LINUX PROJECT

GROUP 3

- **Deepak Hariharan**
- **Rinky Upadhyay**
- **Murugappan Arunachalam Annamalai**
- **Mohammed Shahid Abdul Bashir**

CONTENTS

WEB SERVER

Apache is the most commonly used Web Server on Linux systems. Web Servers are used to serve Web Pages requested by client computers.

BEHAVIOR OF THE PROTOCOL:

The protocol used to transfer Web pages is the Hyper Text Transfer Protocol (HTTP). HTTP uses port 80. By writing an URL to our browser, from this URL, our browser knows which server to contact and what file to ask for.

This is exactly where the http protocol starts: connect a server and transfer a file!

During an http transaction, the client asks for a file to a server.

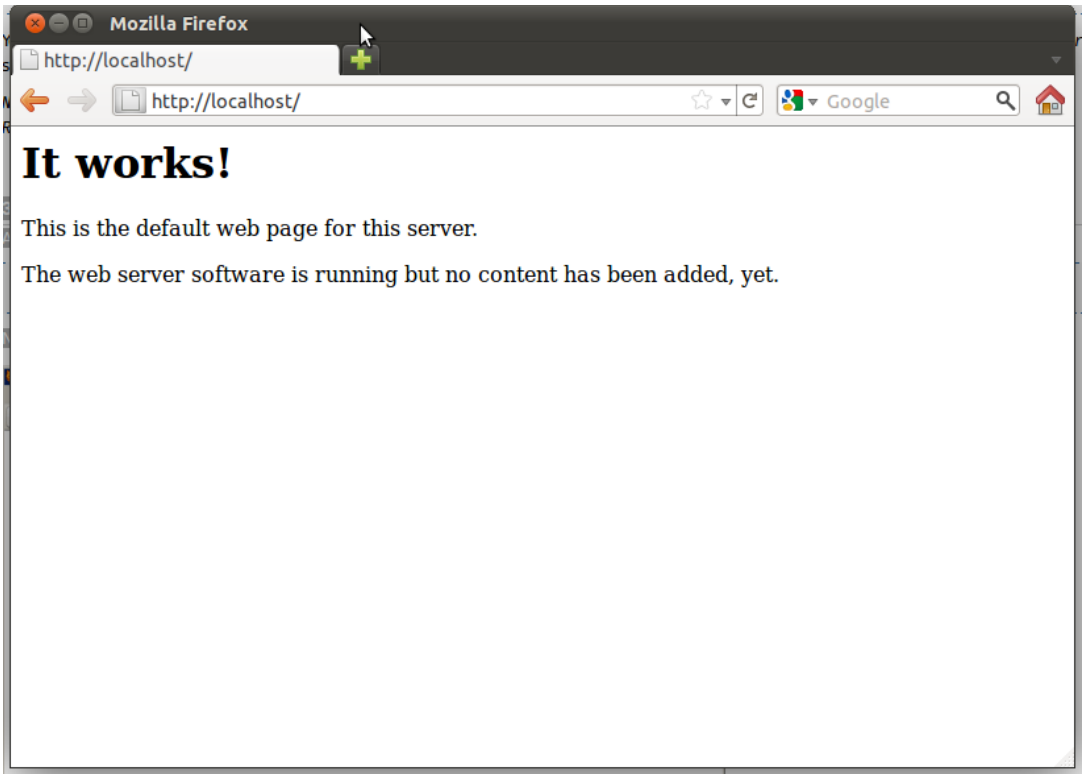
- Client connects to host
- Server accepts connection
- Client request a file
- Server sends a response

IMPLEMENTATION COMMAND USED and TESTING:

Web Server Installation:

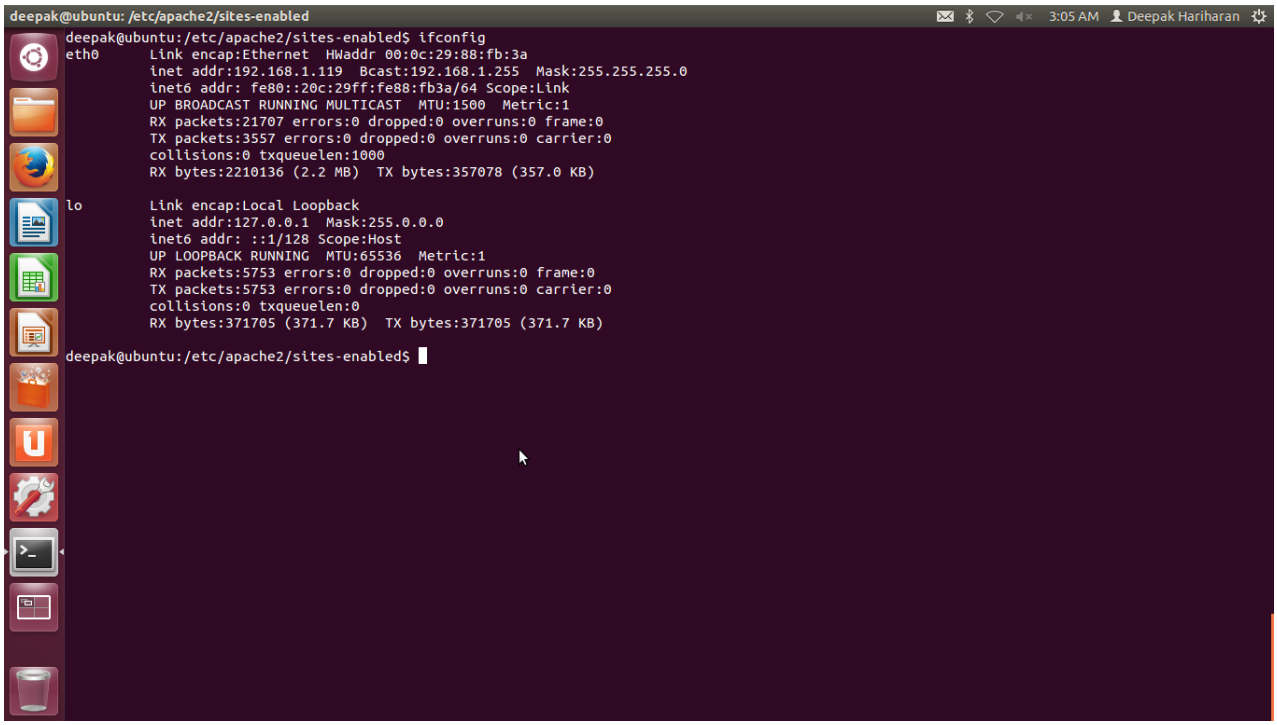
- The Apache Web Server is installed in the Linux machine using the command **sudo apt-get update**
- The command updates the package repositories of Ubuntu Linux to latest versions. **sudo apt-get install apache2**
- The command installs the apache2 package in the Linux machine as the super user.

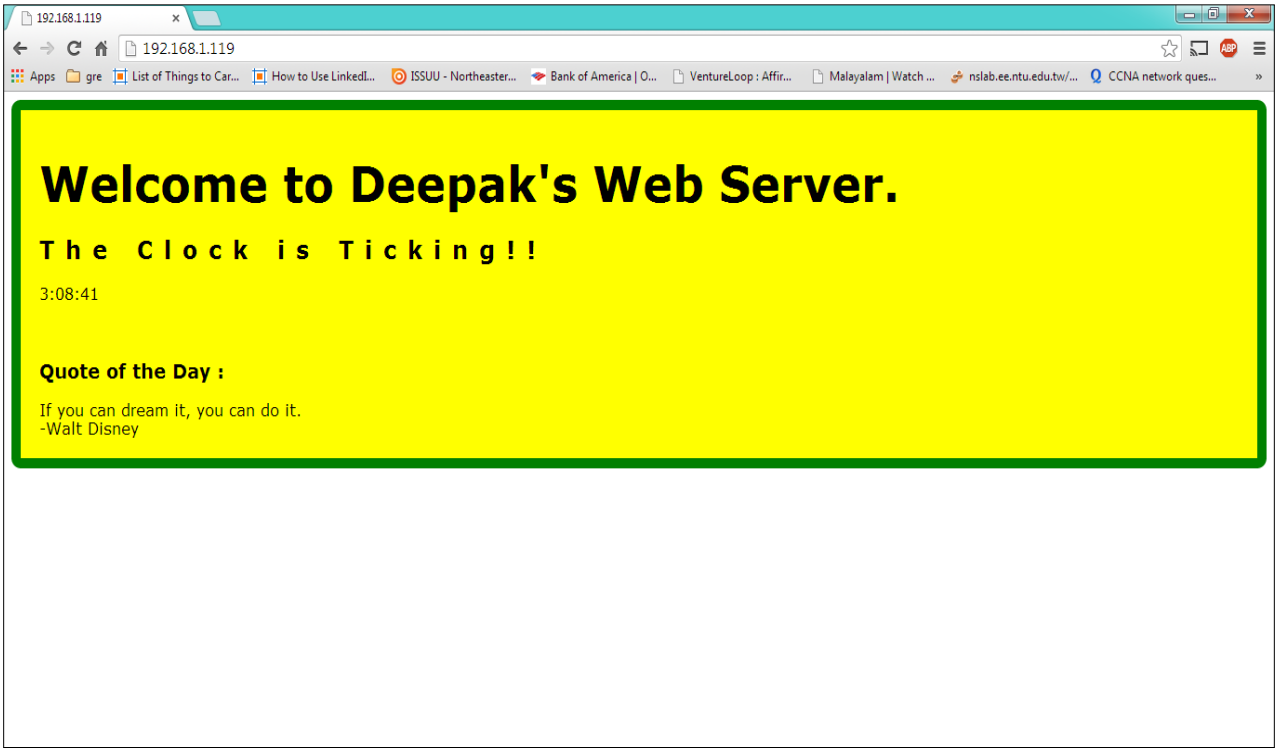
- To verify if the Apache Web Server has been installed properly, open a web browser and enter "**localhost**" in the address bar. The below web page will be loaded in the browser, which is hosted by the web server. If the below web page is loaded in the browser, the server is up and running fine. Next we need to configure the firewall and make the web server more secure.



