**Load Balancer**

**What is Load Balancer?**

* Load Balancer is a virtual machine or appliance that balances your web application load that could be Http or Https traffic that you are getting in. It balances a load of multiple web servers so that no web server gets overwhelmed.
* Elastic Load Balancing automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones.
* It monitors the health of its registered targets, and routes traffic only to the healthy targets.

**Load Balancer Features**

**Security**

When using Amazon Virtual Private Cloud (VPC), you can create and manage security groups associated with Elastic Load Balancing to provide additional networking and security options for Application Load Balancer and Classic Load Balancer. You can configure any of the Load Balancers to be Internet facing or create a load balancer without public IP addresses to serve as an internal (non-internet-facing) load balancer.

**High availability**

An Elastic Load Balancer is highly available. You can distribute incoming traffic across your Amazon EC2 instances in a single Availability Zone or multiple Availability Zones. An Elastic Load Balancer automatically scales its request handling capacity in response to incoming application traffic. To ensure that your targets are available and healthy, Elastic Load Balancer runs health checks on targets on a configurable cadence.

**High throughput**

Elastic Load Balancer is designed to handle traffic as it grows and can load balance millions of requests/sec. It can also handle sudden volatile traffic patterns.

**Health checks**

An Elastic Load Balancer only routes traffic to healthy targets such as EC2 instances, containers, IP addresses, micro services, Lambda functions, and appliances. With Elastic Load Balancing, you get improved insight into the health of your applications in two ways: (1) health check improvements that allow you to configure detailed error codes. The health checks allow you to monitor the health of each of your services behind the load balancer; and (2) new metrics that give insight into traffic for each of the services running on an EC2 instance.

**Sticky sessions**

Sticky sessions are a mechanism to route requests from the same client to the same target. Elastic Load Balancers support sticky sessions. Stickiness is defined at a target group level.

**Delete protection**

You can enable deletion protection on an Elastic Load Balancer to prevent it from being accidentally deleted.

**Types of Elastic Load Balancing**

Elastic Load Balancing supports the following load balancers:

* Application Load Balancers
* Network Load Balancers
* Gateway Load Balancers
* Classic Load Balancers

**Application Load balancers**

* An Amazon Web Services (AWS) launched a new load balancer known as an Application load balancer (ALB) on August 11, 2016.
* It is used to direct user traffic to the public AWS cloud.
* It identifies the incoming traffic and forwards it to the right resources. For example, if a URL has /API extensions, then it is routed to the appropriate application resources.
* It is operated at Layer 7 of the OSI Model.
* It is best suited for load balancing of HTTP and HTTPs traffic.
* Application load balancers are intelligent, sending specific requests to specific web servers.

**Network Load Balancers**

* It is operated at the Layer 4 of the OSI model.
* It makes routing decisions at the transport layer (TCP/SSL), and it can handle millions of requests per second.
* When a load balancer receives a connection, it then selects a target from the target group by using a flow hash routing algorithm. It opens the TCP connection to the selected target of the port and forwards the request without modifying the headers.
* It is best suited for load balancing the TCP traffic when high performance is required.

**Classic Load Balancers**

* It is operated at Layer 4 of the OSI model**.**
* It routes the traffic between clients and backend servers based on IP address
* It can also be used for load balancing the HTTP or HTTPs traffic and use layer 7-specific features, such as X-forwarded and sticky sessions.
* You can also use the Layer 4 load balancing for applications that rely purely on the TCP protocol.

**Gateway Load balancers**

* Gateway Load Balancers enable you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems.
* It combines a transparent network gateway (that is, a single entry and exit point for all traffic) and distributes traffic while scaling your virtual appliances with the demand.
* A Gateway Load Balancer operates at the third layer of the Open Systems Interconnection (OSI) model, the network layer.

**How to Create a Load Balancer:**

Step 1: Configure a target group

Step 2: Register targets

Step 3: Configure a load balancer and a listener

Step 4: Test the load balancer

**Listeners**

* A listener is a process that checks for connection requests, using the protocol and port that you configure.
* The rules that you define for a listener determine how the load balancer routes requests to its registered targets.
* Listeners support the following protocols and ports:
* Protocols: HTTP, HTTPS, TCP, TLS, UDP, TCP\_UDP
* Ports: 1-65535

**Note**: When you create a Listeners for Gateway Load Balancers listen for all IP packets across all ports. You cannot specify a protocol or port when you create a listener for a Gateway Load Balancer. You cannot delete the listener for a Gateway Load Balancer.

**Target Group**

* A target group tells a load balancer where to direct traffic to: EC2 instances, fixed IP addresses; or AWS Lambda functions, amongst others.
* Each target group is used to route requests to one or more registered targets.
* When you create a listener, you specify a target group for its default action.
* Traffic is forwarded to the target group specified in the listener rule.
* You can create different target groups for different types of requests

**Monitoring LB**

You can use the following features to monitor your load balancers, analyze traffic patterns, and troubleshoot issues with your load balancers and targets.

* **Cloud Watch metrics**

You can use Amazon Cloud Watch to retrieve statistics about data points for your load balancers and targets as an ordered set of time-series data, known as metrics. You can use these metrics to verify that your system is performing as expected.

* **Cloud Trail logs**

You can use AWS Cloud Trail to capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in Amazon S3. You can use these Cloud Trail logs to determine which calls were made, the source IP address where the call came from, who made the call, when the call was made, and so on.

* **Access logs**

You can use access logs to capture detailed information about TLS requests made to your load balancer. The log files are stored in Amazon S3. You can use these access logs to analyze traffic patterns and to troubleshoot issues with your targets.