

Week #4

Implementation of a Local DNS Server and Authoritative NameServer

DNS (Domain Name System) is the Internet's phone book; it translates hostnames to IP addresses (and vice versa). This translation is through DNS resolution, which happens behind the scene.

The objectives of this lab are to understand:

- Install, set up and deploy a local DNS server
- Deploy authoritative nameserver for example.com domain

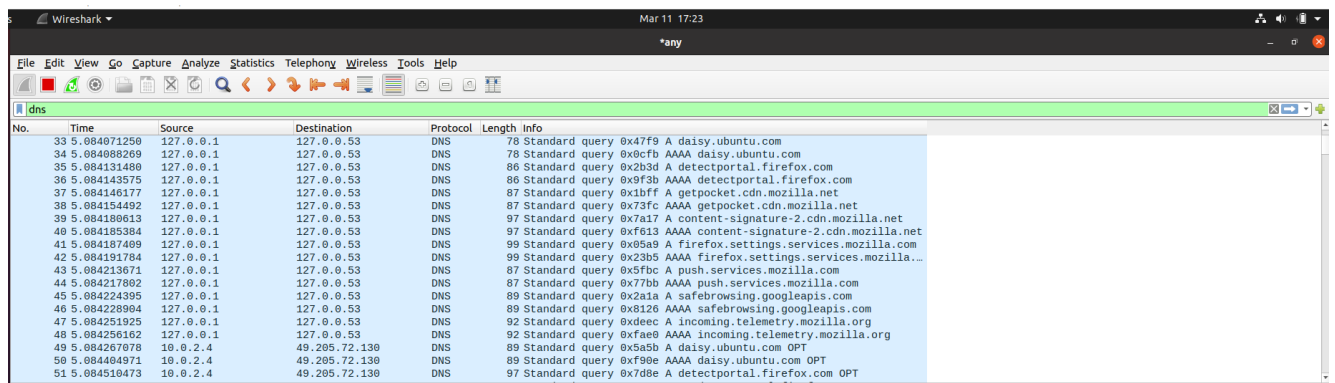
Lab Setup (with Internet Connection)

DNS Server: 10.2.22.184 User/Client: 10.2.22.195 *Note:*

Use the default IP address provided by PESU LAN.

Observation 1:

Ping a computer such as www.google.com (any domain). Please use Wireshark to show the DNS query triggered by your ping command and DNS response. Describe your observation. (Take a screenshot).



No.	Time	Source	Destination	Protocol	Length	Info
33	5.084071250	127.0.0.1	127.0.0.53	DNS	78	Standard query 0x47f9 A daisy.ubuntu.com
34	5.084088269	127.0.0.1	127.0.0.53	DNS	78	Standard query 0x0cfb AAAA daisy.ubuntu.com
35	5.084131480	127.0.0.1	127.0.0.53	DNS	86	Standard query 0x2b3d A detectportal.firefox.com
36	5.084143575	127.0.0.1	127.0.0.53	DNS	86	Standard query 0x9f3b AAAA detectportal.firefox.com
37	5.084146177	127.0.0.1	127.0.0.53	DNS	87	Standard query 0xb1ff A getpocket.cdn.mozilla.net
38	5.084154492	127.0.0.1	127.0.0.53	DNS	87	Standard query 0x73fc AAAA getpocket.cdn.mozilla.net
39	5.084189613	127.0.0.1	127.0.0.53	DNS	97	Standard query 0x7a17 A content-signature-2.cdn.mozilla.net
40	5.084185384	127.0.0.1	127.0.0.53	DNS	97	Standard query 0xf813 AAAA content-signature-2.cdn.mozilla.net
41	5.084187489	127.0.0.1	127.0.0.53	DNS	99	Standard query 0x05a9 A firefox.settings.services.mozilla.com
42	5.084191784	127.0.0.1	127.0.0.53	DNS	99	Standard query 0x23b5 AAAA firefox.settings.services.mozilla.com
43	5.084213671	127.0.0.1	127.0.0.53	DNS	87	Standard query 0x5fbc A push.services.mozilla.com
44	5.084217802	127.0.0.1	127.0.0.53	DNS	87	Standard query 0x77bb AAAA push.services.mozilla.com
45	5.084224395	127.0.0.1	127.0.0.53	DNS	89	Standard query 0x2a1a A safebrowsing.googleapis.com
46	5.084228904	127.0.0.1	127.0.0.53	DNS	89	Standard query 0x8126 AAAA safebrowsing.googleapis.com
47	5.084251925	127.0.0.1	127.0.0.53	DNS	92	Standard query 0xdeec A incoming.telemetry.mozilla.org
48	5.084256162	127.0.0.1	127.0.0.53	DNS	92	Standard query 0xf8ae AAAA incoming.telemetry.mozilla.org
49	5.084267078	10.0.2.4	49.205.72.130	DNS	89	Standard query 0x5a5b A daisy.ubuntu.com OPT
50	5.084404971	10.0.2.4	49.205.72.130	DNS	89	Standard query 0xf90e AAAA daisy.ubuntu.com OPT
51	5.084510473	10.0.2.4	49.205.72.130	DNS	97	Standard query 0x708e A detectportal.firefox.com OPT

