



What Is Load Balancing?

Load Balancing



[Home](#) » Load Balancing Guide

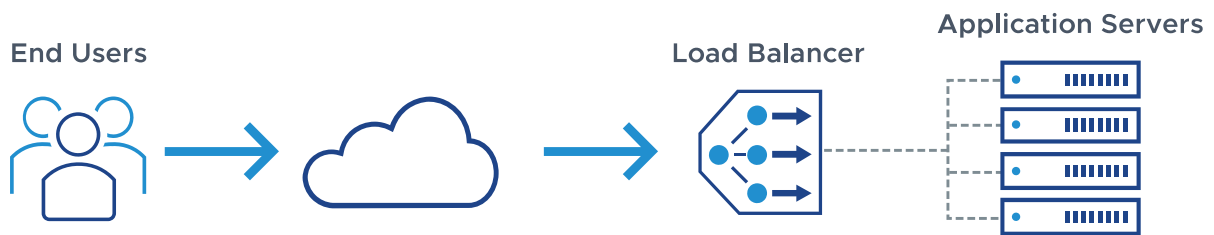
An Introduction to Load Balancing

Load Balancing Definition: Load balancing is the process of distributing network traffic across multiple servers. This ensures no single server bears too much demand. By spreading the work evenly, load balancing improves application responsiveness. It also increases availability of applications and websites.



[Cookie Settings](#)

for users. Modern applications cannot run without load balancers. Over time, **load balancers** have added additional capabilities including **security** and **application acceleration**



When one application server becomes unavailable, the load balancer directs all new application requests to other available servers.

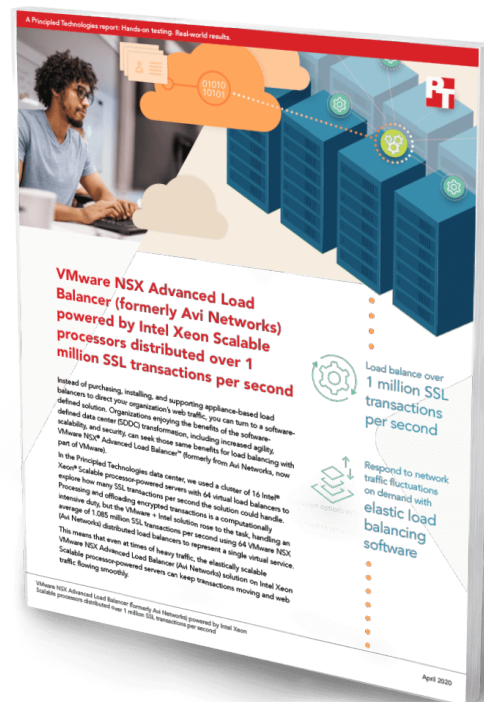
About Load Balancers

As an organization meets demand for its applications, the load balancer decides which servers can handle that traffic. This maintains a good user experience.

Load balancers manage the flow of information between the server and an endpoint device (PC, laptop, tablet or smartphone). The server could be on-premises, in a **data center** or the public cloud. The server can also be physical or virtualized. The load balancer helps servers move data efficiently, optimizes the use of **application delivery** resources and prevents server overloads. Load balancers conduct continuous health checks on servers to ensure they can handle requests. If necessary, the load balancer

removes unhealthy servers from the pool until they are restored. Some load balancers even trigger the creation of new virtualized application servers to cope with increased demand.

