

2.Configure Auto Scaling and Load Balancing Create a launch template for EC2 instances. Set up an auto-scaling group with minimum and maximum instance limits. Attach an application load balancer to distribute traffic.

Auto Scaling Group + Load Balancer on AWS

This guide explains **from scratch** how to set up an Auto Scaling Group (ASG) with a Load Balancer (ALB) to run a simple website on EC2 instances.

We'll do this in 5 main parts:

1. **Launch Template** → Blueprint for EC2.
2. **Security Group** → Firewall for EC2.
3. **Auto Scaling Group (ASG)** → Manages EC2 automatically.
4. **Application Load Balancer (ALB)** → Distributes traffic.
5. **Install Apache Web Server** → Host a webpage.

Step 1: Create a Security Group

A **Security Group** acts like a firewall. It controls what traffic can enter and leave your EC2 instance.

1. Go to **EC2 → Security Groups → Create Security Group**.
2. Fill details:
 - **Name:** asg-alb-sg
 - **Description:** Security Group for ASG + ALB
 - **VPC:** Default VPC
3. Under **Inbound Rules**, add:
 - **SSH (22)** → Source: My IP (to connect to instance)
 - **HTTP (80)** → Source: Anywhere (0.0.0.0/0)
4. Leave **Outbound Rules** as default (all traffic allowed).
5. Click **Create Security Group**

This allows us to connect via SSH and serve a website.

sg-018a21940624ad48f - asg-alb-sg

Details

Security group name asg-alb-sg	Security group ID sg-018a21940624ad48f	Description Security group for Auto Scaling + A LB project	VPC ID vpc-02e06b741fc8ee298
Owner 858265286505	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (3)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0021c2a067334209a	IPv4	SSH	TCP	22
-	sgr-05ef5469b94a9c697	IPv4	HTTPS	TCP	443
-	sgr-025f870caac2641b4	IPv4	HTTP	TCP	80

sg-018a21940624ad48f - asg-alb-sg

Details

Security group name asg-alb-sg	Security group ID sg-018a21940624ad48f	Description Security group for Auto Scaling + A LB project	VPC ID vpc-02e06b741fc8ee298
Owner 858265286505	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

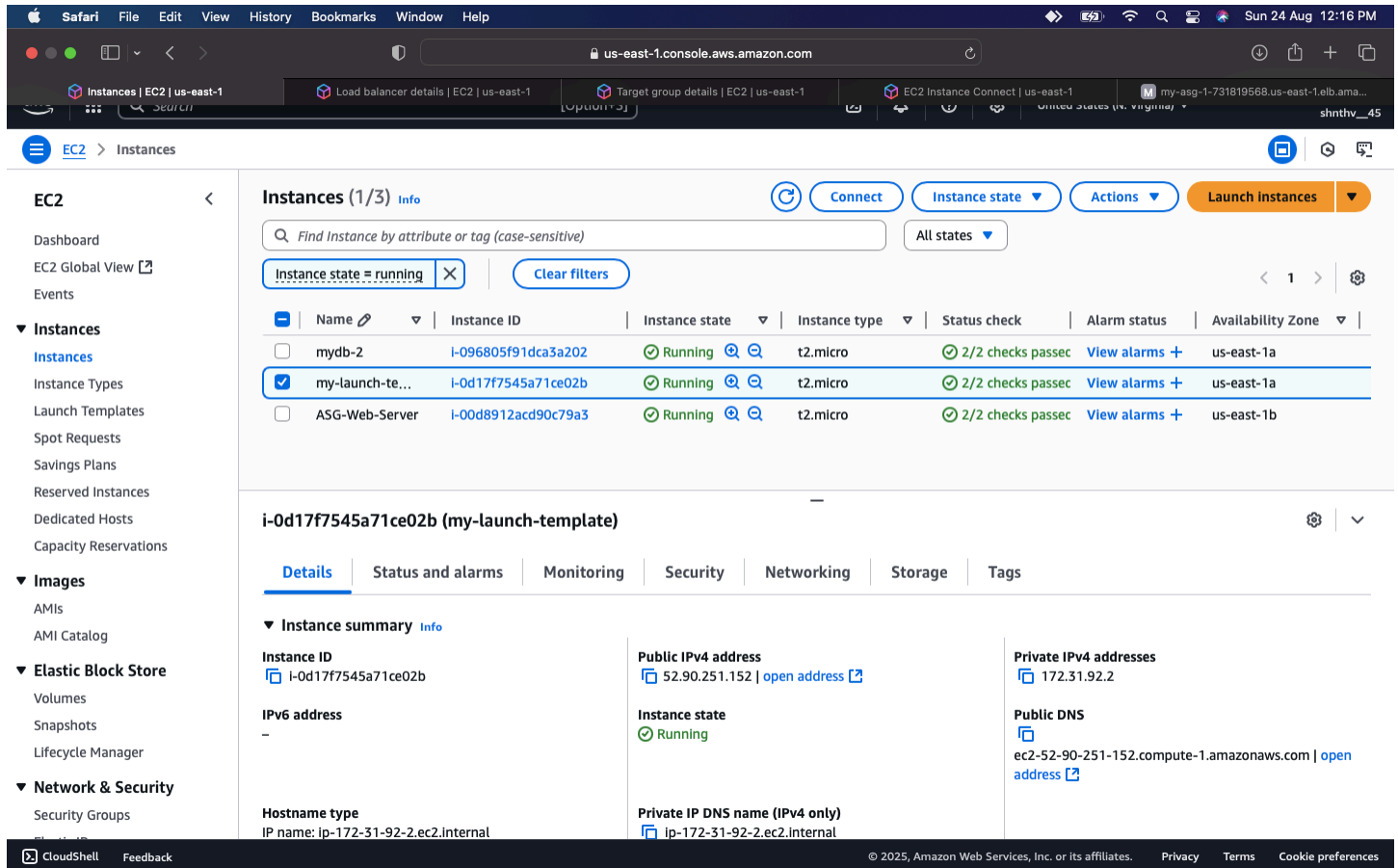
Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-02a2dc2a563a4c63d	IPv4	All traffic	All	All

