**DNS**

DNS which is a core technology that powers most of the activity that you perform on the internet. DNS or the domain name system is essential to everything you do on the internet. It's the thing that allows you to type in www.linuxacademy.com in your browser and get an IP that can be used to make a layer seven connection to the Linux Academy platform. Without DNS, lots of things would simply stop working and even though it's that critical, most people simply don't understand how it works and if you don't understand the DNS, there's a whole host of services and features inside AWS which best case you'll misunderstand worst case will make no sense at all.

**DNS is essentially a huge database. So any DNS name that you can type into a browser including netflix.com, microsoft.com, linuxacademy.com, and even bestcatpicsintheworldever.com these are stored on a DNS server somewhere.** So somewhere in the world there's a DNS server that has details about all those domains. **The database is huge. It has to store all of the DNS names and associated IPs for all domains in the world and it has to handle a huge amount of load.** Every web browser, PC, laptop server, embedded device all performing queries 24/7 365 and as long as it doesn't fail, you probably wouldn't know that DNS exists. **The power of DNS comes from the fact that it's not one single database. It's hierarchical. Information is distributed as needed throughout the system.** Now I want to step through DNS and talk about the theory first. I really want to make sure you understand all of the key components then once I have gone through the theory we'll have a look at how an actual query works.

Now, the first time that I want to talk about is a resolver **and the resolver inside DNS is either a piece of software on your local laptop, your iPhone, your tablet, or it can actually be a server. Its job is to take queries for a given DNS name and find the associated IP address using DNS.** Generally most physical devices use a resolver either in their local internet router or inside their ISP and in most cases, this is delivered automatically using DHCP. **Inside AWS, any resources so EC2 instances or any other VPC capable resources are given the network plus two address and this is provided using a DHCP option set inside the VPC. Whichever method is used the resolver is your initial entry point into DNS. So this is a server that you'll connect to interact with the DNS system.**

Next, I want to talk about the **DNS root.** So when you think of a DNS name, let's say linuxacademy.com what you don't always see is it is actually a hidden **full stop at the end of a DNS name.** It's not always typed. If you don't type it, your browser actually includes it but in most cases it's hidden. It isn't always stated, but it's always there. **This represents the root. It's the top level of DNS. The top level of the global database and this root is actually a database, and it's managed by 13 named servers and these are called the root servers.** Now, these aren't actually 13 servers. There's 13 named entities, each of which are managed by a separate major company but of course, for each of these hostnames, it could be provided by a cluster or even a large data center worth of servers but from a conceptual point of view, each of these individual companies manages one named entity one route server. This is the starting point of DNS. It's the initial point of trust. Your operating system whether you're using a laptop or a PC or a Mac or a tablet an iPhone. Any device essentially trusts this DNS root, and it's given this trust by the vendor of the operating system you're using. So this is the starting point, and everything about DNS flows down from this point.

Now, these **root servers actually manage what's known as the DNS root zone**, and this is a top level database **This root zone contains TLDs or top level domains.** Now examples of these top level domains include .com, .net, .au, .uk, and .org. There are all examples of top level domains some are generic and some are country specific. Now control over these TLDs is delegated to specific large organizations to manage and these large organizations themselves have DNS servers and the **IPs of these DNS servers are stored in the root zone**. So an example of this is this is a full list of all the top level domains stored inside the root zone. So all of these have been delegated to an operating company. An example of this is the .com top level domain and this has been delegated to Verisign global registry services. So this company has been delegated the authority they provide these name servers, which are responsible for this top level domain, and this data is entered into the root zone. So now **we've gone from our computers or our devices trusting just the root zone to now the root zone, pointing at the service, so delegating to the servers and now for .com we also trust these servers because we trust the root zone on the root zone trusts the servers, so it's essentially a chain of trust.**

