



# Linux Networking Guide

---



## Linux Networking Guide

---



### Transfer or Copy Files Through the Same Network

---



## 1. What is Linux Networking?

Linux networking is a collection of kernel subsystems, user-space tools, configuration files, and services that allow a Linux system to exchange data with other systems — whether locally or over the Internet.

At runtime, Linux networking manages:

- ⌚ Moving bits between hardware and applications
  - 🌐 Addressing and routing packets
  - 🔒 Secure transfer (via TCP/SSH)
  - 🧱 Policy enforcement (firewalls, iptables/nftables)
  - ⚡ Virtualization of networks (namespaces, bridges, VLANs, etc.)
- 



## 2. Why Do Networking?

Networking enables:

- 🤝 *Resource sharing*: Files, printers, databases, compute nodes
- 💻 *Service hosting*: Web, mail, DNS, SSH, LDAP
- 🧠 *Centralized management*: Monitoring, automation, logging
- 🚀 *Scalability*: Load balancing, clustering, cloud orchestration
- 🌐 *Remote access*: Admin, APIs, file transfer
- 🛡 *Security & isolation*: Firewalls, namespaces, VLANs
- ⚡ *Performance optimization*: Traffic shaping, QoS, offloading

Networking transforms isolated computers into an interconnected system that powers modern computing.



## 3. The Conceptual Networking Stack

---

Layer	Description	Tools
<i>Physical / Link Layer</i>	NICs, Ethernet/Wi-Fi, MAC, ARP	ethtool, ip link
<i>Network Layer</i>	IPv4/IPv6, routing, subnets	ip addr, ip route, ip neigh
<i>Transport Layer</i>	TCP (reliable), UDP (fast)	ss, netstat, tcpdump
<i>Socket API &amp; Kernel</i>	App ↔ Kernel interface, buffering	socket, netfilter
<i>Application Layer</i>	Services (HTTP, SSH, DNS, NFS)	-
<i>Management Layer</i>	NetworkManager, DHCP, systemd-networkd	-

## ⌚ 4. Basic Network Configuration

### 🔍 View IP Configuration

bash

hostname -l

,

```
vboxuser@ubuntu1:~/Desktop$ hostname -I
10.0.2.15 fd17:625c:f037:2:689d:a074:296:b111 fd17:625c:f037:2:a00:27ff:fe45:641
e
```

## 🔒 SSH (Secure Shell)

### ✳️ Install and Start SSH

bash

sudo apt install openssh-server

sudo systemctl enable ssh

sudo systemctl start ssh

### 🔌 Check SSH Status

bash

sudo systemctl status ssh

### 💻 Connect to Remote System

bash

ssh username@192.168.1.5

### 📦 Copy Files with SCP

bash

scp file.txt username@192.168.1.5:/home/username/

## Copy a Directory

bash

```
scp -r myfolder username@192.168.1.5:/home/username/
```

## Copy SSH Key

bash

```
ssh-copy-id username@192.168.1.5
```

---

# FTP (File Transfer Protocol)

## Install and Start FTP Server

bash

```
sudo apt install vsftpd  
sudo systemctl enable vsftpd  
sudo systemctl start vsftpd
```

## Check FTP Status

bash

```
sudo systemctl status vsftpd
```

## Connect to FTP Server

bash

```
ftp 192.168.1.5
```

*Login Example:*

```
Name (192.168.1.5:yourusername): ftpuser
```

```
Password: *****
```

## 📸 Screenshot Example:

```
vboxuser@ubuntu1:~/Desktop$ sudo systemctl status ssh
[sudo] password for vboxuser:
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Sun 2025-11-02 10:05:03 UTC; 1min 2s ago
    Invocation: 756062c379314049bc6d443abf80ee85
  TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 5544 (sshd)
     Tasks: 1 (limit: 3866)
    Memory: 1.6M (peak: 2.2M)
      CPU: 25ms
     CGroup: /system.slice/ssh.service
             └─5544 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 02 10:05:03 ubuntu1 systemd[1]: Starting ssh.service - OpenBSD Secure Shell...
Nov 02 10:05:03 ubuntu1 sshd[5544]: Server listening on 0.0.0.0 port 22.
Nov 02 10:05:03 ubuntu1 sshd[5544]: Server listening on :: port 22.
Nov 02 10:05:03 ubuntu1 systemd[1]: Started ssh.service - OpenBSD Secure Shell...
lines 1-18/18 (END)
```

