Data Networking

LINUX PROJECT

Wikiversity Link :

https://en.wikiversity.org/wiki/Data_Networking/(TELE_5330)_Linux_Project_Fall_2016:_Sanket_Jabade

Project Objective:

To create a robust, secure, dynamic and intelligent network consisting of a DNS server, DHCP server, Web server with firewall and a client for a start-up company based in Boston.

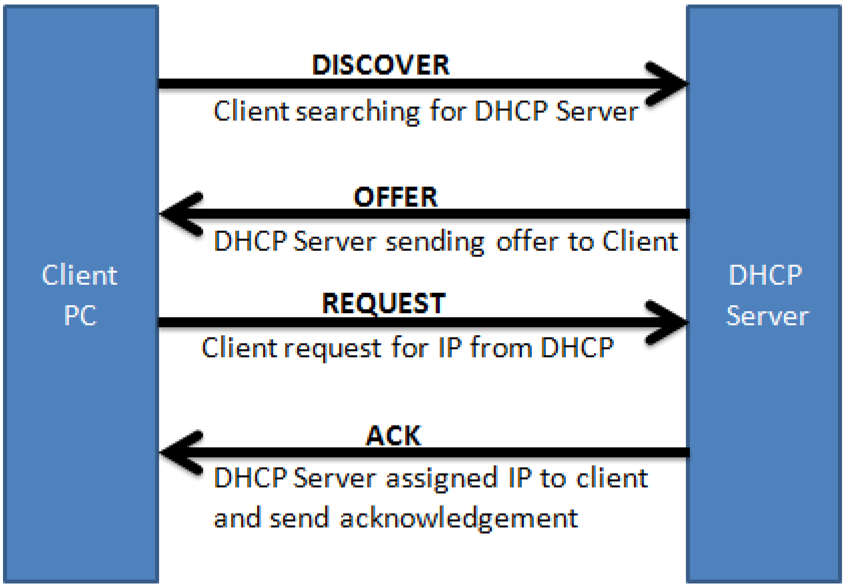
Team Members:

Maheshwar Gurav:	001226694
Riteshkumar Gupta:	001280361
Sanket B Jabade:	001234914
Prajwal Patil:	001280390

Network components:

- DHCP
- DNS
- Web Server
- Client
- Firewall
- Backup

DHCP: Dynamic Host Configuration Protocol is used on IP networks to assign IPs dynamically to computers and other end users connected to the network. It reduces the need to manually assign IP addresses by a network administrator; a centralized DHCP server simply does this task.


Client
PC
DISCOVER
Client searching for DHCP Server
OFFER
DHCP Server sending offer to Client
REQUEST
Client request for IP from DHCP
ACK
DHCP Server assigned IP to client and send acknowledgement
DHCP
Server

Behaviour of the protocol:

DHCP is a client-server protocol wherein a server shares its resources and client establishes a connection with the server to use the resources.
1. DHCP server behaviour: When a DHCP client connects to the server by sending a DHCP message, the server processes it based on current binding state of the client. The various types of DHCP message can be following: DHCPDISCOVER DHCPREQUEST DHCPRELEASE DHCPINFORM
a. DHCPDISCOVER: To every DHCPDISCOVER message from client, the server chooses an IP address from its DHCP pool. If no IP address is available, it reports the system administrator otherwise it simply assigns the IP to the client based on the binding state with the client.
b. DHCP OFFER: When the server offers an IP to the client in reply to DHCPDISCOVER, the clients sends a DHCPREQUEST message to the server, by checking whether the IP is previously allocated or whether the lease has ended, etc. Generally in response to DHCPOFFER given by server, the client sends a DHCP request containing 'Server Identifier' option.
c. DHCP REQUEST: If the network address is not allocated, the client sends DHCP REQUEST message to the server and the server thus makes an entry of client's initialization parameters.
d. DHCP ACK: To every DHCP REQUEST message from the client, the server sends a DHCP ACK message directly to the address specified at 'ciaddr' field of DHCPINFORM message. 2. DHCP client behaviour: A client can receive following messages from the server. DHCPOFFER DHCPACK DHCPNACK
The client sends DHCPINFORM message to the server and waits for DHCPACK. The client completed the configuration process by setting its own parameters.

Benefits of DHCP:

- **Reliable IP address configuration**. DHCP minimizes configuration errors caused by manual IP address configuration, such as typographical errors, or address conflicts caused by the assignment of an IP address to more than one computer at the same time.

- **Reduced network administration**. DHCP includes the following features to reduce network administration:
    - Centralized and automated TCP/IP configuration.
    - The ability to define TCP/IP configurations from a central location.
    - The ability to assign a full range of additional TCP/IP configuration values by means of DHCP options.
    - The efficient handling of IP address changes for clients that must be updated frequently, such as those for portable computers that move to different locations on a wireless network.
    - The forwarding of initial DHCP messages by using a DHCP relay agent, which eliminates the need for a DHCP server on every subnet.

Steps to configure DHCP server in Ubuntu 16.04:

1. Update the apt-get. APT stands for Advanced Packaging Tool which is a free user interface used to install and remove packages in Linux machines.

Command: **sudo apt-get update**

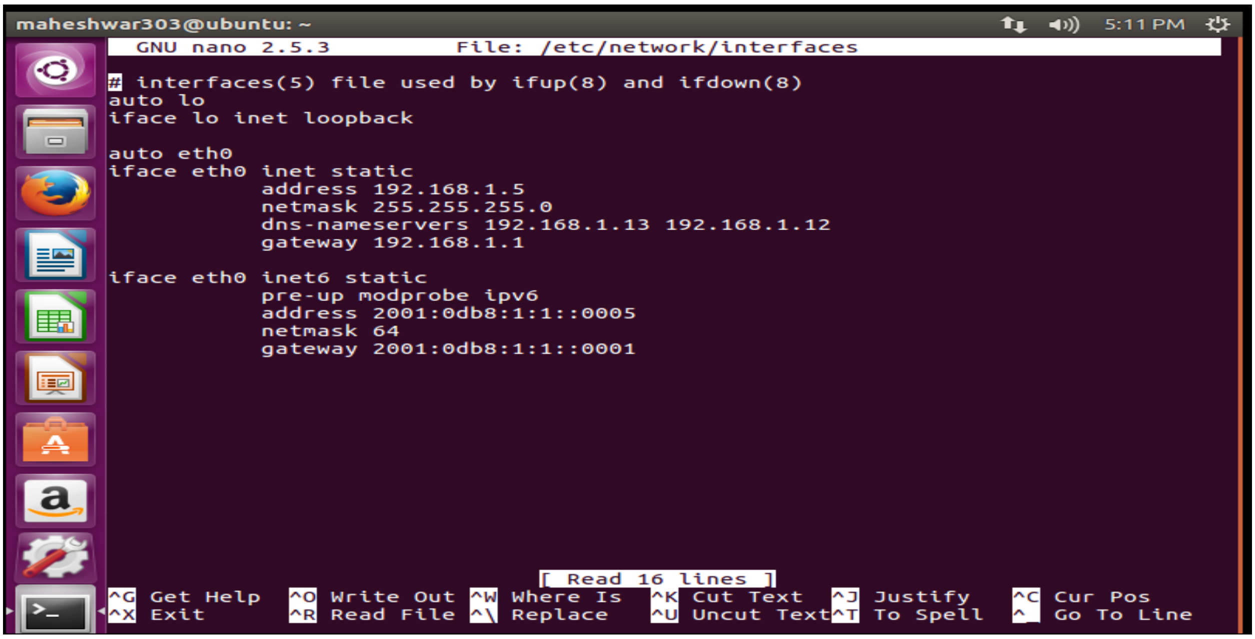
2. Install isc-dhcp server package in order to configure a DHCP server in Ubuntu machine.

Command: **sudo apt-get install isc-dhcp-server**

The package consists of several file out of which we need to edit few important files to configure the server.

3. To give static IPv4 and IPv6 address to interface serving DHCP request:

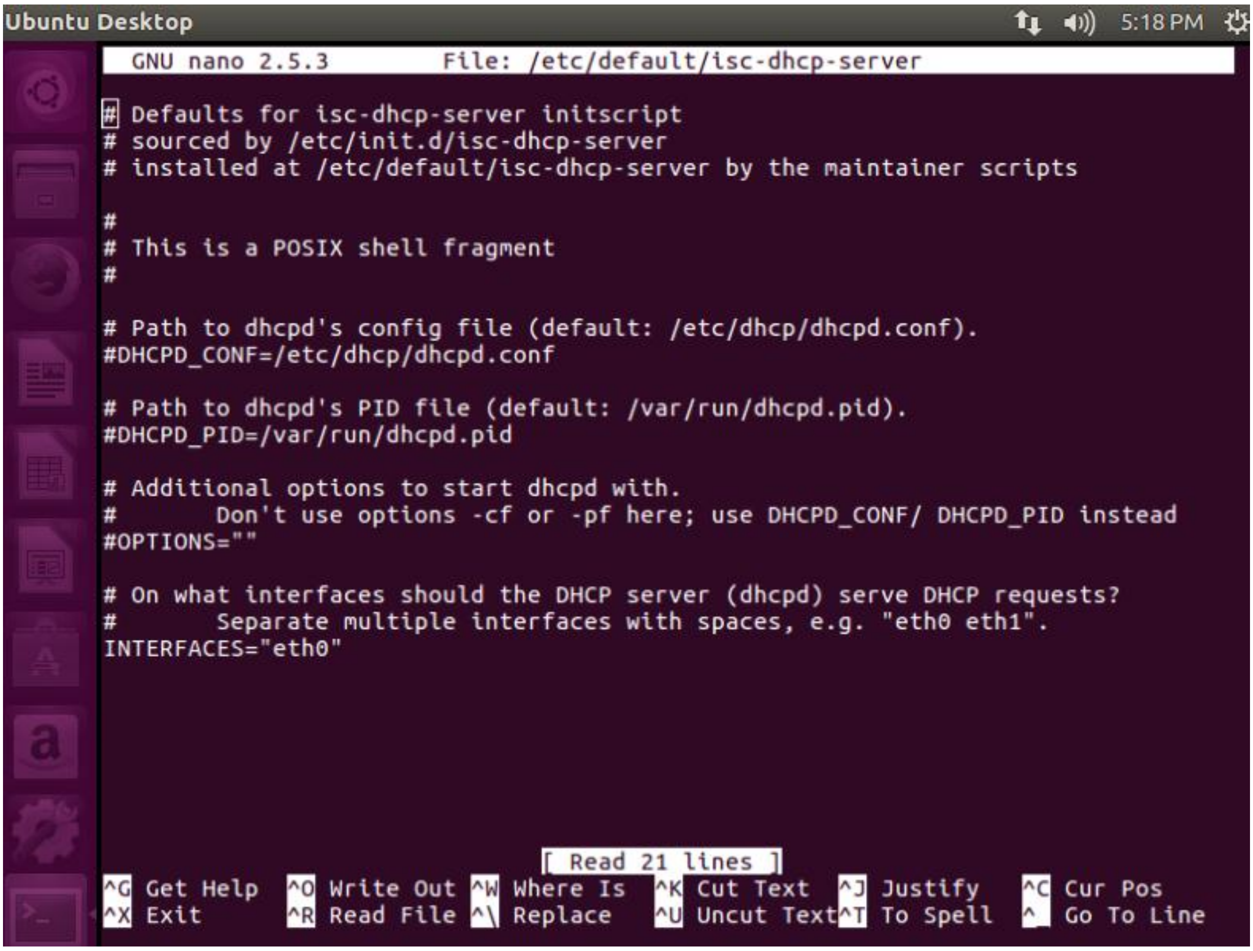
Command: **sudo nano /etc/network/interfaces**



```
  GNU nano 2.5.3            File: /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
           address 192.168.1.5
           netmask 255.255.255.0
           dns-nameservers 192.168.1.13 192.168.1.12
           gateway 192.168.1.1

iface eth0 inet6 static
           pre-up modprobe ipv6
           address 2001:0db8:1:1::0005
           netmask 64
           gateway 2001:0db8:1:1::0001

                                  [ Read 16 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

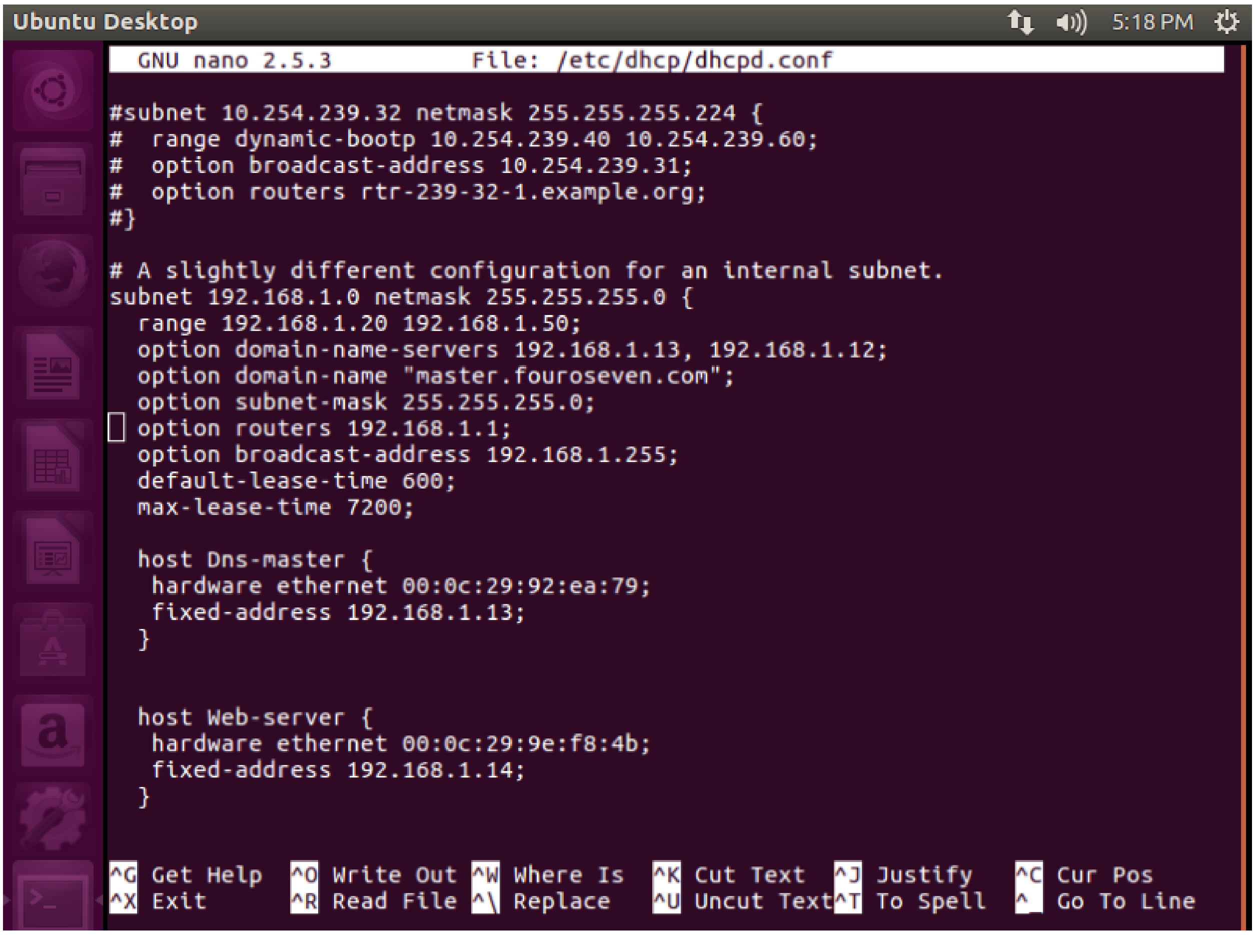
4. Setup interface serving DHCP request

Command: **sudo nano /etc/default/isc-dhcp-server**



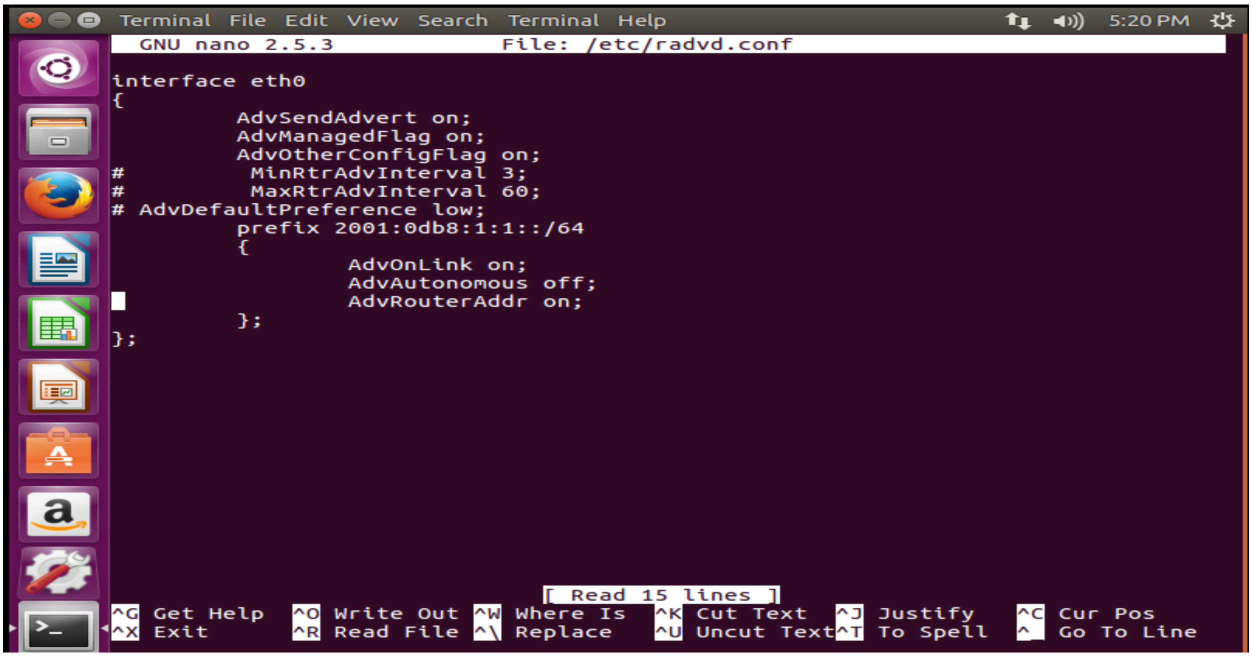
```
  GNU nano 2.5.3          File: /etc/default/isc-dhcp-server

# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth0"

                                  [ Read 21 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

5. dhcpd.conf file configurations to define IPv4 address range for client and fixed address to a particular client and server



```
  GNU nano 2.5.3            File: /etc/dhcp/dhcpd.conf

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.20 192.168.1.50;
  option domain-name-servers 192.168.1.13, 192.168.1.12;
  option domain-name "master.fouroseven.com";
  option subnet-mask 255.255.255.0;
  option routers 192.168.1.1;
  option broadcast-address 192.168.1.255;
  default-lease-time 600;
  max-lease-time 7200;

  host Dns-master {
   hardware ethernet 00:0c:29:92:ea:79;
   fixed-address 192.168.1.13;
  }


  host Web-server {
   hardware ethernet 00:0c:29:9e:f8:4b;
   fixed-address 192.168.1.14;
  }

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

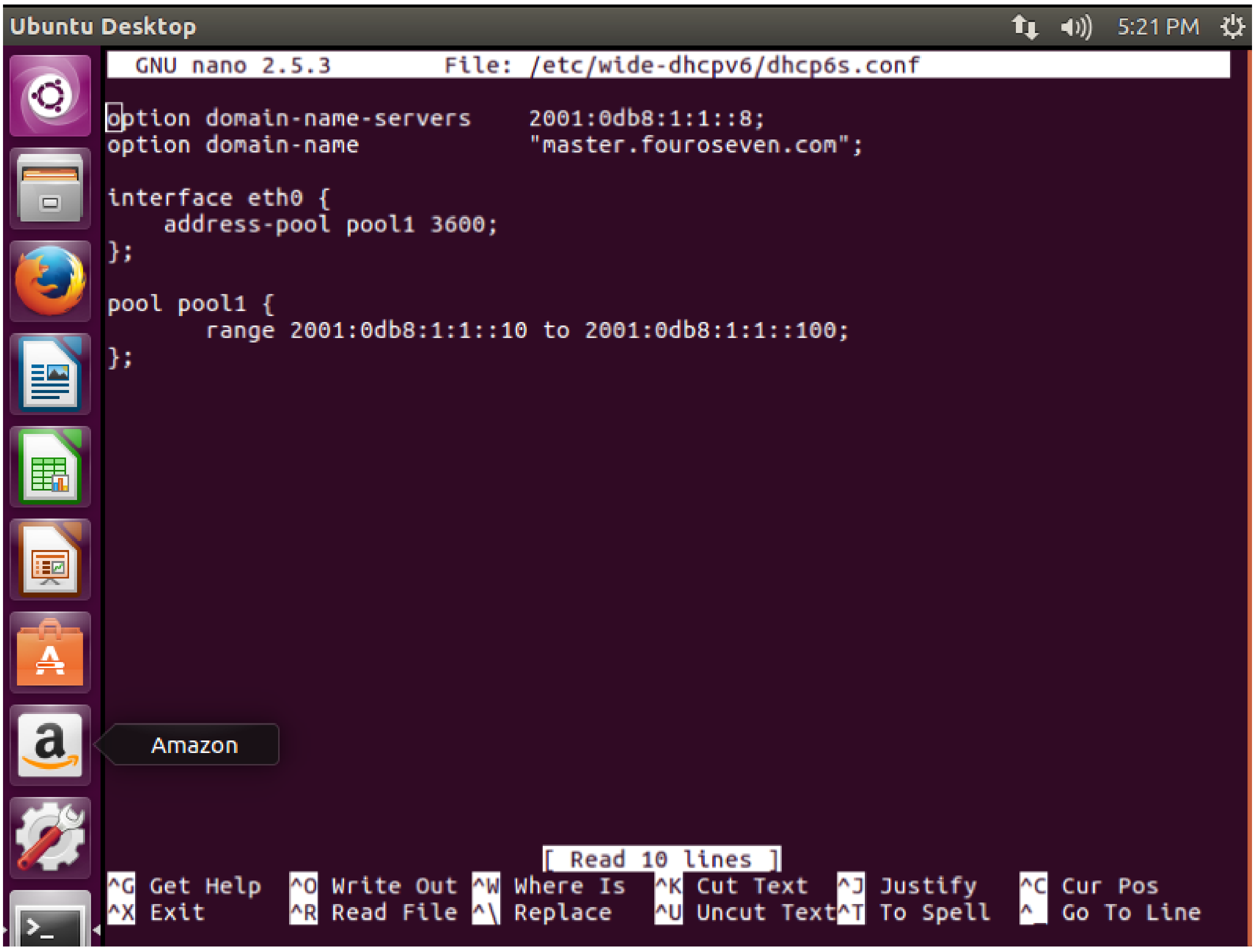
6. To advertise IPv6 address, we install router advertisement demon
Command: **sudo apr-get install radvd**
We need to configure radvd.conf as below



```
  GNU nano 2.5.3              File: /etc/radvd.conf

interface eth0
{
        AdvSendAdvert on;
        AdvManagedFlag on;
        AdvOtherConfigFlag on;
#        MinRtrAdvInterval 3;
#        MaxRtrAdvInterval 60;
# AdvDefaultPreference low;
        prefix 2001:0db8:1:1::/64
        {
                AdvOnLink on;
                AdvAutonomous off;
                AdvRouterAddr on;
        };
};

                                [ Read 15 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Command: **sudo apt-get install wide-dhcpv6-server**

To define IPv6 address range for clients, we need to install wide-dhcpv6-server package



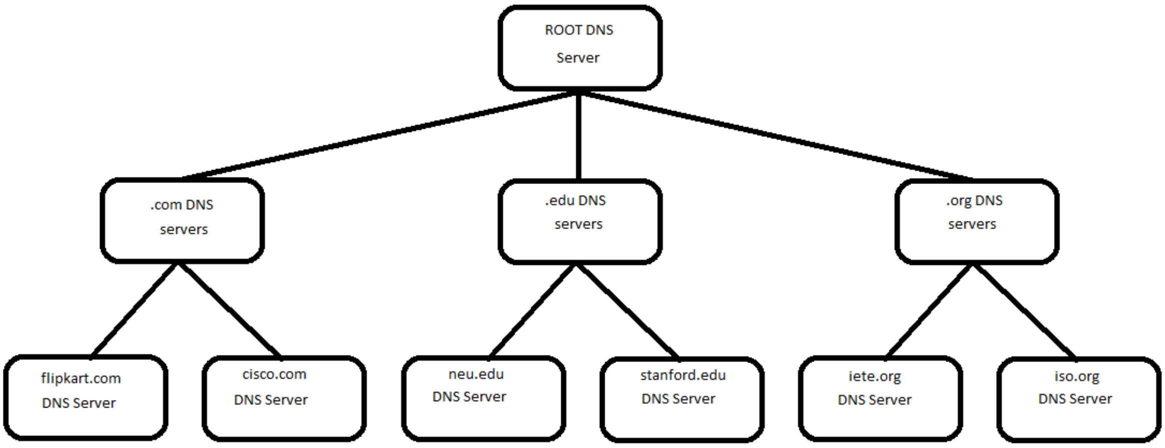
```
  GNU nano 2.5.3         File: /etc/wide-dhcpv6/dhcp6s.conf

option domain-name-servers    2001:0db8:1:1::8;
option domain-name            "master.fouroseven.com";

interface eth0 {
    address-pool pool1 3600;
};

pool pool1 {
       range 2001:0db8:1:1::10 to 2001:0db8:1:1::100;
};

                               [ Read 10 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

DNS:

- Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses.
- This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.
- Information from all the domain name servers across the Internet are gathered together and housed at the Central Registry. Host companies and Internet Service Providers interact with the Central Registry on a regular schedule to get updated DNS information.
- When you type in a web address, e.g., www.google.com, your Internet Service Provider views the DNS associated with the domain name, translates it into a machine friendly IP address (for example 8.8.8.8 is

the IP for google.com) and directs your Internet connection to the correct website.

- After you register a new domain name or when you update the DNS servers on your domain name, it usually takes about 12-36 hours for the domain name servers world-wide to be updated and able to access the information. This 36-hour period is referred to as propagation.



The hierarchical server structure in the above diagram has three levels.
They are:
1) Root DNS servers – These are present at the top level of hierarchy
2) Top level domain (TLD) DNS servers – These are present at the next level to root servers in hierarchy
3) Authoritative or Local DNS servers – These are present at the next level to TLD servers in hierarchy

Query Types
There are two types of queries in DNS hierarchy. They are:
1) Recursive queries
2) Iterative queries

1) Recursive queries:
Whenever a host looks for resolution local Name server associated with that host recursively requests another level of DNS servers for resolution and fetches the name resolution for the requested host. Recursive queries are allowed if the hosts are in the same domain.
2) Iterative queries:
Iterative queries are handled by root DNS servers, TLD DNS servers and authoritative name servers.

Resource Records: In DNS hierarchy, DNS server replies to a query with one or more records.
Format of BIND9 server resource records

Name TTL Class Type Pref Value
TTL is Time To Live
Pref is the preference value (Used in the case of multiple mail servers)
Class is used to identify protocol family
Type=A
Name is the hostname
Value is the IP address
This type is simply a hostname-to-IP address mapping
Type = NS
Name is the domain, e.g., auburn.edu
Value is the hostname of the authoritative name server for this domain
This type is used as a routing function for queries
Type = CNAME
Name is the alias name, e.g., www.ibm.com
Value is the canonical name, e.g., servereast.backup2.ibm.com
This type simply provides the canonical name when requested
Type = MX
Name is domain name
Value is the name of the mail server associated with this domain
Type = PTR

The master server contains original sets of records, whereas slave contains copy of records.

Steps to configure BIND9 Master DNS server and Slave DNS server

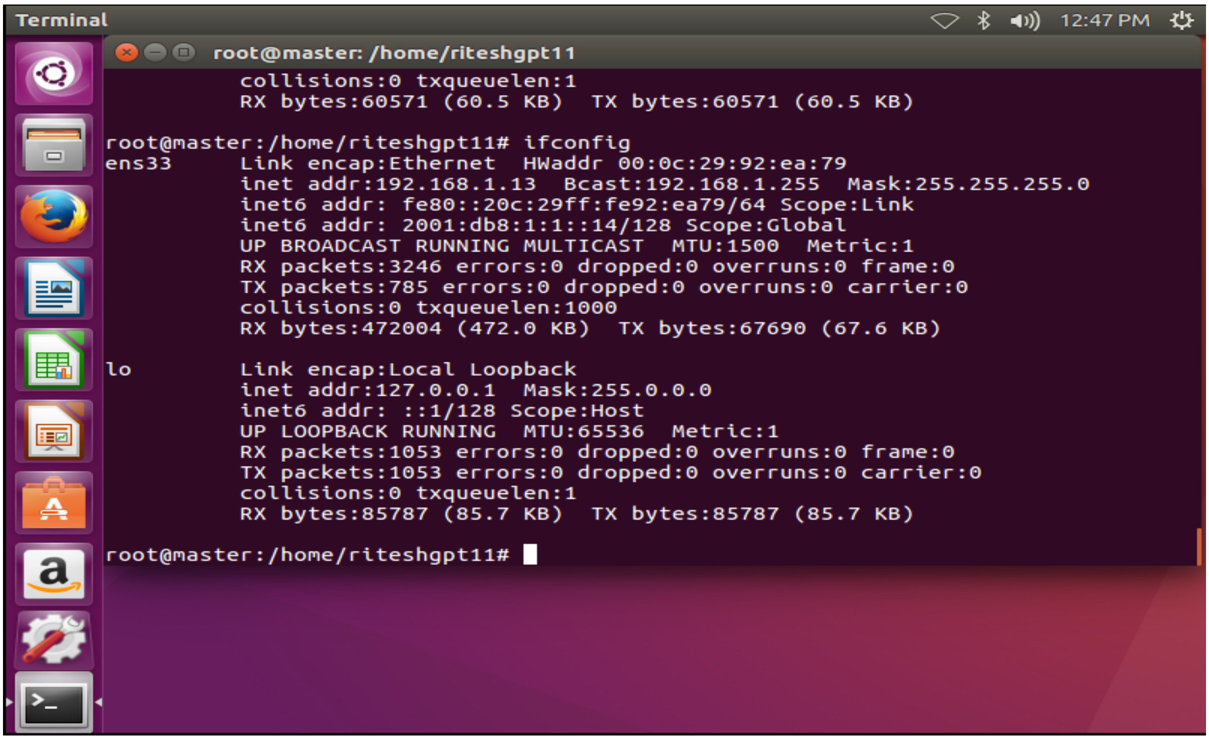
Master DNS configuration:

1) Install all the required updates and packages related to BIND9 using following commands.

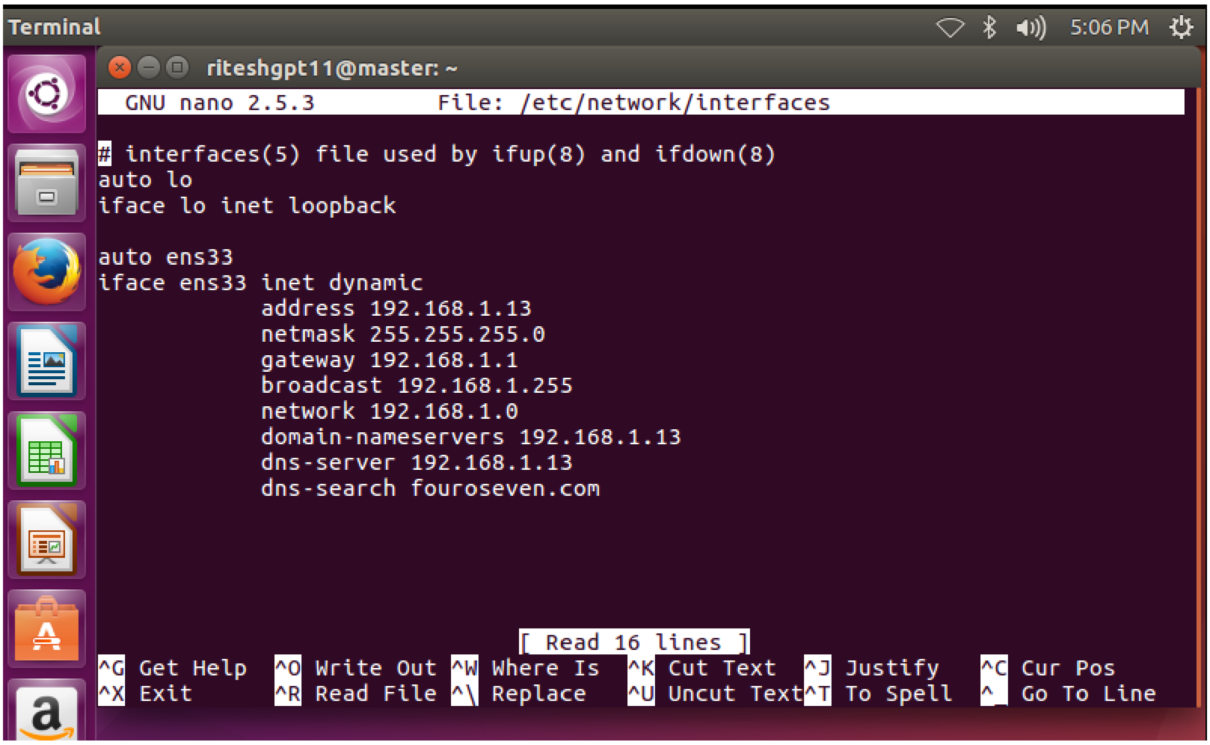
```
sudo apt-get update
sudo apt-get install bind9
```

2) Open network interfaces file to make changes as in the below screenshot using the following command.

