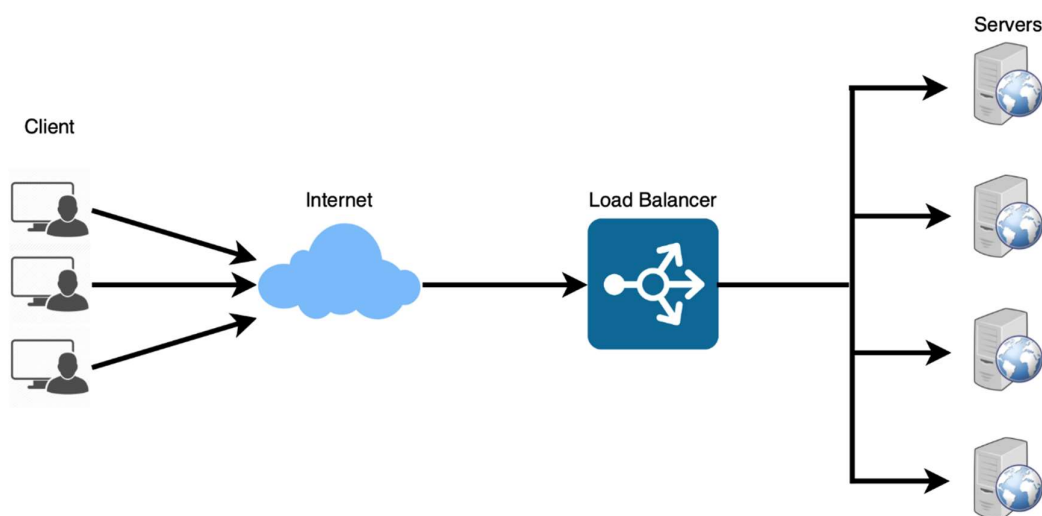


### Elastic Load Balancing:

- Load balancing is the process of distributing workload evenly across multiple servers.
- This helps spread the traffic across a cluster of servers to enhance applications, websites responsiveness and availability.
- Load Balancers also detect the health of back end resource and do not send traffic to servers, which cannot achieve requests, along these lines load balancers can improve network and application performance by controlling and handling applications and network sessions automatically by using various algorithms.
- Load balancers can help Minimize the risk of denial-of-service attacks in addition to providing simple distributed service to multiple servers, grant legitimate users to access services without interruption, protect against single-point failures and avoid network traffic bottlenecks.
- Therefore, we can say that the LB is first line of defense against DDOS.
- ELB automatically distributes your incoming application traffic across all the EC2 instances that you are running.



Elastic Load Balancing provides four types of load balancers that can be used.

### Application Load Balancer:

- Routes and load balances at the application layer (HTTP/HTTPS), and supports path-based routing.
- An Application Load Balancer can route requests to ports on one or more registered targets, such as EC2 instances, in your virtual private cloud (VPC).

### Network Load Balancer:

- Routes and load balances at the transport layer (TCP/UDP Layer-4), based on address information extracted from the Layer-4 header.
- Network Load Balancers can handle traffic bursts, retain the source IP of the client, and use a fixed IP for the life of the load balancer.

### Gateway Load Balancer:

- Gateway Load Balancers work with virtual appliances that support the GENEVE protocol.

### Classic Load Balancer:

- Routes and load balances either at the transport layer (TCP/SSL), or at the application layer (HTTP/HTTPS).

## Create a Classic Load Balancer:

1. Login to AWS Console.
2. Go to EC2.
3. Create a security group for CLB. Here we should allow **only HTTP access** under inbound rule and allow **all traffic** under outbound rule.

EC2 > Security Groups > sg-064098f46b9f61575 - CLB-SG > Edit inbound rules

### Edit inbound rules info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>info</small>	Protocol <small>info</small>	Port range <small>info</small>	Source <small>info</small>	Description - optional <small>info</small>	
-	HTTP	TCP	80	Anywhere-I...	Allow HTTP Access	Delete

[Add rule](#)

EC2 > Security Groups > sg-064098f46b9f61575 - CLB-SG > Edit outbound rules

### Edit outbound rules info

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Security group rule ID	Type <small>info</small>	Protocol <small>info</small>	Port range <small>info</small>	Destination <small>info</small>	Description - optional <small>info</small>	
sg-04d1672922f49e07a	All traffic	All	All	Anywhere-I...	Allow HTTP Access	Delete

[Add rule](#)

4. Before creating a CLB, let's create 3 EC2 instances, on top of which we will create a CLB.
5. Create a separate Security Group for EC2 instances. For the EC2 instances, we will allow both HTTP and HTTPS access.