## 🧠 Basic FTP Commands

```
bash
ls # List files
cd folder # Change directory
get file.txt # Download file
put file.txt # Upload file
mget * # Download multiple files
bye # Exit FTP
```

*Configuration File:*

/etc/vsftpd.conf

## 👥 User & Group Access Management

Action	Command
➕ Create User	sudo adduser username
✖ Delete User	sudo deluser username
👥 Create Group	sudo groupadd developers
➕ Add User to Group	sudo usermod -aG developers username
🔍 View Groups	groups username

Action	Command
Change File Owner	sudo chown username filename
Change Group Owner	sudo chgrp developers filename
Change Permissions	chmod 755 filename
View Permissions	ls -l
Switch User	su - username
Lock / Unlock User	sudo passwd -l username / sudo passwd -u username

---

## Quick Command Summary

Command	Description
ip addr	Show IP address
ping	Test connectivity
ssh user@host	Remote login
scp	Secure file copy
ftp	Connect to FTP
adduser	Create new user
groupadd	Create new group
chmod	Change permissions
chown	Change ownership

---

## File Transfer Between Systems

### ◆ Method 1 — Using SCP

bash

```
scp [source_file] [username]@[destination_IP]:[destination_path]
```

*Example:*

bash

```
scp /home/user/Documents/test.txt ubuntu@192.168.1.15:/home/ubuntu/Desktop/
```

- The file test.txt will be copied to /home/ubuntu/Desktop/ on System B.

### ◆ Method 2 — Copying Folders

bash

```
scp -r /home/user/myfolder ubuntu@192.168.1.15:/home/ubuntu/Documents/
```

### ◆ Method 3 — Using rsync (Faster)

bash

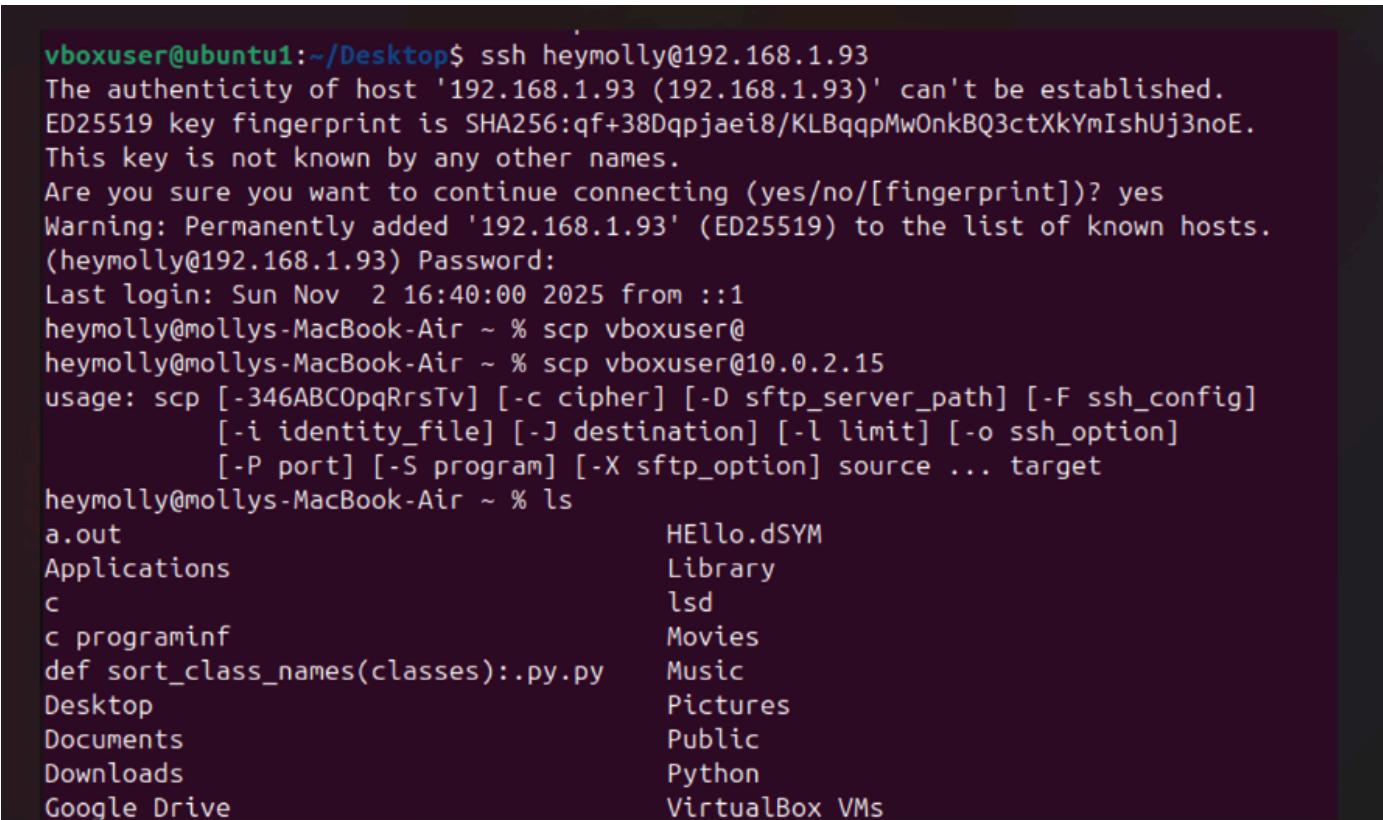
```
sudo apt install rsync
```

```
rsync -avz /home/user/Documents/ ubuntu@192.168.1.15:/home/ubuntu/Backup/
```