Part 1: Setting Up a Local DNS Server

Task 1: Configure the User/Client Machine

On the client machine 10.2.22.195, we need to use 10.2.22.184 as the local DNS server. This is achieved by changing the resolver configuration file (**/etc/resolv.conf**) of the user machine, so the server 10.2.22.184 is added as the first nameserver entry in the file, i.e., this

server will be used as the primary DNS server. Add the following entry to the `/etc/resolvconf/resolv.conf.d/head` file.

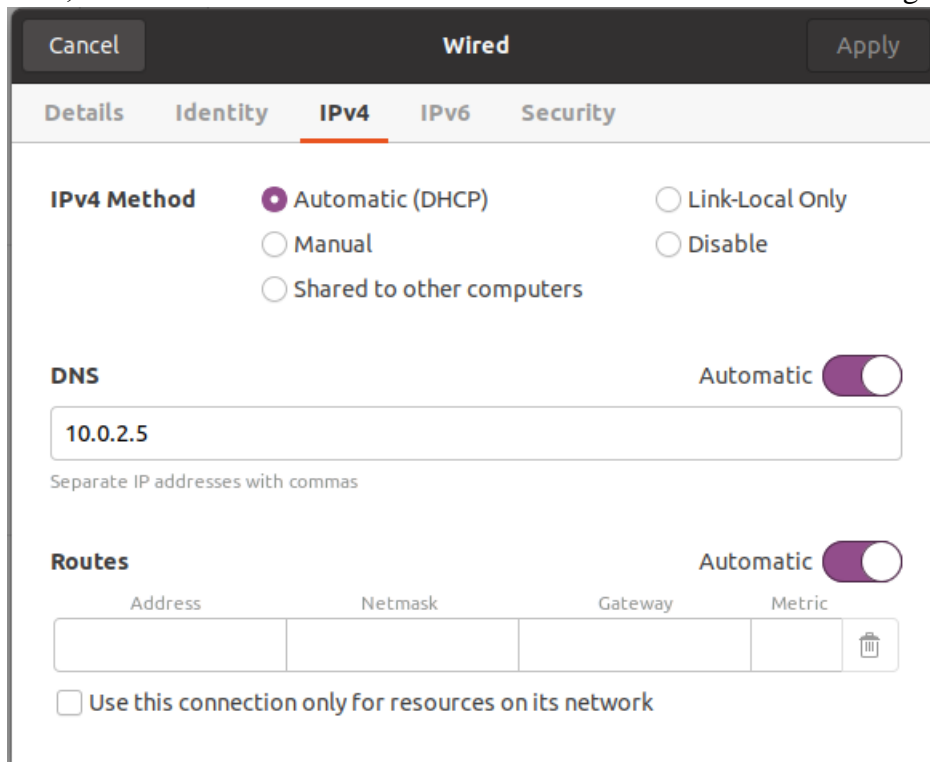
nameserver 10.2.22.184

Run the following command for the change to take effect.
sudo resolvconf -u

The following screenshot shows how to set DNS server on the client machine.

```
gaurav@gaurav-VirtualBox:~$ sudo nano /etc/resolvconf/resolv.conf.d/head
[sudo] password for gaurav:
gaurav@gaurav-VirtualBox:~$ sudo resolvconf -u
gaurav@gaurav-VirtualBox:~$
```

Also, add 10.2.22.184 in ‘Additional DNS servers’ field in IPv4 settings of client machine.



Observation 2:

Ping a computer such as www.google.com. Please use Wireshark to show the DNS query triggered by your ping command and DNS response. Describe your observation. (Take a screenshot).

