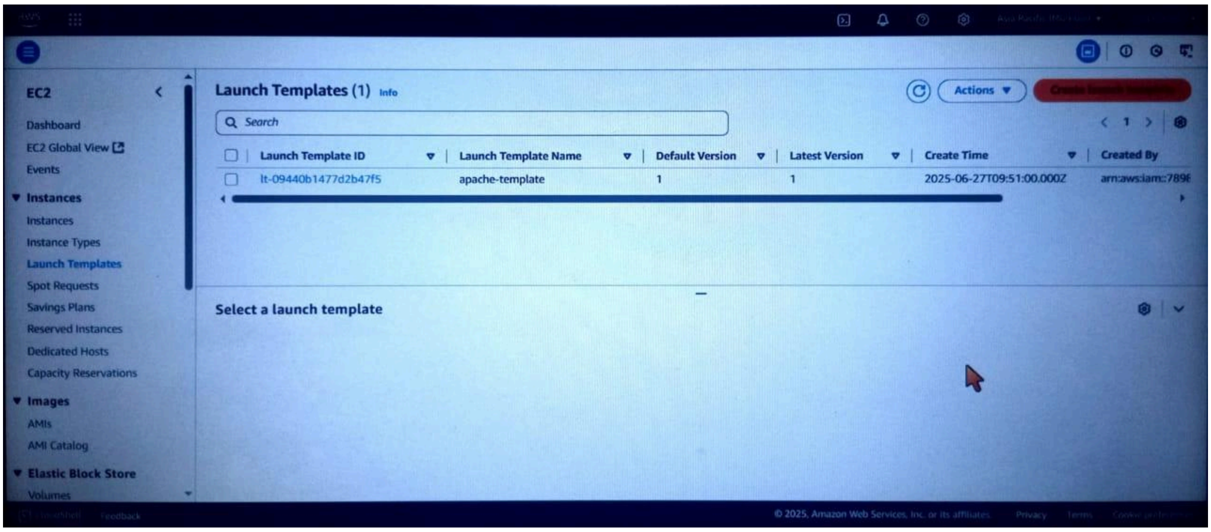


Hello from ip-172-31-37-29

Figure 7: ALB DNS Name Copied and Tested in Browser

4️⃣ Create Launch Template

- Go to EC2 → Launch Templates → Create
- Use same AMI and user data as EC2
- Name: apache-template



Launch Templates (1)

☐	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
☐	lt-09440b1477d2b47f5	apache-template	1	1	2025-06-27T09:51:00.000Z	arn:aws:iam::789

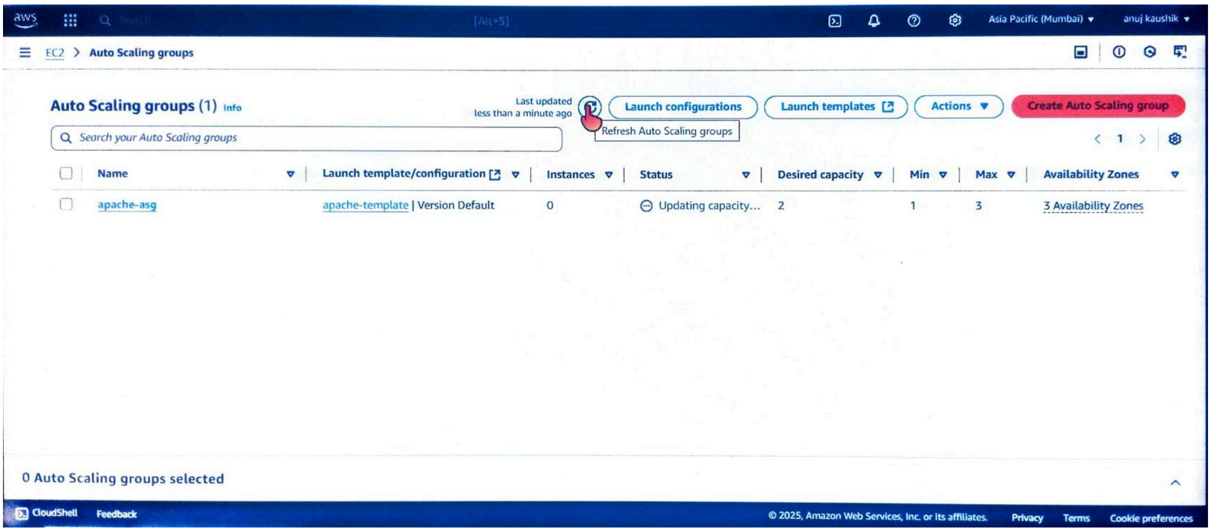
Select a launch template

Figure 8: Launch Template Configuration (AMI, SG, User Data

---

5 Create Auto Scaling Group (ASG)

- Use Launch Template: apache-template
- AZs: All 3 selected
- Attach to Target Group: apache-tg
- Desired: 2, Min: 1, Max: 3
- Skip scaling policies for now