```
sudo nano /etc/network/interfaces
```

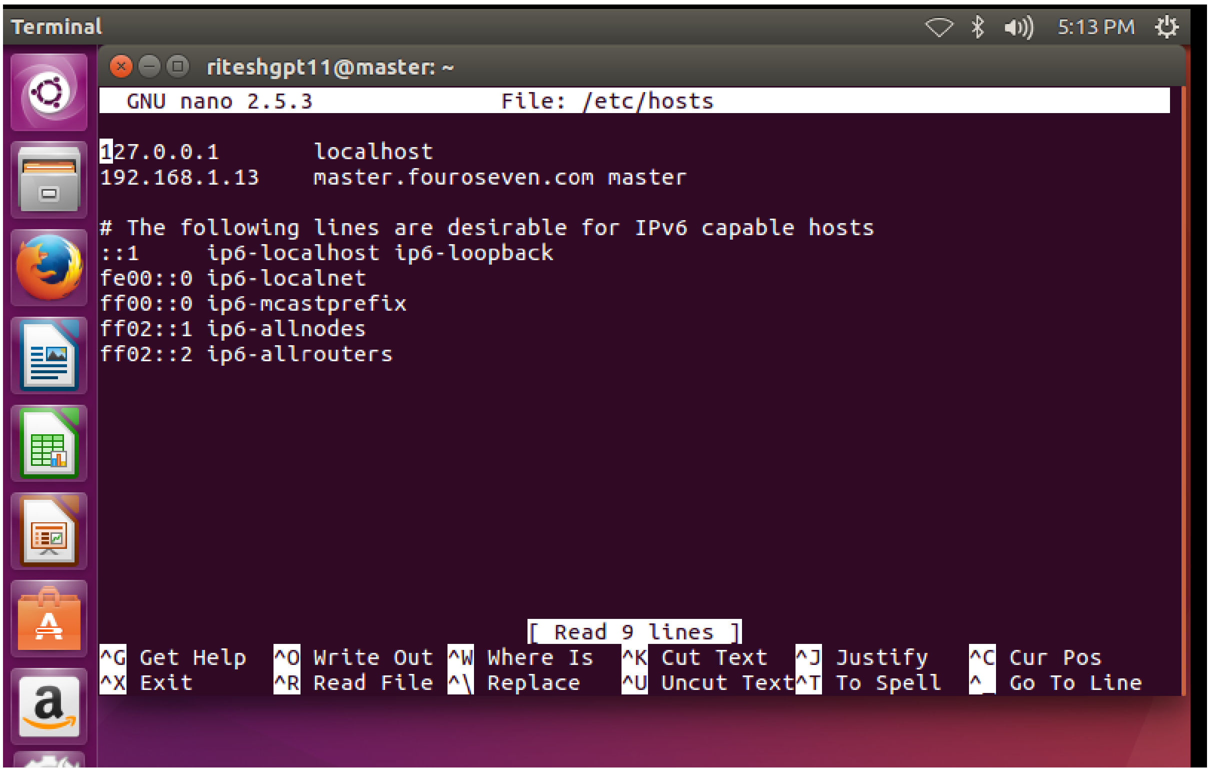


```
          collisions:0 txqueuelen:1
          RX bytes:60571 (60.5 KB)  TX bytes:60571 (60.5 KB)

root@master:/home/riteshgpt11# ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:92:ea:79
          inet addr:192.168.1.13  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe92:ea79/64 Scope:Link
          inet6 addr: 2001:db8:1:1::14/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3246 errors:0 dropped:0 overruns:0 frame:0
          TX packets:785 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:472004 (472.0 KB)  TX bytes:67690 (67.6 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:1053 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1053 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:85787 (85.7 KB)  TX bytes:85787 (85.7 KB)

root@master:/home/riteshgpt11# 
```



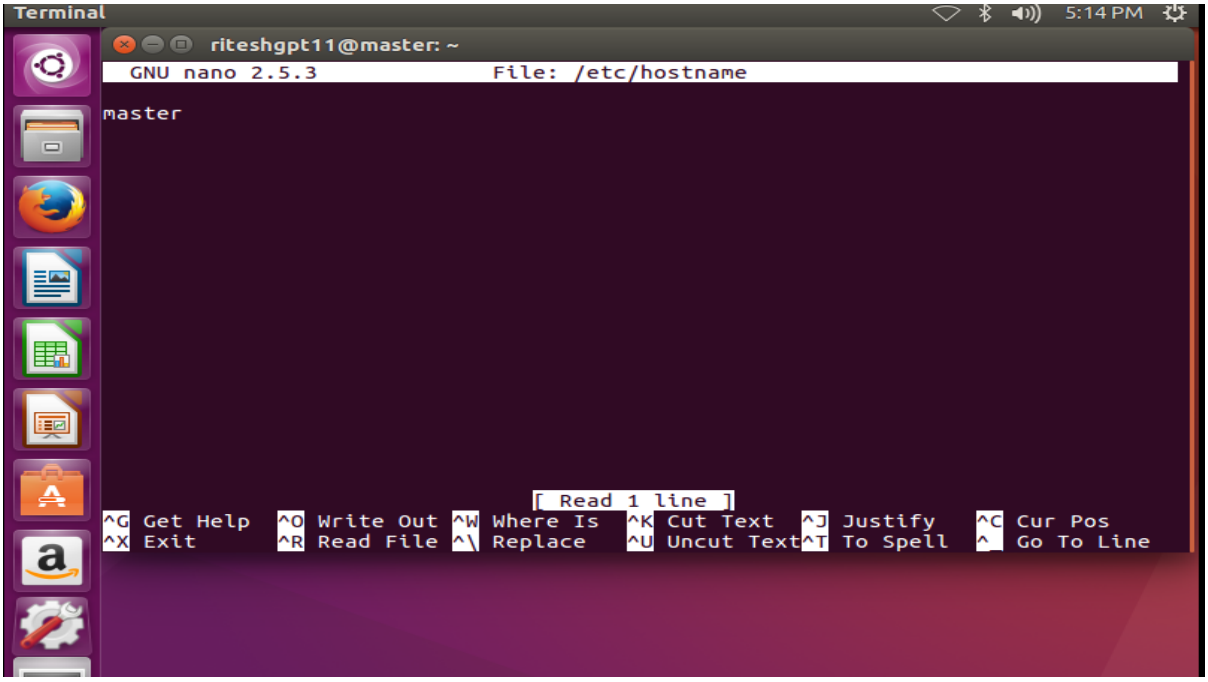
```
  GNU nano 2.5.3           File: /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet dynamic
            address 192.168.1.13
            netmask 255.255.255.0
            gateway 192.168.1.1
            broadcast 192.168.1.255
            network 192.168.1.0
            domain-nameservers 192.168.1.13
            dns-server 192.168.1.13
            dns-search fouroseven.com




                                [ Read 16 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

3) Open hosts file by using the following command and make the changes same as in the below screenshot.
sudo nano /etc/hosts



```
  GNU nano 2.5.3              File: /etc/hosts

127.0.0.1       localhost
192.168.1.13    master.fouroseven.com master

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

                               [ Read 9 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

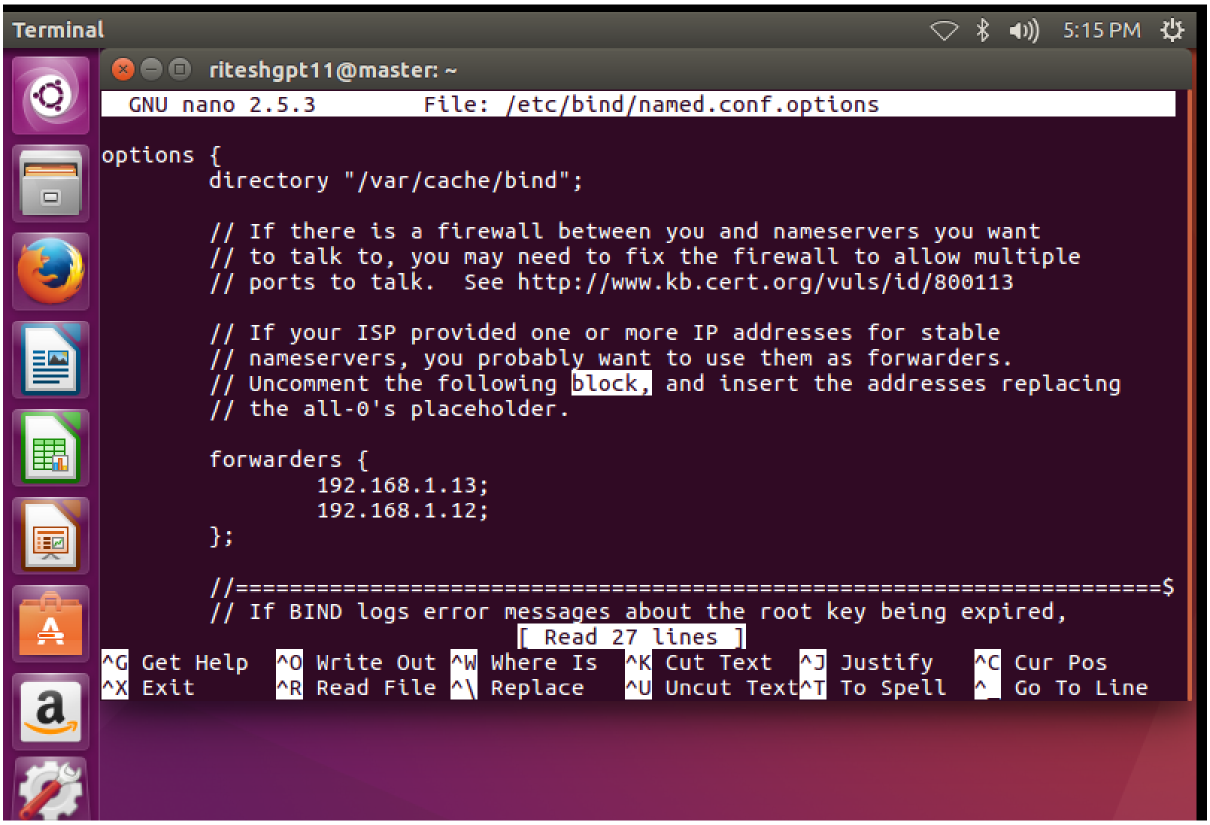
4) Open hostname file using the following command and make below changes.
sudo nano /etc/hostname



