ytoY technologies®
Engineer yocto to Yotta

Elastic Load Balancer

Mohanraj Shanmugam

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Elastic Load Balancing

- Elastic Load Balancing automatically distributes incoming traffic across multiple EC2 instances.
- You create a load balancer and register instances with the load balancer in one or more Availability Zones.
- The load balancer serves as a single point of contact for clients. This enables you to increase the availability of your application.
- You can add and remove EC2 instances from your load balancer as your needs change, without disrupting the overall flow of information.
- If an EC2 instance fails, Elastic Load Balancing automatically reroutes the traffic to the remaining running EC2 instances. If a failed EC2 instance is restored, Elastic Load Balancing restores the traffic to that instance.
- Elastic Load Balancing can also serve as the first line of defense against attacks on your network. You can offload the work of encryption and decryption to your load balancer so that your EC2 instances can focus on their main work.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Features of Elastic Load Balancing

- You can use the operating systems and instance types supported by Amazon EC2. You can configure your EC2 instances to accept traffic only from your load balancer.
- You can configure the load balancer to accept traffic using the following protocols: HTTP, HTTPS (secure HTTP), TCP, and SSL (secure TCP).
- You can configure your load balancer to distribute requests to EC2 instances in multiple Availability Zones, minimizing the risk of overloading one single instance. If an entire Availability Zone goes offline, the load balancer routes traffic to instances in other Availability Zones.
- There is no limit on the number of connections that your load balancer can attempt to make with your EC2 instances. The number of connections scales with the number of concurrent requests that the load balancer receives.
- You can configure the health checks that Elastic Load Balancing uses to monitor the health of the EC2 instances registered with the load balancer so that it can send requests only to the healthy instances.

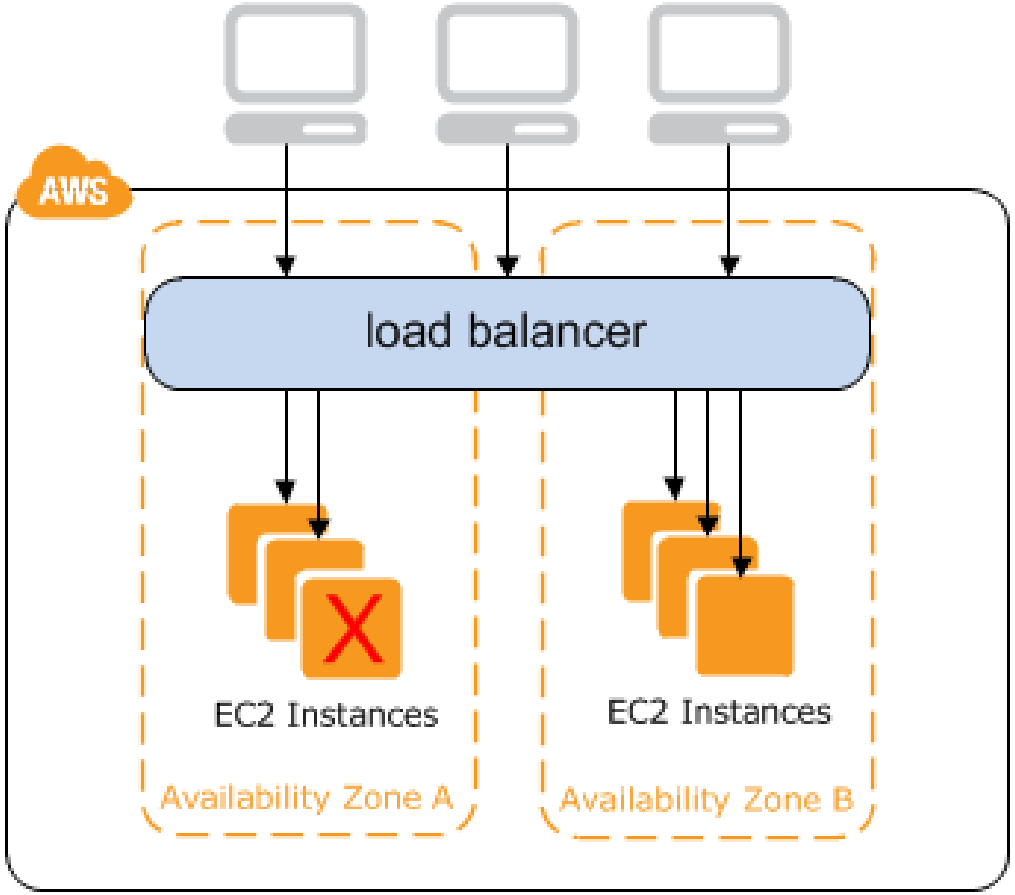
For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Features of Elastic Load Balancing