aws
[Alt+S]
Asia Pacific (Mumbai) ▾
anuj kaushik ▾
EC2 > Auto Scaling groups
Auto Scaling groups (1) Info
Last updated less than a minute ago
Launch configurations
Launch templates
Actions ▾
Create Auto Scaling group
Refresh Auto Scaling groups
Search your Auto Scaling groups
< 1 >

☐	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
☐	apache-asg	apache-template | Version Default	0	Updating capacity...	2	1	3	3 Availability Zones

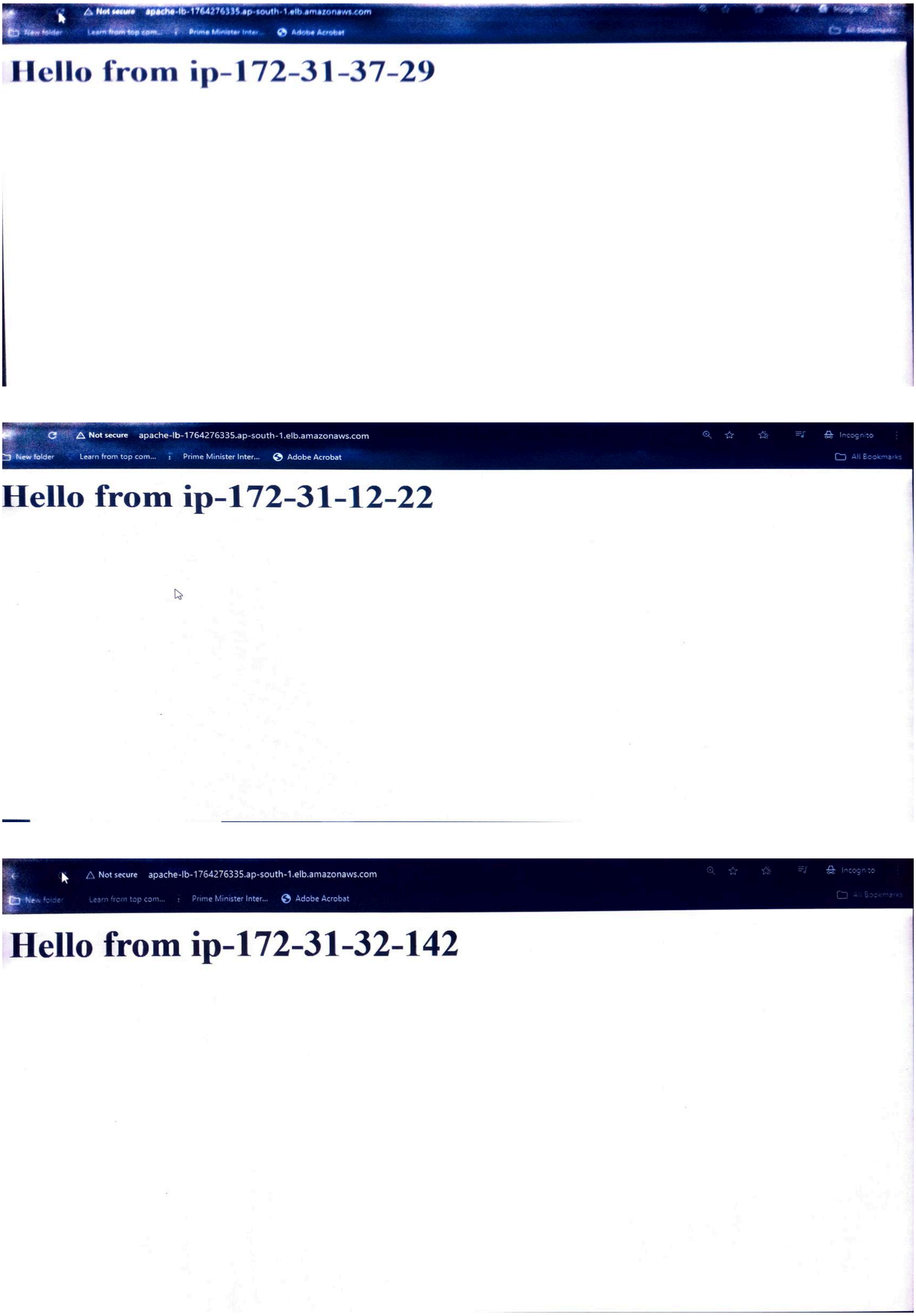
0 Auto Scaling groups selected
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Figure 9: EC2 Instances Launched by ASG

