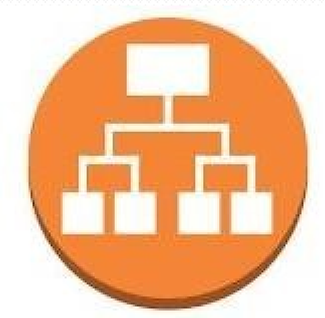
22. Elastic Load Balancing

AWS ELB

Elastic Load Balancing (ELB)

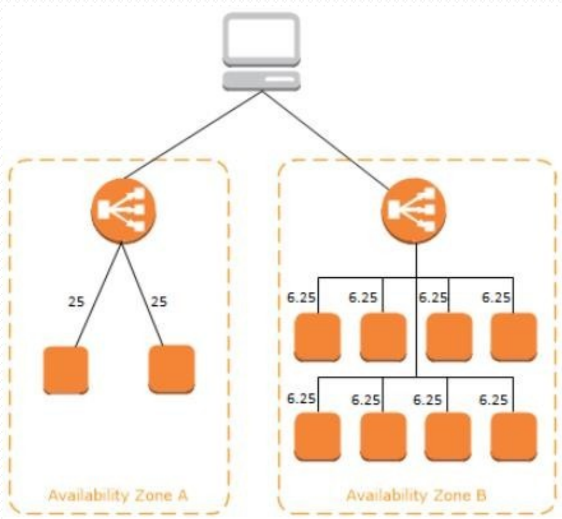


ELB is a traffic load balancing service, which distributes incoming traffic automatically across target instances, IP addresses and containers as per configuration. These targets can be in one Availability Zone or multiples Availability Zones. According to AWS, ELB is a mechanism to distribute traffic among EC2 Instances participating in load balancing.

ELBs are also used to improve capacity of a running application to handle incoming load. ELB supports to various protocols for load balancing such as Hyper Text Transfer Protocol (HTTP), HTTP Secure HTTPS, Secure Socket Layer (SSL), and Transmission Control Protocol (TCP) traffic. ELB provides a fix DNS like entry point to access ELB, which can be mapped in DNS table as CNAME entry.

How ELB Works?

When clients from the Internet do access ELB configured applications, a load balancer receives the traffic and routes requests of clients to ELB registered Instances (virtual machines), which can reside in one or multiple Availability Zones.

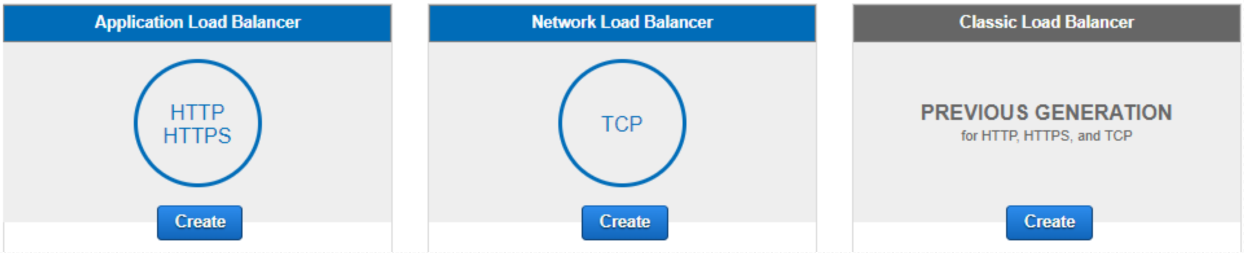


ELB also checks the health of Instances running an applications and those Instances are registered with ELB. In case, ELB finds an unhealthy virtual which is a part of ELB, it stops sending traffic to that unhealthy instance. It would only resume distribution of traffic once ELB detects it as healthy again.

Load Balancer Types

On the basis of configuration, Elastic Load balancer does support three kind of load balancers.

1. Application Load Balancer
2. Network Load Balancer (its new)
3. Classic Load Balancer



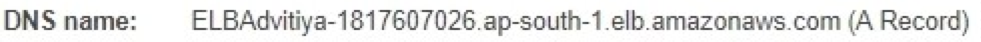
Application Load Balancer

It serves as a single point of contact for your clients, similar to ***DNS of a web site***. Application load balancer operates at 7th layer of OSI. Application Load Balancer uses the SSL/Transport Layer Protocol (TLS) protocol to establish encrypted connections between ELB and client that initiate HTTPS connections. It also provide load balancing for internal servers running in backend. ELB supports and provides security policies which has SSL negotiations, which will be used later when connection between clients and the load balancer will establish.

