

ELB & ASG

(Elastic Load Balancing & Auto Scaling Groups)

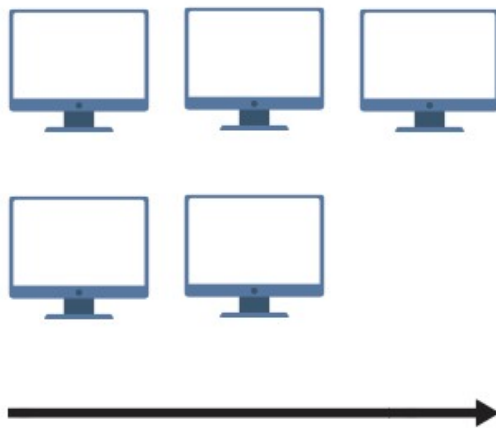
Scalability

- Scalability is basically the ability of a system or application to handle greater workloads by adapting.
- There are two types of scalability:
 1. **Vertical**
 - Increasing the size / capacity of resources.
 - For example, if your application runs on t2.micro you can change it to t2.large
 - Common for non-distributed systems. Like databases.
 - Often times, there is a limit to how much you can vertically scale which depends on the hardware.
 2. **Horizontal**
 - Increasing the number of resources.
 - Suitable for distributed systems.
 - Horizontal scaling is kept in mind while creating web apps or modern apps.
 - Super easy with AWS with on-demand provisioning and ASG.

Horizontal Scaling vs. Vertical Scaling

Horizontal Scaling

Add more instances



Vertical Scaling

Increase size of instances
(RAM, CPU, etc.)



High Availability

- High availability goes hand in hand with horizontal scaling
- Being highly available implies running your application in at least 2 AZs. This ensures survival in case of disaster in one of the AZs (power outage, hardware failure, natural disasters, etc.)

High Availability & Scalability For EC2

- Vertical Scaling: Increase instance size (= scale up / down)
 - From: t2.nano - 0.5G of RAM, 1 vCPU
 - To: u-12tb1.metal – 12.3 TB of RAM, 448 vCPUs
- Horizontal Scaling: Increase number of instances (= scale out / in)
 - Auto Scaling Group
 - Load Balancer
- High Availability: Run instances for the same application across multi AZ
 - Auto Scaling Group multi AZ
 - Load Balancer multi AZ

Scalability vs Elasticity (vs Agility)

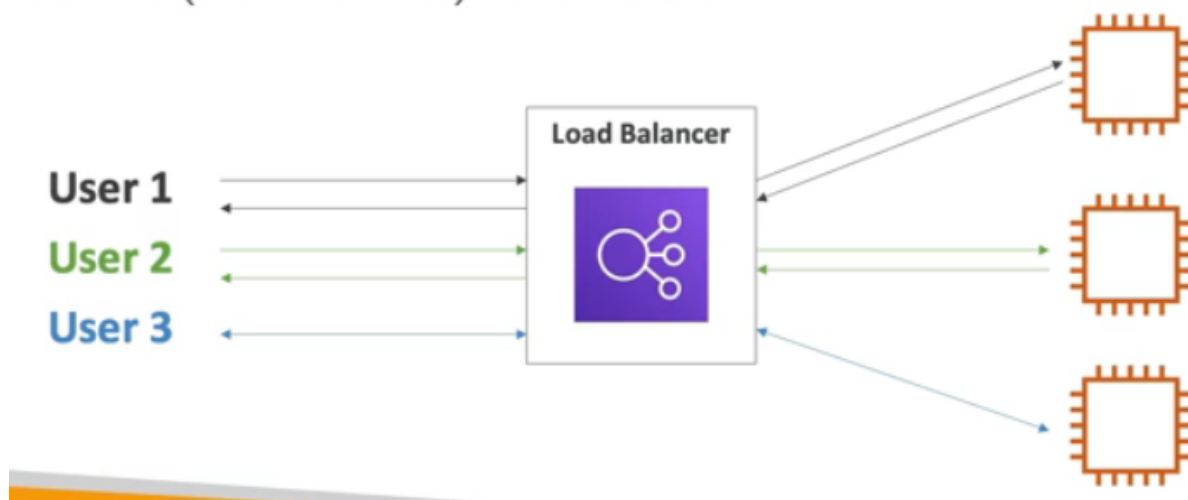
- Scalability: ability to accommodate a larger load by making the hardware stronger (scale up), or by adding nodes (scale out)
- Elasticity: once a system is scalable, elasticity means that there will be some “auto-scaling” so that the system can scale based on the load. This is “cloud-friendly”: pay-per-use, match demand, optimize costs
- Agility: (not related to scalability - distractor) new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes.