```
  GNU nano 2.5.3              File: /etc/hostname

master

                               [ Read 1 line ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

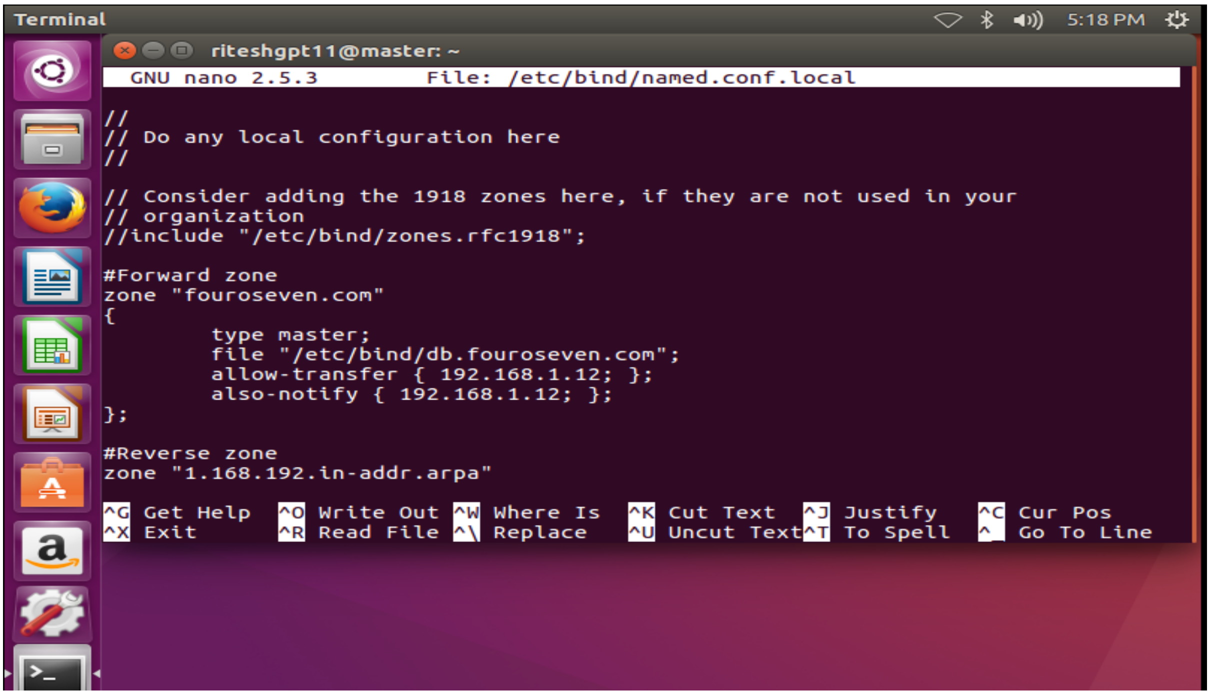
5) Reboot the virtual machine by using the following command to apply changes.
sudo init 6
6) Open the named.conf.options file by using the following command and make the changes as below.

Sudo nano /etc/bind/named.conf.options

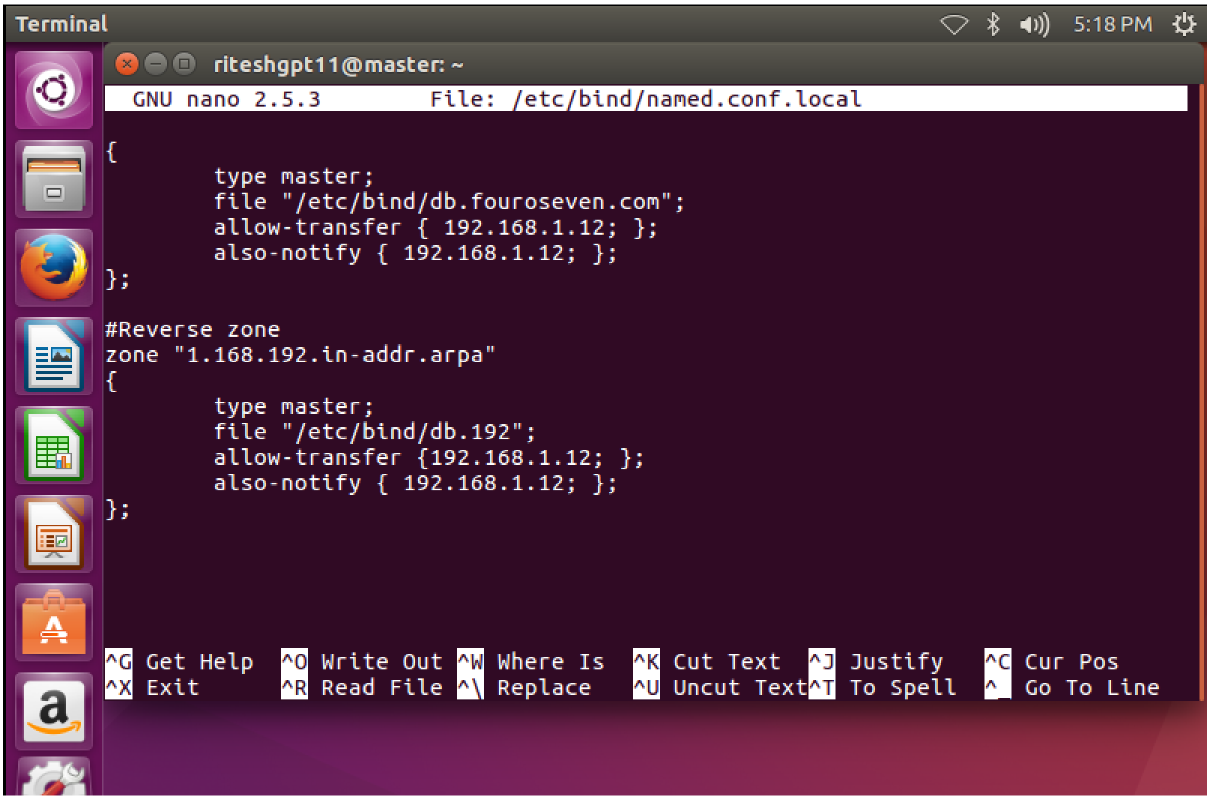


```
  GNU nano 2.5.3          File: /etc/bind/named.conf.options

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 replacing
        // the all-0's placeholder.

        forwarders {
                192.168.1.13;
                192.168.1.12;
        };

        //=====================================================================$
        // If BIND logs error messages about the root key being expired,
                                [ Read 27 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```

7) Open the named.conf.local file and modify as per below
sudo nano /etc/bind/named.conf.local



```
  GNU nano 2.5.3          File: /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

#Forward zone
zone "fouroseven.com"
{
        type master;
        file "/etc/bind/db.fouroseven.com";
        allow-transfer { 192.168.1.12; };
        also-notify { 192.168.1.12; };
};

#Reverse zone
zone "1.168.192.in-addr.arpa"
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```



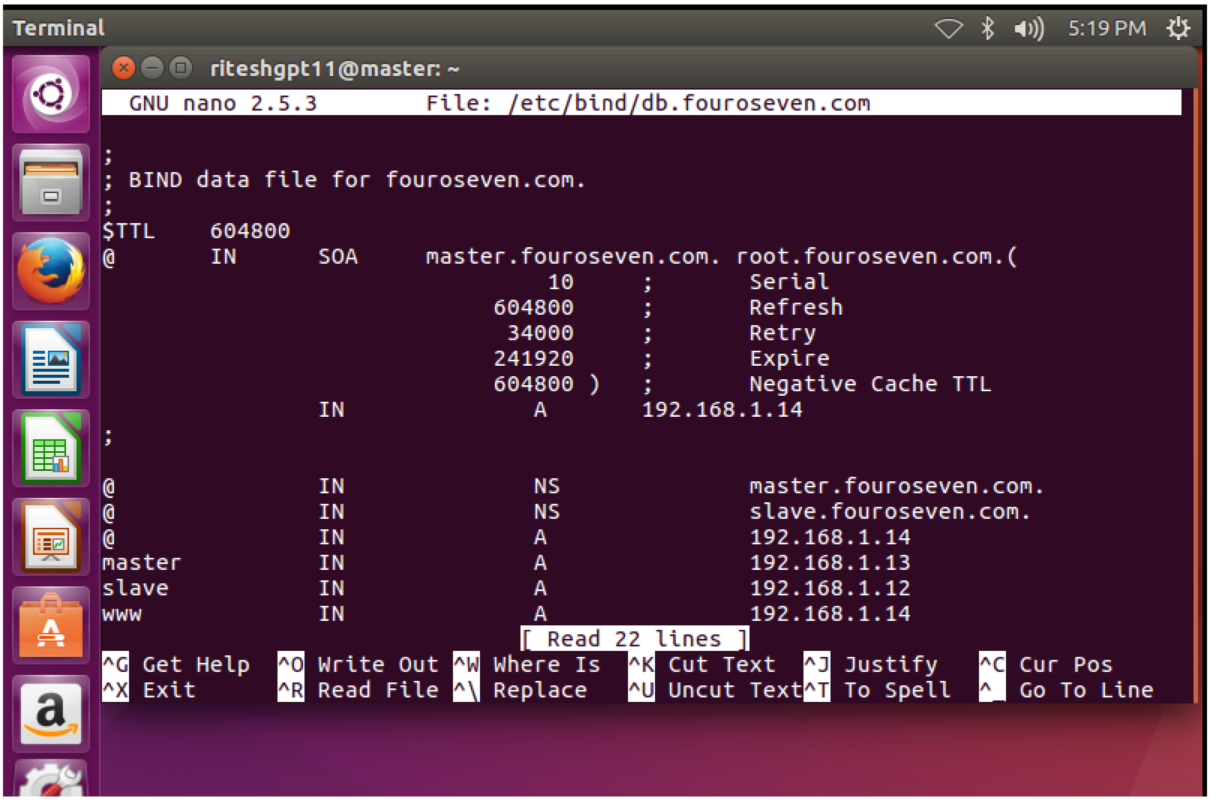
Terminal
5:18 PM
riteshgpt11@master: ~

```
  GNU nano 2.5.3          File: /etc/bind/named.conf.local

{
        type master;
        file "/etc/bind/db.fouroseven.com";
        allow-transfer { 192.168.1.12; };
        also-notify { 192.168.1.12; };
};

#Reverse zone
zone "1.168.192.in-addr.arpa"
{
        type master;
        file "/etc/bind/db.192";
        allow-transfer {192.168.1.12; };
        also-notify { 192.168.1.12; };
};
```

^G Get Help   ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos
^X Exit   ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell   ^_ Go To Line

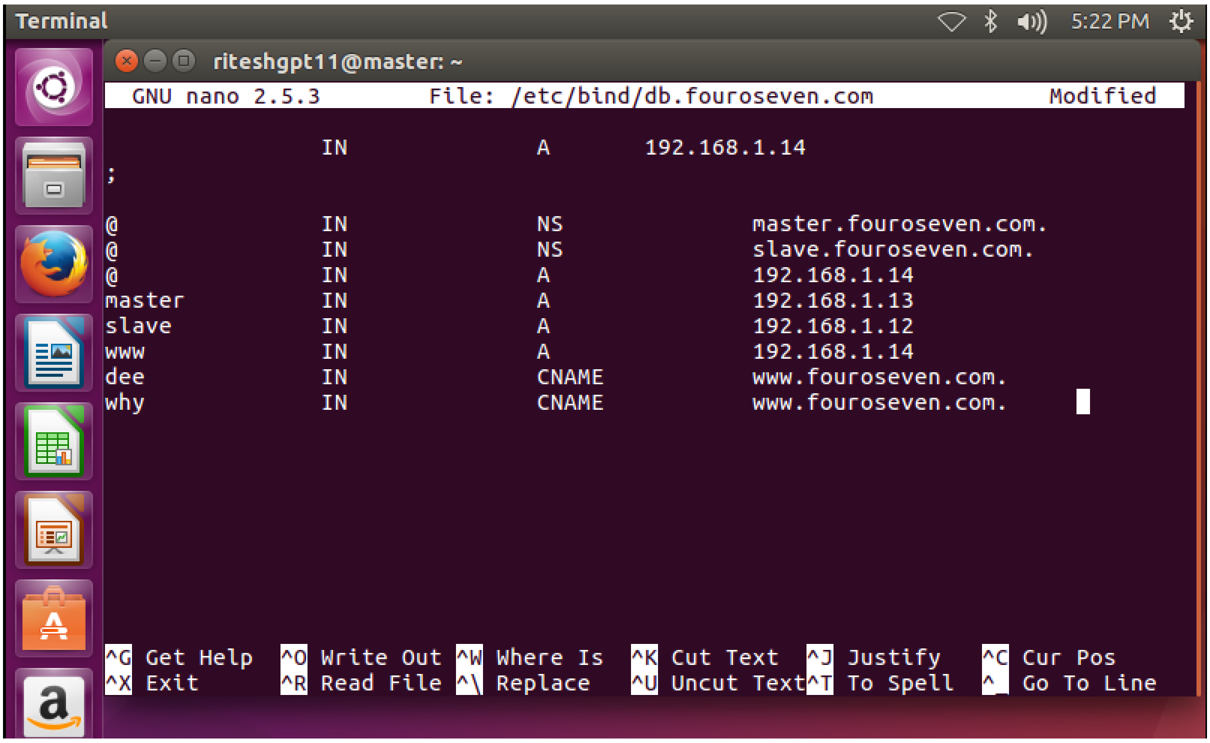
8) Create and open the db.adeptmaster.com forward zone file by using the following command and make the changes as below.

