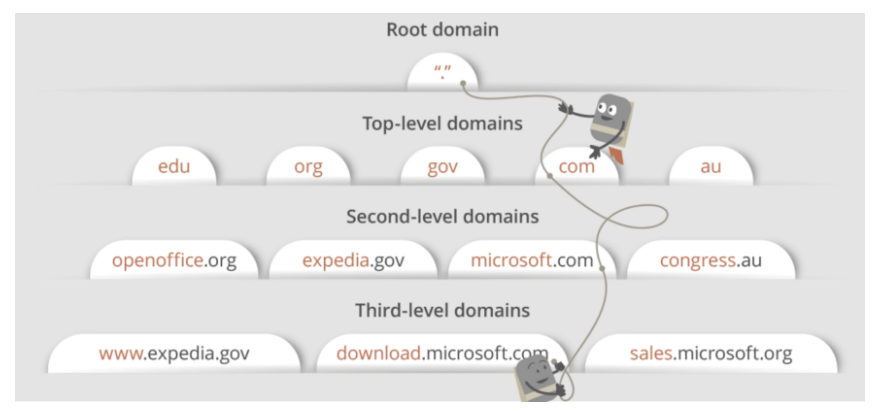
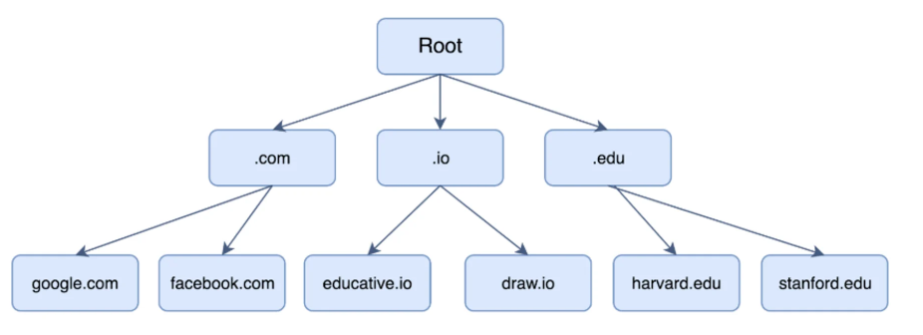
1. **DNS (Domain name system)**

The browser extracts the domain name from the URL.



## DNS lookup to find IP address

After hitting the URL, the first thing that needs to happen is to resolve IP address associated with the **domain name**.



There are 4 local cache to check :

1. The browser’s local cache is checked
2. The operating system’s cache is checked
3. The **router** is checked for the record.
4. Lastly, the query is sent to the **Resolver Server**( Internet Service Provider (ISP)) for it to check its cache.

Hence, if the record cannot be found locally, a full DNS resolution is conducted as follows:

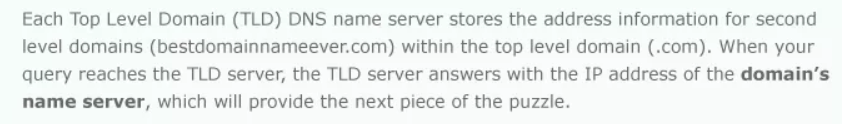
1. The first point of contact for a full resolution is a **root server**. As of the writing of this post, 1017 instances of root servers exist.
2. The root server returns the **IP address** of the relevant **top level domain server**.
3. The top level domain returns **the IP address** of the **second level domain server**.
4. The second-level domain server contains the DNS record of the server we are looking for. The **second-level domain server returns the IP address to the browser**.

This is the overview, but there are **four layers** through which this domain name query goes through. Let’s understand the steps:

* 1. After hitting the URL, the **browser cache** is checked. As browser maintains its DNS records for some amount of time for the websites you have visited earlier. Hence, firstly, DNS query runs here to find the IP address associated with the domain name.
  2. The second place where DNS query runs in **OS cache** followed by **router cache**.
  3. If in the above steps, a DNS query does not get resolved, then it takes the help of resolver server. **Resolver server (Recursive server)** is nothing but your ISP (Internet service provider). The query is sent to ISP where DNS query runs in**ISP cache.**

**Recursive servers/Resolver Servers** are the workhorses in the DNS lookup process. They often have to **make numerous DNS lookups in order to respond with the proper IP for the querying client**. To find the IP, it will recursively query **Root Server**, **TLD**-Top Level Domain Server and **Authoritative Name Server** (2nd level Domain DNS name server). These kinds of servers are typically managed by an ISP (Internet Service Provider) or specialty resolving DNS providers.