Step 2: Create a Launch Template

A **Launch Template** is a blueprint. It tells AWS how new EC2 instances should be created (OS, size, key, firewall).

1. Go to **EC2 → Launch Templates → Create Launch Template**.
2. Fill details:
 - **Name:** my-launch-template
 - **AMI (OS):** Amazon Linux 2023 (free tier eligible)
 - **Instance type:** t2.micro
 - **Key pair:** Choose your .pem key (example: sonu-3.pem)
 - **Security Group:** Select asg-alb-sg (created above)
 - **Storage:** 8GB (default EBS volume)
3. Click **Create Launch Template**



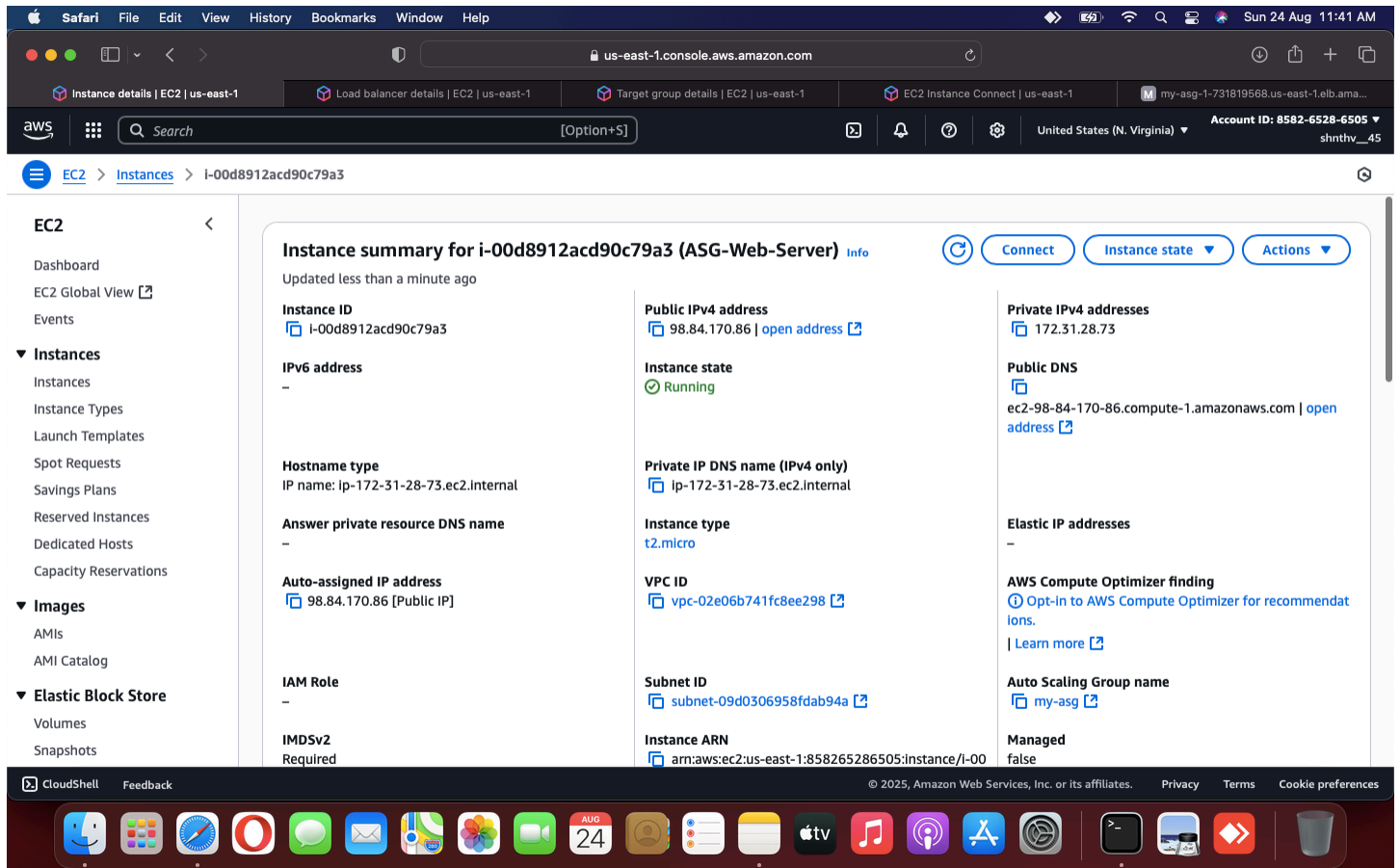
Now AWS knows how to launch new EC2 instances automatically.

Step 3: Create an Auto Scaling Group (ASG)

An **Auto Scaling Group** ensures the right number of EC2 instances are always running. If one crashes, it creates a new one. If load increases, it adds more.

1. Go to **EC2 → Auto Scaling Groups → Create Auto Scaling Group**.
2. Select:
 - **Launch Template:** my-launch-template
 - **Name:** my-asg
3. Select **VPC:** Default VPC
 - Choose **at least 2 subnets** in different Availability Zones (e.g. us-east-1a and us-east-1b) → This ensures high availability.
4. Under **Load Balancing**, select:
 - **Attach to an existing load balancer** → (We'll create ALB next)

- Or choose **Create new ALB** directly here.
- 5. Configure **Group Size**:
 - **Desired capacity**: 1 (start with one instance)
 - **Min**: 1
 - **Max**: 3 (allows scaling up to 3 instances)
- 6. Add **Scaling Policy**:
 - **Target tracking policy** → Keep average CPU utilization at **50%**
- 7. Notifications (Optional):
 - Choose or create **SNS topic** → Get emails when instances launch/terminate.
- 8. Add **Tags**:
 - Example → Key: Name | Value: ASG-Web-Server



Click **Create Auto Scaling Group**

Now AWS will launch your first EC2 instance automatically.

Step 4: Create an Application Load Balancer (ALB)

A **Load Balancer** makes sure user traffic is distributed across healthy instances. If one fails, traffic goes to another.

1. Go to **EC2 → Load Balancers → Create Load Balancer → Application Load Balancer**.
2. Fill details:
 - **Name**: my-asg-alb
 - **Scheme**: Internet-facing
 - **IP type**: IPv4

3. Network:
 - Choose **VPC**: Default
 - Select **2 subnets** in different Availability Zones (same as ASG).
4. Security Group:
 - Select **asg-alb-sg** (so it allows HTTP).
5. Listener:
 - Add **HTTP on port 80**.
6. Create **Target Group**:
 - **Name**: my-asg-target-group
 - **Target type**: Instances
 - **Protocol**: HTTP:80
 - **Health Check Path**: / (root page)
7. Register instances → Select EC2 instances from your ASG.
8. Click **Create Load Balancer**