Now, inside a **top level domain, we've got subdomains and eventually hosts.** So if you're reading any DNS name, you start at the right on the right is the route then you've got the top level domain. So in this case .com and then we've got a subdomain so Linux Academy or linuxacademy.com. **Now on the left side, a lot of times you have host so www or FTP or mail or VPN, essentially on the right hand side, we've got the root and the top level domain on the left hand side, most the time we've got a host, and then in the middle we've got one or more subdomains,** and it is possible to have this format, which is known as a **naked domain**, so it doesn't actually list a host, and that's fine but essentially anything to the left of this top level domain is known as a **subdomain**. Now, **these subdomains are also delegated out to other companies so Verisign who operate the .com top level domain have delegated the operation of linuxacademy.com to a separate organization.** In this case, it's our organization. So we manage this. We have DNS servers that have the database for linuxacademy.com and the .com operating company have delegated that to us.

**So now not only does your machine trust the root zone and the root servers, but because they trust Verisign, which operates .com and Verisign trusts us and we operate Linux Academy, you trust linuxacademy.com.**

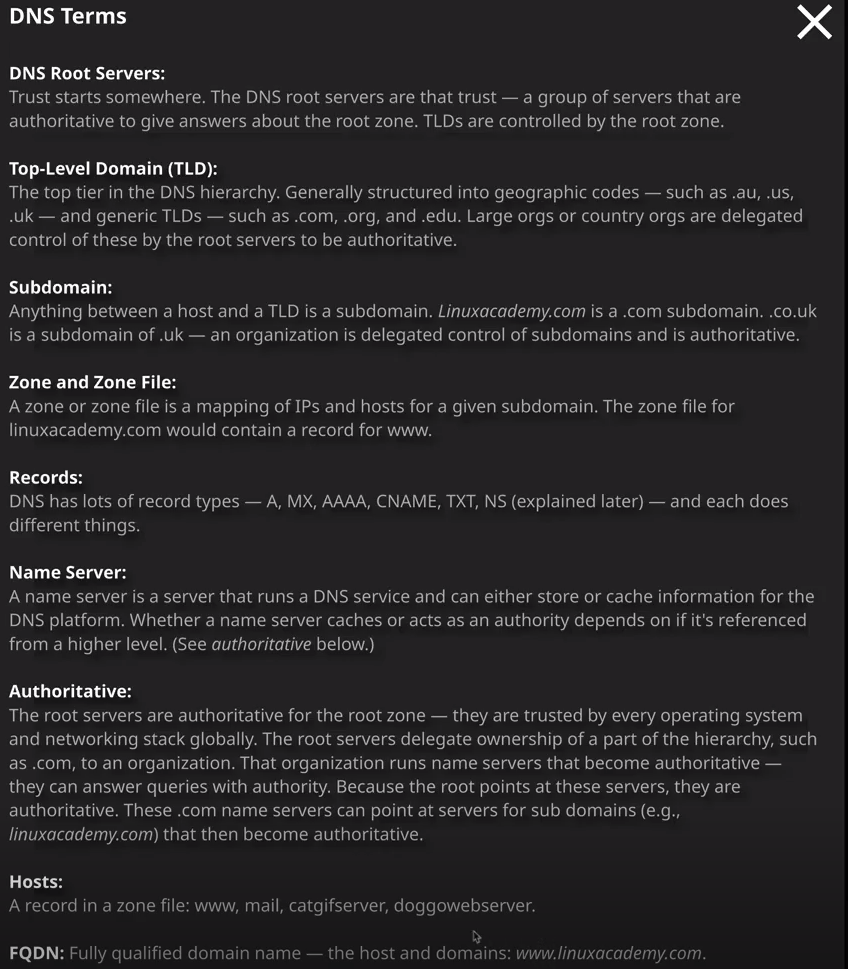
Next, we've got **zone or zone file** and this just represents an area of the global DNS database, so the root zone is a zone, and that's managed by the root servers. They delegate out .com and .com is a zone .com delegates out linuxacademy.com and that's also a zone. These zones are how DNS could be distributed so servers only know the part of the database that's relevant to them. **So the root servers have the root zone. Verisign operate service with the .com zone. Linux Academy operate servers with the linuxacademy.com zone and so they're separated using this delegated process**.

