AWS Network Load Balancer – NLB

# Graphical user interface, application Description automatically generated

# AWS Network Load Balancer – NLB

* Network Load Balancer operates at the connection level (Layer 4), routing connections to targets – EC2 instances, containers and IP addresses based on IP protocol data.
* Network Load Balancer is suited for load balancing of TCP traffic
* **Network Load Balancer is capable of handling millions of requests per second while maintaining ultra-low latencies.**
* **Network Load Balancer is optimized to handle sudden and volatile traffic patterns while using a single static IP address per Availability Zone.**
* Network Load Balancer also supports TLS termination, preserves the source IP of the clients, and provides stable IP support and Zonal isolation.
* **NLB supports long-running connections that are very useful for WebSocket type applications.**
* NLB is integrated with other AWS services such as Auto Scaling, EC2 Container Service (ECS), and CloudFormation.
* NLB support connections from clients over VPC peering, AWS managed VPN, and third-party VPN solutions.
* For TCP traffic,
  + the load balancer selects a target using a flow hash algorithm based on the protocol, source IP address, source port, destination IP address, destination port, and TCP sequence number.
  + TCP connections from a client have different source ports and sequence numbers, and can be routed to different targets.
  + Each individual TCP connection is routed to a single target for the life of the connection.
* For UDP traffic,
  + the load balancer selects a target using a flow hash algorithm based on the protocol, source IP address, source port, destination IP address, and destination port.
  + A UDP flow has the same source and destination, so it is consistently routed to a single target throughout its lifetime.
  + Different UDP flows have different source IP addresses and ports, so they can be routed to different targets.
* **back-end server authentication is not supported**
* **session stickiness is not support**

## Network Load Balancer Features

### Connection-based Load Balancing

* Allows load balancing of  TCP traffic, routing connections to targets – EC2 instances, microservices and containers, and IP addresses.

### High Availability

* is highly available.
* accepts incoming traffic from clients and distributes this traffic across the targets within the same Availability Zone.
* monitors the health of its registered targets and routes the traffic only to healthy targets
* if a health check fails and an unhealthy target is detected, it stops routing traffic to that target and reroutes traffic to remaining healthy targets.
* if configured with multiple AZs and if all the targets in a single AZ fail, it routes traffic to healthy targets in the other AZs

### High Throughput

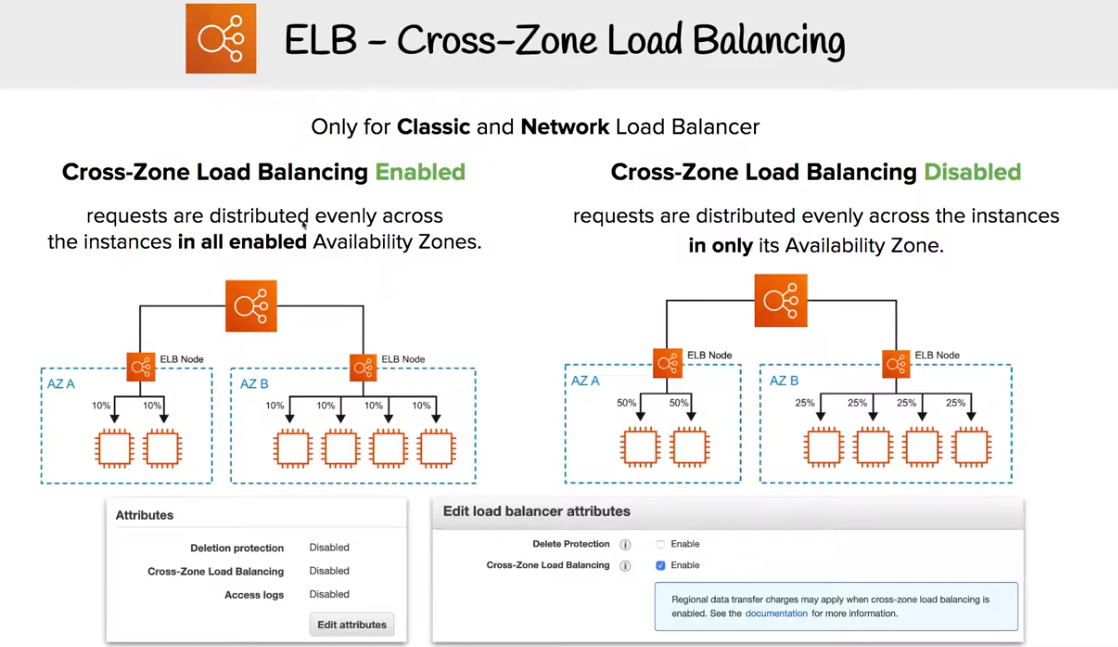
* is designed to handle traffic as it grows and can load balance millions of requests/sec.
* can also handle sudden volatile traffic patterns.

