Elastic Load Balancing & Auto Scaling

1) Create Launch Template

- Go to **EC2 → Launch Templates**
- Click **"Create launch template"**
- Configure the following:

Template name: web-server-template

AMI: Amazon Linux 2 (ami-0abcdef1234567890)

Instance type: t2.micro

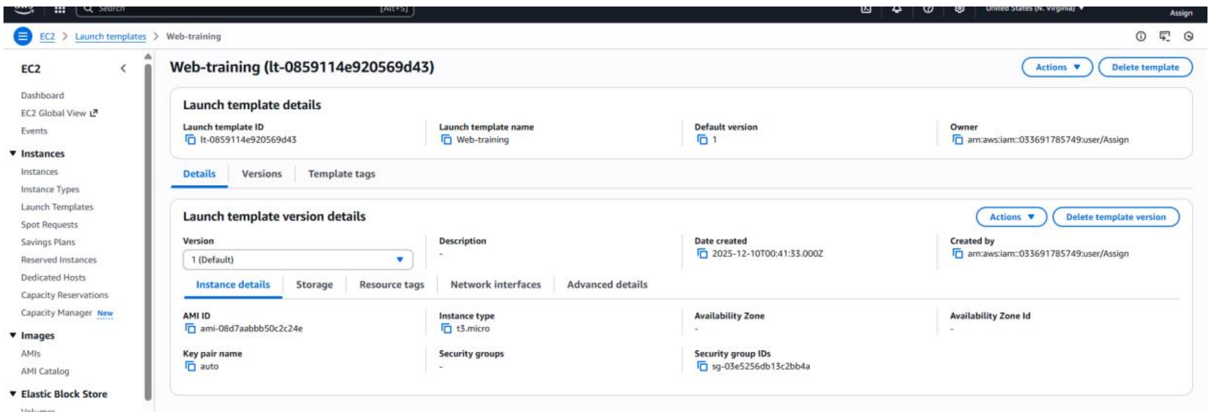
Key pair: Select existing or create new

Security groups: Create new with HTTP (80) and SSH (22)

User Data Script

Add this script in Advanced details → User data:

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Web Server $(hostname -f)</h1>" >
/var/www/html/index.html
```



Web-training (lt-0859114e920569d43)

Launch template details

Launch template ID: lt-0859114e920569d43
Launch template name: Web-training
Default version: 1
Owner: arn:aws:iam::033691785749:user/Assign

Launch template version details

Version: 1 (Default)
Description: -
Date created: 2025-12-10T00:41:33.000Z
Created by: arn:aws:iam::033691785749:user/Assign

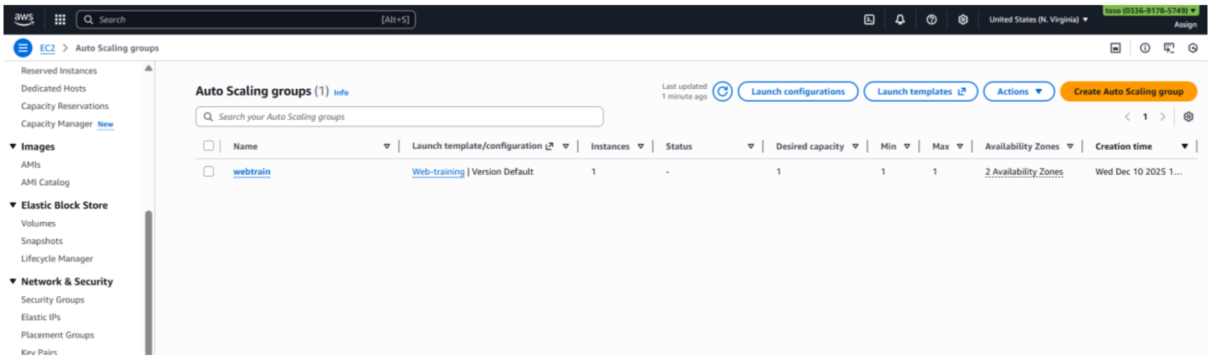
AMI ID: ami-08d7aabbb50c2c24e
Instance type: t3.micro
Availability Zone: -
Availability Zone Id: -
Key pair name: auto
Security groups: -
Security group IDs: sg-03e5256db13c2bb4a

2) Create Auto Scaling Group

1. **Go to EC2 → Auto Scaling Groups**

2. Click "Create Auto Scaling group"
3. Select your launch template

   Auto Scaling group name: webtrain
   Launch template: web-server-template
   VPC: Default VPC
   Subnets: Select 2+ subnets in different AZs



Auto Scaling groups (1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Creation time
webtrain	Web-training | Version Default	1	-	1	1	1	2 Availability Zones	Wed Dec 10 2025 1...

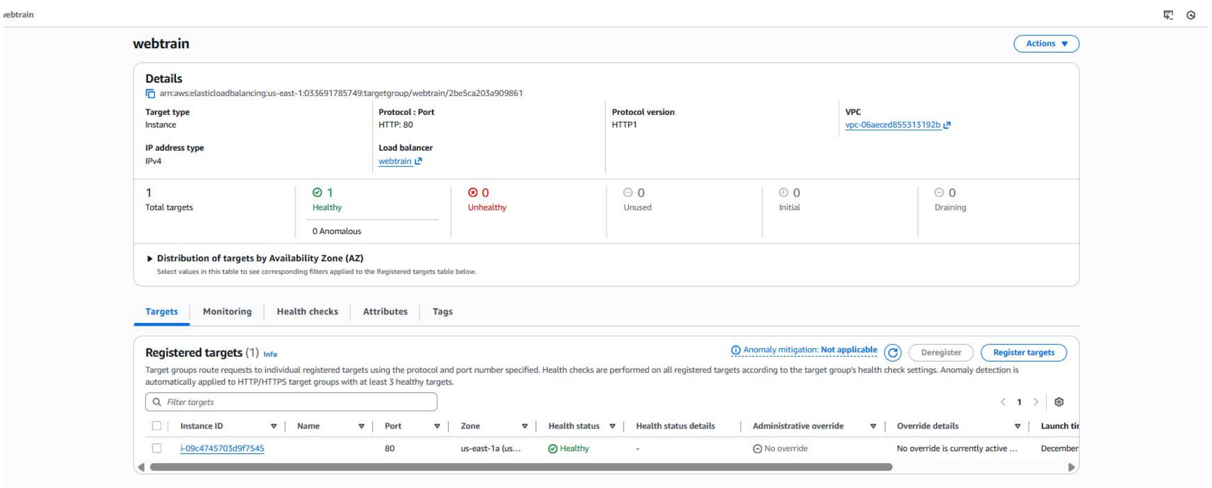
3) Create Application Load Balancer

- Go to **EC2 → Load Balancers**
- Click **"Create Load Balancer"**
- Choose **"Application Load Balancer"**

Name: webtrain
Scheme: Internet-facing
IP address type: IPv4
VPC: Default VPC
Mappings: Select 2+ AZs with public subnets

Target Group Configuration

- **Target type: Instances**
- **Protocol: HTTP**
- **Port: 80**
- **Health check path: /**
- **Health check interval: 30 seconds**



webtrain

Details