* 1. If in 3rd steps as well, no results found, then request sends to **top or root server** of the DNS hierarchy. There it never happens that it says no results found, but actually it tells, from where this information you can get. If you are searching IP address of the top level domain (.com,.net,.Gov,. org). It tells the resolver server to search **TLD server** (Top level domain).

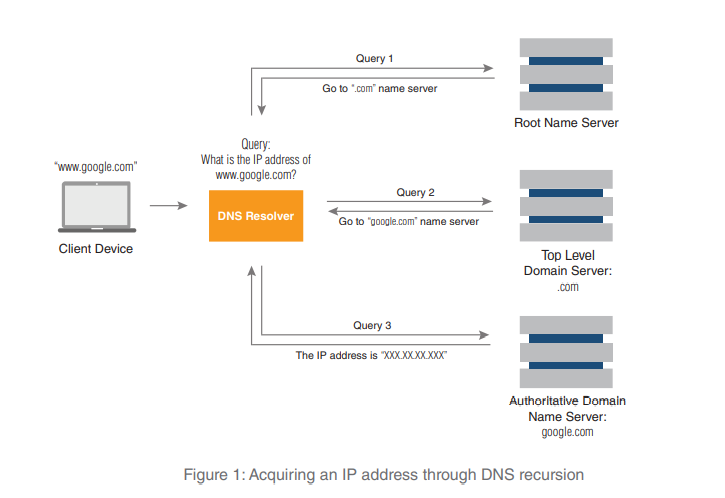


* 1. Now, resolver asks TLD server to give IP address of our domain name. **TLD** stores **address information of domain name**. It tells the resolver to ask it to **Authoritative Name server, which is second-level Domain Server**

The requests that reach Authoritative Name Server are from [Resolving name servers (resolvers)](https://www.cloudns.net/wiki/article/202) and the authoritative servers will either have the complete answer or they will pass to the name server who is responsible for it. The authoritative servers don’t cache query results.

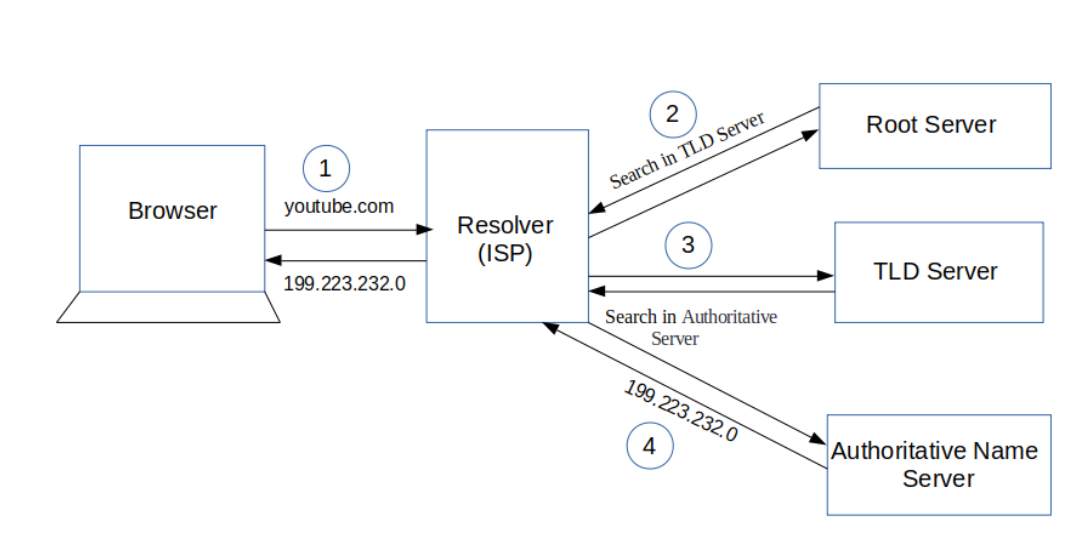
***Authoritative DNS servers (2nd level Domain Server)*** *store the “maps” of your domain names to IP addresses. This domain name to IP mapping is usually configured by system administrators.*

* 1. The authoritative name server is responsible for knowing everything about the IP address of the URL. Finally, resolver (ISP) gets the IP address associated with the domain name and sends it back to the browser.



After getting an IP address, **resolver stores it in its cache** so that next time, if the same query comes then it does not have to go to all these steps again. It can now provide IP address from their cache.

The **resolver** repeatedly obtains destination IP addresses and **updates its cache**. It is called a “**recursive DNS server**” because it uses recursion to perform the resolution process. Once the resolver obtains the IP address requested by the client device, that client can begin communicating with the intended domain.



