ComputerNetworkingNotes

How to Configure and use Linux as a Router

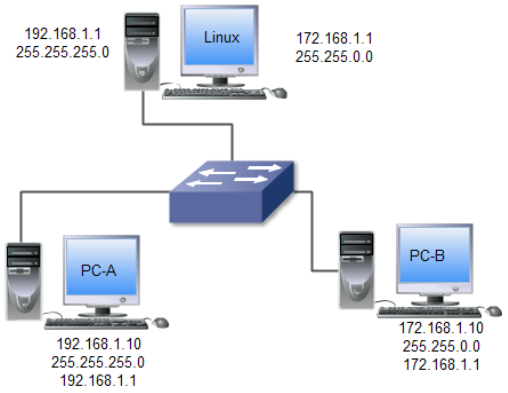
On an IP network, two computers can communicate only if they belong to the same IP subnet. If two computers belong to different IP subnets, they need a router to connect. A router is a special device that not only provides IP forwarding as the main function but also supports many other IP-based features such as; packet filtering, voice over IP, IP firewall, etc.

A router is an expensive device. Configuring it is also a complex task. If you have a Linux system and need only IP forwarding, you can use it. Linux provides a zero-cost solution for IP forwarding.

LAB setup

We need a Linux system to configure the IP forwarding. We need two more systems for testing. We will configure these systems in different IP subnets. We will use the Linux system to provide connectivity between both IP subnets.

The following image shows this setup.



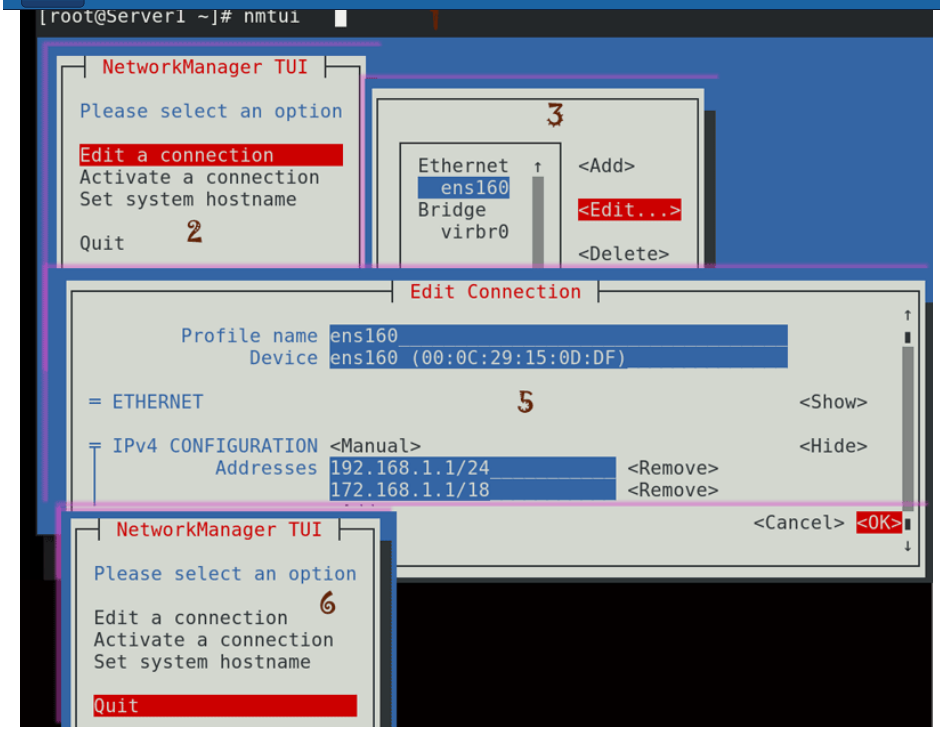
On the Linux system, we need to configure two IP addresses. If you have two interfaces, you can configure one IP address on each. If you have only one interface, you can configure both IP addresses on it. Linux allows you to configure and use multiple IP addresses on the same interface.

To configure both IP addresses on the same interface, use the following steps.

- Start the **nmtui** utility
- Select **Edit a connection** on the main screen
- Select the interface on the left pane and use the **Edit** option to open its configuration
- Select the **Manual** method on the **IPv4 configuration** section
- Assign two IP addresses **192.168.1.1/24** and **172.168.1.1/18**. To assign the second IP address, use the **Add** button.
- Save the configuration and quit the nmtui utility

The following image shows the above process.