No.	Source	Destination	Protocol	Length	Info
569	15.908067523	127.0.0.1	DNS	75	Standard query 0x1cd5 A oosp.pki.goog
570	15.908091044	127.0.0.1	DNS	89	Standard query 0xff61 A googleads.g.doubleclick.net
571	15.908142425	127.0.0.1	DNS	75	Standard query 0xa9d7 AAAA oosp.pki.goog
572	15.908146191	127.0.0.1	DNS	84	Standard query 0x3d23 A adservice.google.co.in
573	15.908188676	127.0.0.1	DNS	84	Standard query 0x8424 AAAA adservice.google.co.in
574	15.908628438	127.0.0.53	DNS	84	Standard query 0x94c5 A adservice.google.co.in
575	15.909153273	127.0.0.1	DNS	126	Standard query response 0x1cd5 A oosp.pki.goog CNAME pki-goog-
576	15.909345398	10.0.2.4	DNS	89	Standard query 0xc364 AAAA googleads.g.doubleclick.net
577	15.909526323	127.0.0.53	DNS	108	Standard query 0xd8d6 A googleads.g.doubleclick.net OPT
578	15.909555590	10.0.2.4	DNS	138	Standard query response 0xa9d7 AAAA oosp.pki.goog CNAME pki-g-
579	15.909768433	10.0.2.4	DNS	95	Standard query 0x82cb A adservice.google.co.in OPT
580	15.909889242	10.0.2.4	DNS	95	Standard query 0x14ca AAAA adservice.google.co.in OPT
581	15.913475976	49.205.72.130	DNS	108	Standard query 0x96ba AAAA googleads.g.doubleclick.net OPT
582	15.913476506	49.205.72.130	DNS	103	Standard query response 0x14ca AAAA adservice.google.co.in CN
583	15.913476606	49.205.72.130	DNS	151	Standard query response 0x82cb A adservice.google.co.in CNAME
584	15.913476811	49.205.72.130	DNS	116	Standard query response 0xd8d6 A googleads.g.doubleclick.net
585	15.913826232	127.0.0.53	DNS	128	Standard query response 0x96ba AAAA googleads.g.doubleclick.n
586	15.913979694	127.0.0.1	DNS	152	Standard query response 0x9424 AAAA adservice.google.co.in CN
			DNS	117	Standard query response 0xc364 AAAA googleads.g.doubleclick.n

Task 2: Set Up a Local DNS Server

Note: If bind9 server is not already installed, install using the command

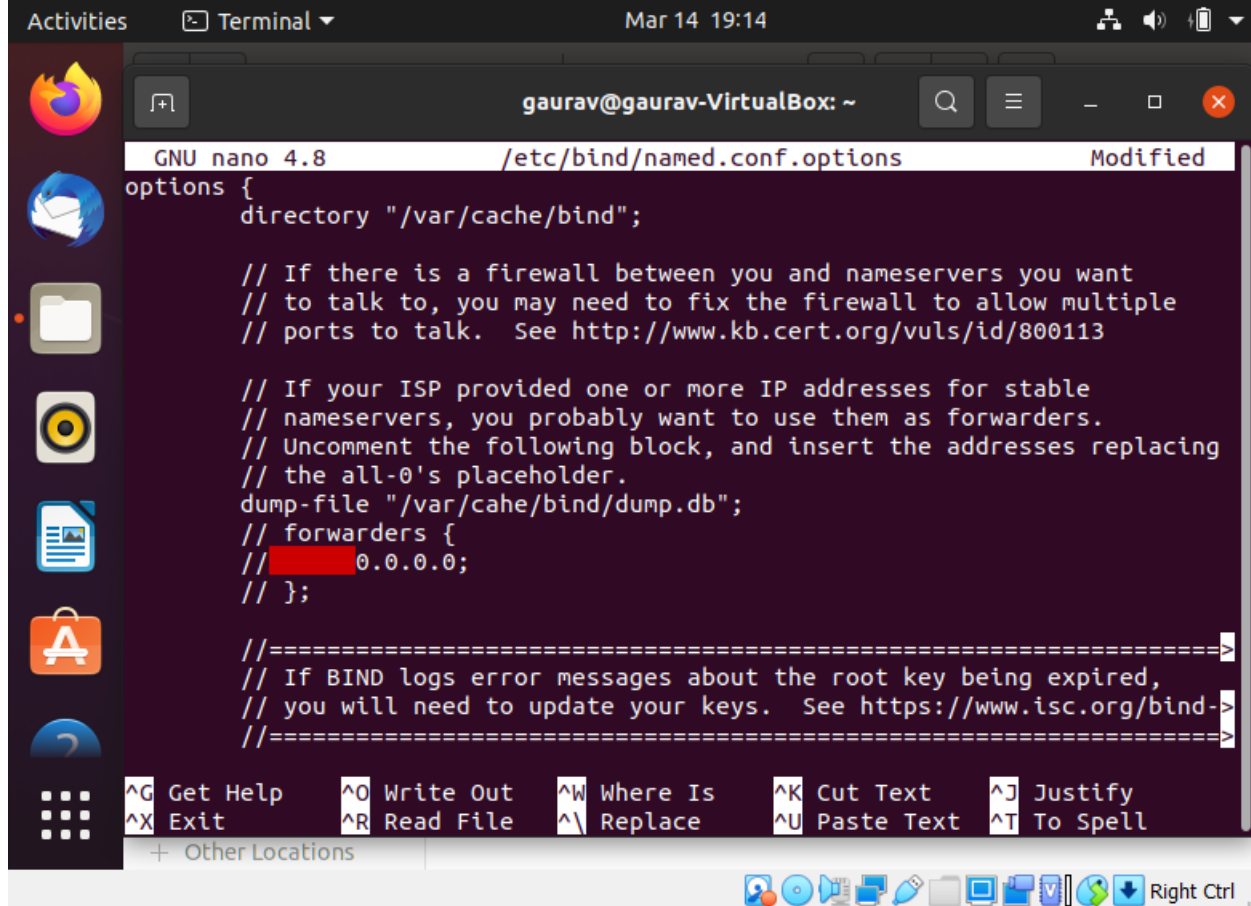
\$ sudo apt-get update

\$ sudo apt-get install bind9

Step 1: Configure the BIND9 Server.