### Low Latency

* offers extremely low latencies for latency-sensitive applications.

### Cross Zone Load Balancing

* enable cross-zone loading balancing only after creating the NLB
* is disabled, by default, and charges apply for inter-az traffic.



### Load Balancing using IP addresses as Targets

* allows load balancing of any application hosted in AWS or **on-premises** using IP addresses of the application backends as targets.
* allows load balancing to an application backend hosted on any IP address and any interface on an instance.
* ability to load balance across AWS and on-premises resources helps migrate-to-cloud, burst-to-cloud or failover-to-cloud
* applications hosted in on-premises locations can be used as targets over a Direct Connect connection and EC2-Classic (using ClassicLink).

### Preserve source IP address

* preserves client side source IP allowing the back-end to see client IP address
* Target groups can be created with target type as instance ID or IP address.
  + **If targets registered by instance ID, the source IP addresses of the clients are preserved and provided to the applications.**
  + **If register targets registered by IP address, the source IP addresses are the private IP addresses of the load balancer nodes.**

Graphical user interface

Description automatically generated

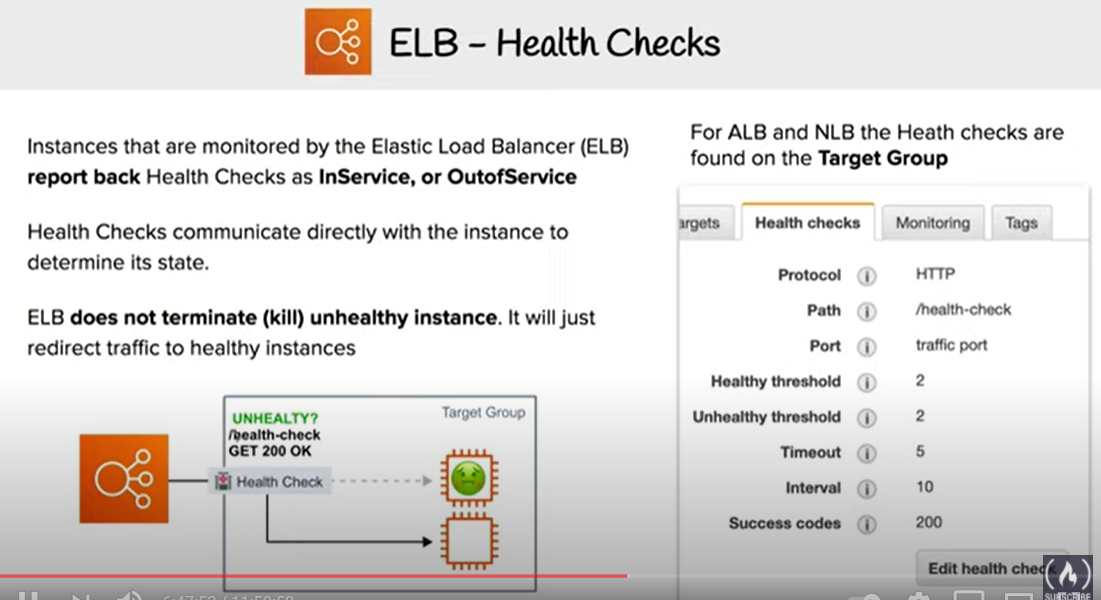
### Static IP support

* automatically provides a static IP per Availability Zone (subnet) that can be used by applications as the front-end IP of the load balancer.
* Elastic Load Balancing creates a network interface for each enabled Availability Zone. Each load balancer node in the AZ uses this network interface to get a static IP address.
* Internet-facing load balancer can optionally associate one Elastic IP address per subnet.