- The above web page is hosted in **/var/www/index.html** file path. The web page can be modified to change its contents using any of the editor
  sudo gedit /var/www/index.html
  The above command opens the source of the web page in the Gedit text editor

- The web server was accessible by all the clients connected to my network using the web browser. The below screenshots show the IP address of the web server in the Linux and how the web server was accessed using the web server IP Address in another client in the network

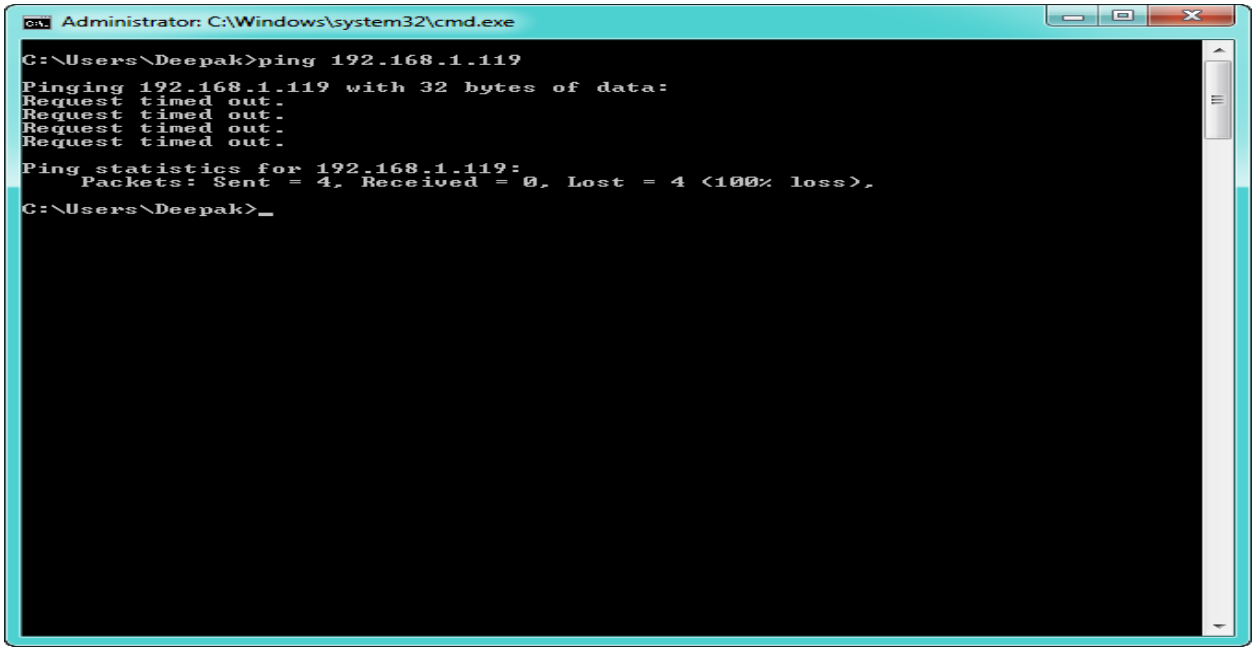

deepak@ubuntu: /etc/apache2/sites-enabled
3:05 AM
Deepak Hariharan
deepak@ubuntu:/etc/apache2/sites-enabled$ ifconfig
eth0 Link encap:Ethernet HWaddr 00:0c:29:88:fb:3a
inet addr:192.168.1.119 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fe88:fb3a/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:21707 errors:0 dropped:0 overruns:0 frame:0
TX packets:3557 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2210136 (2.2 MB) TX bytes:357078 (357.0 KB)
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:5753 errors:0 dropped:0 overruns:0 frame:0
TX packets:5753 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:371705 (371.7 KB) TX bytes:371705 (371.7 KB)
deepak@ubuntu:/etc/apache2/sites-enabled$


192.168.1.119
192.168.1.119
Apps
gre
List of Things to Car...
How to Use LinkedI...
ISSUU - Northeaster...
Bank of America | O...
VentureLoop : Affir...
Malayalam | Watch ...
nslab.ee.ntu.edu.tw/...
CCNA network ques...
Welcome to Deepak's Web Server.
The Clock is Ticking!!
3:08:41
Quote of the Day :
If you can dream it, you can do it.
-Walt Disney

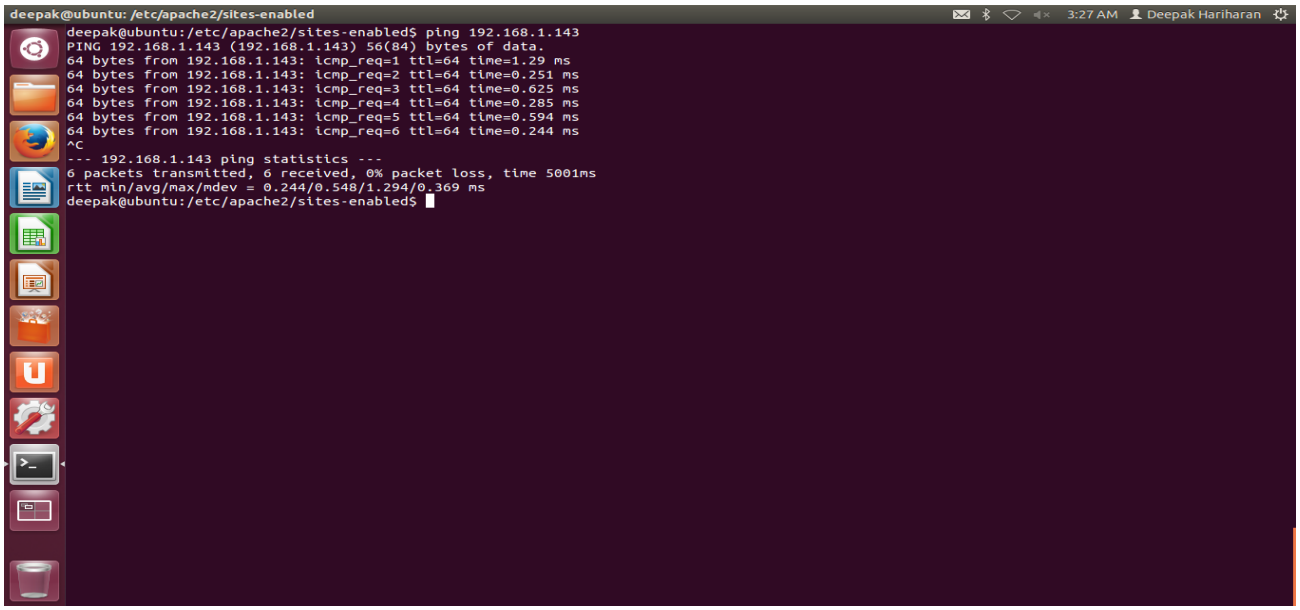
Testing the Web Server

Ping test:

The web server is pinged from a client in the network. However since the web server doesn't accept incoming icmp echo requests due to the rules set in the iptables, the icmp echo requests are timed out as shown in the screenshot.



Similarily the web server allows outgoing icmp echo requests and incoming icmp echo replies. Hence the web server is able to ping the client using the ping command. The below screenshot shows the echo response by the client to the web server



FIREWALL

Firewall Configuration

The firewall can be setup for the web server using the Iptables which are pre installed in the Linux machine. Iptables firewall is used to manage packet filtering and NAT rules. Iptables comes with all Linux distributions. Two chains are being used to set the firewall for the Web server – Input Chain to filter the incoming packets and the Output chain to filter the outgoing packets. Initially the Iptables will be empty and consist of no rules

The Iptables firewall rules are set by executing the following commands in the Terminal:

At first, the iptables are flushed to remove any existing iptable by the command
sudo iptables –F

Input
sudo iptables –A INPUT –j ACCEPT –p tcp --destination-port 80 –i eth0
The rule accepts all incoming TCP port 80 connections to the network interface eth0. In other words, the rule accepts all incoming HTTP requests to the server at port 80 for the network interface named eth0.
sudo iptables –A INPUT –j ACCEPT –p tcp --destination-port 443 –i eth0
The rule accepts all incoming TCP port 443 connections to the network interface eth0. In other words, the rule accepts all incoming HTTPS requests to the server at port 443 for the network interface named eth0.
sudo iptables –A INPUT –j DROP –p tcp –i eth0
The rule drops all incoming TCP connections to the network interface eth0. In other words, the rule drops all incoming TCP connections to any port for the network interface eth0.
sudo iptables –A INPUT –j ACCEPT –p icmp --icmp-type echo reply
The rule accepts all incoming icmp echo replies(icmp ping reply) to the server.
sudo iptables –A INPUT –j DROP –p icmp
The rule drops all incoming icmp connection to the web servers
sudo iptables –A INPUT –j DROP –p udp
The rule drops all incoming udp connections.