BIND9 gets its configuration from a file called **/etc/bind/named.conf**. This file is the primary configuration file, and it usually contains several “include” entries. One of the included files is called **/etc/bind/named.conf.options**. This is where we typically set up the configuration options. Let us first set up an option related to DNS cache by adding a dump-file entry to the options block. The above option specifies where the cache content should be dumped to if BIND is asked to dump its cache.

```
gaurav@gaurav-VirtualBox:~$ sudo nano /etc/bind/named.conf.options
gaurav@gaurav-VirtualBox:~$
```



```
GNU nano 4.8 /etc/bind/named.conf.options Modified
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.
    dump-file "/var/cahe/bind/dump.db";
    // forwarders {
    // 0.0.0.0;
    // };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-
    //=====
```

The above option specifies where the cache content should be dumped to if BIND is asked to dump its cache. If this option is not specified, BIND dumps the cache to a default file called `/var/cache/bind/named_dump.db`.

Step 2: Start DNS server

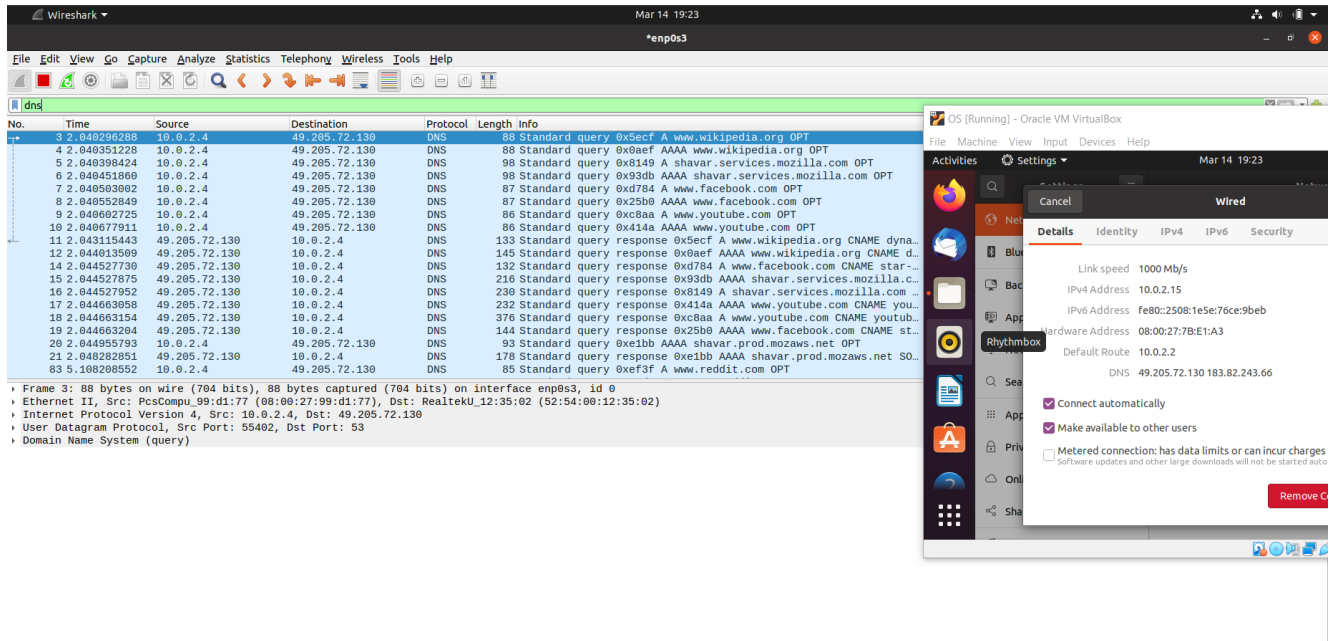
We start the DNS server using the command:

\$ sudo service bind9 restart

```
gaurav@gaurav-VirtualBox:~$ sudo service bind9 restart
gaurav@gaurav-VirtualBox:~$
```

Observation 3:

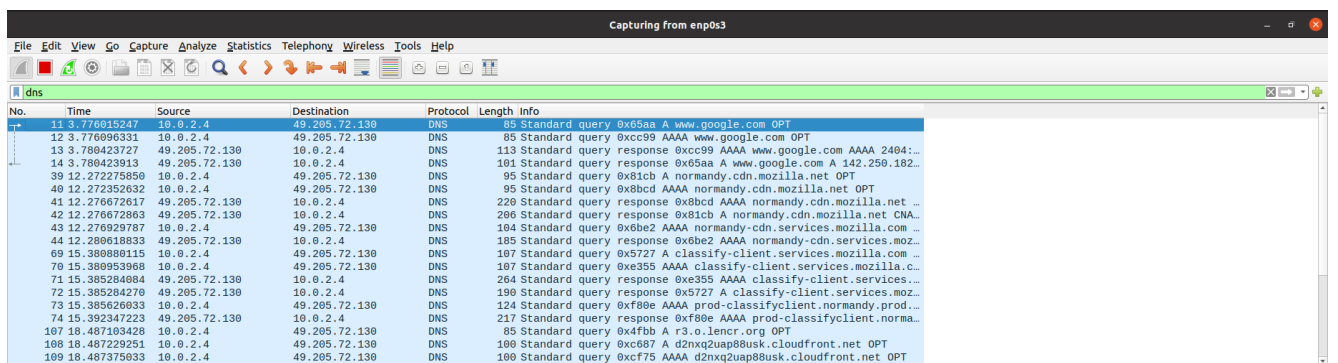
Now, go back to your user machine (10.2.22.195), and ping a computer such as www.google.com and describe your observation. Please use Wireshark to show the DNS query triggered by your ping command. Please also indicate when the DNS cache is used. (Take a screenshot).



Observation 4:

The two commands shown below are related to DNS cache. The first command dumps the content of the cache to the file specified above, and the second command clears the cache. You need extract the DNS cache using ‘grep’ command and take screenshot of www.google.com DNS cache.

```
gaurav@gaurav-VirtualBox:~$ sudo rndc dumpdb -cache
gaurav@gaurav-VirtualBox:~$ sudo rndc flush
gaurav@gaurav-VirtualBox:~$
```



Note: Compare the above three Wireshark DNS packet capture screenshots taken above.

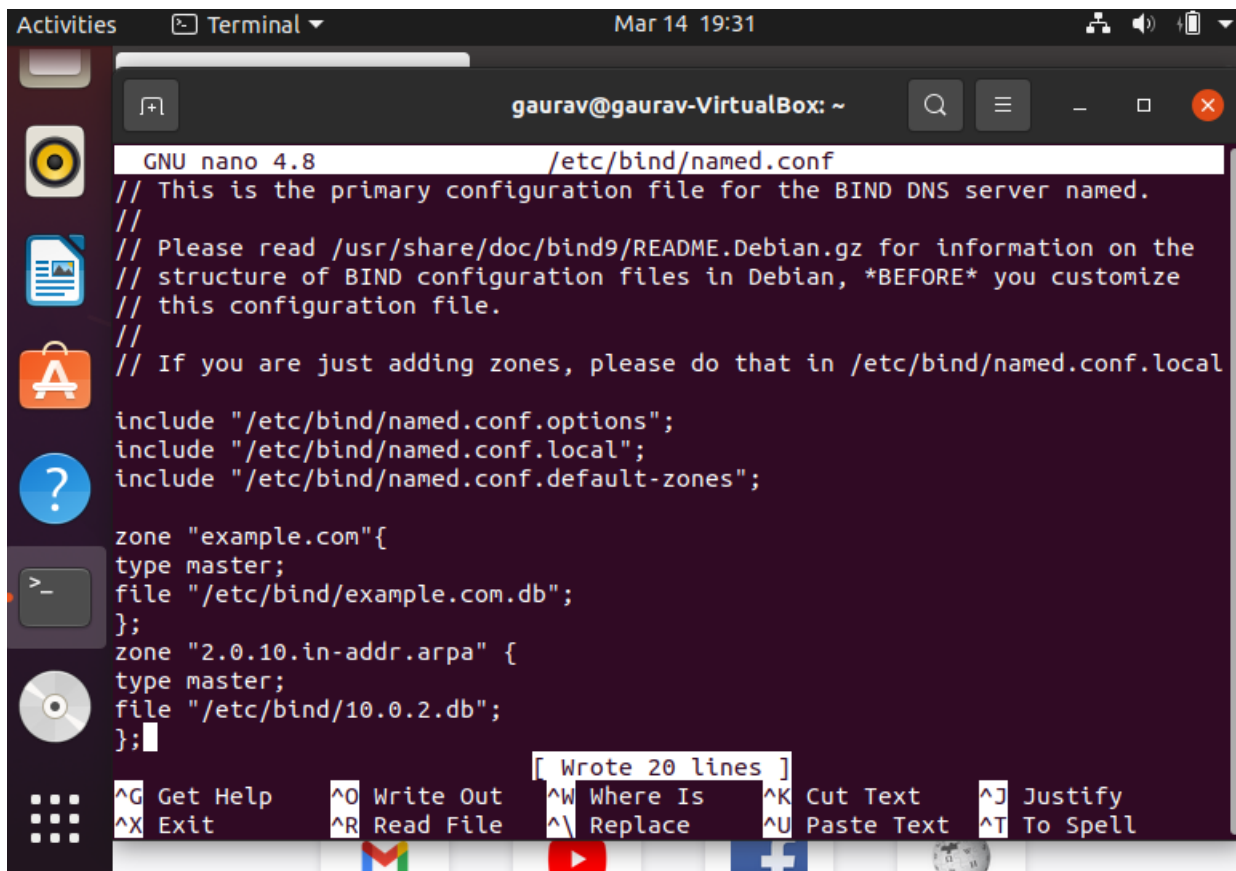
Part 2: Setting Up an Authoritative Nameserver for example.com domain