For HTTPS, SSL protocol will be used and for this you must install an SSL certificate on the load balancer.

Network Load Balancer (NLB)

It operates at 4th layer of OSI in TCP/IP network model. NLB also serves as a single point of contact for your client. For example, DNS name of ELB



DNS name: ELBAdvitiya-1817607026.ap-south-1.elb.amazonaws.com (A Record)

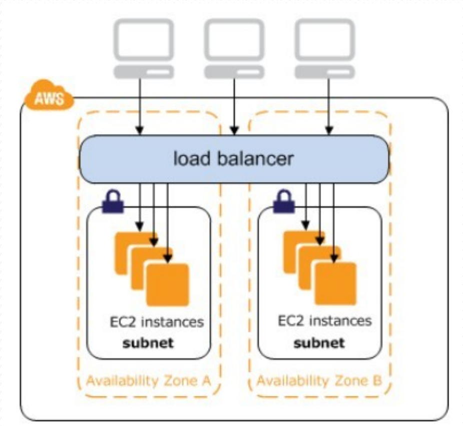
To reach load balancer either you can use ELB DNS name or ELB DNS record can be added in your DNS server as CNAME or A record.

Choose NLB only when you require high performance with low latency and static IP addresses for your application.

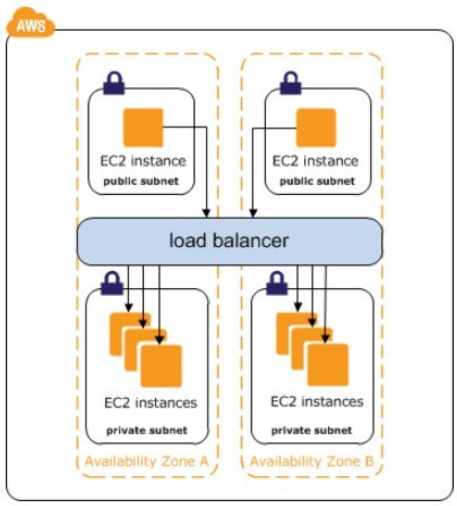
Types of load balancers can also be categorized on the basis of ELB usage, internal or the Internet.

Internet Facing ELB

When clients use the Internet to access a web application which is configured using a load balancer, and that ELB distributes the traffic across multiple Instances which are registered with the load balancer. In brief, an internet facing ELB has public DNS name and requests to access an application come from the Internet. Therefore, Internet facing ELB can route requests from clients over the Internet.



ELB scales in and out to meet traffic demand, AWS does not recommend an application to bind with an IP address. ELB supports only IPv4, only EC2 classic does support both versions of IP address, IPv4 and IPv6.



Internal Load Balancer

If you are using multi-tier application in AWS environment, it is recommended to use load balancer between tiers.

Listeners

To access web applications through AWS services, every client over the Internet require a URL or some kind of configuration which should base on a service name and port number. Therefore a CNAME record is configured to the A record name of the ELB endpoint. Every listener configured with a protocol and a port number.

In Brief a listener checks for connection requests using its configured protocol and port. And ELB uses the listener rules to route requests to targets. You can add, modify or delete the rules.

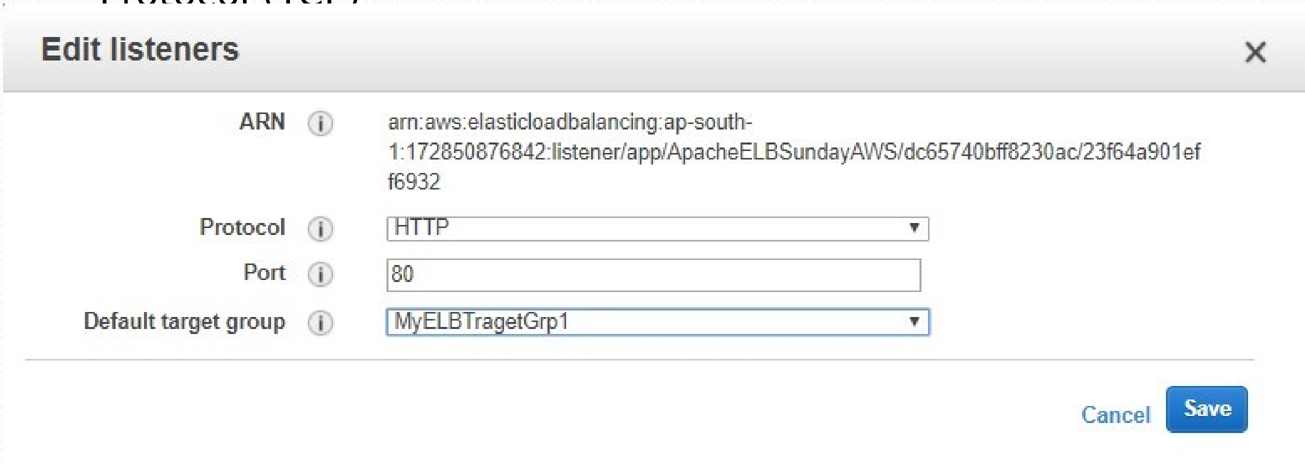
AWS Elastic Load balancing does support to the following protocols:

