

Highly Available Web Application with AWS Auto Scaling and Load Balancing

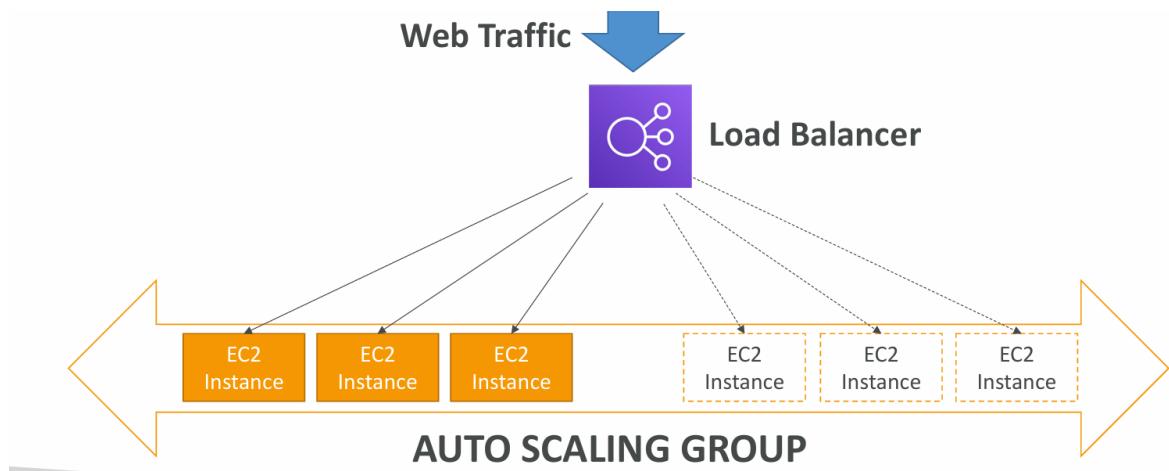
What is ASG?

An Auto Scaling Group (ASG) is a service in AWS that automatically manages the number of EC2 instances in your application.

- When traffic or usage increases (for example, CPU goes above 70%), it adds more EC2 instances up to the maximum capacity to handle the load.
- When traffic goes back to normal, it removes extra EC2 instances, so you only pay for what you use.
- It always keeps your minimum number of servers running and makes sure your application stays available.

Benefits of ASG:

- Ensures scalability → handles sudden demand.
- Provides high availability → no downtime for customers.
- Improves cost efficiency → pay only for the extra servers used during peak time.



Procedure For Creating the ASG group:

- First Create the Load balancer and target Group.
- Create the ASG in console.
- Run and Check it.

Project Description:

This project implements an Auto Scaling Group (ASG) with an Application Load Balancer (ALB) to ensure high availability and scalability of a web application. The ALB distributes traffic evenly across EC2 instances while continuously monitoring their health. The ASG automatically scales out by launching new instances when web traffic increases and scales in by terminating instances during low demand. This dynamic adjustment improves performance during peak loads and reduces costs during idle times. Overall, the solution delivers an efficient, reliable, and cost-effective cloud infrastructure.

Steps For Creating the Target Group and ELB balancer:

1.Create a Target Group

1. Go to EC2 → Target groups → Create target group.
2. Target type: Instances

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

3.Name: myloadtarget (or your myloadtargetgroup)

Target group name

MyLoadTargetgroup

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

4.Protocol: HTTP, Port: 80

Protocol version

HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

a. VPC: your default or the one where instances run

b. Health checks

i. Protocol: HTTP

ii. Path: /

iii. (Leave thresholds/intervals default)

Cancel **Next**

c. Next → Register targets

- i. Select both instances → Include as pending below → Create target group

2. Create the Load Balancer:

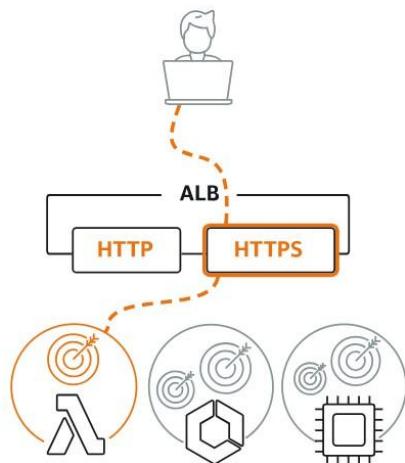
1. Login to AWS Management Console
 - o Go to EC2 Dashboard → Load Balancers (under Load Balancing).
2. Click on “Create Load Balancer”



- o Select Application Load Balancer (ALB).