Output Rules
sudo iptables –A INPUT –j ACCEPT –p icmp --icmp-type echo request
The rule allows all outgoing icmp echo requests (icmp ping requests) from the server.
sudo iptables –A INPUT –j DROP –p icmp
The rule blocks all outgoing icmp connections from the server
sudo iptables –A INPUT –j DROP –p UDP
The rule blocks all outgoing UDP connections from the server

Once the above codes are entered, the iptables have to be saved and restore every time the server is restarted. Iptables-persistent can be used for the same. Iptables-persistent is an “init.d” script to make iptables rules persistent over reboots. Iptables-persistent can be installed using the command:

sudo apt-get install iptables-persistent

Once the iptables-persistent is installed, we can save the iptables rules to iptables-persistent rules of IPv4 rules by the command.
sudo sh –c "iptables-save > /etc/iptables/rules.v4"

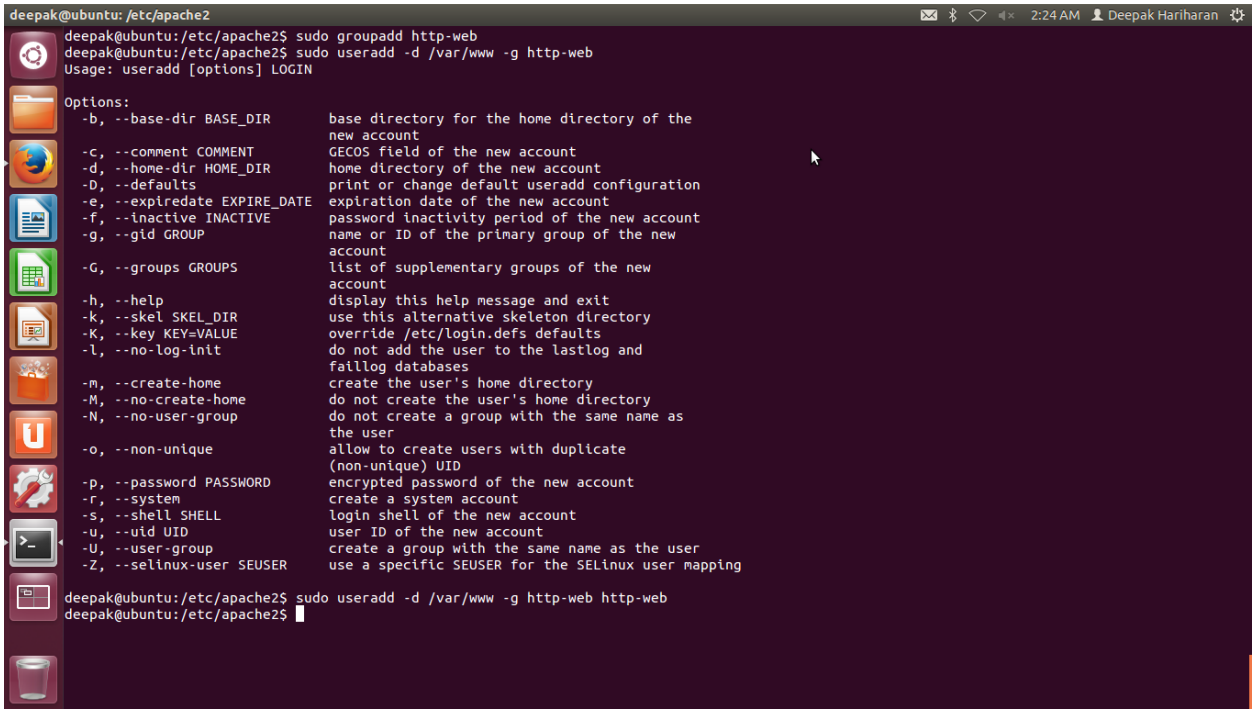
The above code saves the iptables to iptables-persistent IPv4 rules and loads the iptables rules every time iptables-persistent service starts on booting of the linux server. The iptables-persistent service can be started at any point to load the saved iptables by the command:
sudo service iptables-persistent start

Changes Made to Files/Folders

- **Run Apache as Separate user and Separate Group**

  A separate user and user group is created to run the apache web server. When under network attack, the root access will be protected and inaccessible. The apache user and group is created using the below commands:

  sudo groupadd http-web
  The above command creates a group http-**web**
  sudo useradd –d /var/www –g http-web http-web
  The above command creates a user http-web who is added to the group http-web

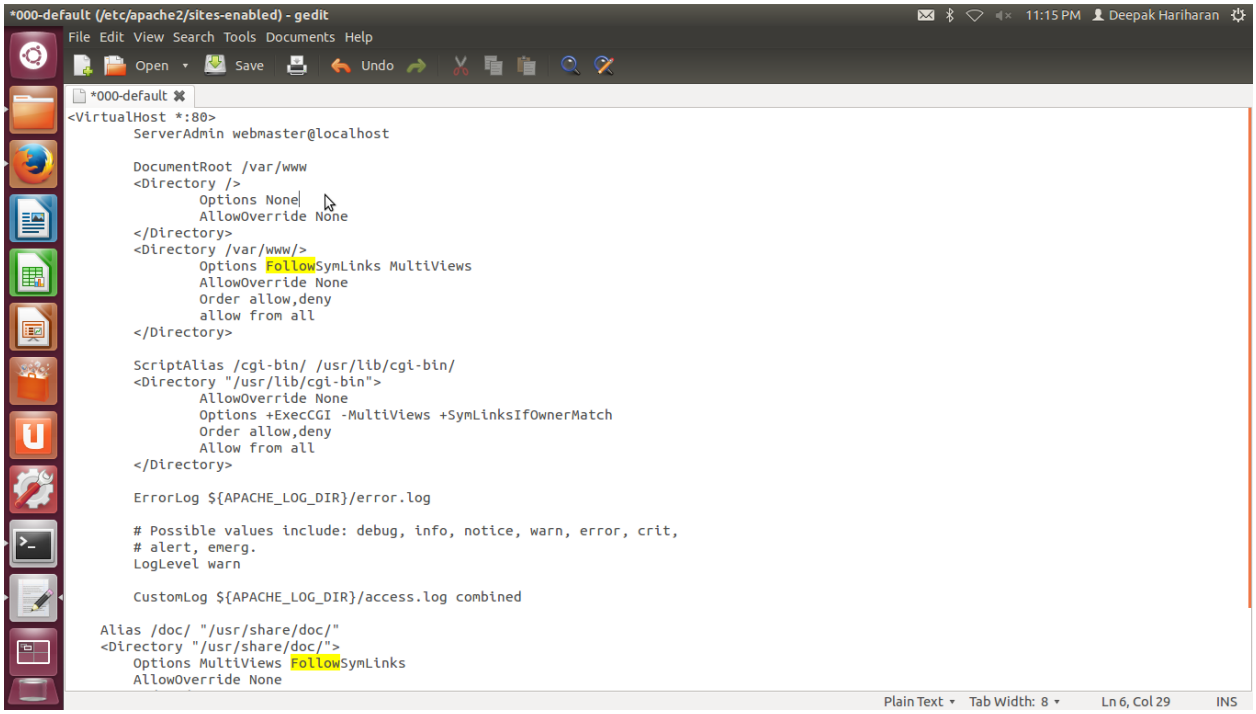


We can verify the same by using the command
ps –ef | grep –i http | awk '{print $1'}'

We can find the apache web server is running as the user http-web

- **Restrict access to root directory**

We can restrict access to directories with "Allow" and "Deny" options in 000-default file in /etc/apache2/sites-enabled. By removing the **FollowSymLinks** options to none in **<Directory />**, the root access can be restricted. The code is set as shown below in the screenshot.
Before changing the code:

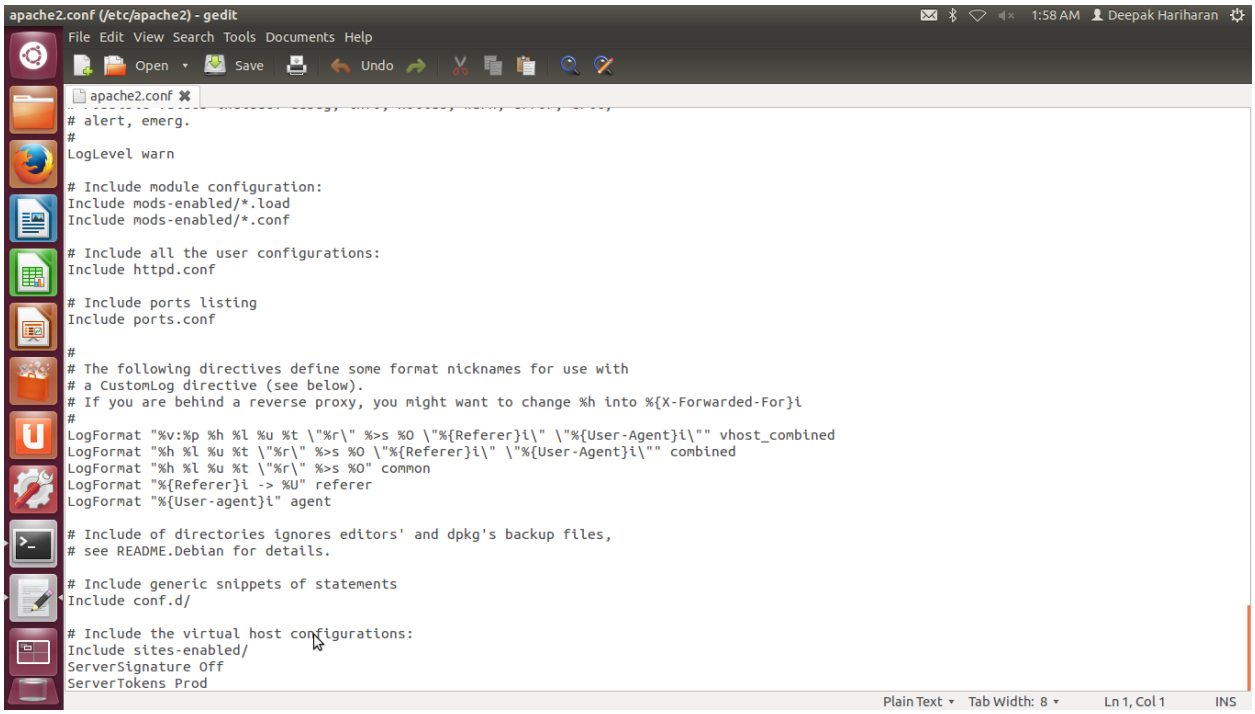


In the above screenshot, for the **<Directory />,** the options is set as **None.** This will prevent the users from enabling any optional features thereby restricting the root access.

- **Hide Apache version and OS Identity in Error page.**

The Apache shows its **version** with the **OS Information and version installed** in the web server. This can be a security threat to your web server as well as your Linux box too. The following changes are made in the files to hide the Apache version and OS Information in the Web Server Error page:

The following code is added at the end of the **apache2.conf** file in **/etc/apache2/**
ServerSignature Off
ServerTokens Prod



In the above command, the Prod option hides the Apache version. The available options for ServerTokens are given below:

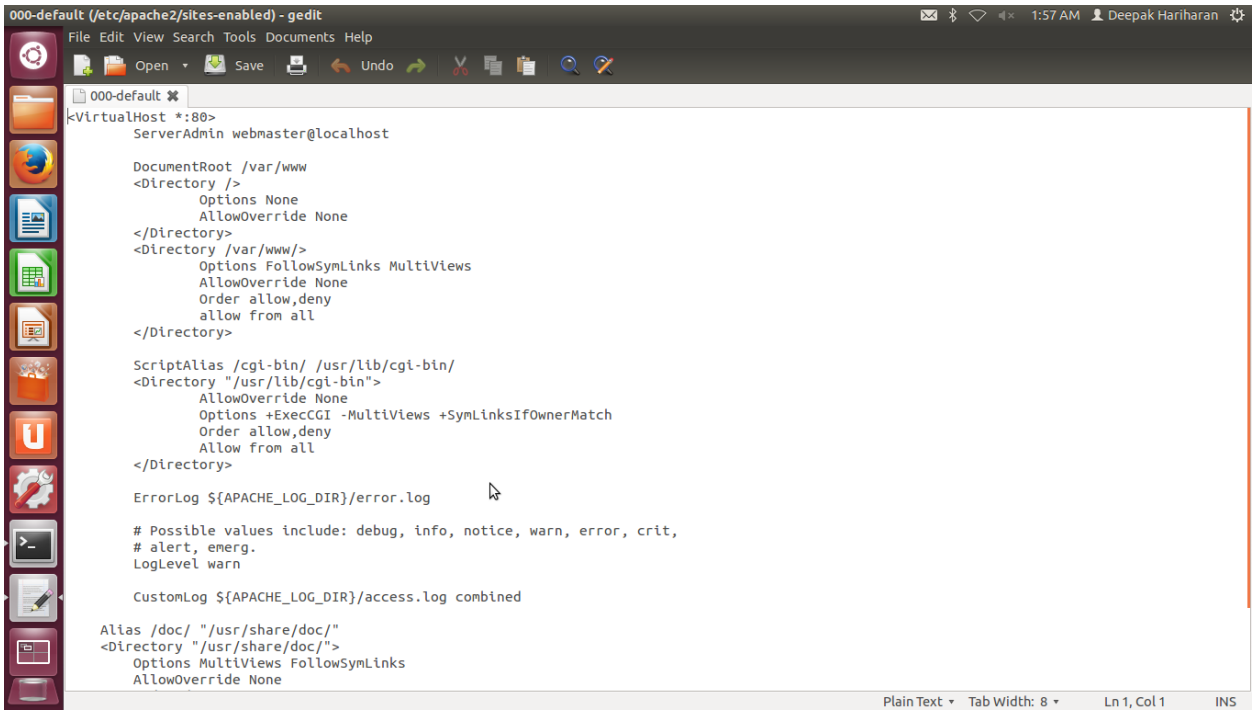
- **ServerTokens Prod** displays "Server: Apache"
- **ServerTokens Major** displays "Server: Apache/2″
- **ServerTokens Minor** displays "Server: Apache/2.2″
- **ServerTokens Min** displays "Server: Apache/2.2.17″
- **ServerTokens Full** displays "Server: Apache/2.2.17 (Unix) PHP/5.3.5″ (If you don't specify any ServerTokens value, this is the default)

Disable Directory Indexing

The user will be able to see all the files and directories under the root if the directory indexing is not disabled. This poses a serious threat to the web server.

If the web server has a missing **index.html**, all the files and directories in the root is visible in the browser. All the files and directories are easily accessible.

To disable directory browsing, we can remove **Indexes from Options directives** in the **000-default** file in **/etc/apache2/sites-enabled/** folder.



Mail Server

Used postfix,courier-imap and squirrelmail

Postfix is the default **Mail Transfer Agent** (MTA) for Ubuntu. It is in Ubuntu's main repository, which means that it receives security updates. This guide explains how to install and configure postfix and set it up as an SMTP server using a secure connection.

Courier is one of the most known mail delivery agent. It only supports Maildir mailboxes and can be integrated with external databases (LDAP, MySQL, etc.).

Squirrelmail is a simple, fast and popular webmail package.

POSTFIX Configuration

To install the postfix, the following command is executed:

sudo apt-get install postfix

To configure postfix, following command is executed,

sudo dpkg-reconfigure postfix
Insert the following details when asked (replacing server1.example.com with your domain name if you have one):

General type of mail configuration: **Internet Site**

NONE *doesn't appear to be requested in current config*
System mail name: **Heisenberg.com**
Root and postmaster mail recipient: **user1,user2**
Other destinations for mail: **mail.Heisenberg.com, Heisenberg.com, localhost**
Force synchronous updates on mail queue?: **No**
Local networks: **127.0.0.0/8, 192.168.1.0/24**
Mailbox size limit (bytes): **0**
Local address extension character: **+**
Internet protocols to use: **all**

The postconf command can be used to configure all postfix parameters. The configuration parameters will be stored in /etc/postfix/main.cf file. Moreover, new mail will be placed in /home/username/Maildir. Hence mail delivery agents need to be configured to the same path

To configure the mailbox format for Maildir:

sudo postconf -e 'home_mailbox = Maildir/'

sudo postconf -e 'mailbox_command ='

Execute the following commands to restart the postfix daemon:

sudo /etc/init.d/postfix restart

COURIER SETUP

The following packages are installed for courier

sudo apt-get install courier-imap

CONFIGURATION

The configuration options are all located in /etc/courier/imapd for imap. For a first installation, the default options perfectly match most of the needs. So no modification will be done.

Maildir setup

Basically, Maildir folders are located in the user home directory. That's a good idea to create Maildir for future users:
sudo maildirmake /etc/skel/Maildir
sudo maildirmake /etc/skel/Maildir/.Drafts
sudo maildirmake /etc/skel/Maildir/.Sent
sudo maildirmake /etc/skel/Maildir/.Trash
sudo maildirmake /etc/skel/Maildir/.Templates

Then, for an existing user:
sudo cp -r /etc/skel/Maildir /home/myuser/

sudo chown -R myuser:usergroup /home/myuser/Maildir
sudo chmod -R 700 /home/myuser/Maildir

SQUIRRELMAIL

Squirrelmail comes with a sample apache configuration file in /etc/squirrelmail/apache.conf.

