Self Hosted DNS

BIND 9

UMD Homelab Club Meeting 2025-03-03

What is DNS?

(D)omain (N)ame (S)ystem

Make DNS queries to lookup DNS records

Very short explanation of some types of DNS records:

- A: "address", hostname -> IP address
- AAAA: "address x4" hostname -> IPv6 address
- PTR: "pointer", IP address -> hostname
- CNAME: "canonical name", hostname -> hostname
- MX: "mail exchange"
- TXT: "text"
- NS: "nameserver"

(Note: This is just a quick review! Check out our last presentation for more info: https://suddenlysixam.club/meetings/past_meetings/2025-02-24-meeting.html)
(Note: IPv4 32 bits, IPv6 128 bits, 32 x 4 = 128)

host

```
labclub@raspberrypi:~ $ host google.com
google.com has address 142.251.16.138
google.com has address 142.251.16.100
google.com has address 142.251.16.139
google.com has address 142.251.16.102
google.com has address 142.251.16.113
google.com has address 142.251.16.101
google.com has IPv6 address 2607:f8b0:4004:c17::8a
google.com has IPv6 address 2607:f8b0:4004:c17::64
google.com has IPv6 address 2607:f8b0:4004:c17::66
google.com has IPv6 address 2607:f8b0:4004:c17::65
google.com mail is handled by 10 smtp.google.com.
labclub@raspberrypi:~ $
```

(man page)

nslookup

```
labclub@raspberrypi:~ $ nslookup google.com
Server:         128.8.120.19
Address:    128.8.120.19#53

Non-authoritative answer:
Name:    google.com
Address: 142.251.16.138
Name:    google.com
Address: 142.251.16.139
Name:    google.com
Address: 142.251.16.102
Name:    google.com
Address: 142.251.16.100
Name:    google.com
Address: 142.251.16.113
Name:    google.com
Address: 142.251.16.101
Name:    google.com
Address: 2607:f8b0:4004:c17::8a
Name:    google.com
Address: 2607:f8b0:4004:c17::64
Name:    google.com
Address: 2607:f8b0:4004:c17::66
Name:    google.com
Address: 2607:f8b0:4004:c17::65

labclub@raspberrypi:~ $
```

(man page)

nslookup (cont.)

```
labclub@raspberrypi:~ $ nslookup -type=ns google.com
Server:         128.8.120.19
Address:        128.8.120.19#53

Non-authoritative answer:
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns1.google.com.

Authoritative answers can be found from:
ns1.google.com  internet address = 216.239.32.10
ns4.google.com  internet address = 216.239.38.10
ns3.google.com  internet address = 216.239.36.10
ns2.google.com  internet address = 216.239.34.10
ns1.google.com  has AAAA address 2001:4860:4802:32::a
ns4.google.com  has AAAA address 2001:4860:4802:38::a
ns3.google.com  has AAAA address 2001:4860:4802:36::a
ns2.google.com  has AAAA address 2001:4860:4802:34::a

labclub@raspberrypi:~ $
```

(man page)

nslookup (cont.):

```
labclub@raspberrypi:~ $ nslookup google.com
Server:         128.8.120.19
Address:     128.8.120.19#53

Non-authoritative answer:
Name:    google.com
Address: 142.251.16.138
Name:    google.com
Address: 142.251.16.139
Name:    google.com
Address: 142.251.16.102
Name:    google.com
Address: 142.251.16.100
Name:    google.com
Address: 142.251.16.113
Name:    google.com
Address: 142.251.16.101
Name:    google.com
Address: 2607:f8b0:4004:c17::8a
Name:    google.com
Address: 2607:f8b0:4004:c17::64
Name:    google.com
Address: 2607:f8b0:4004:c17::66
Name:    google.com
Address: 2607:f8b0:4004:c17::65

labclub@raspberrypi:~ $
```

```
labclub@raspberrypi:~ $ nslookup google.com 1.1.1.1
Server:         1.1.1.1
Address:     1.1.1.1#53

Non-authoritative answer:
Name:    google.com
Address: 142.250.31.100
Name:    google.com
Address: 142.250.31.139
Name:    google.com
Address: 142.250.31.101
Name:    google.com
Address: 142.250.31.138
Name:    google.com
Address: 142.250.31.113
Name:    google.com
Address: 142.250.31.102
Name:    google.com
Address: 2607:f8b0:4004:c07::66
Name:    google.com
Address: 2607:f8b0:4004:c07::8a
Name:    google.com
Address: 2607:f8b0:4004:c07::65
Name:    google.com
Address: 2607:f8b0:4004:c07::8b

labclub@raspberrypi:~ $
```