Load balancer types

Application Load Balancer [Info](#)



3. Step 1: Configure Load Balancer

- o Name: Give your ALB a name (example: myasgloadbalancer).
- o Scheme: Select Internet-facing (if public access is needed).
- o IP Address type: Choose IPv4.

Basic configuration

Load balancer name

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

myasgloadbalancer

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 address types. Public IPv4 addresses have an additional cost.

IPv4

Includes only IPv4 addresses.

Dualstack

Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4

Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

o Availability Zones:

- Select your VPC.
- Choose at least 2 subnets

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 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-04ab91921d0e632c4

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-071d01d853fb6d886

IPv4 subnet CIDR: 172.31.0.0/20

ap-south-1c (aps1-az2)

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-05df90917fed42b1

IPv4 subnet CIDR: 172.31.16.0/20

4. Step 2: Configure Security Settings (Optional if HTTPS)

- o If HTTPS listener, upload/create SSL certificate.
- o If HTTP only → Skip this.

5. Step 3: Configure Security Groups

- o Create or select a Security Group for ALB.
- o Ensure rules allow inbound traffic on port 80 (HTTP)

The screenshot shows the AWS CloudFormation Inbound and Outbound rules configuration interface. It includes sections for Inbound rules (Protocol: HTTP, Port range: 80, Source: Any... 0.0.0.0/0) and Outbound rules (Protocol: All traffic, Port range: All, Destination: Cust... 0.0.0.0/0). There are 'Add rule' buttons for both sections.

6. Step 4: Configure Listeners & Routing

- Listener: By default, ALB creates an HTTP listener on port 80.
- Default Action: Forward traffic to a Target Group.

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you registered targets.

The screenshot shows the AWS CloudFormation Listener configuration interface for Listener HTTP:80. It displays the Protocol (HTTP), Port (80), and Default action (Forward to target group myelastictarget). A 'Create target group' button is also present.

7. Step 5: Configure Target Group

- Click Create Target Group.
- Target Type: Instance (for EC2).
- Target Group Name: Example my-target-group.
- Protocol: HTTP (port 80).
- VPC: Choose same VPC as ALB.
- Health Checks: Keep default (/ on HTTP).
- Register Targets: Select EC2 instances (like your web servers).

8. Step 6: Review and Create

- Review all configurations

Steps for Creating the Autoscaling group:

1. Set the name for Auto Scaling group name

Name

Auto Scaling group name

Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

2.launch template

1.Create a launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - required

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance | Info

Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Enter the template name.

3.template version description (optional)

4.in Quick Start Select amazon linux (preferred)

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

[Recents](#)[Quick Start](#)

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-0861f4e788f5069dd (64-bit (x86), uefi-preferred) / ami-0fad8318b9405c6fb (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

5.Instance type (t2.micro free tier eligible)

The screenshot shows the 'Instance type' section of the AWS console. It highlights the 't2.micro' instance type, which is described as 'Free tier eligible'. Below the instance type, there is a table of pricing information for various operating systems. To the right, there are buttons for 'All generations' and 'Compare instance types'. A note at the bottom states 'Additional costs apply for AMIs with pre-installed software'.

6.leave key pair as default (not required)

7 Network Settings

In that select the security groups (launch -wizard 1)

The screenshot shows the 'Network settings' section of the AWS console. It includes fields for 'Subnet' (set to 'Don't include in launch template'), 'Availability Zone' (set to 'Not applicable for EC2 Auto Scaling'), and 'Firewall (security groups)' (set to 'Select existing security group'). A dropdown menu shows a selected security group: 'launch-wizard-1 sg-07e7d0cf5f773a586 X'. There is also a 'Create new subnet' and 'Enable additional zones' button. A 'Compare security group rules' link is available. A 'Security groups' section is also visible.

8.Advanced Details for default Web Server Creation.

```

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

```

Step 3 :

Choose instance launch Options

Choose the Availability Zones and subnets.

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

aps1-az1 (ap-south-1a) | subnet-04ab91921d0e632c4
172.31.32.0/20 Default

aps1-az2 (ap-south-1c) | subnet-05dfe90917fed42b1
172.31.16.0/20 Default

aps1-az3 (ap-south-1b) | subnet-071d01d853fb6886
172.31.0.0/20 Default

Create a subnet

And leave all as default.