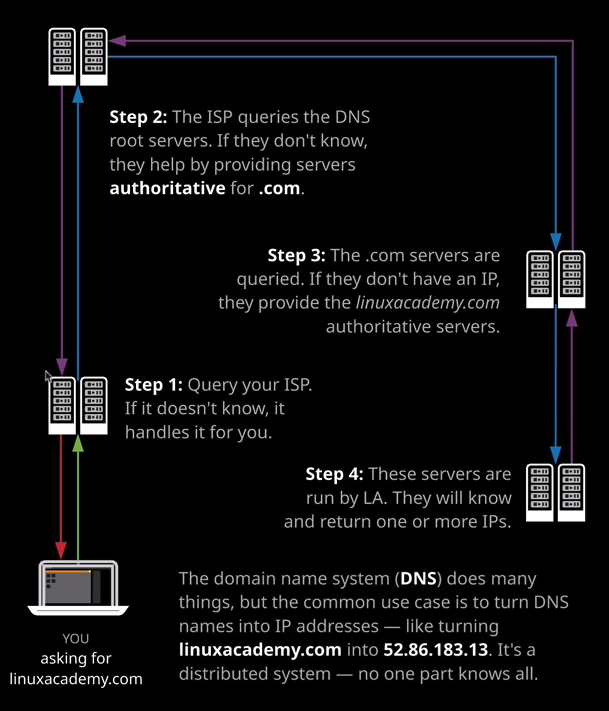
Now then, we've got **records**. DNS can have lots of different type of records. We got A records, MX records, AAA records, text records and don't worry, I've got a lesson specifically talking about records that's coming a little bit later on in the course an A record, though, is how we can get from this DNS name so linuxacademy.com to the IP address. So I want to introduce an A record straight away because that's the most common type of DNS record. When you type www.netflix.com into your browser, that is an A record, and that generally points at one or more IP addresses.

Then we've got a name server and a **name server inside DNS** is just a DNS server that hosts one or more zones. Remember, a zone is just a portion of the DNS database. So linuxacademy.com would run name servers that host the linuxacademy.com zone. So name servers remember, **they're just servers that host databases**.

Next, we've got the term **authoritative**. Now that simply means that something is an authority with a distributed system like DNS there needs to be one or more entities that are the authority for certain components of that database. **Now that just insures that you've got security all along the chain.** **So the only servers that are authoritative for the root zone are the root servers the only servers that are authoritative linuxacademy.com are our name servers.** When the root servers delegate the .com top level domain into an organization to remember, that's Verisign. They allow Verisign to give them some name server IP addresses and that gets added to the root zone so straightaway, Verisign name servers become authoritative for that domain so the .com domain. When you query a server for something in DNS, it can either give you an authoritative response meaning you could rely on it 100% or it's not on authoritative response. Now, this might be a server caching your results or maybe your ISP server 999 times out of 1000 it's the same thing but you will know when you do a query, whether the response is authoritative or not. If it's authoritative, you know it's coming from a name server that has been delegated the ability to be an authority for that zone. So if you do a query for linuxacademy.com, the only servers that can be authoritative are our name servers, you could in theory, run a DNS server with linuxacadmemy.com zone on it but the only responses that you would give would be not authoritative. **So the only way to give authoritative responses are if you've been delegated the right to be an authority and that comes from the root zone delegating to the operator of .com and then that company** **delegating the authority to Linux Academy,**

now a **host** inside DNS is everything that's not a domain. So anything like www, mail, cat GIF server, doggo web server, anything that is not included in the domain but that's ambiguous, that could refer to www.linuxacademy.com or www.bestcatpicsintheworldever.com. **The only way to be specific is to use an FQDN which is the fully qualified name.** **So you've got to specify the host and then any subdomains and then the top level domain and although you don't have to specify it explicitly, there is the root right at the end on the right hand side.** So while you don't have to put that final period, it is their implicitly, even if it's hidden.





