Comparison of ALB, NLB and Classic Load Balancer

Comparison of Elastic Load Balancing Products

You can select the appropriate load balancer based on your application needs. If you need flexible application management, we recommend that you use an **Application Load Balancer**. If extreme performance and static IP is needed for your application, we recommend that you use a **Network Load Balancer**. If you have an existing application that was built within the EC2-Classic network, then you should use a **Classic Load Balancer**.

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocols	HTTP, HTTPS	TCP	TCP, SSL, HTTP, HTTPS
Platforms	VPC	VPC	EC2-Classic, VPC
Health checks	✓	✓	✓
CloudWatch metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal fail-over	✓	✓	✓
Connection draining (deregistration delay)	✓	✓	✓
Load Balancing to multiple ports on the same instance	✓	✓	
WebSockets	✓	✓	
IP addresses as targets	✓	✓	
Load balancer deletion protection	✓	✓	
Path-Based Routing	✓		
Host-Based Routing	✓		
Native HTTP/2	✓		
Configurable idle connection timeout	✓		✓
Cross-zone load balancing	✓		✓
SSL offloading	✓		✓
Server Name Indication (SNI)	✓		
Sticky sessions	✓		✓
Back-end server encryption	✓		✓
Static IP		✓	
Elastic IP address		✓	
Preserve Source IP address		✓	

Details for Elastic Load Balancing Products

Application Load Balancer | Network Load Balancer | Classic Load Balancer

Application Load Balancer operates at the request level (layer 7), routing traffic to targets - EC2 instances, containers and IP addresses based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer simplifies and improves the security of your application, by ensuring that the latest SSL/TLS ciphers and protocols are used at all times.

Key Features

Layer-7 Load Balancing
You can load balance HTTP/HTTPS applications and use layer 7-specific features, such as X-Forwarded-For headers.

HTTPS Support
An Application Load Balancer supports HTTPS termination between the clients and the load balancer. Application Load Balancers also offer management of SSL certificates through AWS Identity and Access Management (IAM) and AWS Certificate Manager for pre-defined security policies.

Server Name Indication (SNI)
Server Name Indication (SNI) is an extension to the TLS protocol by which a client indicates the hostname to connect to at the start of the TLS handshake. The load balancer can present multiple certificates through the same secure listener, which enables it to support multiple secure websites using a single secure listener. Application Load Balancers also support a smart certificate selection algorithm with SNI. If the hostname indicated by a client matches multiple certificates, the load balancer determines the best certificate to use based on multiple factors including the capabilities of the client.

IP addresses as Targets
You can load balance any application hosted in AWS or on-premises using IP addresses of the application backends as targets. This allows load balancing to an application backend hosted on any IP address and any interface on an instance. Each application hosted on the same instance can have an associated security group and use the same port. You can also use IP addresses as targets to load balance applications hosted in on-premises locations (over a Direct Connect or VPN connection), peered VPCs and EC2-Classic (using ClassicLink). The ability to load balance across AWS and on-prem resources helps you migrate-to-cloud, burst-to-cloud or failover-to-cloud.

High Availability
An Application Load Balancer requires you to specify more than one Availability Zone. You can distribute incoming traffic across your targets in multiple Availability Zones. An Application Load Balancer automatically scales its request handling capacity in response to incoming application traffic.

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

Content-Based Routing
If your application is composed of several individual services, an Application Load Balancer can route a request to a service based on the content of the request.

- **Host-based Routing**
  You can route a client request based on the Host field of the HTTP header allowing you to route to multiple domains from the same load balancer.
- **Path-based Routing**
  You can route a client request based on the URL path of the HTTP header.

Application Load Balancer | **Network Load Balancer** | Classic Load Balancer

Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, containers and IP addresses based on IP protocol data. Ideal 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. It is integrated with other popular AWS services such as Auto Scaling, Amazon EC2 Container Service (ECS), and Amazon CloudFormation.

Key Features

Connection-based Load Balancing
You can load balance TCP traffic, routing connections to targets - Amazon EC2 instances, microservices and containers, and IP addresses.

