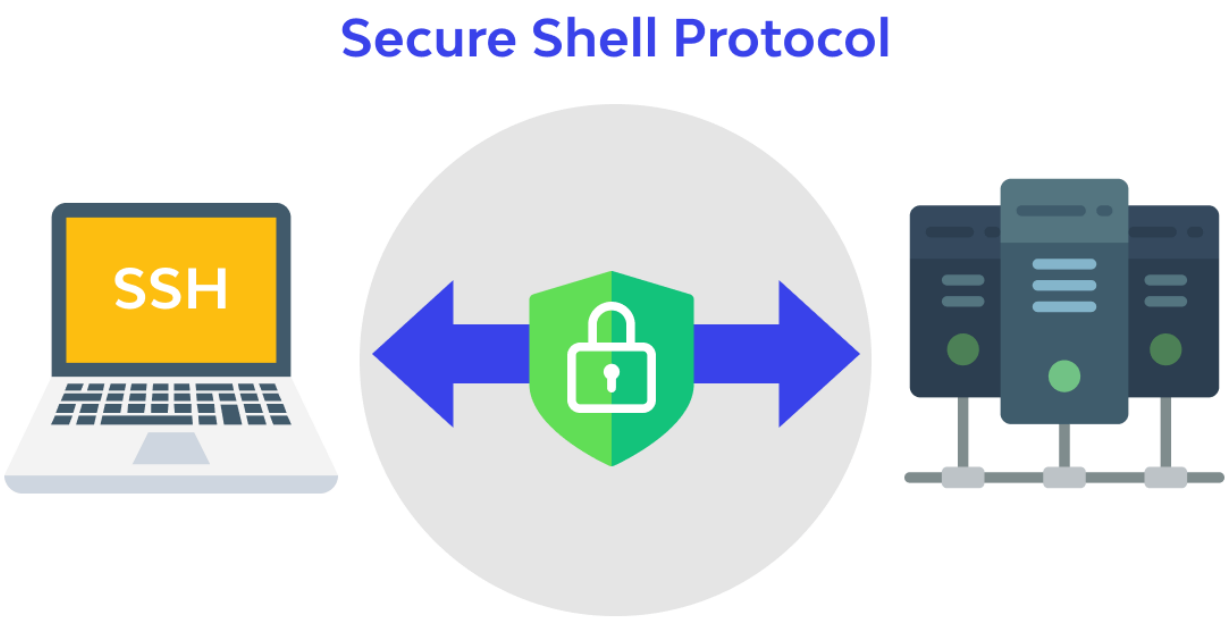
General Overview — Linux Networking & SSH



Scope & Purpose

This document is a concise, practical overview of Linux networking essentials and Secure Shell (SSH). It combines foundational networking concepts, common commands, and SSH workflows (keys, agents, forwarding, file transfer, and troubleshooting). All screenshots and images already present in the file are preserved for reference and examples.

Linux Networking Essentials (quick reference)

- Network interfaces and addresses
  - View interfaces: `ip addr show` or `hostname -I`

```
ujjwaltyagi@ujjwaltyagi:~$ sudo systemctl status ssh
[sudo] password for ujjwaltyagi:
○ ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: inactive (dead)
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
ujjwaltyagi@ujjwaltyagi:~$
```

- Routing & connectivity
  - Show routing table: `ip route`
  - Test reachability: `ping <host>` / `traceroute <host>`
- DNS

    - Query DNS: `dig example.com` or `nslookup example.com`

- Firewall basics (ufw / iptables)
    - Allow SSH: `sudo ufw allow ssh`
    - Check rules: `sudo ufw status verbose`

- Common troubleshooting
    - Check links: `ip link`
    - Check active sockets: `ss -tulpn`
    - View logs: `sudo journalctl -u NetworkManager -e`

SSH — Conceptual Overview

- What is SSH?
    - Secure remote login, command execution and file transfer over an encrypted channel.
    - Default TCP port: 22.