Load Balancing

- An Elastic Load Balancer (ELB) in AWS is like a smart traffic cop for your website. It helps distribute the incoming traffic to different servers to make


sure no single server is overwhelmed.

- Load Balancers are servers that forward internet traffic to multiple servers (EC2 Instances) downstream. This allows us to scale our backend better and does not overload one particular instance.
- Another advantage is a seamless experience at the user's end due to reduced latency or response time.
- ELB is a service managed by aws.
- How it works is:
 - We expose our load balancer publicly to our users.
 - When the users interact with our application, they interact with the load balancer at the backend instead of directly accessing the servers (ec2 instances)
 - The load balancer then redirects that user or traffic to different servers allowing better response time and optimization.




- Advantages of using a load balancer
 - Spread workloads across multiple downstream instances
 - Expose a single point of access (DNS - Domain Name Server) to your application

- Seamlessly handle failure of instances
- Regular health checks of instances - if an instance is down, the load balancer will not direct any traffic to that instance. This way we can hide our backend failures.
- Provide SSL (Secure Sockets Layer is standard technology for securing an internet connection by encrypting data sent between a website and a browser (or between two servers)) termination for websites (HTTPS)


You

how is "Provide SSL termination for websites (HTTPS)" an advantage of using ELB in AWS. Explain in simple terms as im only a beginner


ChatGPT





Sure, let's break it down in simple terms!

Imagine you have a website, and you want to make sure that when people visit it, their connection is secure. One way to do this is by using HTTPS, which encrypts the data between the user's browser and your website.

Now, an Elastic Load Balancer (ELB) in AWS is like a smart traffic cop for your website. It helps distribute the incoming traffic to different servers to make sure no single server is overwhelmed.

The advantage of using ELB with "SSL termination for websites" is that it can handle the encryption part for you. When a user tries to connect to your website, ELB can take care of the secure connection (HTTPS) so that your web servers don't have to worry about it. It's like having a helper that deals with the security handshake and encryption, allowing your servers to focus on serving the actual content of your website.

So, in simple terms, using ELB for "SSL termination" means offloading the task of securing connections to ELB, making it easier for your website to provide secure and encrypted connections without your servers having to do all the heavy lifting.

- Highly available across different zones.
- AWS ensures ELB works, and takes care of upgrades, maintenance, availability. Not as customizable as your own ELB (cheaper and more specific but comes with a lot of hassle)

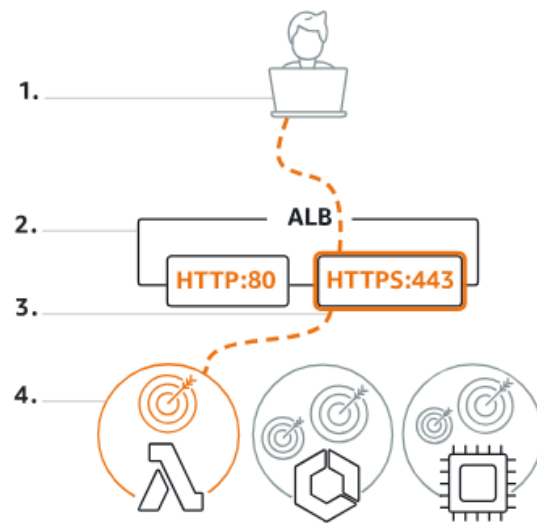
Types of ELB offered by AWS:

1. Application LB

- Layer 7
- User traffic: HTTP / HTTPS / gRPC protocols
- Super simple and helpful if you want a static DNS (URL)

▼ How Elastic Load Balancing works

1. Clients make requests to your application.
2. The listeners in your load balancer receive requests matching the protocol and port that you configure.
3. The receiving listener evaluates the incoming request against the rules you specify, and if applicable, routes the request to the appropriate target group. You can use an HTTPS listener to offload the work of TLS encryption and decryption to your load balancer.
4. Healthy targets in one or more target groups receive traffic based on the load balancing algorithm, and the routing rules you specify in the listener.



