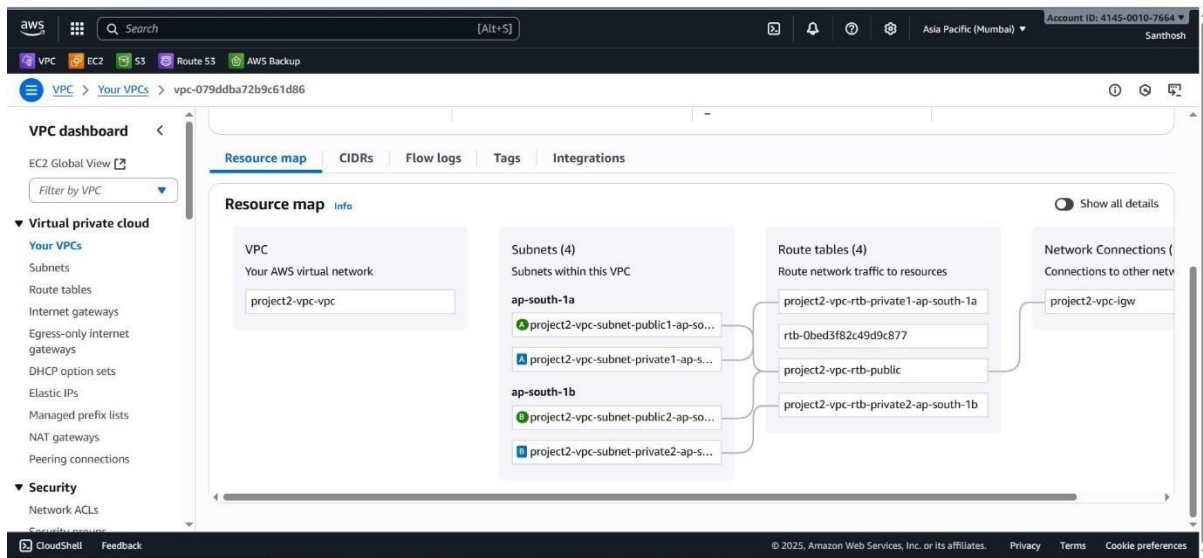
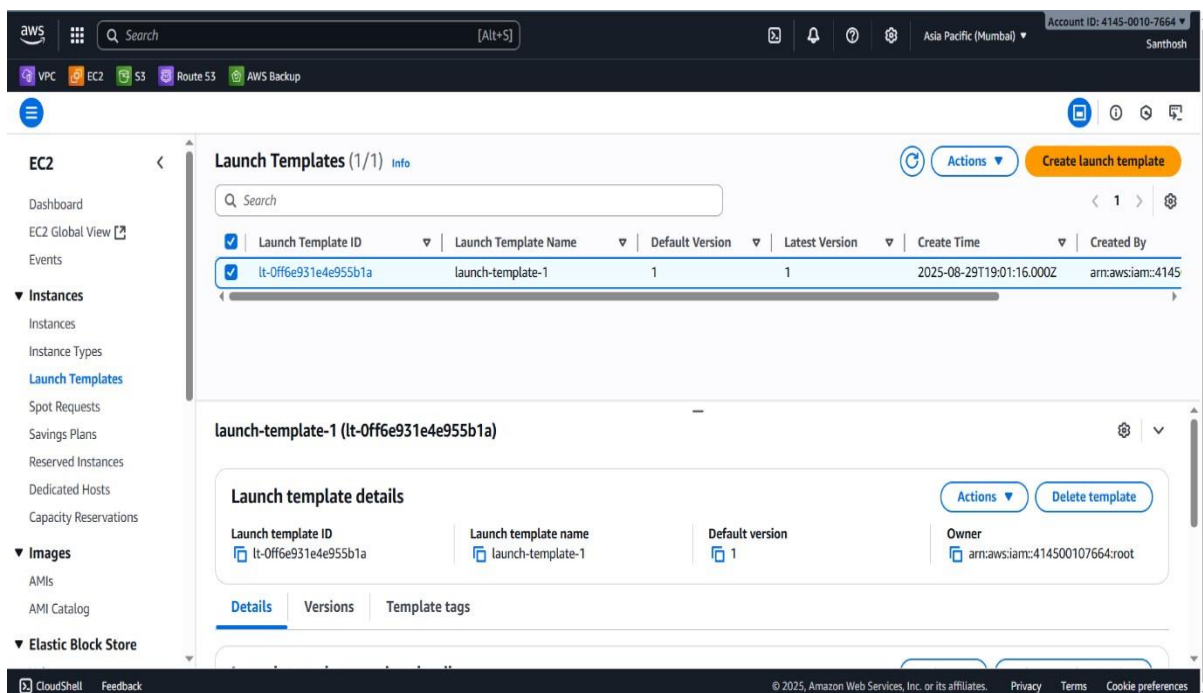


Step 1– 5: Creating vpc, public subnets, private subnet, attaching internet gateway to the public route table.