EC2 > Security Groups > sg-0b0e46ba953101b16 - ec2-sg > Edit inbound rules

### Edit inbound rules info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>info</small>	Protocol <small>info</small>	Port range <small>info</small>	Source <small>info</small>	Description - optional <small>info</small>	
sg-01b884a4e17d5343d	SSH	TCP	22	Custom	For SSH Access	Delete
sg-027617b4c39f37e0e	HTTP	TCP	80	Custom	Request from CLB	Delete
-	HTTPS	TCP	443	Custom		Delete

[Add rule](#)

Source dropdown menu options:

- 0.0.0.0/0
- sg-064098f46b9f61575
- Use: "sg-064098f46b9f61575"
- CIDR blocks
- Security Groups
- CLB-SG | sg-064098f46b9f61575
- Prefix lists

[Cancel](#) [Preview changes](#) [Save](#)

6. While launching EC2 instances, add following script under user data section.

```
#!/bin/bash
apt-get update
apt-get install nginx -y
echo "<h1> This is $(hostname) </h1>" > /var/www/html/index.html
```

7. Go to **EC2 Dashboard** and click on **Load Balancers**.
8. Click on **Create Load Balancer**, select **Classic Load Balancer** and click on **Create**.
9. Give a name to Load Balancer.
10. Scheme – Internet-Facing.
11. Select all the default availability zones.
12. Assign the security group that we have created previously for CLB.
13. No changes required under **Listeners and routing** and **Health checks**.
14. Under **Instances**, add the 3 EC2 instances here.

### Add instances

Select EC2 instances to register to your load balancer. Requests will be routed to registered instances that meet the health check requirements. For maximum fault tolerance, we recommend maintaining approximately equivalent numbers of instances in each Availability Zone enabled for the load balancer. If demand on your instances changes, you can register or deregister instances without disrupting the flow of requests to your application. [Learn more](#)

VPC  
vpc-0ae7fd134cd2696c

Available instances (3/3)

< 1 > ⚙

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Public IPv4 address	Subnet ID
<input checked="" type="checkbox"/>	i-07c287bab52147bf2	instance	Running	ec2-sg	ap-south-1b	3.108.59.118	subnet-09bf6f985da526fa0
<input checked="" type="checkbox"/>	i-02d695816d237ec06	instance	Running	ec2-sg	ap-south-1b	3.110.185.21	subnet-09bf6f985da526fa0
<input checked="" type="checkbox"/>	i-01219de2895557f10	instance	Running	ec2-sg	ap-south-1b	13.235.83.102	subnet-09bf6f985da526fa0

Cancel

Confirm

15. Now click on **Create Load balancer**, Load Balancer will be created.
16. Copy the DNS Name of the CLB, and access in browser.

### Delete a Load Balancer:

Select the Load Balancer, Click on Actions and Click on Delete Load balancer. After giving confirmation, it will get deleted.

### Auto Scaling:

- AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.
- Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.
- The service provides a simple, powerful user interface that lets you build scaling plans for AWS resources.
- Create a **Launch Template** before creating **Auto Scaling Group**.

### Create a Launch Template:

1. Go to EC2 Dashboard and click on **Launch Templates** present under **Instances**.
2. Click on **Create New 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

► Template tags

► Source template

### 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) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents

Quick Start

Don't include in launch template













Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

#### Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-0dee22c13ea7a9a67 (64-bit (x86)) / ami-0c8eea98010057bd0 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

#### 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

64-bit (x86) ▼

AMI ID

ami-0dee22c13ea7a9a67

Username ⓘ

ubuntu

Verified provider

## ▼ Instance type [Info](#) | [Get advice](#)

Advanced

### 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  
On-Demand Windows base pricing: 0.017 USD per Hour  
On-Demand RHEL base pricing: 0.0268 USD per Hour  
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour  
On-Demand SUSE base pricing: 0.0124 USD per Hour