- Key benefits
    - Confidentiality (encryption), integrity, and strong authentication (public/private keys).
    - Supports port forwarding, X11 forwarding, agent forwarding, and file transfer (SCP/SFTP).

Practical SSH: Commands & Workflows

1) Generate keypair (recommended ed25519)

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Example (from this repo):

```
ujjwaltyagi@ujjwaltyagi:~$ ssh-keygen -t ed25519 -C "ujjwaltyagi458@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ujjwaltyagi/.ssh/id_ed25519):
Created directory '/home/ujjwaltyagi/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ujjwaltyagi/.ssh/id_ed25519
Your public key has been saved in /home/ujjwaltyagi/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:0j2kZRBeWWKnJEP6S2mOh2ixJk3hxioPI9115X93ev4 ujjwaltyagi458@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|     .+ =o+ .    |
|     . = = +     |
|  . .   o *      |
| o . . o O       |
|  *   B S +      |
| * = B +   o     |
|B B + +     . . o|
|oB   .       . oo|
|  .           .oE|
+----[SHA256]-----+
ujjwaltyagi@ujjwaltyagi:~$ ls ~/.ssh
id_ed25519  id_ed25519.pub
ujjwaltyagi@ujjwaltyagi:~$ eval "$(ssh-agent -s)"
Agent pid 4677
ujjwaltyagi@ujjwaltyagi:~$ ssh-add ~/.ssh/id_ed25519
Identity added: /home/ujjwaltyagi/.ssh/id_ed25519 (ujjwaltyagi458@gmail.com)
ujjwaltyagi@ujjwaltyagi:~$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIN74CRUJb/1CpEZQD1gDhrAEeRM7syD4c2OdZ0+CQzCR ujjwaltyagi458@gmail.com
ujjwaltyagi@ujjwaltyagi:~$
```

2) Install public key on remote

Preferred:

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub user@remote.host
```

Manual:

```
cat ~/.ssh/id_ed25519.pub | ssh user@remote.host 'mkdir -p ~/.ssh && cat >>
~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys'
```

Example image:

```
ujjwaltyagi@ujjwaltyagi:~$ ssh-copy-id -i ~/.ssh/id_ed25519.pub ujjwaltyagi@10.0.2.15
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ujjwaltyagi/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ujjwaltyagi@10.0.2.15's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ujjwaltyagi@10.0.2.15'"
and check to make sure that only the key(s) you wanted were added.
```

3) Connect & run commands

Interactive:

```
ssh user@192.168.1.100
```

Single command:

```
ssh user@192.168.1.100 'uname -a && uptime'
```

Example:

```
ujjwaltyagi@ujjwaltyagi:~$ ssh ujjwaltyagi@10.0.2.15
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is enabled.

118 updates can be applied immediately.
69 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

SSH Agent & Config

- Start agent and add key:

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
ssh-add -l
```

```
ujjwaltyagi@ujjwaltyagi:~$ eval "$(ssh-agent -s)"
Agent pid 4677
ujjwaltyagi@ujjwaltyagi:~$ ssh-add ~/.ssh/id_ed25519
Identity added: /home/ujjwaltyagi/.ssh/id_ed25519 (ujjwaltyagi458@gmail.com)
```

- Useful `~/.ssh/config` entry:

```
Host local-server
  HostName 192.168.1.100
  User youruser
  IdentityFile ~/.ssh/id_ed25519
  IdentitiesOnly yes
```

```
ujjwaltyagi@ujjwaltyagi:~$ nano ~/.ssh/config
ujjwaltyagi@ujjwaltyagi:~$ cat ~/.ssh/config
Host uni
        HostName 10.0.2.15
        User ujjwaltyagi
        Port 22
        IdentityFile ~/.ssh/id_ed25519
        IdentitiesOnly yes
ujjwaltyagi@ujjwaltyagi:~$ chmod 700 ~/.ssh
ujjwaltyagi@ujjwaltyagi:~$ chmod 600 ~/.ssh/config
```

Connect with:

```
ssh local-server
```

```
ujjwaltyagi@ujjwaltyagi:~$ ssh uni
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is enabled.

