D.N.S
(Domain Name System)

[What, How and Where]

What is DNS? *(Normal and in terms of Networking)*

It is like the **phonebook of the internet**.

- It translates **human-readable names** like www.google.com or any Other URL
  → into **machine-readable IP addresses** like 142.250.64.100

So instead of remembering numbers (IP addresses), you just type a name — DNS finds the right number for you.

Now Technical Definition:

(DNS) is a **distributed hierarchical system** that maps:

- **Domain names → IP addresses** (via A/AAAA records)
- **IP addresses → Domain names** (via PTR records)

Also handles things like:

- **Mail routing** (MX records)
- **Aliases** (CNAME records)
- **Service discovery** (SRV records)
- **Text/config info** (TXT records)

****We Will Study about records in later in this Guide****

Some Popular DNS Providers:

Provider	Primary DNS	Secondary DNS	Features
Google DNS	8.8.8.8	8.8.4.4	Fast, reliable, minimal logging
Cloudflare	1.1.1.1	1.0.0.1	Privacy-focused, DNS over HTTPS (DoH), DoT
OpenDNS	208.67.222.222	208.67.220.220	Parental control, phishing protection
Quad9	9.9.9.9	149.112.112.112	Security and malware filtering
CleanBrowsing	185.228.168.9	185.228.169.9	Family-safe DNS filtering
Comodo Secure DNS	8.26.56.26	8.20.247.20	Malware and phishing protection

Servers Role in DNS Operation

1-Recursive Resolver

- First server your device talks to.
- **Does all the work** of finding the answer by querying other DNS servers on your behalf.

Meaning:

Scenario: You type www.abhishekchaudhary.com into your web browser.

Your Device Talks to the Recursive Resolver First:

- Your computer doesn't know the IP address for www. abhishekchaudhary.com.
- It's configured to use a specific **Recursive Resolver** (let's say its IP address is 8.8.8.8, which is Google's Public DNS).
- Your computer sends a query to 8.8.8.8 asking, "What's the IP address for www. abhishekchaudhary.com?"

2-Root DNS Servers

- Top of the DNS hierarchy.
- They don’t know domain IPs, but they tell you where to find TLD servers (like .com, .org).

Meaning:
Continuing above example

You typed www.abhishekchaudhary.com into your web browser, and your Recursive Resolver (let's say 8.8.8.8) has received your query.

1. **Your Device Talks to the Recursive Resolver First (as before):**
   - Your computer asks 8.8.8.8, "What's the IP address for www.abhishekchaudhary.com?"
2. **The Recursive Resolver Starts Its Work – And Immediately Consults a Root DNS Server:**
   - Recursive Resolver (8.8.8.8) receives your query.
   - It doesn't know the answer directly. It’s very first step in its iterative query process is to contact a Root DNS Server.
   - **Role of the Root DNS Server in Action:**
     - 8.8.8.8 asks one of the Root DNS Servers (e.g., A.ROOT-SERVERS.NET, whose IP address is hardcoded into almost all recursive resolvers): "I need to resolve www.abhishekchaudhary.com. Can you tell me where to find the authoritative information for this?"
     - The Root DNS Server's Response (Crucial Role): The Root DNS Server doesn't know the IP address for www.abhishekchaudhary.com itself. Its job is to point the

recursive resolver to the next level in the hierarchy. It looks at the **.com part** of www.abhishekchaudhary.com and replies: "I don't know the IP for www.abhishekchaudhary.com, but I can tell you that the servers responsible for **the .com** Top-Level Domain (TLD) are located at these IP addresses (e.g., 192.0.2.10, 192.0.2.11, etc. - these are the TLD Name Servers for .com).

3. TLD DNS Servers (Top-Level Domain Servers)

- Handle extensions like .com, .net, .org, .in, etc.
- They tell you which **authoritative name server** holds records for your domain.