The file is copied to /etc/apache2/sites-available/squirrelmail with the command:

sudo cp /etc/squirrelmail/apache.conf /etc/apache2/sites-available/squirrelmail

then link it to the sites-enabled directory with the command:

sudo ln -s /etc/apache2/sites-available/squirrelmail /etc/apache2/sites-enabled/squirrelmail

Reload Apache Configuration:

sudo /etc/init.d/apache2 force-reload

Users can be added as follows:

sudo useradd -m -s /bin/bash user1

sudo passwd user1

sudo cp -r /etc/skel/Maildir /home/user1/

sudo chown -R user1:usergroup /home/user1/Maildir

sudo useradd -m -s /bin/bash user2

sudo passwd user2

sudo cp -r /etc/skel/Maildir /home/user2/

sudo chown -R user2:usergroup /home/user2/Maildir

FTP SERVER

The FTP server is installed by executing the following command:

sudo apt-get install vsftpd

To configure vsftpd to authenticate system users and allow them to upload files edit /etc/vsftpd.conf:

local_enable=YES

write_enable=YES

The vsftpd daemon is restarted

sudo /etc/init.d/vsftpd restart

Now when system users login to FTP they will start at their home directories

THE FTP can be made more secure by editing in the vsftpd.conf

chroot_local_user=YES

chroot_list_enable=YES

chroot_list_file=/etc/vsftpd.chroot_list

Uncomment the above options and create a /etc/vsftpd.chroot_list containing a list of users one per line.

Restart vsftpd:

sudo /etc/init.d/vsftpd restart

DNS SERVER:

Description:-

Domain Name System is a distributed naming system which is used by devices when connecting to the internet. DNS helps to translate the IP address to internet domain names. It provides the IP address of web site when we enter it through the browser. A registered system known as DNS server is used to join the Domain System. A DNS server runs special-purpose networking software and it features a public IP addresses. It also contains a database of network names and addresses for other Internet hosts.

In this project the Ubuntu machine is made to work as local DNS server by configuring commands on Linux terminal. The DNS server will act as local DNS. So when the request for any data from the website, system will pass that request to root DNS server. The requested address is resolved by the root DNS.

Configuration steps and explanation

Configure Server With Static IP

It is necessary to configure the DNS with static IP. The reason for this is, when the DHCP fails to renew the IP, or when the timeout happen then the DNS server won't be reached. So it's better to assign static IP to the DNS server.

Edit the file /etc/network/interfaces

Using the sudo command would allow the user to perform admin's privilege. And assign staticIP to the interfaces

```
auto lo
iface lo inet loopback
#iface lo inet6 loopback

auto eth0
iface eth0 inet static
address 192.168.1.110
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1

iface eth0 inet6 static
address fe80::d1
netmask 64
gateway fe80::1
```

.

Check hosts file - Cat hosts

Cat /etc/hostname
At times, it might have been configured incorrectly by network-manager.

Restart network by the following command
sudo /etc/init.d/networking restart

If the files are installed then uninstall network-manager and network-manager-gnome.
Clear all dynamic IP entries out of /etc/hosts by the following
Sudo apt-get remove network-manager network-manager-gnome

Install the DNS Daemon by the following command
sudo apt-get install bind9

Edit BIND Configuration Files
Edit named.conf.options and add forwarders.
sudo nano /etc/bind/named.conf.options

```
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 replaci$
        // the all-0's placeholder.

         forwarders {
                192.168.1.1;
                8.8.8.8;
                8.8.4.4;
         };
```

Restart the bind using the bellow command to check if the configuration is correct.

```
heisenberguser@Heisenberg:/$ sudo /etc/init.d/bind9 restart
 * Stopping domain name service... bind9
waiting for pid 1018 to die
                                                                    [ OK ]
 * Starting domain name service... bind9                           [ OK ]
heisenberguser@Heisenberg:/$
```

Configure the named.conf.local file with forward and reverse lookup zones
The forward lookup zone resolves hostname to IP address translation and reverse lookup is used

to resolve IP address to hostname translation

```
//include "/etc/bind/zones.rfc1918";
#Forward Lookup Zone
zone "Heisenberg.com"{
        type master;
        file "/etc/bind/Heisenberg.com.db";
        allow-transfer { 192.168.1.111; };
        notify yes;
        also-notify { 192.168.1.111; };
};

#Reverse Lookup Zone
zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192";
        allow-transfer { 192.168.1.111; };
        notify yes;
        also-notify { 192.168.1.111; };
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa" {
        type master;
        file "/etc/bind/db.ipv6";
        allow-transfer { 192.168.1.111;};
        notify yes;

};
```

Create the Forward Lookup Zones database file using the following command
sudo nano /etc/bind/Heisenber.com.db

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.Project. murugappan89.gmail.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.Project.com.
client1 IN      A       192.168.1.10
client2 IN      A       192.168.1.15
client3 IN      A       192.168.1.20
client4 IN      A       192.168.1.25
client5 IN      A       192.168.1.30
@       IN      AAAA    ::1
ns      IN      A       192.168.1.5

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Create the Reverse Lookup Zones database file using the following command
sudo nano /etc/bind/bd.192

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     Heisenberg.com. root.Heisenberg.com. (
                              42         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      Heisenberg.com.
110     IN      PTR     ns.Heisenberg.com.
111     IN      PTR     ns1.Heisenberg.com.
139     IN      PTR     Heisenberg.com.
140     IN      PTR     win7pc1.Heisenberg.com.
145     IN      PTR     win7pc2.Heisenberg.com.
150     IN      PTR     win7pc3.Heisenberg.com.
155     IN      PTR     win7pc4.Heisenberg.com.
116     IN      PTR     gappa.com.
```

sudo nano /etc/bind/bd.ipv6

```
; start of 7.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa
$TTL 604800 ;
@       SOA Heisenberg.com. root.Heisenberg.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

@       IN      NS      ns.Heisenberg.com.
1.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     ns.Heisenberg.com.
2.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     ns1.Heisenberg.com
7.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     Heisenberg.com.
0.4.1.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     win7pc1.Heisenberg.com.
5.4.1.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     win7pc2.Heisenberg.com.
0.5.1.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     win7pc3.Heisenberg.com.
5.5.1.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         IN      PTR     win7pc4.Heisenberg.com.

; end of zone file
```

Restart the bind to check if the configuration is correct.

```
heisenberguser@Heisenberg:/$ sudo /etc/init.d/bind9 restart
 * Stopping domain name service... bind9
waiting for pid 1018 to die
                                                                    [ OK ]
 * Starting domain name service... bind9                           [ OK ]
heisenberguser@Heisenberg:/$
```

Now the forward and reverse zone files are configures. It can be checked by using the below

command
named-checkzone Heisenberg.com /etc/bind/Heisenberg.com.bd
named-checkzone Heisenberg.com /etc/bind/db.192
named-checkzone Heisenberg.com /etc/bind/db.ipv6

```
heisenberguser@Heisenberg:/etc/bind$ named-checkzone Heisenberg.com.db /etc/bind/Heisenberg.com.db
/etc/bind/Heisenberg.com.db:14: ignoring out-of-zone data (ns.Heisenberg.com)
/etc/bind/Heisenberg.com.db:15: ignoring out-of-zone data (ns.Heisenberg.com)
/etc/bind/Heisenberg.com.db:18: ignoring out-of-zone data (Heisenberg.com)
/etc/bind/Heisenberg.com.db:19: ignoring out-of-zone data (Heisenberg.com)
/etc/bind/Heisenberg.com.db:20: ignoring out-of-zone data (win7pc1.Heisenberg.com)
/etc/bind/Heisenberg.com.db:21: ignoring out-of-zone data (win7pc1.Heisenberg.com)
/etc/bind/Heisenberg.com.db:22: ignoring out-of-zone data (win7pc2.Heisenberg.com)
/etc/bind/Heisenberg.com.db:23: ignoring out-of-zone data (win7pc2.Heisenberg.com)
/etc/bind/Heisenberg.com.db:24: ignoring out-of-zone data (win7pc3.Heisenberg.com)
/etc/bind/Heisenberg.com.db:25: ignoring out-of-zone data (win7pc3.Heisenberg.com)
/etc/bind/Heisenberg.com.db:26: ignoring out-of-zone data (win7pc4.Heisenberg.com)
/etc/bind/Heisenberg.com.db:27: ignoring out-of-zone data (win7pc4.Heisenberg.com)
/etc/bind/Heisenberg.com.db:28: ignoring out-of-zone data (gappa.com)
/etc/bind/Heisenberg.com.db:29: ignoring out-of-zone data (shahidbashir.com)
zone Heisenberg.com.db/IN: loaded serial 43
OK
heisenberguser@Heisenberg:/etc/bind$
```

Edit the configuration file /etc/resolv.conf to reflect the DNS server

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.110
search Heisenberg.com
```

