GLOSSARY: CYBERSECURITY TERMS & DEFINITIONS

What Is a Load Balancer?

A load balancer enables distribution of network traffic dynamically across resources (on-premises or cloud) to support an application.

A load balancer is a solution that acts as a traffic proxy and distributes network or application traffic across endpoints on a number of servers. Load balancers are used to distribute capacity during peak traffic times, and to increase reliability of applications. They improve the overall performance of applications by decreasing the burden on individual services or clouds, and distribute the demand across different compute surfaces to help maintain application and network sessions.

Modern applications must process millions of sessions simultaneously and return the correct text, videos, images, and other data to each user in a fast and reliable manner. To handle such high volumes of traffic, most applications have many resource servers with duplicate data among them.

Load balancing distributes network traffic dynamically across a network of resources that support an application. A load balancer is the device or service that sits between the user and the server group and acts as an invisible facilitator, ensuring that all resource servers are used equally. A load balancer helps increase reliability and availability, even in times of high usage and demand, and ensures more uptime and a better user experience.

In some cases, it is essential that all requests from a client are sent to the same server for the duration of a session, for example when a client is putting items in a shopping cart and then completing the purchase. Maintaining the connection between client and server is known as ***session persistence***. Without session persistence, information has to be synchronized across servers and potentially fetched multiple times, creating performance inefficiencies.

Benefits of Load Balancing

Users and customers depend on near-real-time ability to find information and conduct transactions. Lag time or unreliable and inconsistent responses—even during peak demand and usage times—can turn a customer away forever. And high spikes in compute need can cause havoc to an internal server or server system if the incoming demand—or "load"—is too high to be

Ski

Chat with Codey

Advantages of using a load balancer include:

- **Application availability:** Users both internal and external need to be able to rely on application availability. If an application or function is down, lagging, or frozen, precious time is lost—and a potential source of friction is introduced that might drive a customer to a competitor.
- **Application scalability:** Imagine you run a ticketing company, and tickets for a popular performance are announced to be available at a certain date and time. There could be thousands or even more people trying to access your site to buy tickets. Without a load balancer, your site would be limited to whatever your single/first server can accommodate—which likely won't be much with that much demand. Instead, you can plan for this big spike in traffic by having a load balancer to direct requests and traffic to other available compute surfaces. And that means more customers can get their desired tickets.
- **Application security:** Load balancing also lets organizations scale their security solutions. One of the primary ways is by distributing traffic across multiple backend systems, which helps to minimize the attack surface and makes it more difficult to exhaust resources and saturate links. Load balancers can also redirect traffic to other systems if one system is vulnerable or compromised. In addition, load balancers can offer an extra layer of protection against DDoS attacks by rerouting traffic between servers if a particular server becomes vulnerable.
- **Application performance:** By doing all of the above, a load balancer boosts application performance. By increasing security, by optimizing uptime, and by enabling scalability through spikes in demand, load balancers keep your applications working as designed—and the way you, and your customers, want them to.

Load Balancing Algorithms

There are two types of load-balancing algorithms in terms of how they operate: static and dynamic. Static load balancing measures the incoming load on a server using algorithms that have performance capacity information about the existing servers in the distributed network. Dynamic load balancing can dynamically identify the amount of load that needs to be shed during runtime and which system should bear the load. It is designed for systems with high fluctuation in incoming load.

The following are some of the common types of load balancing algorithms.

Ski

- **Round robin:** This algorithm sends traffic to a list of servers in rotation using the Domain Name System (DNS). (Note: DNS load balancing can also be a dynamic solution.)
- **Threshold:** This algorithm distributes tasks based on a threshold value that is set by the administrator.
- **Random with two choices:** The "power of two" algorithm