arn:aws:elasticloadbalancing:us-east-1:033691785749:targetgroup/webtrain/2be5ca203a909861

Target type Instance	**Protocol : Port** HTTP: 80	**Protocol version** HTTP1	**VPC** vpc-06aeced855313192b
IP address type IPv4	**Load balancer** webtrain		

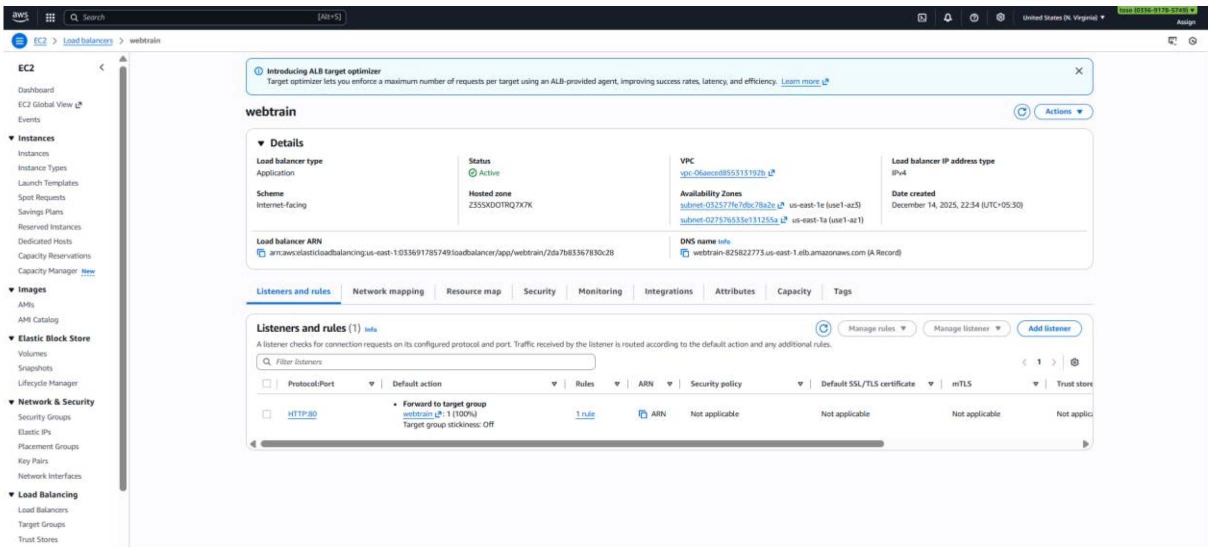
1 Total targets	1 Healthy 0 Anomalous	0 Unhealthy	0 Unused	0 Initial	0 Draining

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (1) Info

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative override	Override details	Launch ti
i-09c4745703d9f7545		80	us-east-1a (us...	Healthy	-	No override	No override is currently active ...	December



Introducing ALB target optimizer
Target optimizer lets you enforce a maximum number of requests per target using an ALB-provided agent, improving success rates, latency, and efficiency. Learn more

webtrain

Details

Load balancer type Application	**Status** Active	**VPC** vpc-06aeced855313192b	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** Z35SXDOTRQ7X7K	**Availability Zones** subnet-032577fe7dbc78a2e us-east-1e (use1-az3) subnet-027576533e131255a us-east-1a (use1-az1)	**Date created** December 14, 2025, 22:34 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:033691785749:loadbalancer/app/webtrain/2da7b83367830c28

DNS name Info
webtrain-825822773.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

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 webtrain: 1 (100%) Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applica

4) Attach Load Balancer to Auto Scaling Group

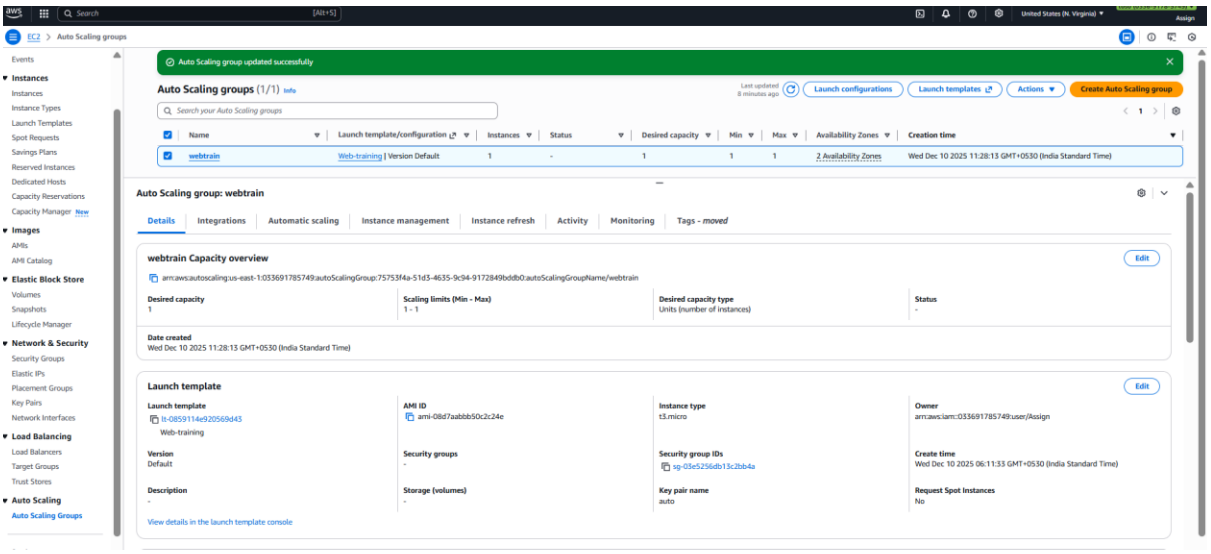
- Go back to **Auto Scaling Groups**
- Select your ASG: **web-server-asg**
- Go to **Details** tab