- You can use end-to-end traffic encryption on those networks that use secure (HTTPS/SSL) connections.
- [EC2-VPC] You can create an *Internet-facing* load balancer, which takes requests from clients over the Internet and routes them to your EC2 instances, or an *internal-facing*load balancer, which takes requests from clients in your VPC and routes them to EC2 instances in your private subnets. Load balancers
- You can monitor your load balancer using CloudWatch metrics, access logs, and AWS CloudTrail.
- You can associate your Internet-facing load balancer with your domain name. Because the load balancer receives all requests from clients, you don't need to create and manage public domain names for the EC2 instances to which the load balancer routes traffic.
- You can point the instance's domain records at the load balancer instead and scale as needed (either adding or removing capacity) without having to update the records with each scaling activity.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Elastic Load Balancing



- A load balancer accepts incoming traffic from clients and routes requests to its registered EC2 instances in one or more Availability Zones
- The load balancer also monitors the health of its registered instances and ensures that it routes traffic only to healthy instances.
- When the load balancer detects an unhealthy instance, it stops routing traffic to that instance, and then resumes routing traffic to that instance when it detects that the instance is healthy again.
- You configure your load balancer to accept incoming traffic by specifying one or more *listeners*.
- A listener is a process that checks for connection requests. It is configured with a protocol and port number for connections from clients to the load balancer and a protocol and port number for connections from the load balancer to the instances.
- Recommend that you configure your load balancer across multiple Availability Zones. If one Availability Zone becomes unavailable or has no healthy instances, the load balancer can route traffic to the healthy registered instances in another Availability Zone.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Request Routing

- Before a client sends a request to your load balancer, it resolves the load balancer's domain name using a Domain Name System (DNS) server.
- The DNS entry is controlled by Amazon, because your instances are in the `amazonaws.com` domain.
- The Amazon DNS servers return one or more IP addresses to the client, These are the IP addresses of the load balancer nodes for your load balancer.
- Note that the DNS entry also specifies the time-to-live (TTL) as 60 seconds, which ensures that the IP addresses can be remapped quickly in response to changing traffic.
- The client uses DNS round robin to determine which IP address to use to send the request to the load balancer.
- The load balancer node that receives the request uses a routing algorithm to select a healthy instance.
- It uses the round robin routing algorithm for TCP listeners, and the least outstanding requests routing algorithm (favors the instances with the fewest outstanding requests) for HTTP and HTTPS listeners.