Authoritative DNS servers are managed by or on behalf of the domain owner. Internet service providers often host the service.

Because they have complete and up-to-date information about their zones. these servers are the authoritative source for IP addresses. That fact makes authoritative DNS servers crucial to an organization’s availability on the Internet. Constant change is the nature of a network. Traffic growth, re-allocation of capacity, and the evolution of web-based services are frequent drivers. **Since each network element requires an accurate IP address**, every change requires a corresponding and timely update to the authoritative DNS server. This ups the ante for authoritative DNS servers because not only must they be reliable; they must also be well-managed and secure

**DNS how it works**

DNS system can be divided into **three tiers**. They are:

* root DNS servers
* top-level domain DNS servers
* authoritative DNS servers

There's another class of DNS Server usually called local DNS server or **Resolver server** whose IP address is specified on your operating system.

When your browser connects to a website say example.com, the browser check multiple cache and then queries your **Resolver server** to get the IP address of example.com.

* If the **Resolver server** doesn't have the A record of example.com, it will query one of the **root DNS servers.**
* The root DNS server will say: I don't have the A record but I know the **top-level domain DNS** server which is responsible for .com domains.
* Then your **Resolver server** query the top-level domain DNS server which is responsible for .com domains. The TLD DNS server will respond: I don't know either but I know which DNS server is **authoritative** for example.com.
* So your **Resolver server** queries the authoritative DNS server. Because the **actual DNS record is stored on that authoritative DNS server,** so it will give your local DNS server an answer.
* Then this query result is cached on your local DNS server but it can be outdated. When the TTL time has expired, your **Resolver server** will update the query result from the authoritative DNS server. Whenever you query a DNS record on your **Resolver server**, it returns a **non-authoritative (unofficial) answer**. If **you want an authoritative answer**, you must explicitly **specify the authoritative DNS server** when you use **nslookup** or other utilities.
* The answer you've received is essentially a cached or forwarded response from your local DNS server. Basically, a non-authoritative name server is one that does not contain the records for the zone being queried; your local DNS is likely not going to have Google's name records, for example.

You can get the name servers that are authoritative for a given domain by running **host -t ns example.com** to retrieve the NS record for example.com.

In the case of Google, we see:

**$ host -t ns google.com**

**google.com name server ns4.google.com.**

**google.com name server ns1.google.com.**

**google.com name server ns2.google.com.**

**google.com name server ns3.google.com.**

If you subsequently run your nslookup command against one of those servers, you will get the authoritative answer:

$ **nslookup www.google.com** ns1.google.com

Server: ns1.google.com

Address: 216.239.32.10#53

www.google.com canonical name = www.l.google.com.

Name: www.l.google.com

Address: 173.194.43.49

Name: www.l.google.com

Address: 173.194.43.50

Name: www.l.google.com

Address: 173.194.43.48

Name: www.l.google.com

Address: 173.194.43.52

Name: www.l.google.com

Address: 173.194.43.51

If you're using nslookup, to get the NS record type, you can run something like this in interactive mode:

$ nslookup

> set querytype=ns

> google.com

Server: 127.0.0.1

Address: 127.0.0.1#53

Non-authoritative answer:

google.com nameserver = ns3.google.com.

google.com nameserver = ns4.google.com.

google.com nameserver = ns1.google.com.

google.com nameserver = ns2.google.com.

Authoritative answers can be found from:

ns1.google.com internet address = 216.239.32.10

So, setting querytype=ns does what the above host command did.

Basically, it's what the name says it is. An authoritative answer comes from a nameserver that is considered authoritative for the domain which it's returning a record for (one of the nameservers in the list for the domain you did a lookup on), and a non-authoritative answer comes from anywhere else (a nameserver not in the list for the domain you did a lookup on).

It's basically a distinction between a nameserver that's an official nameserver for the domain you're querying, and a nameserver that isn't. Nameservers that aren't authoritative are getting their answers second (or third or fourth...) hand - just relaying the information along from somewhere else.

So, for example, If I did an nslookup of maps.google.com right now, I would get a response from one of my configured nameservers. (Either from my ISP, or my domain.) It would come back as non-authoritative because neither my ISP's nameservers, nor my own are in the list of nameservers for google.com. They aren't Google's nameservers, so they're not the authoritative source that creates the NS records.

