# So, when I type www.google.com and press enter:

* first of all: this URL doesn't have protocol (prefix before domain that means set of rules to control any communication on the web)
* every protocol has layers, I'll cover TCP/IP model layers that used in this process later (application -> transport -> network -> data link or physical layer)
* the browser automatically adds https protocol if it not found, this protocol uses the same principle of http but with more secure and encryption
* https stands for: hypertext transfer protocol (with security), it's responsible for transfer web documents between server and client
* server is a physical device or super computer that stores all of website data
* every website has unique IP address (it's an octet string represents the address of this website on the server)
* the model starts with application layer having protocols like https, IMAP, SSL, DNS and more
* once I press enter, we need the IP address of google, so the browser sends query to DNS server
* DNS server is the name system of the internet, it translates domain name to its IP address, it has logs or registers to many websites, returns the IP to browser to resolve it with https (if not cached and ready)
* After finding it, https sends a request to server, secure it with SSL protocol (Transport Layer Security) to encrypt the data on traffic (like login and user information) and verify the website with a certificate to make it safer
* After that, the TCP protocol (transmission control protocol) in the transport layer encapsulates from the application layer that has done all of above, to check errors and send packets to IP protocol (internet protocol) that verifies access to internet (physical layer like switches or routers convert all to bits and to signal to transport it)
* Google is a large company, millions of traffic in google.com and big data, so one server is not perfect and will be heavy load on it, so google has many servers (html content servers, images servers, database primary servers, database replica servers, JS servers and more), to access the correct server, we need load balancer
* Load balancer is a server redirects the request or packets to the correct server, uses the best algorithm like robin, weight, least traffic, fastest road and more, the problem that it has SPOF (single point of failure) means if the load balancer corrupted, may all web system crashes, so we need a reliable and fast load balancer like HAproxy
* If any errors in server, https returns 5xx code (server error) to web browser, or errors in content, it returns 4xx, or in redirection, it returns 3xx, but when success, it returns 200
* Redirecting to the server, all of the basic process depends on the simple document content are resolved by web server, JS and database are resolved by application server like Apache or nginx
* To prevent database from any interrupts like error accrued, updating it or debugging it, there're many of database server (primary, replica) to collect data with the best method
* Response all of above, the client needs more security, may the website be dangerous and contains viruses, so we'll use firewall
* Firewall is a software on client that monitors and controls traffic, blocks unwanted traffic depending on your needs, or it doesn't allow else the traffics that you modified it
* If it safe, it allows it to transport, every response uses the same methods above, until you reach the web browser and display the web site, if you pressed reload, it'll do all of above again else the cached information like IP address
* Enjoy Google!