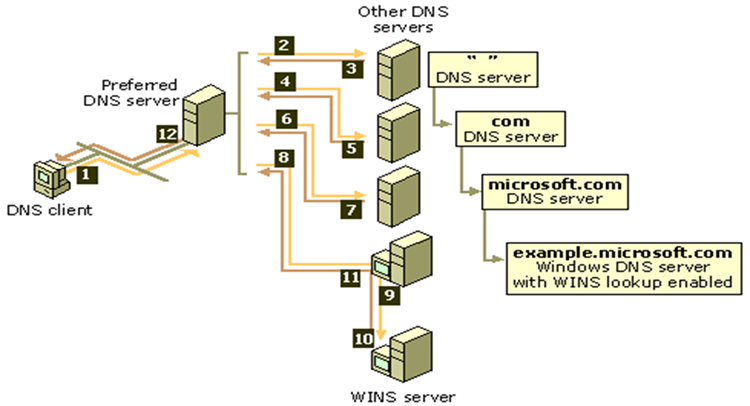
DNS (Domain Name System)

# Review of DNS Terminology

* Root Servers
  + *13 Root Servers – know the names of the Top-level DNS servers*
    - [(https://www.iana.org/domains/root/servers](https://www.iana.org/domains/root/servers))
  + *For example, if you access mysql.org:*
    - Your local DNS SERVER will refer to the root server
    - The root server will reply with the address of a DNS which is closer to the actual target.
    - Your server will continue querying until it finds a DNS server that actually has that information.
    - The DNS SERVER will then **cache** that address for future reference.



# Two important files for finding other hosts in Linux:

1. /etc/hosts 🡨 We know this one!
2. Text

   Description automatically generated/etc/nssswitch.conf
   * The line “**hosts: files mdns4\_minimal [NOTFOUND=return] dns**” tells your local machine how to find other host addresses
     + *First entry: “files” tells your machine to look in the /etc/hosts file first*
     + *Second entry: “mdns4\_minimal” says look for a machine ip using multicastdns*
       - If not found continue
     + *Third entry indicates to use “dns”*

# Some Useful Commands

1. **nslookup “hostname”**

Examples:

Text

Description automatically generatedText

Description automatically generated

*The output contains the following information:*

* + - the DNS server “Server”
    - the “Address” of the DNS server, including port number
    - the “real” name of the requested site
    - The IP address

We can use a different DNS Server:

**nslookup www.saskpolytech.ca 8.8.8.8** 🡨 Google’s recursive DNS server

Text

Description automatically generated

Note the internal and external addresses for the saskpoly website are different (10.34.60.42 & 142.99.246.43), this is because when we use the Google DNS server we are hitting SaskPoly from the outside – public internet!

1. **Ping**
   1. some useful options for ping: (man ping to see all of them!)
      1. **ping -q (quiet)**
      2. **ping -w (wait – in seconds)**
      3. **ping -c (count)**
2. **DIG** – Domain Information Groper – part of the DNS utility package
   1. Install: **apt-get install dnsutils**
   2. Syntax: **dig [server][name][type]**
   3. Reference: <https://www.hostinger.com/tutorials/how-to-use-the-dig-command-in-linux/>
   4. Examples:

Timeline

Description automatically generated

Text

Description automatically generated

# DNS Record Types

|  |  |
| --- | --- |
| **Record Type** | **Meaning** |
| A | IPv4 Host Address |
| AAAA | IPv6 Host Address |
| NS | Name Server |
| MX | Mail Exchange |
| PTR | Pointer (Reverse look ups) |
| SOA | Start of Authority |
| CNAME | Canonical Name (similar to an alias) |

# Types of DNS Servers

1. **Authoritative** – nameserver that gives the definitive answer to a question about a name in its zone.
2. **Caching** – nameserver (DNS caches) store DNS query results but they don’t have authoritative answers.
3. **Forwarding** – nameserver that sends all requests on to other nameservers – often these will be set up in a network so that all requests for outside DNS information go through one machine. These can also be caching**.**
4. **Recursive –** nameserver that will recursively query the authoritative DNS servers – first check the root server, then the top-level domain server, etc. These can also be caching**.**

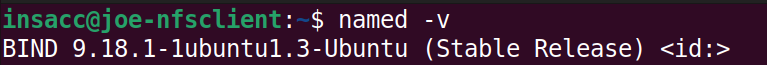
Notes:

* In practice, many nameservers will do multiple of these – most forwarding namerservers will cache results to prevent excess requests.
* Some of them don’t make sense to combine – a server can NOT be both forwarding and recursive in general.
* For security reasons, you often want authoritative namesesrvers separated from forwarding or recursive nameservers.

# BIND Exercise:

* BIND stands for Berkeley Internet Name Daemon
* Primary Linux Based DNS Server <https://www.isc.org/bind/>
* We will configure BIND as an authoritative name server…

1. Set up our own named server:
   1. **sudo apt install bind9 bind9utils bind9-dnsutils bind9-doc bind9-host**
   2. check the version of named: **named -v**

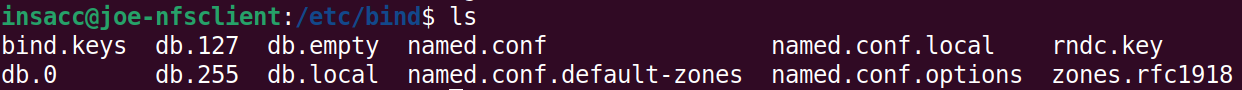


* 1. Note: the name of the BIND service is **named**
  2. It is running upon install: **systemctl status named**

Text

Description automatically generated

* 1. BIND listens on port 53
     + **ss -lp “sport = 53”**
       - The ss command shows open ports, -l = listen state, -p = the process that owns the port (“man ss” for more information)
       - **sport = 53**, filters for the source port 53.
     + **ss -tunl** 🡨 a different set of options with different output.