Hyper Text Transfer Protocol (HTTP)

Hyper Test Transfer Protocol Secure (HTTPS)

Secure Socket Layer (SSL)

Transmission Control Protocol (TCP)



Edit listeners

ARN: arn:aws:elasticloadbalancing:ap-south-1:172850876842:listener/app/ApacheELBSundayAWS/dc65740bff8230ac/23f64a901eff6932

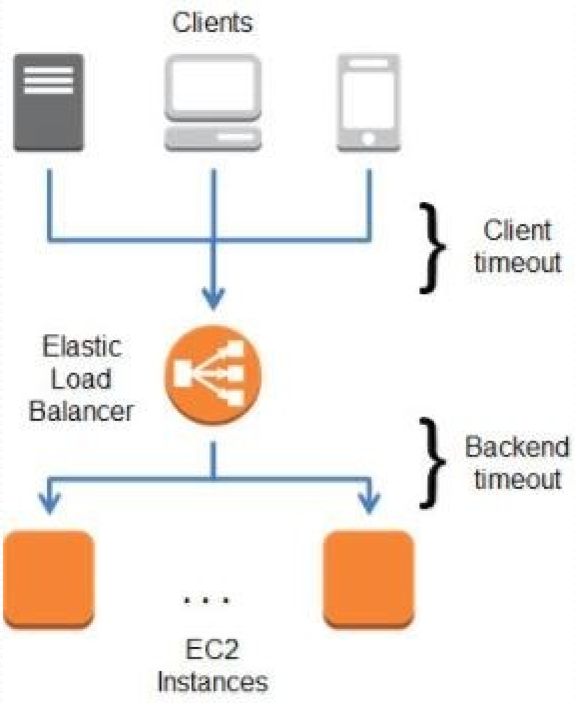
Protocol: HTTP

Port: 80

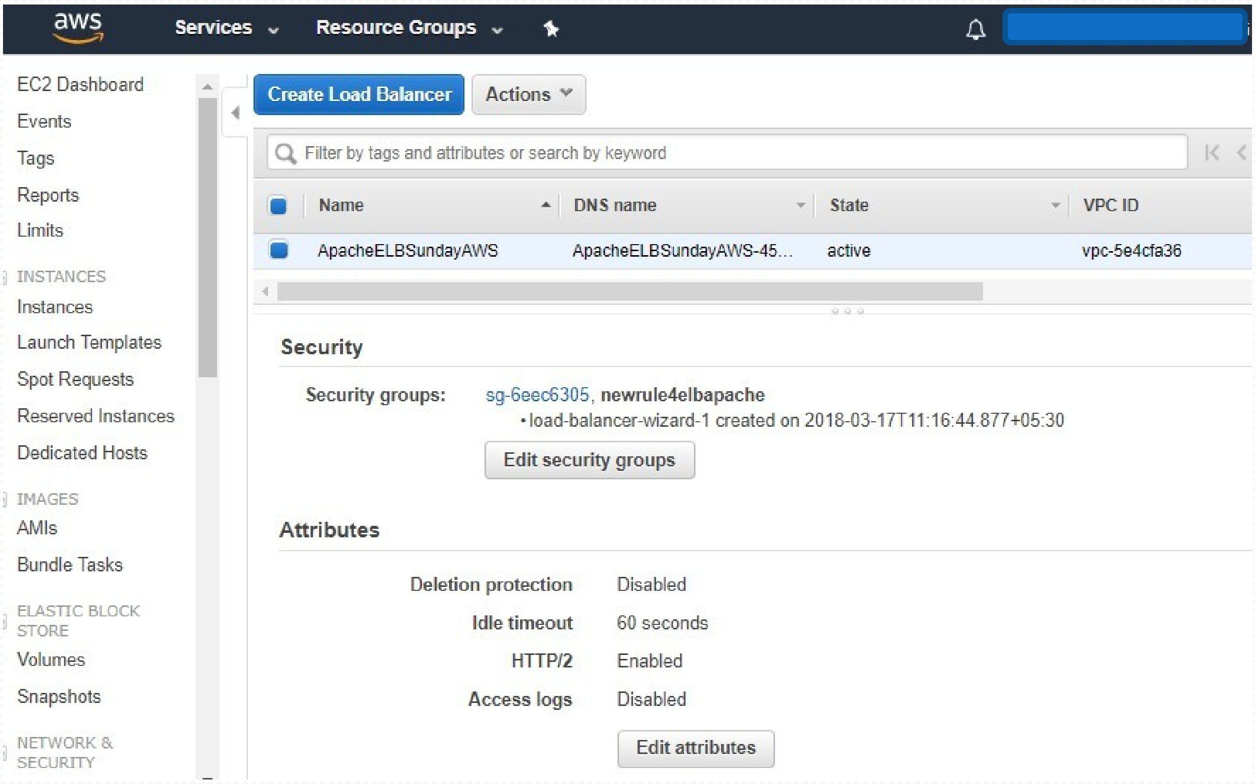
Default target group: MyELBTragetGrp1

Idle connection timeout

Any request which makes through your web browser or mobile device to the ELB, the established connection is used for the request and response. The connection remains open for the possible communication between client and the ELB. By default, this time in ELB is set for 60 seconds. This time period is known as the **idle connection timeout**.