---

6 Test Load Balancer

- Open http://<ALB-DNS>
- Refresh the page — hostname should change



Hello from ip-172-31-37-29

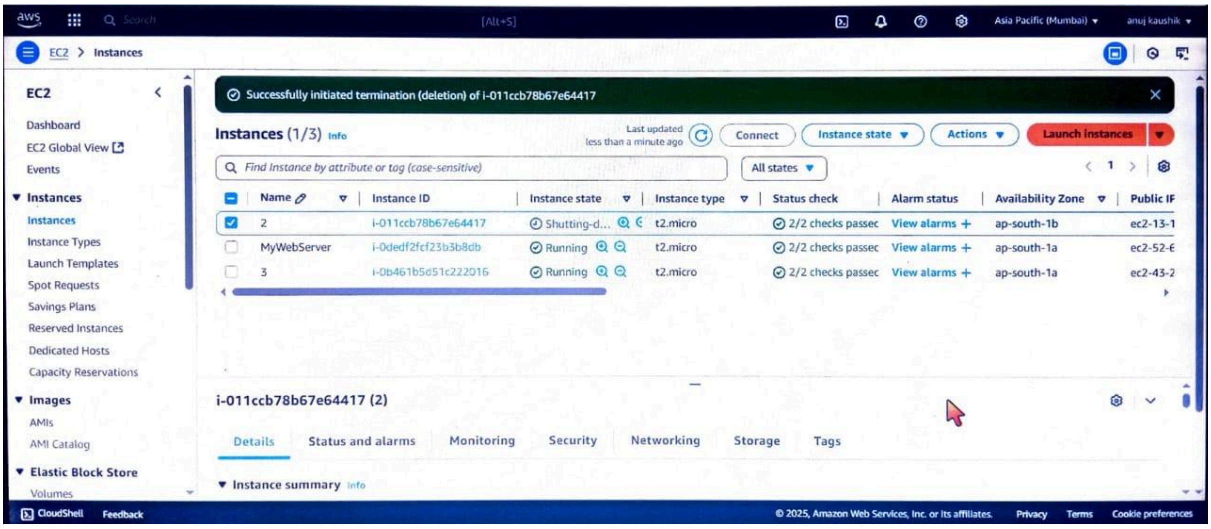
Hello from ip-172-31-12-22

Hello from ip-172-31-32-142

Figure 10: Browser Output Rotating Hostnames via ALB

7 Test Auto Healing

- Terminate 1 EC2 instance
- ASG should auto-launch new one
- App remains available on browser