### Elastic IP support

* an Elastic IP per Availability Zone (subnet) can also be assigned, optionally, thereby providing a fixed IP.

### Health Checks



* supports both network and application target health checks.
* Network-level health check
  + is based on the overall response of the underlying target (instance or a container) to normal traffic.
  + target is marked unavailable if it is slow or unable to respond to new connection requests
* Application-level health check
  + is based on a specific URL on a given target to test the application health deeper

### DNS Fail-over

* integrates with Route 53
* Route 53 will direct traffic to load balancer nodes in other AZs, if there are no healthy targets with NLB or if the NLB itself is unhealthy
* if NLB is unresponsive, Route 53 will remove the unavailable load balancer IP address from service and direct traffic to an alternate Network Load Balancer in another region.

### Integration with AWS Services

* is integrated with other AWS services such as Auto Scaling, EC2 Container Service (ECS), CloudFormation, CodeDeploy, and AWS Config.

### Long-lived TCP Connections

* **supports long-lived TCP connections ideal for WebSocket type of applications**

### Central API Support

* uses the same API as Application Load Balancer.
* enables you to work with target groups, health checks, and load balance across multiple ports on the same EC2 instance to support containerized applications.

### Robust Monitoring and Auditing

* integrated with CloudWatch to report Network Load Balancer metrics.
* CloudWatch provides metrics such as Active Flow count, Healthy Host Count, New Flow Count, Processed bytes, and more.
* integrated with CloudTrail to track API calls to the NLB

### Enhanced Logging

* use the Flow Logs feature to record all requests sent to the load balancer.
* Flow Logs capture information about the IP traffic going to and from network interfaces in the VPC
* Flow log data is stored using CloudWatch Logs

### Zonal Isolation

* is designed for application architectures in a single zone.
* can be enabled in a single AZ to support architectures that require zonal isolation
* automatically fails-over to other healthy AZs, if something fails in a AZ
* its recommended to configure the load balancer and targets in multiple AZs for achieving high availability

## Graphical user interface Description automatically generated

SSL/TLS Termination

Diagram

Description automatically generated with medium confidence

## Advantages over Classic Load Balancer

* Ability to handle volatile workloads and scale to millions of requests per second, without the need of pre-warming
* Support for static IP/Elastic IP addresses for the load balancer
* Support for registering targets by IP address, including targets outside the VPC (on-premises) for the load balancer.
* Support for routing requests to multiple applications on a single EC2 instance. Single instance or IP address can be registered with the same target group using multiple ports.
* Support for containerized applications. Using Dynamic port mapping, ECS can select an unused port when scheduling a task and register the task with a target group using this port.
* Support for monitoring the health of each service independently, as health checks are defined at the target group level and many CloudWatch metrics are reported at the target group level. Attaching a target group to an Auto Scaling group enables scaling each service dynamically based on demand

## Network Load Balancer Pricing

* charged for each hour or partial hour that an NLB is running and the number of Load Balancer Capacity Units (LCU) used per hour.
* An LCU is a new metric for determining NLB pricing
* An LCU defines the maximum resource consumed in any one of the dimensions (new connections/flows, active connections/flows, bandwidth and rule evaluations) the Network Load Balancer processes your traffic.

## AWS Certification Exam Practice Questions

1. A company wants to use load balancer for their application. However, the company wants to forward the requests without any header modification. What service should the company use?
   1. Classic Load Balancer
   2. **Network Load Balancer**
   3. Application Load Balancer
   4. Use Route 53
2. A company is hosting an application in AWS for third party access. The third party needs to whitelist the application based on the IP. Which AWS service can the company use in the whitelisting of the IP address?
   1. AWS Application Load Balancer
   2. AWS Classic Load balancer
   3. **AWS Network Load Balancer**
   4. AWS Route 53