If request does not complete in the specified idle connection timeout period, the ELB closes the connection, even if the data is being copied or transferred. So you can change the setting for the connection timeout to make sure that the running task must be completed. You can find idle timeout in **Attributes** settings under **Description** tab of load balancer.



Name	DNS name	State	VPC ID
ApacheELBSundayAWS	ApacheELBSundayAWS-45…	active	vpc-5e4cfa36

Security

Security groups: sg-6eec6305, newrule4elbapache
• load-balancer-wizard-1 created on 2018-03-17T11:16:44.877+05:30

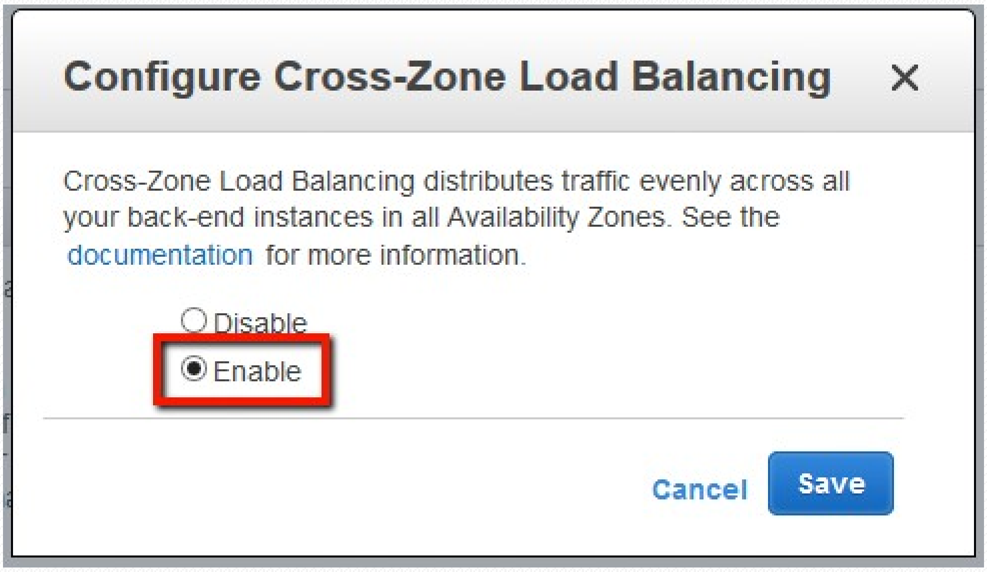
Attributes

Deletion protection	Disabled
Idle timeout	60 seconds
HTTP/2	Enabled
Access logs	Disabled

If you configure the ELB listener for HTTP/HTTPS services, you must enable keep-alive option for Instances. And in order to avoid configuration conflict between keep-alive time and idle timeout, ensure the keep-alive time must be greater than idle connection timeout for the ELB.

