Chapter . 7

Network Configuration for Linux

VEDHA DATE: / /201 Page No.:

Network Configuration Tools.

All the initial network configuration for Red Hat Linux is typically done during installations.

There are two basic types of configuration.

1) command line configuration.

configuring the network using this method can be done in two ways.

a) use command to change your current settings.

The commands are.

1) /sbin / ifconfig command
2) /sbin / route command
3) /sbin / netstat command.

b) edit seven different configuration files.

The files are

1) /etc / hosts
2) /etc/ services
3) /etc/ nsswitch . conf
4) /etc/ resolv . conf.
5) /etc / host . conf.
6) /etc / hostname
7) /etc/sysconfig / network

II Graphical Configuration

linuxconf tool is a graphical tool used to configure a network.

To open linuxconf tool

1) type /sbin / linuxconf command at command line
2) start it from KDE or GNOME.

To start linuxconf from KDE or GNOME

click on Networking → click on client Tasks

VEDHA
DATE: / /201
Page No.:

→ select Hostname & IP Network devices.

Network configuration commands.

1) ifconfig command.
- It is used to configure network interface.
Also, To create an IP alias to allow more than one IP address on the NIC
- To change machine's IP address, Netmask or broadcast address.
- To activate & deactivate NIC [Network Interface card].

syntax

/sbin/ifconfig [Network device] options

Ex :- # /sbin/ifconfig

lo Link encap : Local Loopback

What Is Samba Server And How To Setup Samba Server In Ubuntu Linux

Samba is an open-source software suite that runs on Unix/Linux based platforms but is able to communicate with Windows clients like a native application. So Samba is able to provide this service by employing the Common Internet File System (CIFS).
At the heart of this CIFS is the Server Message Block (SMB) protocol. Samba does this by performing these 4 key things -

- File & print services
- Authentication and Authorization
- Name resolution
- Service announcement (browsing)

Samba can be run on many different platforms including Linux, Unix, OpenVMS and operating systems other than Windows and allows the user to interact with a Windows client or server natively. It can basically be described as the Standard Windows interoperability suite of programs for Linux and Unix.

How to setup Samba Server on Ubuntu?

Let's take a look at setting up a Samba Server on Ubuntu to share files with Windows clients. Note that we will create two forms of Samba server, one setup won't require a password to share files with any client on the network which is the anonymous share and another setup will require setting up users and permissions.

1. Installation Of Samba

Binary packages of Samba are included in almost any Linux distribution. There are also some packages available at the Samba homepage. In fact, there are now several GUI interfaces to Samba available to help with configuration and management. This tutorial will set up Samba via the Linux terminal.

We install the samba package from the terminal in Ubuntu with the following code -

```
sudo apt-get update

sudo apt-get install samba
```

2. Configure File Server - Anonymous Share

a. Create shared folder called "shared folder".

```
sudo mkdir -p /srv/samba/sharedfolder
```

b. Allow anyone to access and store files in folder.

```
sudo chown nobody:nogroup /srv/samba/sharedfolder/
```

c. Edit configuration file to enable sharing.

Make a backup of the configuration file before editing

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.old
```

Open the conf file and make the following changes -

```
sudo vi /etc/samba/smb.conf
```

Add the following at the end of the file to enable sharing -

d. Restart smbd service -

```
sudo systemctl restart smbd.service nmbd.service
```

3. Configure File Server - Secured Share

a. Create shared folder called "securedfolder".