Sudo nano /etc/bind/db.fouroseven.com



```
  GNU nano 2.5.3          File: /etc/bind/db.fouroseven.com

;
; BIND data file for fouroseven.com.
;
$TTL    604800
@       IN      SOA     master.fouroseven.com. root.fouroseven.com.(
                              10      ;       Serial
                          604800      ;       Refresh
                           34000      ;       Retry
                          241920      ;       Expire
                          604800 )    ;       Negative Cache TTL
                IN              A       192.168.1.14
;

@               IN              NS              master.fouroseven.com.
@               IN              NS              slave.fouroseven.com.
@               IN              A               192.168.1.14
master          IN              A               192.168.1.13
slave           IN              A               192.168.1.12
www             IN              A               192.168.1.14
                               [ Read 22 lines ]
```

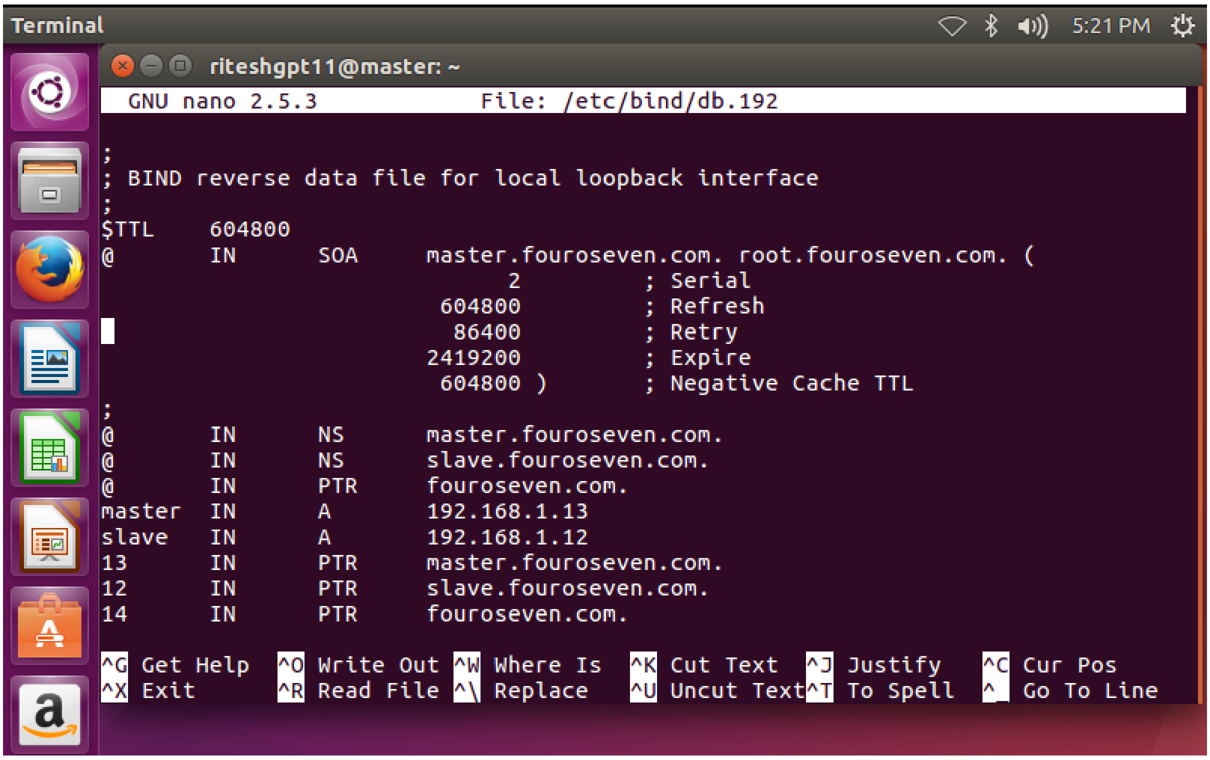
^G Get Help   ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos
^X Exit   ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell   ^_ Go To Line



```
  GNU nano 2.5.3          File: /etc/bind/db.fouroseven.com              Modified

                IN              A       192.168.1.14
;
@               IN              NS              master.fouroseven.com.
@               IN              NS              slave.fouroseven.com.
@               IN              A               192.168.1.14
master          IN              A               192.168.1.13
slave           IN              A               192.168.1.12
www             IN              A               192.168.1.14
dee             IN              CNAME           www.fouroseven.com.
why             IN              CNAME           www.fouroseven.com.

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

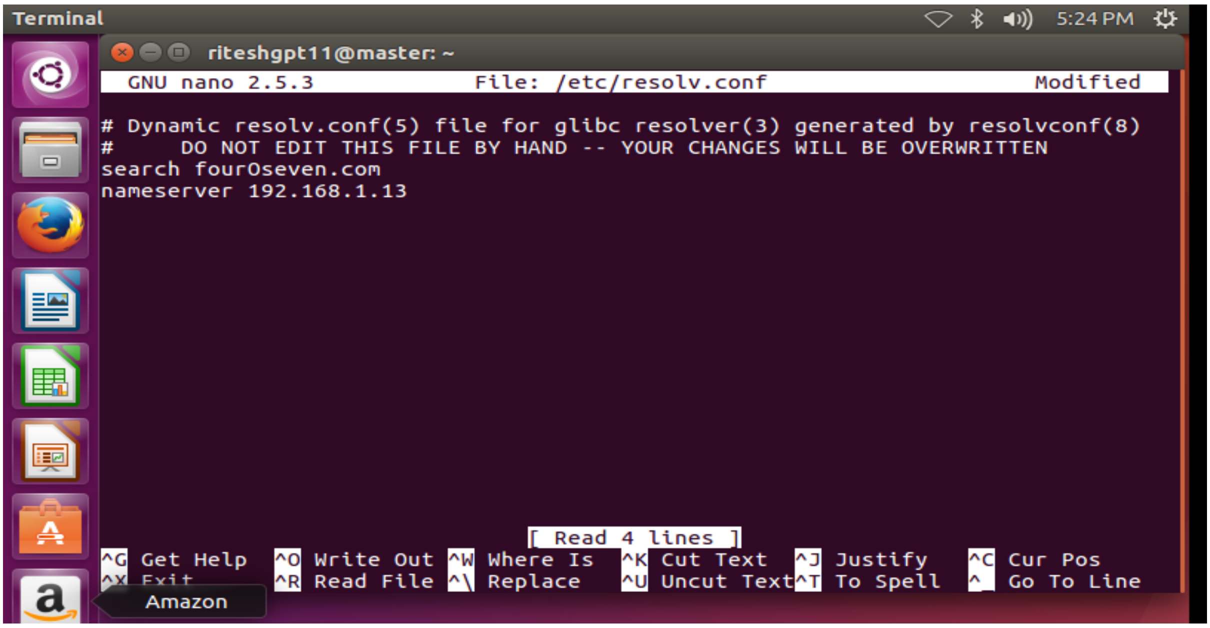
9) Create and open the db.adeptmaster.com reverse zone file by using the following command and make the changes as below.

Sudo nano /etc/bind/db.192



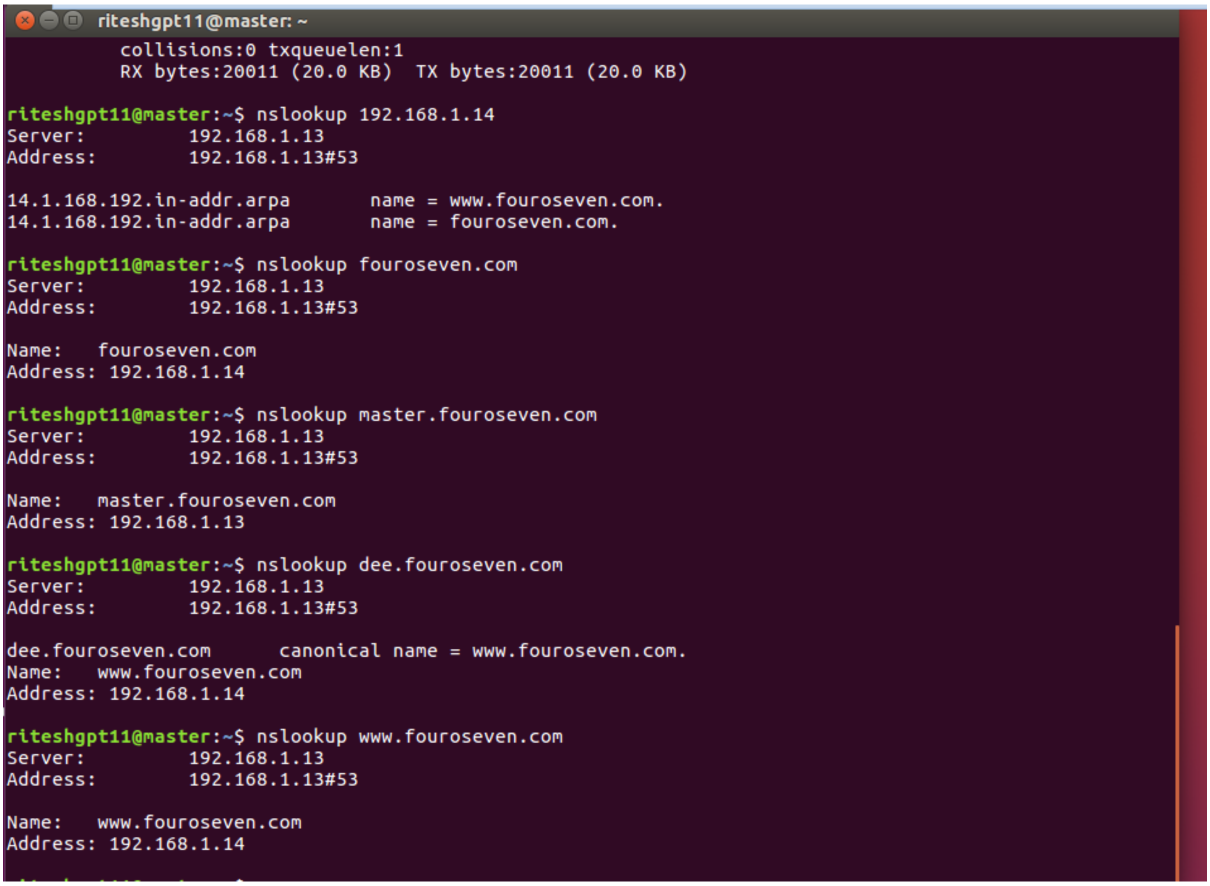
```
  GNU nano 2.5.3              File: /etc/bind/db.192

;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     master.fouroseven.com. root.fouroseven.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      master.fouroseven.com.
@       IN      NS      slave.fouroseven.com.
@       IN      PTR     fouroseven.com.
master  IN      A       192.168.1.13
slave   IN      A       192.168.1.12
13      IN      PTR     master.fouroseven.com.
12      IN      PTR     slave.fouroseven.com.
14      IN      PTR     fouroseven.com.

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

10) Open the head file by using the following command and make the changes as below.
Sudo nano /etc/resolv.conf/

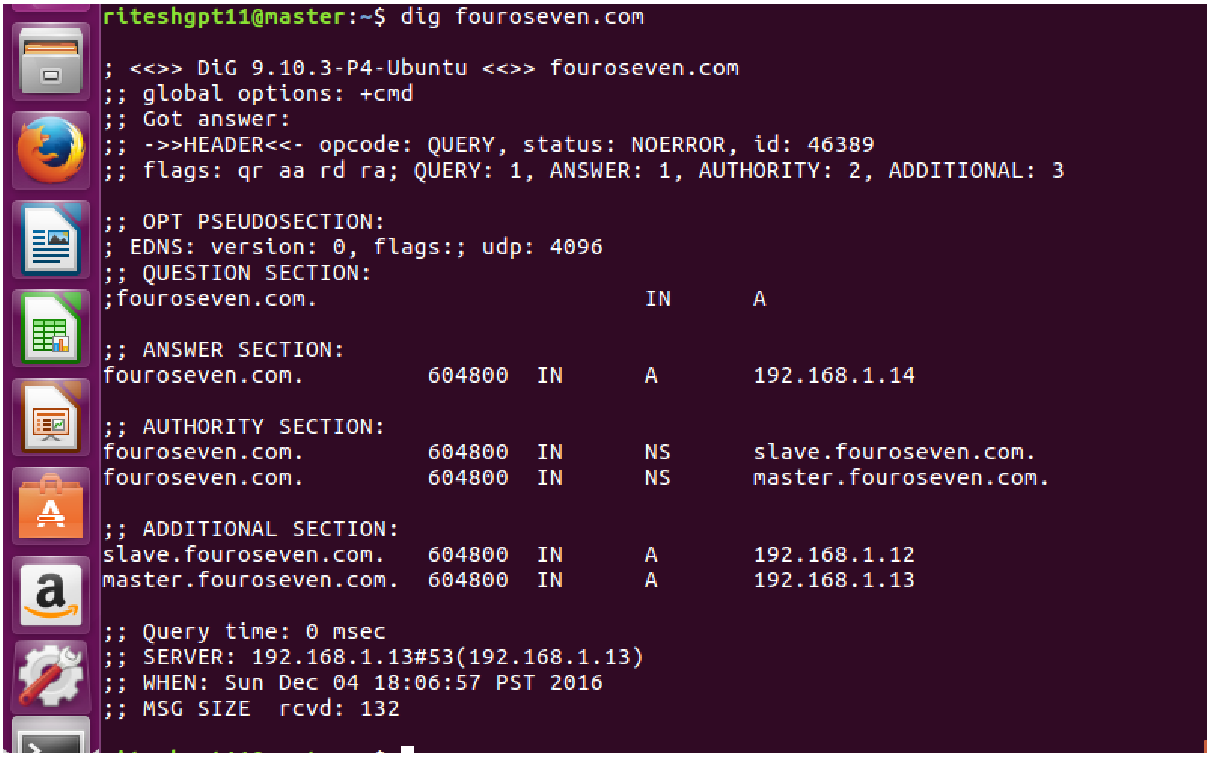


```
  GNU nano 2.5.3              File: /etc/resolv.conf                    Modified

# 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
search fourOseven.com
nameserver 192.168.1.13
```

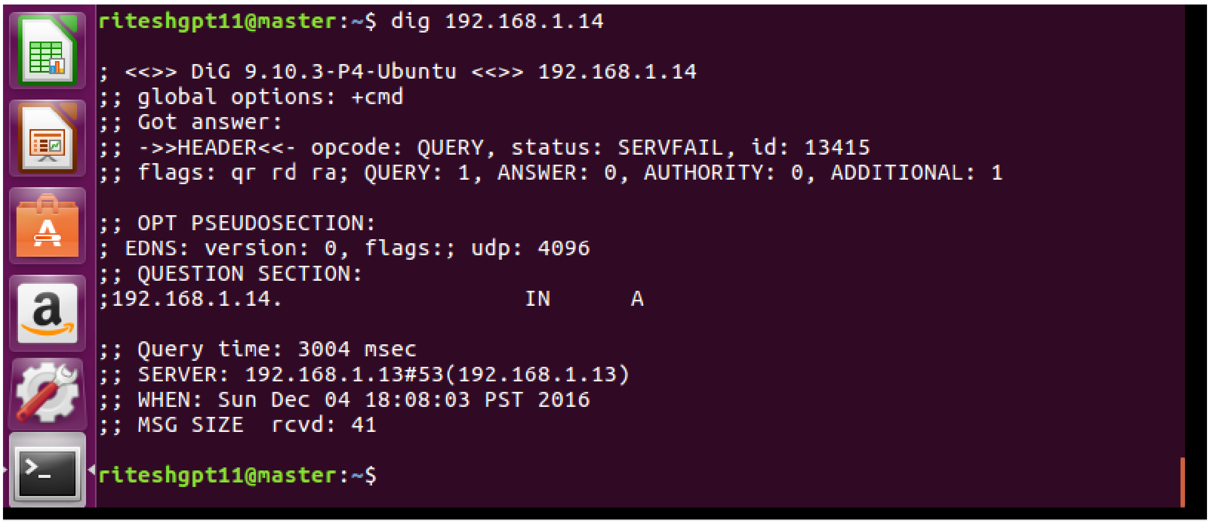
11) Restart BIND9 server using the below command.
Sudo service bind9 restart
Test results for records updated in master:



```
          collisions:0 txqueuelen:1
          RX bytes:20011 (20.0 KB)  TX bytes:20011 (20.0 KB)

riteshgpt11@master:~$ nslookup 192.168.1.14
Server:         192.168.1.13
Address:        192.168.1.13#53

14.1.168.192.in-addr.arpa       name = www.fouroseven.com.
14.1.168.192.in-addr.arpa       name = fouroseven.com.

riteshgpt11@master:~$ nslookup fouroseven.com
Server:         192.168.1.13
Address:        192.168.1.13#53

Name:   fouroseven.com
Address: 192.168.1.14

riteshgpt11@master:~$ nslookup master.fouroseven.com
Server:         192.168.1.13
Address:        192.168.1.13#53

Name:   master.fouroseven.com
Address: 192.168.1.13

riteshgpt11@master:~$ nslookup dee.fouroseven.com
Server:         192.168.1.13
Address:        192.168.1.13#53

dee.fouroseven.com      canonical name = www.fouroseven.com.
Name:   www.fouroseven.com
Address: 192.168.1.14

riteshgpt11@master:~$ nslookup www.fouroseven.com
Server:         192.168.1.13
Address:        192.168.1.13#53

Name:   www.fouroseven.com
Address: 192.168.1.14
```



```
riteshgpt11@master:~$ dig fouroseven.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> fouroseven.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46389
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;fouroseven.com.                        IN      A

;; ANSWER SECTION:
fouroseven.com.         604800  IN      A       192.168.1.14

;; AUTHORITY SECTION:
fouroseven.com.         604800  IN      NS      slave.fouroseven.com.
fouroseven.com.         604800  IN      NS      master.fouroseven.com.

;; ADDITIONAL SECTION:
slave.fouroseven.com.   604800  IN      A       192.168.1.12
master.fouroseven.com.  604800  IN      A       192.168.1.13

;; Query time: 0 msec
;; SERVER: 192.168.1.13#53(192.168.1.13)
;; WHEN: Sun Dec 04 18:06:57 PST 2016
;; MSG SIZE  rcvd: 132
```



```
riteshgpt11@master:~$ dig 192.168.1.14

; <<>> DiG 9.10.3-P4-Ubuntu <<>> 192.168.1.14
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 13415
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;192.168.1.14.                  IN      A

;; Query time: 3004 msec
;; SERVER: 192.168.1.13#53(192.168.1.13)
;; WHEN: Sun Dec 04 18:08:03 PST 2016
;; MSG SIZE  rcvd: 41

riteshgpt11@master:~$
```

