

What is DNS? | How DNS works

DNS lets users connect to websites using domain names instead of IP addresses. Learn how DNS works.

What is DNS?

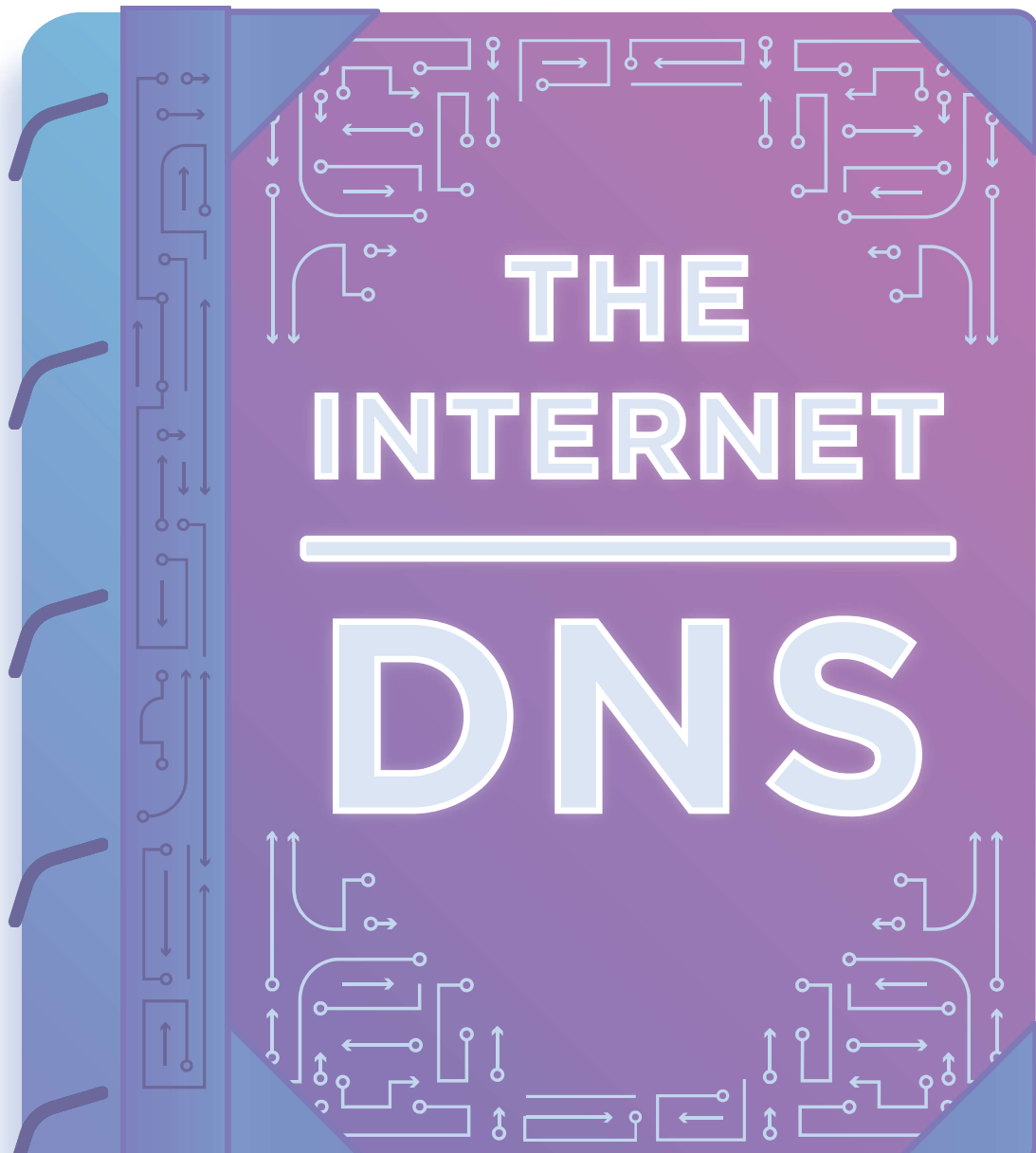


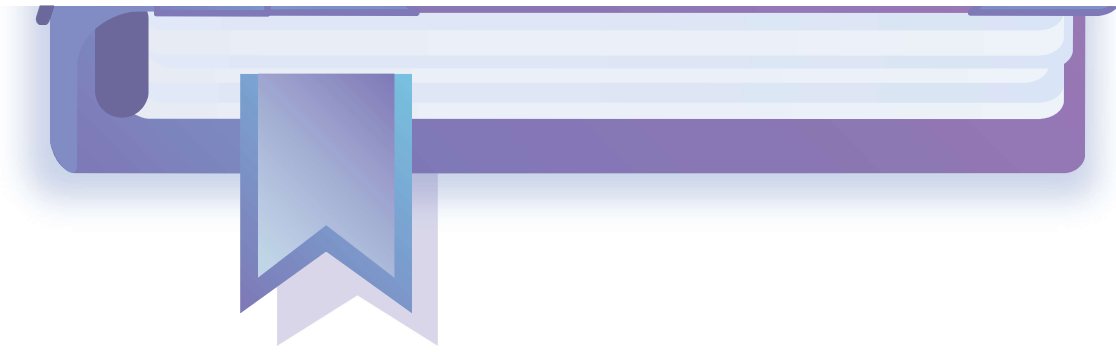
[Copy article link](#) 

What is DNS?

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through [domain names](#), like nytimes.com or espn.com. Web browsers interact through [Internet Protocol \(IP\)](#) addresses. DNS translates domain names to [IP addresses](#) so browsers can load Internet resources.

Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).





How does DNS work?

The process of DNS resolution involves converting a hostname (such as `www.example.com`) into a computer-friendly IP address (such as `192.168.1.1`). An IP address is given to each device on the Internet, and that address is necessary to find the appropriate Internet device - like a street address is used to find a particular home. When a user wants to load a webpage, a translation must occur between what a user types into their web browser (`example.com`) and the machine-friendly address necessary to locate the `example.com` webpage.