## Distributed Denial of Service (DDoS). This is the most infamous type of DNS attack and has many sub-variants. In a DDoS attack authoritative DNS servers are overwhelmed with messages, queries, zone transfers, TCP, UDP, and other traffic. This traffic may be amplified in packet size and query volume for greater effect. If a DDoS attack is successful, the authoritative DNS server resources are consumed responding to attack traffic, thereby ignoring legitimate access to services. This amounts to a “denial of service” to valid users and their traffic. Cache poisoning. A cache poisoning attacker sends bogus IP addresses in a forged DNS response to a recursive DNS server and if the attack is successful, that server caches and maps a bogus IP address to a domain. Once the bogus address is cached, that DNS server will respond to a legitimate DNS query with the false IP address. An unsuspecting client could then transact information with the attacker rather than the intended web service. Man-in-the-middle attack (MITM). This is often a means to perform another attack, such as DNS spoofing. A MITM attack uses a machine that compromises the network to intercept traffic. It then spoofs DNS transactions and delivers counterfeit IP addresses to clients, similar to cache poisoning. Here again unsuspecting traffic is routed to a host controlled by the attacker rather than the intended service.

## Berkeley Internet Name Domain (BIND) is the most popular Domain Name System (DNS) server

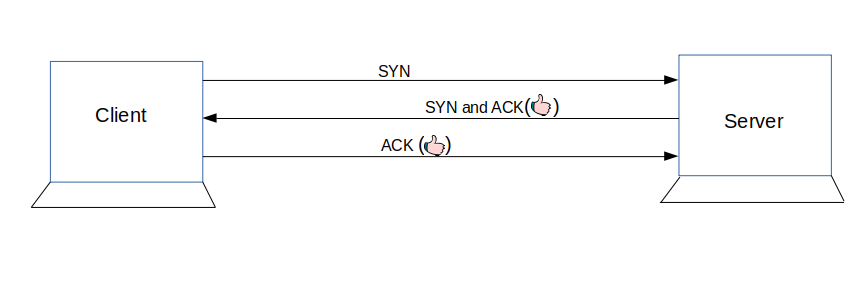
## DNS Zone

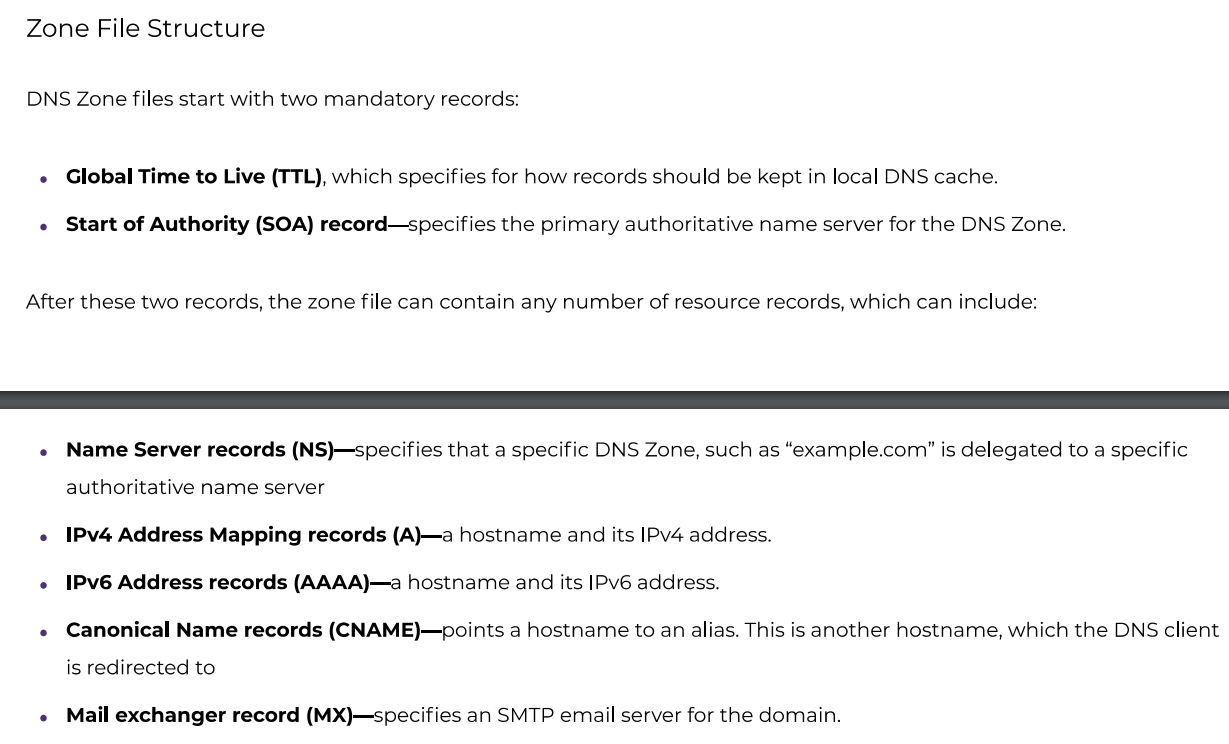
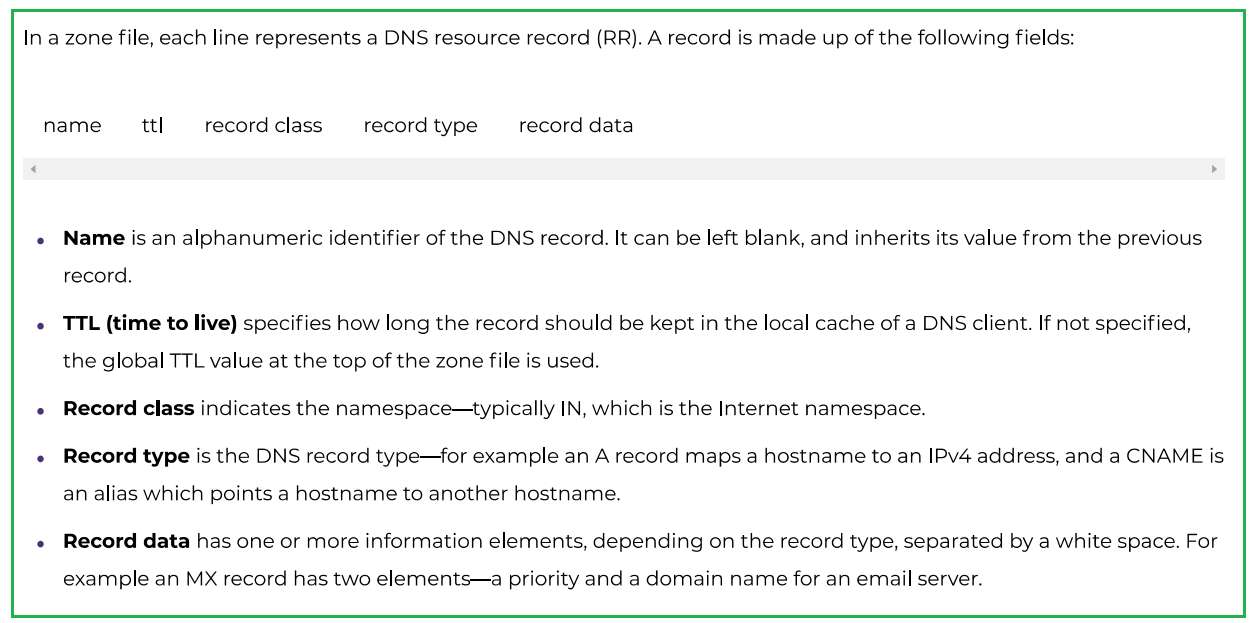
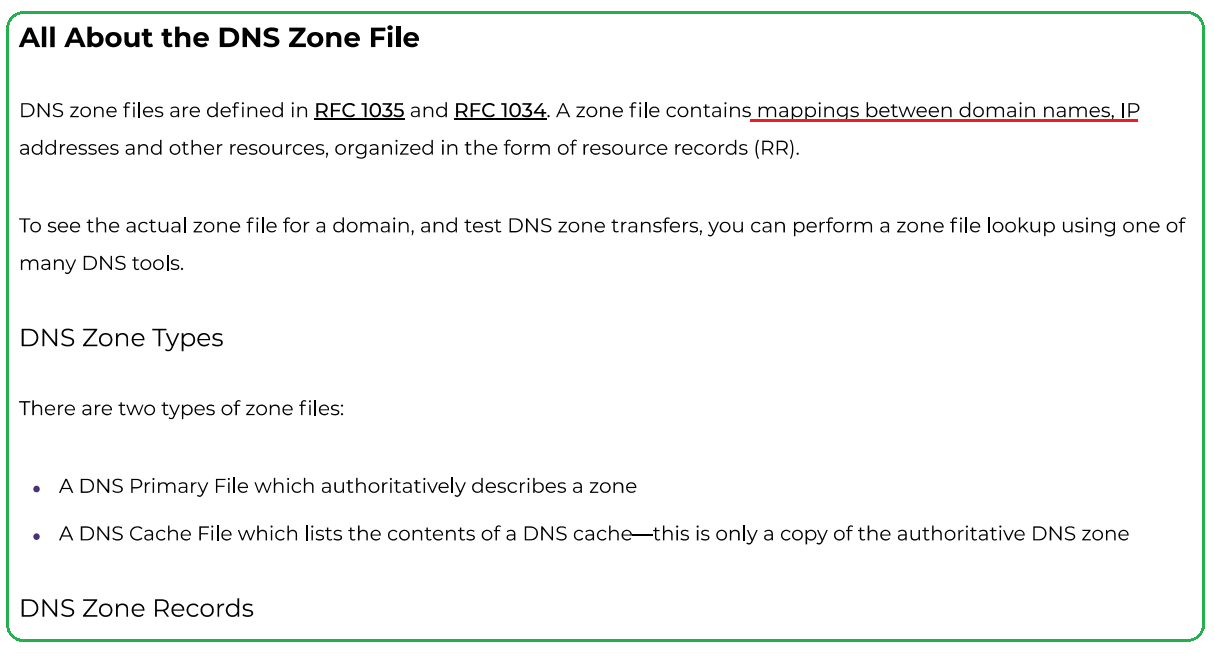
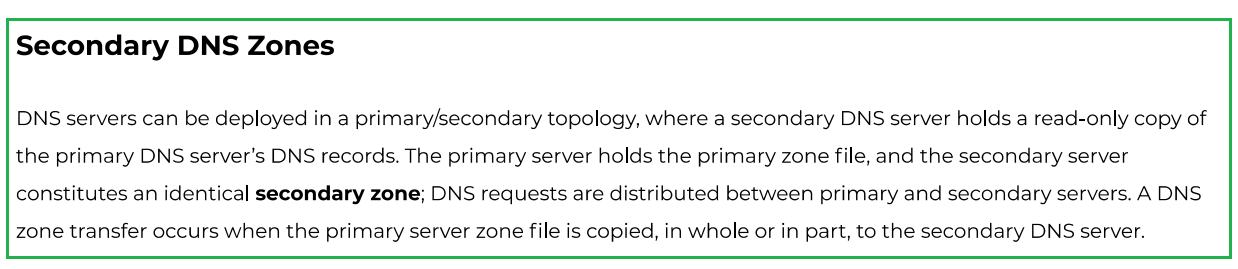