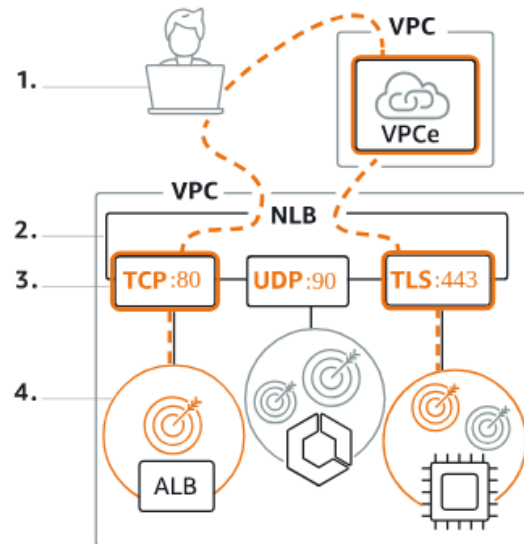
2. Network LB

- Layer 4
- **Ultra high performance:** millions of requests per second
- Gives you a static IP through Elastic IP

- User traffic: Allows for TCP and UDP

▼ How Elastic Load Balancing works

1. Clients make requests to your application.
2. The load balancer receives the request either directly or through an endpoint for private connectivity (via AWS PrivateLink).
3. The listeners in your load balancer receive requests of matching protocol and port, and route these requests based on the default action that you specify. You can use a TLS listener to offload the work of encryption and decryption to your load balancer.
4. Healthy targets in one or more target groups receive traffic according to the flow hash algorithm.



3. Gateway LB

- Layer 3
- GENEVE protocol on IP Packets
- Use case is to route traffic to firewalls that you manage on EC2 instances (for intrusion detection or deep packet introspection)

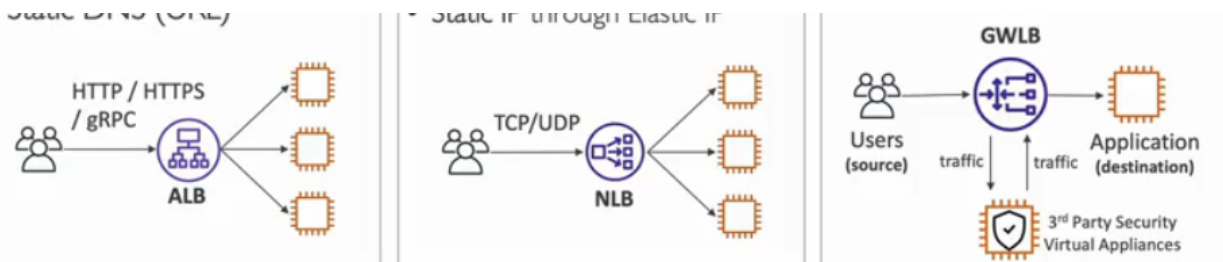
▼ How Gateway Load Balancers work

1. Clients make requests to your application.
2. The load balancer receives the request based on the route table configurations that are set within your VPC, Internet Gateway, or Transit Gateway.
3. The load balancer routes requests to a target group consisting of a scalable fleet of appliances (for example, firewalls, deep packet inspection systems, URL filtering systems etc.) to process traffic flows.
4. The virtual appliance processes the traffic, and forwards it back to the load balancer, or drops the traffic based on its configuration. This type of load balancer acts as a bump-in-the-wire between the source and destination.
5. The load balancer forwards the traffic to its destination.



4. Classic LB

- Layer 4 and 7
- Older generation LB
- Being retired in 2023
- It's usually replaced by the application LB or network LB
- Not visible in the console atp.
- Architecture:



Application LB Hands-on

- Launch 2 new EC2 instances without key pairs. Access the public IP using microsoft edge (or any other browser except chrome).
- With ALB, we basically want to access both these URLs using a single point and balance the load between them.
- Create an ALB

Basic configuration

Load balancer name

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


Demo ALB

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

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#) 

☐ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)

Select the type of IP addresses that your subnets use.

☒ IPv4

Recommended for internal load balancers.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

- Deploy it in all the AZs

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ **ap-south-1a (aps1-az1)**

Subnet

subnet-0c5e537c39c7e8a6b

IPv4 address

Assigned by AWS

☒ **ap-south-1b (aps1-az3)**

Subnet

subnet-0b376f9be7efaafc9

IPv4 address

Assigned by AWS

☒ **ap-south-1c (aps1-az2)**

Subnet

subnet-02c51bc76a4f0d2e9

IPv4 address

Assigned by AWS

- Create a new security group which allows only HTTP traffic to the load balancer

Basic details

Security group name [Info](#)

demo-sg-for-alb

Name cannot be edited after creation.