Task 3: Host a Zone in the Local DNS server.

Assume that we own a domain, we will be responsible for providing the definitive answer regarding this domain. We will use our local DNS server as the authoritative nameserver for the domain. In this lab, we will set up an authoritative server for the **example.com** domain. This domain name is reserved for use in documentation, and is not owned by anybody, so it is safe to use it.

Step 1: Create Zones

We had two zone entries in the DNS server by adding the following contents to **/etc/bind/named.conf** as shown in the below screenshot. The first zone is for forward lookup (from hostname to IP), and the second zone is for reverse lookup (from IP to hostname).



The screenshot shows a terminal window titled "gaurav@gaurav-VirtualBox: ~" with a date and time of "Mar 14 19:31". The terminal is running the GNU nano 4.8 editor, editing the file **/etc/bind/named.conf**. The configuration file contains the following content:

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "example.com"{
type master;
file "/etc/bind/example.com.db";
};
zone "2.0.10.in-addr.arpa" {
type master;
file "/etc/bind/10.0.2.db";
};
```

At the bottom of the terminal, there is a status bar showing "Wrote 20 lines" and a list of keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^X Exit, ^R Read File, ^\ Replace, ^U Paste Text, ^T To Spell.

Note: In above screenshot, 10.2.22.0 is the subnet mask of your IP address. This applies to all part of the experiment.

Step 2: Setup the forward lookup zone file

We create **example.com.db** zone file with the following contents in the **/etc/bind/** directory where the actual DNS resolution is stored.

