

Disable Root Login Via SSH in RHEL 8 | Forbid SSH Root Login in Linux

```
[root@localhost ssh]# cd /etc/ssh
[root@localhost ssh]# ls -l
total 604
-rw-r--r--. 1 root root      228 Nov 12 05:48 banner.txt
-rw-r--r--. 1 root root  577388 Jan  8 2020 moduli
-rw-r--r--. 1 root root     1716 Jan  8 2020 ssh_config
drwxr-xr-x. 2 root root      28 Nov  6 16:01 ssh_config.d
-rw-----. 1 root root    4439 Nov 12 05:49 sshd_config
-rw-----. 1 root ssh_keys   492 Nov  6 16:13 ssh_host_ecdsa_key
-rw-r--r--. 1 root root    162 Nov  6 16:13 ssh_host_ecdsa_key.pub
-rw-----. 1 root ssh_keys   387 Nov  6 16:13 ssh_host_ed25519_key
-rw-r--r--. 1 root root     82 Nov  6 16:13 ssh_host_ed25519_key.pub
-rw-----. 1 root ssh_keys  2578 Nov  6 16:13 ssh_host_rsa_key
-rw-r--r--. 1 root root    554 Nov  6 16:13 ssh_host_rsa_key.pub
[root@localhost ssh]#
```

```
# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys
```

```
# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

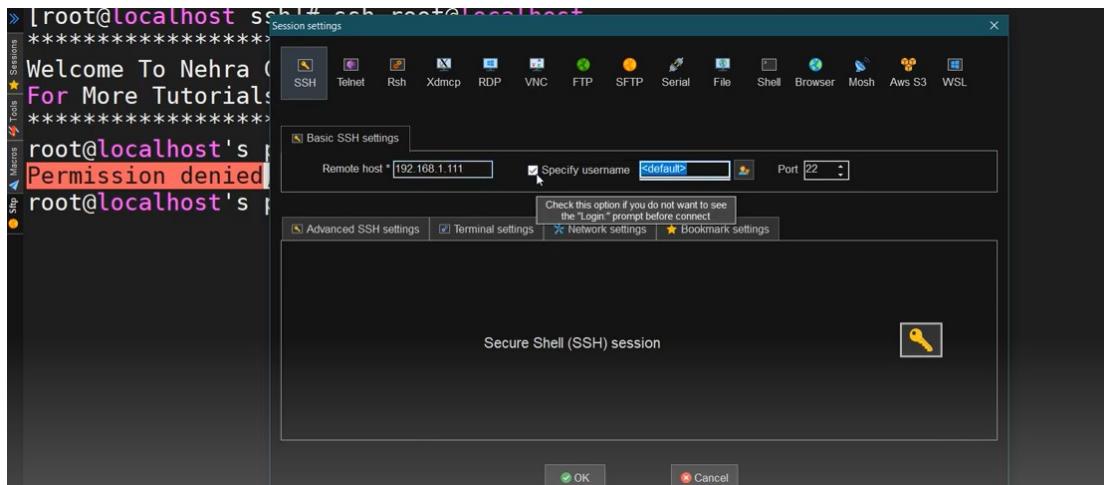
# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys
```

```
[root@localhost ssh]# systemctl restart sshd
[root@localhost ssh]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-11-13 21:00:41 IST; 8s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 3243 (sshd)
    Tasks: 1 (limit: 11160)
   Memory: 1.3M
      CGroup: /system.slice/sshd.service
              └─3243 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@
```

Nov 13 21:00:41 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
Nov 13 21:00:41 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Nov 13 21:00:41 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Nov 13 21:00:41 localhost.localdomain sshd[3243]: Server listening on 0.0.0.0 port 22.
Nov 13 21:00:41 localhost.localdomain sshd[3243]: Server listening on :: port 22.

lines 1-16/16 (END)

```
[root@localhost ssh]# ssh root@localhost
*****
Welcome To Nehra Classes.
For More Tutorials On Linux Please Subscribe Our Youtube Channel.
*****
root@localhost's password:
Permission denied, please try again.
root@localhost's password:
```



```
Pre-authentication banner message from server:
*****
Welcome To Nehra Classes.
For More Tutorials On Linux Please Subscribe Our Youtube Channel.
*****
End of banner message from server
```

```

Pre-authentication banner message from server:
*****
Welcome To Nehra Classes.
For More Tutorials On Linux Please Subscribe Our Youtube Channel.
*****
End of banner message from server
Access denied
root@192.168.1.111's password:
Access denied
root@192.168.1.111's password:

# allows local user Account
[root@localhost ssh]# ssh vikasnehra@localhost
*****
Welcome To Nehra Classes.
For More Tutorials On Linux Please Subscribe Our Youtube Channel.
*****
vikasnehra@localhost's password:
*****
Hi, Welcome To Nehra Classes.
Be Careful as this is a production server.
*****
Web console: https://localhost:9090/

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

Last login: Fri Nov 13 20:39:14 2020 from 192.168.1.107
[vikasnehra@localhost ~]$ 

```

```

From user to Root account ssh
[vikasnehra@localhost ~]$ whoami
vikasnehra
[vikasnehra@localhost ~]$ ssh root@localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:tAWgj27HXBq5VR6QryB4PHROWoleGBJxDbNrW7joagc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
*****
Welcome To Nehra Classes.
For More Tutorials On Linux Please Subscribe Our Youtube Channel.
*****
root@localhost's password:
Permission denied, please try again.
root@localhost's password: 

```

Allow Or Deny Selected Users/Groups To Login Via SSH in Linux || Nehra Classes

```

[root@localhost ~]# vim /etc/ssh/sshd_config

```

End of line

```
# no default banner path
#Banner none

# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASURE
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS

# override default of no subsystems
Subsystem      sftp    /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
AllowUsers vikasnehra tom anuj
```

```
[root@localhost ~]# systemctl restart sshd
[root@localhost ~]# 

[root@localhost ~]# ssh vikasnehra@localhost
vikasnehra@localhost's password:
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://access.redhat.com/insights/
To register this system, run: insights-client --register

Last failed login: Sun Dec 20 22:11:12 IST 2020 from ::1 on
There was 1 failed login attempt since the last successful login.
Last login: Sun Dec 20 22:03:49 2020 from ::1
[vikasnehra@localhost ~]$
```

```
[root@localhost ~]# ssh test1@localhost
test1@localhost's password:
Permission denied, please try again.
test1@localhost's password:
```

```
[root@localhost ~]# groupadd linux
[root@localhost ~]# vim /etc/group
```

```
Mac: ~
$ cat /etc/passwd
rpcuser:x:29:
gdm:x:42:
gnome-initial-setup:x:975:
sshd:x:74:
slocate:x:21:
rngd:x:974:
tcpdump:x:72:
vikasnehra:x:1000:
tom:x:1001:
anuj:x:1002:
john:x:1003:
test1:x:1004:
test2:x:1005:
linux:x:1006:test1,test2
-- INSERT --
```

```
Quick Connect...
[ root@localhost ~]# id test1
uid=1004(test1) gid=1004(test1) groups=1004(test1),1006(linux)
[ root@localhost ~]# id test2
uid=1005(test2) gid=1005(test2) groups=1005(test2),1006(linux)
[ root@localhost ~]#
```

```
Session Servers Tools Games Sessions View Split MultiDEC Partition Packages Settings Help
Quick connect...
[ root@localhost ~]# vim /etc/ssh/sshd_config
```

```

# no default banner path
#Banner none

# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MON
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUR
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS

# override default of no subsystems
Subsystem      sftp      /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
AllowGroups linux
-- INSERT --

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

[root@localhost ~]# systemctl restart sshd
[root@localhost ~]#

```

```

[root@localhost ~]# ssh test1@localhost
test1@localhost's password:
Activate the web console with: systemctl enable --now cockpit

This system is not registered to Red Hat Insights. See https://
To register this system, run: insights-client --register

Last failed login: Sun Dec 20 22:18:20 IST 2020 from ::1 on s
There was 1 failed login attempt since the last successful logi
Last login: Sun Dec 20 22:10:54 2020 from ::1
[test1@localhost ~]$ 

```

```

[test1@localhost ~]$ ssh vikasnehra@localhost
vikasnehra@localhost's password:
Permission denied, please try again.
vikasnehra@localhost's password: 

```

only allow speci user/ grps

```
[root@localhost ~]# vim /etc/ssh/sshd_config
[root@localhost ~]# cl■

# no default banner path
#Banner none
★
# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONI
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUR
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
● AcceptEnv XMODIFIERS

# override default of no subsystems
Subsystem      sftp      /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
DenyUsers test1 tom anuj■
-- INSERT --
```

```
[root@localhost ~]# systemctl restart sshd
[root@localhost ~]# ■
```

```
> [root@localhost ~]# ssh vikasnehra@localhost
vikasnehra@localhost's password:
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://access.redhat.com
To register this system, run: insights-client --register

Last failed login: Sun Dec 20 22:20:37 IST 2020 from ::1 on s
There was 1 failed login attempt since the last successful login.
Last login: Sun Dec 20 22:17:30 2020 from ::1
[vikasnehra@localhost ~]$ ■
```

```
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help  
Quick connect... 2 192.168.1.115 (!)  
» [root@localhost ~]# ssh tom@localhost  
tom@localhost's password:  
Permission denied, please try again.  
tom@localhost's password: █
```

```
# Deny grp  
#Banner none  
  
# Accept locale-related environment variables  
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES  
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT  
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE  
AcceptEnv XMODIFIERS  
  
# override default of no subsystems  
Subsystem sftp /usr/libexec/openssh/sftp-server  
  
# Example of overriding settings on a per-user basis  
#Match User anoncvs  
#     X11Forwarding no  
#     AllowTcpForwarding no  
#     PermitTTY no  
#     ForceCommand cvs server  
DenyGroups linux █  
-- INSERT --  
REGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

```
[root@localhost ~]# vim /etc/ssh/sshd_config  
[root@localhost ~]# systemctl restart sshd  
[root@localhost ~]# █
```

```
[root@localhost ~]# id test1  
uid=1004(test1) gid=1004(test1) groups=1004(test1),1006(linu█  
[root@localhost ~]# id test2  
uid=1005(test2) gid=1005(test2) groups=1005(test2),1006(linu█  
[root@localhost ~]# █
```