Description [Info](#)

Allow HTTP into ALB

VPC [Info](#)

vpc-0021041a806890855

Inbound rules [Info](#)

Inbound rule 1

Delete

Type

HTTP

Protocol

TCP

Port range

80

Source type

Anywhere-IPv4

Source

Q

Description - optional

0.0.0.0/0 X

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

demo-sg-for-alb

sg-0f5034866d2bfd3e8 VPC: vpc-0021041a806890855



- Create a target group. A target group is simply a group of instances you want the load balancer to direct user traffic to.

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

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.

☐ IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

☐ Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

☐ Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

demo-tg-for-alb

Protocol : Port

HTTP

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

☒ IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

☐ IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

-
vpc-0021041a806890855
IPv4: 172.31.0.0/16

Protocol version

☒ HTTP1

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

☐ HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

☐ gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter instances

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Private IP
<input checked="" type="checkbox"/>	i-0775567b4dd1eea93	My first instance	Running	launch-wizard-7	ap-south-1b	172.31.1
<input checked="" type="checkbox"/>	i-0647739eaad972f5b	My second instance	Running	launch-wizard-7	ap-south-1b	172.31.7

2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include as pending below

- Select this target group as listener while creating the ALB

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 define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Remove

Protocol

Port

Default action

[Info](#)

HTTP ▼

:

80

1-65535

Forward to

demo-tg-for-alb

Target type: Instance, IPv4

HTTP ▼

↻

[Create target group](#)

- Create the load balancer and wait for its state to turn from provisioning to active.
- Once it is active, you will get a DNS, copy and paste it in a new tab.
- If you keep refreshing the page, you will notice that the target is changing. This is because the ALB is directing traffic to different ec2 instances. Plus you can now access both the servers using the same URL / DNS.





- You can also **health check** your instances using the target group created.

Target group: demo-tg-for-alb

Details

Targets

Monitoring

Health checks

Attributes

Tags

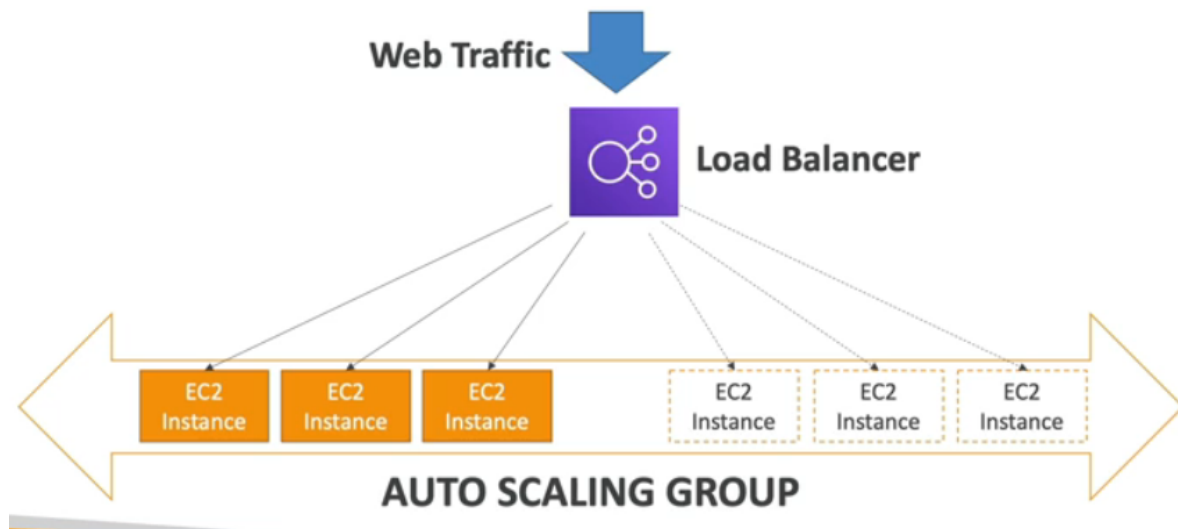
Registered targets (2)

Filter targets

<input type="checkbox"/>	Instanc... ▾	Name ▾	Port ▾	Zone ▾	Health ... ▾	Health st...	Launch time
<input type="checkbox"/>	i-064773...	My secon...	80	ap-south-1b	<div><div></div>Healthy</div>		November 16, 2023, 20:39 (UTC+05:30)
<input type="checkbox"/>	i-077556...	My first in...	80	ap-south-1b	<div><div></div>Healthy</div>		November 16, 2023, 20:39 (UTC+05:30)

Auto Scaling Groups

- Now that we have learned how to load balance using the ELB, we need to know how to automate the creation of these instances in case of traffic fluctuations (highs and lows) (horizontal scaling).
- ASG and ELB go hand in hand (duh).