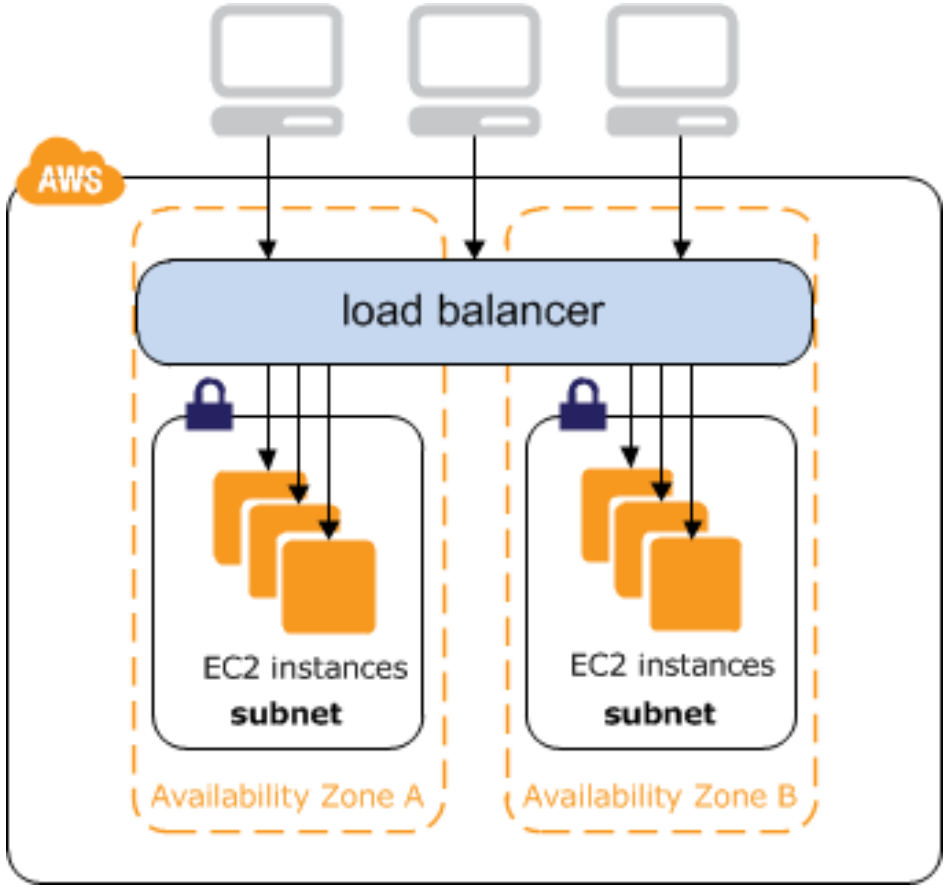
For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

cross-zone load balancing

- The *cross-zone load balancing* setting also determines how the load balancer selects an instance.
- If cross-zone load balancing is disabled, the load balancer node selects the instance from the same Availability Zone that it is in.
- If cross-zone load balancing is enabled, the load balancer node selects the instance regardless of Availability Zone. The load balancer node routes the client request to the selected instance.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Internet-Facing Load Balancers



- An Internet-facing load balancer takes requests from clients over the Internet and distributes them across the EC2 instances that are registered with the load balancer.
- When your load balancer is created, it receives a public DNS name that clients can use to send requests.
- The DNS servers resolve the DNS name of your load balancer to the public IP addresses of the load balancer nodes for your load balancer.
- Each load balancer node is connected to the back-end instances.
- Load balancers in a VPC support IPv4 addresses only. The console displays a public DNS name with the following form:

```
name-1234567890.region.elb.amazonaws.com
```