```
[root@localhost ~]# ssh test1@localhost
test1@localhost's password:
Permission denied, please try again.
test1@localhost's password:
```

Setup SSH Idle Timeout in Linux | Configure SSH Inactivity Timeout in RHEL (CentOS)

Setup Session Idle Timeout (inactivity timeout) For SSH in Linux:
=====

There are two options related to ssh inactivity in `/etc/ssh/sshd_config` file:

`ClientAliveInterval`
`ClientAliveCountMax`

So the timeout value is calculated by multiplying `ClientAliveInterval` with `ClientAliveCountMax`.

`timeout interval = ClientAliveInterval * ClientAliveCountMax`
The meaning of the two parameters can be found in the man page of `sshd_config`:

man sshd_config
 `ClientAliveCountMax`
 Sets the number of client alive messages (see below) which may be sent
 to the client before the connection is closed.
 `ClientAliveInterval`
 Sets a timeout interval in seconds after which if no data has been
 received from the client, it is closed.

There are 2 methods to configure the inactivity timeout. For example in this

Method 1:

```
File Edit View Search Terminal Help
[root@localhost ~]# vim /etc/ssh/sshd_config
```

It is recommended to use pam_motd in /etc/pam.d/sshd instead of PrintMotd,
as it is more configurable and versatile than the built-in version.
PrintMotd no

#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval
#ClientAliveCountMax
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

/Clien

```

File Edit View Search Terminal Help
# It is recommended to use pam_motd in /etc/pam.d/sshd instead of PrintMotd,
# as it is more configurable and versatile than the built-in version.
PrintMotd no

#printLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
ClientAliveInterval 1m
ClientAliveCountMax 0
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

[root@localhost ~]# systemctl restart sshd
[root@localhost ~]#

```

\$ Putty connection Failed .. fatal error to connect to Server

```

root@localhost:~
login as: root
root@192.168.1.115's password:
Last login: Tue Dec 15 16:50:21 2020 from 192.168.1.114
root@localhost ~]# date
Tue Dec 15 16:59:29 IST 2020
root@localhost ~]#

```



Change SSH Default Port in Linux | OS Hardening By Secure SSH | Nehra Classes

```

root@localhost ~]# vim /etc/ssh/sshd_config

```

```
» # sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
/port
```

```
# port 5152 -tcp/udp
» # The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 5152
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
-- TNSFRT --
```

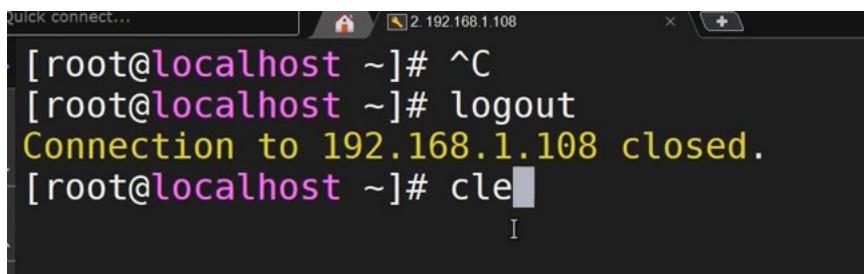
```
» [root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpcv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@localhost ~]#
```

```
Quick connect... 192.168.1.108 x +  
[root@localhost ~]# firewall-cmd --permanent --add-port=5152/tcp  
success  
[root@localhost ~]# firewall-cmd --reload  
success  
[root@localhost ~]# clear  
  
success  
[root@localhost ~]# firewall-cmd --list-all  
public (active)  
  target: default  
  icmp-block-inversion: no  
  interfaces: ens160  
  sources:  
  services: cockpit dhcpcv6-client ssh  
  ports: 5152/tcp  
  protocols:  
  masquerade: no  
  forward-ports:  
  source-ports:  
  icmp-blocks:  
  rich rules:  
  
[root@localhost ~]#
```

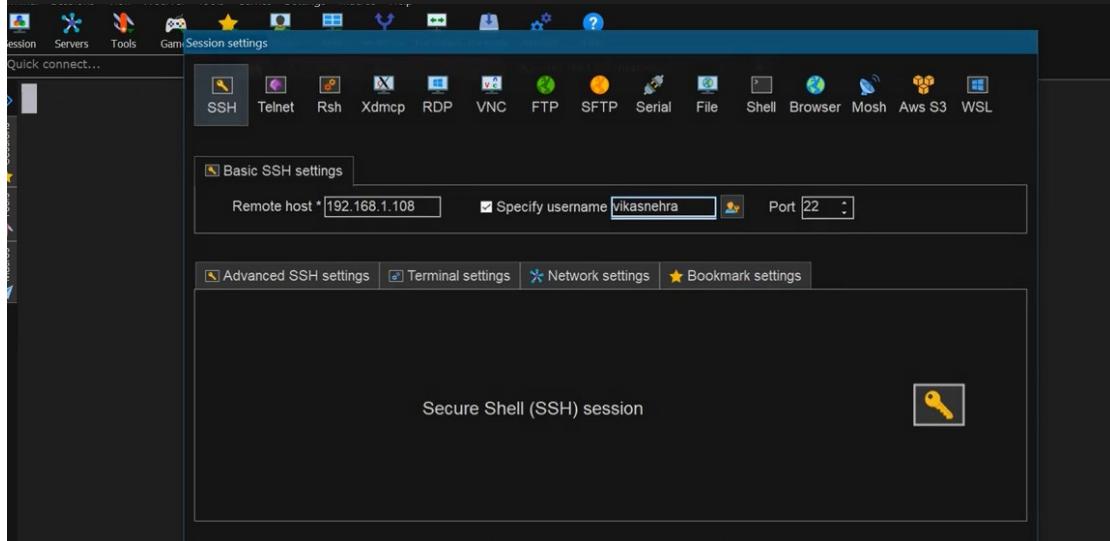
```
Quick connect... 192.168.1.108 x +  
[root@localhost ~]# semanage port -l | grep ssh  
ssh_port_t          tcp    22  
[root@localhost ~]# semanage port -a -t ssh_port_t -p tcp 5152  
[root@localhost ~]# semanage port -l | grep ssh  
ssh_port_t          tcp    5152, 22  
[root@localhost ~]#  
  
root@localhost ~]# systemctl restart sshd  
root@localhost ~]# cl
```

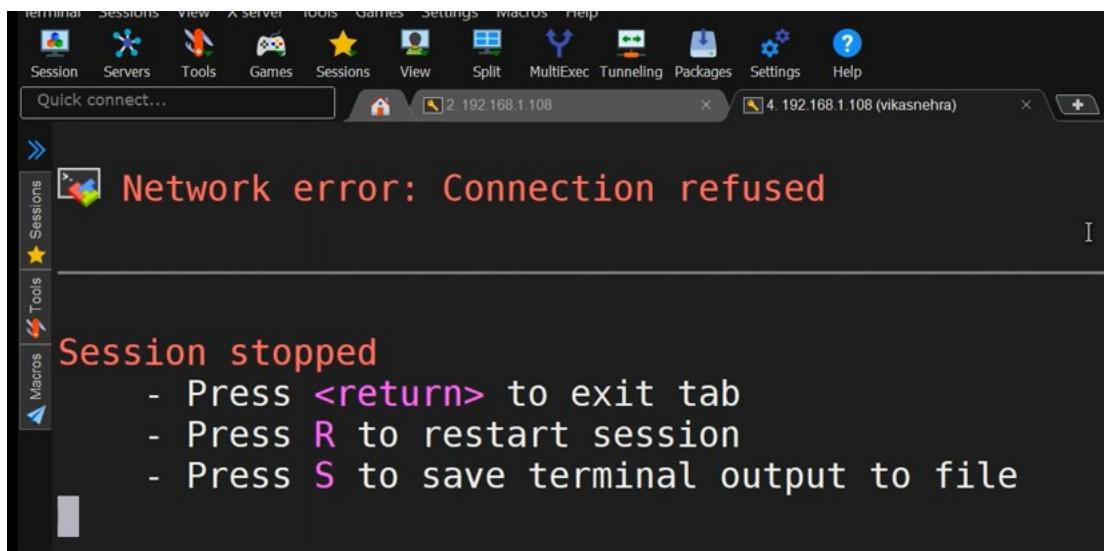
```
» [root@localhost ~]# ssh root@192.168.1.108 -p 5152
The authenticity of host '[192.168.1.108]:5152 ([192.168.1.108]:5152)' can't
be established.
ECDSA key fingerprint is SHA256:ptQP4kwbvdWwu3f+oBtDzcB6pzTx1AqetL4Cw8dmaLE
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.1.108]:5152' (ECDSA) to the list of known hosts.
root@192.168.1.108's password:
Activate the web console with: systemctl enable --now cockpit.socket
This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register
Last login: Thu Oct  8 19:21:59 2020
[root@localhost ~]#
```



```
Quick connect... 2.192.168.1.108 ×
[ root@localhost ~]# ^C
[ root@localhost ~]# logout
Connection to 192.168.1.108 closed.
[ root@localhost ~]# cle
```