In order to understand the process behind the DNS resolution, it's important to learn about the different hardware components a DNS query must pass between. For the web browser, the DNS lookup occurs "behind the scenes" and requires no interaction from the user's computer apart from the initial request.

There are 4 DNS servers involved in loading a webpage:

- **DNS recursor** - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically the recursor is then responsible for making additional requests in order to satisfy the client's DNS query.
- **Root nameserver** - The [root server](#) is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.

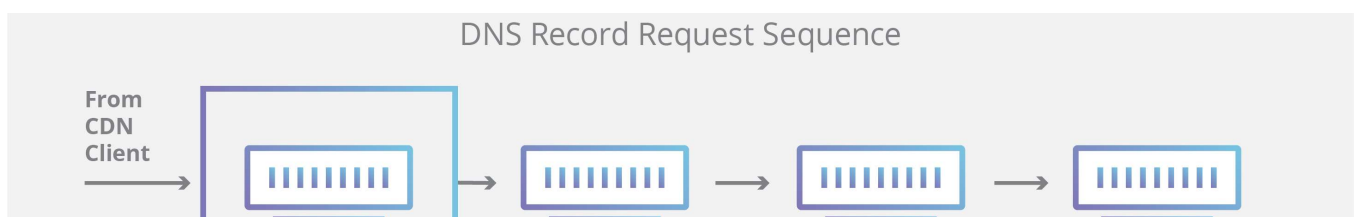
- **TLD nameserver** - The top level domain server (**TLD**) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is "com").
- **Authoritative nameserver** - This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

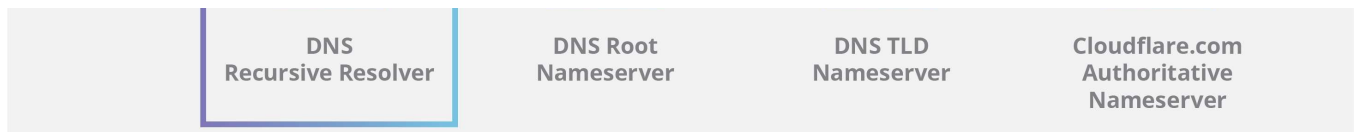
What's the difference between an authoritative DNS server and a recursive DNS resolver?