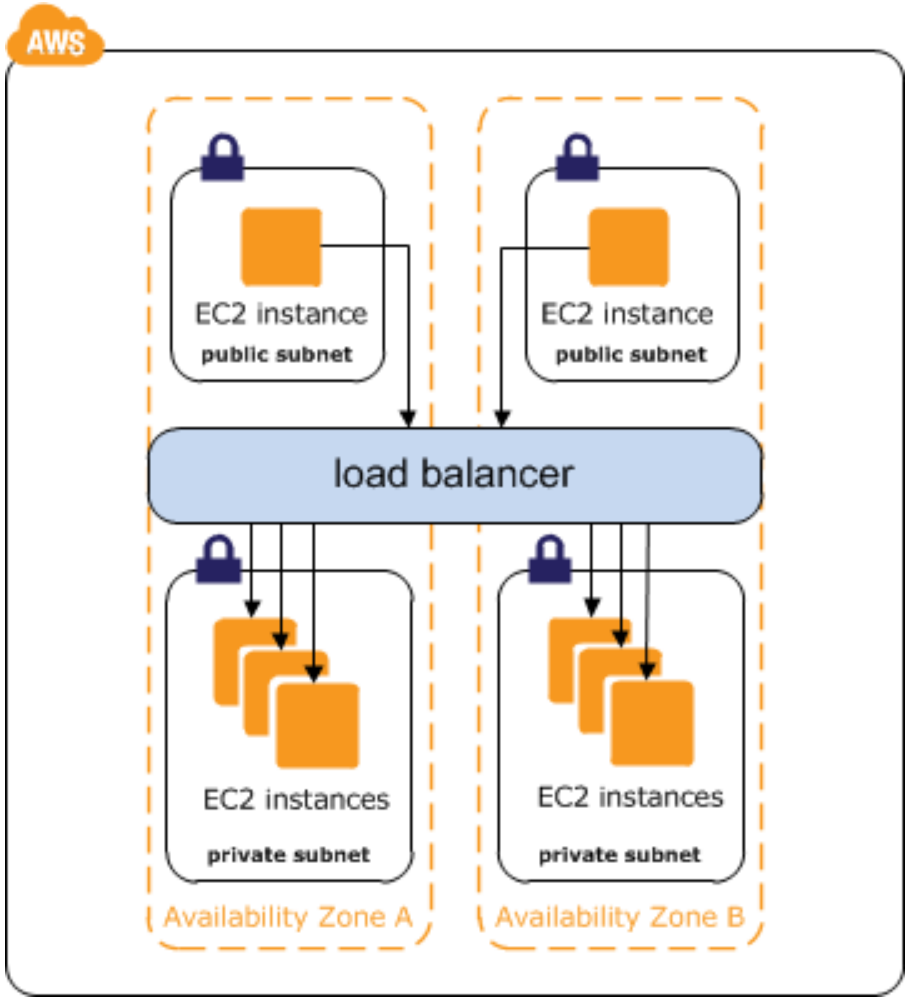
For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Create ELB

- When you create your load balancer, you configure:
  - listeners,
  - health checks,
  - register back-end instances.
- You configure a listener by specifying a protocol and a port for front-end (client to load balancer) connections, You can configure multiple listeners for your load balancer.
- protocol and a port for back-end (load balancer to back-end instances) connections.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Internal Load Balancers



- When you create a load balancer in a VPC, as internal load balancer
- An internal load balancer routes traffic to your EC2 instances in private subnets. The clients must have access to the private subnets.
- If your application has multiple tiers, for example web servers that must be connected to the Internet and database servers that are only connected to the web servers, you can design an architecture that uses both internal and Internet-facing load balancers.
- When an internal load balancer is created, it receives a public DNS name with the following form:

```
internal-name-123456789.region.elb.amazonaws.com
```

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

HTTPS Load Balancers

- You can create a load balancer that uses the SSL/TLS protocol for encrypted connections (also known as *SSL offload*).
- This feature enables traffic encryption between your load balancer and the clients that initiate HTTPS sessions, and for connections between your load balancer and your back-end instances.
- Elastic Load Balancing provides Secure Sockets Layer (SSL) negotiation configurations, known as *security policy*, to negotiate connections between the clients and the load balancer.
- you can use either a predefined security policy or a custom security policy.
- You must install an SSL certificate on your load balancer. The load balancer uses this certificate to terminate the connection and then decrypt requests from clients before sending them to the back-end instances.
- You can optionally choose to enable authentication on your back-end instances.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Back-end Instances

