Load balancing is used to improve the performance of websites, databases, and other services through the distribution of the workload across multiple servers. This also allows for websites to maintain an improved standard of reliability. Both reliability and improved performance are important in the digital society today.

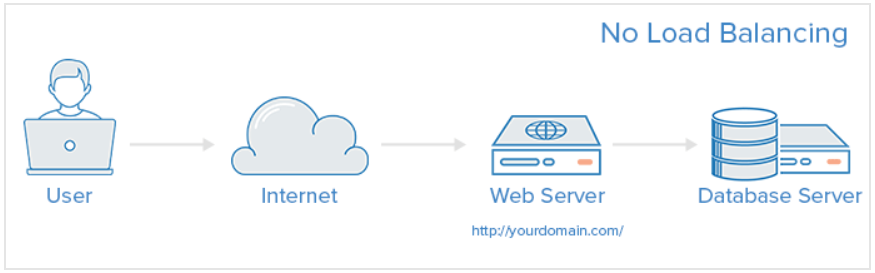
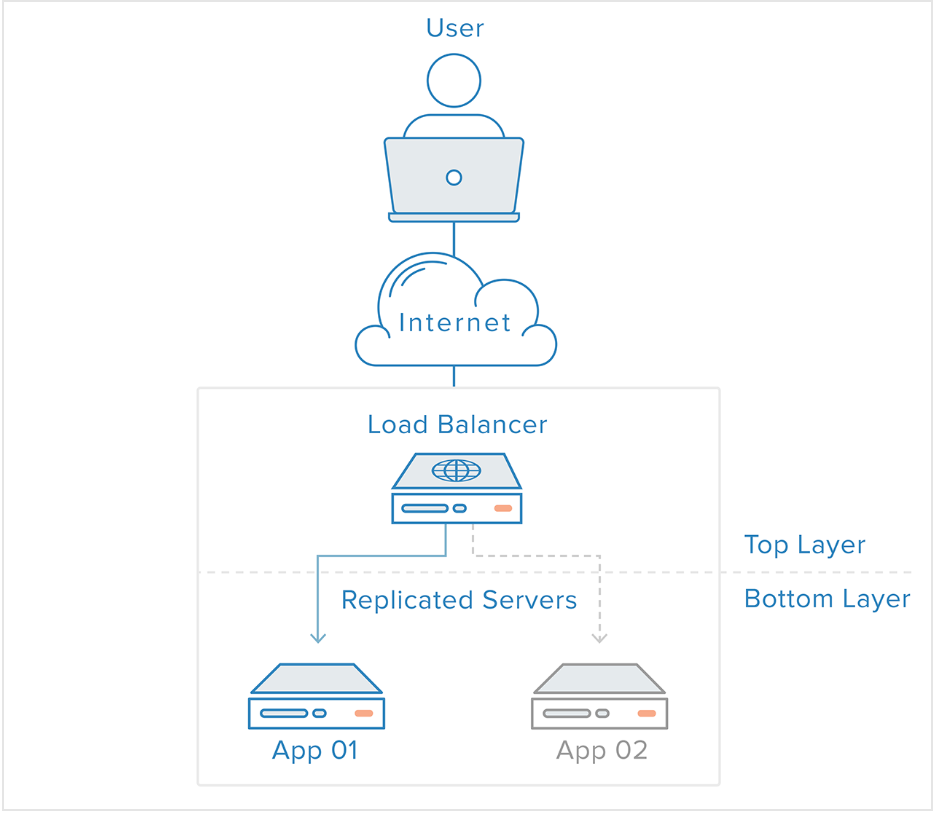
Web infrastructure that does not contain any load balancing is very simple. A user connected to the internet and connects directly to the web server (yourdomain.com) would be an example (See Figure 1 Below)(Anderson, 2017). If the single web server goes down the user will then no longer be able to access the website. Also, with this set up if many users are trying to access the server at the same time, the server may not be able to handle the sudden load and may act slowly or may be unable to connect users to the site at all. By introducing a load balancer and another web server on the backend. Usually all of the backend servers will supply the same content so that clients visiting or using the site will receive consistent content no matter what server responds to their request. In another example case (See Figure 2 Below) where a single load balancer is in place the user accesses the load balancer, the request is then forwarded to the backend server, which then responds to the user’s request (Anderson, 2017). Thus, the single failure is now the load balancer itself and the same issue on overloading the two servers can occur from the first example. The problem is solved through the addition of a second load balancer (Anderson, 2017). 