Flags:

- -a → Archive mode
- -v → Verbose
- -z → Compression

#### 👉 Screenshot:



```
vboxuser@ubuntu1:~/Desktop$ ssh heymolly@192.168.1.93
The authenticity of host '192.168.1.93 (192.168.1.93)' can't be established.
ED25519 key fingerprint is SHA256:qf+38Dqpjaei8/KLBqqpMwOnkBQ3ctXkYmIshUj3noE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.93' (ED25519) to the list of known hosts.
(heymolly@192.168.1.93) Password:
Last login: Sun Nov  2 16:40:00 2025 from ::1
heymolly@mollys-MacBook-Air ~ % scp vboxuser@
heymolly@mollys-MacBook-Air ~ % scp vboxuser@10.0.2.15
usage: scp [-346ABC0pqRrsTv] [-c cipher] [-D sftp_server_path] [-F ssh_config]
           [-i identity_file] [-J destination] [-l limit] [-o ssh_option]
           [-P port] [-S program] [-X sftp_option] source ... target
heymolly@mollys-MacBook-Air ~ % ls
a.out                               Hello.dSYM
Applications                         Library
c                                     lsd
c_programinf                         Movies
def sort_class_names(classes):.py.py   Music
Desktop                             Pictures
Documents                           Public
Downloads                           Python
Google Drive                         VirtualBox VMs
```

```

hey Molly@Mollys-MacBook-Air ~ % cd ~/Desktop
hey Molly@Mollys-MacBook-Air Desktop % nano hello.txt
hey Molly@Mollys-MacBook-Air Desktop % ls
| Google Chrome Mail Prime Video UTM
Brave Browser hello.txt Numbers Telegram Visual Studio Code.app
Copilot Keynote PagesTextEdit WhatsApp
Easy Streaming Launchpad Perplexity TV Xcode
hey Molly@Mollys-MacBook-Air Desktop % scp ~/Desktop/test.txt hey Molly@192.168.1.93:/users/heymolly/Desktop/
scp: stat local "/Users/heymolly/Desktop/test.txt": No such file or directory
hey Molly@Mollys-MacBook-Air Desktop % scp ~/Desktop/test.txt hey Molly@192.168.1.93:/users/heymolly/Desktop/
scp: stat local "/Users/heymolly/Desktop/test.txt": No such file or directory
hey Molly@Mollys-MacBook-Air Desktop % scp ~/Desktop/hello.txt hey Molly@192.168.1.93:/users/heymolly/Desktop/
The authenticity of host '192.168.1.93 (192.168.1.93)' can't be established.
ED25519 key fingerprint is SHA256:qf+38DqpjaeI8/KLBqqpMwOnkBQ3ctXkYmIshUj3noE.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: localhost
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.93' (ED25519) to the list of known hosts.
(hey Molly@192.168.1.93) Password:
hello.txt
hey Molly@Mollys-MacBook-Air Desktop %