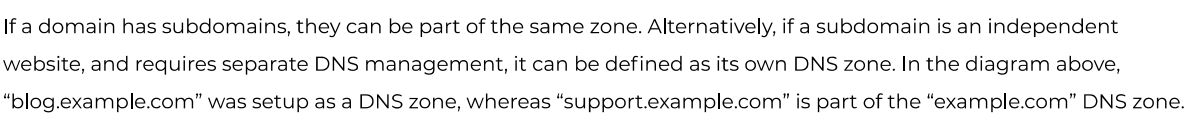
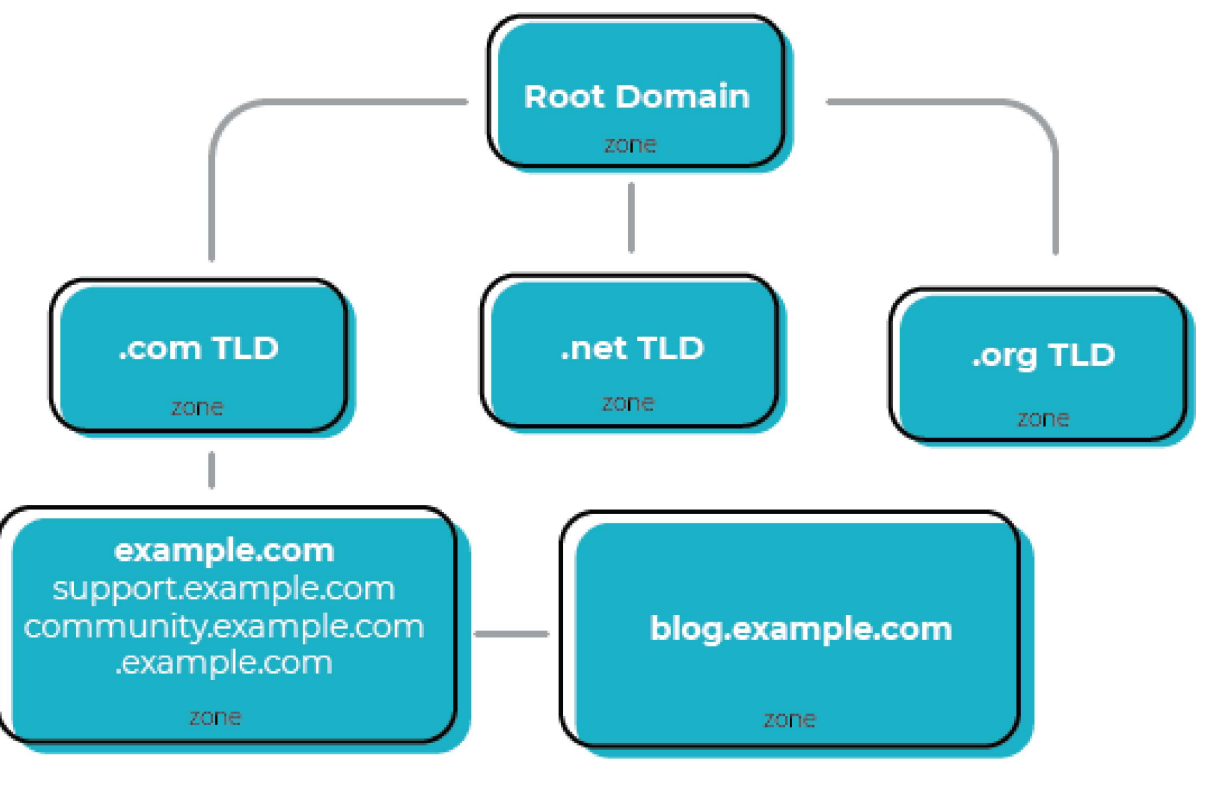
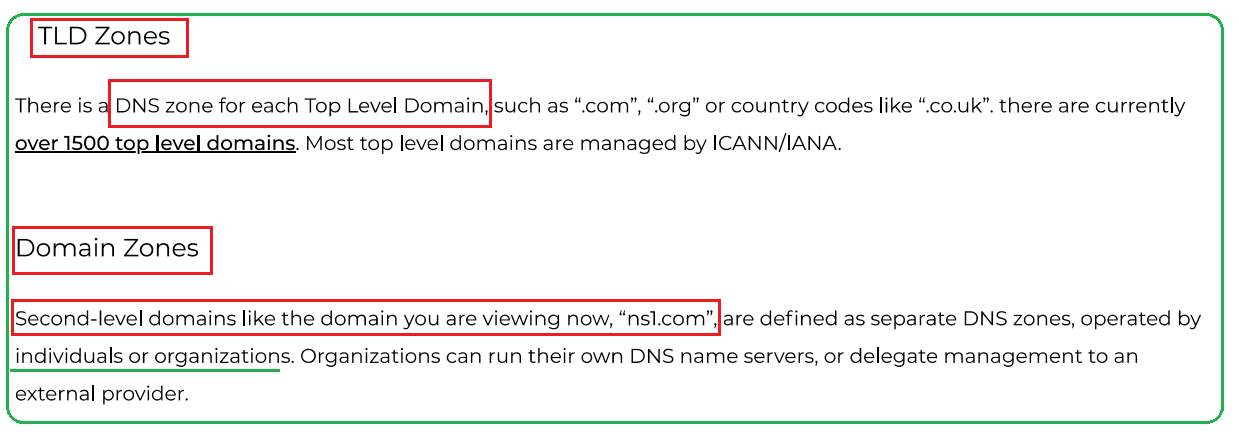
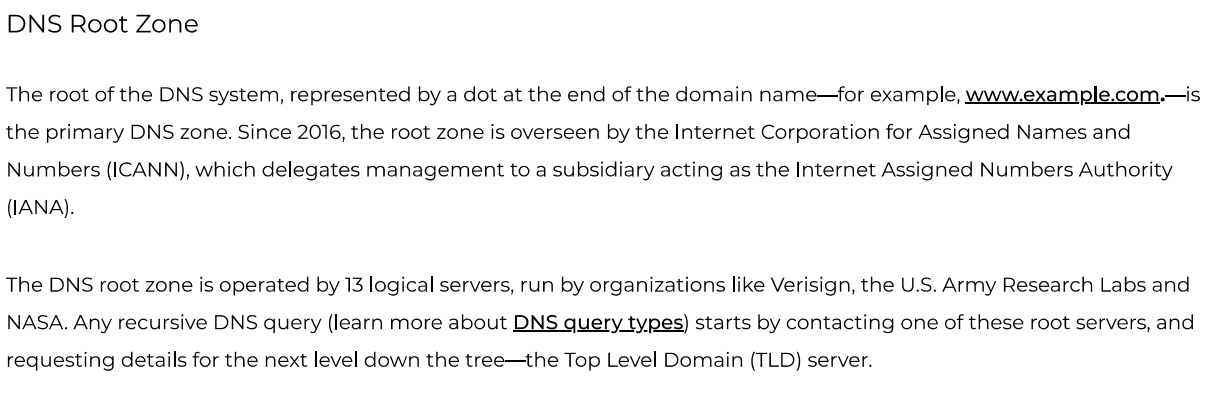
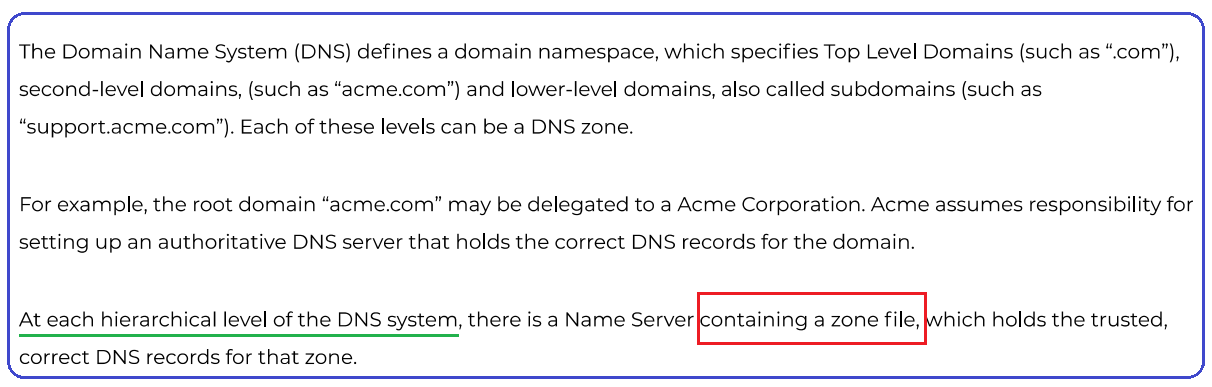
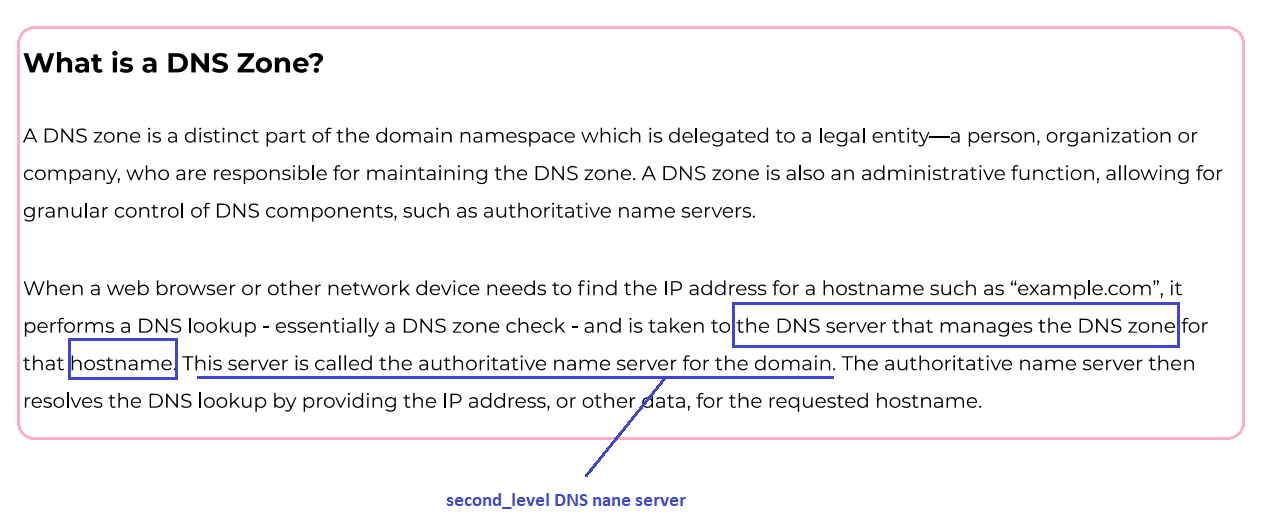
## TCP connection initiates with the server by Browser

Once the **IP address** of the computer (where your website information is there) is**found**, it **initiates connection** with it. To communicate over the network,**internet protocol** is followed. **TCP/IP** is most common protocol. A connection is built between two using a process called **‘TCP 3-way handshake’**. Let’s understand the process in brief:

1. A client computer sends a **SYN** (synchronize) **message** means, whether second computer is open for new connection or not.
2. Then**another computer**, if open for new connection, it sends **acknowledge ACK** with SYN message as well.
3. After this, **first computer** receives its message and acknowledge by **sending** an**ACK message.**

To better  understand, look below diagram.





**How to Flush DNS Cache in macOS, Windows, & Linux**

<https://phoenixnap.com/kb/how-to-flush-dns-cache>