Both concepts refer to servers (groups of servers) that are integral to the DNS infrastructure, but each performs a different role and lives in different locations inside the pipeline of a DNS query. One way to think about the difference is the **recursive** resolver is at the beginning of the DNS query and the authoritative nameserver is at the end.

Recursive DNS resolver

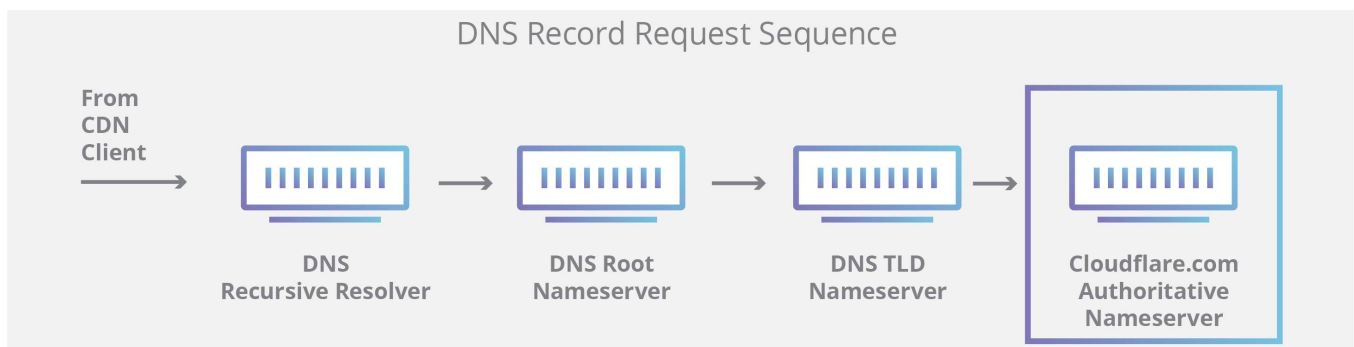
The recursive resolver is the computer that responds to a recursive request from a client and takes the time to track down the **DNS record**. It does this by making a series of requests until it reaches the authoritative DNS nameserver for the requested record (or times out or returns an error if no record is found). Luckily, recursive DNS resolvers do not always need to make multiple requests in order to track down the records needed to respond to a client; **caching** is a data persistence process that helps short-circuit the necessary requests by serving the requested resource record earlier in the DNS lookup.