```
sudo mkdir -p /srv/samba/securedfolder
```

b. Create a new user group named "selected".
sudo addgroup selected

c. Modify permission and ownership for the folder.
sudo chown root:selected /srv/samba/securedfolder/

sudo chmod 770 /srv/samba/securedfolder/

d. Edit configuration file to allow sharing

Make a backup of the configuration file before editing,
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.old

Open the conf file and make the following changes -
sudo vi /etc/samba/smb.conf

Add the following at the end of the file to enable sharing -
[sharedfolder]

comment = secured shared folder

path = /srv/samba/securedfolder

Valid users = @selected

guest ok = no

writable = yes

browsable = yes

e. Restart smbd service -
sudo systemctl restart smbd.service nmbd.service

f. Once Samba has restarted, use this command to check your smb.conf for any syntax errors.
testparm

g. Add new users

We are going to create and add a user "Jack" to the user group "selected" with a restricted shell access.
sudo useradd jack -s /usr/sbin/nologin -G selected

To add a password for the user -
sudo smbpasswd -a jack

h. Add existing user Jane to the group to the group "selected".
sudo usermod jane -G selected

Conclusion
You should be able to access and browse files from these servers from your Windows client. If you do not see your client automatically, you can try accessing it va its IP address. You can access the Ubuntu sharing in Windows by entering "\\sharedfolder" or "\\securedfolder" in the windows search field of the menu or use the network browser of the Windows file explorer to connect to the share. In the case of the secured share, the user will be required to enter the password before being able to access the shared folder.

DHCP Server Configuration:

Dynamic Host Configuration Protocol (DHCP) automatically assigns IP addresses and other network configuration information (subnetmask, broadcast address, etc) to computers on a network. A client configured for DHCP will send out a broadcast request to the DHCP server requesting an address. The DHCP server will then issue a "lease" and assign it to that client. The time period of a valid lease can be specified on the server. DHCP reduces the amount of time required to configure clients and allows one to move a computer to various networks and be configured with the appropriate IP address, gateway, and subnet mask. For ISP's it conserves the limited number of IP addresses it may use. DHCP servers may assign a "static" IP address to specified hardware. Microsoft NetBios information is often included in the network information sent by the DHCP server.

DHCP assignment:

1. Lease Request: Client broadcasts request to DHCP server with a source address of 0.0.0.0 and a destination address of 255.255.255.255. The request includes the MAC address which is used to direct the reply.
2. IP lease offer: DHCP server replies with an IP address, subnet mask, network gateway, name of the domain, name servers, duration of the lease and the IP address of the DHCP server.
3. Lease Selection: Client recieves offer and broadcasts to al DHCP servers that will accept given offer so that other DHCP server need not make an offer.
4. The DHCP server then sends an ack to the client. The client is configured to use TCP/IP.
5. Lease Renewal: When half of the lease time has expired, the client will issue a new request to the DHCP server.

DHCP server installation:

- Red Hat/CentOS/Fedora: `rpm -ivh dhcp-x.xxx.elx.i386.rpm`
- Ubuntu/Debian 8: `apt-get install dhcp3-server`
  (Later releases of Ubuntu (11.04) used the busybox release known as udhcpd and the configuration is NOT shown here)

Starting DHCP server:

- Red Hat/CentOS/Fedora: `service dhcpd start`
  (or `/etc/rc.d/init.d/dhcpd start` for Red Hat, Fedora and CentOS Linux distributions)
- Ubuntu/Debian: `/etc/init.d/networking restart`

Sample DHCP server config file: (DHCP v3.0.1)

- Red Hat/CentOS/Fedora: `/etc/dhcpd.conf`
  (See `/usr/share/doc/dhcp-3.X/dhcp.conf.sample`)
  [Potential Pitfall]: Its `/etc/dhcpd.conf` NOT `/etc/dhcp.conf` !!
- Ubuntu/Debian: `/etc/default/dhcp3-server`

```
                                                    # Required for dhcp 3.0+
ddns-update-style interim;
/ Red Hat 8.0+
ignore client-updates;

subnet 192.168.1.0 netmask 255.255.255.0 {

                                                    # Range of IP addresses

        range 192.168.1.128 192.168.1.254;
to be issued to DHCP clients
```

```
        option subnet-mask              255.255.255.0;    # Default subnet mask to
be used by DHCP clients

        option broadcast-address        192.168.1.255;    # Default
broadcastaddress to be used by DHCP clients

        option routers                  192.168.1.1;      # Default gateway to be
used by DHCP clients

        option domain-name              "your-domain.org";

        option domain-name-servers      40.175.42.254, 40.175.42.253;          #
Default DNS to be used by DHCP clients

        option netbios-name-servers     192.168.1.100;    # Specify a WINS server
for MS/Windows clients.