```

1 $TTL 3D
2 @      IN      SOA      ns.example.com. admin.example.com. (
3        2008111001
4        8H
5        2H
6        4W
7        1D)
8
9 @      IN      NS       ns.example.com.
10 @     IN      MX       10 mail.example.com.
11
12 www   IN      A        10.0.2.22
13 mail  IN      A        10.0.2.23
14 ns    IN      A        10.0.2.24
15 *.example.com. IN      A        10.0.2.100

```

The symbol '@' is a special notation representing the origin specified in **named.conf** (the string after "zone"). Therefore, '@' here stands for **example.com**. This zone file contains 7 resource records (RRs), including a SOA (Start Of Authority) RR, a NS (Name Server) RR, a MX (Mail eXchanger) RR, and 4 A (host Address) RRs.

Step 3: Setup the reverse lookup zone file

We create a reverse DNS lookup file called **10.0.2.db** for the example.net domain to support DNS reverse lookup, i.e., from IP address to hostname in the **/etc/bind/** directory with the following contents.

10.0.2.db	example.com.db
1 \$TTL 3D	
2 @ IN SOA ns.example.com. admin.example.com. (
3 2008111001	
4 8H	
5 2H	
6 4W	
7 1D)	
8 @ IN NS ns.example.com.	
9	
10 101 IN PTR www.example.com.	
11 102 IN PTR mail.example.com.	
12 10 IN PTR ns.example.com.	

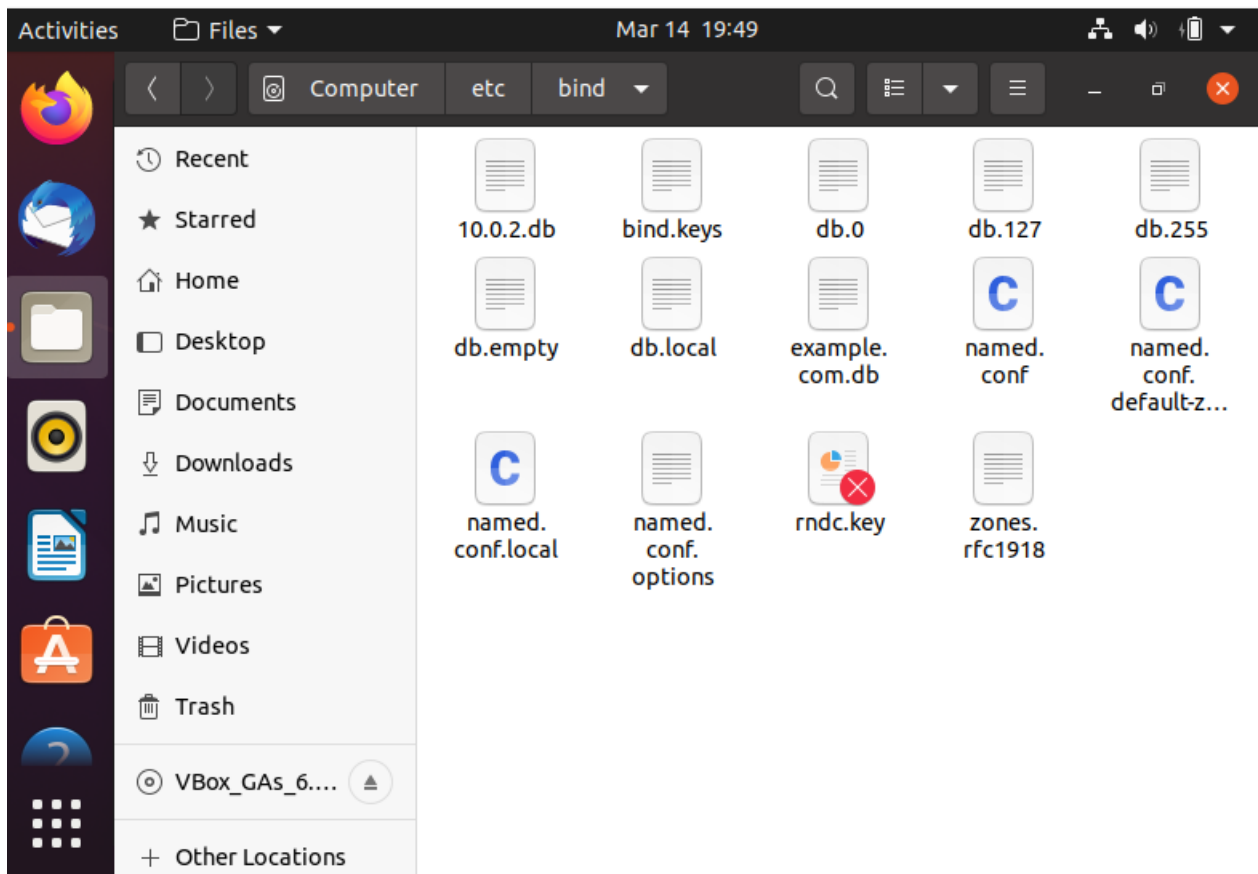
Note: You can download the above two db files from Edmodo. Indent spacing is essential.

Step 4: Copy the above files into **/etc/bind** location.

```

gaurav@gaurav-VirtualBox:~$ sudo cp 10.0.2.db /etc/bind
[sudo] password for gaurav:
gaurav@gaurav-VirtualBox:~$ sudo cp example.com.db /etc/bind
Show Applications gaurav@gaurav-VirtualBox:~$

```

Task 4: Restart the BIND server and test

Step 1: When all the changes are made, remember to restart the BIND server. Now we will restart the DNS server using the following command:

\$ sudo service bind9 restart

```
isfcr@isfcr-H110M-H:~$ sudo service bind9 restart
isfcr@isfcr-H110M-H:~$
```

Step 2: Now, go back to the client machine and ask the local DNS server for the IP address of www.example.com using the dig command.

Dig stands for (Domain Information Groper) is a network administration command-line tool for querying DNS name servers. It is useful for verifying and troubleshooting DNS problems and also to perform DNS lookups and displays the answers that are returned from the name server that were queried. dig is part of the BIND domain name server software suite.


```

gaurav@gaurav-VirtualBox:~$ dig www.example.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 41790
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags::; udp: 65494
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                356     IN      A      93.184.216.34