- After you've created your load balancer, you must register your EC2 instances with the load balancer.
- You can select EC2 instances from a single Availability Zone or multiple Availability Zones within the same region as the load balancer.
- Elastic Load Balancing routinely performs health checks on registered EC2 instances, and automatically distributes incoming requests to the DNS name of your load balancer across the registered, healthy EC2 instances.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Best Practices for Your Back-end Instances

- Install a web server, such as Apache or Internet Information Services (IIS), on all instances that you plan to register with your load balancer.
- For HTTP and HTTPS listeners, we recommend that you enable the keep-alive option in your EC2 instances, which enables the load balancer to re-use the connections to your instances for multiple client requests. This reduces the load on your web server and improves the throughput of the load balancer. The keep-alive timeout should be at least 60 seconds to ensure that the load balancer is responsible for closing the connection to your instance.
- Elastic Load Balancing supports Path Maximum Transmission Unit (MTU) Discovery. To ensure that Path MTU Discovery can function correctly, you must ensure that the security group for your instance allows ICMP fragmentation required
- The security group for your back-end instances should have inbound rules for the listener port and the health check port for your load balancer, with the source as the security group for your load balancer.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Configure Health Checks

- To discover the availability of your EC2 instances, the load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances. These tests are called *health checks*.
- The status of the instances that are healthy at the time of the health check is `InService`. The status of any instances that are unhealthy at the time of the health check is `OutOfService`.
- The load balancer performs health checks on all registered instances, whether the instance is in a healthy state or an unhealthy state.
- The load balancer routes requests only to the healthy instances. When the load balancer determines that an instance is unhealthy, it stops routing requests to that instance. The load balancer resumes routing requests to the instance when it has been restored to a healthy state.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Configure Health Checks

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

- The load balancer checks the health of the registered instances using either the default health check configuration provided by Elastic Load Balancing or a health check configuration that you configure.
- The load balancer sends a request to each registered instance at the ping port and ping path every `Interval` seconds. It waits for the instance to respond within the response timeout period. If the health checks exceed the threshold for consecutive failed responses, the load balancer takes the instance out of service. When the health checks exceed the threshold

stered trademarks of their respective owners.

ytoY technologies®
Engineer yocto to Yotta

Security Groups for Your Load Balancer

- **Internet-facing Load Balancer: Recommended Rules**

Inbound			
Source	**Protocol**	**Port Range**	**Comment**
0.0.0.0/0	TCP	*listener*	Allow all inbound traffic on the load balancer listener port
Outbound			
Destination	**Protocol**	**Port Range**	**Comment**
instance security group	TCP	*instance listener*	Allow outbound traffic to back-end instances on the instance listener port
instance security group	TCP	*health check*	Allow outbound traffic to back-end instances on the health check port

For perso

Security Groups for Your Load Balancer

- **Internal Load Balancer: Recommended Rules**

Inbound			
Source	**Protocol**	**Port Range**	**Comment**
VPC CIDR	TCP	*listener*	Allow inbound traffic from the VPC CIDR on the load balancer listener port
Outbound			
Destination	**Protocol**	**Port Range**	**Comment**
instance security group	TCP	*instance listener*	Allow outbound traffic to back-end instances on the instance listener port
instance security group	TCP	*health check*	Allow outbound traffic to back-end instances on the health check

Fo ive owners.

Network ACLs for Load Balancers in a VPC

- **Internet-Facing Load Balancer: Recommended Rules**

Inbound			
Source	**Protocol**	**Port**	**Comment**
0.0.0.0/0	TCP	*listener*	Allow all inbound traffic on the load balancer listener port
VPC CIDR	TCP	1024-65535	Allow inbound traffic from the VPC CIDR on the ephemeral ports
Outbound			
Destination	**Protocol**	**Port**	**Comment**
0.0.0.0/0	TCP	*instance listener*	Allow all outbound traffic on the instance listener port
0.0.0.0/0	TCP	*health check*	Allow all outbound traffic on the health check port
0.0.0.0/0	TCP	1024-65535	Allow all outbound traffic on the ephemeral ports