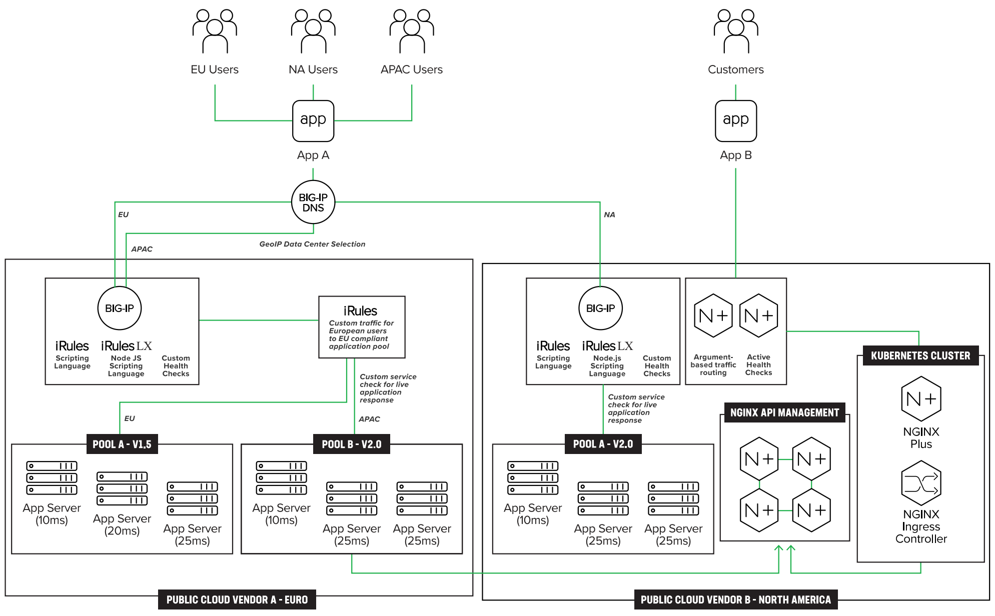
Chat with Codey

one that is selected by then applying the Least Connections algorithm or the Least Time algorithm, if so configured.

- **Least connections:** A new request is sent to the server with the fewest current connections to clients. The relative computing capacity of each server is factored into determining which one has the least connections or which is using the least amount of bandwidth or resources.
- **Least time:** In this algorithm, a request is sent to the server selected by a formula that combines the fastest response time and fewest active connections.
- **URL hash:** This algorithm generates a hash value based on the URL present in client requests. The requests are forwarded to servers based on the hash value. The load balancer caches the hashed value of the URL, so subsequent requests that use the same URL result in a cache hit and are forwarded to the same server.
- **Source IP hash:** This algorithm uses the client's source and destination IP addresses to generate a unique hash key to tie the client to a particular server. As the key can be regenerated if the session disconnects, this allows reconnection requests to get redirected to the same server used previously.
- **Consistent hashing:** This algorithm maps both clients and servers onto a ring structure, with each server assigned multiple points on the ring based on its capacity. When a client request comes in, it is hashed to a point on the ring, and is then dynamically routed clockwise to the next available server.

How Does Load Balancing Work?

Load balancing works by either statically or dynamically responding to a user request, and distributing that request to one of the backend servers capable of fulfilling the request. If one of the servers goes down, the load balancer redirects traffic to the remaining online servers.



Examples of Load Balancing

An example of static load balancing: A company hosts a website with content that is largely static. This scenario would be ideal for and consistent. The company can use two (or more) identical web servers across which the static load balancer can distribute traffic.

An example of dynamic load balancing: A company experiences surges, spikes, and drops in traffic. Some are predictable and some are not. These organizations would benefit from dynamic load balancing. Such companies might include an e-commerce retailer announcing Black Friday hours and dates; a healthcare company which has just announced it can schedule online appointments for a seasonal vaccine; a government unemployment agency which requires unemployment insurance recipients to file a weekly claim on a certain day of the week; a relief organization that may need to respond quickly online to a natural disaster. Some of these surges and spikes in traffic and demand can be planned for, but some cannot. In these scenarios, a dynamic load balancing algorithm will help ensure access to apps and resources when customers and users need them most.

Different Types of Load Balancers

Different types of load balancers with different capabilities reside in the architecture called the Open System Interconnection (OSI) model. In this model are seven layers. Network firewalls are at levels one to three (L1-physical wiring, L2-data link and L3-network). Meanwhile, load balancing happens at layers four to seven (L4-transport, L5-session, L6-presentation and L7-application). **Load balancers are generally used at Layer 4 and Layer 7**.

- **Layer 4 load balancers** direct traffic based on data from network and transport layer protocols (IP, TCP, FTP, UDP). Load balancing at the IP layer refers to a deployment where the load balancer's IP address is the one advertised to clients for a website, and therefore recorded as the destination address. When the load balancer gets the request, it changes the recorded destination IP address to that of the content server it has chosen.

- **Layer 7 load balancers distribute requests based upon data found in application layer protocols such as HTTP headers, cookies,** uniform resource identifier, SSL session ID, and HTML form data. They also enable routing decisions based on data **within the application message itself, such as the value of a specific parameter. Layer 7 adds content switching to load balancing.**

Cloud-Based Load Balancers

Cloud-based load balancers are not just traffic controllers for spikes in traffic and for optimizing server use. Cloud-native load balancers can also provide predictive analytics to help you visualize traffic bottlenecks before they happen. That in turn delivers actionable insights to help any company optimize its IT solutions.