```

## Deletes Files Remotely

### Option 1 — SSH into the Second System

```

bash
ssh ubuntu@192.168.1.15
cd /home/ubuntu/Desktop
rm test.txt
exit

```

### Option 2 — Delete Directly from Source

```

bash
ssh ubuntu@192.168.1.15 "rm /home/ubuntu/Desktop/test.txt"

```

To Delete a Folder:

```

bash
ssh ubuntu@192.168.1.15 "rm -r /home/ubuntu/Desktop/foldername"

```

## Screenshot:

```

hey Molly@Mollys-MacBook-Air Desktop % ls
| Google Chrome Mail Prime Video UTM
Brave Browser hello.txt Numbers Telegram Visual Studio Code.app
Copilot Keynote PagesTextEdit WhatsApp
Easy Streaming Launchpad Perplexity TV Xcode
hey Molly@Mollys-MacBook-Air Desktop % exit
Connection to 192.168.1.93 closed.
vboxuser@ubuntu1:~/Desktop$ rm -rf hello.txt
vboxuser@ubuntu1:~/Desktop$ ls
590025385_prog_1 590025385_prog_14 590025385_prog_3.c 590025385_prog_8 macros1.c
590025385_prog_10 590025385_prog_14.c 590025385_prog_3clear 590025385_prog_8.c m.c
590025385_prog_10.c 590025385_prog_15 590025385_prog_4 590025385_prog_9 parent_folder
590025385_prog_11 590025385_prog_15.c 590025385_prog_4.c 590025385_prog_9.c prime.sh
590025385_prog_11.c 590025385_prog_16 590025385_prog_5 armstrong.sh product
590025385_prog_11clear 590025385_prog_16.c 590025385_prog_5.c c Projects
590025385_prog_12 590025385_prog_1.c 590025385_prog_6 check_file_permission.sh script
590025385_prog_12.c 590025385_prog_2 590025385_prog_6.c count_line_words.txt string_ops.sh
590025385_prog_13 590025385_prog_2.c 590025385_prog_7 file1.txt sum_of_digits
590025385_prog_13.c 590025385_prog_3 590025385_prog_7.c m
vboxuser@ubuntu1:~/Desktop$ 

```

```
ds_starter
'Engineering Physics by Hitendra K Malik A K Singh (z-lib.org).pdf'
f1.txt
factorial.sh
fdkjn
file1.txt
file2.txt
folder
g2.png
HelloWorld.class
j1.java
JoplinBackup
LAB0.pdf
lab1.md
LAB1.pdf
lab2.md
LAB2.pdf
lab3.md
LAB3.pdf
'lab4 (1).md'
lab4.md
LAB4.pdf
LAB5.pdf
Lab6_Shell_Scripting_Basics.md
Linux_lab
Linux_Lab
'Linux-lab_Newcommands.md at main · muditsharmacr2-del_LInux-lab.pdf'
linuxtransfer.txt
managingself1.pdf
'Managing self.pdf'
'Managing self.pptx'

qw5.sh
qw6.sh
qw7.sh
qw8.sh
qw9.sh
school.pdf
'Screenshot From 2025-09-24 16-07-03.png'
scripts
_sh.1
Shell
snake.c
snap
starterkit
starter_kit.sh
starter_kit_streamlit.sh
temp
Templates
testfile1.txt
transfertest.txt
Videos
w.sh.save
ws.txt
zf.py
zx1.sh
zx2.sh
zx3.sh
zx4.sh
zx5.sh
zx6.sh
zxn.sh
```

## 🏁 Conclusion

With Linux networking tools like SSH, SCP, and FTP, you can easily transfer, manage, and secure files across systems — empowering automation, collaboration, and control in any networked environment.

💡 *Tip:* Always verify IP addresses, permissions, and connectivity before file transfer to prevent errors or unauthorized access.