118 updates can be applied immediately.
69 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Tue Oct 28 13:21:11 2025 from 10.0.2.15
```

Port Forwarding & Tunneling

- Local forward (access remote web app locally):

```
ssh -L 8080:localhost:8000 user@192.168.1.100
# then browse http://localhost:8080
```

```
ujjwaltyagi@ujjwaltyagi:~$ ssh -L 8080:localhost:8080 ujjwaltyagi@10.0.2.15
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is enabled.

118 updates can be applied immediately.
69 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Tue Oct 28 13:35:04 2025 from 10.0.2.15
ujjwaltyagi@ujjwaltyagi:~$
```

- Remote forward and dynamic SOCKS proxy (SSH as proxy):

```
ssh -R 9000:localhost:22 user@remote.host
ssh -D 1080 user@remote.host   # dynamic SOCKS proxy
```

File Transfer (SCP / SFTP)

- Copy single file:

```
scp localfile.txt user@192.168.1.100:/home/user/
```

```
ujjwaltyagi@ujjwaltyagi:~$ scp modify1.txt ujjwaltyagi@10.0.2.15:/home/ujjwaltya
gi/
modify1.txt                                   100%  378    61.5KB/s   00:00
ujjwaltyagi@ujjwaltyagi:~$
```

- Copy directory:

```
scp -r project/ user@192.168.1.100:/home/user/project_backup/
```

```
ujjwaltyagi@ujjwaltyagi:~$ scp -r unit_8/ ujjwaltyagi@10.0.2.15:/home/ujjwaltyag
i/scripts/
check_file_permission.sh                      100%   30    17.6KB/s   00:00
```

```
ujjwaltyagi@ujjwaltyagi:~$ ls scripts
unit_8
```

- Interactive SFTP:

```
sftp user@192.168.1.100
sftp> put localfile
```

```
ujjwaltyagi@ujjwaltyagi:~$ sftp ujjwaltyagi@10.0.2.15
Connected to 10.0.2.15.
sftp> pwd
Remote working directory: /home/ujjwaltyagi
sftp> put modify1.txt
Uploading modify1.txt to /home/ujjwaltyagi/modify1.txt
modify1.txt                                   0%    0     0.0KB/s   --:-- ETA
sftp> put floatint.txt
Uploading floatint.txt to /home/ujjwaltyagi/floatint.txt
floatint.txt                                  0%    0     0.0KB/s   --:-- ETA
sftp> bye
ujjwaltyagi@ujjwaltyagi:~$
```

🛠 Troubleshooting & Best Practices

- Permissions: `chmod 700 ~/.ssh && chmod 600 ~/.ssh/authorized_keys`
- Debugging: `ssh -vvv user@host`
- Check server logs: `sudo journalctl -u ssh -e` or `sudo tail -n 200 /var/log/auth.log`
- Use strong keys (ed25519), protect private keys with passphrase
- Disable root login and password auth on servers for production:
  - Edit `/etc/ssh/sshd_config`:
    - `PermitRootLogin no`
    - `PasswordAuthentication no`
  - Restart: `sudo systemctl restart ssh`

📌 Quick Commands Summary

```
# networking
ip addr show
ip route
ss -tulpn
ping 8.8.8.8

# SSH basics
ssh-keygen -t ed25519
ssh-copy-id user@host
ssh user@host
scp file user@host:/path/
ssh -L 8080:localhost:8000 user@host
```

📎 References & Next Steps

- man pages: `man ssh`, `man sshd`, `man scp`, `man sftp`
- Further topics to add (recommended): VPNs (WireGuard/Tailscale), SSH certificate auth, SSH jump hosts, X11 vs Wayland caveats.