```
[root@localhost ~]# ssh root@192.168.1.108
ssh: connect to host 192.168.1.108 port 22: Connection refused
[root@localhost ~]#
```





now port 5152

The terminal window shows an SSH session setup. The host is set to 192.168.1.108, port 5152, and the username is specified as vikasnehra. The session is titled "Secure Shell (SSH) session". A modal dialog box is open, showing the configuration for the SSH session:

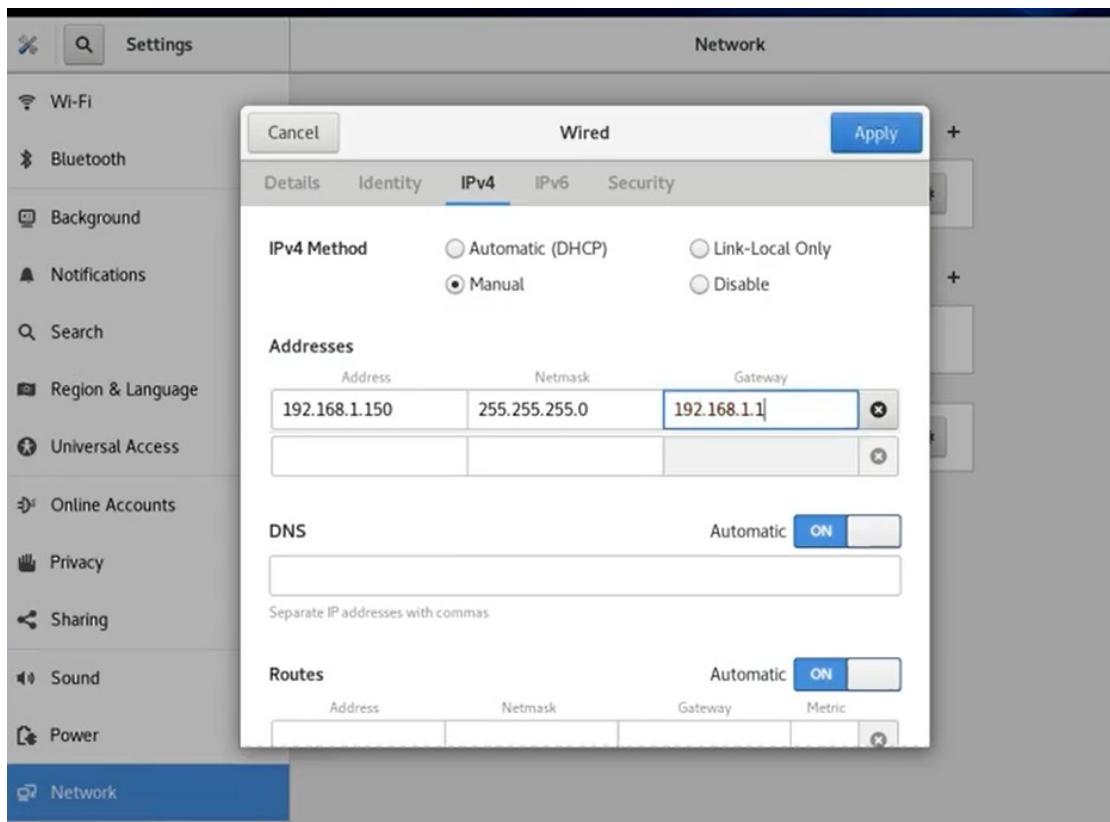
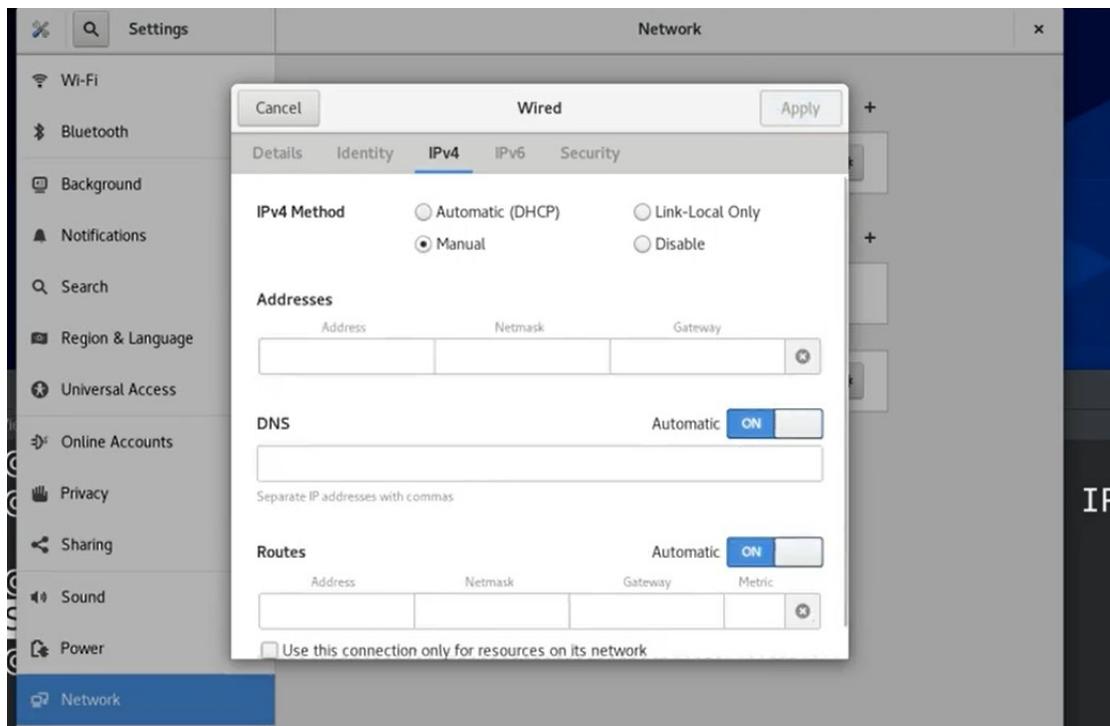
- SSH session to vikasnehra@192.168.1.108
- SSH compression : ✓
- SSH-browser : ✓
- X11-forwarding : ✓ (remote display is forwarded)
- DISPLAY : ✓ (automatically set on remote)

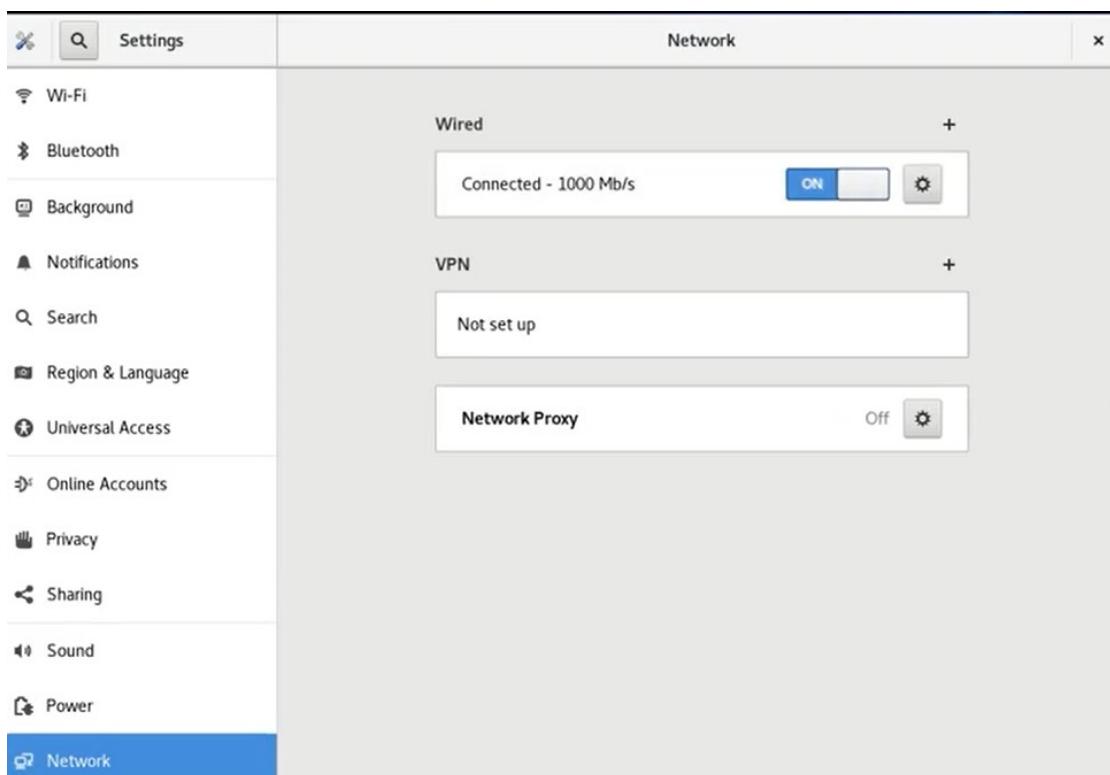
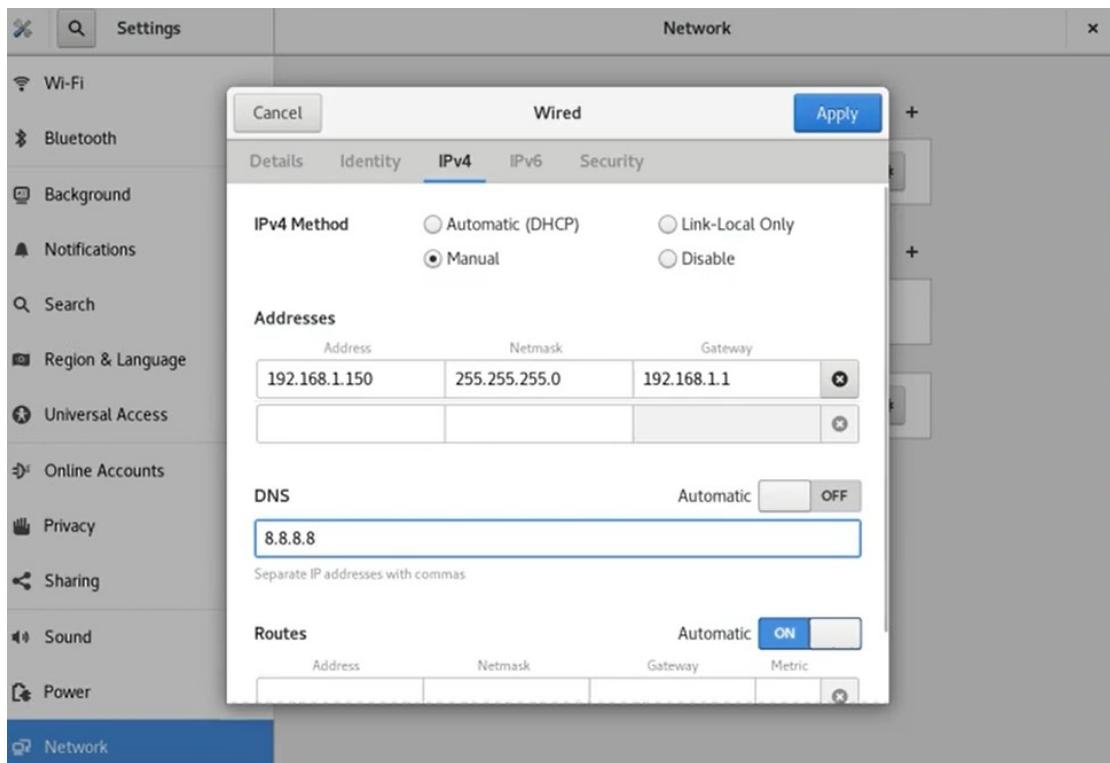
The modal also includes a note: "For more info, ctrl+click on help or visit our website".

The terminal window also displays the following information:

- File browser sidebar showing files like .bash_history, .bash_logout, .bash_profile, .bashrc, .esd_auth, and .Xauthority.
- Message: "Activate the web console with: systemctl enable --now cockpit-webconsole".
- Message: "This system is not registered to Red Hat Insights. See https://access.redhat.com/".
- Message: "To register this system, run: insights-client --register".
- Message: "Last login: Thu Oct 8 19:04:11 2020 from 192.168.1.107".
- Terminal prompt: [vikasnehra@localhost ~]\$

Learn All Methods To Assign The Static IP Address in Linux | NMCLI, NMTUI, GUI ||



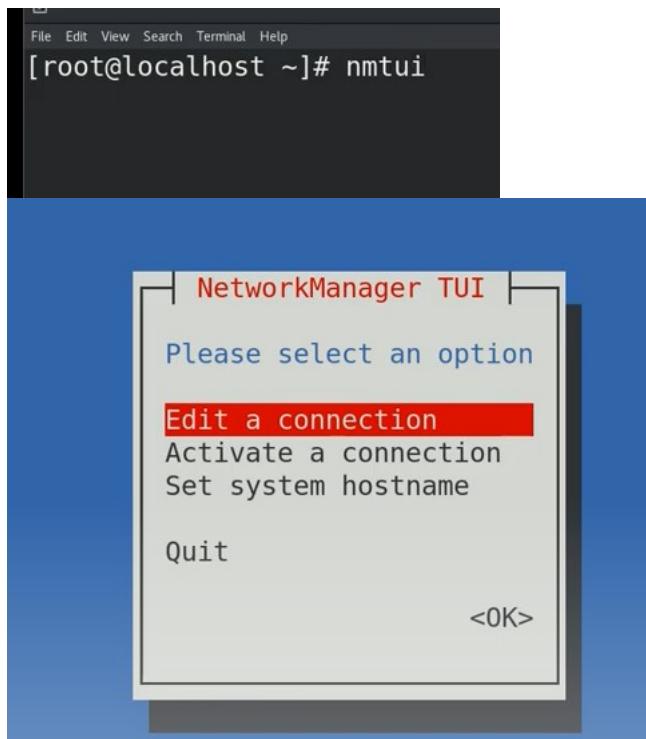


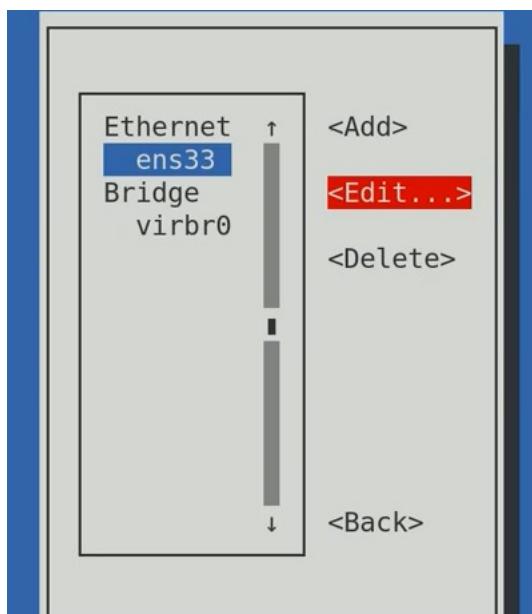
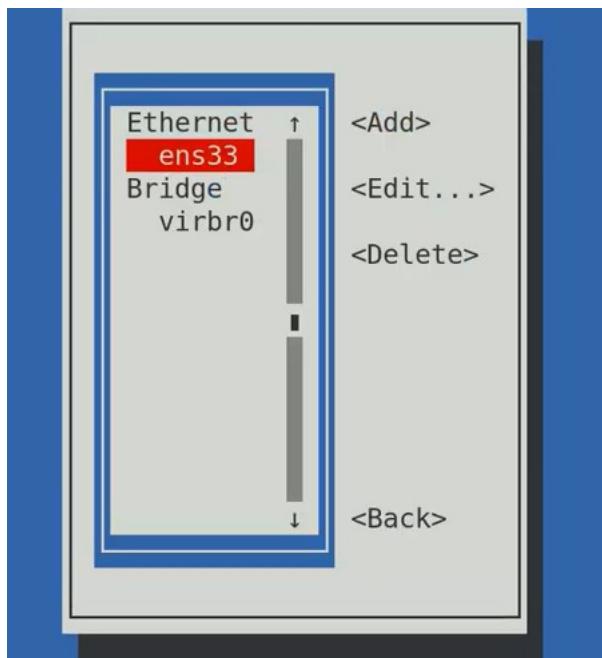
```
[root@localhost ~]# #Different Methods Used To Assign Static IP Address in Linux#
[root@localhost ~]# cat /etc/redhat-release
CentOS Linux release 8.2.2004 (Core)
[root@localhost ~]# clear
[root@localhost ~]# #WELCOME TO NEHRA CLASSES#
[root@localhost ~]# #Different Methods Used To Assign Static IP Address in Linux#
[root@localhost ~]# cat /etc/redhat-release
CentOS Linux release 8.2.2004 (Core)
[root@localhost ~]#
```

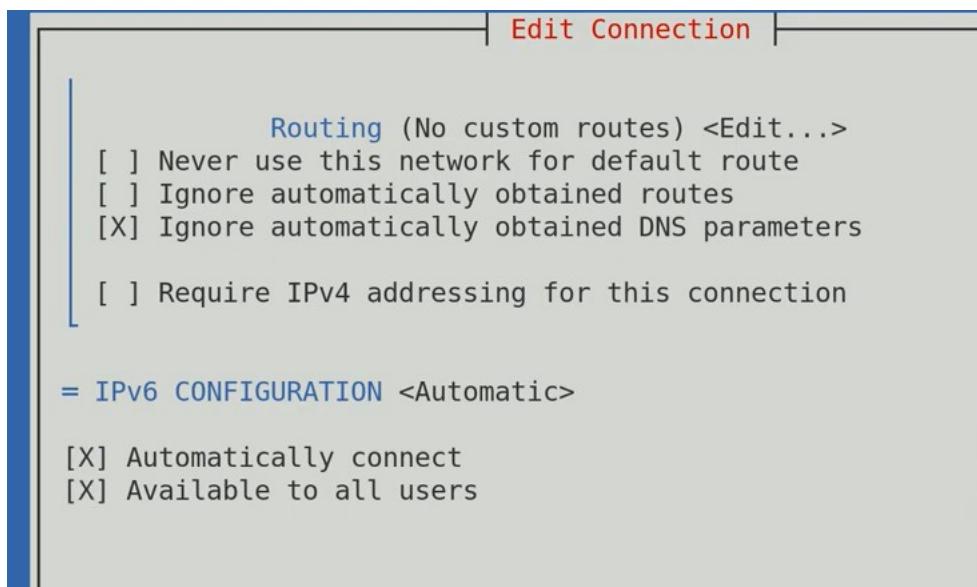
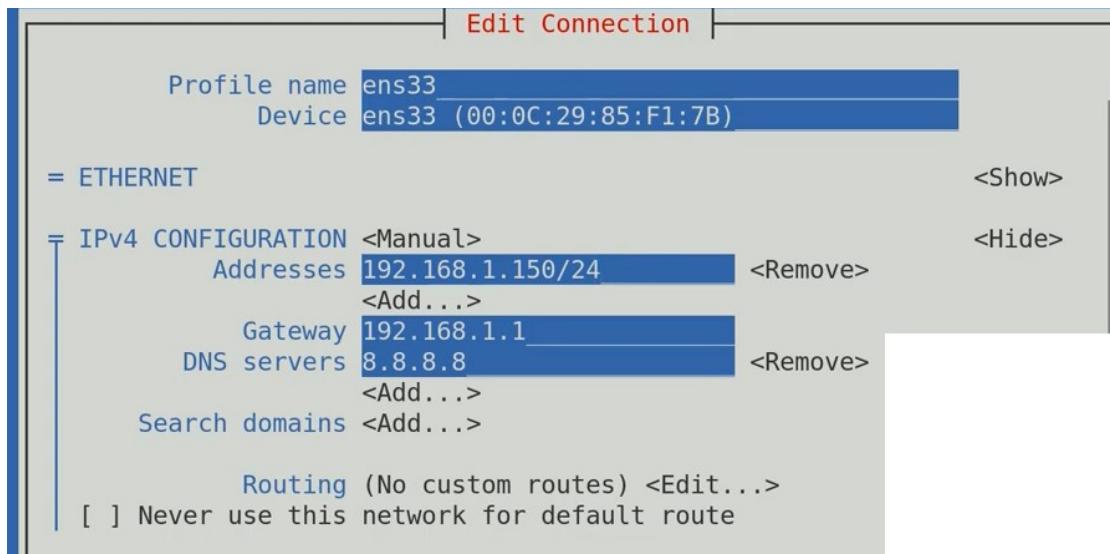
```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.150 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::f3a0:5d96:7902:aa55 prefixlen 64 scopeid 0x20<link>
          ether 00:0c:29:85:f1:7b txqueuelen 1000 (Ethernet)
            RX packets 2104 bytes 245635 (239.8 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 46 bytes 6304 (6.1 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