(man page)

dig

```
labclub@raspberrypi:~ $ dig google.com

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 274
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 2c2057ec0075e2ea0100000067c5ecae05519ddeafe9bc01 (good)
;; QUESTION SECTION:
;google.com.            IN      A

;; ANSWER SECTION:
google.com.         300     IN      A       142.251.16.102
google.com.         300     IN      A       142.251.16.101
google.com.         300     IN      A       142.251.16.139
google.com.         300     IN      A       142.251.16.100
google.com.         300     IN      A       142.251.16.138
google.com.         300     IN      A       142.251.16.113

;; Query time: 11 msec
;; SERVER: 128.8.120.19#53(128.8.120.19) (UDP)
;; WHEN: Mon Mar 03 12:53:50 EST 2025
;; MSG SIZE  rcvd: 163

labclub@raspberrypi:~ $
```

(Note: the answer section may remind you of a DNS record)
(man page)

ping

```
labclub@raspberrypi:~ $ ping google.com -c 4
PING google.com (142.251.16.101) 56(84) bytes of data.
64 bytes from bl-in-f101.1e100.net (142.251.16.101): icmp_seq=1 ttl=53 time=4.36 ms
64 bytes from bl-in-f101.1e100.net (142.251.16.101): icmp_seq=2 ttl=53 time=4.04 ms
64 bytes from bl-in-f101.1e100.net (142.251.16.101): icmp_seq=3 ttl=53 time=4.11 ms
64 bytes from bl-in-f101.1e100.net (142.251.16.101): icmp_seq=4 ttl=53 time=13.0 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 4.039/6.382/13.018/3.832 ms
labclub@raspberrypi:~ $
```

(man page)

Let’s get DNS set up on the Raspberry Pi!

Package installation:

```
sudo apt update
sudo apt upgrade

sudo apt install bind9-dnsutils
```

(Note: `bind9-dnsutils` gives us (among other things) `nslookup` & `dig`. It has a package dependency for `bind9-host` which gives us `host`.)
(Note: `dnsutils` is a transitional package for `bind9-dnsutils`. So we will not use it here, but you may see `dnsutils` used in other guides.)

Set hostname:

```
sudo vi /etc/hosts
sudo vi /etc/hostname
sudo shutdown -r now
```

Once the host has finished rebooting, check to see if your changes applied:

```
hostname
```

(Note: You might ask, why not use `sudo hostnamectl set-hostname <your-hostname>`? If you are curious, you can run this and then `cat` the mentioned files, and see what has changed.)
(Note: After you set the hostname and before you reboot, it may yell at you about the previous hostname name/service not being known.)

Local name resolution - more in /etc/hosts:

Add a different hostname to /etc/hosts:

```
sudo vi /etc/hosts
```

Then let's test with some of the commands we went over before and see their behavior:

```
host <hostname>
nslookup <hostname>
dig <hostname>
ping <hostname>
```

(Note: You can test what happens with your own device's hostname that we configured in the last slide too.)

Setting a static IP:

In our case, we will be using NetworkManager to set a static IP.

```
nmcli
nmcli connection show
```

Take note of the "Name" of the connection you are using. (NOT the "Device". Sometimes these will be the same, but not always.)

```
nmcli con mod "<connection-name>" ipv4.addresses <ip-address>/<subnet-mask>
nmcli con mod "<connection-name>" ipv4.gateway <gateway-ip-address>
nmcli con mod "<connection-name>" ipv4.method manual
nmcli con down "<connection-name>" && nmcli con up "<connection-name>"
```

```
nmcli
```

(Note: How you set a static IP will greatly vary by OS and OS version.)
(Note: Some parts of these commands can be shortened, such as `connection` > `con`. Some can be shortened even further.)
(Note: `sudo systemctl restart NetworkManager` will add the new config, but will not remove the old, which is not ideal. This is why we are running the `up` and `down` commands for the connection. You could also reboot your pi.)
(Note: `<subnet-mask>` based on the netmask[0])
[0] Subnet mask cheat sheet: https://dnsmadeeasy.com/support/subnet

BIND9 - packages:

```
sudo apt install bind9 bind9-utils
```

(optional) `sudo apt install bind9-doc`

(Note: BIND9 is not the only choice, it is just our choice)
(Note: bind9 for the service, bind9-utils for ways to check our work, bind9-docs for documentation.)
(Note: bind9utils is a transitional package for bind9-utils. So we will not use it here, but you may see bind9utils used in other guides.)
[0] Additional package info: https://www.kali.org/tools/bind9/

BIND9 - Configuration files

First, lets look at the systemd service for bind9:
```
cat /etc/systemd/system/bind9.service
```

Take note of:
```
EnvironmentFile=-/etc/default/named
ExecStart=/usr/sbin/named -f $OPTIONS
Alias=bind9.service
```

Next lets take a look at the current status of the service:
```
systemctl status bind9
systemctl status named
```

Because of what we have now observed, lets look at the named service script:
```
cat /etc/init.d/named | less
```

We can see that this has a comment about creating/changing /etc/default/named (and this was the *EnvironmentFile* that we noted earlier), so let's take a look at that next.