Successfully initiated termination (deletion) of i-011ccb78b67e64417

Instances (1/3) Info

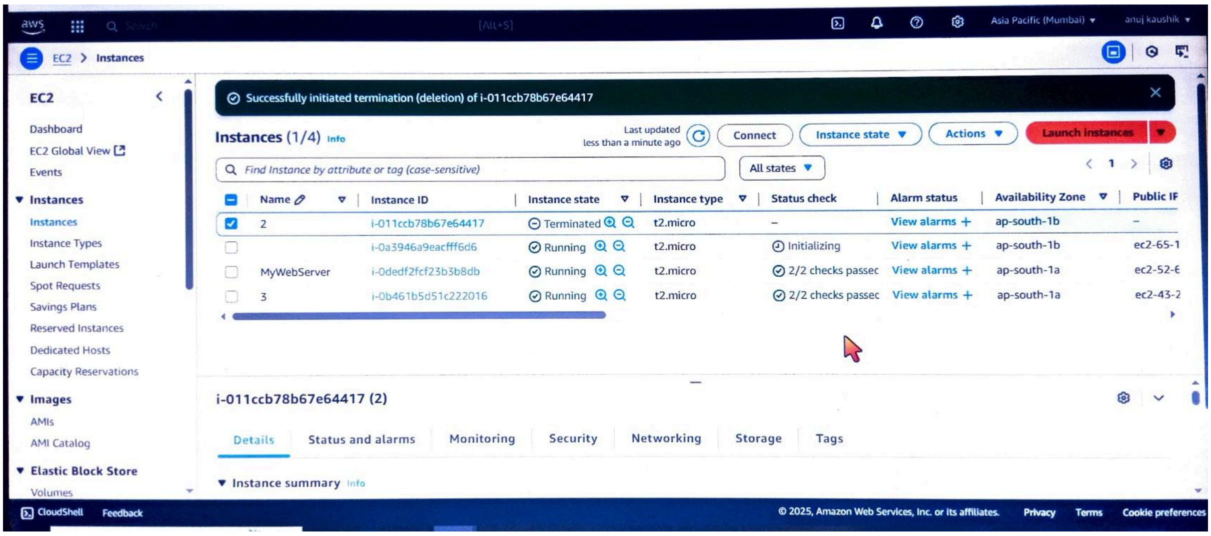
Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IF
☑	2	i-011ccb78b67e64417	Shutting-d...	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-1
□	MyWebServer	i-0dedf2fcf23b3b8db	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-52-6
□	3	i-0b461b5d51c222016	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-43-2

i-011ccb78b67e64417 (2)

Figure 11: EC2 Instance Terminated from ASG



Successfully initiated termination (deletion) of i-011ccb78b67e64417

Instances (1/4) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IF
☑	2	i-011ccb78b67e64417	Terminated	t2.micro	–	View alarms +	ap-south-1b	–
□		i-0a3946a9eacfff6d6	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-65-1
□	MyWebServer	i-0dedf2fcf23b3b8db	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-52-6
□	3	i-0b461b5d51c222016	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-43-2

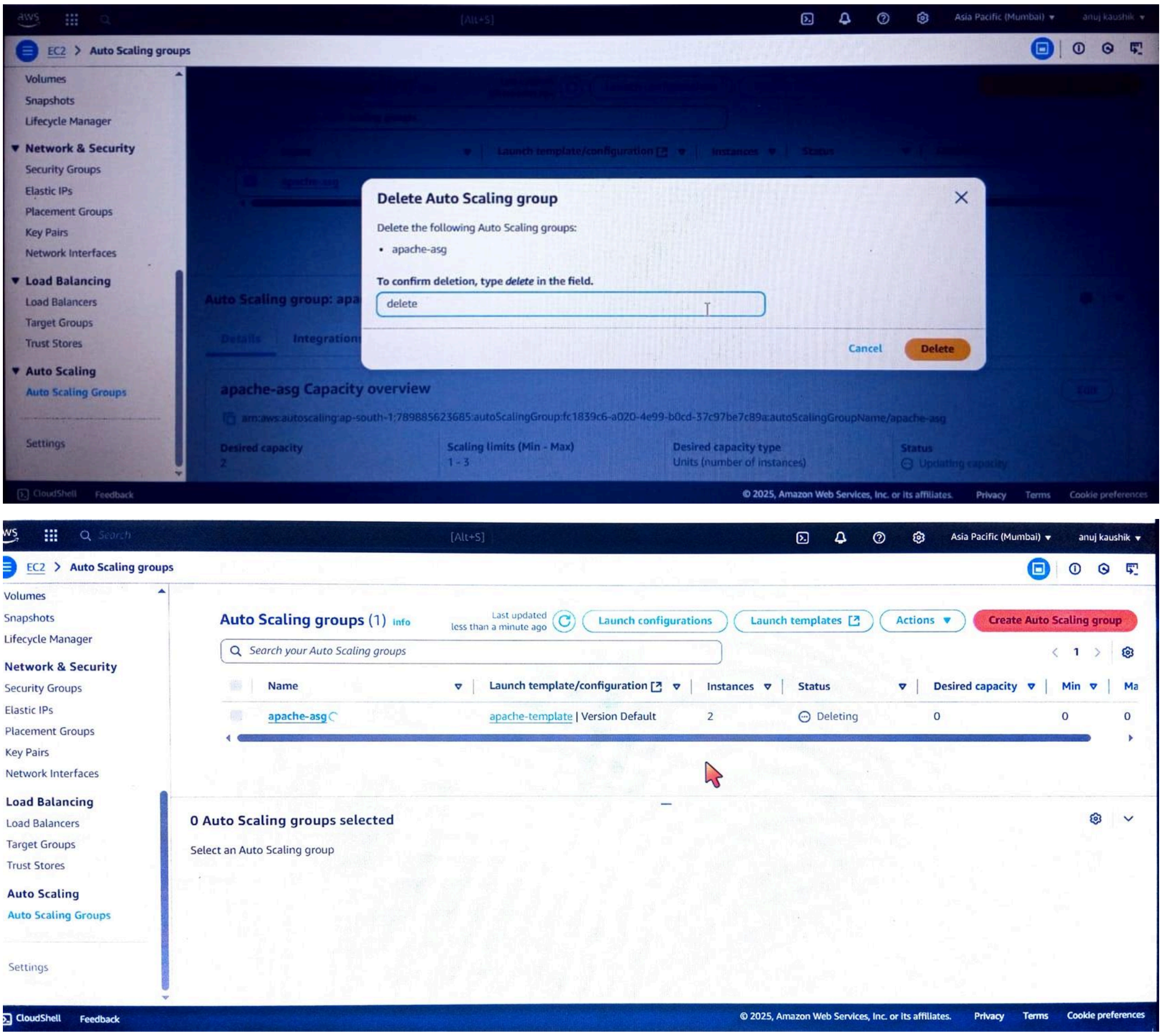
i-011ccb78b67e64417 (2)