Slave DNS configuration:

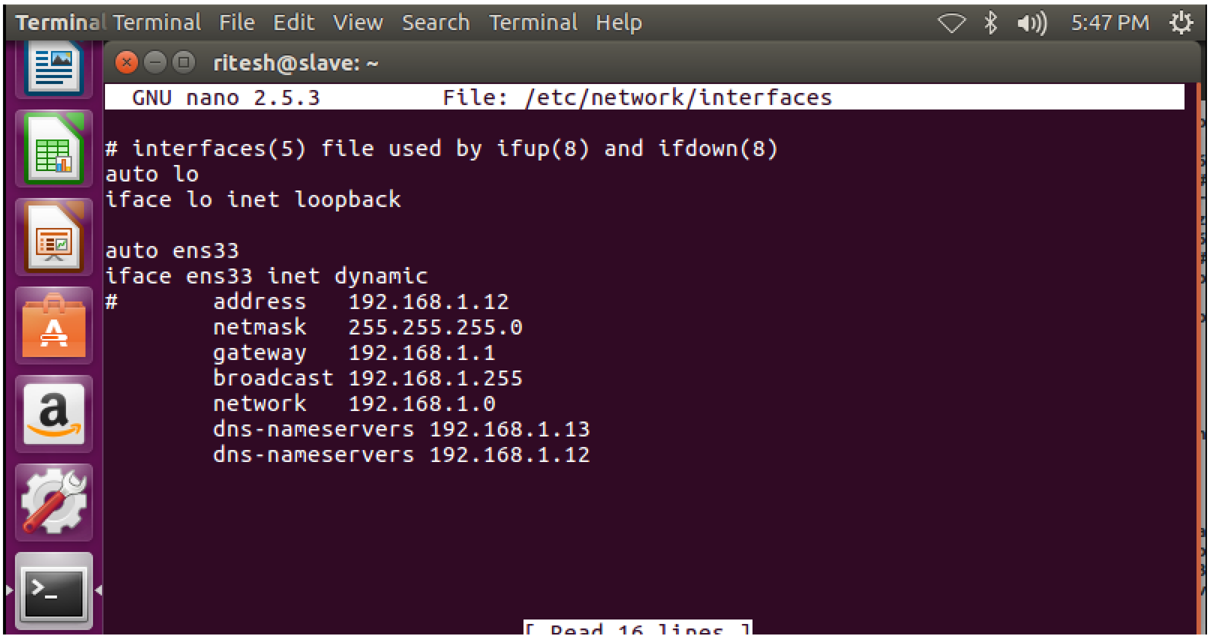
1) Install all the required updates and packages related to BIND9 using following commands.

sudo apt-get update

sudo apt-get install bind9

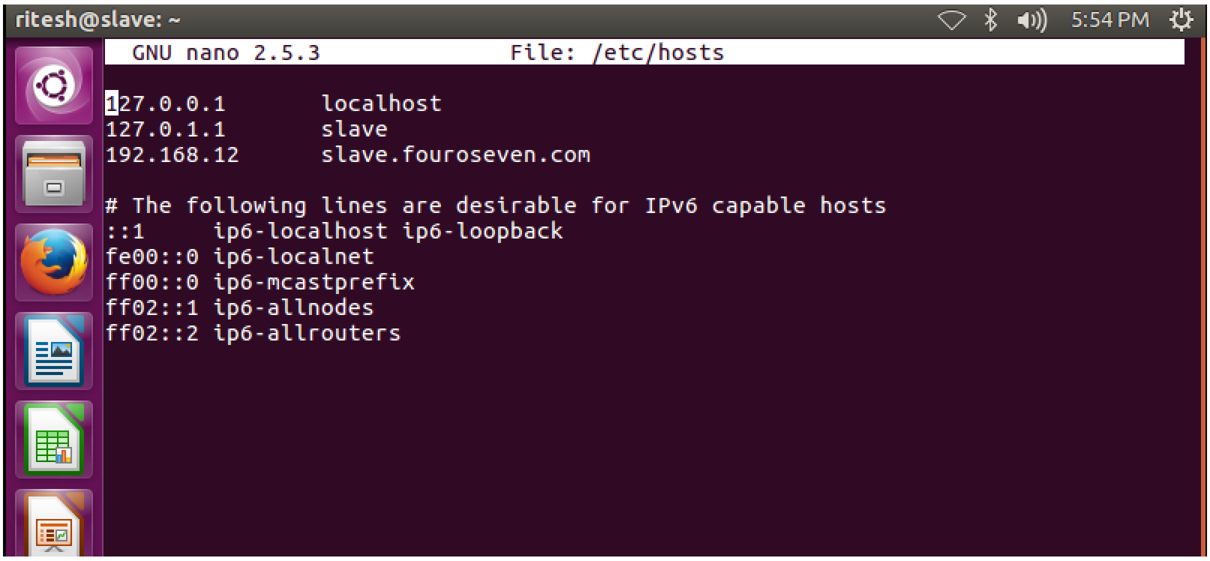
2) Open network interfaces file to make changes as in the below screenshot using the following command.

sudo nano /etc/network/interfaces



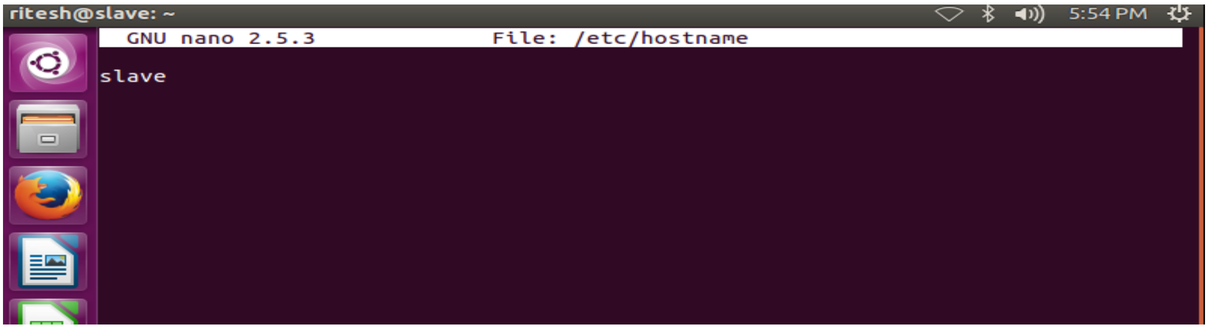
```
  GNU nano 2.5.3           File: /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet dynamic
#        address   192.168.1.12
         netmask   255.255.255.0
         gateway   192.168.1.1
         broadcast 192.168.1.255
         network   192.168.1.0
         dns-nameservers 192.168.1.13
         dns-nameservers 192.168.1.12



                               [ Read 16 lines ]
```

3) Open hosts file by using the following command and make the changes same as in the below screenshot.
sudo nano /etc/hosts



```
  GNU nano 2.5.3                File: /etc/hosts

127.0.0.1       localhost
127.0.1.1       slave
192.168.12      slave.fouroseven.com

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

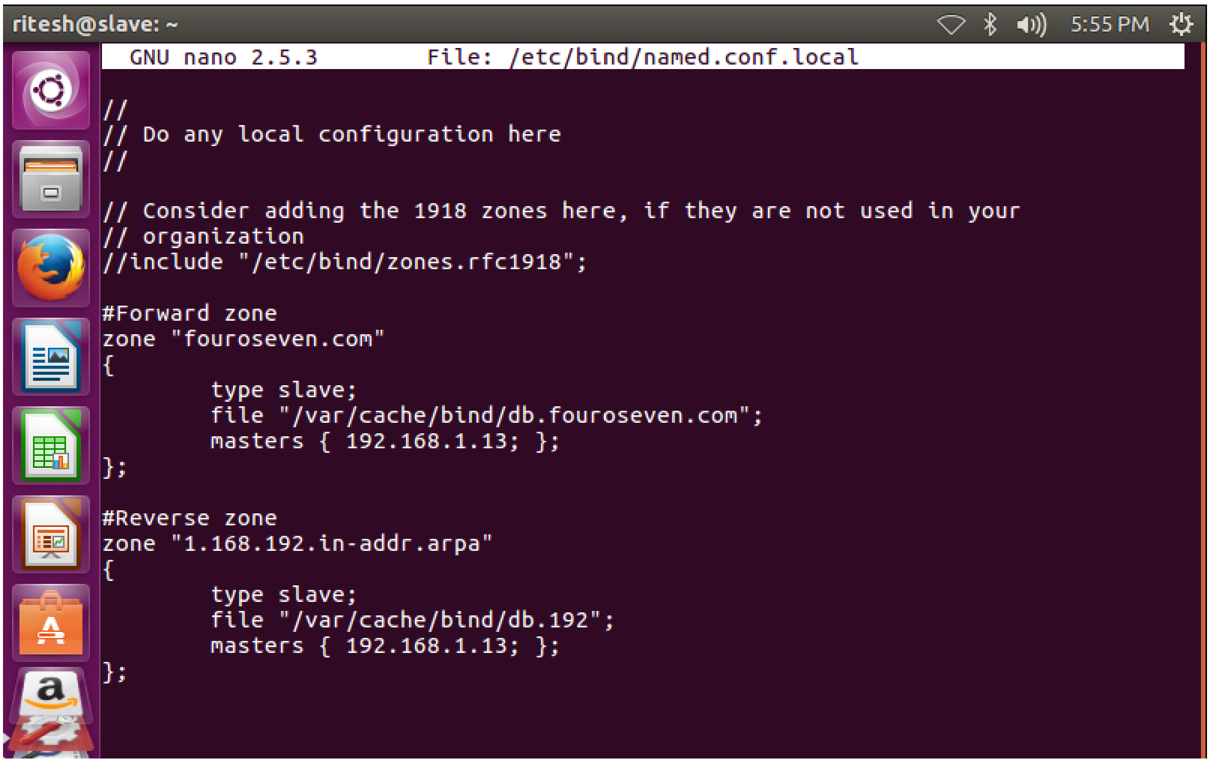
4) Open hostname file using the following command and make below changes.
sudo nano /etc/hostname