(Note: In the bind9.service file, the =-, indicates that if the file does not exist, it will not be read and no error or warning message is logged.)

BIND9 - Configuration files (cont.)

sudo vi /etc/default/named

Lets configure this to run only on IPv4 by adding -4 to the options.

Default:

```
OPTIONS="-u bind"
```

Change to:

```
OPTIONS="-u bind -4"
```

BIND9 - Configuration files (cont.)

```
cat /etc/bind/named.conf
```

We see 3 included conf files, which we are going to begin configuring for different things:

/etc/bind/named.conf.options (ACLs, forwarders, port, etc.)
/etc/bind/named.conf.local (declare our zones)
/etc/bind/named.conf.default-zones (default zone declarations)

Throughout this you may also want to be checking your work, but I am not going to list this command on every slide. Validate your changes as you go.

```
sudo named-checkconf
```

named.conf.options:

```
sudo vi /etc/bind/named.conf.options
```

(A)ccess (C)ontrol (L)ist

Put the following statement above the options {...} statement:

```
acl trusted {
        10.70.50.0/24;
        localhost;
        localnets;
};
```

BIND has the following built-in ACLs:

none: Matches no hosts.
any: Matches all hosts.
localhost: 127.0.0.1 and ::1, as well as the IP addresses of all interfaces on the server that runs BIND.
localnets: 127.0.0.1 and ::1, as well as all subnets the server that runs BIND is directly connected to.

(Note: If you would like to look at all the options, and installed bind9-doc, you can take a look at /usr/share/doc/bind9/options.gz.)
(Note: You could put the ACL directly in named.conf, however since we are going to use it in named.conf.options I'm putting it here.)

named.conf.options (cont.):

Now add some configuration within the options {...} statement:

sudo vi /etc/bind/named.conf.options

Allow DNS queries from the ACL we defined:

```
    allow-query     { trusted; };
    allow-recursion { trusted; };    # allow them to recursively query authoritative DNS servers for the queried domain
```

Forward requests for records that this server does not have:

```
    forward only;    # don't attempt to contact other NS if forwarders not available
    forwarders {
        1.1.1.1;
        1.0.0.1;
    };
```

Only IPv4. Change the second IP to that of the pi.

```
    listen-on port 53 { 127.0.0.1; 10.70.50.104; };
    listen-on-v6 { none; };
```

Others:

```
    auth-nxdomain no;    # conform to RFC1035 - yes/no answer authoritative if NXDOMAIN
    allow-transfer { none; };    # Do not transfer the zone information to the secondary DNS
```

(Note: If you would like to look at all the options, and installed bind9-doc, you can take a look at /usr/share/doc/bind9/options.gz.)
[0] https://wiki.debian.org/Bind9

named.conf.local:

```
sudo vi /etc/bind/named.conf.local
```

Declare the zones associated with this server's domain(s). Replace domain(s) and IP address(s) as appropriate for your setup:

```
### Forward zones
zone "umdhomelab.local" {
    type master;
    file "/etc/bind/zones/umdhomelab.zone";
    allow-update { none; };    # no DDNS by default
};

### Reverse zones
# 10.70.50.0/24 subnet
zone "50.70.10.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/10.70.50.zone";
    allow-update { none; };    # no DDNS by default
};
```

[0] https://wiki.debian.org/Bind9

BIND9 - Configuration files (cont.):

You may want to do additional configuration, such as logging, but we aren't doing any more for the sake of brevity.

[0] https://wiki.debian.org/Bind9

BIND9 - Configure zones

```
cd /etc/bind/
sudo mkdir ./zones
sudo cp db.local ./zones/umdhomelab.zone
sudo cp db.127 ./zones/10.70.50.zone
cd zones
```

umdhomelab.zone

Replace domain(s) and IP address(s) as appropriate for your setup:

```
;
; BIND data file for forward umdhomelab.local
;
$TTL    604800
@       IN      SOA     druid.umdhomelab.local. admin.umdhomelab.local. (
                     2025030300         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
; name servers - NS records
        IN      NS      druid.umdhomelab.local.

$ORIGIN umdhomelab.local.

; name servers - A records
druid               IN      A       10.70.50.130

; 10.70.50.0/24
paladin             IN      A       10.70.50.104
```

(Note: The serial that I've configured here is the date plus a two digit integer YYYYMMDDxx. It needs to be updated / incremented by at least 1 every time you make changes. You could simply make this an integer starting at 1, but that would go against my training.)
(Note: $ORIGIN defines a base name from which 'unqualified' names (those without a terminating dot) substitutions are made when processing the zone file.)