Meaning:
Continuing above example

The Recursive Resolver Now Consults a TLD DNS Server (This is where the TLD's role comes in):

- Recursive Resolver (8.8.8.8) now takes the information from the Root Server. It chooses one of the **.com** TLD DNS Servers (let's say 192.0.2.10).
- Role of the TLD DNS Server in Action:
  - 8.8.8.8 sends a query to 192.0.2.10 (a .com TLD DNS Server): "I need to find www.abhishekchaudhary.com Can you tell me which servers are authoritative for abhishekchaudhary.com?"
  - The TLD DNS Server's Response (Crucial Role): The **.com** TLD DNS Server doesn't know the IP address for www.abhishekchaudhary.com itself. Its job is to manage all the domains ending in **.com**. It looks at abhishekchaudhary.com (the second-level domain) and knows which specific servers are responsible for holding the records for *that particular domain*.
  - It replies: "I don't have the IP for www.abhishekchaudhary.com, but the Authoritative Name Servers for the abhishekchaudhary.com domain are ns1.abhishekchaudhary.com (with IP 203.0.113.5) and ns2.abhishekchaudhary.com (with IP 203.0.113.6)."

4. The Recursive Resolver Continues its journey (now knowing where to find the authoritative servers for abhishekchaudhary.com):

- Armed with the IP addresses of the authoritative name servers for abhishekchaudhary.com, the Recursive Resolver (8.8.8.8) will then proceed to query one of those authoritative servers to get the final IP address for www.abhishekchaudhary.com.

4- Authoritative DNS Servers

- The final and actual source of truth.
- They store DNS records (like A, AAAA, CNAME, MX, etc.) for a domain.

Meaning:
Continuing above example

The Recursive Resolver Now Consults an Authoritative DNS Server ***(This is where the final answer is found):***

- Recursive Resolver (8.8.8.8) now takes the information from the TLD Server. It chooses one of the authoritative DNS servers for example.com (let's say 203.0.113.5, which is ns1.abhishekchaudhary.com).
- Role of the Authoritative DNS Server in Action:
  - 8.8.8.8 sends a query directly to 203.0.113.5 (the Authoritative DNS Server for abhishekchaudhary.com): "What is the IP address for www.abhishekchaudhary.com?"
  - The Authoritative DNS Server's Response **(The Final Truth):** This server is the "source of truth" for all records within the abhishekchaudhary.com domain. It has a comprehensive list of all DNS records for abhishekchaudhary.com, including www.abhishekchaudhary.com. It finds the specific record.
  - It replies: "The IP address for www.abhishekchaudhary.com is 192.0.2.1." *(This is an "A" record, mapping a hostname to an IPv4 address).*

5. The Recursive Resolver Delivers the Answer to Your Device:

- The Recursive Resolver (8.8.8.8) now has the definitive IP address: 192.0.2.1.
- It sends this IP address back to your computer/laptop.
- Your computer then uses 192.0.2.1 to establish a connection with the web server hosting www.abhishekchaudhary.com and load the website content.

5- Local DNS Cache

- Before any network query, your browser or OS may cache DNS answers to improve speed.
- Not a DNS server per se, but it can answer queries without any external request.

Meaning:
Continuing above example

You've previously visited www.abhishekchaudhary.com. Now, you want to visit it again.

1. **Your Device Initiates a Query (and Checks its Local Cache First):**
   - You type www.abhishekchaudhary.com into your web browser (or click a link to it).
   - Role of Local DNS Cache in Action:
     - **Browser Cache:** Your web browser (e.g., Chrome, Firefox) first checks its own internal DNS cache. If it recently resolved www.abhishekchaudhary.com and the TTL (Time-To-Live) for that entry hasn't expired, it will find the IP address (192.0.2.1) right there.
     - Operating System (OS) Cache: If the browser cache doesn't have it, or if the browser is configured to defer to the OS, your operating system (e.g., Windows, macOS, Linux) checks its own DNS cache. Again, if the IP address for www.abhishekchaudhary.com (192.0.2.1) is stored and still valid, it will be retrieved from here.

Scenario A: Cache Hit (The Good, Fast Path!!!!!!!)

- If the IP address for www.abhishekchaudhary.com (i.e., 192.0.2.1) is found in your browser's or OS's local DNS cache and is still valid (not expired):
  - Your device immediately uses 192.0.2.1 to connect directly to the web server hosting www.abhishekchaudhary.com.
  - No external DNS queries are made. This is why subsequent visits to the same website **often load much faster!** The entire process involving the Recursive Resolver, Root Servers, TLD Servers, and Authoritative Servers is completely bypassed/Skipped.