Restart the BIND9 Daemon to update the new configuration
sudo /etc/init.d/bind9 restart

Test
Check forward Zones and reverse zone respectively.
Nslookup Heisenberg.com
Nslookup 192.168.1.139
Dig Heisenberg.com
Dig 192.168.1.139

The outputs are shown below

```
heisenberguser@Heisenberg:/$ nslookup Heisenberg.com
Server:         192.168.1.110
Address:        192.168.1.110#53

Name:   Heisenberg.com
Address: 192.168.1.139

heisenberguser@Heisenberg:/$ nslookup 192.168.1.139
Server:         192.168.1.110
Address:        192.168.1.110#53

139.1.168.192.in-addr.arpa      name = Heisenberg.com.
```

```
heisenberguser@Heisenberg:/$ dig Heisenberg.com

; <<>> DiG 9.8.1-P1 <<>> Heisenberg.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19501
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; QUESTION SECTION:
;Heisenberg.com.                        IN      A

;; ANSWER SECTION:
Heisenberg.com.         604800  IN      A       192.168.1.139

;; AUTHORITY SECTION:
Heisenberg.com.         604800  IN      NS      ns.Heisenberg.com.

;; ADDITIONAL SECTION:
ns.Heisenberg.com.      604800  IN      A       192.168.1.110
ns.Heisenberg.com.      604800  IN      AAAA    fe80::d1

;; Query time: 1 msec
;; SERVER: 192.168.1.110#53(192.168.1.110)
;; WHEN: Sun Apr 13 21:13:30 2014
;; MSG SIZE  rcvd: 109
```

```
heisenberguser@Heisenberg:/$ dig 192.168.1.139

; <<>> DiG 9.8.1-P1 <<>> 192.168.1.139
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 44627
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;192.168.1.139.                 IN      A

;; AUTHORITY SECTION:
.                       1590    IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2014041301 1800 900 604800 8

;; Query time: 1 msec
;; SERVER: 192.168.1.110#53(192.168.1.110)
;; WHEN: Sun Apr 13 21:13:42 2014
;; MSG SIZE  rcvd: 106
```

DNS SLAVE

The slave DNS comes to power when the Master DNS fails. Any files updated the Master will reflect in the slave DNS.

On the slave server install bind serve
Sudo apt-get install bind9

Once bind has been installed, we have to setup the zones that we will be hosting. In our project we have Heisenberg.com is your domain name, 192.168.1.111 is the IP address of the slave server, and 192.168.1.110 is the IP address of the master server

Add the details of the zone in sudo nano /etc/bind/named.conf.local

```
//include "/etc/bind/zones.rfc1918";
#Forward Lookup Zone
zone "Heisenberg.com"{
        type slave;
        file "/var/lib/bind/Heisenberg.com.db";
        masters { 192.168.1.110; };
        allow-transfer {"none";};
        allow-notify {"none";};
};

#Reverse Lookup Zone
zone "1.168.192.in-addr.arpa" {
        type slave;
        file "/var/lib/bind/db.192";
        masters { 192.168.1.110; };
        allow-transfer {"none";};
        allow-notify {"none";};
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa" {
        type slave;
        file "/var/lib/bind/db.ipv6";
        masters { 192.168.1.110;};
        allow-transfer {"none";};
        allow-notify {"none";};
};
```

The files from the Master DNS automatically gets transferred and updated in the Slave in the following location /var/lib/bind/
The output is as shown below

```
$ORIGIN .
$TTL 604800     ; 1 week
Heisenberg.com          IN SOA  Heisenberg.com. root.Heisenberg.com. (
                                44         ; serial
                                604800     ; refresh (1 week)
                                86400      ; retry (1 day)
                                2419200    ; expire (4 weeks)
                                604800     ; minimum (1 week)
                                )
                        NS      ns.Heisenberg.com.
                        A       192.168.1.139
                        AAAA    fe80::d7
$ORIGIN Heisenberg.com.
ns1.Heisenberg.com      A       192.168.1.111
                        AAAA    fe80::d2
ns                      A       192.168.1.110
                        AAAA    fe80::d1
win7pc1                 A       192.168.1.140
                        AAAA    fe80::d140
win7pc2                 A       192.168.1.145
                        AAAA    fe80::d145
win7pc3                 A       192.168.1.150
                        AAAA    fe80::d150
win7pc4                 A       192.168.1.155
                        AAAA    fe80::d155
```

db.192

```
$ORIGIN .
$TTL 604800     ; 1 week
1.168.192.in-addr.arpa  IN SOA  Heisenberg.com. root.Heisenberg.com. (
                                42         ; serial
                                604800     ; refresh (1 week)
                                86400      ; retry (1 day)
                                2419200    ; expire (4 weeks)
                                604800     ; minimum (1 week)
                                )
                        NS      Heisenberg.com.
$ORIGIN 1.168.192.in-addr.arpa.
110                     PTR     ns.Heisenberg.com.
111                     PTR     ns1.Heisenberg.com.
116                     PTR     gappa.com.
139                     PTR     Heisenberg.com.
140                     PTR     win7pc1.Heisenberg.com.
145                     PTR     win7pc2.Heisenberg.com.
150                     PTR     win7pc3.Heisenberg.com.
155                     PTR     win7pc4.Heisenberg.com.
```

db.ipv6

```
$ORIGIN .
$TTL 604800     ; 1 week
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa IN SOA Heisenberg.com. root.Heisenberg.com. (
                                2          ; serial
                                604800     ; refresh (1 week)
                                86400      ; retry (1 day)
                                2419200    ; expire (4 weeks)
                                604800     ; minimum (1 week)
                                )
                        NS      ns.Heisenberg.com.
$ORIGIN d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa.
1                       PTR     ns.Heisenberg.com.
2                       PTR     ns1.Heisenberg.com.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa.
7                       PTR     Heisenberg.com.
$ORIGIN 4.1.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa.
0                       PTR     win7pc1.Heisenberg.com.
5                       PTR     win7pc2.Heisenberg.com.
$ORIGIN 5.1.d.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa.
0                       PTR     win7pc3.Heisenberg.com.
5                       PTR     win7pc4.Heisenberg.com.
```

Now restart bind to make the changes active:

sudo services bind9 restart

Hierarchy

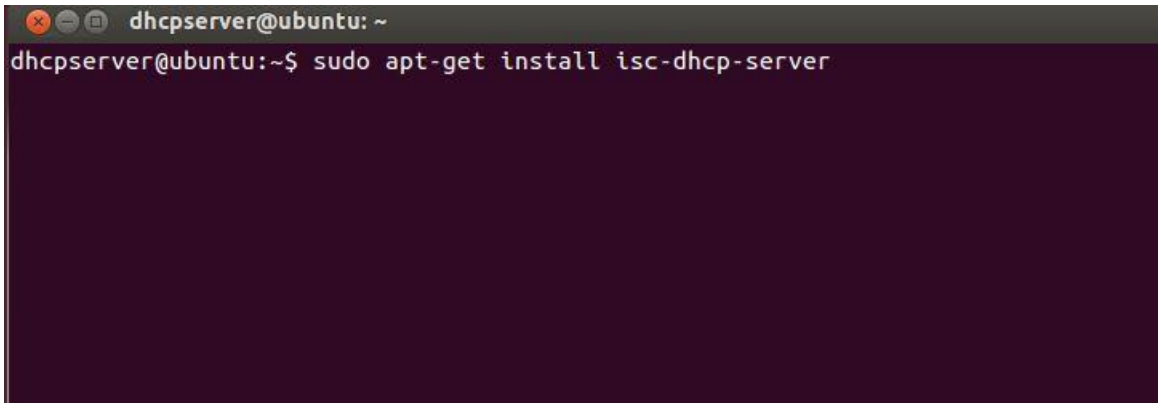
The hierarchy of the DNS is classified as Root Servers, Top Level Domains, and Authoritative Domain Server. The lookup of the IP address follows the hierarchy of Root, TLD and Authoritative DNS servers.

- When the browser enters Heisenberg.com, hostname is extracted from URL and then passed to the client DNS.
- The client DNS sends the DNS query to the local DNS server for the hostname.
- When the local DNS is not able to resolve the hostname IP address, it forwards the query to the Root DNS server.
- The root dns server differentiates the .com suffix. It responds with the set of IP addresses of the Top Level DNS server which are responsible for .com
- Next the local DNS server sends the query to Top level DNS servers for Heisenberg.com and the TLD responds with the IP addresses of authoritative DNS servers Heisenberg.com.
- Finally the local DNS server queries to the authoritative DNS for the IP address of Heisenberg.com.
- The authoritative DNS responds with the actual IP address of Heisenberg.com and then the browser forwards this request to the webserver.