Step 6: Created a Launch Template with custom AMI.



Using the same in Autoscaling config.

The screenshot shows the 'Create Auto Scaling group' wizard in the AWS Management Console, specifically Step 2: 'Choose instance launch options'. The left sidebar shows a progress bar with steps 2 through 7. The main content area has a 'Name' section with 'Autoscaling-group-1' entered. Below that is the 'Launch template' section, which includes a dropdown menu set to 'launch-template-1' and a 'Version' dropdown set to 'Default (1)'. A blue information box at the top of the launch template section states: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.'

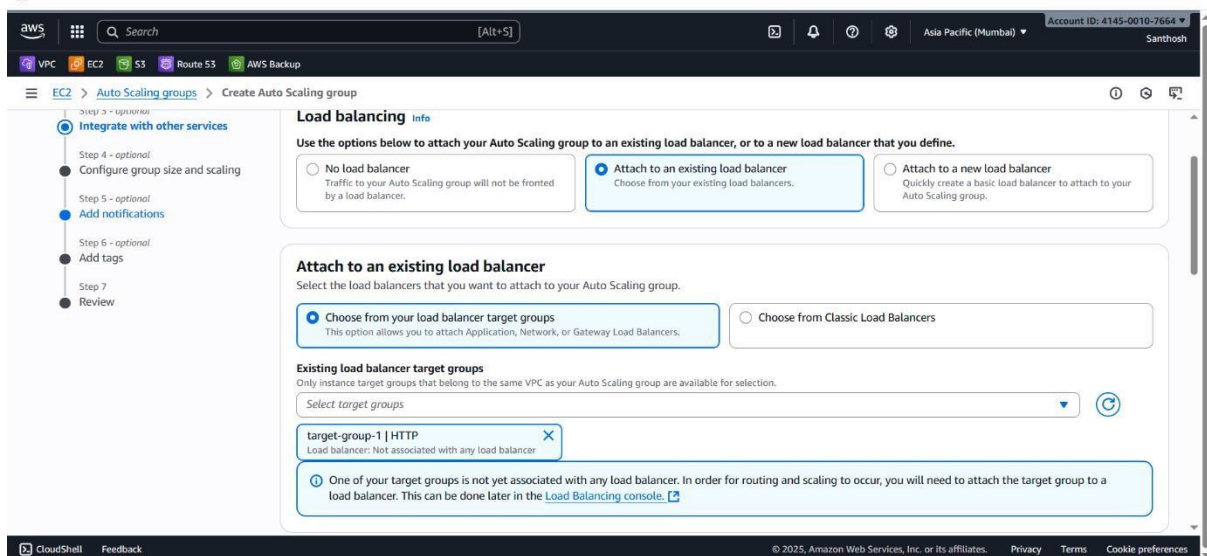
Defining the Minimum: 2, Max: 5, Desired: 2 capacity.

The screenshot shows the 'Create Auto Scaling group' wizard in the AWS Management Console, specifically Step 6: 'Add tags'. The left sidebar shows a progress bar with steps 6 and 7. The main content area has a 'Desired capacity' section with the value '2' entered. Below that is the 'Scaling' section, which includes 'Scaling limits' with 'Min desired capacity' set to '2' and 'Max desired capacity' set to '5'. The 'Automatic scaling - optional' section has two radio buttons: 'No scaling policies' (selected) and 'Target tracking scaling policy'. The 'Instance maintenance policy' section is visible at the bottom.

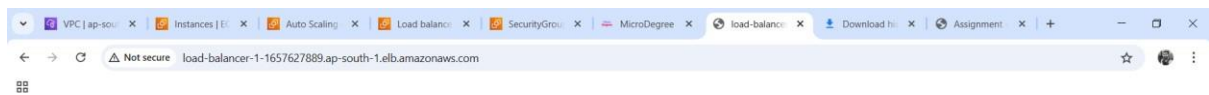
Attaching a Load balancer in the same.

The screenshot shows the 'Create Auto Scaling group' wizard in the AWS Management Console, specifically Step 3: 'Integrate with other services'. The left sidebar shows a progress bar with steps 3, 6, and 7. The main content area has a 'Load balancing' section with a 'Load balancer 1' section. The 'Name' field is 'load-balancer-1', the 'Type' is 'Application/HTTP', and the 'Target group' is 'target-group-1'. Below that is the 'VPC Lattice integration options' section with 'VPC Lattice target groups' set to '-'. The 'Application Recovery Controller (ARC) zonal shift' section has 'ARC zonal shift' set to 'Disabled'.

Step 7: Using application loadbalancer, create a target group and launch the instances using autoscaling.