                                                          # (Optional. Specify if
used on your network)


#       DHCP requests are not forwarded. Applies when there is more than one
ethernet device and forwarding is configured.
#       option ipforwarding off;


        default-lease-time 21600;                         # Amount of time in
seconds that a client may keep the IP address

        max-lease-time 43200;


        option time-offset              -18000;           # Eastern Standard Time

#       option ntp-servers              192.168.1.1;      # Default NTP server to
be used by DHCP clients

#       option netbios-name-servers     192.168.1.1;

# --- Selects point-to-point node (default is hybrid). Don't change this unless you
understand Netbios very well

#       option netbios-node-type 2;


        # We want the nameserver "ns2" to appear at a fixed address.

        # Name server with this specified MAC address will recieve this IP.


        host ns2 {

                next-server ns2.your-domain.com;

                hardware ethernet 00:02:c3:d0:05:83;

                fixed-address 40.175.42.254;

        }


        # Laser printer obtains IP address via DHCP. This assures that the

        # printer with this MAC address will get.this IP address every time.
```

```
        host laser-printer-lex1 {
                hardware ethernet 08:00:2b:4c:a3:82;
                fixed-address 192.168.1.120;
        }
}
```

Test configuration file for errors with the following command: /etc/rc.d/init.d/dhcpd configtest.
(Other distributions may use: /usr/sbin/dhcpd -t)
Note: The MAC addresses for the static address name server (ns2.*your-domain.com*), can be obtained with either of the two commands:

- /sbin/ip addr show:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:02:c3:d0:e5:83 brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.214/24 brd 192.168.42.255 scope global dynamic eth0
       valid_lft 82646sec preferred_lft 82646sec
    inet6 f080::477:3e0e:d5fd:803a/64 scope link
       valid_lft forever preferred_lft forever
```

OR

- /sbin/ifconfig:

```
eth0      Link encap Ethernet  HWaddr 00:02:C3:D0:E5:83
          inet addr:192.168.42.214  Bcast:192.168.42.255  Mask:255.255.255.0
          inet6 addr: fe80::202:3ff:fed0:e583/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4070 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3878 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

```
RX bytes:3406445 (3.2 MiB)  TX bytes:439612 (429.3 KiB)
```

When dhcpd is running it will generate entries in the file: `/var/lib/dhcp/dhcpd.leases`