10.70.50.zone:

Replace domain(s) and IP address(s) as appropriate for your setup:

```
;
; BIND data file for reverse 50.70.10.in-addr.arpa
;
$TTL    604800
@       IN      SOA     druid.umdhomelab.local. admin.umdhomelab.local. (
                     2025030300         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
; name servers - NS records
        IN      NS      druid.umdhomelab.local.

$ORIGIN 50.70.10.in-addr.arpa.

; Name Servers - PTR Records
130      IN     PTR     druid.umdhomelab.local.

; PTR Records
104      IN     PTR     paladin.umdhomelab.local.
```

But does it work?:

Check your work. Is it OK?

```
named-checkzone umdhomelab.local umdhomelab.zone
named-checkzone 50.70.10.in-addr.arpa 10.70.50.zone
```

Restart the service. Make sure its still running properly.

```
sudo systemctl restart named
systemctl status named
```

Now what happens when we run:

```
nslookup <hostname>
nslookup <hostname> 127.0.0.1
nslookup <hostname>.umdhomelab.local 127.0.0.1
dig @127.0.0.1 <hostname>.umdhomelab.local
```

What is our current DNS server?:

```
cat /etc/resolv.conf
```

(Note: Don't overlook the lack of an e in resolv.conf. Tab complete is your friend.)
(Note: Additionally note the Generated by line at the top of the file, if there is one. This may indicate that we do not want to edit this file directly.)

Setting different DNS servers:

In our case, we will be using NetworkManager to modify our DNS servers.

```
nmcli
nmcli connection show
```

Take note of the "Name" of the connection you are using. (NOT the "Device". Sometimes these will be the same, but not always.)

```
nmcli con mod "<connection-name>" ipv4.dns "<space-separated-dns-ips>"
nmcli con mod "<connection-name>" ipv4.ignore-auto-dns yes
nmcli con down "<connection-name>" && nmcli con up "<connection-name>"

nmcli
```

(Optional) Set a search domain

```
nmcli con mod "<connection-name>" ipv4.dns-search "<domain>"
```

(Note: How you modify your DNS servers will greatly vary by OS and OS version.)
(Note: As mentioned previously in these slides, some parts of these commands can be shortened, such as `connection` > `con`. Some can be shortened even further.) (Note: `sudo systemctl restart NetworkManager` will add the new config, but will not remove the old, which is not ideal. This is why we are running the `up` and `down` commands for the connection. You could also reboot your pi.)

But does it work (pt 2.)

Now what happens when we run:

```
nslookup <hostname>
nslookup <hostname>.umdhomelab.local
dig <hostname>
dig <hostname>.umdhomelab.local
```

General reminders:

Increment the serial each time you make changes!

Check your work:

```
named-checkconf
named-checkzone <zonename> <filename>
```

e.g.

```
named-checkzone umdhomelab.local umdhomelab.zone
named-checkzone 50.70.10.in-addr.arpa 10.70.50.zone
```

Troubleshooting - Can you connect to port 53?:

```
telnet <remote-server> 53
```

```
bash-3.2$ telnet 1.1.1.1 53
Trying 1.1.1.1...
Connected to one.one.one.one.
Escape character is '^]'.
```

```
nc -vz -w 1 <remote-server> 53
nc -vuz -w 1 <remote-server> 53
```

```
bash-3.2$ nc -vz -w 1 1.1.1.1 53
Connection to 1.1.1.1 port 53 [tcp/domain] succeeded!
bash-3.2$ nc -vuz -w 1 1.1.1.1 53
Connection to 1.1.1.1 port 53 [udp/domain] succeeded!
bash-3.2$
```

Troubleshooting - Is port 53 open?:

```
sudo netstat -tulpn | grep :53
sudo lsof -Pi | grep LISTEN
sudo nmap -sS localhost
sudo nmap -sU localhost
```

Have you configured any variety of firewall? And the follow up question, have you configured it to allow DNS / port 53? (e.g. `iptables`, `nftables`, `firewalld`, `ufw`)

(Note: If you are following this guide with the same hardware/software, you should not ***need*** to configure this out of the box. I am including it because different OS'es may ship with a firewall already in place, and in practice you would want to configure this further.)

Troubleshooting - named service:

```
systemctl status named
sudo journalctl -u named
```

Extras:

Want to take this a step further?

Combine with git for backups & version control. Check out our Self Hosted Git[0] project for how you can run git yourself in your homelab.

[0] https://suddenlysixam.club/meetings/past_meetings/2025-02-17-meeting.html

Thank you!

Don't forget to join the Discord!

https://suddenlysixam.club/discord

Home is where the lab is. ~Megan