```
  GNU nano 2.5.3              File: /etc/hostname

slave
```

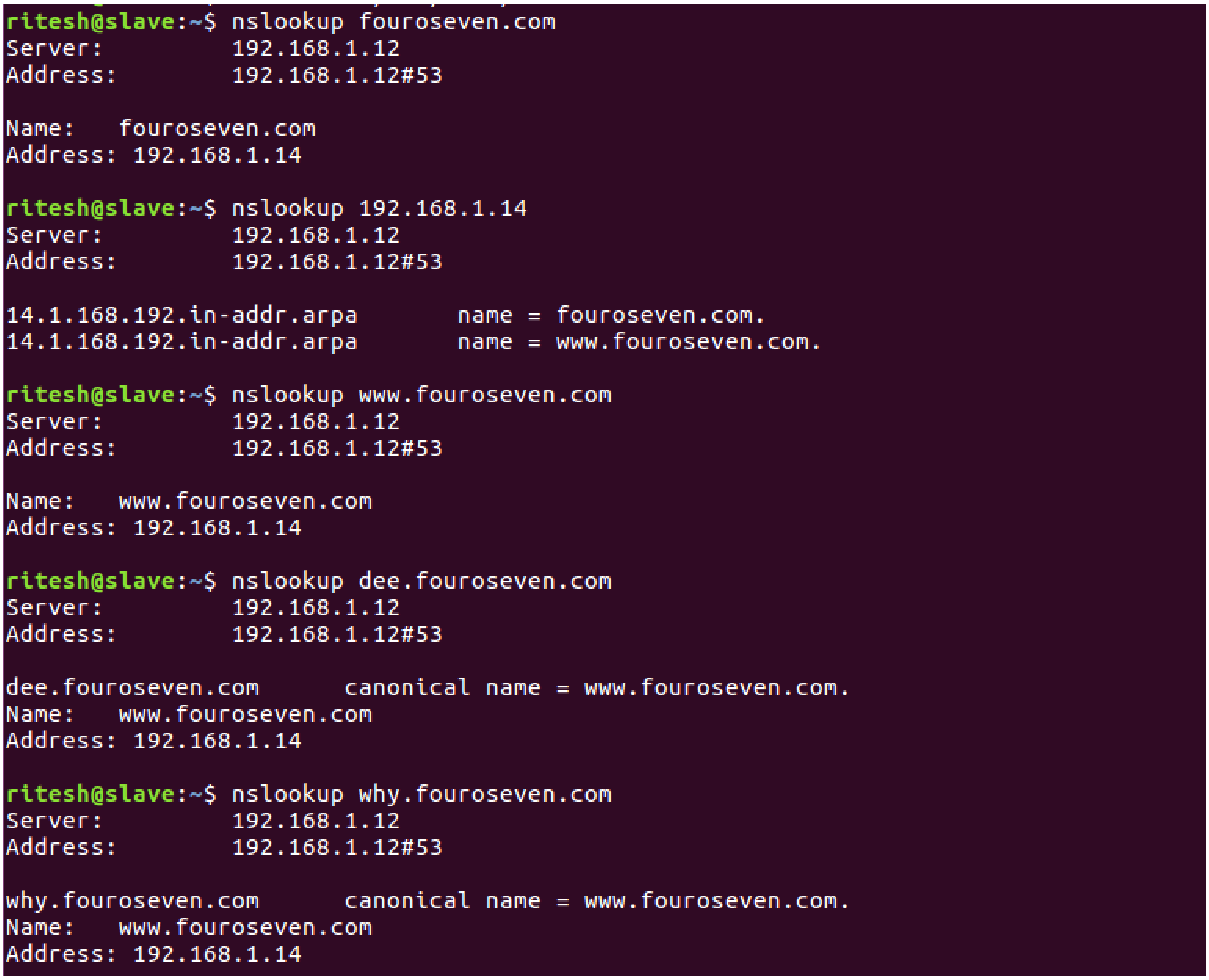
5) Reboot the virtual machine by using the following command to apply changes.
sudo init 6
6) Open the named.conf.local file by using the following command and make the changes as below.
Sudo nano /etc/bind/named.conf.local

7) Open the named.conf.local file by using the following command and make the changes as below.
Sudo nano /etc/bind/named.conf.local

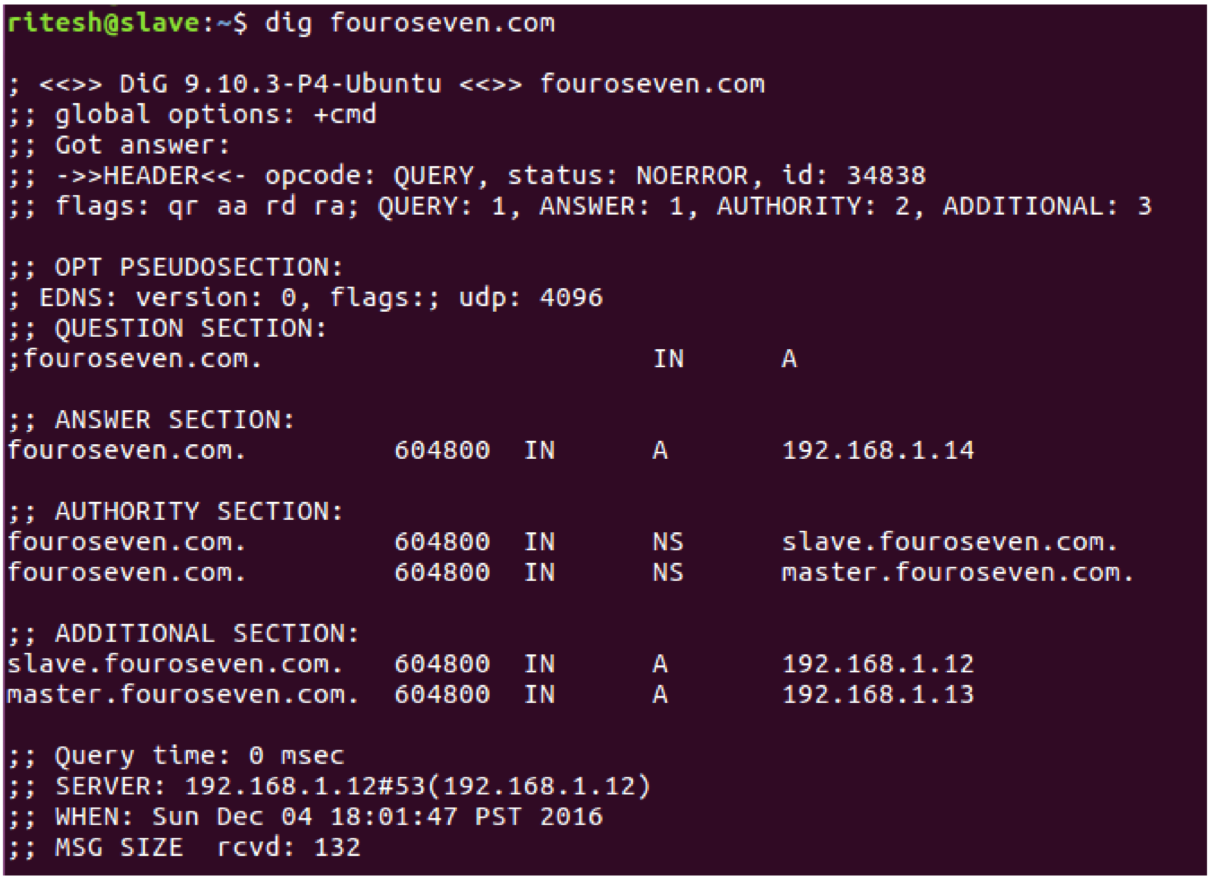


```
  GNU nano 2.5.3          File: /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

#Forward zone
zone "fouroseven.com"
{
        type slave;
        file "/var/cache/bind/db.fouroseven.com";
        masters { 192.168.1.13; };
};

#Reverse zone
zone "1.168.192.in-addr.arpa"
{
        type slave;
        file "/var/cache/bind/db.192";
        masters { 192.168.1.13; };
};
```

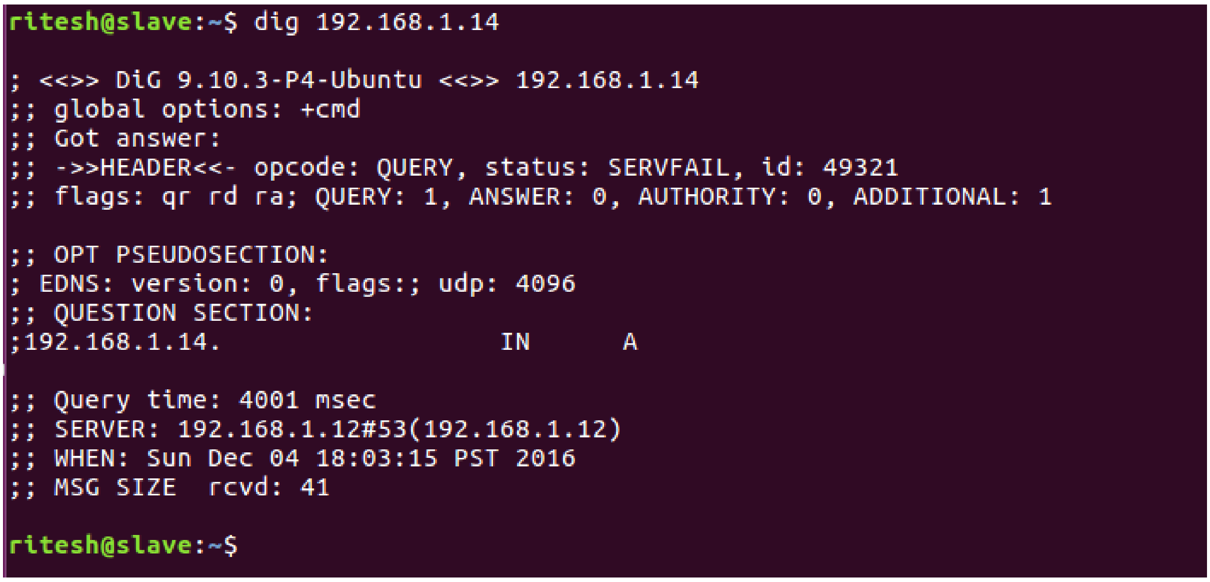
Test results for records updated in slave:



```
ritesh@slave:~$ nslookup fouroseven.com
Server:         192.168.1.12
Address:        192.168.1.12#53

Name:   fouroseven.com
Address: 192.168.1.14

ritesh@slave:~$ nslookup 192.168.1.14
Server:         192.168.1.12
Address:        192.168.1.12#53

14.1.168.192.in-addr.arpa       name = fouroseven.com.
14.1.168.192.in-addr.arpa       name = www.fouroseven.com.

ritesh@slave:~$ nslookup www.fouroseven.com
Server:         192.168.1.12
Address:        192.168.1.12#53

Name:   www.fouroseven.com
Address: 192.168.1.14

ritesh@slave:~$ nslookup dee.fouroseven.com
Server:         192.168.1.12
Address:        192.168.1.12#53

dee.fouroseven.com      canonical name = www.fouroseven.com.
Name:   www.fouroseven.com
Address: 192.168.1.14

ritesh@slave:~$ nslookup why.fouroseven.com
Server:         192.168.1.12
Address:        192.168.1.12#53

why.fouroseven.com      canonical name = www.fouroseven.com.
Name:   www.fouroseven.com
Address: 192.168.1.14
```

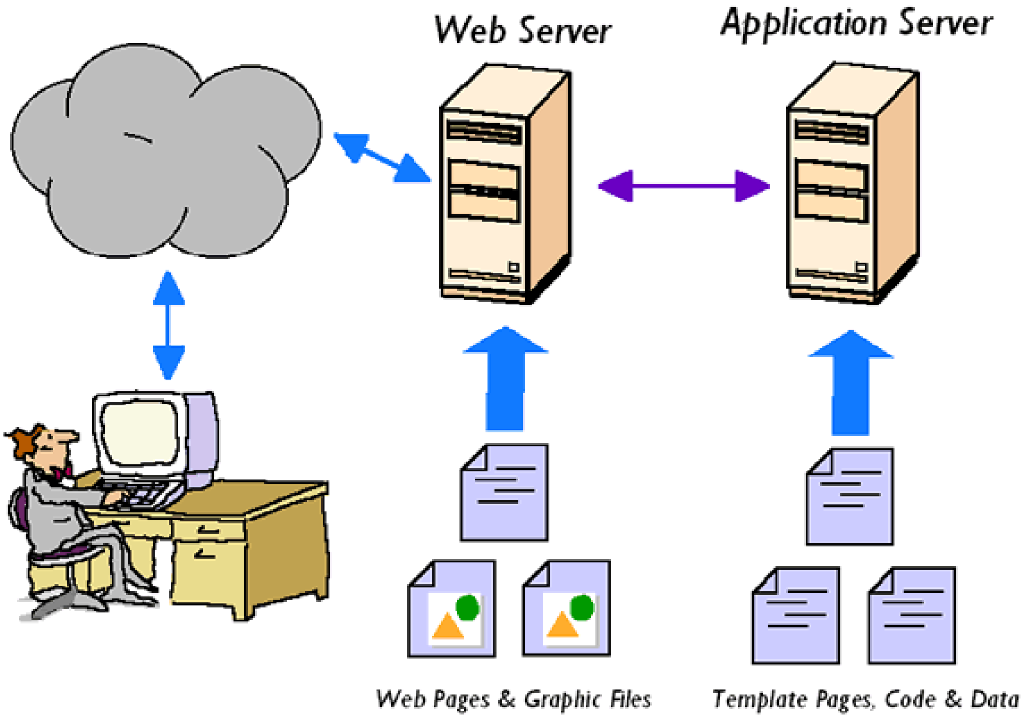


```
ritesh@slave:~$ dig fouroseven.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> fouroseven.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34838
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;fouroseven.com.                        IN      A

;; ANSWER SECTION:
fouroseven.com.         604800  IN      A       192.168.1.14

;; AUTHORITY SECTION:
fouroseven.com.         604800  IN      NS      slave.fouroseven.com.
fouroseven.com.         604800  IN      NS      master.fouroseven.com.

;; ADDITIONAL SECTION:
slave.fouroseven.com.   604800  IN      A       192.168.1.12
master.fouroseven.com.  604800  IN      A       192.168.1.13

;; Query time: 0 msec
;; SERVER: 192.168.1.12#53(192.168.1.12)
;; WHEN: Sun Dec 04 18:01:47 PST 2016
;; MSG SIZE  rcvd: 132
```



```
ritesh@slave:~$ dig 192.168.1.14

; <<>> DiG 9.10.3-P4-Ubuntu <<>> 192.168.1.14
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 49321
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;192.168.1.14.                  IN      A

;; Query time: 4001 msec
;; SERVER: 192.168.1.12#53(192.168.1.12)
;; WHEN: Sun Dec 04 18:03:15 PST 2016
;; MSG SIZE  rcvd: 41

ritesh@slave:~$
```

WEB SERVER:



A Web Server is a host machine, or workstation, which has the capability to host multiple web sites. The first step is to build a list of available options for the type of Web Server we will use for our project. The available Web Servers are hosted by either Windows (Windows Web Server) or Linux (APACHE HTTP Server or NGINX). There are a lot of differences between the previously mentioned web servers, but the most important factor for a network engineer with low funding would be the cost of the system. Windows Web Servers require purchasing a license to be able to download and configure their web server. On the other hand, Linux is an open source system which provides free license for hosting websites via their web servers. Therefore, from our previous discussion, we exclude Windows Web Servers from our list of options. The other option we are left with is Apache. Apache uses a parent process, which dispatches child processes, and each HTTP request is handled by a separate process. Apache offers a trade-off between speed and compatibility with add- on modules, hence providing more flexibility in terms of server configuration options. For our project, the web server is required to host a small number of websites for a small network. Since speed will not be a major problem in our design, therefore we will configure and implement Apache2 web server.

The second step is to decide the type of host machine or workstation that will host or web server and web sites. There are multiple options, including virtual machines, standalone machines and computers. One of the drawbacks of using Server-Client architecture is that we need an always-on device that will host the web server and serve incoming requests. Therefore, a standalone machine, that supports client-server architecture, is the preferred choice, but since we are required to demonstrate our networking project, hence we created dual-boot for our laptops, hosting Linux on a separate partition than Windows.

APACHE2 WEB SERVER Configuration:

Steps to install and configure the Web server on Ubuntu 16.04 LTS:

Step 1

We need to install apache2 on our webserver:

Command:
sudo apt-get update

apt-get install apache2
In CLI:
netstat –a | more

This is to check if the web server is listening on port 80.

Restart the web server:

```
/etc/init.d/apache2 stop
/etc/init.d/apache2 start
```

To check that Apache2 is successfully installed, open a web browser and type the following

http://localhost

Step 2

Creating Directory files
After successfully installing Apache2, we'll create the html file that will contain our web site information and design. To create the directory folder, use the following command

sudo mkdir /var/www/html/fouroseven.com/

Change the properties of the directory to allow the web server to read the html file
sudo chmod 755 /var/www/html/
sudo chown $USER:$USER /var/www/html/fouroseven.com/

Create and edit sample.html to the layout you prefer
sudo nano /var/www/html/fouroseven.com/sample.html

Step 3

Configuring Apache2 apache2.conf
Now we have to configure it to meet our needs. The Apache2 configuration file is apache2.conf and can be accessed using the following command

```
sudo nano /etc/apache2/apache2.conf
```

We are required to do the following modifications, add the following lines to apache2.conf:

```
<Directory /var/www/html/fouroseven.com/>

Options Indexes FollowSymLinks AllowOverride None

Require all granted

</Directory>
```

Step 4

Configuring Apache2 dir.conf
Since we didn't use the default .html file syntax (index.html), we have to add our 'sample.html' to the list of accessible directory indices. Modify the following file

```
sudo nano /etc/apache2/modsavailable/dir.conf
```

Append sample.html to the list of indices

```
<IfModule mod_dir.c>
Directory Index index.html index.cgi index.pl index.php index.xhtml index.htm sample.html
</IfModule>
```

Step 5

Configuring Apache2 000default.conf
000default.conf is used by the Server to identify itself. For our web site, we need to modify/add the following lines to direct the web server to our web site :

```
ServerAdmin webmaster@localhost
ServerName fouroseven.com
DocumentRoot /var/www/html/fouroseven.com
```

Step 6

Restart Apache2
Now that the Apache2 configuration is complete, we need to reload the web server, so that the new configurations can be implemented

```
sudo service Apache2 restart
```

Step 7

Verification
To verify that our changes have been successfully configured and implemented, simply reload the web site (you might need to delete your browser's cache)

http://localhost

FIREWALL:

A firewall is a system that provides network security by filtering incoming and outgoing network traffic based on a set of user-defined rules. In general, the purpose of a firewall is to reduce or eliminate the occurrence of unwanted network communications while allowing all legitimate communication to flow freely. In most server infrastructures, firewalls provide an essential layer of security that combined with other measures; prevent attackers from accessing your servers in malicious ways.

Firewall Rules:

Network traffic that travels through a firewall is matched against rules to determine if it should be allowed to pass through or not.

Suppose there is a server with this list of firewall rules that apply to incoming traffic:

- Accept new and established incoming traffic to the public network interface on port 80 and 443 (HTTP and HTTPS web traffic)
- Drop incoming traffic from IP addresses of the non-technical employees in your office to port 22 (SSH)
- Accept new and established incoming traffic from your office IP range to the private network interface on port 22 (SSH)

Commands:

1. Install ip-tables persistent
   "sudo apt-get iptables-persistent"

2. Start the service using the command
   "sudo service iptables-persistent start"

3. Accepts and deny the ports according to the necessity
   "sudo iptables –A INPUT –p tcp –dport 22 –j ACCEPT"
   "sudo iptables –A INPUT –p tcp –dport 80 –j ACCEPT"
   "sudo iptables –A INPUT –j DROP"

   “sudo iptables -A INPUT -p tcp -s 192.168.1.25 --dport 80 -j REJECT”

BACKUP:

Why do we need backup?

This is required to take the backups of the HTML page webserver at constant intervals and make sure that we do not lose the HTML page and this can be hosted on to another IP.

Configuration:

```
sudo ssh-keygen -t rsa #Generating a pair of public keys
```

```
ssh maheshwar404@192.168.1.16 mkdir -p .ssh #creating the .ssh on the host machine
```

```
cat .ssh/id_rsa.pub | ssh maheshwar404@192.168.1.16 'cat >> .ssh/authorized_keys'
```

# appending the Web server's public key to host

```
sudo tar –cvpzf /home/webserver/minbkup.tar.gz /var/www/html/fouroseven.com/sample.html
```

#compress the backup and keep it in the server.

```
sudo crontab –e
* * * * * sudo tar -cvpzf

/home/webserver/minbkup.tar.gz/var/www/html/fouroseven.com/sample.html

* * * * * sudo scp /home/webserver/minbkup.tar maheshwar404@192.168.1.16:home/maheshwar404/backup
```

# We do Cron jobs to take backup every minute

Algorithm and Flowchart:

1. A client machine tries connecting to the server.
2. For this, client sends an IP address request to the DHCP server.
3. 'If' DHCP request is successful,

   DHCP server will reply with an IP address.

   'Else'

   DHCP will send a reply with 'request failed'

   In this case, reconnect or contact the network administrator for help resolving the issue.

4. Client wants to access the web page.

“If" the domain name is correct, a request will be sent to DNS for IP address of the domain.

DNS will reply with IP address of the website.

''Else if'' DNS reply fails
An error message will be displayed saying, ‘server not found.’

''Else if '' URL entered is wrong
An error message will be displayed saying, ‘webpage unavailable.’

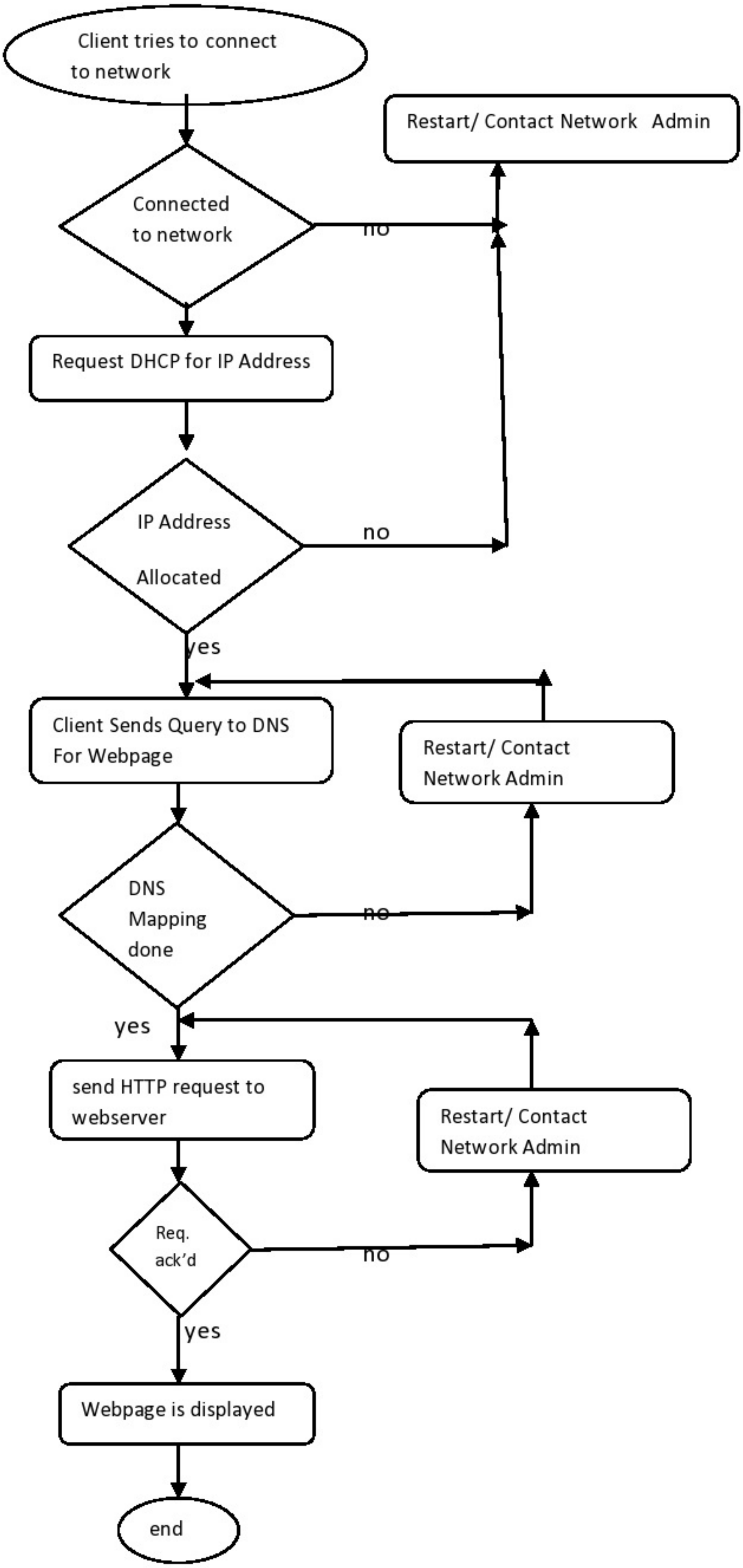
''Retry''

5. Client accessed the web server. Now he sends HTTP request to the server.

''If'' the request is successful,
Web page will be displayed

''Else''
Error message like ‘no data received’ will be displayed.

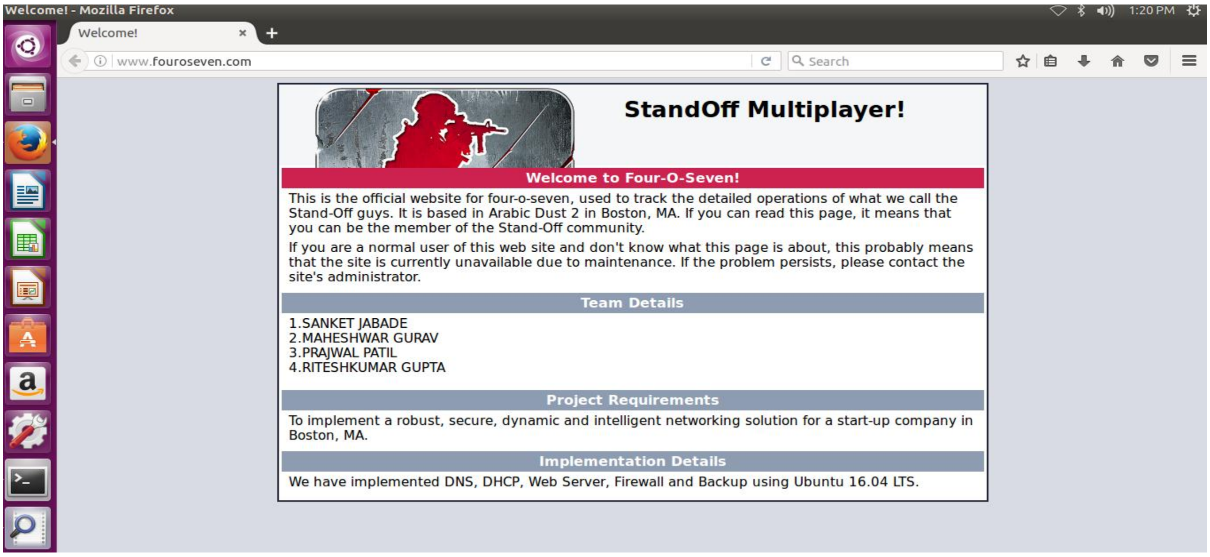
''Retry''


Client tries to connect to network
Connected to network
no
Restart/ Contact Network Admin
Request DHCP for IP Address
IP Address Allocated
no
yes
Client Sends Query to DNS For Webpage
Restart/ Contact Network Admin
DNS Mapping done
no
yes
send HTTP request to webserver
Restart/ Contact Network Admin
Req. ack'd
no
yes
Webpage is displayed
end

Integration:
We have used Linux Ubuntu for implementing DHCP, DNS, Webserver, Firewall and backup. We have used the 192.168.1.0/24 network. 192.168.1.1 is our default gateway and 192.168.1.255 is the broadcast address. IP address assigned:

Master: **192.168.1.13** and **2001:0db8:1:1::13**
Slave: **192.168.1.12** and **2001:0db8:1:1::12**
Webserver: **192.168.1.14** and **2001:0db8:1:1::14**
Back-up Server: **192.168.1.16**
DHCP: **192.168.1.5** and **2001:0db8:1:1::5**

The client can also access the webpage with the url, www.fouroseven.com. The below page is displayed.



StandOff Multiplayer!

Welcome to Four-O-Seven!

This is the official website for four-o-seven, used to track the detailed operations of what we call the Stand-Off guys. It is based in Arabic Dust 2 in Boston, MA. If you can read this page, it means that you can be the member of the Stand-Off community.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Team Details

1.SANKET JABADE
2.MAHESHWAR GURAV
3.PRAJWAL PATIL
4.RITESHKUMAR GUPTA

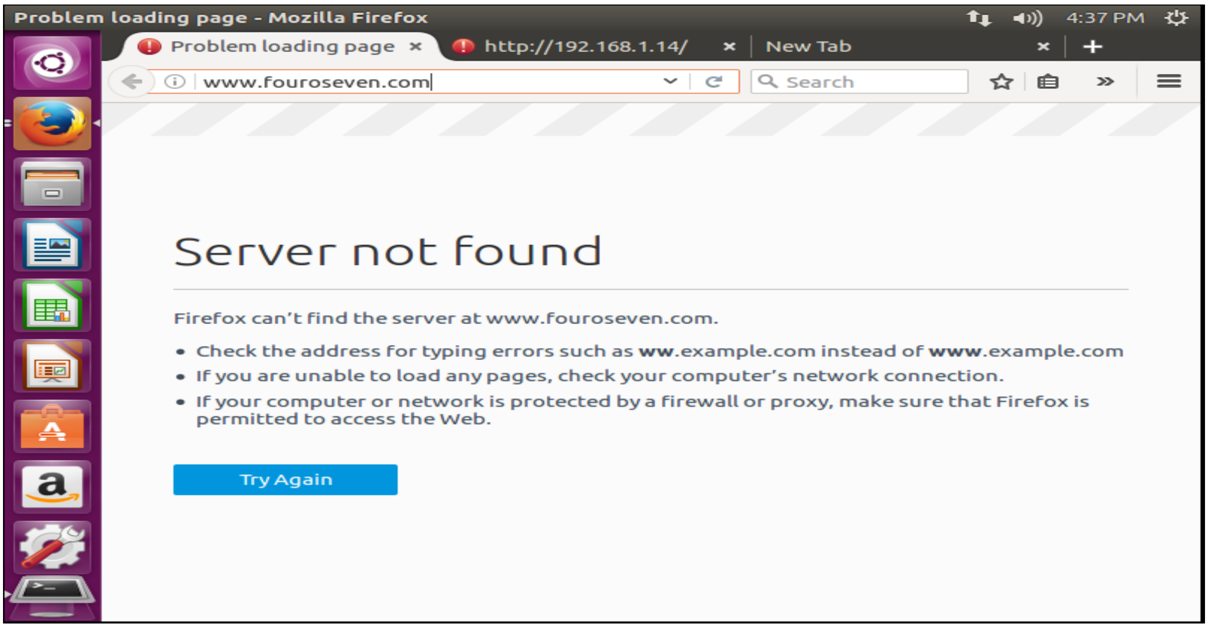
Project Requirements

To implement a robust, secure, dynamic and intelligent networking solution for a start-up company in Boston, MA.

Implementation Details

We have implemented DNS, DHCP, Web Server, Firewall and Backup using Ubuntu 16.04 LTS.

To check firewall we have blocked 192.168.1.25 and when it tries to access the webpage, it’s not displayed.



Server not found

Firefox can’t find the server at www.fouroseven.com.

- Check the address for typing errors such as **ww**.example.com instead of **www**.example.com
- If you are unable to load any pages, check your computer’s network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Citation:

Websites:

https://www.ostechnix.com/install-and-configure-dns-server-ubuntu-16-04-lts/

https://help.ubuntu.com/

https://www.digitalocean.com

http://askubuntu.com/questions/

Textbook:

Computer Networking, A Top-Down Approach by James F Kurose and Keith W. Ross (Fifth Edition)