```
lease 192.168.1.128 {
starts 2 2004/12/01 20:07:05;
ends 3 2004/12/02 08:07:05;
hardware ethernet 00:00:e8:4a:2c:5c;
uid 01:00:00:e8:4c:5d:31;
client-hostname "Node1";
}
```

Options:

- ddns-update-style:
  - interim: allows your DHCP server to update a DNS server whenever it hands out a lease. Allows your DNS server to know which IP addresses are associated with which computers in your network. Requires that your DNS server support DDNS (Dynamic DNS).
  - none: to disable dynamic DNS updates or DNS is not supporting DDNS.
  - ad-hoc: been deprecated and shouldn't be used
- Default options (Red Hat/CentOS/Fedora) are set in `/etc/sysconfig/dhcpd`

LANs separated by routers: In order to have your DHCP broadcast pass through a router on the the next network, one must configure the router to allow DHCP relay. (Cisco: `ip-helper address`, Juniper: `dhcp-relay`) The local lan `subnet { }` configuration must come before the configuration directives of the remote LANs.

Look for errors in `/var/log/messages`

See dhcp-options man page below.

DHCP man pages:

- Client: dhclient: DHCP client daemon (replaces dhcpcd)
  - dhclient.conf: DHCP client configuration file (`/etc/dhcp/dhcpd.conf`)
  - dhclient.leases: DHCP client lease database (`/var/lib/dhcpd/dhcpd.leases`)
  - dhclient-script: DHCP client network configuration script
- Server: dhcpd: Dynamic Host Configuration Protocol Server daemon
  - dhcpd.conf: dhcpd configuration file
  - dhcpd.leases: dhcpd DHCP client lease database
  - dhcp-options: dhcpd Dynamic Host Configuration Protocol options
- dhcrelay: bootp relay agent. One DHCP server to service multiple network segments.
- omshell: OMAPI Command Shell - interactive way to connect to, query, and possibly change, the ISC DHCP Server's state via OMAPI One can make the changes while the server is running.

DHCP RFC's: RFC2131, RFC1541 (obsolete), RFC2132

Note: DHCP client will overwrite your `/etc/resolv.conf` file with new information recieved from the DHCP server.

Introduction to the Network File System (NFS) on Linux

What is the Network File System (NFS)?

The Network File System (NFS) is a way of mounting Linux discs/directories over a network. An NFS server can *export* one or more directories that can then be mounted on a remote Linux machine. Note, that if you need to mount a Linux filesystem on a Windows machine, you need to use Samba/CIFS instead.

NFS is a way of mounting Linux discs/directories over a network..

Why use the Network File System (NFS)?

The main use of NFS in the home context, is to share out data on a central server (-for example, your music collection) to all the PCs in the house. This way, you have a single copy of data (-hopefully, well backed up) accessible from a central location.

Can I use Samba (CIFS) Instead?

The short answer is "Yes" -but the consensus opinion is: "only use Samba if you *have* to"! If you have a Linux server and a Linux client, those two should share data via NFS rather than Samba/CIFS.

Samba was designed to let Windows machines talk to machines running other types of O/S - it therefore like a translator. Having Samba connect two Linux machines is like two native english speakers trying to communicate via a native spanish speaker (-who has to internally convert english to spanish and then spanish back to english), however entertaining that might sound!

If a Linux box needs to talk to another Linux box, they can do so using their native protocols, without any additional overhead or conversion, using NFS - which is why it is much more efficient (-and more reliable, in our experience) than Samba.

When to use NFS and when to use Samba

Here are some examples of when to use Samba and when to use NFS:

Linux	Linux	NFS
Windows	Linux	Samba
Linux	Windows	Samba
Windows	Windows	Samba

To sum up: in a *heterogeneous* network (-i.e. containing more than one O/S), you'd use NFS to connect the Linux members and Samba only when one O/S is talking to a different O/S.

Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all webservers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules. Most WordPress hosting providers use Apache as their web server software. However, WordPress can run on other web server software as well.

What is a Web Server?

Wondering what the heck is a web server? Well a web server is like a restaurant host. When you arrive in a restaurant, the host greets you, checks your booking information and takes you to your table. Similar to the restaurant host, the web server checks for the web page you have requested and fetches it for your viewing pleasure. However, A web server is not just your host but also your server. Once it has found the web page you requested, it also serves you the web page. A web server like Apache, is also the Maitre D' of the restaurant. It handles your communications with the website (the kitchen), handles your requests, makes sure that other staff (modules) are ready to serve you. It is also the bus boy, as it cleans the tables (memory, cache, modules) and clears them for new customers.

So basically a web server is the software that receives your request to access a web page. It runs a few security checks on your HTTP request and takes you to the web page. Depending on the page you have requested, the page may ask the server to run a few extra modules while generating the document to serve you. It then serves you the document you requested. Pretty awesome isn't it.

How to Install and Configure DNS Server in Linux
Domain Name Service (DNS) is an internet service that maps IP addresses to fully qualified domain names (FQDN) and vice versa.

BIND stands for Berkley Internet Naming Daemon.

BIND is the most common program used for maintaining a name server on Linux.

In this tutorial, we will explain how to install and configure a DNS server.

If you are new to DNS, you should first understand the fundamentals of DNS and how it works.

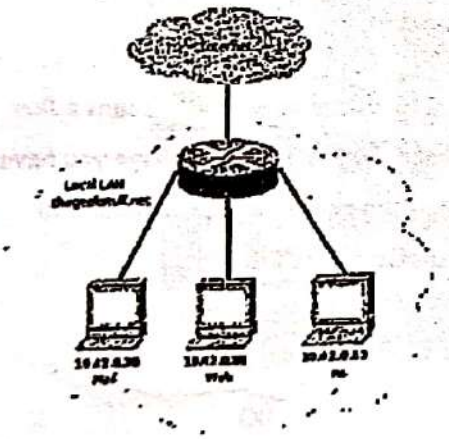
1. Network Information

In this tutorial, we are going to setup a local DNS server for the network shown in the below diagram.

We'll use "thegeekstuff.net" domain as an example for this DNS installation. "mail", "web", "ns" are the hosts that resides within this domain.

It is possible to configure a single system to act as a caching name server, primary/master and secondary/slave. We will configure this DNS as a Primay/Master as well as Caching DNS server.

We'll be installing DNS server on "10.42.0.83".



2. Install Bind

Install the bind9 package using the appropriate package management utilities for your Linux distributions.

On Debian/Ubuntu flavors, do the following:

$ sudo apt-get install bind9

On Redhat/CentOS/Fedora flavors, do the following:

yum install bind9

All the DNS configurations are stored under /etc/bind directory. The primary configuration is /etc/bind/named.conf which will include other needed files. The file named /etc/bind/db.root describes the root nameservers in the world.

3. Configure Cache NameServer

The job of a DNS caching server is to query other DNS servers and cache the response. Next time when the same query is given, it will provide the response from the cache. The cache will be updated periodically.

Please note that even though you can configure bind to work as a Primary and as a Caching server, it is not advised to do so for security reasons. Having a separate caching server is advisable.

All we have to do to configure a Cache NameServer is to add your ISP (Internet Service Provider)'s DNS server or any OpenDNS server to the file /etc/bind/named.conf.options. For Example, we will use google's public DNS servers, 8.8.8.8 and 8.8.4.4.

Uncomment and edit the following line as shown below in /etc/bind/named.conf.options file.

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

After the above change, restart the DNS server.