1. Let’s look at some default files in /etc/bind
   1. **named.conf** – primary BIND config file
   2. **db.127** – localhost IPv4 reversename mapping
   3. **db.local** – localhost forward IPv4 and IPv6 mapping zone file
   4. 

Note: Take a look at the root DNS servers list in **/usr/share/dns/root.hints**

Graphical user interface, text

Description automatically generated

We can see all 13 previously mentioned root DNS servers (A-M) listed here.

### Class DNS Data:

|  |  |
| --- | --- |
| Name | IP Address (/23) |
| Tweety | 10.28.53.19 |
| Hugo | 10.28.52.24 |
| Marvin | 10.28.53.21 |
| Yosemite | 10.28.52.25 |
| Taz | 10.28.53.7 |
| Lola | 10.28.53.20 |
| Bugs | 10.28.53.8 |
| RoadRunner | 10.28.53.22 |
| Charles | 10.28.53.11 |
| Sylvester | 10.28.52.20 |

### Set up our DNS Servers

We are all going to be working in a domain called **looneytunes.com**

* All the records will be similar form.
* Create a sub directory: **sudo mkdir /etc/bind/zones**
* Then **sudo vi /etc/bind/named.conf.local** and add the following lines*:*

zone "52.28.10.in-addr.arpa" in {

type master;

file "/etc/bind/zones/db.10.28.52";

};

zone "looneytunes.com" {

type master;

file "/etc/bind/zones/db.looneytunes.com";

};

Next we will create the DNS database file: **sudo vi /etc/bind/zones/db.looneytunes.com** and add the following lines:

$TTL 3H

@ IN SOA localhost. root.localhost. (

2 ; Serial

3h ; Refresh

1h ; Retry

1w ; Expire

1h ) ; Negative Cache TTL

;

Followed by our DNS records… (See the screenshot below)

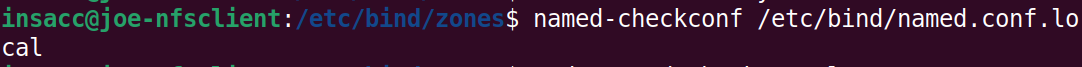
Text

Description automatically generated

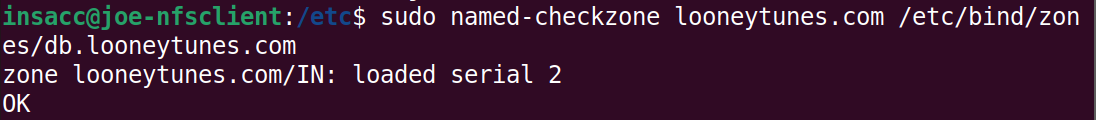
* Important Syntax for the db.loooneytunes.com file:
  + Don’t forget the “.” after the .com – this ends the entry, if the “.” is missed it will add looneytunes.com again and get confused.
  + There are syntax shortcuts:
* for the host records - you can just list the name and leave out the domain.
* Other notes for this file:
  + **TTL** the amount of time we are going to keep the records before we must query an authoritative names server again
  + **SOA** - Start of Authority for the given zone.
  + **Serial** - Should be changed every time you change this file
    - a common format is: yyyymmdd##
  + **Refresh** - the time between update between primary and secondary DNS Controllers
  + **Retry** - the time the secondary server will wait before retrying to update off the primary server
  + **Expire** - the amount of time it will consider records valid if primary DNS cannot be reached
  + **Negative** **Cache TTL** - (Previous versions used this as the TTL value) – specifies how long a negative cache (host not found) record will exist
  + **NS** – DNS record identifying the name of the nameserver for this domain
  + **A** – DNS records listing the addresses for the various hosts on our network

**Check the Configurations:**

* To check that named.conf.local is configured:
  + **named-checkconf /etc/bind/named.conf.local**



* To check that our looneyTunes is set up correctly:
  + **sudo named-checkzone looneytunes.com /etc/bind/zones/db.looneytunes.com**



DNS is very touchy about configurations! If you have any problems with the conf file the service will stop. Keep in mind that DNS is a mission critical service on your network, so always be careful when editing the files!

Next we want to edit the /etc/resolv.conf file as follows: **sudo vi /etc/resolv.conf and a**dd the following three lines:

**nameserver 127.0.0.1  
nameserver 8.8.8.8  
search looneytunes.com**

* BUT… It’s a sym link! So first **rm resolv.conf**, then **sudo vi resolv.conf** and edit. ☺
* Restart named/BIND: ***sudo systemctl restart named***



You can then ping your neighbours by name (tweety, hugo, marvin, etc. Just make sure they’re entered correctly in your db.looneytunes.com file!)

Text

Description automatically generated with medium confidence

Note: you can ping with either “name” or name.looneytunes.com, this is because of the **search directive in resolv.conf**

Next try and dig for one of the names you added to your db.looneytunes.com file:

* **dig name.looneytunes.com**

**Troubleshooting notes:** We added an entry in /etc/hosts for name.looneytunes.com to 127.0.0.1. Because of this you will need to put your current hostname (from your prompt) into your /etc/hosts file using the 127/0/0/1 address as well.

* Here’s my hosts file:

Text

Description automatically generated