☒ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

## ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

ec2-lab

[Create new key pair](#)

## ▼ Network settings [Info](#)

### Subnet [Info](#)

subnet-0f4f3c2ce9c2d1f55

VPC: vpc-0ae7fd134c8d2696c Owner: 834362069350  
Availability Zone: ap-south-1a Zone type: Availability Zone  
IP addresses available: 4091 CIDR: 172.31.32.0/20

[Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

### Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Select existing security group

☐ Create security group

### Common security groups [Info](#)

Select security groups

ec2-sg sg-0b0e46ba953101b16 X

VPC: vpc-0ae7fd134c8d2696c

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

### ► Advanced network configuration

▼ **Storage (volumes)** [Info](#)

**EBS Volumes** [Hide details](#)

▼ **Volume 1 (AMI Root)**  
AMI Volumes are not included in the template unless modified

Storage type <a href="#">Info</a>	Device name - <i>required</i> <a href="#">Info</a>	Snapshot <a href="#">Info</a>
EBS	/dev/sda1	snap-032f6375d7735bd06
Size (GiB) <a href="#">Info</a>	Volume type <a href="#">Info</a>	IOPS <a href="#">Info</a>
<input type="text" value="8"/>	<input type="text" value="gp3"/>	<input type="text" value="2000"/>
Delete on termination <a href="#">Info</a>	Encrypted <a href="#">Info</a>	KMS key <a href="#">Info</a>
<input type="text" value="Yes"/>	<input type="text" value="Not encrypted"/>	<input type="text" value="Don't include in launch tem..."/>
<small>KMS keys are only applicable when encryption is set on this volume.</small>		
Throughput <a href="#">Info</a>		
<input type="text" value="125"/>		

*Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage*

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

- Now click on create launch template.
- Go to EC2 Dashboard, click on **Auto Scaling Groups** present under **Auto Scaling**.
- Create Auto Scaling Groups.
- Provide a name and select launch template.

**Choose launch template or configuration** [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

**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.

**Launch template** [Info](#) [Switch to launch configuration](#)

Launch template  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.  
   
[Create a launch template](#)

Version  
   
[Create a launch template version](#)

Description v1	Launch template <a href="#">EC2-Template</a> lt-0fb7499cc7c71be4d	Instance type t2.micro
AMI ID ami-0dee22c13ea7a9a67	Security groups -	Request Spot Instances No



On the next screen, select the Availability Zones.

## Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

### VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0ae7fd134c8d2696c

172.31.0.0/16 Default



[Create a VPC](#)

### 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



ap-south-1a | subnet-0f4f3c2ce9c2d1f55 X

172.31.32.0/20 Default

ap-south-1b | subnet-09bf6f985da526fa0 X

172.31.0.0/20 Default

ap-south-1c | subnet-0b41bfca92ee33bcd X

172.31.16.0/20 Default

[Create a subnet](#)

### Availability Zone distribution - new

Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

#### ☒ Balanced best effort

If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

#### ☐ Balanced only

If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

## Configure advanced options - optional [Info](#)

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

### 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.

#### ☒ No load balancer

Traffic to your Auto Scaling group will not be fronted by a load balancer.

#### ☐ Attach to an existing load balancer

Choose from your existing load balancers.

#### ☐ Attach to a new load balancer

Quickly create a basic load balancer to attach to your Auto Scaling group.

### VPC Lattice integration options [Info](#)

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

#### Select VPC Lattice service to attach

#### ☒ No VPC Lattice service

VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

#### ☐ Attach to VPC Lattice service

Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

[Create new VPC Lattice service](#)

## Configure group size and scaling - *optional* [Info](#)

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

### 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 by using automatic scaling.

#### Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance 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

Equal or less than desired capacity

##### Max desired capacity

5

Equal or greater than desired capacity

### Automatic scaling - *optional*

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.



#### No scaling policies

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.



#### Target tracking scaling policy

Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Next, Next, Next and create Auto Scaling Group.

Now you can check, 2 minimum instances must get created.

### Delete Auto Scaling Group.

Go to Auto Scaling, Select Auto Scaling Groups, select the auto scaling group, and delete this.

If you delete the Auto Scaling Group, then associated EC2 instances will get deleted.