- Click **"Edit"** in Load balancing section

Load balancing: Enable

Target groups: Select your target group

Health check type: ELB

Health check grace period: 300 seconds



Auto Scaling group updated successfully

Auto Scaling groups (1/1)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Creation time
webtrain	Web-training | Version Default	1	-	1	1	1	2 Availability Zones	Wed Dec 10 2025 11:28:13 GMT+0530 (India Standard Time)

Auto Scaling group: webtrain

webtrain Capacity overview

arn:aws:autoscaling:us-east-1:033691785749:autoScalingGroup:75753f4a-51d3-4635-9c94-9172849bddb0:autoScalingGroupName/webtrain

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
1	1 - 1	Units (number of instances)	-

Date created
Wed Dec 10 2025 11:28:13 GMT+0530 (India Standard Time)

Launch template

Launch template: lt-0859114e920569d43 Web-training	AMI ID: ami-08d7aabbb50c2c24e	Instance type: t3.micro	Owner: arn:aws:iam::033691785749:user/Assign
Version: Default	Security groups: -	Security group IDs: sg-03e5256db13c2bb4a	Create time: Wed Dec 10 2025 06:11:33 GMT+0530 (India Standard Time)
Description: -	Storage (volumes): -	Key pair name: auto	Request Spot Instances: No

View details in the launch template console

5) Configure Dynamic Scaling Policies

6) n ASG, go to **Automatic scaling** tab

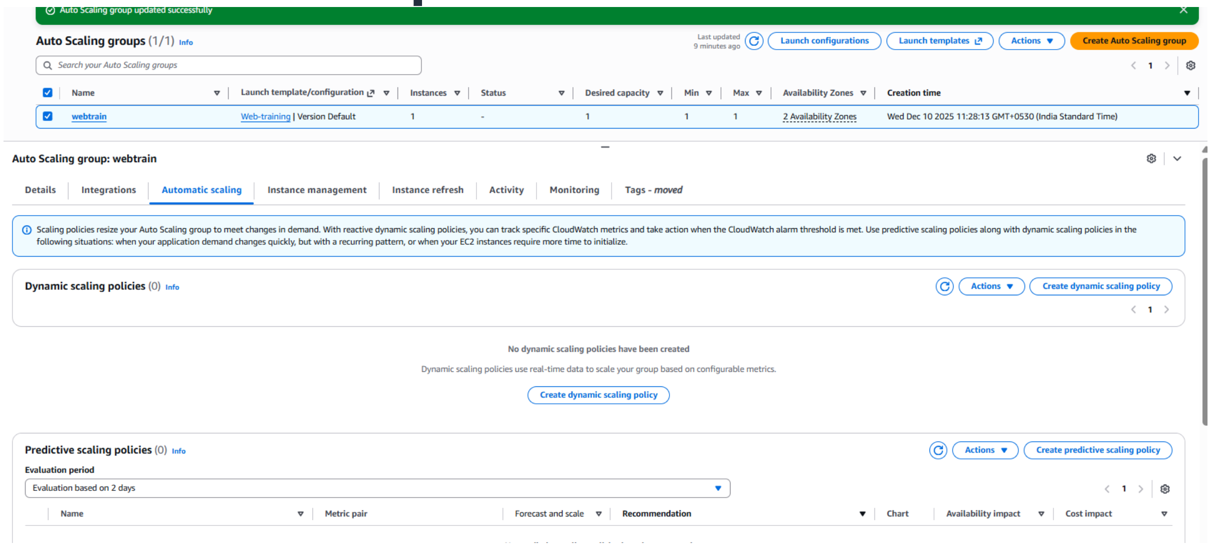
7) Click **"Create dynamic scaling policy"**

Policy type: Target tracking scaling

Metric type: Average CPU Utilization

Target value: 70%

Instance warmup: 300 seconds



Auto Scaling group updated successfully

Auto Scaling groups (1/1)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Creation time