Figure 12: ASG Auto-Healed with New EC2 Launch

🧹8️⃣Delete AWS Resources (Clean-Up)

Delete all resources in this order to avoid cost:

1. **Auto Scaling Group**
   - ○ EC2 → Auto Scaling Groups → Select → Delete



Delete Auto Scaling group

Delete the following Auto Scaling groups:

- apache-asg

To confirm deletion, type *delete* in the field.

delete

Cancel | Delete

apache-asg Capacity overview

arn:aws:autoscaling:ap-south-1:789885623685:autoScalingGroup:fc1839c6-a020-4e99-b0cd-37c97be7c89a:autoScalingGroupName/apache-asg

Desired capacity: 2 | Scaling limits (Min - Max): 1 - 3 | Desired capacity type: Units (number of instances) | Status: Updating capacity

Auto Scaling groups (1) Info — Last updated less than a minute ago

Launch configurations | Launch templates | Actions | Create Auto Scaling group

Search your Auto Scaling groups

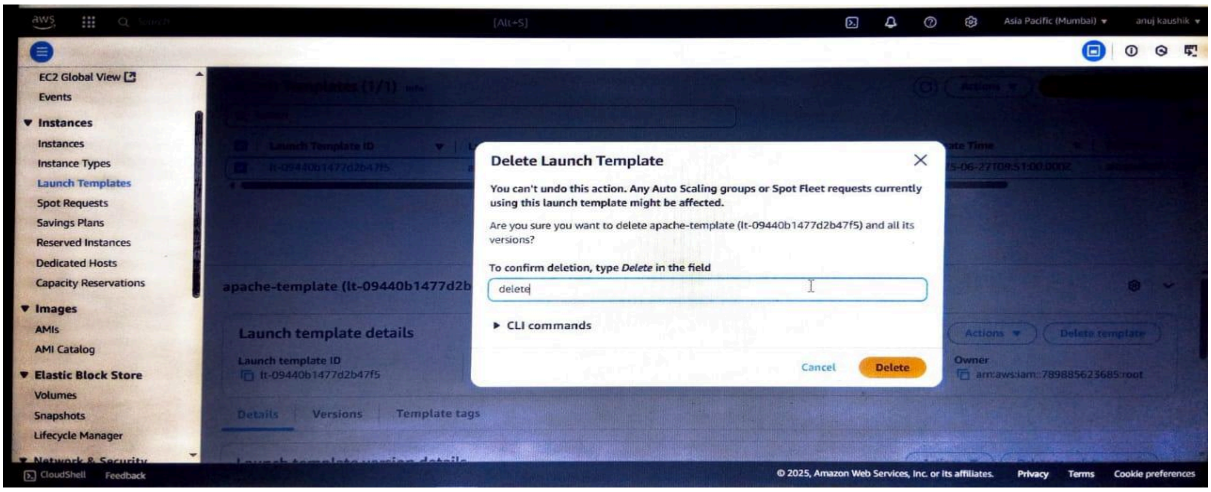
Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Ma
apache-asg	apache-template | Version Default	2	Deleting	0	0	0

0 Auto Scaling groups selected

Select an Auto Scaling group

Figure 13: Deleting Auto Scaling Group (ASG)

2. **Launch Template**
   - ○ EC2 → Launch Templates → Delete



Delete Launch Template

You can't undo this action. Any Auto Scaling groups or Spot Fleet requests currently using this launch template might be affected.