Okay, so now I want to talk through the actual flow of DNS. So let's say that you're sat there right now attempting to open linuxacademy.com using your laptop. You want to browse to this very training video, and so you're browsing to linuxacademy.com in your browser. **Well, what you do is you open your browser and you type in the DNS name, but in order to make that connection your browser needs an IP address so your browser goes to your local operating system on the DNS client on the operating system communicates with your local DNS server. Remember, the DNS resolver that's a server that's either inside your internet router or inside your local ISP. It essentially asks for linuxacademy.com and let's assume for now it's not cached. Let's say that you're the first person to ask for it today. Well, if it doesn't know the answer, it performs what's known as a recursive query, so it gets the answer for you. The first thing it does, is it queries the root servers, so it does a query against the root servers for www.linuxacademy.com.** So that's step two in this diagram. Now, the root servers won't have an answer because they don't have any data inside the root zone for linuxacademy.com but what they will be able to do is to give you another step along the chain so they will know, for example, **the name servers that are authoritative for the .com top level domain. So they'll give the DNS resolver server that was addresses. Once the resolver server gets those details back for the .com top level domain name servers it'll go ahead and query those. Again it'll ask for linuxacademy.com or www.linuxacademy.com. Now those top level domain servers won't know the answer again, but they will have more details. They'll know the name servers that are authoritative for linuxacademy.com itself. Now these are the servers that we as a company manage, and so it will provide those details back to the resolver server.** **The final step is that initially solver will contact the linuxacademy.com name servers and then do a query for linuxacademy.com or www.linuxacademy.com. This time it will get an authoritative answer back because our name servers will have that information that'll be returned to the resolver the resolver will return that to you, and then you can browse through the linuxacademy.com website.**

Now this process is followed every single time, no matter what the query is that you perform. So let's look an actual example. So I'm going to do a query for cantrill.io which is my personal domain, and I want to show you what the output is. So I'll press enter and they'll just scroll up and show you this step by step. Now, I use 8888 as my local DNS resolver. So that's the first step is to query that for cantrill.io that's not going to have that result of assuming it's not cached which it doesn't look to be but what it is going to do is give me **the next step along the chain, which is the root servers. The next step is to query those root servers for cantrill.io Those roots servers won't have an answer, but what they will be able to do is to give the authoritative name servers for the .IO top level domain and that's what these are so ns-.a3.io is one example ns-a1 is another example and so on. These are all authoritative for the .IO top level domain. Now, I don't have the answers that I need yet, so the next step is to query these name servers that are authoritative for the .IO domain. So that's the next step. So I query the .IO name server for cantrill.io again, they can't give me an authoritative answer, but they can give me the name servers that are authoritative for cantrill.io in this particular case these are servers provided by Route 53. One example is ns-96.awsdns-12.com. Now the final step because we know that these are authoritative for cantrill.io is to perform a query against these names servers for cantrill.io itself. This could either be cantrill.io or it could be www.cantrill.io These servers are able to give an authoritative answer back because they have been delegated the operation of cantrill.io and in this particular case, I get a record, which gives an IP address. So this is 185.119.110.153** and I do get four different alternatives.

**So these are four different options to go to to get to cantrill.io and that is this process which the same process all I do is a query my local resolver then the root servers, then the server that's been delegated the top level domain operational responsibility and then I get the name servers that are responsible for the actual domain, and then they can provide an authoritative answer. It's the same process end to end whatever you query inside DNS. Now that's just a small taste of an example of exactly how DNS works. DNS is one of the most important systems on the internet. It is elegant in its simplicity it's a multilevel system. You make a query, and you either get an authoritative answer or you get another server to go to to get you one step closer to the authoritative answer because of that, the data in DNS is spread across thousands or even millions of zones which are hosted on name servers globally. Now Route 53 inside AWS is a DNS name server product in part so it does lots of things, but you can use it to host DNS zones. They get hosted globally on highly scalable name servers it can host public zones which are accessible from the public, internet or private zones which only work inside a VPC but it can also register domains** and that's what we're going to look at nex.