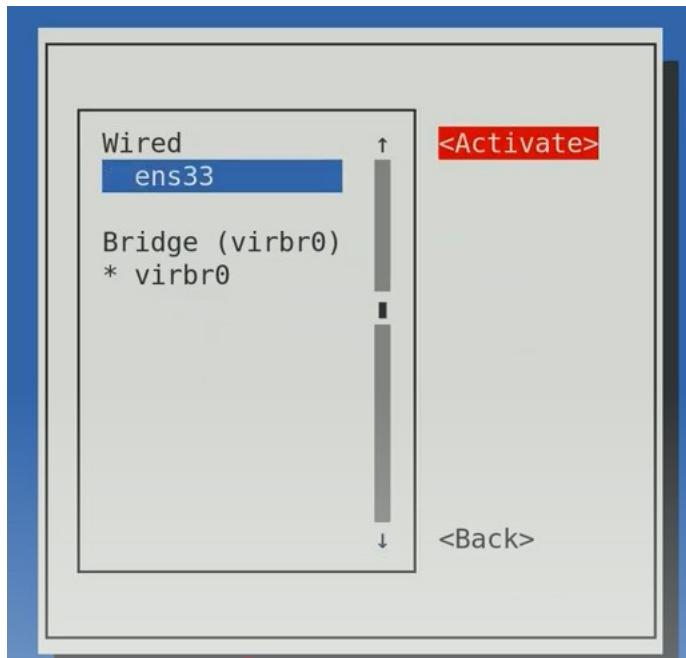
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 48 bytes 4080 (3.9 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 48 bytes 4080 (3.9 KiB)
```

cli -- static ip addr









```
[root@localhost ~]# clear

[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.1.140  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::f3a0:5d96:7902:aa55  prefixlen 64  scopeid 0x20<link>
          ether 00:0c:29:85:f1:7b  txqueuelen 1000  (Ethernet)
            RX packets 3606  bytes 412412 (402.7 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 142  bytes 18694 (18.2 KiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 76  bytes 6464 (6.3 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 76  bytes 6464 (6.3 KiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

```
File Edit View Search Terminal Help
root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-ens33 █

Ip addr / subnet / gateway
```

```
File Edit View Search Terminal Help  
TYPE=Ethernet  
PROXY_METHOD=none  
BROWSER_ONLY=no  
BOOTPROTO=none  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=no  
IPV6INIT=yes  
IPV6_AUTOCONF=yes  
IPV6_DEFROUTE=yes  
IPV6_FAILURE_FATAL=no  
IPV6_ADDR_GEN_MODE=stable-privacy  
NAME=ens33  
UUID=2bf7d0e1-0fa5-4456-bd21-9486e02e2718  
DEVICE=ens33  
ONBOOT=yes  
IPADDR=192.168.1.140 ↵  
PREFIX=24  
GATEWAY=192.168.1.1  
DNS1=8.8.8.8  
PEERDNS=no  
-- INSERT --
```

save the file then Activate

```
[root@localhost ~]# ifdown ens33  
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/1)  
[root@localhost ~]# ifup ens33  
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)  
[root@localhost ~]#
```