webtrain	Web-training | Version Default	1	-	1	1	1	2 Availability Zones	Wed Dec 10 2025 11:28:13 GMT+0530 (India Standard Time)

Auto Scaling group: webtrain

Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize.

Dynamic scaling policies (0)

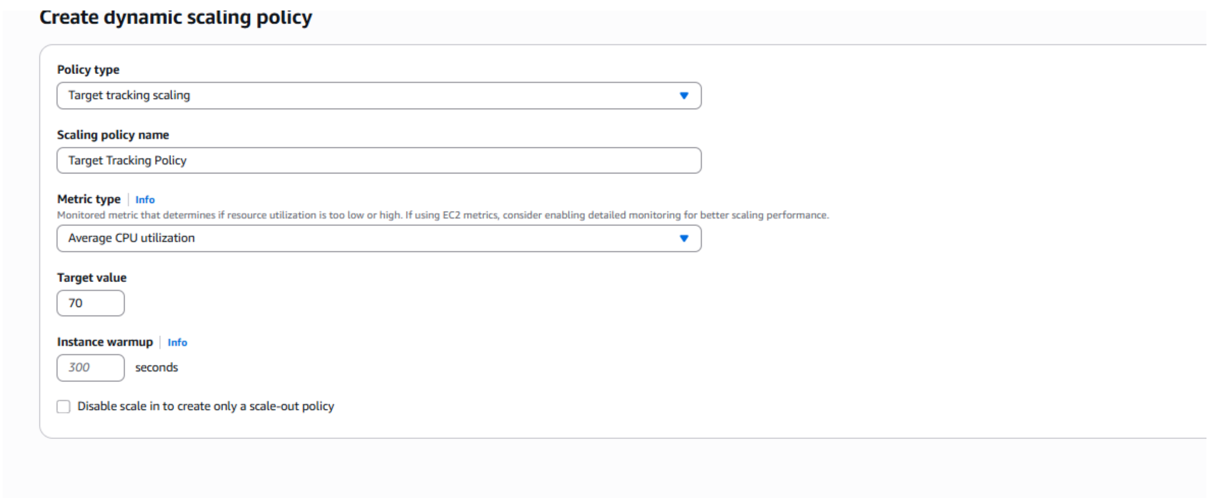
No dynamic scaling policies have been created

Dynamic scaling policies use real-time data to scale your group based on configurable metrics.

Predictive scaling policies (0)

Evaluation period: Evaluation based on 2 days

Name	Metric pair	Forecast and scale	Recommendation	Chart	Availability impact	Cost impact



Create dynamic scaling policy

Policy type
Target tracking scaling

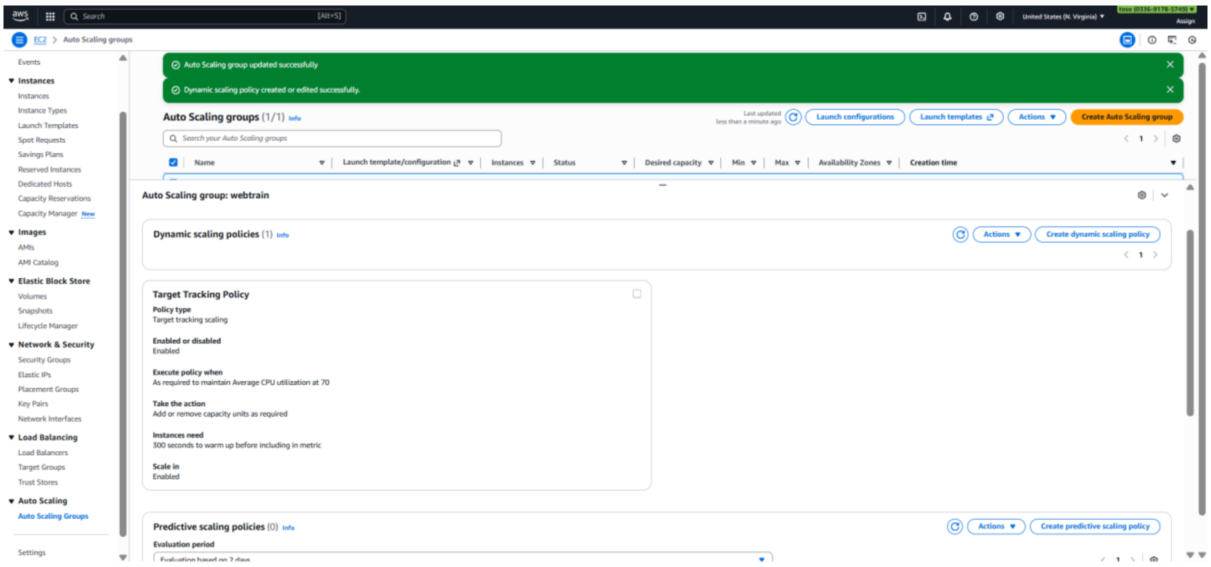
Scaling policy name
Target Tracking Policy

Metric type Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.
Average CPU utilization

Target value
70

Instance warmup Info
300 seconds

☐ Disable scale in to create only a scale-out policy



EC2 > Auto Scaling groups

Auto Scaling group updated successfully

Dynamic scaling policy created or edited successfully.

Auto Scaling groups (1/1) Info

Last updated less than a minute ago

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Creation time

Auto Scaling group: webtrain

Dynamic scaling policies (1) Info

Target Tracking Policy

Policy type
Target tracking scaling

Enabled or disabled
Enabled

Execute policy when
As required to maintain Average CPU utilization at 70

Take the action
Add or remove capacity units as required

Instances need
300 seconds to warm up before including in metric

Scale in
Enabled

Predictive scaling policies (0) Info

Evaluation period
Evaluation based on 2 days