- The goal of an Auto Scaling Group (ASG) is to:
 - Scale out (add EC2 instances) to match an increased load
 - Scale in (remove EC2 instances) to match a decreased load
 - Ensure we have a minimum and a maximum number of machines running
 - Automatically register new instances to a load balancer
 - Replace unhealthy instances
- Cost optimization max
- ASG in AWS require you to set a minimum size, a maximum size and the actual size is the desired capacity at any given point in time (it will scale in and out as needed).

ASG Hands-on

- Terminate the 2 instances created previously.
- Create an auto scaling group.
- Create a launch template (this template will be used to tell asg how to or what type of instances to create within it)

Launch template contents

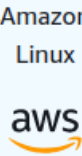
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

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


An AMI 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

 Search our full catalog including 1000s of application and OS images


Quick Start




Amazon Linux




macOS




Ubuntu



Windows



Red Hat



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community


Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-02a2af70a66af6dfb (64-bit (x86)) / ami-07d67603110c131fb (64-bit (Arm))

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

Free tier eligible



Description

Amazon Linux 2023 AMI 2023.2.20231113.0 x86_64 HVM kernel-6.1

- Skip key pair and subnet. Choose from existing security groups.

▼ 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

Don't include in launch template ▼



[Create new key pair](#)

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template ▼



[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

Security groups [Info](#)

Select security groups ▼

launch-wizard-7 sg-08b3080f32b0d6e27 ✕
VPC: vpc-0021041a806890855



[Compare security group rules](#)

► Advanced network configuration

- Paste this user data script within advanced details

```
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
yum update -y
yum install -y httpd
```

```
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello World from $(hostname -f)</h1>" > /var/www/html.
```

- Select multiple AZs (for high availability) within the region wherever you want your instances to be.

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-0021041a806...

<input checked="" type="checkbox"/>	ap-south-1a subnet-0c5e537c39c7e8a6b 172.31.32.0/20 Default
<input checked="" type="checkbox"/>	ap-south-1b subnet-0b376f9be7efaafc9 172.31.0.0/20 Default
<input checked="" type="checkbox"/>	ap-south-1c subnet-02c51bc76a4f0d2e9 172.31.16.0/20 Default

Select Availability ...

ap-south-1c | su

- Attach an existing LB

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.

- This will tell the ASG to register all the instances created to the demo target group, so then the load balancer will be able to direct the traffic to those instances along w the ones previously created.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

- ☒ **Choose from your load balancer target groups**
This option allows you to attach Application, Network, or Gateway Load Balancers.

- ☐ **Choose from Classic Load Balancers**

Existing load balancer target groups

Only Instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▼



demo-tg-for-alb | HTTP ✕
Application Load Balancer: Demo-ALB

- Now we will configure scaling setting of the ASG.

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.

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

Equal or less than
desired capacity

Max desired capacity

Equal or greater than
desired capacity

- Review details and create the ASG.
- Since we set the desired capacity to 2, the asg has created 2 new running ec2 instances and they are fully managed by the asg.

DemoASG

< Details Activity Automatic scaling **Instance management** Monitoring >

Instances (2)



Actions ▼

🔍 Filter instances

< 1 > ⚙️

<input type="checkbox"/>	Instance ID ▲	Lifecycle ▼	Instanc... ▼	Weight... ▼	Lau
<input type="checkbox"/>	i-031314b259a145dc8	InService	t2.micro	-	Den
<input type="checkbox"/>	i-053af1248060577e2	InService	t2.micro	-	Den

- If you check in the target group (the one used to create the asg), you will find that we now have 2 targets.

Details

 arn:aws:elasticloadbalancing:ap-south-1:248640396187:targetgroup/demo-tg-for-alb/edecb5c37e0692b3

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1
IP address type IPv4	Load balancer Demo-ALB	

Total targets 2	Healthy ✔ 2	Unhealthy ✖ 0	Unused ⊖ 0
--------------------	----------------	------------------	---------------

► Distribution of targets by Availability Zone (AZ)

 Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets
Monitoring
Health checks
Attributes
Tags

Registered targets (2)

Filter targets

<input type="checkbox"/>	Instanc... ▾	Name ▾	Port ▾	Zone ▾	Health ... ▾	Health st...	Launch time
<input type="checkbox"/>	i-053af12...		80	ap-south-1a	✔ Healthy		November 16, 2023, 22:01 (UTC+05:30)
<input type="checkbox"/>	i-031314...		80	ap-south-1b	✔ Healthy		November 16, 2023, 22:01 (UTC+05:30)

- Access your website using the DNS with the load balancer and you will be redirected to two different IP addresses. Same as before, except, this whole thing was automated.
- FUN - terminate any one instance or even both and then watch the ASG creating new ones automatically. nice.
- Check the activity history of the ASG to monitor these changes.