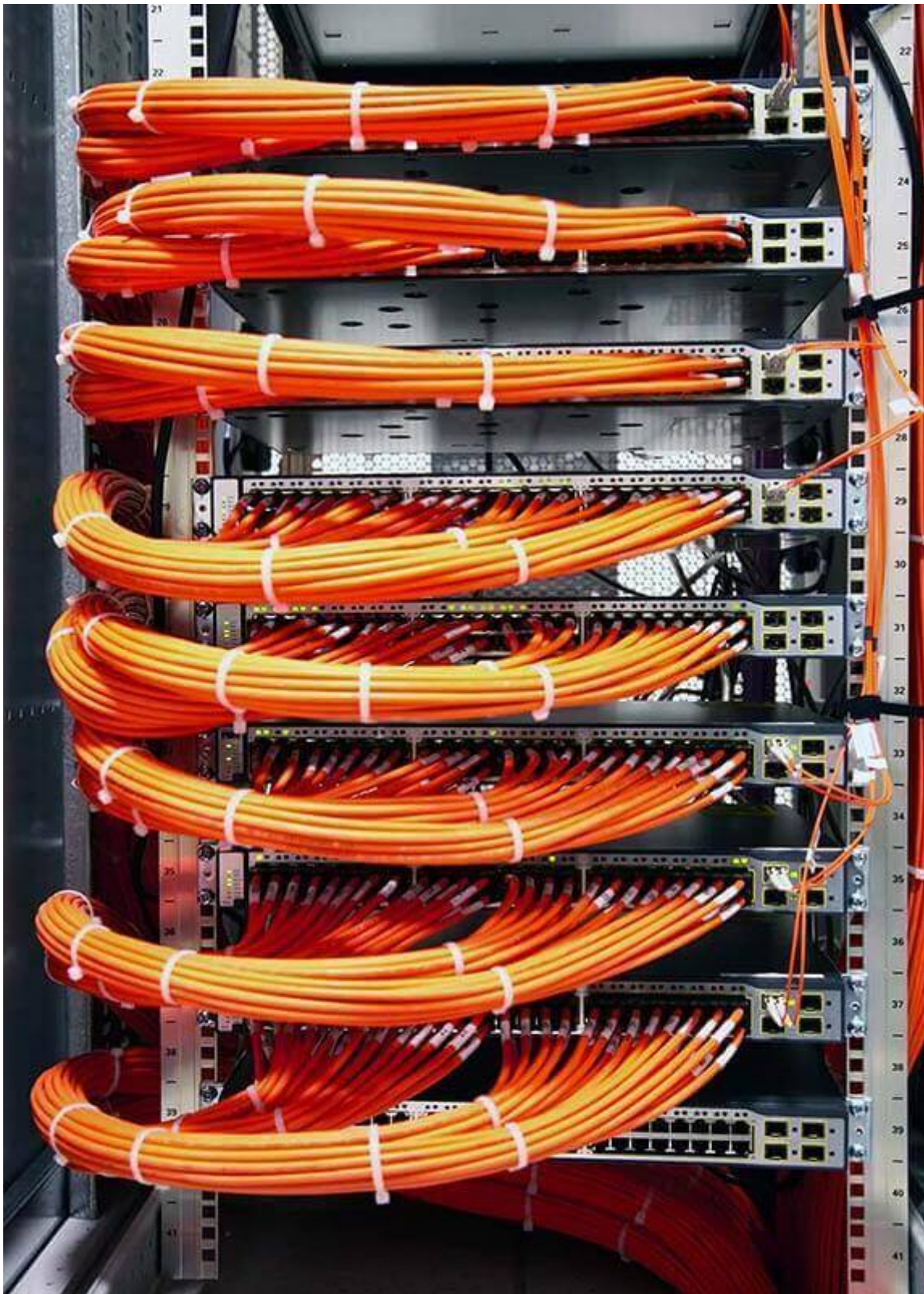
Traditionally, load balancers consist of a hardware appliance. Yet they are increasingly becoming software-defined. This is why load balancers are an essential part of an organization's digital strategy.



Download this report to learn how the NSX Advanced Load Balancer (Avi Networks) distributed over 1 million SSL transactions per second with the help of Intel's high-performance CPUs

[DOWNLOAD HERE](#) ➤

History of Load Balancing



Load balancing got its start in the 1990s as hardware appliances distributing traffic across a network. Organizations wanted to improve accessibility of applications running on servers. Eventually, [load balancing](#) took on more responsibilities with the advent of [Application Delivery Controllers \(ADCs\)](#). They provide security along with seamless access to applications at peak times.

ADCs fall into three categories: hardware appliances, virtual appliances (essentially the software extracted from legacy hardware) and software-native load balancers. As computing moves to the cloud, software ADCs perform similar tasks to hardware. They also come with added functionality and flexibility. They let an organization quickly and securely scale up its application services based on demand in the cloud. Modern ADCs allow organizations to consolidate network-based services. Those services include [SSL/TLS offload](#), caching, compression, intrusion detection and [web application firewalls \(WAF\)](#). This creates even shorter delivery times and greater scalability.

Load Balancing and SSL

Secure Sockets Layer (SSL) is the standard security technology for establishing an encrypted link between a web server and a browser. [SSL](#) traffic is often decrypted at the load balancer. When a load balancer decrypts traffic before passing the request on, it is called SSL termination. The load balancer saves the web servers from having to expend the extra CPU cycles required for decryption. This improves [application performance](#).