ComputerNetworkingNotes



To learn more about the above process, you can check the following tutorial.

How to configure multiple IP addresses on Linux

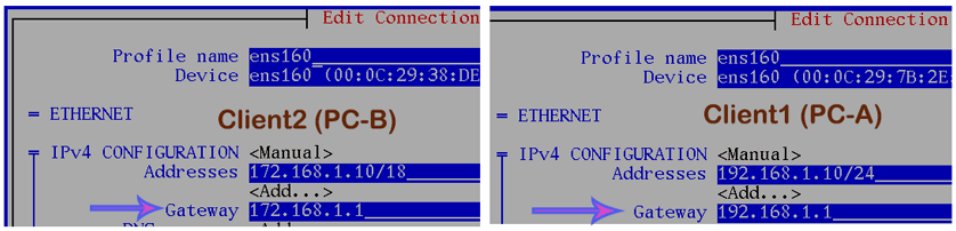
To check and verify the new IP configuration, you can use the **'ip addr show [*interface*]'** command.

```
[root@Server1 ~]# ip addr show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 q
oup default qlen 1000
    link/ether 00:0c:29:15:0d:df brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global
       valid_lft forever preferred_lft forever
    inet 172.168.1.1/18 brd 172.168.63.255 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe15:ddf/64 scope link noprefi
       valid_lft forever preferred_lft forever
[root@Server1 ~]#
```

Use the same process, and configure IP addresses on both systems that you want to use for testing. On these systems, you also need to add the gateway IP address. Add the Gateway IP address from the same subnet.

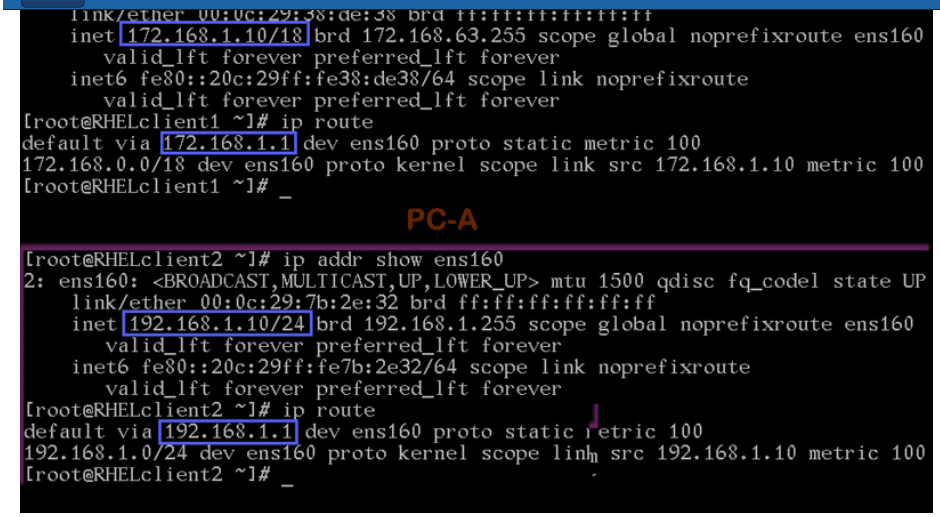
On PC-A, assign an IP address (192.168.1.10/24) from the network 192.168.1.0/24 and configure the gateway to 192.168.1.1.

On PC-B, assign an IP address (172.168.1.10/18) from the network 172.168.1.0/18 and configure the gateway to 172.168.1.1.



After adding the IP configuration on both systems, verify the IP address and Gateway IP address.

