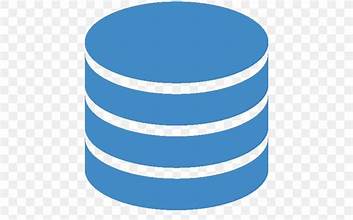
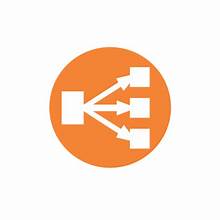
**EXPLAINING WHAT HAPPENS WHEN YOU TYPE google.com IN YOUR BROWSER AND PRESS ENTER**



1. **DNS server** resolved the domain name and provides the IP Address for www.google.com(64.233.160.0)
2. Request is sent to DNS server to provie the IP Address or **www.google.com**
3. **Firewall** uses the rule to permit/restrict access to 64.233.160.0

Response from Server to Client



DB server

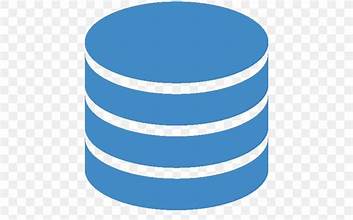


Web server

App server



Code Base



DB server



Web server

App server



Code Base

1. TCP/IP---->
2. **Load Balancer**

Uses the set algorithm to share load between the servers.

**Server 1.**

**Server 2**

When a user types the domain name (eg "www.google.com") into the browser at **Point 1**, several components and processes come into play to enable the loading of the webpage.

**DNS request:** The browser sends this domain name as a request to the DNS (Domain Name System) server to lookup this domain name ("www.google.com") and provide the IP address. THis happens at **Point 2**. The DNS server responds with the IP address associated with that domain.

**TCP/IP:** With the provided IP address, a TCP (Transmission Control Protocol) connection which ensures reliable communication between the browser and the server by breaking down the data into packets and ensuring their proper deliveryis established with the web server at that IP address. .

**Firewall**: are security measures that monitor and control network traffic. It has rules that allow or restrict access based on different criteria, such as IP addresses, ports, or protocols. If the connection is allowed, it proceeds to the next step. The TCP connection passes through these firewalls.

**HTTPS/SSL:** If the website uses HTTPS (Hypertext Transfer Protocol Secure), the browser initiates an SSL/TLS (Secure Sockets Layer/Transport Layer Security) handshake with the web server. This process establishes an encrypted connection between the browser and the server to ensure the confidentiality and integrity of the data exchanged.

**Load-balancer:** In the case of large websites like Google, multiple servers are often involved to handle the traffic efficiently. A load balancer distributes incoming requests across these servers to optimize performance and ensure scalability. It selects an appropriate server and forwards the request to it.

**Web server:** The web server, receiving the request from the load balancer, processes it. In this case, Google's web server would handle the request for the "www.google.com" webpage. It retrieves the relevant files and resources required to generate the webpage's content.

**Application server:** In complex web applications, an application server may be involved to handle dynamic content generation and business logic. However, for a simple website like Google's homepage, an application server may not be necessary, and the web server can handle the request directly.

**Database:** Depending on the content being served, the web or application server may need to retrieve data from a database. For example, in a search engine like Google, the web server may query a massive database to retrieve search results. However, the specifics of Google's infrastructure are not publicly disclosed, so the exact workings are unknown.

Once the web server (and potentially other components) has processed the request and gathered the necessary data, it generates an HTML response, which includes the webpage's content and other resources like CSS and JavaScript. This response is sent back to the browser via the established TCP connection, allowing the browser to render and display the webpage to the user.