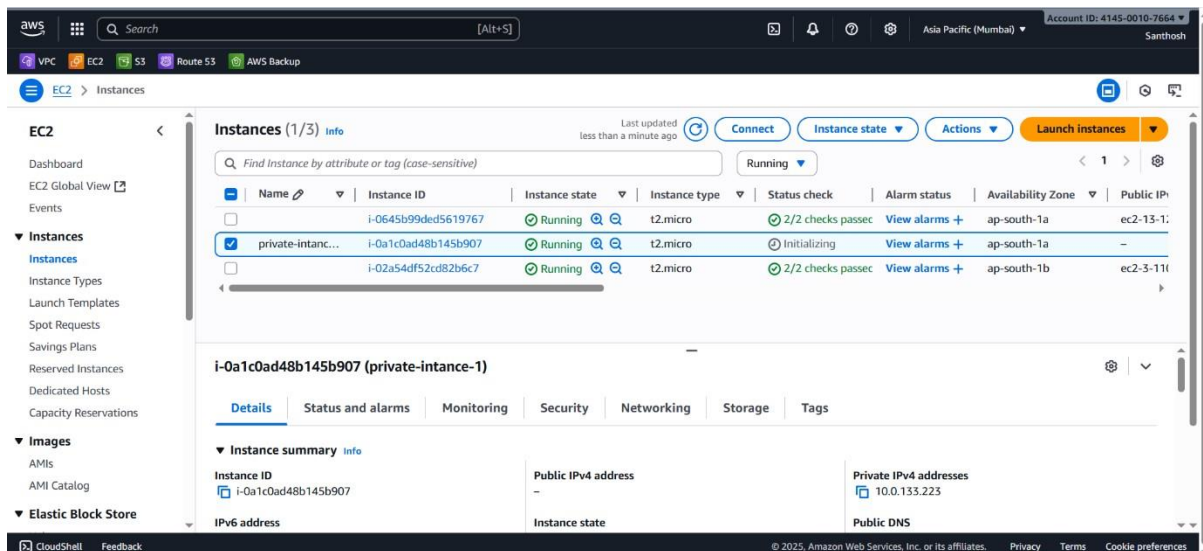


Launched the instance using load balancer and auto scaling group.



**This webpage is launched using custom AMI,in a auto scaling group**

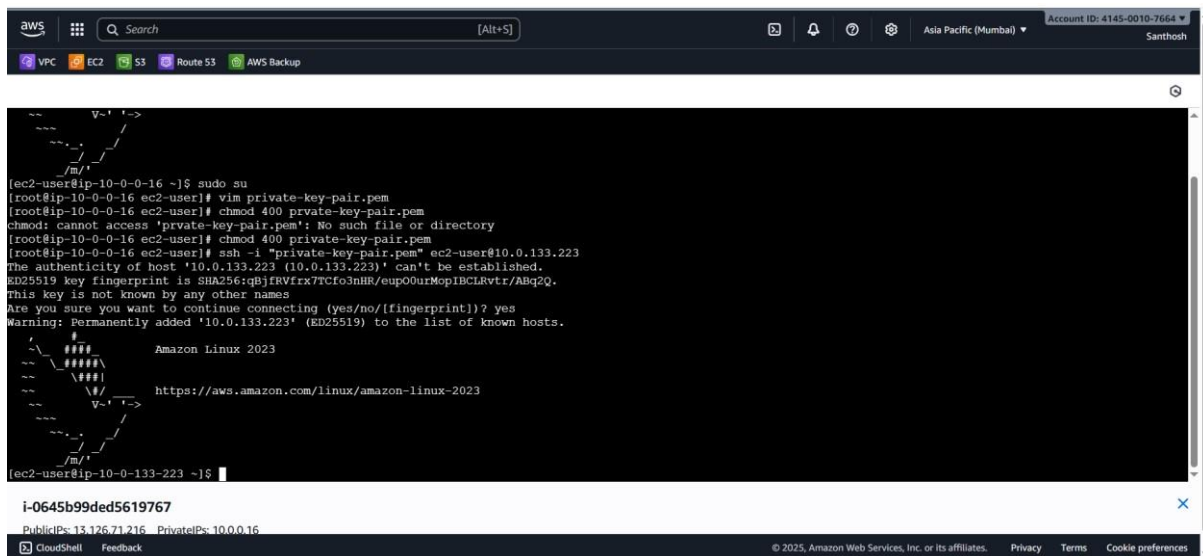
Step 8: creating EC2 instance in a private subnet.