Are you sure you want to delete apache-template (lt-09440b1477d2b47f5) and all its versions?

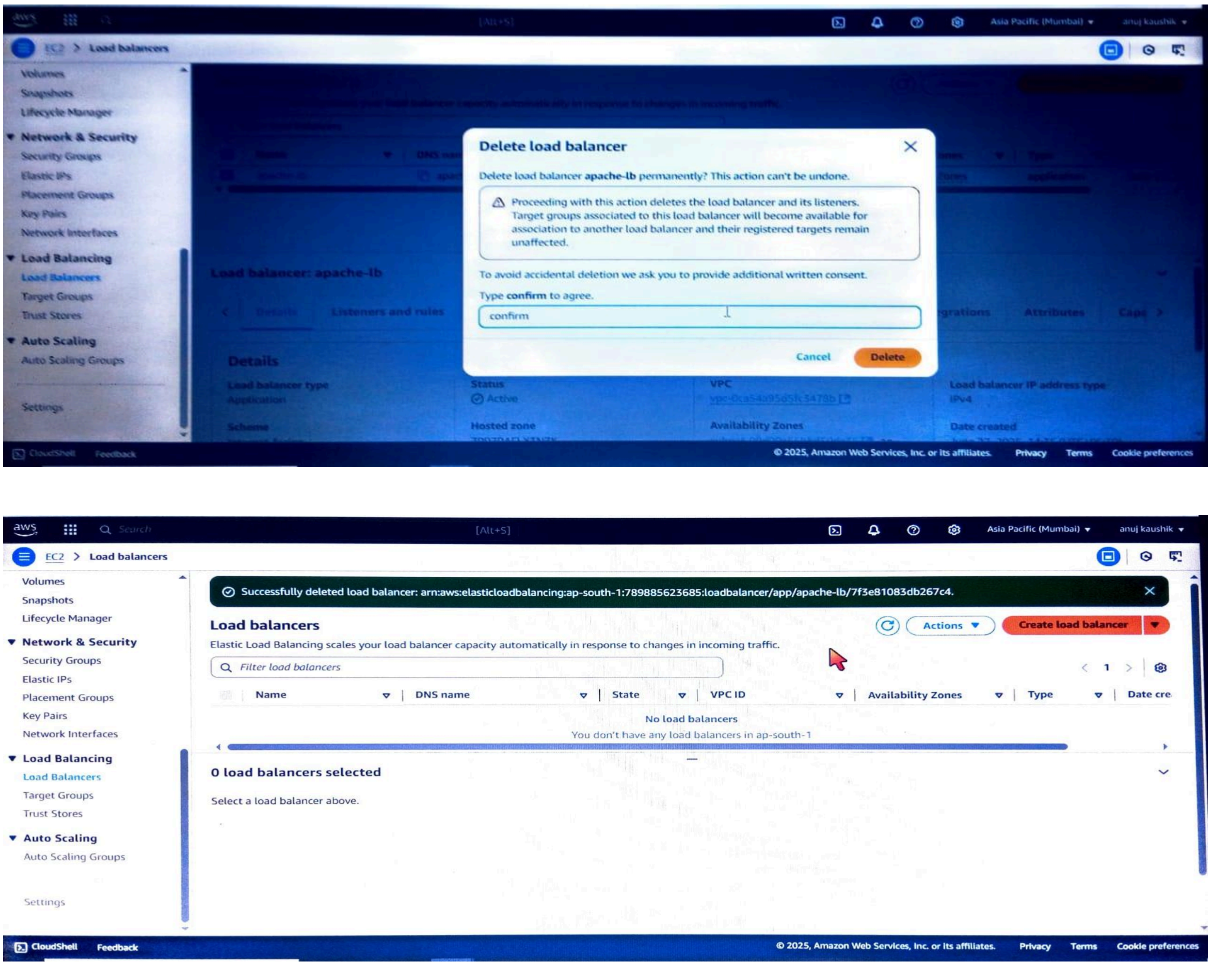
To confirm deletion, type *Delete* in the field

delete

Figure 14: Deleting Launch Template

3. **Load Balancer**

- EC2 → Load Balancers → Delete



Delete load balancer

Delete load balancer **apache-lb** permanently? This action can't be undone.