Figure (Anderson, 2017)

Figure (Anderson, 2017)

Load Balancers can handle and maintain four different types of network traffic. These include HTTP, HTTPS, TCP, and UDP. The administrators of a load balancer can generate the forwarding rules to maintain these types of traffic. For HTTP, the standard is to balance the requests by directing them based upon the standard HTP mechanisms. The load balancer then sets the headers (X-Forwarded-For, X-Forwarded-Proto, and X-Forwarded-Port) to give the backends information about the original requests (Anderson, 2017). HTTPS is similar to HTTP balancing but has a level of encryption. The encryption is handled by either SSL passthrough which keeps the encryption all the way to the backends or uses SSL termination which places the decryption on the load balancer but sends the traffic unencrypted to the back end (Anderson, 2017). With non-HTTP(S), either TCP or UDP traffic is involved and can be balanced. For example, with TCP, traffic that is going to a database cluster can be spread out across all of the servers. While with UDP not all load balancers support this type of traffic, but some have added support for balancing core internet protocols like that of DNS and syslogd, which use UDP(Anderson, 2017).

Load Balancers are tasked with the job of choosing which backend server to send a request to. There are two factors that are typically involved, responsiveness and the health of a server. Health checks are monitored because the load balancer should only forward to a healthy server. Health checks regularly attempt to connect to backend servers using the protocol and port defined in the forwarding rules, thus, ensuring that the servers are listening. If a server fails, then they are reported to be unable to serve requests and is removed from the load balancers pool. Traffic will not be forwarded to them until they respond correctly to a health check. Once the healthy pool is generated it will then use load balancing algorithms to determine where to send the requests. Some of the most common Algorithms include Round Robin, Least Connections, and Source (Anderson, 2017). Round Robin places the selected servers sequentially and will select the first server on its list for the first request and will loop once it reaches the end of its list. Least connections send the request to the server with the least connections. It is recommended for traffic with longer sessions. With the Source algorithm, the load balancer will select what server to go to based on a hash of the source IP of the request, like the client’s IP address (Anderson, 2017). Thus, each user is connected consistently to the same server. Another way to achieve his is with Sticky Sessions, in which the load balancer sets a cookie and all of the requests for that session are directed to the same physical server (Anderson, 2017). However, algorithms vary due to availability to administrators and the load balancing technology that is being used.

Load balancers can also be redundant in order to benefit the clients. When two load balancers are used they can be connected and form a cluster. They then can monitor each other’s’ health and detect failure and recovery. If the primary load balancer fails the second one can be used and the concept of floating IPs is then used to make sure that the DNS changes do not take too long. On demand IP address remapping eliminates the propagation and caching issues inherent in DNS changes by providing a static IP address that can be easily remapped when needed (Anderson, 2017). The domain name can remain associated with the same IP address, while the IP address itself is moved between servers (Anderson, 2017).

Load balancers are important to have in place so that a website can have improved performance and reliability for its users. Load balancers handle different types of traffic and use different algorithms and checks to send requests to back end servers. They can also be placed redundantly to work together to eliminate another single point of failure. Overall, load balancers are an important aspect of web applications that should be applied.

Work Cited

Anderson, Melissa. “What Is Load Balancing?” *What Is Load Balancing? | DigitalOcean*, DigitalOcean, 18 July 2017, www.digitalocean.com/community/tutorials/what-is-load-balancing.