Ski|

Chat with Codey

Application Load Balancing: As enterprises rely more and more on application performance and availability, application load balancing can help them scale, streamline operations, and save money.

Global Server Load Balancing: With users and customers around the world, companies can enhance their load availability with global server load balancing, which sends users to the nearest endpoint to them.

DNS Load Balancing: The practice of configuring a domain in the Domain Name System (DNS) so that user requests to the domain are distributed across a group of server machines is called DNS load balancing.

Network Load Balancing: Application delivery controllers (ADCs), physical or virtual appliances functioning as proxies for physical servers, manage application or network functions, and rely on a network load balancing solution to support them. ADCs also use other techniques, including caching, compression, and offloading of SSL processing, to improve the performance of web applications. In the usual configuration, the ADC sits in front of a group of web and application servers and mediates requests and responses between them and their clients, effectively making the group look like a single virtual server to the end user.

HTTP(S) Load Balancing: The technique for distributing traffic across multiple web or application server groups to optimize resource utilization is called HTTP(S) load balancing.

Internal Load Balancing: An internal load balancer is assigned to a private subnet and does not have a public IP. It typically works within a server farm.

Diameter: A diameter load balancer distributes signaling traffic across multiple servers in a network. One of the most cost-effective ways to do this is to scale the diameter control plane rather than the data transport layer. (Diameter load balancing can also be static or dynamic.)

Load Balancer Technology

There are other types of load balancer solutions, which can be used alone or in a network with cloud-native load balancers. Here are some notable types.

Hardware Load Balancer: A hardware load balancer is a physical device with a specialized operating system that can be programmed to distribute web traffic across several application servers, usually on-premises.

Software Load Balancer: A software load balancer operates like a physical load balancer, but it runs on software programs. The software keeps apps available through all kinds of traffic demands, using both static and dynamic load balancing to eliminate single points of failure.

Ski|

Chat with Codey

Virtual Load Balancer: A type of load balancer that combines hardware and software load balancers is a virtual load balancer. It uses application delivery controller software that helps to distribute network traffic load among hardware backend servers.