The screenshot displays the AWS Management Console interface for an Application Load Balancer named 'my-asg-1'. The top navigation bar shows the user is logged in as 'shnthv_45' with account ID '8582-6528-6505' in the 'us-east-1' region. The left sidebar contains navigation links for EC2, Load balancers, and my-asg-1. The main content area shows the 'Details' tab for the load balancer, which is in an 'Active' state. Key details include: Load balancer type: Application; Scheme: Internet-facing; Status: Active; Hosted zone: Z35SXDOTRQ7X7K; VPC: vpc-02e06b741fc8ee298; Availability Zones: subnet-0c3892124977de756 (us-east-1a) and subnet-09d0306958fdab94a (us-east-1b); Load balancer IP address type: IPv4; Date created: August 24, 2025, 10:18 (UTC+05:30); Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:858265286505:loadbalancer/app/my-asg-1/6601d3f6fe1dbdde; DNS name: my-asg-1-731819568.us-east-1.elb.amazonaws.com (A Record). The bottom of the screen shows a macOS dock with various application icons.

my-asg-1			
Load balancer type Application	Status Active	VPC vpc-02e06b741fc8ee298	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-0c3892124977de756 (us-east-1a) subnet-09d0306958fdab94a (us-east-1b)	Date created August 24, 2025, 10:18 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:858265286505:loadbalancer/app/my-asg-1/6601d3f6fe1dbdde		DNS name my-asg-1-731819568.us-east-1.elb.amazonaws.com (A Record)	

The screenshot shows the AWS Management Console for the 'us-east-1' region. The left sidebar displays the navigation menu with categories like EC2, Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled 'Load balancer details | EC2 | us-east-1' and shows the configuration for the 'my-asg-1' load balancer.

Load balancer details:

- Load balancer type:** Application
- Status:** Active
- VPC:** vpc-02e06b741fc8ee298
- Load balancer IP address type:** IPv4
- Scheme:** Internet-facing
- Hosted zone:** Z35SXDOTRQ7X7K
- Availability Zones:** subnet-0c3892124977de756 (us-east-1a), subnet-09d0306958fdab94a (us-east-1b)
- Date created:** August 24, 2025, 10:18 (UTC+05:30)
- Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:858265286505:loadbalancer/app/my-asg-1/6601d3f6e1dbdde
- DNS name:** my-asg-1-731819568.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1):

- Protocol:** HTTP
- Port:** 80
- Default action:** Forward to target group my-asg-target-group (100%)
- Rules:** 1 rule
- ARN:** Not applicable
- Security policy:** Not applicable

The screenshot shows the AWS Management Console for the 'us-east-1' region, specifically the 'Target group details | EC2 | us-east-1' page for the 'my-asg-target-group'.

Target group details:

- IPV4:** my-asg-1
- Total targets:** 1
- Healthy:** 1
- Unhealthy:** 0
- Unused:** 0
- Initial:** 0
- Draining:** 0
- Anomalous:** 0

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):

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...
i-00d8912acd90c79a3	ASG-Web-Server	80	us-east-1b (us...)	Healthy	-	No override

Now ALB sits in front of your ASG and distributes traffic.

Step 5: Install Apache on EC2 Instance

Now we'll install a **web server** (Apache) to serve a webpage.

1. Go to **EC2 → Instances → Select instance → Connect**
(or SSH from terminal: `ssh -i sonu-3.pem ec2-user@<Public-IP>`).
2. Run these commands one by one:

Update all packages

```
sudo yum update -y
```

Install Apache (httpd)

```
sudo yum install -y httpd
```

Start Apache

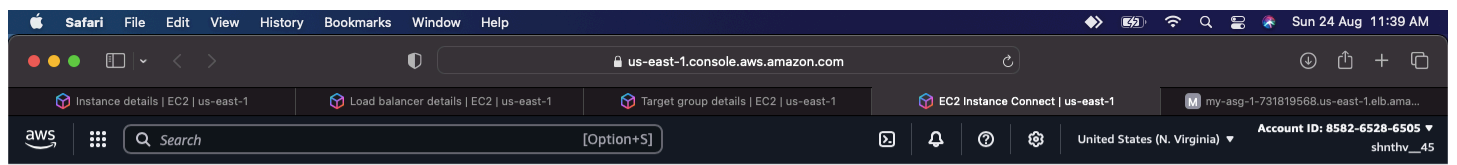
```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

Create a custom HTML page