However, [SSL termination](#) comes with a security concern. The traffic between the load balancers and the web servers is no longer encrypted. This can expose the application to possible attack. However, the risk is lessened when the load balancer is within the same data center as the web servers.

Another solution is the [SSL pass-through](#). The load balancer merely passes an encrypted request to the web server. Then the web server does the decryption. This uses more CPU power on the web server. But organizations that require extra security may find the extra overhead worthwhile.

Load Balancing and Security

Load Balancing plays an important security role as computing moves evermore to the cloud. The off-loading function of a load balancer defends an organization against [distributed denial-of-service \(DDoS\) attacks](#). It does this by shifting attack traffic from a corporate server to a public cloud provider. [DDoS attacks](#) represent a large portion of cybercrime as their number and size continues to rise. Hardware defense, such as a

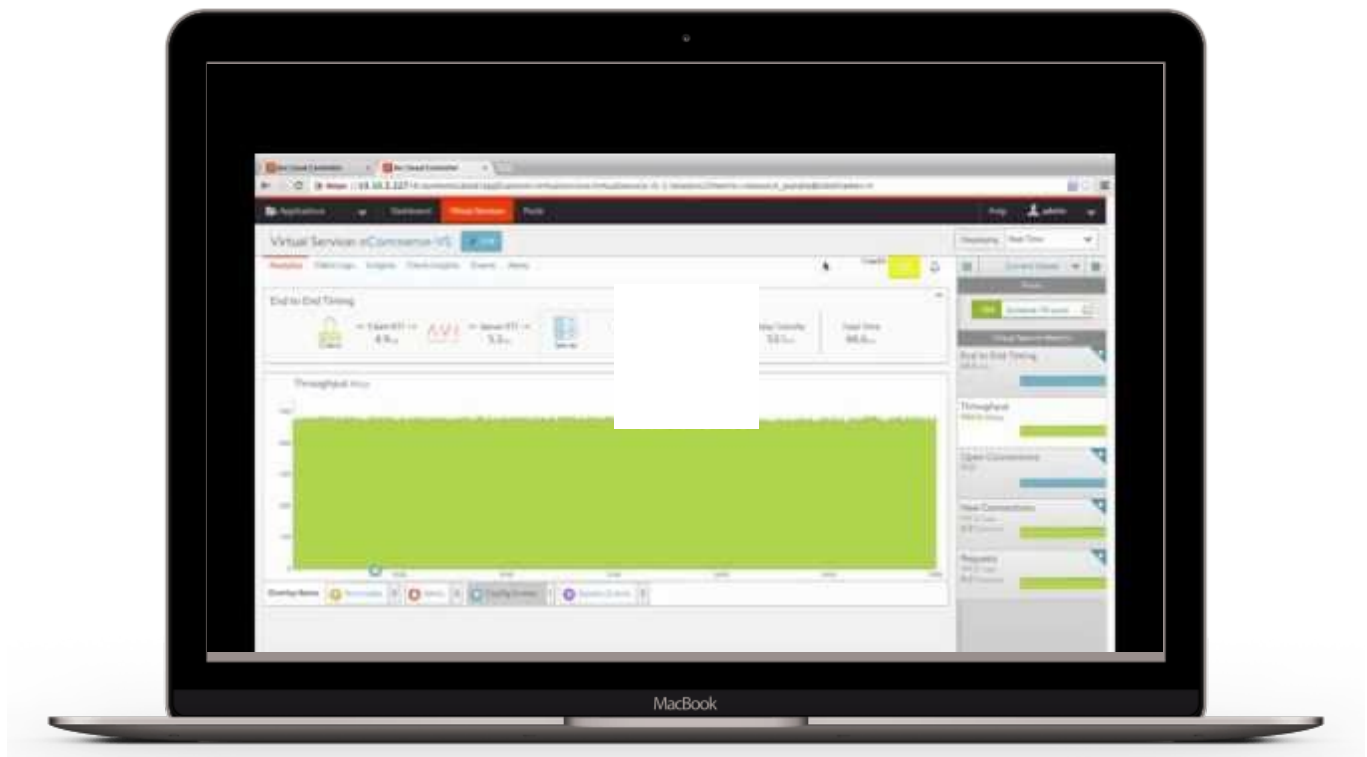
perimeter firewall, can be costly and require significant maintenance. Software load balancers with cloud offload provide efficient and cost-effective protection.