ComputerNetworkingNotes



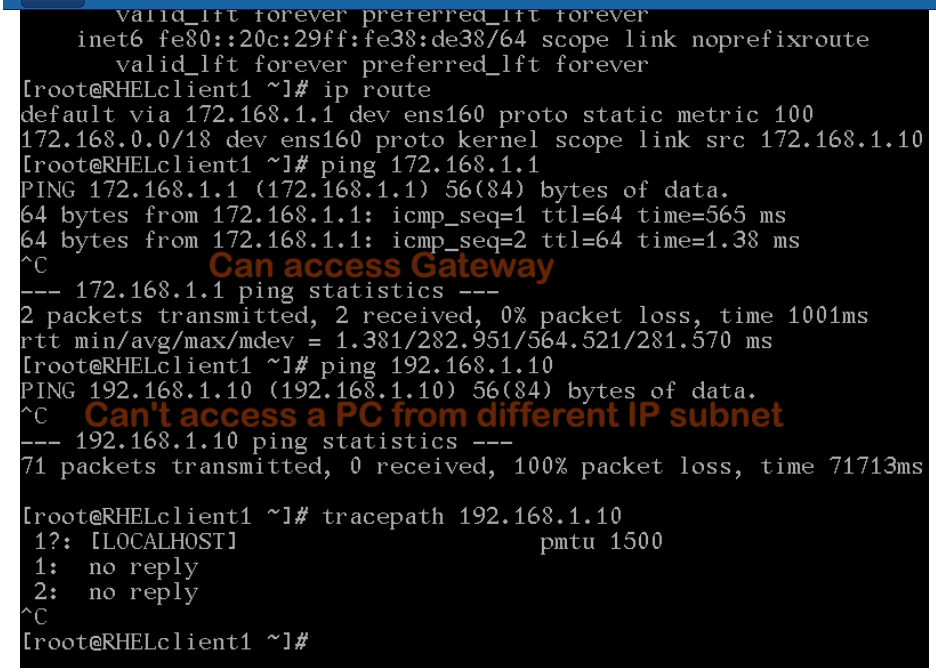
Testing the LAB setup

To test this lab setup, perform the following steps.

Send ping requests from PCA (192.168.1.10) to gateway IP (192.168.1.1/24) configured on the Linux system. You should get replies from the gateway IP. After it, send ping requests to PCB (172.168.1.10/18). Since we have not configured the IP forwarding on the Linux system, you should not get replies from PCB. You can also use the **tracepath** command to view the path packets take to reach PCB.

Perform the same testing from PCB.

ComputerNetworkingNotes



Enabling the IP forwarding

To configure Linux as a router, we need to enable IP forwarding. To enable and disable IP forwarding, Linux uses two configuration values **0** and **1**. It uses the value **0** to disable and the value **1** to enable IP forwarding.

You can enable or disable IP forwarding in two ways temporary and permanent.

Temporary enabling/disabling the IP forwarding

Linux keeps all running processes in the */proc* directory. The */proc* directory represents the current state of the kernel. It allows applications and users to view currently running processes and update their settings on the live system.

Linux controls the IP forwarding through the */proc/sys/net/ipv4/ip_forward* process. There are two ways to change the value of this process. You can use the **echo** command to directly update this value. Or if you don't want to directly update this value, you can use the **sysctl** command to change this value.

Let's understand both methods.

Using the echo command

The following command displays the current state of the IP forwarding.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
[root@Server1 ~]# cat /proc/sys/net/ipv4/ip_forward
1
[root@Server1 ~]# echo 0 > /proc/sys/net/ipv4/ip_forward
[root@Server1 ~]# cat /proc/sys/net/ipv4/ip_forward
0
[root@Server1 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@Server1 ~]# cat /proc/sys/net/ipv4/ip_forward
1
[root@Server1 ~]#
```

Using the sysctl command

The following command enables IP forwarding.

```
#sysctl -w net.ipv4.ip_forward=1
```

The following command disables the IP forwarding.

```
#sysctl -w net.ipv4.ip_forward=0
```

```
[root@Server1 ~]# cat /proc/sys/net/ipv4/ip_forward
1
[root@Server1 ~]# sysctl -w net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
[root@Server1 ~]# cat /proc/sys/net/ipv4/ip_forward
0
[root@Server1 ~]# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[root@Server1 ~]# cat /proc/sys/net/ipv4/ip_forward
1
[root@Server1 ~]#
```

Permanently enabling/disabling the IP forwarding

Linux uses the */etc/sysctl.conf* file to enable or disable the IP forwarding at the boot time.

```
[root@Server1 ~]# cat /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
[root@Server1 ~]#
```

To permanently enable the IP forwarding, add the following lines at end of the file.

```
#Permnanenlty enabling the IP forwarding
net.ipv4.ip_forward = 1
```

To permanently disable the IP forwarding, add the following lines at end of the file.