⚠ Proceeding with this action deletes the load balancer and its listeners. Target groups associated to this load balancer will become available for association to another load balancer and their registered targets remain unaffected.

To avoid accidental deletion we ask you to provide additional written consent.

Type **confirm** to agree.

confirm

Successfully deleted load balancer: arn:aws:elasticloadbalancing:ap-south-1:789885623685:loadbalancer/app/apache-lb/7f3e81083db267c4.

Load balancers

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date cre
No load balancers You don't have any load balancers in ap-south-1						

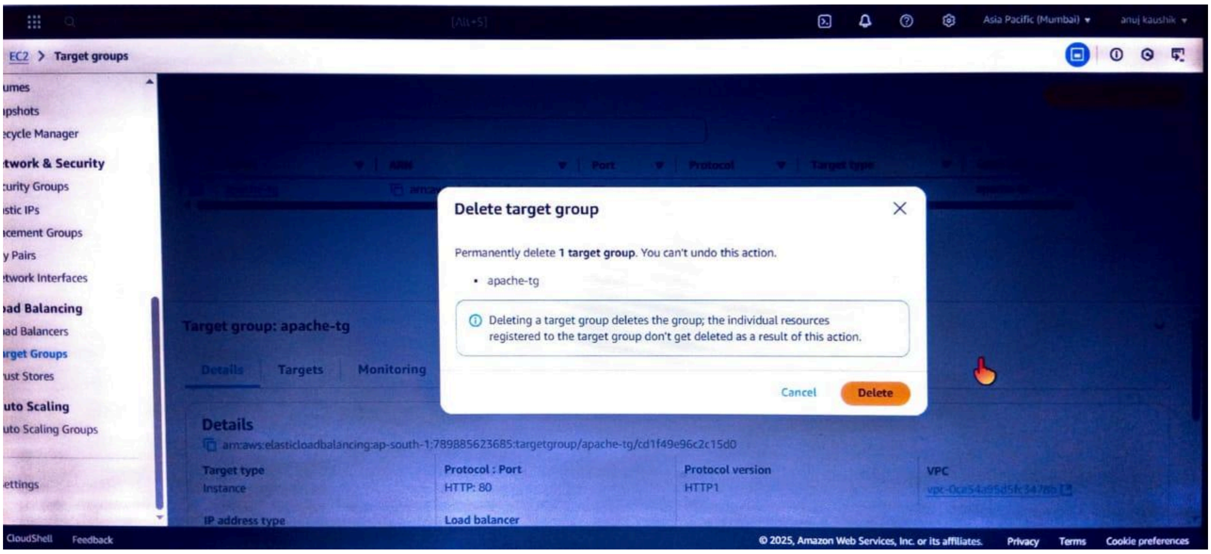
0 load balancers selected

Select a load balancer above.

Figure 15: Deleting Application Load Balancer

4. Target Group

- EC2 → Target Groups → Delete



Delete target group

Permanently delete 1 target group. You can't undo this action.

- apache-tg

Deleting a target group deletes the group; the individual resources registered to the target group don't get deleted as a result of this action.