High Availability
Network Load Balancer is highly available. It accepts incoming traffic from clients and distributes this traffic across the targets within the same Availability Zone. The load balancer also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets. When the load balancer detects an unhealthy target, it stops routing traffic to that target and reroutes traffic to remaining healthy targets. If all of your targets in one Availability Zone are unhealthy, and you have set up targets in another Availability Zone, Network Load Balancer will automatically fail-over to route traffic to your healthy targets in the other Availability Zones.

High Throughput
Network 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.

Low Latency
Network Load Balancer offers extremely low latencies for latency-sensitive applications.

Preserve source IP address
Network Load Balancer preserves the client side source IP allowing the back-end to see the IP address of the client. This can then be used by applications for further processing.

Static IP support
Network Load Balancer 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 IP support
Network Load Balancer also allows you the option to assign an Elastic IP per Availability Zone (subnet) thereby providing your own fixed IP.

Health Checks
Network Load Balancer supports both network and application target health checks. Network-level health is based on the overall response of your target to normal traffic. If the target becomes unable, or too slow, to respond to new connections then the load balancer will mark the target as unavailable. Application-level health checks can also be used to go deeper. By periodically probing a specific URL on a given target, it can integrate the health of the actual application. For quick diagnosis and powerful debugging, full visibility into health checks and why they may be failing is also available through 'reason codes' in the Network Load Balancer API, and the Amazon CloudWatch metrics attached to target health checks.

DNS Fail-over
If there are no healthy targets registered with the Network Load Balancer or if the Network Load Balancer nodes in a given zone are unhealthy, then Amazon Route 53 will direct traffic to load balancer nodes in other Availability Zones.

Integration with Amazon Route 53
In the event that your Network load balancer is unresponsive, integration with 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
Network Load Balancer is integrated with other AWS services such as Auto Scaling, Amazon EC2 Container Service (ECS), AWS CloudFormation, AWS CodeDeploy, and AWS Config.

Long-lived TCP Connections
Network Load Balancer supports long-lived TCP connections that are ideal for WebSocket type of applications.

Central API Support
Network Load Balancer uses the same API as Application Load Balancer. This will enable you to work with target groups, health checks, and load balance across multiple ports on the same Amazon EC2 instance to support containerized applications.

Robust Monitoring and Auditing
Amazon CloudWatch reports Network Load Balancer metrics. CloudWatch provides metrics such as Active Flow count, Healthy Host Count, New Flow Count, Processed bytes, and more. Network Load Balancer is also integrated with AWS CloudTrail. CloudTrail tracks API calls to the Network Load Balancer.

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

Zonal Isolation
The Network Load Balancer is designed for application architectures in a single zone. If something in the Availability Zone fails, we will automatically fail-over to other healthy Availability Zones. While we recommend customers configure the load balancer and targets in multiple AZs for achieving high availability, Network Load Balancer can be enabled in a single Availability Zone to support architectures that require zonal isolation.

Load Balancing using IP addresses as Targets
You can load balance any application hosted in AWS or on-premises using IP addresses of the application backends as targets. This allows load balancing to an application backend hosted on any IP address and any interface on an instance. Each application hosted on the same instance can have an associated security group and use the same port. You can also use IP addresses as targets to load balance applications hosted in on-premises locations (over a Direct Connect connection) and EC2-Classic (using ClassicLink). The ability to load balance across AWS and on-prem resources helps you migrate-to-cloud, burst-to-cloud or failover-to-cloud.

Application Load Balancer | Network Load Balancer | **Classic Load Balancer**

Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network. We recommend Application Load Balancer for Layer 7 and Network Load Balancer for Layer 4 when using Virtual Private Cloud (VPC).

Key Features

High Availability
You can distribute incoming traffic across your Amazon EC2 instances in a single Availability Zone or multiple Availability Zones. Classic Load Balancer automatically scales its request handling capacity in response to incoming application traffic.

Health Checks
Classic Load Balancer can detect the health of Amazon EC2 instances. When it detects unhealthy EC2 instances, it no longer routes traffic to those instances and spreads the load across the remaining healthy instances.