Load Balancing Algorithms

There is a variety of [load balancing methods](#), which use different algorithms best suited for a particular situation.

- Least Connection Method — directs traffic to the server with the fewest active connections. Most useful when there are a large number of persistent connections in the traffic unevenly distributed between the servers.
- Least Response Time Method — directs traffic to the server with the fewest active connections and the lowest average response time.
- Round Robin Method — rotates servers by directing traffic to the first available server and then moves that server to the bottom of the queue. Most useful when servers are of equal specification and there are not many persistent connections.
- IP Hash — the IP address of the client determines which server receives the request.



Load balancing has become a necessity as applications become more complex, user demand grows and traffic volume increases. Load balancers allow organizations to build flexible networks that can meet new challenges without compromising security, service or performance.

Load Balancing Benefits

Load balancing can do more than just act as a network traffic cop. [Software load balancers](#) provide benefits like predictive analytics that determine traffic bottlenecks before they happen. As a result, the software load balancer gives an organization actionable insights. These are key to automation and can help drive business decisions.

In the seven-layer [Open System Interconnection \(OSI\) model](#), network firewalls are at levels one to three (L1-Physical Wiring, L2-Data Link and L3-Network). Meanwhile, load balancing happens between layers four to seven (L4-Transport, L5-Session, L6-Presentation and L7-Application).

Load balancers have different capabilities, which include:



- L4 — directs traffic based on data from network and transport layer protocols, such as IP address and TCP port.
- L7 — adds content switching to load balancing. This allows routing decisions based on attributes like HTTP header, uniform resource identifier, SSL session ID and HTML form data.
- GSLB — [Global Server Load Balancing](#) extends L4 and L7 capabilities to servers in different geographic locations.

More enterprises are seeking to deploy cloud-native applications in data centers and public clouds. This is leading to significant changes in the capability of load balancers. In turn, this creates both challenges and opportunities for infrastructure and operations leaders.

For more on the actual implementation of load balancers, check out our [Application Delivery How-To Videos](#) or watch the Global Server Load Balancing How To Video here:



Software load balancers provide predictive analytics that determine traffic bottlenecks before they happen.



Actionable insights by load balancers that can help drive business decisions.



Global Server Load Balancing extends L4 and L7 load balancing capabilities to servers in different geographic locations.



Load Balancing with App Insights

Using a Software Load Balancer for Application Monitoring, Security, and End User Intelligence

- Administrators can have actionable application insights at their fingertips
- Reduce troubleshooting time from days to mere minutes
- Avoid finger-pointing and empowers collaborative issue resolution



Software Load Balancers vs. Hardware Load Balancers

Load balancers run as hardware appliances or are software-defined. Hardware appliances often run proprietary software optimized to run on custom processors. As traffic increases, the vendor simply adds more load balancing appliances to handle the volume. [Software defined load balancers](#) usually run on less-expensive, standard Intel x86 hardware. Installing the software in cloud environments like AWS EC2 eliminates the need for a physical appliance.

SOFTWARE PROS

- Flexibility to adjust for changing needs.
- Ability to scale beyond initial capacity by adding more software instances.
- Lower cost than purchasing and maintaining physical machines. Software can run on any standard device, which tends to be cheaper.
- Allows for load balancing in the cloud, which provides a managed, off-site solution that can draw resources from an elastic network of servers. [Cloud computing](#) also allows for the flexibility of hybrid hosted and in-house solutions. The main load balancer could be in-house while the backup is a cloud load balancer.

SOFTWARE CONS

- When scaling beyond initial capacity, there can be some delay while configuring load balancer software.
- Ongoing costs for upgrades.

HARDWARE PROS

- Fast throughput due to software running on specialized processors.
- Increased security since only the organization can access the servers physically.
- Fixed cost once purchased.

HARDWARE CONS


- Require more staff and expertise to configure and program the physical machines.
- Inability to scale when the set limit on number of connections has been made. Connections are refused or service degraded until additional machines are purchased and installed.
- Higher cost for purchase and maintenance of physical [network load balancer](#). Owning a hardware load balancer may also require paying for consultants to manage it.