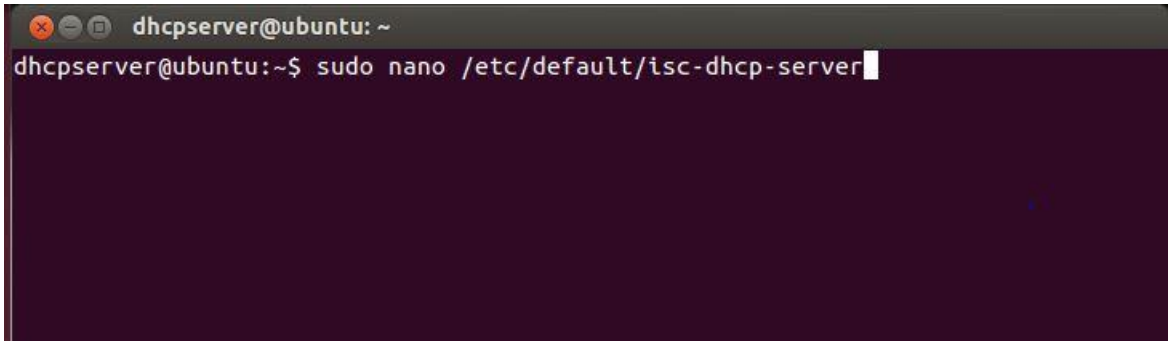
DHCP

DHCP

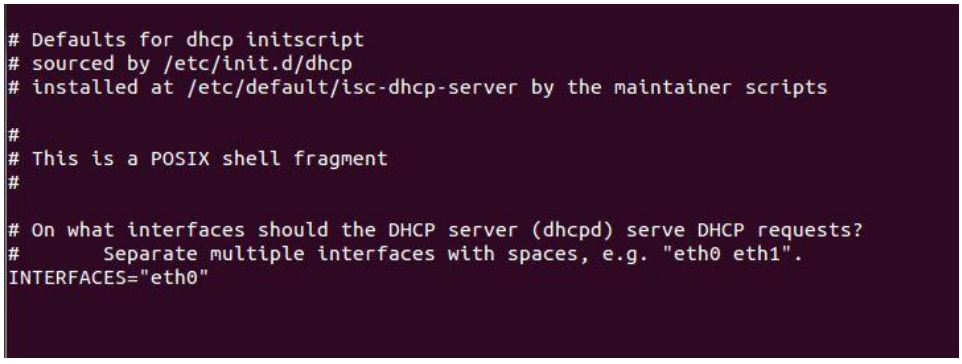
Step 1: Install the DHCP server with the command **sudo apt-get install isc-dhcp-server**



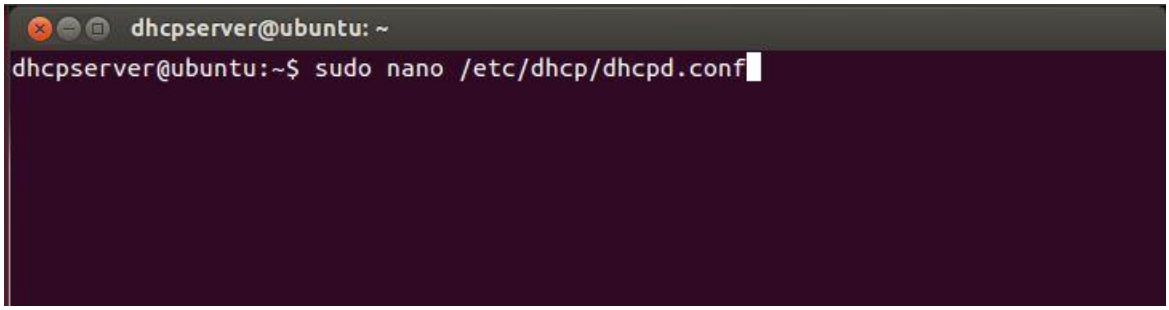
Step 2: After the DHCP is installed, the configuration file has to be edited to change the interface where dhcp is listening to



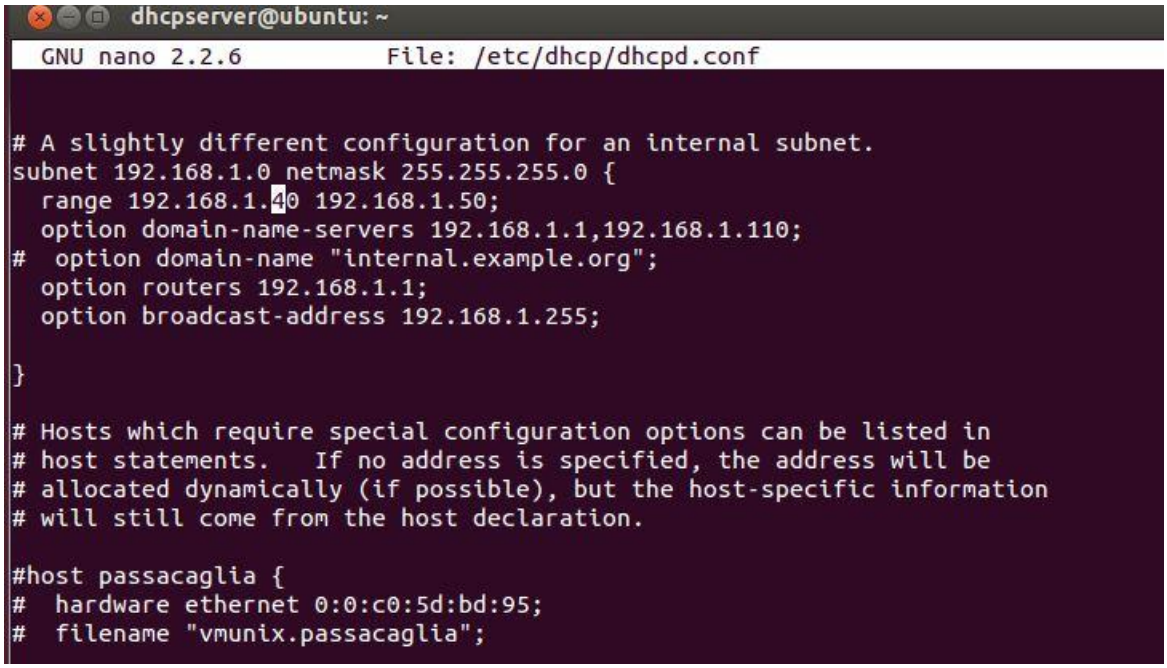
Step 3 : Now after opening the file, edit the interface where the dhcp is listening.
In this case the dhcp is listening to the interface eth0. So enter the name of the interface.



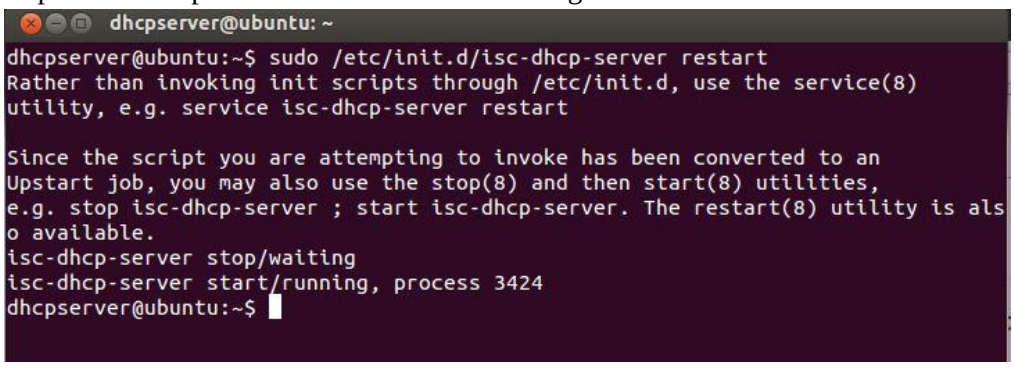
Step 4: Now the DHCP has to be configured in accordance with the network. In order to configure the dhcp, edit the file **/etc/dhcp/dhcpd.conf**



Step 5: Configure the dhcp as shown below. Edit the file in accordance with the network .Enter the range of IP address the DHCP should generate.



Step 6: The Dhcp server is restarted for the changes to take effect.



DHCP v6:

Step 1: To enable IPv6 the following command should be queried.

sudo nano /etc/sysctl.conf

uncomment the command **net.ipv6.conf.default.forwarding=1**

Step 2: Install radvd using the command

sudo apt-get install radvd

Step 3: Edit the file sudo nano /etc/radvd.conf and insert the following

```
interface eth0 {
AdvSendAdvert on;
        #AdvManagedFlag on
MinRtrAdvInterval 3;
MaxRtrAdvInterval 10;
prefix 2001:0db8:0100:f101::/64 {
AdvOnLink on;
AdvAutonomous on;
AdvRouterAddr on;
        };
};
```

Step 4 : Edit the file sudo nano /etc/dhcp/dhcpd.conf

```
default-lease-time 600;
max-lease-time 7200;
log-facility local7;
subnet6 2001:db8:0:1:: /64 {
# Range for clients
range6 2001:db8:0:1::129 2001:db8:0:1::254;
# Additional options
#option dhcp6.name-servers fec0:0:0:1::1;
#option dhcp6.domain-search "domain.example";
# Prefix range for delegation to sub-routers
prefix6 2001:db8:0:100:: 2001:db8:0:f00:: /56;
# Example for a fixed host address
host specialclient {
host-identifier option dhcp6.client-id 00:01:00:01:4a:1f:ba:e3:60:b9:1f:01:23:45;
fixed-address6 2001:db8:0:1::127;
        }
 }
```

Step 5: Restart radvd using the command

sudo /etc/init.d/radvd restart

DH CLIENT

TESTING

The Client is connected to the DHCP through an Ethernet cable.