```
echo '<h1 style="font-size:50px; color:green; text-align:center;">Hello from Auto Scaling  
Group Instance</h1>
```

```
<p style="text-align:center; font-size:20px;">This page is served through Apache on an EC2  
instance managed by an Auto Scaling Group and Load Balancer.</p>' | sudo tee  
/var/www/html/index.html
```



```
Complete!
[ec2-user@ip-172-31-28-73 ~]$ sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
echo '<h1 style="font-size:50px; color:green; text-align:center;">Hello from Auto Scaling Group Instance</h1><p style="text-align:center; font-size:20px;">This page is served through Apache on an EC2 instance managed by an Auto Scaling Group and Load Balancer.</p>' | sudo tee /var/www/html/index.html
Last metadata expiration check: 0:45:29 ago on Sun Aug 24 05:23:27 2025.
=====
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

  Version 2023.8.20250808:
    Run the following command to upgrade to 2023.8.20250808:

      dnf upgrade --releasever=2023.8.20250808

  Release notes:
    https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.8.20250808.html

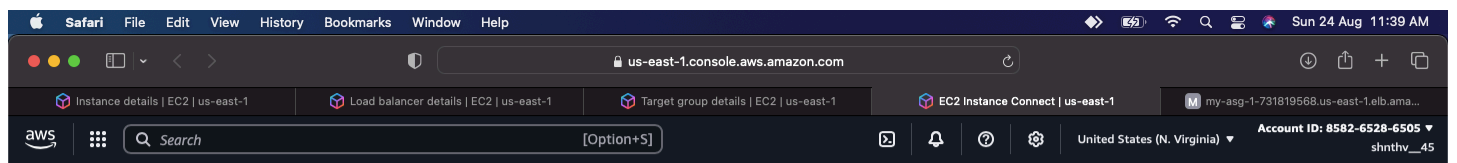
  Version 2023.8.20250818:
    Run the following command to upgrade to 2023.8.20250818:

      dnf upgrade --releasever=2023.8.20250818

  Release notes:
    https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.8.20250818.html
```

i-00d8912acd90c79a3 (ASG-Web-Server)

PublicIPs: 98.84.170.86 PrivateIPs: 172.31.28.73



```
Version 2023.8.20250808:
  Run the following command to upgrade to 2023.8.20250808:

    dnf upgrade --releasever=2023.8.20250808

  Release notes:
    https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.8.20250808.html

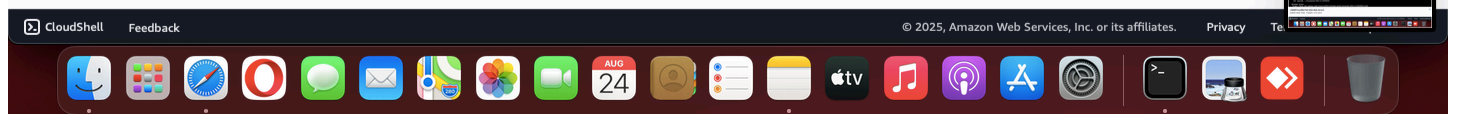
Version 2023.8.20250818:
  Run the following command to upgrade to 2023.8.20250818:

    dnf upgrade --releasever=2023.8.20250818

  Release notes:
    https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.8.20250818.html
=====
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:45:30 ago on Sun Aug 24 05:23:27 2025.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
<h1 style="font-size:50px; color:green; text-align:center;">Hello from Auto Scaling Group Instance</h1><p style="text-align:center; font-size:20px;">This page is served through Apache on an EC2 instance managed by an Auto Scaling Group and Load Balancer.</p>
[ec2-user@ip-172-31-28-73 ~]$
```

i-00d8912acd90c79a3 (ASG-Web-Server)

PublicIPs: 98.84.170.86 PrivateIPs: 172.31.28.73



Now your EC2 is serving a webpage.

Step 6: Test Your Setup

1. Copy the **DNS of your Load Balancer** (looks like my-asg-1-731819568.us-east-1.elb.amazonaws.com).
2. Open it in your browser.
3. You should see:

“Hello from Auto Scaling Group Instance” in big green letters.

If you refresh multiple times, ALB may send requests to different EC2 instances.

If one instance fails, ASG will replace it automatically.