Security Features
When using Amazon Virtual Private Cloud (Amazon VPC), you can create and manage security groups associated with Classic Load Balancer to provide additional networking and security options. You can also create a Classic Load Balancer without public IP addresses to serve as an internal (non-internet-facing) load balancer.

SSL Offloading
Classic Load Balancer supports SSL termination, including offloading SSL decryption from application instances, centralized management of SSL certificates, and encryption to back-end instances with optional public key authentication. Flexible cipher support allows you to control the ciphers and protocols the load balancer presents to clients.

Sticky Sessions
Classic Load Balancer supports the ability to stick user sessions to specific Amazon EC2 instances using cookies. Traffic will be routed to the same instances as the user continues to access your application.

IPv6 Support
Classic Load Balancer supports the use of both the Internet Protocol version 4 and 6 (IPv4 and IPv6) for EC2-Classic networks.

Layer 4 or Layer 7 Load Balancing
You can load balance HTTP/HTTPS applications and use Layer 7-specific features, such as X-Forwarded and sticky sessions. You can also use strict Layer 4 load balancing for applications that rely purely on the TCP protocol.

Operational Monitoring
Classic Load Balancer metrics such as request count and request latency are reported by Amazon CloudWatch.

Logging
Use the Access Logs feature to record all requests sent to your load balancer, and store the logs in Amazon S3 for later analysis. The logs are useful for diagnosing application failures and analyzing web traffic. You can use AWS CloudTrail to record Classic Load Balancer API calls for your account and deliver log files. The API call history enables you to perform security analysis, resource change tracking, and compliance auditing.

Containerized Application Support
Application Load Balancer provides enhanced container support by load balancing across multiple ports on a single Amazon EC2 instance. Deep integration with the Amazon EC2 Container Service (ECS), provides a fully-managed container offering. ECS allows you to specify a dynamic port in the ECS task definition, giving the container an unused port when it is scheduled on the EC2 instance. The ECS scheduler automatically adds the task to the load balancer using this port.

HTTP/2 Support
HTTP/2 is a new version of the HyperText Transfer Protocol (HTTP) that uses a single, multiplexed connection to allow multiple requests to be sent on the same connection. It also compresses header data before sending it out in binary format and supports TLS connections to clients.

WebSockets Support
WebSockets allows a server to exchange real-time messages with end-users without the end users having to request (or poll) the server for an update. The WebSockets protocol provides bi-directional communication channels between a client and a server over a long-running TCP connection.

Native IPv6 Support
Application Load Balancers support native Internet Protocol version 6 (IPv6) in aVPC. This will allow clients to connect to the Application Load Balancer via IPv4 or IPv6.

Sticky Sessions
Sticky sessions are a mechanism to route requests from the same client to the same target. Application Load Balancer supports sticky sessions using load balancer generated cookies. If you enable sticky sessions, the same target receives the request and can use the cookie to recover the session context. Stickiness is defined at a target group level.

Health Checks
An Application Load Balancer routes traffic only to healthy targets. With an Application Load Balancer, 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 from 200-499. 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.

Operational Monitoring
Amazon CloudWatch reports Application Load Balancer metrics such as request counts, error counts, error types, and request latency.

Logging
You can use the Access Logs feature to record all requests sent to your load balancer, and store the logs in Amazon S3 for later analysis. The logs are compressed and have a gzip file extension. The compressed logs save both storage space and transfer bandwidth and are useful for diagnosing application failures and analyzing web traffic.

You can also use AWS CloudTrail to record Application Load Balancer API calls for your account and deliver log files. The API call history enables you to perform security analysis, resource change tracking, and compliance auditing.

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

Request Tracing
The Application Load Balancer injects a new custom identifier “X-Amzn-Trace-Id” HTTP header on all requests coming into the load balancer. Request tracing allows you to track a request by its unique ID as the request makes its way across various services that make up the your websites and distributed applications. You can use the unique trace identifier to uncover any performance or timing issues in your application stack at the granularity of an individual request.

Web Application Firewall
You can now use AWS WAF to protect your web applications on your Application Load Balancers. AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.