Step 1: Configure the network interfaces of the client by editing the file /etc/network/interfaces

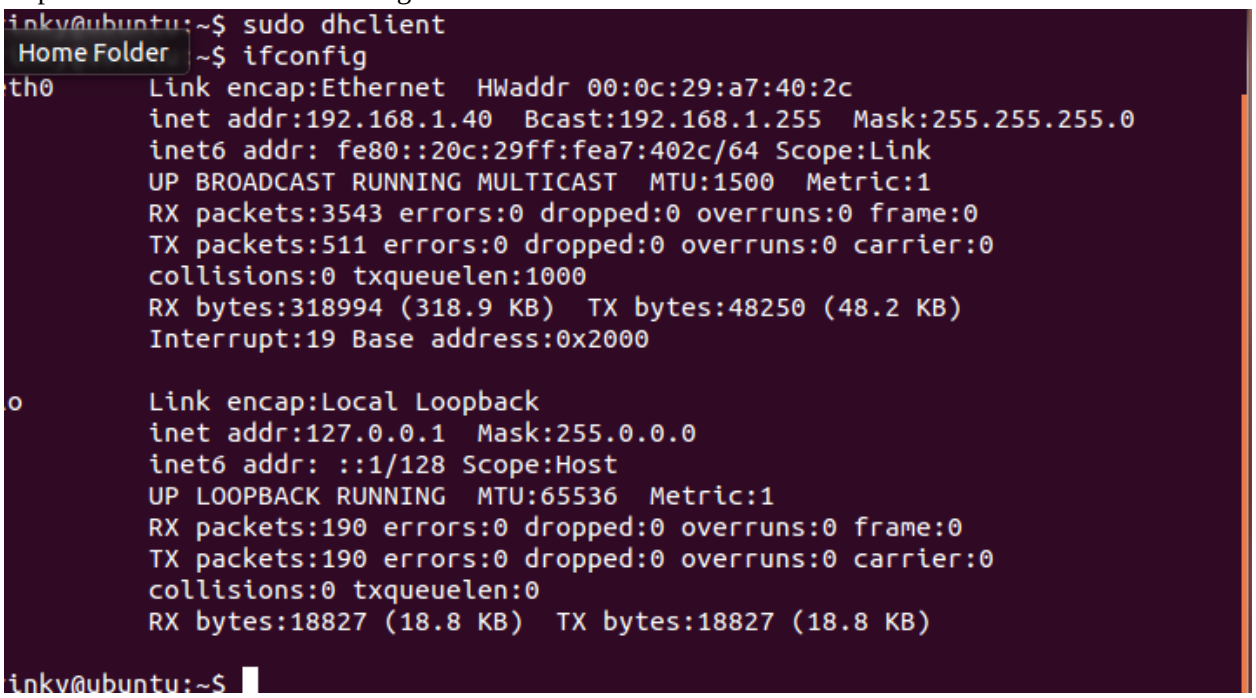
```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Step 2: After configuring the interfaces, restart the networking in the machine.

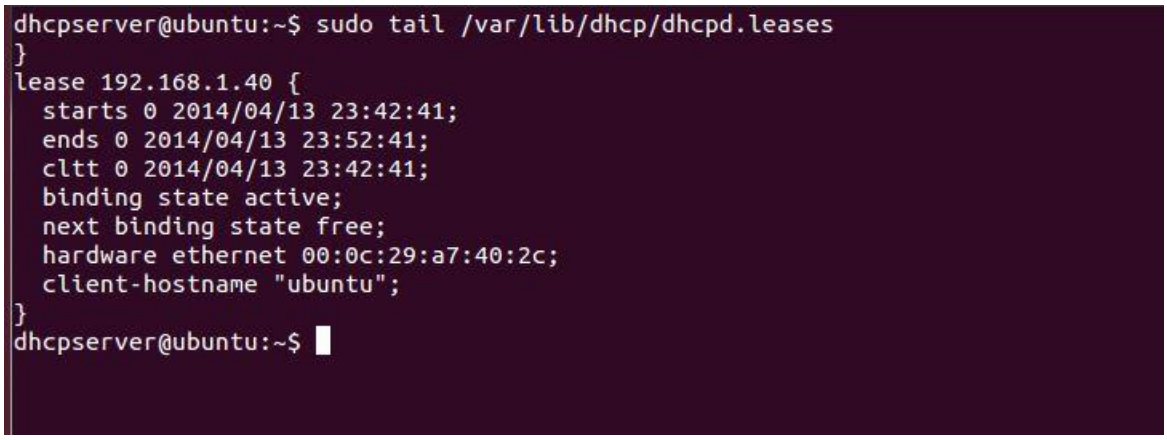
```
rinky@ubuntu:~$ sudo /etc/init.d/networking restart
[sudo] password for rinky:
 * Running /etc/init.d/networking restart is deprecated because it may not enabl
e again some interfaces
 * Reconfiguring network interfaces...
ssh stop/waiting
ssh start/running, process 3571
                                                                        [ OK ]
rinky@ubuntu:~$
```

Step 3: Now the client must have got an IP address from the DHCP.

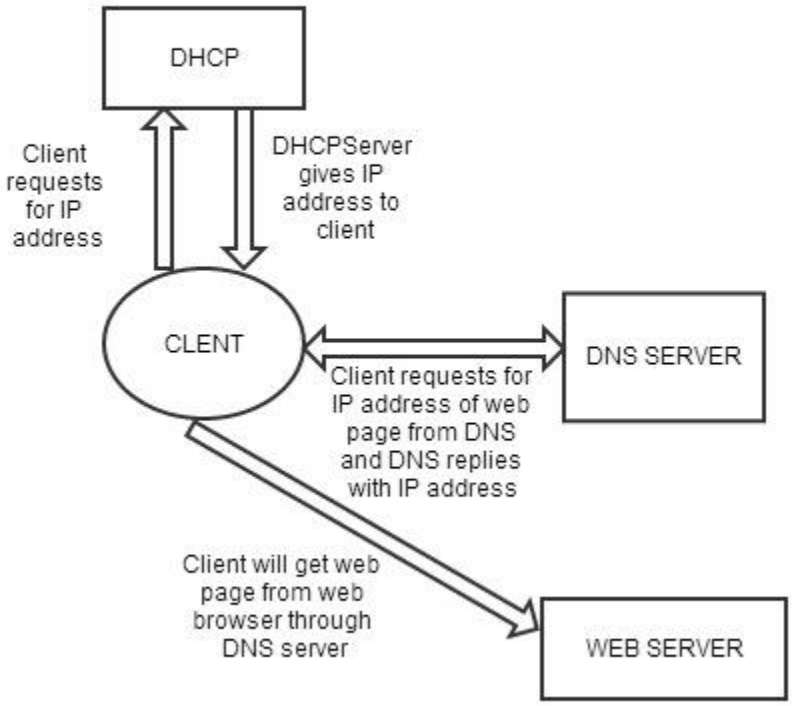


It is seen that the IP address for the client is generated by the DHCP.
Thus the DHCP server is properly configured.

The IP address lease can be checked in the DHCP as below.



FLOWCHART



ADD ONs:

NTP: NTP is a TCP/IP protocol for synchronizing time over a network. Basically a client requests the current time from a server, and uses it to set its own clock.

Commands used:

- To set up ntpd:
  sudo apt-get install ntpd

- For ntpd edit /etc/ntp.conf to include additional server lines:
  server Heisenberg.com

- Restart ntpd
  sudo /etc/init.d/ntp restart

- Check it using
  sudo ntpq –p

VPN: A VPN is a private network that uses a public network (usually the Internet) to connect remote sites or users together.

- Install pptp server using apt-get
  sudo apt-get install pptpd

- Configure the pptpd
  sudo nano /etc/pptpd.conf

- Add server IP and client IP at the end of the file

- Configure DNS servers to use when clients connect to this PPTP server
  Sudo nano /etc/ppp/pptpd.options

- Uncomment the ms-dns and add DNS IP

- Now add a VPN user and password in /etc/ppp/chap-secrets file.

- Finally restart the VPN server

NFS: NFS allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems almost as if they were local files.

NIS: NIS is a client–server directory service protocol for distributing system configuration data such as user and host names between computers on a computer network.

FUTURE IMPROVEMENTS:

1. More addons can be implemented like VLAN, a end system acting as a router to represent a real time network.

2. Security can be tightened by implementing alerts to the network admin, whenever a user attempts to perform an action as the root user.

3. Implementation of LDAP.

Citations

www.help.ubuntu.com

www.ubuntuforums.org

www.askubuntu.com

www.ubuntugeek.com