Verify

```
File Edit View Search Terminal Help  
[root@localhost ~]# ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
      inet 192.168.1.130 netmask 255.255.255.0 broadcast 192.168.1.255  
        ether 00:0c:29:85:f1:7b txqueuelen 1000 (Ethernet)  
          RX packets 4249 bytes 487606 (476.1 KiB)  
          RX errors 0 dropped 0 overruns 0 frame 0  
          TX packets 230 bytes 29308 (28.6 KiB)  
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
      inet 127.0.0.1 netmask 255.0.0.0  
        inet6 ::1 prefixlen 128 scopeid 0x10<host>  
          loop txqueuelen 1000 (Local Loopback)  
            RX packets 128 bytes 10888 (10.6 KiB)  
            RX errors 0 dropped 0 overruns 0 frame 0  
            TX packets 128 bytes 10888 (10.6 KiB)  
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
      inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255  
        ether 52:54:00:ba:7b:50 txqueuelen 1000 (Ethernet)
```

```

File Edit View Search Terminal Help
[root@localhost ~]# nmcli connection modify ens33 ipv4.addresses 192.168.1.120/24 ipv4.method manual
[root@localhost ~]# ifdown ens33
Connection 'ens33' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)
[root@localhost ~]# ifup ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@localhost ~]# clear

```

How To Assign Static IPv4 & IPv6 in Linux | Set Static IP Address in RHEL 8

	Internet Protocol version 4 (IPv4)	Internet Protocol version 6 (IPv6)
Deployed	1981	1999
Address Size	32-bit number	128-bit number
Address Format	Dotted Decimal Notation: 192.149.252.76	Hexadecimal Notation: 3FFE:F200:0234:AB00: 0123:4567:8901:ABCD
Prefix Notation	192.149.0.0/24	3FFE:F200:0234::/48
Number of Addresses	$2^{32} = \sim 4,294,967,296$	$2^{128} = \sim 340,282,366,$ 920,938,463,463,374, 607,431,768,211,456

Configure ipv4 / ipv6

```

[root@localhost ~]# uname -a
Linux localhost.localdomain 4.18.0-193.el8.x86_64 #1 SMP Fri May 8 10:59:10 UTC 2020 x86
64 x86_64 x86_64 GNU/Linux
[root@localhost ~]# nmcli connection modify ens33 ipv4.addresses 192.168.1.110/24 ipv4.m
ethod manual
[root@localhost ~]# nmcli connection reload
[root@localhost ~]# systemctl restart network-online.target
[root@localhost ~]#

```

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.110 netmask 255.255.255.0 broadcast 192.168.1.255
              inet6 fe80::e28c:3f81:b607:e2f8 prefixlen 64 scopeid 0x20<link>
                ether 00:0c:29:a0:f8:78 txqueuelen 1000 (Ethernet)
                  RX packets 1552 bytes 201313 (196.5 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 137 bytes 15502 (15.1 KiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
              inet6 ::1 prefixlen 128 scopeid 0x10<host>
                loop txqueuelen 1000 (Local Loopback)
                  RX packets 2432 bytes 211256 (206.3 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 2432 bytes 211256 (206.3 KiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

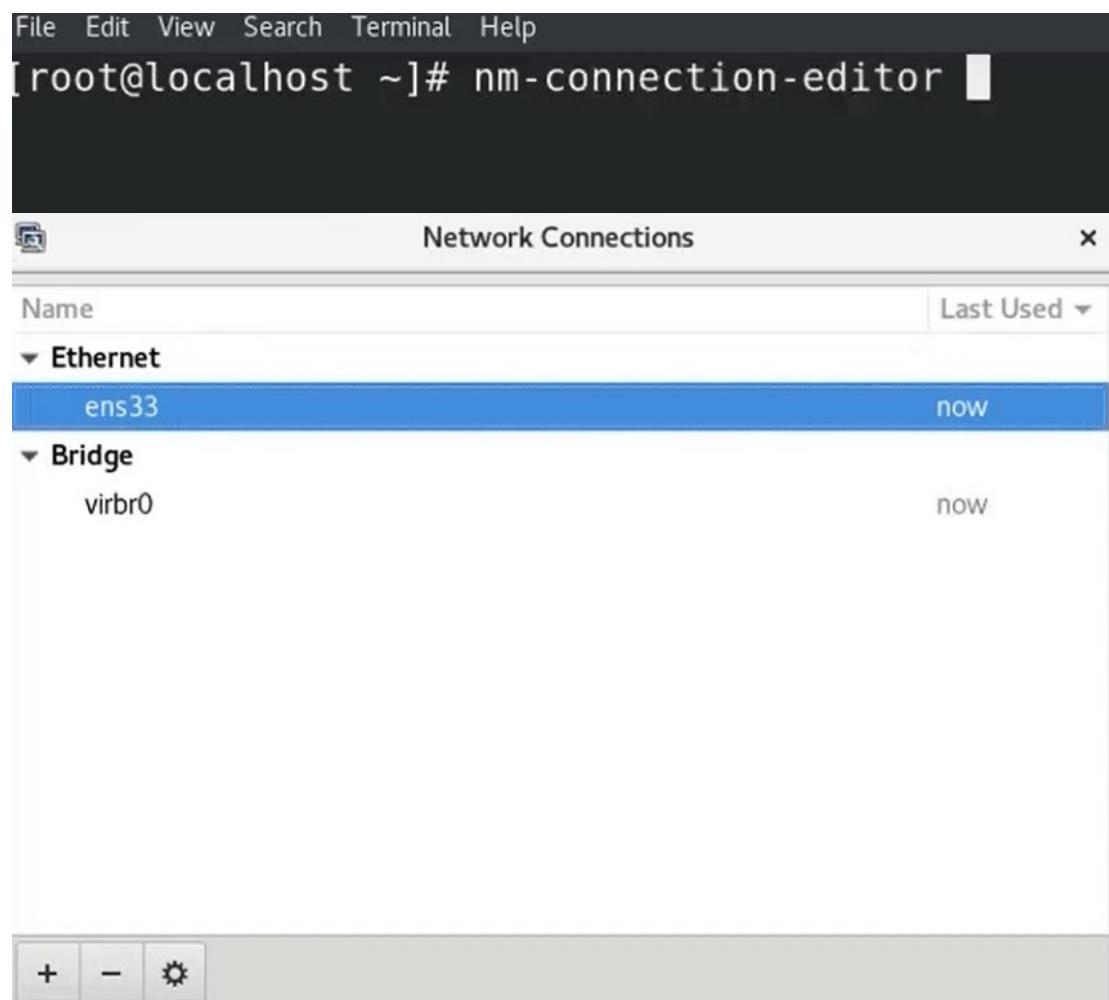
```
vikasnehra@VikasNehra-PC:/mnt/c/Users/vikasnehra
vikasnehra@VikasNehra-PC:/mnt/c/Users/vikasnehra$ ping 192.168.1.110
PING 192.168.1.110 (192.168.1.110) 56(84) bytes of data.
4 bytes from 192.168.1.110: icmp_seq=1 ttl=64 time=1.49 ms
4 bytes from 192.168.1.110: icmp_seq=2 ttl=64 time=0.836 ms
```

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.110 netmask 255.255.255.0 broadcast 192.168.1.255
              inet6 fe80::e28c:3f81:b607:e2f8 prefixlen 64 scopeid 0x20<link>
                ether 00:0c:29:a0:f8:78 txqueuelen 1000 (Ethernet)
                  RX packets 1607 bytes 210653 (205.7 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 152 bytes 16804 (16.4 KiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
              inet6 ::1 prefixlen 128 scopeid 0x10<host>
                loop txqueuelen 1000 (Local Loopback)
                  RX packets 2432 bytes 211256 (206.3 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 2432 bytes 211256 (206.3 KiB)
```

set hostname First

```
[root@localhost ~]# hostnamectl set-hostname CentOS.nehraclasses.local
[root@localhost ~]# hostnamectl
  Static hostname: CentOS.nehraclasses.local
    Icon name: computer-vm
    Chassis: vm
  Machine ID: b2fda1f0531c4727959fa225a5eb4c4b
    Boot ID: fdaa42a1f7c14a74a5a2de94e82842e7
  Virtualization: vmware
Operating System: CentOS Linux 8 (Core)
  CPE OS Name: cpe:/o:centos:centos:8
    Kernel: Linux 4.18.0-193.el8.x86_64
  Architecture: x86-64
[root@localhost ~]#
# now set ipv6
```



```
# Remove ipv4
```

Connection name ens33

General Ethernet 802.1X Security DCB Proxy **IPv4 Settings** IPv6 Settings

Method Manual

Addresses

Address	Netmask	Gateway	Add

DNS servers

Search domains

DHCP client ID

Require IPv4 addressing for this connection to complete

Routes...

select + save changes

Connection name ens33

General Ethernet 802.1X Security DCB Proxy **IPv4 Settings** IPv6 Settings

Method **Automatic (DHCP)**



Additional static addresses

Address	Netmask	Gateway	Add

Additional DNS servers

Additional search domains

DHCP client ID

Require IPv4 addressing for this connection to complete

Routes...