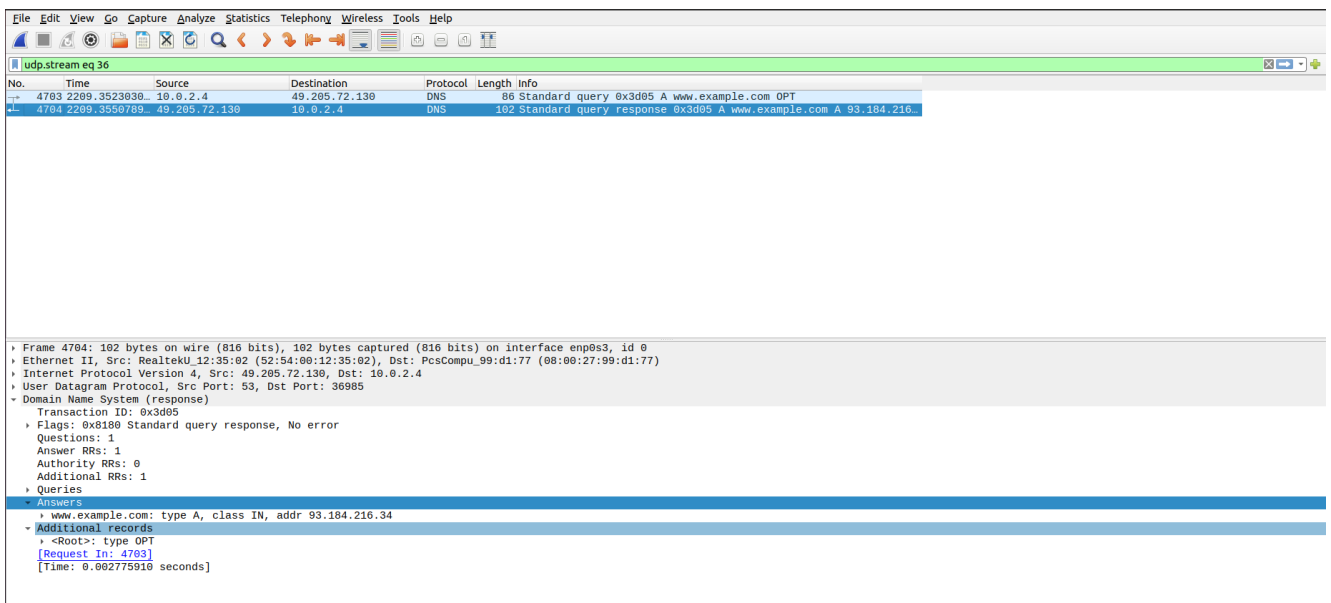
;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Mon Mar 14 20:36:31 IST 2022
;; MSG SIZE rcvd: 60

gaurav@gaurav-VirtualBox:~$

```

We can see that the ANSWER SECTION contains the DNS mapping. We can see that the IP address of www.example.com is now 10.2.22.101, which is what we have setup in the DNS server.

Step 3: Observe the results in Wireshark capture.



To load and clear DNS cache, use the below commands.

```

gaurav@gaurav-VirtualBox:~$ sudo rndc dumpdb -cache
gaurav@gaurav-VirtualBox:~$ sudo rndc flush
gaurav@gaurav-VirtualBox:~$

```

```

1;
2; Start view _default
3;
4;
5; Cache dump of view '_default' (cache _default)
6;
7; using a 604800 second stale ttl
8 $DATE 20220226133151
9; secure
10 .                1123046      IN NS    a.root-servers.net.
11                1123046      IN NS    b.root-servers.net.
12                1123046      IN NS    c.root-servers.net.
13                1123046      IN NS    d.root-servers.net.
14                1123046      IN NS    e.root-servers.net.
15                1123046      IN NS    f.root-servers.net.
16                1123046      IN NS    g.root-servers.net.
17                1123046      IN NS    h.root-servers.net.
18                1123046      IN NS    i.root-servers.net.
19                1123046      IN NS    j.root-servers.net.
20                1123046      IN NS    k.root-servers.net.
21                1123046      IN NS    l.root-servers.net.
22                1123046      IN NS    m.root-servers.net.
23; secure
24                1123046      RRSIG   NS 8 0 518400 (
25                20220318050000 20220305040000 9799 .
26                J+F4rD6WQQWJCFvr+GPQ0GhU9sJkNgdGGEaV
27                LRMrpufULqnEJnfhRPhU4tEsJvq/GIcfZw1W
28                Pv1rIQfem11aepTsQ/mHbn8h2n5pslm6fSvJ
29                CzXsDHM00904AzCIVrjDZQpcM92aWVUdxRy+
30                XPzZPyK5Ge+MQXjRB/yV+3IaBdUIEjxmCw0a
31                arGgvKJ9ufu1ABw29qDGmdMAoPuZk+gno3d4
32                cQGsBPd6/50XE1PA800xIvufuqCDKJ3HxFLB
33                VN0wnmMGvtTSU7L3ITEAWKjHK1PJHLTb/ckC
34                yScTf0XQYiXGozpE+yfdt/tfNvwC/24MDBtR
35                AI23MvrJAOnNBIBHQ== )
36; secure
37                777446  DNSKEY  256 3 8 (

```

Edmodo Requirements:

- 1) Wireshark packet capture screenshots (Observations 1-3)
- 2) DNS cache for www.google.com (Observation 4)
- 3) **dig www.example.com** command (in Terminal)
- 4) Wireshark packet capture – **dig www.example.com** command
- 5) DNS cache on server machine after dig command