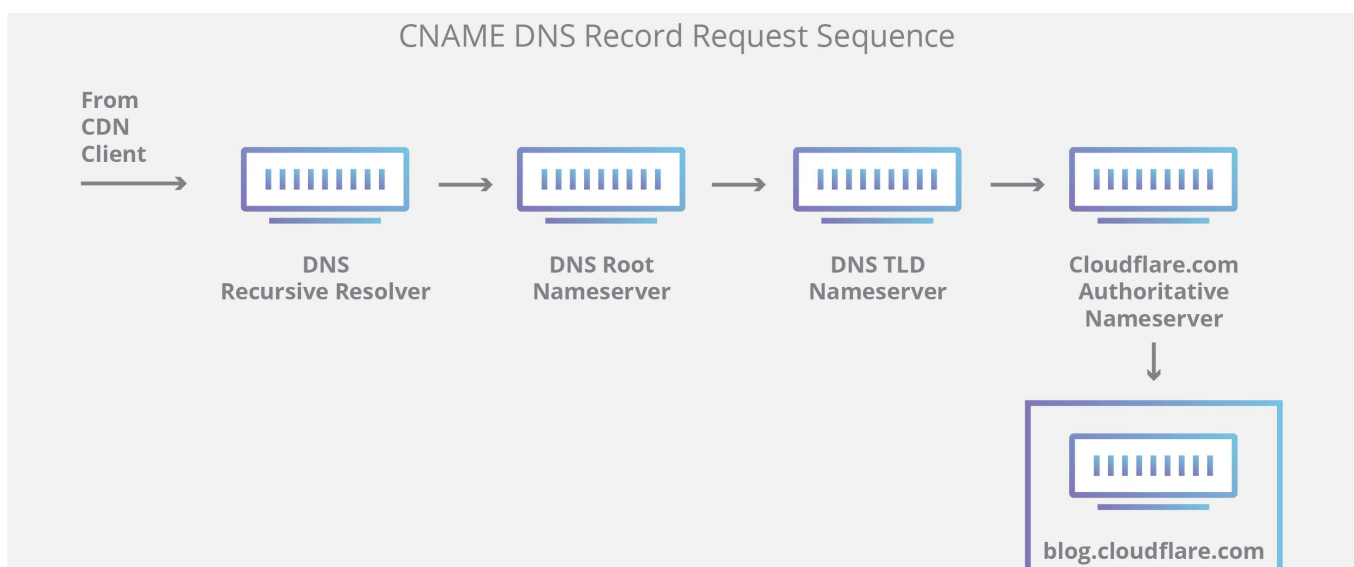


Authoritative DNS server

Put simply, an authoritative DNS server is a server that actually holds, and is responsible for, DNS resource records. This is the server at the bottom of the DNS lookup chain that will respond with the queried resource record, ultimately allowing the web browser making the request to reach the IP address needed to access a website or other web resources. An authoritative nameserver can satisfy queries from its own data without needing to query another source, as it is the final source of truth for certain DNS records.



It's worth mentioning that in instances where the query is for a subdomain such as `foo.example.com` or `blog.cloudflare.com`, an additional nameserver will be added to the sequence after the authoritative nameserver, which is responsible for storing the subdomain's [CNAME record](#).



There is a key difference between many DNS services and the one that Cloudflare provides. Different DNS recursive resolvers such as Google DNS, OpenDNS, and providers like Comcast all maintain data center installations of DNS recursive resolvers. These resolvers allow for quick and easy queries through optimized clusters of DNS-optimized computer systems, but they are fundamentally different than the nameservers hosted by Cloudflare.

Cloudflare maintains infrastructure-level nameservers that are integral to the functioning of the Internet. One key example is the [f-root server network](#) which Cloudflare is partially responsible for hosting. The F-root is one of the root level DNS nameserver infrastructure components

responsible for the billions of Internet requests per day. Our [Anycast network](#) puts us in a unique position to handle large volumes of DNS traffic without service interruption.

What are the steps in a DNS lookup?

For most situations, DNS is concerned with a domain name being translated into the appropriate IP address. To learn how this process works, it helps to follow the path of a DNS lookup as it travels from a web browser, through the DNS lookup process, and back again. Let's take a look at the steps.

Note: Often DNS lookup information will be cached either locally inside the querying computer or remotely in the DNS infrastructure. There are typically 8 steps in a DNS lookup. When DNS information is cached, steps are skipped from the DNS lookup process which makes it quicker. The example below outlines all 8 steps when nothing is cached.

The 8 steps in a DNS lookup:

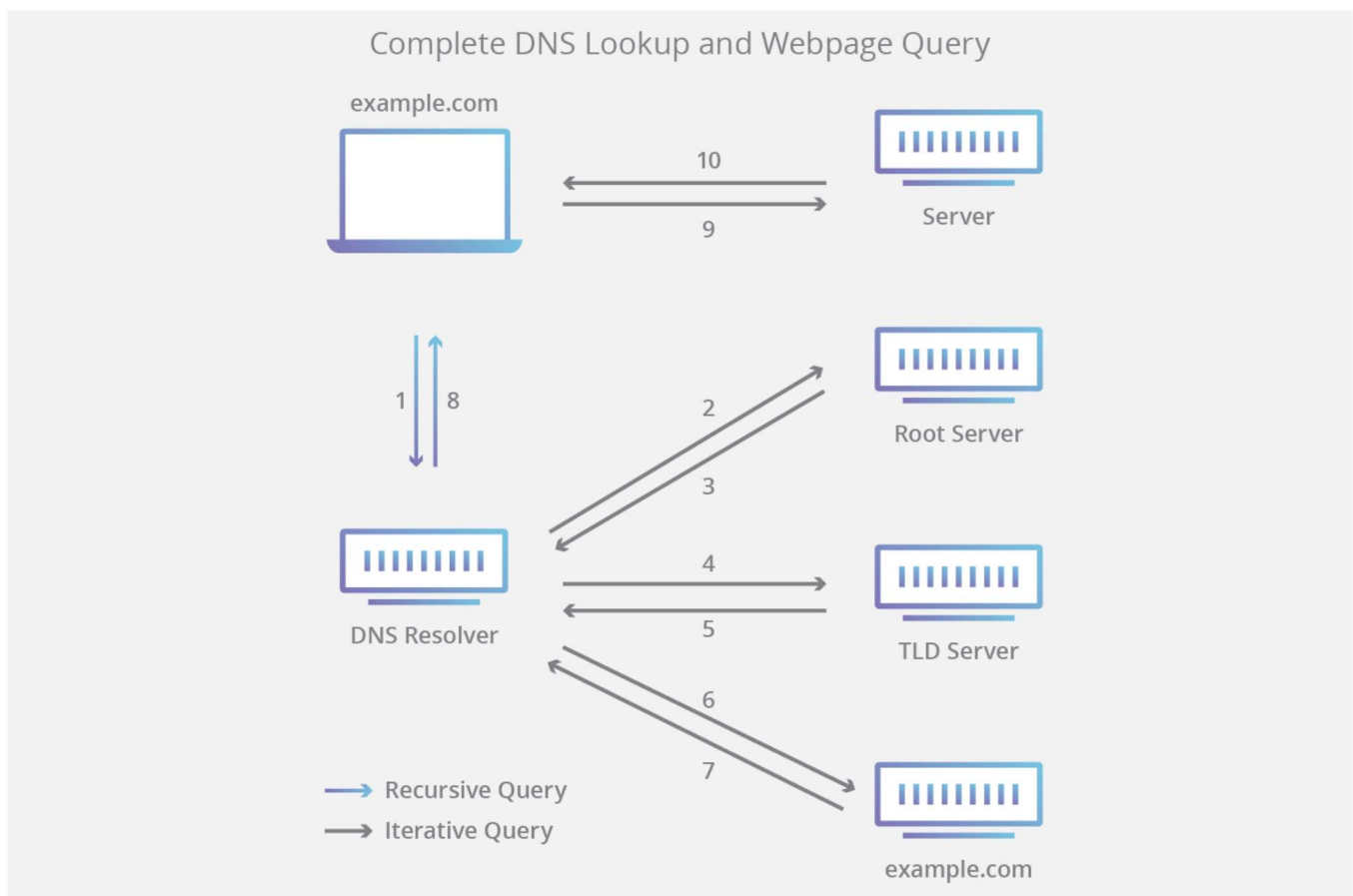
1. A user types 'example.com' into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
2. The resolver then queries a DNS root nameserver (.).
3. The root server then responds to the resolver with the address of a Top Level Domain (TLD)

DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.

4. The resolver then makes a request to the .com TLD.
5. The TLD server then responds with the IP address of the domain's nameserver, example.com.
6. Lastly, the recursive resolver sends a query to the domain's nameserver.
7. The IP address for example.com is then returned to the resolver from the nameserver.
8. The DNS resolver then responds to the web browser with the IP address of the domain requested initially.

Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:

9. The browser makes a [HTTP](#) request to the IP address.
10. The server at that IP returns the webpage to be rendered in the browser (step 10).



What is a DNS resolver?



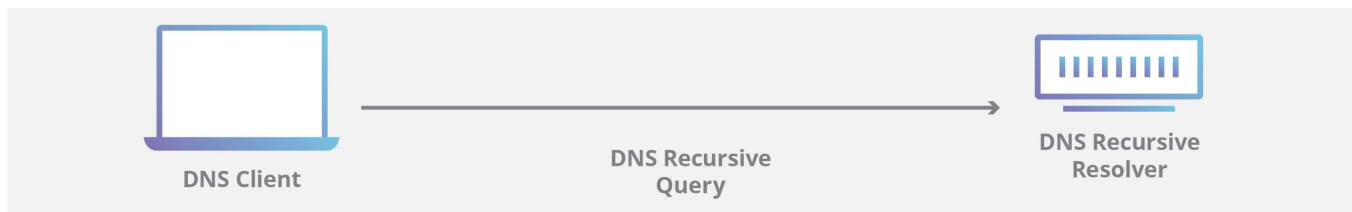
Log In



leads to a URL being translated into the necessary IP address.