```
[root@localhost ~]# nmcli connection modify ens33 ipv6.addresses 'fdd8:fe2a:ad1e::c0a8:8  
/64' ipv6.method manual
[root@localhost ~]# nmcli connection reload
[root@localhost ~]# systemctl restart network-online.target
[root@localhost ~]# ifup ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
[root@localhost ~]#
```

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.1.105  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fd80:fe2a:adle::c0a8:8  prefixlen 64  scopeid 0x0<global>
          inet6 fe80::e28c:3f81:b607:e2f8  prefixlen 64  scopeid 0x20<link>
            ether 00:0c:29:a0:f8:78  txqueuelen 1000  (Ethernet)
              RX packets 1889  bytes 269270 (262.9 KiB)
              RX errors 0  dropped 0  overruns 0  frame 0
              TX packets 220  bytes 26863 (26.2 KiB)
              TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 2472  bytes 214640 (209.6 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 2472  bytes 214640 (209.6 KiB)
```

```
[root@localhost ~]# ping localhost
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.114 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.078 ms
^C
--- localhost ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 55ms
rtt min/avg/max/mdev = 0.078/0.096/0.114/0.018 ms
[root@localhost ~]# ping fd80:fe2a:adle::c0a8:8
PING fd80:fe2a:adle::c0a8:8(fd80:fe2a:adle::c0a8:8) 56 data bytes
64 bytes from fd80:fe2a:adle::c0a8:8: icmp_seq=1 ttl=64 time=0.187 ms
64 bytes from fd80:fe2a:adle::c0a8:8: icmp_seq=2 ttl=64 time=0.103 ms
```

```
[root@localhost ~]# ping -6 fd80:fe2a:adle::c0a8:8
PING fd80:fe2a:adle::c0a8:8(fd80:fe2a:adle::c0a8:8) 56 data bytes
64 bytes from fd80:fe2a:adle::c0a8:8: icmp_seq=1 ttl=64 time=0.275 ms
64 bytes from fd80:fe2a:adle::c0a8:8: icmp_seq=2 ttl=64 time=0.077 ms
64 bytes from fd80:fe2a:adle::c0a8:8: icmp_seq=3 ttl=64 time=0.097 ms
64 bytes from fd80:fe2a:adle::c0a8:8: icmp_seq=4 ttl=64 time=0.074 ms
^C
--- fd80:fe2a:adle::c0a8:8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 70ms
rtt min/avg/max/mdev = 0.074/0.130/0.275/0.084 ms
[root@localhost ~]# █
```

What is NIC Teaming (Network Bonding) Or Link Aggregation? | Configure NIC Teaming in Linux

Network Bonding (NIC Teaming) / Link Aggregation:

=====
In computer networking, the term link aggregation refers to various methods of combining (aggregating) multiple network connections in parallel in order to increase throughput beyond what a single connection could sustain, and to provide redundancy in case one of the links should fail. A link aggregation group (LAG) is the collection of physical ports combined together.

NIC teaming is the process of combining multiple network cards together for performance, load balancing, and redundancy reasons. Use NIC teaming to group two or more physical NICs into a single logical network device called a bond. NIC teaming is also called as Load Balancing FailOver (LBFO), bandwidth aggregation, traffic failover, and so on.

Benefits of NIC teaming

NIC teaming helps in:

Load balancing – Traffic is automatically load balanced based on destination address between the available physical NICs.

Fault tolerance – If one of the underlying physical NICs is broken or its cable is unplugged, server/target host detects the fault condition and automatically move traffic to another NIC in the bond.

Teaming Terminologies

Before we roll our sleeves, it's crucial to familiarize yourself with the following terminologies:

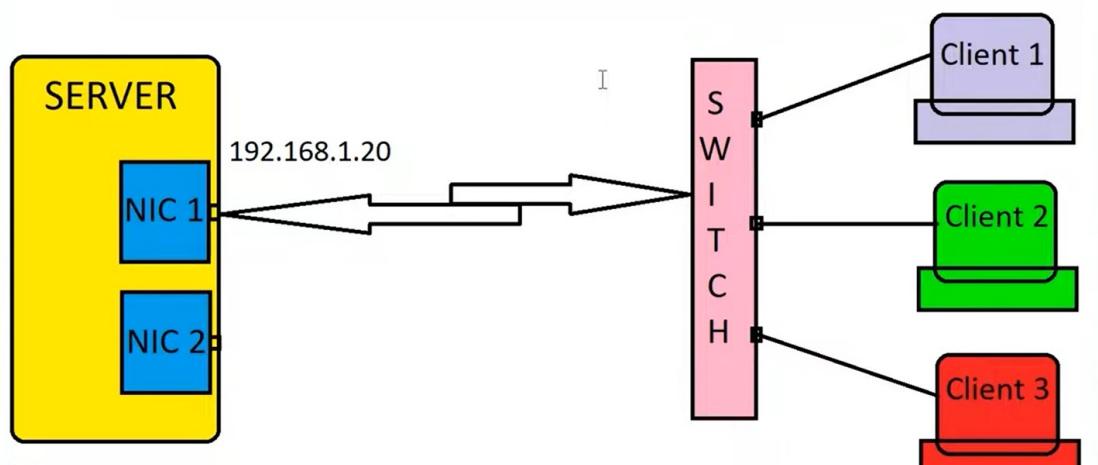
Teamd – This is the nic teaming daemon that uses the libteam library to communicate with team devices via the Linux kernel.

Teamdctl – This is a utility that allows users to control an instance of teamd. You can check and change the port status, as well as switch between backup and active states.

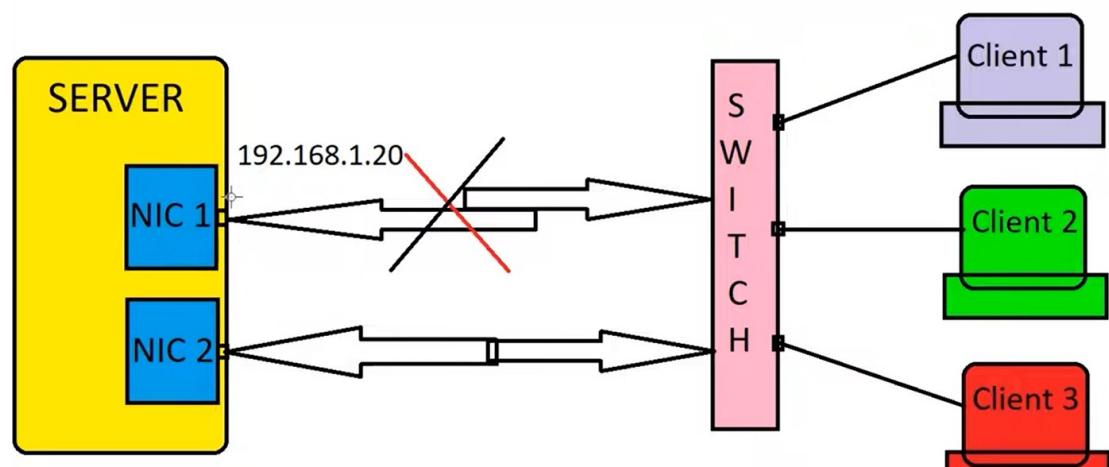
Runner – These are units of code written in JSON and are used for the implementation of various NIC teaming concepts. Examples of runner modes include Round robin, load balancing, broadcast, and active backup.

```
# rpm -qi teamd
```

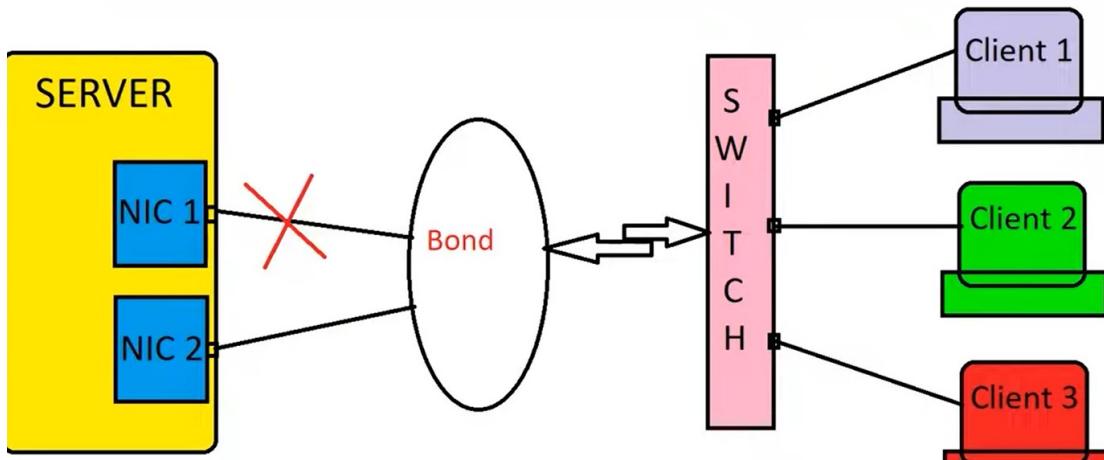
Network Bonding (NIC Teaming) / Link Aggregation



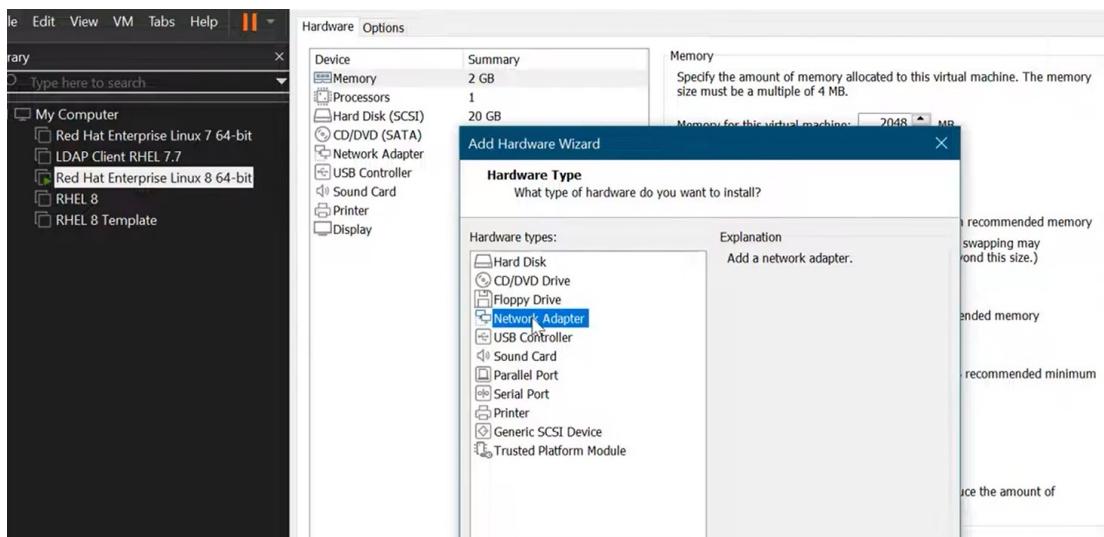
Network Bonding (NIC Teaming) / Link Aggregation