Cross Zone Load Balancing

This is the feature of load balancing through which incoming traffic is equally distributed among the Instances participating in load balancing regardless of the Availability Zones in which these Instances are located.



Configure Cross-Zone Load Balancing ×

Cross-Zone Load Balancing distributes traffic evenly across all your back-end instances in all Availability Zones. See the documentation for more information.

○ Disable

● Enable

Connection Draining

When the connection draining option is enable, the ELB ensures that requests to unhealthy or deregistered Instances are stopped, while keeping the existing communication open. By default connection draining time is 300 seconds, and can be adjusted between 1 and 3,600 seconds.

Health Checks

ELB checks the health of Instances taking part in load balancing. If the status of the Instance is healthy, at the time of health check ELB will report as ***InService*** or if it is unhealthy, the ELB will report as ***OutOfService.***

Follow the Health Check configuration setting in below table:

Field	Description
Ping Protocol	The protocol to use to connect with the instance. Valid values: TCP, HTTP, HTTPS, and SSL Console default: HTTP CLI/API default: TCP
Ping Port	The port to use to connect with the instance, as

	a protocol:port pair. If the load balancer fails to connect with the instance at the specified port within the configured response timeout period, the instance is considered unhealthy. Ping protocols: TCP, HTTP, HTTPS, and SSL Ping port range: 1 to 65535 Console default: HTTP:80 CLI/API default: TCP:80
Ping Path	The destination for the HTTP or HTTPS request. An HTTP or HTTPS GET request is issued to the instance on the ping port and the ping path. If the load balancer receives any response other than "200 OK" within the response timeout period, the instance is considered unhealthy. If the response includes a body, your application must either set the Content-Length header to a value greater than or equal to zero, or specify Transfer-Encoding with a value set to 'chunked'. Default: /index.html
Response Timeout	The amount of time to wait when receiving a response from the health check, in seconds. Valid values: 2 to 60 Default: 5
HealthCheck Interval	The amount of time between health checks of an individual instance, in seconds. Valid values: 5 to 300 Default: 30
Unhealthy Threshold	The number of consecutive failed health checks that must occur before declaring an EC2 instance unhealthy. Valid values: 2 to 10 Default: 2
Healthy Threshold	The number of consecutive successful health checks that must occur before declaring an EC2 instance healthy. Valid values: 2 to 10 Default: 10

Sticky Sessions

By default, a load balancer routes each request independently to the registered instance with the smallest load. However, you can use the sticky session feature (also known as session affinity), which enables the load balancer to bind a user's session to a specific instance. This ensures that all requests from the user during the session are sent to the same instance.

Proxy Protocol

In AWS, proxy protocol is an Internet protocol. According to Proxy Protocol it

carries connection information from the source to the destination for which

the connection was requested.

The Proxy Protocol header helps you identify the IP address of a client when you have a load balancer that uses TCP for back-end connections. Because load balancers intercept traffic between clients and your instances, the access logs from your instance contain the IP address of the load balancer instead of the originating client

Reference:

https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-proxy-protocol.html

Summary

Important about ELB

1. AWS ELB main two tasks

a. ELB distributes incoming traffic to number of Instances, registered to ELB
b. ELB continuously does check health of Instances, if any Instance found unhealthy, it stops sending traffic to that unhealthy Instance.

2. ELB can be public or private.
3. For better performance and high availability, you are recommended to choose more than one subnet in different Availability Zones to launch the ELB.
4. Distribution of traffic in ELB among multiple Availability Zones happens in round-robin fashion.
5. To access ELB over the Internet, ELB DNS name is used.
6. ELB scales up or down the load balancer as the incoming traffic to ELB endpoint varies over time.
7. ELB can be integrated with AWS Auto Scaling service to manage traffic load at backend to meet requirement.
8. ELB functions with VPC, as a result of this ELB incorporates advanced features of security and networking.
9. ELB can be Internet facing or Internal. Internal load balancer routes traffic using private IPs within the VPC.
10. You can enhance the feature of application availability by using Rout 53 health check and DNS failover features.