```
$ sudo service bind9 restart
```

4. Test the Cache NameServer

You can use the dig command to test DNS services. DIG command examplesexplains more about how to perform DNS lookups.

```
$ dig ubuntu.com
```

```
;; Query time: 1323 msec
```

Now when the second time you execute the dig, there should be an improvement in the Query time. As you see below, it took only 3 msec the second time, as it is getting the info from our caching DNS server.

$ dig ubuntu.com

;; Query time: 3 msec

5. Configure Primary/Master Nameserver

Next, we will configure bind9 to be the Primary/Master for the domain/zone "thegeekstuff.net".

As a first step in configuring our Primary/Master Nameserver, we should add Forward and Reverse resolution to bind9.

To add a DNS Forward and Reverse resolution to bind9, edit /etc/bind9/named.conf.local,

```
zone "thegeekstuff.net" {
    type master;
    file "/etc/bind/db.thegeekstuff.net";
};

zone "0.42.10.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/db.10";
};
```

Now the file /etc/bind/db.thegeekstuff.net will have the details for resolving hostname to IP address for this domain/zone, and the file /etc/bind/db.10 will have the details for resolving IP address to hostname.

6. Build the Forward Resolution for Primary/Master NameServer

Now we will add the details which is necessary for forward resolution into /etc/bind/db.thegeekstuff.net.

First, copy /etc/bind/db.local to /etc/bind/db.thegeekstuff.net