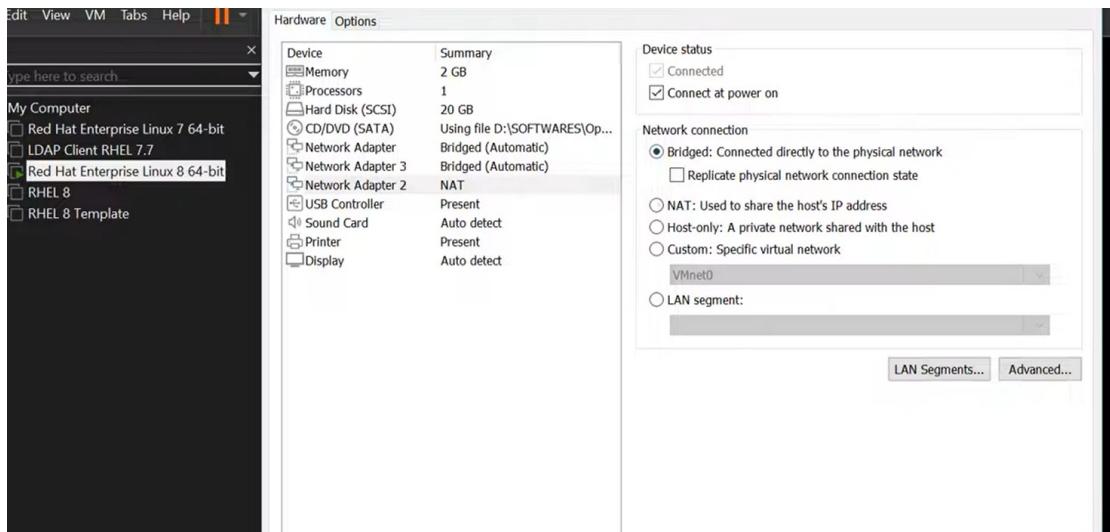


Network Bonding (NIC Teaming) / Link Aggregation



```
[root@primary-dns ~]# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
ens160      ethernet  connected  ens160
virbr0      bridge    connected  virbr0
lo          loopback  unmanaged  --
virbr0-nic  tun       unmanaged  --
[root@primary-dns ~]#
```





```

DEVICE      TYPE      STATE      CONNECTION
ens160      ethernet  connected   ens160
virbr0      bridge    connected   virbr0
ens192      ethernet  disconnected --
ens24       ethernet  disconnected --
lo          loopback unmanaged  --
virbr0-nic  tun       unmanaged  --
[ root@primary-dns ~]#

```

```

[root@primary-dns ~]# nmcli connection show
NAME      UUID
ens160    62b4bfa9-cbce-4326-91bf-3d189fe04486
virbr0    cf3dc632-0217-4564-ba97-fa4c912e4e22
[root@primary-dns ~]#

```

```

> [root@primary-dns ~]# nmcli connection add type team con-name team0 ifname team0 config '{"runner": {"name": "activebackup"} }'
Connection 'team0' (62425d53-6a78-44ab-bf70-53ea48d211dc) successfully added.
[root@primary-dns ~]# nmcli connection show
NAME      UUID
team0    62425d53-6a78-44ab-bf70-53ea48d211dc
ens160    62b4bfa9-cbce-4326-91bf-3d189fe04486
virbr0    cf3dc632-0217-4564-ba97-fa4c912e4e22
[root@primary-dns ~]# nmcli

```

```

> [root@primary-dns ~]# nmcli connection modify team0 ipv4.addresses 192.168.1.120/24 ipv4.gateway 192.168.1.1 connection.autoconnect yes ipv4.method manual
[root@primary-dns ~]#

```

```

» [root@primary-dns ~]# nmcli device status
DEVICE      TYPE      STATE           CONNECTION
ens160      ethernet  connected       ens160
virbr0      bridge    connected       virbr0
team0       team      connecting (getting IP configuration) team0
ens192      ethernet  disconnected   --
ens224      ethernet  disconnected   --
lo          loopback  unmanaged      --
virbr0-nic  tun       unmanaged      --
[root@primary-dns ~]#

```

```

[root@primary-dns ~]# nmcli connection add type team-slave con-name team0-port1 ifname ens192 master team0
Connection 'team0-port1' (7e5be0e7-6920-489f-b4d0-eb213bbac71c) successfully added.
[root@primary-dns ~]# nmcli connection add type team-slave con-name team0-port2 ifname ens224 master team0
Connection 'team0-port2' (2d2666a8-66ce-48e8-acfb-dfcfc23fa74e7) successfully added.
[root@primary-dns ~]#

```

```

[root@primary-dns ~]# nmcli device status
DEVICE      TYPE      STATE           CONNECTION
ens160      ethernet  connected       ens160
team0       team      connected       team0
virbr0      bridge    connected       virbr0
ens192      ethernet  connected       team0-port1
ens224      ethernet  connected       team0-port2
lo          loopback  unmanaged      --
virbr0-nic  tun       unmanaged      --
[root@primary-dns ~]#

```

Team Bonding Working fine

```

Terminal Sessions View X server C:\WINDOWS\system32\cmd.exe - ping 192.168.1.120 -t
Session Servers Tools Games
Quick connect...
» [root@primary-
[root@primary-dns ~]# ping 192.168.1.120 -t
Pinging 192.168.1.120 with 32 bytes of data:
Reply from 192.168.1.120: bytes=32 time<1ms TTL=64
Reply from 192.168.1.120: bytes=32 time<1ms TTL=64
Reply from 192.168.1.120: bytes=32 time<1ms TTL=64
-
```

```

» [root@primary-dns ~]# nmcli connection down team0-port1
Connection 'team0-port1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@primary-dns ~]#

```

Connectivity still working

```
[root@primary-dns ~]# nmcli connection down team0-port1
Connection 'team0-port1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@primary-dns ~]# nmcli connection down team0-port2
```

#no connection as both network down

```
# check details
```

```
Quick connect...  2.192.168.1.115 (root) X +  
» [root@primary-dns ~]# teamdctl team0 state  
setup:  
    runner: activebackup  
runner:  
    active port:  
[root@primary-dns ~]#
```

```
# bringConnection up

    active port:
[root@primary-dns ~]# nmcli connection up team0-port1
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)
[root@primary-dns ~]# nmcli connection up team0-port2
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
[root@primary-dns ~]# clear
```

```
[root@primary-dns ~]# teamdctl team0 state
setup:
  runner: activebackup
ports:
  ens192
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
  ens224
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
runner:
  active port: ens192
[root@primary-dns ~]#
```

Now Connectivity

```
Request timed out.
Request timed out.
Reply from 192.168.1.120: bytes=32 time<1ms TTL=64
```

make one port down & check the connection

```
[root@primary-dns ~]# nmcli connection down team0-port2
Connection 'team0-port2' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
[root@primary-dns ~]#
```