Scenario B: Cache Miss (Back to the Full Resolution Process), *as discussed above*

What Are the 13 Root Servers?

There are **13 logical root server names**, labeled from **A to M**.

Root Server	Operator	Example Root Server Name
A	Verisign, Inc.	a.root-servers.net
B	USC-ISI (University of Southern California)	b.root-servers.net
C	Cogent Communications	c.root-servers.net
D	University of Maryland	d.root-servers.net
E	NASA Ames Research Center	e.root-servers.net
F	Internet Systems Consortium (ISC)	f.root-servers.net
G	U.S. Department of Defense (DISA)	g.root-servers.net

H	U.S. Army Research Lab	h.root-servers.net
I	Netnod (Sweden)	i.root-servers.net
J	Verisign, Inc.	j.root-servers.net
K	RIPE NCC (Europe)	k.root-servers.net
L	ICANN	l.root-servers.net
M	WIDE Project (Japan)	m.root-servers.net

Importance:

- They serve the **root zone file**, which contains pointers to all **Top-Level Domains (TLDs)** like .com, .org, .net, etc.
- Without them, DNS resolution wouldn't know where to begin.
- They're **duplicated globally using anycast**, not just 13 physical machines.

Types of Records in DNS

Record Type	Purpose	Example
A (Address)	Maps a domain name to an **IPv4 address**	example.com → 93.184.216.34
AAAA	Maps a domain name to an **IPv6 address**	example.com → 2606:2800:220:1:248
CNAME	Alias of another domain name	www → myapp.hosting.net
MX	Mail server for a domain	mail.example.com priority 10
NS	Specifies **name servers** for the domain	ns1.exampledns.com
PTR	Reverse lookup: IP address → domain name	34.216.184.93.in-addr.arpa → example.com
SOA	Start of Authority — contains zone info	Primary NS, admin email, serial #
TXT	Text info (for SPF, DKIM, verification)	"v=spf1 include:_spf.google.com"

Now, Understand each Record with Example:

1. **A (Address) Record**

- **Purpose:** Maps a domain name or hostname to an IPv4 (32-bit) address. This is the most fundamental record for pointing a domain to a server.

Example:

- example.com. IN A 192.0.2.1
- www.example.com. IN A 192.0.2.1
- mail.example.com. IN A 192.0.2.2
    - **Explanation:** When someone tries to access example.com or www.example.com, their browser will be directed to the server at IPv4 address 192.0.2.1. Similarly, mail.example.com points to 192.0.2.2.

2. **AAAA (Quad-A) Record**

- **Purpose:** Maps a domain name or hostname to an IPv6 (128-bit) address. This is the IPv6 equivalent of an A record.

Example:

- example.com. IN AAAA 2001:0db8:85a3:0000:0000:8a2e:0370:7334
- www.example.com. IN AAAA 2001:0db8:85a3:0000:0000:8a2e:0370:7334
    - **Explanation:** Similar to an A record, but for IPv6. When a device request example.com and prefers IPv6, it will be directed to 2001:0db8:85a3:0000:0000:8a2e:0370:7334.

3. CNAME (Canonical Name) Record

- **Purpose:** Creates an alias from one domain name to another. When a DNS resolver queries a CNAME record, it will then perform a new lookup for the aliased name.

Example:

- blog.example.com. IN CNAME example.wordpress.com.
- www.example.com. IN CNAME example.com.
    - **Explanation:**
        - blog.example.com is an alias for example.wordpress.com. If you try to access blog.example.com, the DNS resolver will then look up the IP address for example.wordpress.com.
        - www.example.com is an alias for example.com. This is common to ensure both example.com and www.example.com point to the same resource.

4. **MX (Mail Exchange) Record**

- **Purpose:** Specifies the mail servers responsible for accepting email messages on behalf of a domain. MX records also include a "preference" value (lower is preferred) to indicate priority if multiple mail servers are listed.

Example:

- example.com. IN MX 10 mail.example.com.
- example.com. IN MX 20 backup-mail.example.com.
    - **Explanation:** When an email is sent to user@example.com, the sending mail server will first try to deliver it to mail.example.com (preference 10). If that fails, it will then try backup-mail.example.com (preference 20). Note that

mail.example.com and backup-mail.example.com would also need their own A or AAAA records.