DNS Load Balancing vs Hardware Load Balancing

[DNS load balancing](#) is a software-defined approach to load balancing where client requests to a domain within the [Domain Name System \(DNS\)](#) are distributed across different server machines. The DNS system sends a different version of the list of IP addresses each time it responds to a new client request using the round-robin method, therefore distributing the DNS requests evenly to different servers to handle the overall load. This in turn provides DNS load balancing failover protection through automatic removal of non-responsive servers. ^

DNS load balancing differs from [hardware load balancing](#) in a few instances, although both can be a very effective solution for distributing traffic. One main advantage of DNS level load balancing is the scalability and price. A DNS load balancer distributes traffic to several different IP addresses, whereas the hardware solution uses a single IP address and splits traffic leading to it on multiple servers. As for pricing, hardware load balancers require a large upfront cost whereas DNS load balancers can be scaled as needed.

Types of Load Balancing

- SDN — Load balancing using [SDN \(software-defined networking\)](#) separates the control plane from the data plane for application delivery. This allows the control of multiple load balancing. It also helps the network to function like the virtualized versions of compute and storage. With the centralized control, networking policies and parameters can be programmed directly for more responsive and efficient application services. This is how networks can become more agile.
- UDP — A UDP load balancer utilizes User Datagram Protocol (UDP). [UDP load balancing](#) is often used for live broadcasts and online games when speed is important and there is little need for error correction. UDP has low latency because it does not provide time-consuming health checks.
- TCP — A TCP load balancer uses transmission control protocol (TCP). [TCP load balancing](#) provides a reliable and error-checked stream of packets to IP addresses, which can otherwise easily be lost or corrupted.
- SLB— Server Load Balancing (SLB) provides network services and content delivery using a series of load balancing algorithms. It prioritizes responses to the specific requests from clients over the network. [Server load balancing](#) distributes client traffic to servers to ensure consistent, high-performance application delivery.
- Virtual — Virtual load balancing aims to mimic software-driven infrastructure through virtualization. It runs the software of a physical load balancing appliance on a virtual machine. [Virtual load balancers](#), however, do not avoid the architectural challenges of traditional hardware appliances which include limited scalability and automation, and lack of central management.
- Elastic — [Elastic Load Balancing](#) scales traffic to an application as demand changes over time. It uses system health checks to learn the status of application pool memb  (application servers) and routes traffic appropriately to available servers, manages fail-over to high availability targets, or automatically spins-up additional capacity.

- Geographic — Geographic load balancing redistributes application traffic across data centers in different locations for maximum efficiency and security. While local load balancing happens within a single data center, [geographic load balancing](#) uses multiple data centers in many locations.
- Multi-site — Multi-site load balancing, also known as global server load balancing (GSLB), distributes traffic across servers located in multiple sites or locations around the world. The servers can be on-premises or hosted in a public or private cloud. [Multi-site load balancing](#) is important for quick disaster recovery and business continuity after a disaster in one location renders a server inoperable.
- Load Balancer as a Service (LBaaS) — Load Balancer as a Service (LBaaS) uses advances in load balancing technology to meet the agility and application traffic demands of organizations implementing private cloud infrastructure. Using an as-a-service model, [LBaaS](#) creates a simple model for application teams to spin up load balancers.

Per App Load Balancing

A per-app approach to load balancing equips an application with a dedicated set of application services to scale, accelerate, and secure the application. Per app load balancing provides a high degree of application isolation, avoids over-provisioning of load balancers, and eliminates the constraints of supporting numerous applications on one load balancer.

Load balancing automation tools deploy, configure, and scale load balancers as needed to maintain performance and availability of applications, eliminating the need to code custom scripts per-app or per-environment. Per [application load balancing](#) offers a cost-efficient, elastic scale based on learned traffic thresholds and is particularly beneficial for applications that have matured beyond the limitations of a traditional, hardware load balancer.

What is Weighted Load Balancing?



Weighted load balancing is the process of permitting users to set a respective weight for each origin server in a pool. It's important to consider weighted load balancing because of its ability to rebalance traffic when an origin becomes unhealthily crowded. Depending on their respective weights and the load balancing weight priority, traffic will be rebalanced to the remaining accessible origins.

An underestimated aspect to weighted load balancing are the nodes. Nodes that restart begin again with an empty cache, and while the cache is repopulating it makes the node slower, which results in slowing down the entire collection. This is where heat weighted load balancing comes into focus by aiming to have low latency. The heat of each node is a factor in enhancing the node selection in the coordinator, so as a node is being rebooted, latency remains at a low level.