Cancel Delete

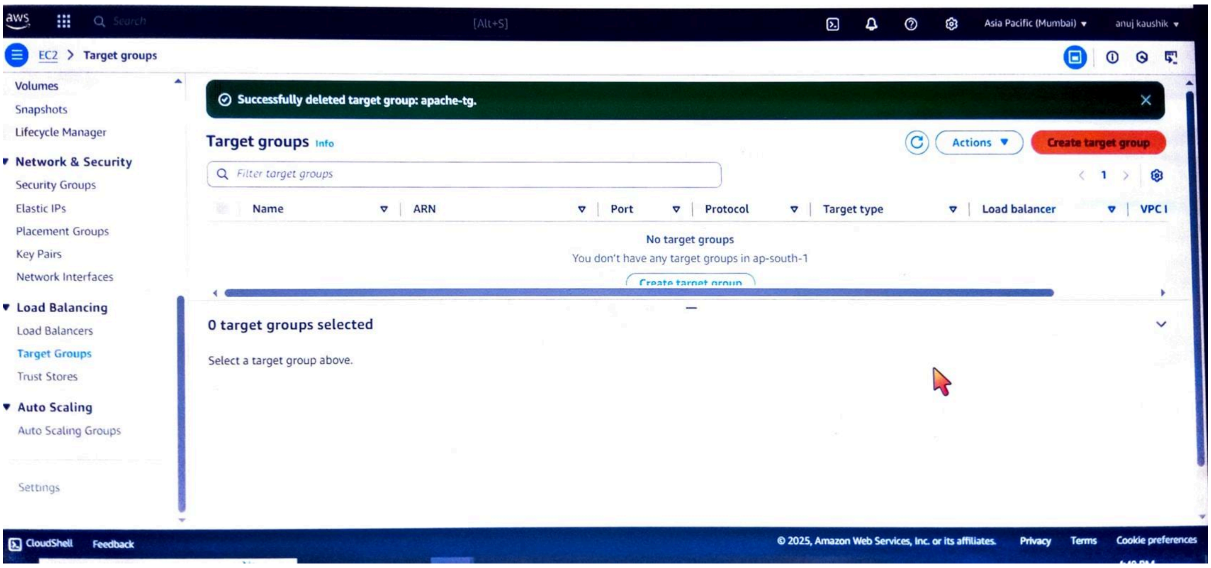
Details

arn:aws:elasticloadbalancing:ap-south-1:789885623685:targetgroup/apache-tg/cd1f49e96c2c15d0

Target type: Instance

Protocol : Port: HTTP: 80

Protocol version: HTTP1



Successfully deleted target group: apache-tg.

No target groups

You don't have any target groups in ap-south-1

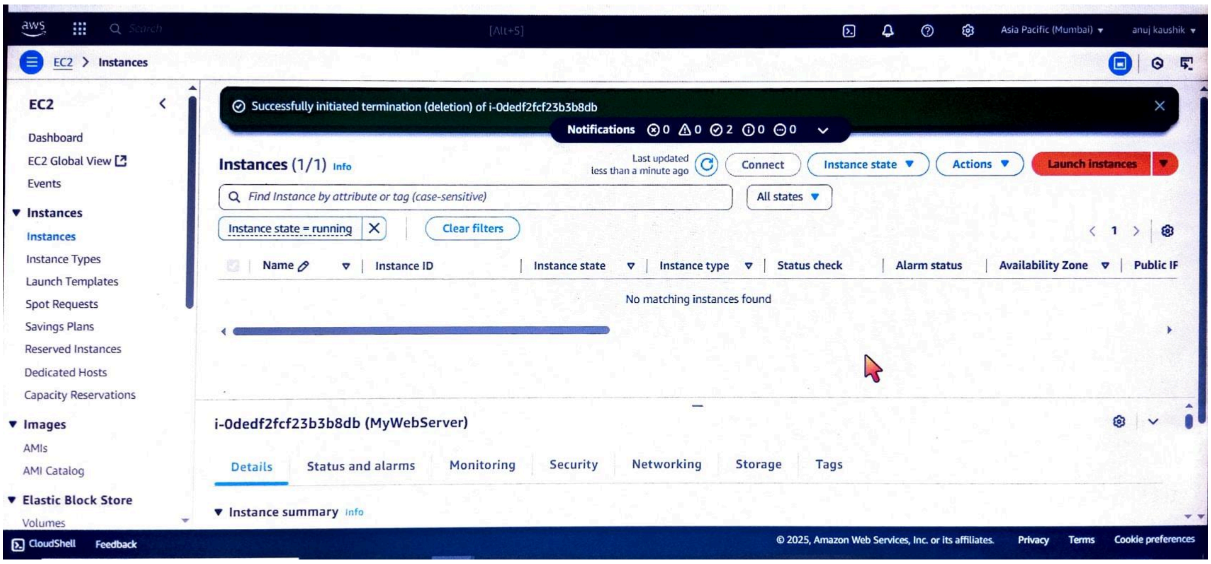
0 target groups selected

Select a target group above.

Figure 16: Deleting Target Group

5. EC2 Instances

- Terminate any leftover instances



Successfully initiated termination (deletion) of i-0dedf2fcf23b3b8db

Instance state = running

No matching instances found

i-0dedf2fcf23b3b8db (MyWebServer)

Figure 17: Terminating EC2 Instances

✅ Conclusion

✅ Deployed Apache app on EC2
✅ Load balanced with ALB
✅ Auto Healing via ASG
✅ Resources deleted to stay within Free Tier