5. NS (Name Server) Record

- **Purpose:** Specifies the authoritative DNS servers for a domain. These are the servers that hold the actual DNS records for that domain. They delegate authority for a zone.

Example:

- example.com. IN NS ns1.example.com.
- example.com. IN NS ns2.example.com.
    - **Explanation:** These records tell other DNS servers (like TLD servers) that ns1.example.com and ns2.example.com are the servers responsible for providing DNS information about anything within the example.com domain. ns1.example.com and ns2.example.com would also have corresponding A or AAAA records (sometimes called "glue records").

6. **PTR (Pointer) Record**

- **Purpose:** Maps an IP address to a domain name. This is used for reverse DNS lookups, primarily for anti-spam measures and logging. It's the inverse of an A or AAAA record. PTR records are configured in special reverse DNS zones, often managed by the IP address owner (ISP or hosting provider).

Example (for IP 192.0.2.1):

- 1.2.0.192.in-addr.arpa. IN PTR www.example.com.
    - **Explanation:** If a mail server receives an email from 192.0.2.1, it might perform a reverse DNS lookup. This PTR record would tell it that 192.0.2.1 corresponds to www.example.com. Note the reversed IP octets and the .in-addr.arpa suffix for IPv4 reverse lookups.

7. **SOA (Start of Authority) Record**

- **Purpose:** Provides authoritative information about a DNS zone, including the primary name server, the email address of the domain administrator, the domain serial number, and various timers (e.g., refresh, retry, expire, TTL). Every zone must have exactly one SOA record.

Example:

- example.com. IN SOA ns1.example.com. admin.example.com. (
- 2024060101 ; Serial
- 7200 ; Refresh (2 hours)
- 3600 ; Retry (1 hour)
- 1209600 ; Expire (2 weeks)
- 3600 ; Minimum TTL (1 hour)
    - **Explanation:**
        - ns1.example.com.: The primary name server for this zone.

- admin.example.com.: The email address of the person responsible for this zone (the first dot . is replaced by @).
- Serial: A version number that must be incremented each time the zone file is updated. This tells secondary DNS servers when to refresh their copy.
- Refresh: How often secondary name servers should check for updates.
- Retry: How long secondary name servers should wait before retrying a failed refresh.
- Expire: If a secondary server fails to refresh its data after this time, it should no longer answer queries for the zone.
- Minimum TTL: The default TTL for any records in the zone that don't have their own TTL specified.

8. TXT (Text) Record

- **Purpose:** Holds arbitrary human-readable text information or machine-readable data for various purposes. They are often used for verification, security, and policy settings.

Example:

- example.com. IN TXT "v=spf1 include:_spf.google.com ~all"
- _dmarc.example.com. IN TXT "v=DMARC1; p=quarantine; rua=mailto:dmarc-reports@example.com"
- _acme-challenge.www.example.com. IN TXT "random_string_for_cert_verification"
    - **Explanation:**
        - The first example is a Sender Policy Framework (SPF) record, used to prevent email spoofing by specifying which mail servers are authorized to send email on behalf of example.com.
        - The second is a DMARC record, used for email authentication and reporting policies.
        - The third example is often used for domain verification by services like Let's Encrypt for issuing SSL certificates.

Important Points:

A and CNAME: You can't have both for the same domain (e.g., www) — it's either one or the other.

- **CNAME** must always point to a domain name, **not** an IP address.
- **TXT** records are widely used for verification (e.g., Google site verification, SPF, DKIM, etc.).

Some Common DNS Command Line tool for troubleshooting

1. DIG (Domain Information Groper)

- Powerful, flexible, and commonly used on Linux, macOS, and even Windows (via WSL or install).
- Let you query specific DNS record types (A, MX, TXT, etc.).
- Shows detailed information, like query time, flags, and authoritative responses.

2. NSLOOKUP (Name server Lookup)

- It is a command-line tool used for querying the Domain Name System (DNS) to obtain domain name or IP address mapping.
- most common use is to find the IP address associated with a given domain name.
- It can also take an IP address and attempt to find the domain name associated with it.