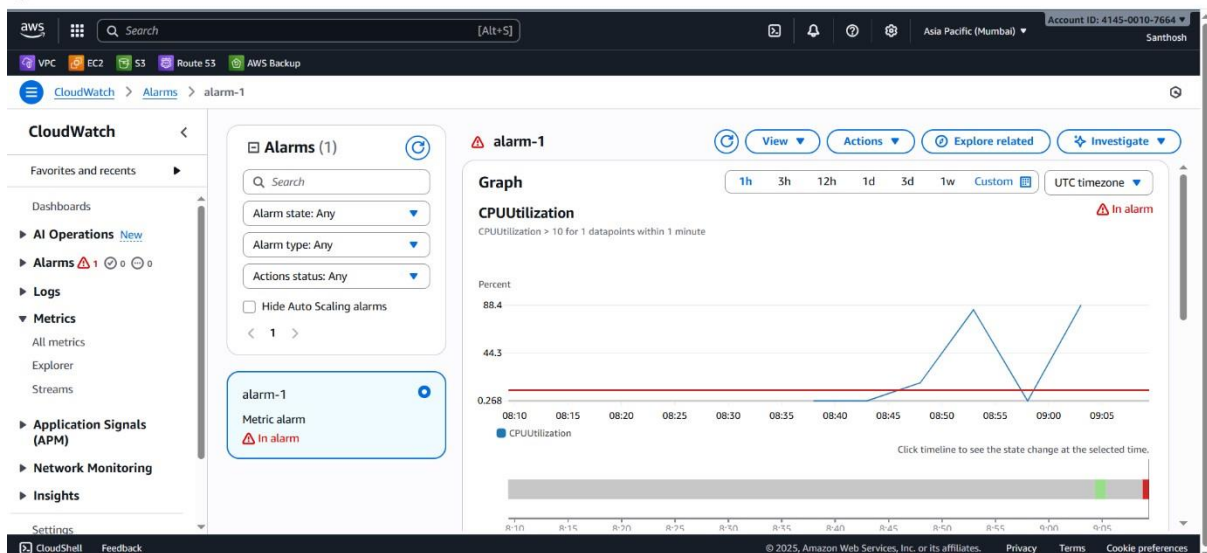
Accessing private instance from public instance.

public Ip of public EC2: 13.126.71.216 and private Ip of public EC2: 10.0.0.16.

private Ip of private EC2: 10.0.133.223



Step 9: Create an Alarm in CloudWatch.



Sending notification whenever a new instance is launched/terminated using autoscaling.

The screenshot shows the AWS Auto Scaling console, specifically the 'Add notifications - optional' step. The notification is configured to send notifications to the SNS topic 'project2-notification' whenever instances are launched or terminated. The recipients are 'santhoshsanthu97401@gmail.com'. The event types are 'Launch' and 'Terminate'. The notification is optional, as indicated by the 'optional' label in the title.

Step 10: created S3 bucket static website hoisting and added index.html and other required files through EC2 instance maintained by IAM role.

EC2VPCRoute 53IAMS3

Amazon S3> Buckets> bucket97401

bucket97401

Info

Objects (3)

Copy S3 URICopy URLOpenDownloadDeleteActionsCreate folderUpload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
404.html	html	September 8, 2025, 20:06:46 (UTC+05:30)	451.0 B	Standard
image.PNG	PNG	September 8, 2025, 20:06:46 (UTC+05:30)	3.6 KB	Standard
index.html	html	September 8, 2025, 20:04:53 (UTC+05:30)	1.2 KB	Standard

EC2VPCRoute 53IAMS3

Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Mon Sep 8 14:13:23 2025 from 13.233.177.3

(ec2-user@ip-10-0-9-178 ~)\$ sudo su

[root@ip-10-0-9-178 ec2-user]# ls

[root@ip-10-0-9-178 ec2-user]# cd /var/www/html

[root@ip-10-0-9-178 ec2-user]# ls

[root@ip-10-0-9-178 ec2-user]# vim index.html

[root@ip-10-0-9-178 ec2-user]# aws s3 cp index.html s3://bucket97401/

upload: ./index.html to s3://bucket97401/index.html

[root@ip-10-0-9-178 ec2-user]#

i-Oe978197fdc954002 (static-website-hosting)

PublicIPs: 65.2.188.116 PrivateIPs: 10.0.9.178

EC2VPCRoute 53IAMS3

IAM> Roles> ec2-s3accessrole

ec2-s3accessrole

Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation dateSeptember 08, 2025, 19:15 (UTC+05:30)

ARNarn:aws:iam:414500107664:role/ec2-s3accessrole

Last activity17 minutes ago

Maximum session duration1 hour

Instance profile ARNarn:aws:iam:414500107664:instance-profile/ec2-s3accessrole

PermissionsTrust relationshipsTagsLast AccessedRevoke sessions

Trusted entities

Entities that can assume this role under specified conditions.

1-23456-{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {

# Micro Degree

## AWS Trainer - Kavya

Bangalore

Karnataka

Contact