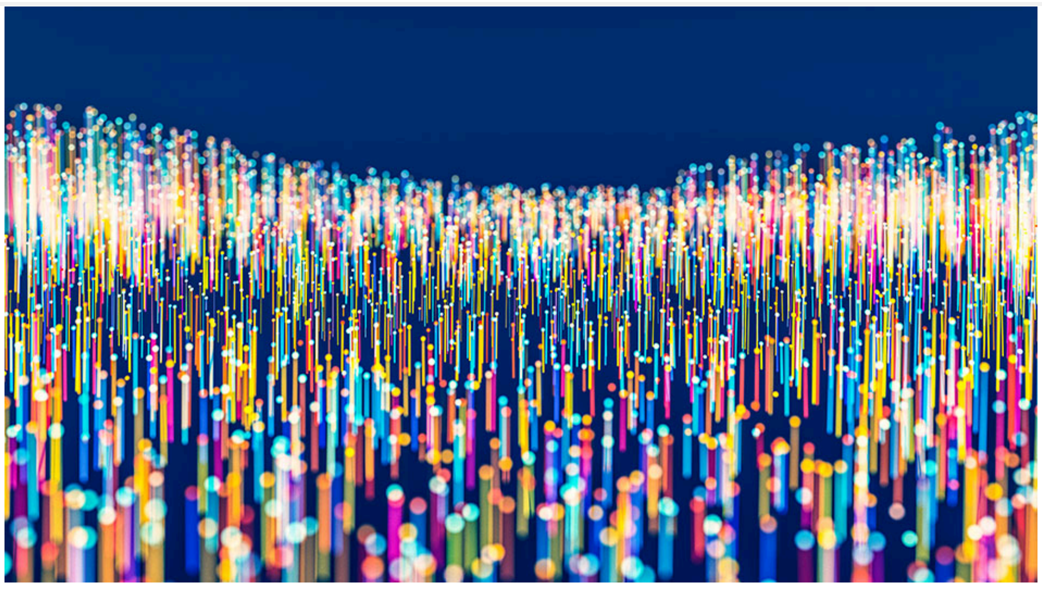
How F5 Can Help

- Finding the right load balancer for your organization's needs is critical to keeping your systems available and optimized, your data accessible, and your users and customers happy.
- F5 can address your organization's specific load balancer needs, from a static solution to an integrated, global solution that combines the strengths of hardware, software, and cloud-based load balancers. F5 has a load balancing algorithm or solution for your unique business needs.
- F5 NGINX Plus and NGINX are the best-in-class load-balancing solutions used by high-traffic websites such as Dropbox, Netflix, and Zynga. More than 350 million websites worldwide rely on NGINX Plus and NGINX Open Source to deliver their content quickly, reliably, and securely. As a software-based application delivery controller and load balancer, NGINX Plus is significantly less expensive than hardware solutions with similar capabilities. It combines web serving, load balancing, caching, media delivery, and more, making it an ideal choice for controlling the delivery of your applications.
- BIG-IP application services provide an integrated solution to manage, scale, and optimize your digital application services. And BIG-IP Local Traffic Manager (LTM) includes static and dynamic load balancing to eliminate single points of failure. F5 BIG-IP DNS takes load balancing across applications and applies it globally, ensuring that your applications are on and responding to your customer's needs.
- F5 Distributed Cloud DNS Load Balancer delivers a simple load-balancing solution with reliable disaster recovery, so your development teams can focus on helping your business innovate.
- F5 Distributed Cloud App Connect helps enable load balancing by securely connecting your apps and services across any type of environment, including the edge.
- F5 offers a comprehensive suite of load balancing solutions to keep your apps, traffic, data, and compute surface optimized.

Ski

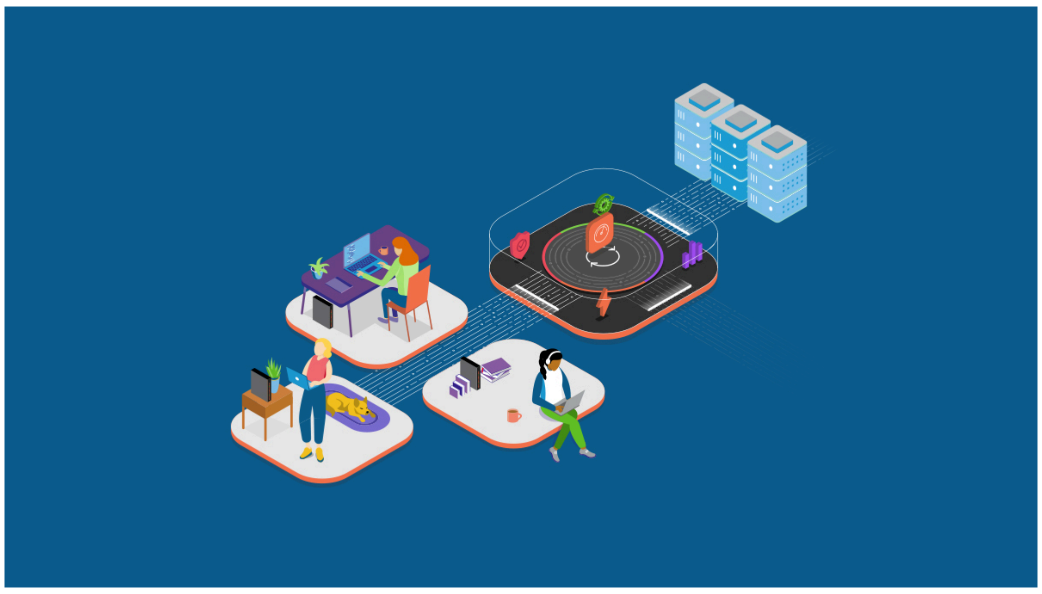
Resources

Chat with Codey



WHITE PAPER

[Load Balancing 101: Nuts and Bolts ›](#)



USE CASES

[Load Balancing Your Applications ›](#)