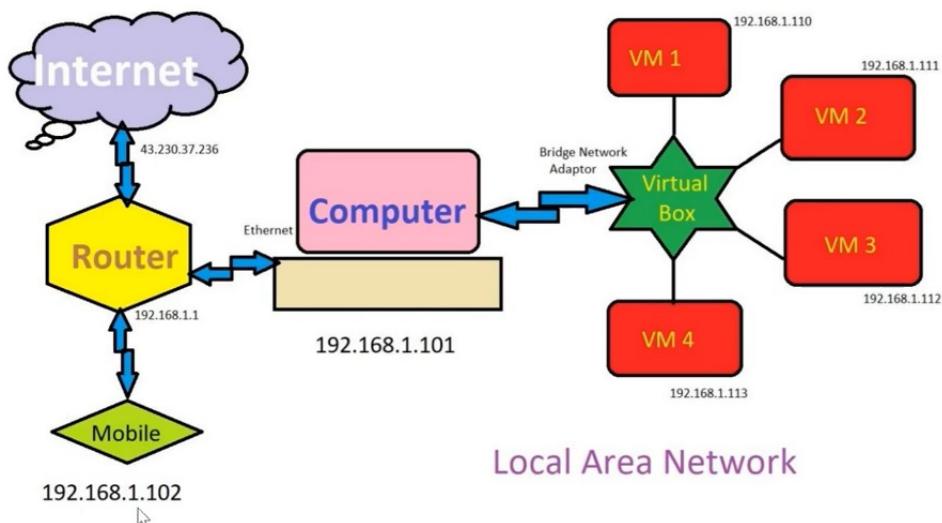
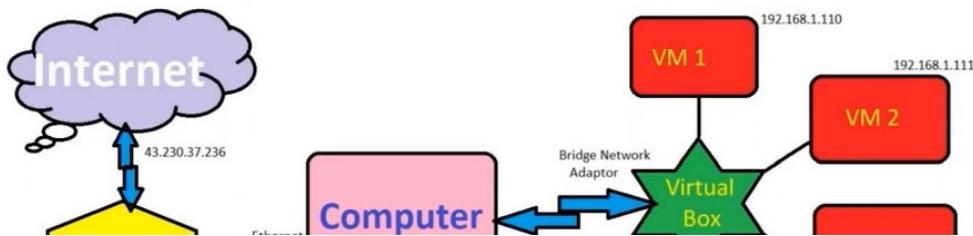
REGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```
[root@primary-dns ~]# teamdctl team0 state
setup:
  runner: activebackup
ports:
  ens192
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
runner:
  active port: ens192
[root@primary-dns ~]#
```

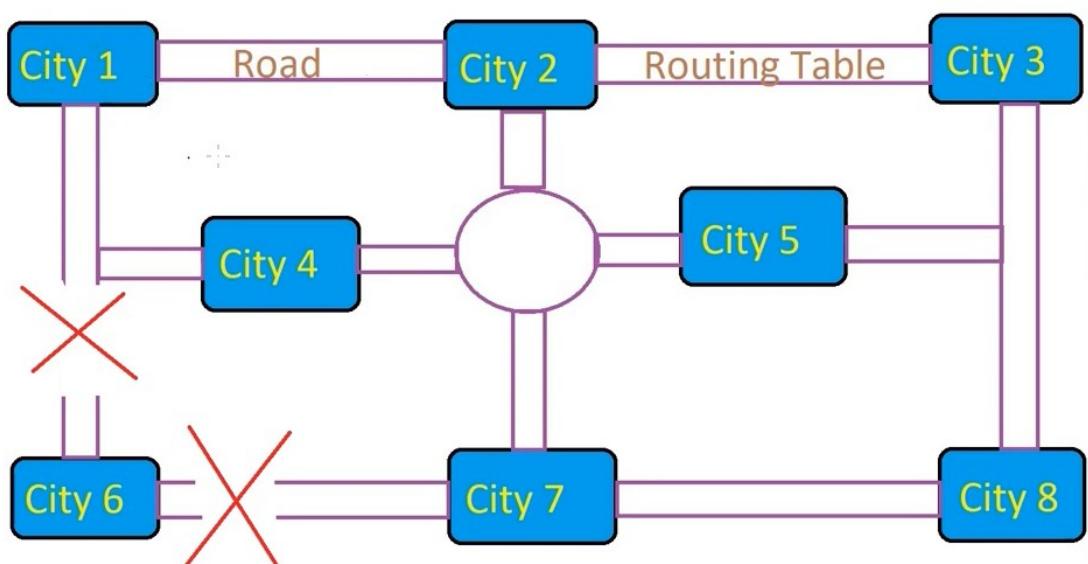
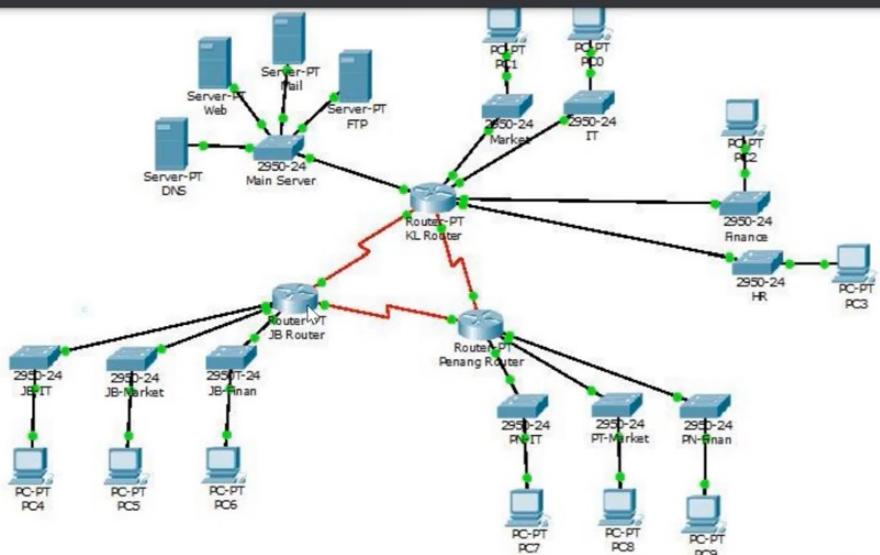
All About Routing & Routing Table | Routing Table Configuration in Linux

ROUTING:

Routing is the process of selecting a path for traffic in a network or between or across multiple networks. Broadly, routing is performed in many types of networks, including circuit-switched networks, such as the public switched telephone network (PSTN), and computer networks, such as the Internet.



Routing is a process which is performed by layer 3 (or network layer) devices in order to deliver the packet by choosing an optimal path from one network to another.



Types Of Routing

Static
Routing

Dynamic
Routing

Default
Routing

There are 3 types of routing:

1. Static routing:

Static routing is a process in which we manually add the routes in routing table.

Advantages

- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.

Disadvantage

- For a large network, it is a hectic task for administrator to manually add each route for the network in the routing table on each router.
- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.

2. Default Routing:

This is the method where the router is configured to send all packets towards a single router (next hop). It doesn't matter to which network the packet belongs, it is forwarded out to router which is configured for default routing. It is generally used with stub routers. A stub router is a router which has only one route to reach all other networks.

3. Dynamic Routing:

Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and the routes to

reach it. RIP and OSPF are the best examples of dynamic routing protocol. Automatic adjustment will be made to reach the network destination if one route goes down.

A dynamic protocol has following features:

1. The routers should have the same dynamic protocol running in order to exchange routes.
2. When a router finds a change in the topology then router advertises it to all other routers.

Advantages:

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage:

- Consumes more bandwidth for communicating with other neighbours.
- Less secure than static routing.

In computer networking a **routing table**, or **routing information base (RIB)**, is a data table stored in a router or a network host that lists the routes to particular network destinations, and in some cases, metrics (distances) associated with those routes. The routing table contains information about the topology of the network immediately around it.

The routing table consists of at least three information fields:

The routing table consists of at least three information fields:

1. **network ID:** The destination subnet
2. **metric:** The routing metric of the path through which the packet is to be sent. The route will go in the direction of the gateway with the lowest metric.
3. **next hop:** The next hop, or gateway, is the address of the next station to which the packet is to be sent on the way to its final destination

Depending on the application and implementation, it can also contain additional values that refine path selection:

1. *quality of service* associated with the route. For example, the U flag indicates that an IP route is up.
2. *filtering criteria*: Access-control lists associated with the route
3. *interface*: Such as eth0 for the first Ethernet card, eth1 for the second Ethernet card, etc.

Shown below is an example of what the table above could look like on an average computer connected to the internet via a home router:

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10

The column **Network Destination** and **Netmask** together describe the **Network ID** as mentioned earlier. For example, destination **192.168.0.0** and netmask **255.255.255.0** can be written as network ID **192.168.0.0/24**.

The **Gateway** column contains the same information as the **Next hop**, i.e. it points to the gateway through which the network can be reached.

- The **Interface** indicates what locally available interface is responsible for reaching the gateway. In this example, gateway **192.168.0.1** (the internet router) can be reached through the local network card with address **192.168.0.100**.
- Finally, the **Metric** indicates the associated cost of using the indicated route. This is useful for determining the efficiency of a certain route from two points in a network. In this example, it is more efficient to communicate with the computer itself through the use of address **127.0.0.1** (called “localhost”) than it would be through **192.168.0.100** (the IP address of the local network card).

```
> [root@localhost ~]# ip route
default via 192.168.1.1 dev ens160 proto dhcp metric 100
192.168.1.0/24 dev ens160 proto kernel scope link src 192.168.1.109 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
[root@localhost ~]# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
0.0.0.0         192.168.1.1    0.0.0.0       UG        0 0          0 ens160
192.168.1.0     0.0.0.0        255.255.255.0  U        0 0          0 ens160
192.168.122.0   0.0.0.0        255.255.255.0  U        0 0          0 virbr0
[root@localhost ~]#
```

```
[root@centos7 ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref Use Iface
0.0.0.0         192.168.1.254  0.0.0.0       UG    100    0    0 eno16777736
192.168.1.0     0.0.0.0        255.255.255.0  U    100    0    0 eno16777736
192.168.122.0   0.0.0.0        255.255.255.0  U    0      0    0 virbr0
```

Adding a static route using IP command

Suppose you want to take a backup of a Linux machine and push the backup file to another backup server in the subnet **10.0.2.0/24**. However for one reason or the other you cannot reach the backup server.

```
> [root@localhost ~]# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref Use Iface
default         _gateway       0.0.0.0       UG    100    0    0 ens160
192.168.1.0     0.0.0.0        255.255.255.0  U    100    0    0 ens160
192.168.122.0   0.0.0.0        255.255.255.0  U    0      0    0 virbr0
[root@localhost ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref Use Iface
0.0.0.0         192.168.1.1    0.0.0.0       UG    100    0    0 ens160
192.168.1.0     0.0.0.0        255.255.255.0  U    100    0    0 ens160
192.168.122.0   0.0.0.0        255.255.255.0  U    0      0    0 virbr0
[root@localhost ~]#
```

```
» [root@localhost ~]# cat /proc/net/route
Iface Destination Gateway Flags RefCnt Use Metric Mask MTU W
inow IRTT
ens160 00000000 0101A8C0 0003 0 0 100 00000000 0 0
0 ens160 0001A8C0 00000000 0001 0 0 100 00FFFFFF 0 0
0 virbr0 007AA8C0 00000000 0001 0 0 0 00FFFFFF 0 0
0
[root@localhost ~]#
```

```
[root@localhost ~]# cat /proc/net/rt_cache
Iface Destination Gateway Flags RefCnt Use Metric Source M
TU Window IRTT TOS HHRef HHUptod SpecDst
[root@localhost ~]# clear
```

```
[root@localhost ~]# cat /proc/net/ipv6_route
00000000000000000000000000000001 80 00000000000000000000000000000000 00 00000000000000000000000000000000
00000000 00000100 00000001 00000000 00000001 lo
fe800000000000000000000000000000 40 00000000000000000000000000000000 00 00000000000000000000000000000000
00000000 00000064 00000001 00000000 00000001 ens160
00000000000000000000000000000000 00 00000000000000000000000000000000 00 00000000000000000000000000000000
00000000 ffffffff 00000001 00000000 00200200 lo
00000000000000000000000000000001 80 00000000000000000000000000000000 00 00000000000000000000000000000000
00000000 00000003 00000000 80200001 lo
fe800000000000888c148e41e5b3f3 80 00000000000000000000000000000000 00 00000000000000000000000000000000
00000000 00000002 00000000 80200001 ens160
ff000000000000000000000000000000 08 00000000000000000000000000000000 00 00000000000000000000000000000000
00000000 00000100 00000003 00000000 00000001 ens160
00000000000000000000000000000000 00 00000000000000000000000000000000 00 00000000000000000000000000000000
00000000 ffffffff 00000001 00000000 00200200 lo
[root@localhost ~]#
```

Adding a static route using IP command

Suppose you want to take a backup of a Linux machine and push the backup file to another backup server in the subnet **10.0.2.0/24**. However, for one reason or the other, you cannot reach the backup server via the default gateway. In this case, you will have to create a

new route for backup server subnet via another IP,
say **192.168.43.223** via the interface **enp0s3**.

The command for this will be

```
# ip route add 10.0.2.0/24 via 192.168.43.223 dev enp0s3
```

```
[root@localhost ~]# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
default         _gateway       0.0.0.0        UG    100    0        0 ens160
192.168.1.0    0.0.0.0        255.255.255.0   U     100    0        0 ens160
192.168.122.0  0.0.0.0        255.255.255.0   U     0      0        0 virbr0
[root@localhost ~]# ip route add 10.0.2.0/24 via 192.168.1.1 dev ens160
[root@localhost ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1   0.0.0.0        UG    100    0        0 ens160
10.0.2.0        192.168.1.1   255.255.255.0   UG    0      0        0 ens160
192.168.1.0    0.0.0.0        255.255.255.0   U     100    0        0 ens160
192.168.122.0  0.0.0.0        255.255.255.0   U     0      0        0 virbr0
[root@localhost ~]#
```

Permanently adding static route (Ubuntu / Debian)

For Debian distributions, edit the file `/etc/network/interfaces`

```
#!/bin /etc/network/interfaces
```

Append the following line:

```
up route add -net 10.0.2.0 netmask 255.255.255.0 gw 192.168.43.1  
dev enp0s3
```

```
[root@localhost ~]# ip route del 10.0.2.0/24 via 192.168.1.1 dev ens160
[root@localhost ~]# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.1.1    0.0.0.0        UG    100    0        0 ens160
192.168.1.0      0.0.0.0        255.255.255.0  U     100    0        0 ens160
192.168.122.0    0.0.0.0        255.255.255.0  U     0      0        0 virbr0
[root@localhost ~]#
```

```
# route del default
```

To add a default route run below ‘ip route add’ command,

```
$ sudo ip route add default via <ip-address> dev interface
```

For example:

```
# ip route add default via 192.168.43.1 dev eth0
```

We hope that this tutorial was informative and provided you with insights into how you can go about adding and deleting static route in Linux.