Activity history (5)

Status ▾	Description ▾	Cause ▾	Start time ▾	End time
✔ Successful	Launching a new EC2 instance: i-0306abffa5e2c534e	At 2023-11-16T16:41:15Z an instance was launched in response to an unhealthy instance needing to be replaced.	2023 November 16, 10:11:17 PM +05:30	2023 November 16, 10:11:49 PM +05:30
⌛ Connection draining in progress	Terminating EC2 instance: i-053af1248060577e2 - Waiting For ELB Connection Draining.	At 2023-11-16T16:41:15Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2023 November 16, 10:11:15 PM +05:30	
✔ Successful	Launching a new EC2 instance: i-053af1248060577e2	At 2023-11-16T16:31:35Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2023 November 16, 10:01:37 PM +05:30	2023 November 16, 10:02:09 PM +05:30
✔ Successful	Launching a new EC2 instance: i-031314b259a145dc8	At 2023-11-16T16:31:11Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2023-11-16T16:31:25Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2023 November 16, 10:01:27 PM +05:30	2023 November 16, 10:01:59 PM +05:30

- Section Clean up:
 - Terminating the instances created by the ASG using the usual EC2 console won't work as the ASG will keep on creating new instances. So, delete the ASG altogether.
 - Also delete the load balancer.
 - You don't need to delete the target group as they cost nothing and the targets are anyway set to zero bec you deleted the asg.

ASG Scaling Strategies

** imp for exam

- **Manual Scaling:** Update the size of an ASG manually
- **Dynamic Scaling:** Respond to changing demand
 - **Simple / Step Scaling**
 - When a CloudWatch alarm is triggered (example CPU > 70%), then add 2 units
 - When a CloudWatch alarm is triggered (example CPU < 30%), then remove 1
 - **Target Tracking Scaling**
 - Example: I want the average ASG CPU to stay at around 40%
 - **Scheduled Scaling**
 - Anticipate a scaling based on known usage patterns
 - Example: increase the min. capacity to 10 at 5 pm on Fridays

- **Predictive Scaling**
 - Uses Machine Learning to predict future traffic ahead of time
 - Automatically provisions the right number of EC2 instances in advance

- Useful when your load has predictable time-based patterns



Summary

- High Availability vs Scalability (vertical and horizontal) vs Elasticity vs Agility in the Cloud
- Elastic Load Balancers (ELB)
 - Distribute traffic across backend EC2 instances, can be Multi-AZ
 - Supports health checks
 - 4 types: Classic (old), Application (HTTP – L7), Network (TCP – L4), Gateway (L3)
- Auto Scaling Groups (ASG)
 - Implement Elasticity for your application, across multiple AZ
 - Scale EC2 instances based on the demand on your system, replace unhealthy
 - Integrated with the ELB

**Good job!**

Vertical scaling means increasing the size of the instance. Changing from a t3a.medium to a t3a.2xlarge is an example of size increase.

Question 3:

Changing an EC2 Instance Type from a t3a.medium to a t3a.2xlarge is an example of?

☐ Horizontal scaling

☐ High Availability

☐ Agility

☒ Vertical scaling



Good job!

Auto Scaling Groups can add or remove instances, but from the same type. They cannot change the EC2 Instance Types on the fly.

Question 5:

Which of the following statements is INCORRECT regarding Auto Scaling Groups?

- ☐ Replace unhealthy instances
- ☐ Are cost-effective by running at optimal capacity
- ☐ Automatically register new instances to a load balancer
- ☒ Automatically changing the EC2 Instance Types



Good job!

Load Balancers cannot help with back-end autoscaling. You should use Auto Scaling Groups.

This was discussed in Lecture 64: [Elastic Load Balancing \(ELB\) Overview](#) >

Question 9:

Which of the following statements is NOT a feature of Load Balancers?

- ☐ Do regular health checks to your instances
- ☐ Spread load across multiple downstream instances
- ☐ Handle failures of downstream instances
- ☒ Back-end autoscaling