Step 4:

Integrate with other Services-optional

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

<input type="radio"/> No load balancer Traffic to your Auto Scaling group will not be fronted by a load balancer.	<input checked="" type="radio"/> Attach to an existing load balancer Choose from your existing load balancers.	<input type="radio"/> Attach to a new load balancer Quickly create a basic load balancer to attach to your Auto Scaling group.
--	---	---

And Enable the Turn on Elastic Load Balancing health check if there is any unwanted health issues or error that will automatically terminated

and it will create the new instances.

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks

Always enabled

Additional health check types - optional | Info

Turn on Elastic Load Balancing health checks Recommended

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check

Configure Group Size and scaling optional:

Set as Desired capacity means it manual I will set as 2

Scaling set Min Desired Capacity 2 and max Desired capacity I will set as 3 and leave all it as default

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or automatically.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configuration attributes.

Units (number of instances)



Desired capacity

Specify your group size.

2

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

2

Max desired capacity

3

Equal or less than desired capacity

Equal or greater than desired capacity

Step 7:

If you need notification set as your id it is optional

Add tags -optional

Review and create it.

Auto Scaling groups (1) Info									
Last updated less than a minute ago C Launch configurations Launch templates Actions Create Auto Scaling group									
<input type="text"/> Search your Auto Scaling groups									
Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Available		
myasg-webtemplate	myasg-webtemplate Version Default	0	Updating capacity...	2	2	3	3 Available		

Wait for active ..

Go To load balancer and check it :

Auto Scaling groups (1) Info									
Last updated less than a minute ago C Launch configurations Launch templates Actions Create Auto Scaling group									
<input type="text"/> Search your Auto Scaling groups									
Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Available		
myasg-webtemplate	myasg-webtemplate Version Default	0	Updating capacity...	2	2	3	3 Available		

Test the Load Balancer

Load balancers (1/1)									
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.									
Actions Create load balancer									
<input type="text"/> Filter load balancers									
Name	State	Type	Schema	IP address type	VPC ID				
myasgloadbalancer	Active	application	Internet-facing	IPv4	vpc-0727d00193d5261b2				

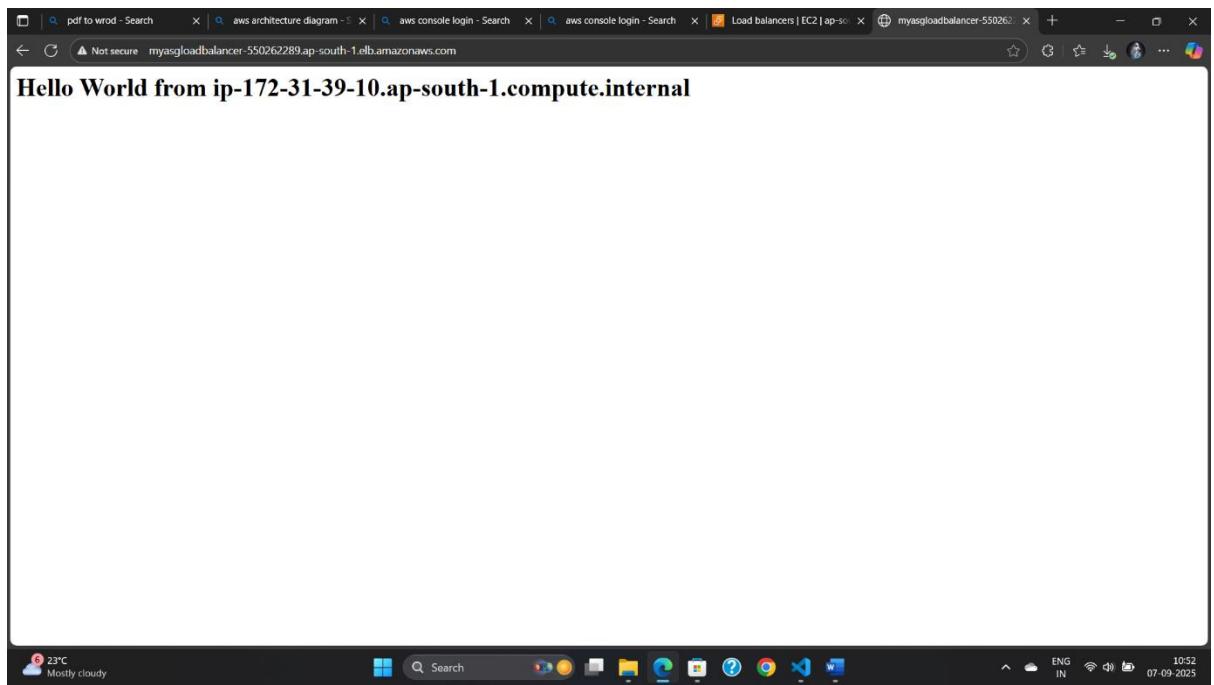
1. Open the ALB's DNS name from Load balancer → Description