```
#Permnanenlty disabling the IP forwarding
net.ipv4.ip_forward = 0
```

ComputerNetworkingNotes

```
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).

#Permanently eabling IP Forwarding
net.ipv4.ip_forward = 1
_

:wq
[root@Server1 ~]# sysctl -p /etc/sysctl.conf
net.ipv4.ip_forward = 1
[root@Server1 ~]#
```

Testing the IP forwarding

To test the IP forwarding, enable the IP forwarding on the Linux system and send ping requests from PCA to PCB again. This time, PCA should get replies from PCB.

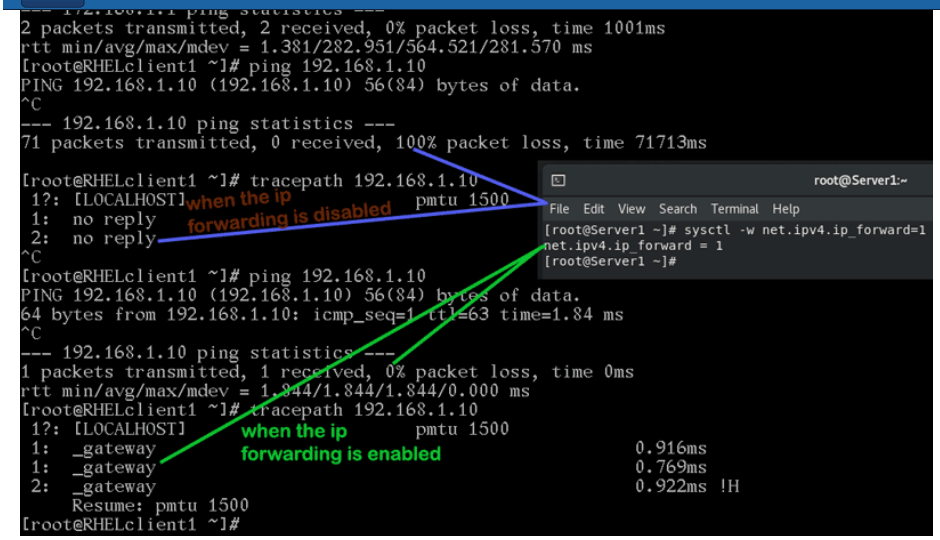
```
[root@RHELclient2 ~]# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=197 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 1 received, 50% packet loss, time 1069ms
rtt min/avg/max/mdev = 197.050/197.050/197.050/0.000 ms
[root@RHELclient2 ~]# ping 172.168.1.10
PING 172.168.1.10 (172.168.1.10) 56(84) bytes of data.
^C
--- 172.168.1.10 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9428ms

[root@RHELclient2 ~]# tracepath 172.168.1.10
 1?: [LOCALHOST]                      pmtu 1500
 1:  no reply
 2:  no reply
^C
[root@RHELclient2 ~]# ping 172.168.1.10
PING 172.168.1.10 (172.168.1.10) 56(84) bytes of data.
64 bytes from 172.168.1.10: icmp_seq=1 ttl=63 time=214 ms
From 192.168.1.1 icmp_seq=2 Redirect Host(New nexthop: 172.168.1.10)
64 bytes from 172.168.1.10: icmp_seq=2 ttl=64 time=24.9 ms
^C
--- 172.168.1.10 ping statistics ---
2 packets transmitted, 2 received, +1 errors, 0% packet loss, time 100
rtt min/avg/max/mdev = 24.870/119.204/213.538/94.334 ms
[root@RHELclient2 ~]# tracepath 172.168.1.10
 1?: [LOCALHOST]                      pmtu 1500
 1:  172.168.1.10                                          274.968ms !H
 1:  172.168.1.10                                            1.251ms !H
     Resume: pmtu 1500
[root@RHELclient2 ~]# _
```

when the ip forwarding is disabled

when the ip forwarding is enabled

Repeat the same testing from PCB.



If both PCs can access each other, it verifies that the Linux system is working as the router and the IP forwarding is enabled on it.

That's all for this tutorial. In this tutorial, we learned how to use the Linux system as a router and enable IP forwarding on it.

By ComputerNetworkingNotes Updated on 2024-06-09

ComputerNetworkingNotes › Linux Tutorials › How to Configure and use Linux as a Router

Network Configuration Files in Linux Explained

Linux user Profile Management and Environment Variable