```
$ sudo cp /etc/bind/db.local /etc/bind/db.thegeekstuff.net
```

Next, edit the /etc/bind/db.thegeekstuff.net and replace the following.

1. In the line which has SOA: localhost. -- This is the FQDN of the server in charge for this domain. I've installed bind9 in 10.42.0.83, whose hostname is "ns". So replace the "localhost." with "ns.thegeekstuff.net.". Make sure it end's with a dot(.).
2. In the line which has SOA: root.localhost. -- This is the E-Mail address of the person who is responsible for this server. Use dot(.) instead of @. I've replaced with lak.localhost.
3. In the line which has NS: localhost. -- This is defining the Name server for the domain (NS). We have to change this to the fully qualified domain name of the name server. Change it to "ns.thegeekstuff.net.". Make sure you have a "." at the end.

Next, define the A record and MX record for the domain. A record is the one which maps hostname to IP address, and MX record will tell the mailserver to use for this domain.

Once the changes are done, the /etc/bind/db.thegeekstuff.net file will look like the following:

```
$TTL    604800

@   IN  SOA ns.thegeekstuff.net. lak.localhost. (

              1024         ; Serial

            604800         ; Refresh

             86400         ; Retry

           2419200         ; Expire

            604800 )       ; Negative Cache TTL

;

@   IN  NS  ns.thegeekstuff.net.

thegeekstuff.net.   IN    MX    10    mail.thegeekstuff.net.

ns  IN  A   10.42.0.83

web IN  A   10.42.0.80

mail IN A   10.42.0.70
```

6. Build the Reverse Resolution for Primary/Master NameServer

We will add the details which are necessary for reverse resolution to the file /etc/bind/db.10. Copy the file /etc/bind/db.127 to /etc/bind/db.10

$ sudo cp /etc/bind/db.127 /etc/bind/db.10

Next, edit the /etc/bind/db.10 file, and basically changing the same options as /etc/bind/db.thegeekstuff.net

```
$TTL    604800
@   IN  SOA ns.thegeekstuff.net. root.localhost. (
              20     ; Serial
         604800     ; Refresh
          86400     ; Retry
        2419200     ; Expire
         604800 )   ; Negative Cache TTL
;
@   IN  NS  ns.
```

Next, for each A record in /etc/bind/db.thegeekstuff.net, add a PTR record.

```
$TTL    604800
@   IN  SOA ns.thegeekstuff.net. root.thegeekstuff.net. (
              20     ; Serial
         604800     ; Refresh
          86400     ; Retry
        2419200     ; Expire
         604800 )   ; Negative Cache TTL
;
@   IN  NS  ns.
```

83 IN PTR ns.thegeekstuff.net.

70 IN PTR mail.thegeekstuff.net.

80 IN PTR web.thegeekstuff.net.

Whenever you are modifying the file db.thegeekstuff.net and db.10, you need to increment the "Serial" number as well. Typically admin uses DDMMYYSS for serial numbers and when they modify, the change the serial number appropriately.

Finally, restart the bind9 service;

$ sudo service bind9 restart

7. Test the DNS server

Now we have configured the DNS server for our domain. We will test our DNS server by pinging mail.thegeekstuff.net from web.thegeekstuff.net.

If the ping is success, then we have configured the DNS successfully.

You can also use nslookup and dig to test DNS servers.
On web.thegeekstuff.net server, add the following to /etc/resolv.conf

nameserver 10.42.0.83

Now ping, mail.thegeekstuff.net, which should resolve the address appropriately from the DNS server that we just configured.

$ ping mail.thegeekstuff.net

PING mail.thegeekstuff.net (10.42.0.70) 56(84) bytes of data.

64 bytes from mail.thegeekstuff.net (10.42.0.70): icmp_req=1 ttl=64 time=0.482 ms

64 bytes from mail.thegeekstuff.net (10.42.0.70): icmp_req=2 ttl=64 time=0.532 ms