- Looks like: my-elastic-load-balancer-123456.ap-south-1.elb.amazonaws.com

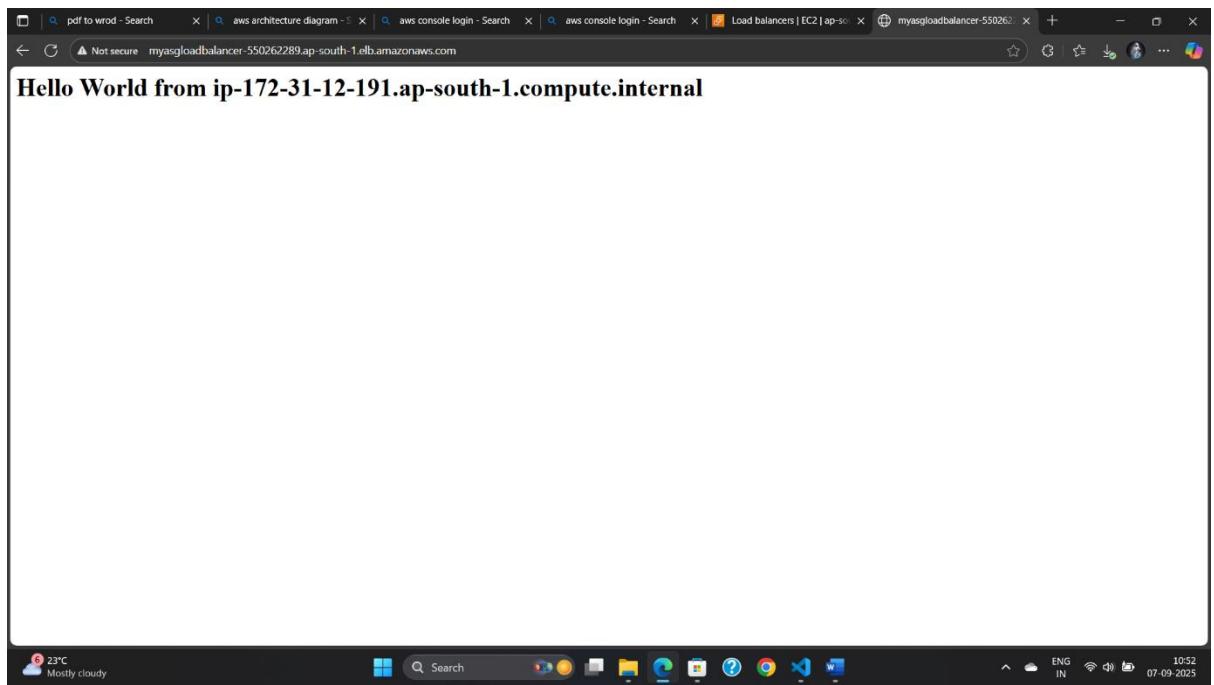
DNS name [Info](#)

 MyLoadBalancer-582097276.ap-south-1.elb.amazonaws.com (A Record)

First Instance :



Second Instance 2 :



Instance Check in Instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon
		i-06145837dc0871794	Terminated	t2.micro	-	View alarms +	ap-south-1b
		i-0503b834b8e9a31c6	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b
		i-0e4a158392ee6b667	Terminated	t2.micro	-	View alarms +	ap-south-1a
		i-008cbcd0c7b282e6f	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a

More Projects on GitHub

Explore more of my cloud, AWS, Linux, projects on my GitHub profile:

- [Sanjaykumar-P \(Sanjaykumar-P\)](#)

Portfolio

Visit my personal portfolio website to see detailed case studies, projects, and achievements:

- [Sanjaykumar P - Linux & Cloud Specialist](#)