A typical nslookup output will show:

Server
Address
Non-authoritative answer
Authoritative answer
Name
Addresses (or Address)
Aliases

3. set q=all in nslookup

- A command within nslookup interactive mode.
- It sets the query type to ALL, meaning it will try to return all DNS records available for a domain.

TTL Value Concept (very Important)

TTL (Time to Live) is a value (in seconds) that tells DNS resolvers how long to cache a DNS record before requesting a fresh copy from the authoritative DNS server.

Simple terms:-> TTL controls **how long** other DNS servers (like ISPs, Google DNS, Cloudflare) should **remember a DNS record** (like A, CNAME, MX, etc.).

For example-: You have "A Record" for www.abhishekchaudhary.com with TTL=3600

```
www.abhishekchaudhary.com.  IN  A  192.0.2.10  TTL = 3600
```

- This means **caching DNS resolvers** (like 8.8.8.8 or your ISP's DNS) will remember that www.abhishekchaudhary.com maps to 192.0.2.10 for **1 hour (3600 seconds)**.
- After that, they'll **ask again** from the authoritative server.

Key Points/Best Practices to Remember while Doing any DNS Changes or doing any Migration Activity.

- A few days **before migration**, **lower the TTL** (e.g., to 300 seconds).
- This ensures resolvers only cache records for **5 minutes**.
- Then, when you make the change, most users will see the **new IP quickly**.
- After successful migration, you can **increase the TTL** again (e.g., to 3600 or 86400) to reduce DNS traffic.

What is DNS Traffic

DNS traffic refers to the **data packets exchanged** between computers and DNS servers during **DNS resolution** — the process of translating a **domain name into an IP address.**

Here's How It Works (Simplified Flow):

1. You type www.google.com in a browser.
2. Your device **sends a DNS query** (e.g., "What is the IP for www.google.com?").
3. That query **travels to a DNS resolver** (like 8.8.8.8 or your ISP).

4. The resolver looks up the DNS records and **sends back a response** (IP address).
5. The browser uses that IP to connect to the web server.

👉 This **query–response exchange** is **DNS traffic**.

Why DNS Traffic Matters?

- It's the **first step in every internet request**.
- Heavy DNS traffic might indicate:
  - High website usage
  - DNS misconfiguration
  - **DNS-based attacks** (e.g., DDoS, amplification attacks)

Is DNS Traffic Secure?

By default, **no** — DNS traffic is **unencrypted**.

That's why many systems now use:

- **DoH** (DNS over HTTPS)
- **DoT** (DNS over TLS)
- These protect DNS queries from **snooping, spoofing, or tampering**

Top DNS Management Consoles & Providers (Web-Based)

Platform	Description
Infoblox	Leader in DNS, DHCP, IPAM (DDI); robust UI, automation, and security tools
BlueCat	Competes with Infoblox; full DDI solution with great automation support
Microsoft DNS (Windows Server)	GUI via DNS Manager (MMC); widely used in corporate AD environments
Men & Mice	DNS/DHCP/IPAM with hybrid cloud support and web console
EfficientIP	Advanced DNS security and DDI automation

Cloud DNS Providers (Public Internet-Facing):

Provider	Console / Platform Name	Notes
AWS Route 53	AWS Console	Highly available, scalable, supports latency/routing policies
Cloudflare DNS	Cloudflare Dashboard	Fast DNS + WAF, DDoS protection, CDN integration
Google Cloud DNS	Google Cloud Console	Highly scalable, integrated with GCP
Azure DNS	Azure Portal	Integrated with Microsoft Azure services
DigitalOcean DNS	DigitalOcean Control Panel	Simple and developer-friendly
Namecheap / GoDaddy / others	Web dashboards for DNS record management	Easy but less advanced (good for basic domains)

Open Source/Self-Hosted:

Tool	Description
PowerDNS (with PowerAdmin)	Popular open-source DNS server + web GUI
Bind9 + Webmin	Classic BIND DNS with web-based admin tool
Pi-hole (for home DNS)	Local DNS sinkhole with web UI, often used for blocking ads

References:

https://data.iana.org/TLD/tlds-alpha-by-domain.txt

https://www.icann.org/en/contracted-parties/registry-operators/resources/list-of-top-level-domains

https://www.cloudflare.com/learning/dns/what-is-dns/