**Elastic Load Balancer :ELB**

* After we kept autoscaling in between the sever the load should be control in that time Elastic load balancer as introduced .
* When we want to conect to the ELB only we can use DNS name.
* Based on the load balance the request will be send to the servers {mainly the request will be validate before sending the response}.
* Any one server completely initiate in private that server should not know by anyone then we will use ELB as a mediator {in this scenario the server will not expose only the load balancer will expose}.
* Instead of seen ip in the details we can see only the cname or DNS name.
* When we are creating ELB compulsory we should have two subnets.
* The subnets should be in different availabilty zones then we can create ELB

**Major Benefits**:

1. **Load balancer**:how many servers in that case the load balancing will happen
2. **health checks**:every adding instances the ELB will check.what ever time we have given like 1 mins or 5 mins in that mean time it will ping those instances.if it is healthy then the rquest will send through that instances.
3. **SSL encryption and decryption**:every request will encrypt

**Types of ELB:**

1. **Classic Load Balancer:**

CLB will work on protocals:http,https,tccp and ssl

http:80

https:443

tcp:80

ssl:443

* we should have equal number ec2 in both the availabilty zones

1. **Application Load Balancer**:

* protocal:http and https
* ALB supports multiple protocals
* In this ALB it will support multiple listeners
* **Rules:**this will decide to which particular target group the trafic should redirected
* applications load balanacer works on layer 7 I.e., application layer of osi model .
* we can configure different rules,listeners it will work based on path preifix.

**use case to use application Load balancer:**

* if we are aware of traffic accessing our application and our application is working on http and https protocols’s then we can use the application load balancer.
* Web apps
* REST APIs
* Microservices architecture

1. **Network Load Balancer**:

protocol: tcp and udp

* **TCP**:is very secure and reliable means the data is transmited is stable.
* **Udp** :it is fast but there might be lot of data missing which data is transmit.
* network load bancer works on layer4 I.e, Transport layer of OSI model.
* we can configure different rules,listeners it will work based on path preifix.
* we can configure one elastic IP.

**use case to use network Load balancer:**

* if we are expection sudden spikes In traffic then network load balancer is smoething which will hep us to manage the traffic and application will be accessible to end user.
* High-performance applications
* Real-time gaming
* VoIP, chat applications
* Load balancing databases or services over TCP/UDP

1. **Gateway Load Balancer (GWLB):**

Works with Layer 3/4)

* Gateway Load Balancer (GWLB) is designed to deploy, scale, and manage third-party virtual network appliances like:

Firewalls (e.g., Palo Alto, Fortinet)

Intrusion Detection/Prevention Systems (IDS/IPS)

Packet analyzers

Deep packet inspection tools

Custom security appliances

* Combines a transparent network gateway with load balancing
* Integrates with VPC traffic mirroring

**🧩 Use Case:**

* Security appliances
* Deep packet inspection
* Network traffic inspection and filtering