GLOSSARY

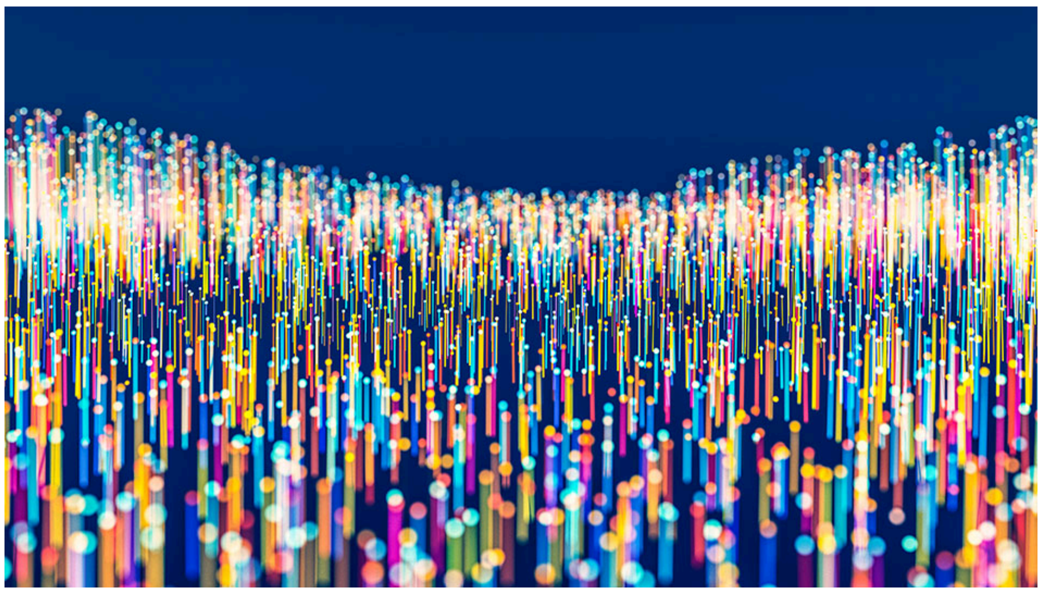
[What Is a Diameter Load Balancer? ›](#)

Ski

Chat with Codey

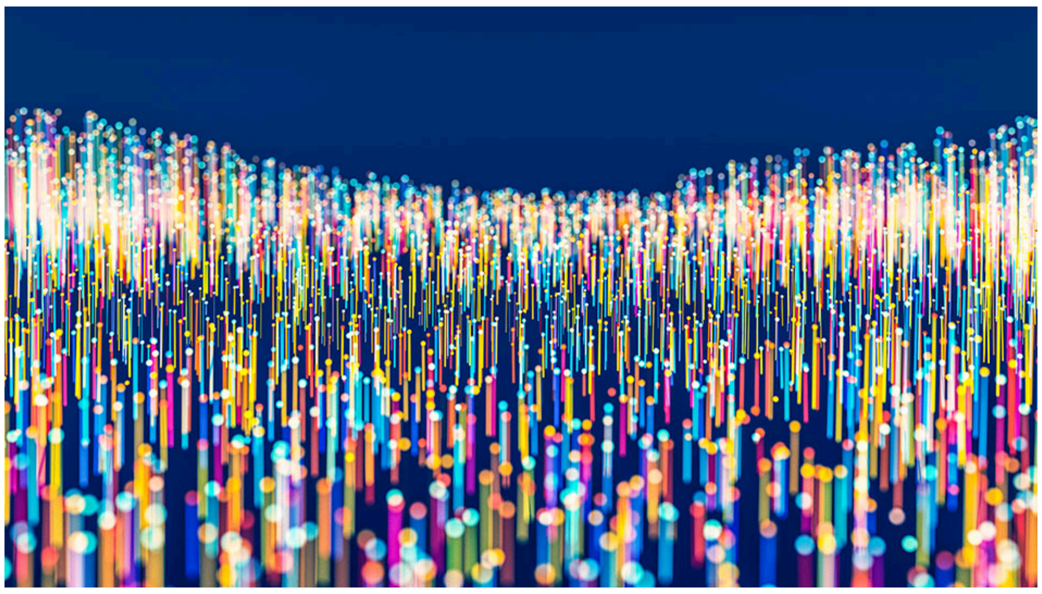
DATA SHEET

Big-IP Local Traffic Manager: Application Delivery, at Scale ›



WHITE PAPER

Load Balancing on AWS: Know Your Options ›



WHITE PAPER

Load Balancing 101: The Evolution to ADCs ›

Secure and Deliver Extraordinary Digital Experiences

F5's portfolio of automation, security, performance, and insight capabilities empowers our customers to create, secure, and operate adaptive applications that reduce costs, improve operations, and better protect users. Learn more ›

WHAT WE OFFER

RESOURCES

SUPPORT

PARTNERS

Ski

COMPANY

Chat with Codey

CONNECT WITH US

Trademarks Policies Privacy California Privacy Do Not Sell My Personal Information Cookie Preferences

Chat with Codey