

# DNS

## Basic Settings

To configure basic DNS for a new domain, we will only really need to create two records, after ensuring our nameservers are pointed to the correct location.

If you are just messing around with NGINX or Apache, there is no real need to purchase a domain to simply resolve your IP with DNS. Check out [Freenom](#) for a free domain name, granted it may not be your first choice - but they provide a wide range of **free domains** for up to 12 months.

## Nameservers

A nameserver defines the path the DNS will take to resolve your domain name's IP address. If you purchased your domain already, chances are you created an account with the vendor you purchased from. Login to this account, and locate a 'DNS Records / Settings' panel to modify your DNS records using the vendors supplied control panel. If you would rather use another control panel, for example DigitalOcean, you would need to login to your domain provider's control panel and alter your domain's nameservers to reflect the below -

```
ns1.digitalocean.com
ns2.digitalocean.com
ns3.digitalocean.com
```

This allows your domain to resolve using the appropriate servers on which we have set our DNS records using their respective control panels.

## DNS Records

To get started using our new domain, we will only need the below basic DNS settings -

Type	Hostname	Value	TTL
A	www.website.com	0.0.0.0	3600
A	@.website.com	0.0.0.0	3600
A	*.website.com	0.0.0.0	3600

These basic settings will allow you to further configure DNS on the host directly using a webserver if you so choose, or if you'd rather the interface using the control panel associated with your nameservers is fine as well.

## DNS Record Types / Definitions

### A Records

An A record maps an IPv4 address to a domain name. This determines where to direct any requests for a domain name.

### AAAA Records

An AAAA record, also called a Quad A record, maps an IPv6 address to a domain name. This determines where to direct requests for a domain name in the same way that an A record does for IPv4 addresses.

### CNAME Records

A CNAME record defines an alias for an A record; it points one domain to another domain instead of to an IP address. When the associated A record's IP address changes, the CNAME will follow to the new address.

### MX Records

An MX record specifies the mail servers responsible for accepting email on behalf of your domain. Providers often make multiple name servers available so that if one is offline, another can respond. Each server needs its own MX record.

### NS Records

An NS record specifies the *name servers*, or servers that provide DNS services, for a domain or subdomain. You can use these to direct part of your traffic to another DNS service or to delegate DNS administration for a subdomain.

## Web Query Path

In general, when we visit google.com for example, these are the steps that are taken to resolve the destination IP address, which is resolved from right-to-left.

1. Client queries google.com via some application
2. The browser checks if the DNS entry is within the local cache, then the OS checks it's local cache. If either is found to exist it returns it to the client immediately and the DNS resolution is complete; Otherwise, continue to next step
3. Client query is sent to router on the LAN
4. The router sends the query to the DNS Recursive Resolver if there is no entry in the cache

5. The DNS Recursive Resolver routes the request to the DNS Root Nameserver (.)
6. The Root server responds with the Top Level Domain (TLD) DNS server (.com, .net, .io, etc)
7. The DNS Recursive Resolver routes the query to the TLD server
8. The TLD server responds with the destination IP of the domain's nameserver (shaunreed.com, google.com, etc)
9. The Recursive Resolver sends the request to the domain's Authoritative Nameserver
10. If the request has a subdomain (like git.sh Shaunreed.com), the Authoritative Nameserver returns the final DNS Authoritative nameserver that is responsible for storing the subdomain's CNAME record. If there is no subdomain, this step is skipped and the destination IP is returned
11. The final DNS Authoritative Nameserver returns the destination IP address back to the Recursive Resolver
12. The Recursive Resolver returns the resolved destination IP address to the router which initially sent the query
13. The router delivers the resolved destination IP to the client IP that initially requested it on the router's LAN
14. The client caches the result in the OS and web browser, assigning a Time To Live (TTL) value to indicate when the cached result should expire

Once all of these steps are completed, or a cached result is returned, the client's actual request is carried out to the destination. This is true whether the request is HTTP, HTTPS, FTP, or any other protocol within the Application layer of the TCP/IP or OSI network models.

On Ubuntu Linux systems, Mozilla stores the cache in ~/.cache/mozilla, where everything from DNS resolutions, mozilla settings, and images / thumbnails are cached.

On Ubuntu Linux systems, by default the systemd-resolved service is enabled but the /etc/systemd/resolved.conf has a default Cache value of no-negative, which means the service will not cache DNS resolutions on an OS level.

## Resources and Links

### Cloudflare - DNS

---

Revision #12

Created 19 April 2019 06:49:19 by Shaun Reed

Updated 25 March 2022 13:07:50 by Shaun Reed