Note: A typical uncached DNS lookup will involve both recursive and iterative queries.

It's important to differentiate between a [recursive DNS](#) query and a recursive DNS resolver. The query refers to the request made to a DNS resolver requiring the resolution of the query. A DNS recursive resolver is the computer that accepts a recursive query and processes the response by making the necessary requests.



What are the types of DNS queries?

In a typical DNS lookup three types of queries occur. By using a combination of these queries, an optimized process for DNS resolution can result in a reduction of distance traveled. In an ideal situation cached record data will be available, allowing a DNS name server to return a non-recursive query.

3 types of DNS queries:

1. **Recursive query** - In a recursive query, a DNS client requires that a DNS server (typically a DNS recursive resolver) will respond to the client with either the requested resource record or an error message if the resolver can't find the record.
2. **Iterative query** - in this situation the DNS client will allow a DNS server to return the best answer it can. If the queried DNS server does not have a match for the query name, it will return a referral to a DNS server authoritative for a lower level of the domain namespace. The DNS client will then make a query to the referral address. This process continues with

additional DNS servers down the query chain until either an error or timeout occurs.

3. **Non-recursive query** - typically this will occur when a DNS resolver client queries a DNS server for a record that it has access to either because it's authoritative for the record or the record exists inside of its cache. Typically, a DNS server will cache DNS records to prevent additional bandwidth consumption and load on upstream servers.

What is DNS caching? Where does DNS caching occur?

The purpose of caching is to temporarily stored data in a location that results in improvements in performance and reliability for data requests. DNS caching involves storing data closer to the requesting client so that the DNS query can be resolved earlier and additional queries further down the DNS lookup chain can be avoided, thereby improving load times and reducing bandwidth/CPU consumption. DNS data can be cached in a variety of locations, each of which will store DNS records for a set amount of time determined by a [time-to-live \(TTL\)](#).

Browser DNS caching

Modern web browsers are designed by default to cache DNS records for a set amount of time. The purpose here is obvious; the closer the DNS caching occurs to the web browser, the fewer processing steps must be taken in order to check the cache and make the correct requests to an IP address. When a request is made for a DNS record, the browser cache is the first location checked for the requested record.

In Chrome, you can see the status of your DNS cache by going to <chrome://net-internals/#dns>.

Operating system (OS) level DNS caching

The operating system level DNS resolver is the second and last local stop before a DNS query leaves your machine. The process inside your operating system that is designed to handle this query is commonly called a "stub resolver" or DNS client. When a stub resolver gets a request

from an application, it first checks its own cache to see if it has the record. If it does not, it then sends a DNS query (with a recursive flag set), outside the local network to a DNS recursive resolver inside the Internet service provider (ISP).

When the recursive resolver inside the ISP receives a DNS query, like all previous steps, it will also check to see if the requested host-to-IP-address translation is already stored inside its local persistence layer.

The recursive resolver also has additional functionality depending on the types of records it has in its cache:

1. If the resolver does not have the [A records](#), but does have the [NS records](#) for the authoritative nameservers, it will query those name servers directly, bypassing several steps in the DNS query. This shortcut prevents lookups from the root and .com nameservers (in our search for example.com) and helps the resolution of the DNS query occur more quickly.
2. If the resolver does not have the NS records, it will send a query to the TLD servers (.com in our case), skipping the root server.
3. In the unlikely event that the resolver does not have records pointing to the TLD servers, it will then query the root servers. This event typically occurs after a DNS cache has been purged.

Learn about what differentiates [Cloudflare DNS](#) from other DNS providers.

RELATED CONTENT

DNS security

DNS Zone

DNS Root Server

Primary vs. secondary DNS

Reverse DNS

Sales

[Enterprise Sales](#)

[Enterprise Sales](#)

[Become a Partner](#)

[Contact Sales:](#)

[8003014062](#)

[About DNS](#)

[DNS Servers](#)

[DNS Records](#)

[DNS Glossary](#)

[Learning Center Navigation](#)