Weighted Load Balancing vs Round Robin

Round robin load balancing has client requests allocated throughout a group of servers that are readily available, then is followed by all requests redirected to each server in turn. In contrast to the weighted load balancing algorithm, the weighted [round robin load balancing](#) algorithm is used to schedule data flows and processes in networks. This process becomes cyclical when the algorithm commands the load balancer to return to the beginning of the list and repeat its procedure again. Reliable and efficient, weighted round robin load balancing is a simple method and the most commonly used load balancing algorithm.

Load Balancer Health Check

Periodically, load balancers will perform a series of health checks to make sure registered instances are being monitored. Regardless of the instances being in a healthy or

unhealthy state, all registered instances will receive [load balancer health checks](#). An instance health status shows as such:

- Healthy Instance = “InService”
- Unhealthy Instance = “OutOfService”

The load balancer will only send requests to healthy instances, so it will not send requests to an instance with an unhealthy status. Once the instance has returned to a healthy state, the load balancer will continue to route requests to that instance.


Stateful vs Stateless Load Balancing

Stateful Load Balancing

A stateful load balancer is able to keep track of all current sessions using a session table. Before picking the right server to handle a request, it is able to look at a number of things using a distributed load balancing algorithm, such as the load of the different servers. Once a session is initiated and the load distribution algorithms have chosen its destination server, it sends all the upcoming packets to the server until the session comes to a close.

Stateless Load Balancing

Contrary to the process of stateful load balancing, stateless load balancing is a much simpler process. The most common method of a stateless load balancer is by making a hash of the IP address of the client down to a small number. The number is used for the balancer to decide which server to take the request. It also has the ability to pick a server entirely by random, or even go round-robin.

The hashing algorithm is the most basic form of stateless load balancing. Since one cli  can create a log of requests that will be sent to one server, hashing on source IP will

generally not provide a good distribution. However, a combination of IP and port can create a hash value as a client creates individual requests using a different source port.

Application Load Balancer

An application load balancer is one of the features of elastic load balancing and allows simpler configuration for developers to route incoming end-user traffic to applications based in the public cloud. In addition to load balancing's essential functionality, it also ensures no single server bears too much demand. As a result, it enhances user experiences, improves application responsiveness and availability, and provides protection from distributed denial-of-service (DDoS) attacks.

Load Balancing Router

A load balancing router, also known as a [failover](#) router, is designed to optimally route internet traffic across two or more broadband connections. Broadband users that are simultaneously accessing internet applications or files will be more likely to have a better experience. This becomes especially important for businesses that have a lot of employees trying to access the same tools, applications, etc.

Adaptive Load Balancing

Adaptive load balancing provides a simpler and more efficient solution to correcting an imbalance in traffic by using a feedback mechanism. In order to achieve efficient traffic distribution across the links in an aggregated Ethernet (AE) bundle, the imbalanced weights need to be corrected by adapting the bandwidth and packet stream of links.

Configuring adaptive load balancing requires interfaces with an IP address and protocol family to be configured. The membership for the AE bundle is made up of these interfaces. In order to create an AE bundle, a set of router interfaces need to be configured as aggregated Ethernet with a specific AE group identifier.

Cloud Load Balancing

[Cloud load balancing](#) is heavily involved in cloud computing to distribute workloads and compute resources. Contrary to traditional on-premises load balancing technology, cloud load balancing can help enterprises achieve high performance levels at a lower cost. Another benefit of cloud load balancing is its ability to utilize the cloud's scalability and agility to meet rerouted workload demands, which can improve overall availability. In the meantime it is hosting the distribution of demands and workload traffic residing over the Internet, it can also provide health checks for cloud applications.

Active Active vs Active Passive Load Balancing

To begin your understanding of active active vs active passive load balancing, we'll start by going over active active load balancing. Active active load balancing is essentially when there are two load balancer appliances running at the same time, processing the connections to virtual servers. Active active uses all of the power it can access to making these connections with the "real services" running. Active passive load balancing also has two load balancer appliances, however, only one is working while the other remains "passive", staying on stand-by to monitor the working appliance and implementing health checks.



Check out this webinar to learn everything you need to know about the NSX Advanced Load Balancer (Avi Networks)

[WATCH HERE](#) ➤

Are you interested in learning more
about Avi?

START YOUR FREE TRIAL TODAY

Why Avi

What We Do

Platform Overview

Platform Architecture

Products