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Network ACLs for Load Balancers in a VPC

- **Internal Load Balancer: Recommended Rules**

Inbound			
Source	**Protocol**	**Port**	**Comment**
VPC CIDR	TCP	*listener*	Allow inbound traffic from the VPC CIDR on the load balancer listener port
VPC CIDR	TCP	1024-65535	Allow inbound traffic from the VPC CIDR on the ephemeral ports
Outbound			
Destination	**Protocol**	**Port**	**Comment**
VPC CIDR	TCP	*instance listener*	Allow outbound traffic to the VPC CIDR on the instance listener port
VPC CIDR	TCP	*health check*	Allow outbound traffic to the VPC CIDR on the health check port
VPC CIDR	TCP	1024-65535	Allow outbound traffic to the VPC CIDR on the ephemeral ports

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Network ACLs for Load Balancers in a VPC

- **Back-End Instances: Recommended Rules**

Inbound			
Source	**Protocol**	**Port**	**Comment**
VPC CIDR	TCP	*instance listener*	Allow inbound traffic from the VPC CIDR on the instance listener port
VPC CIDR	TCP	*health check*	Allow inbound traffic from the VPC CIDR on the health check port
Outbound			
Destination	**Protocol**	**Port**	**Comment**
VPC CIDR	TCP	1024-65535	Allow outbound traffic to the VPC CIDR on the ephemeral ports

For p s.

Add or Remove Subnets for Your Load Balancer

- For load balancers in a VPC, we recommend that you attach one subnet per Availability Zone for at least two Availability Zones.
- By default, the load balancer routes requests evenly across the enabled subnets.
- To route requests evenly across the instances, regardless of subnet, enable cross-zone load balancing.
- You can modify the list of subnets for your load balancer at any time. To expand the availability of your application, launch instances in an additional subnet, register the new instances with your load balancer, and then attach the new subnet to your load balancer.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Connection Draining

- To ensure that the load balancer stops sending requests to instances that are de-registering or unhealthy, while keeping the existing connections open, use *connection draining*.
- This enables the load balancer to complete in-flight requests made to instances that are de-registering or unhealthy.
- When you enable connection draining, you can specify a maximum time for the load balancer to keep connections alive before reporting the instance as de-registered.
- The maximum timeout value can be set between 1 and 3,600 seconds (the default is 300 seconds). When the maximum time limit is reached, the load balancer forcibly closes connections to the de-registering instance.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Proxy Protocol

- Proxy Protocol is an Internet protocol used to carry connection information from the source requesting the connection to the destination for which the connection was requested.
- By default, when you use Transmission Control Protocol (TCP) or Secure Sockets Layer (SSL) for both front-end and back-end connections, your load balancer forwards requests to the back-end instances without modifying the request headers.
- If you enable Proxy Protocol, a human-readable header is added to the request header with connection information such as the source IP address, destination IP address, and port numbers. The header is then sent to the back-end instance as part of the request.
- You can enable Proxy Protocol on ports that use either the SSL and TCP protocols. You can use Proxy Protocol to capture the source IP of your client when you are using a non-HTTP protocol, or when you are using HTTPS and not terminating the SSL connection on your load balancer.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

Sticky Sessions

- By default, a load balancer routes each request independently to the registered instance with the smallest load.
- 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.
- The key to managing sticky sessions is to determine how long your load balancer should consistently route the user's request to the same instance.
- f your application has its own session cookie, then you can configure Elastic Load Balancing so that the session cookie follows the duration specified by the application's session cookie.
- If your application does not have its own session cookie, then you can configure Elastic Load Balancing to create a session cookie by specifying your own stickiness duration.

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.

ytoY technologies®
Engineer yocto to Yotta

Demo of ELB

For personal use only not for distribution | Copyright YTOY Technologies | AWS is Trade Mark of Amazon and All brand and product names are registered trademarks of their respective owners.