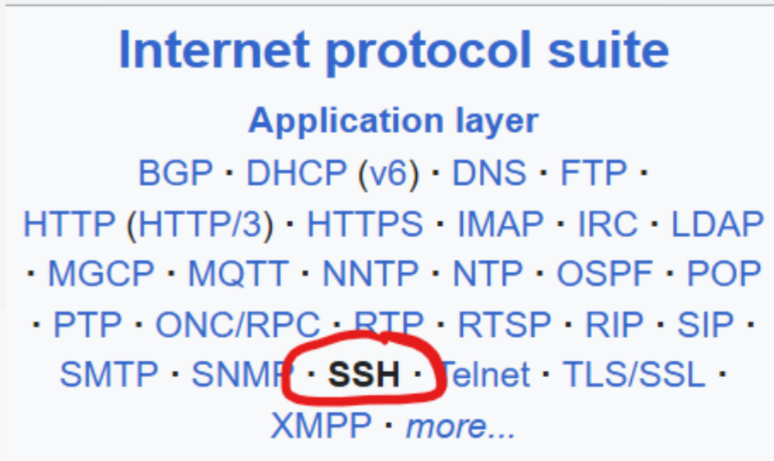
```
$ nmcli connection show --active
NAME          UUID                                  TYPE     DEVICE
Comtrend7FB9  2f5a6217-37c7-449f-bfaa-1d3fa5283482  wifi     wlo1
virbr0        13b49d1a-145e-4332-bbf1-6de6f6816531  bridge   virbr0
$ nmcli device show | head -10
GENERAL.DEVICE:                         wlo1
GENERAL.TYPE:                           wifi
GENERAL.HWADDR:                         7C:67:A2:CF:9F:EF
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     Comtrend7FB9
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManage
nnection/1
IP4.ADDRESS[1]:                         192.168.0.22/24
```

Connecting and Configuring your RPi using SSH, mDNS and Network Manager(nmcli)

Introduction to SSH

- **SSH** (Secure Shell) is a protocol that provides a secure way to access a remote computer or server.
- It encrypts the communication between your device and the remote device, ensuring data security.
- SSH is widely used for
  - Accessing remote hosts/servers
  - Secure File Transfer (SCP and SFTP)
  - Access to Cloud Services and Virtual Machines
  - Tunneling and Port Forwarding



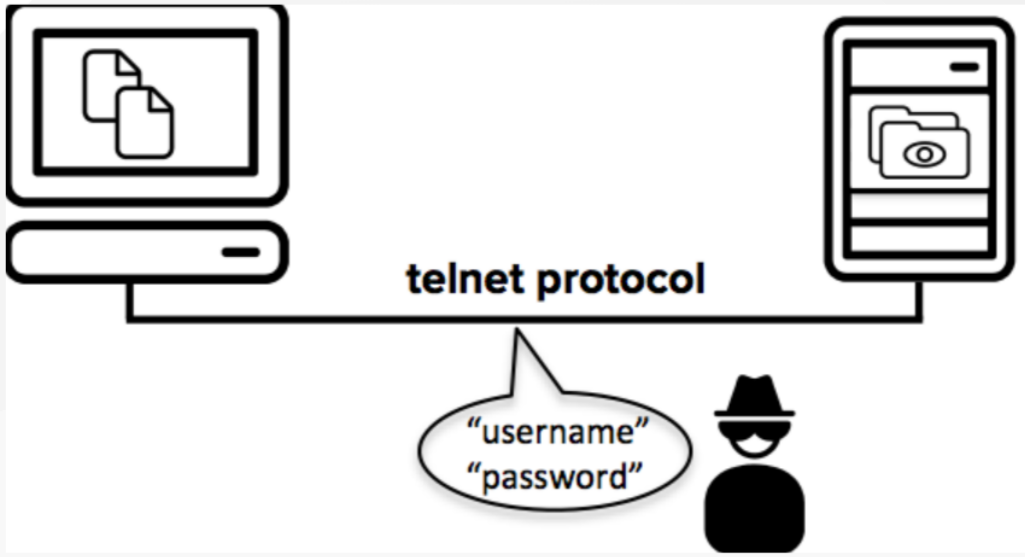
Internet protocol suite

Application layer

BGP · DHCP (v6) · DNS · FTP · HTTP (HTTP/3) · HTTPS · IMAP · IRC · LDAP · MGCP · MQTT · NNTP · NTP · OSPF · POP · PTP · ONC/RPC · RTP · RTSP · RIP · SIP · SMTP · SNMP · **SSH** · Telnet · TLS/SSL · XMPP · *more...*

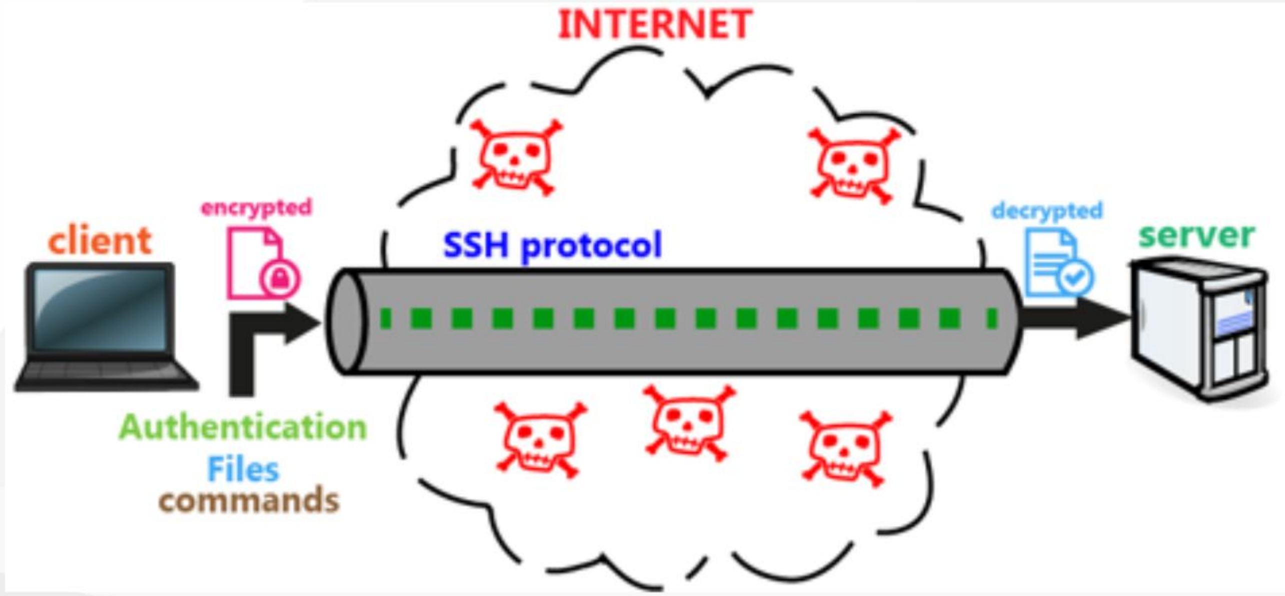
Why SSH is Important

- Need for security in remote communication
  - Telnet was its predecessor. Not good enough for today
- Vulnerability of unencrypted protocols(last lab - UDP payload unencrypted)
- SSH allow you to:
  - protecting sensitive data
  - ensure authentication
  - preventing unauthorized access.



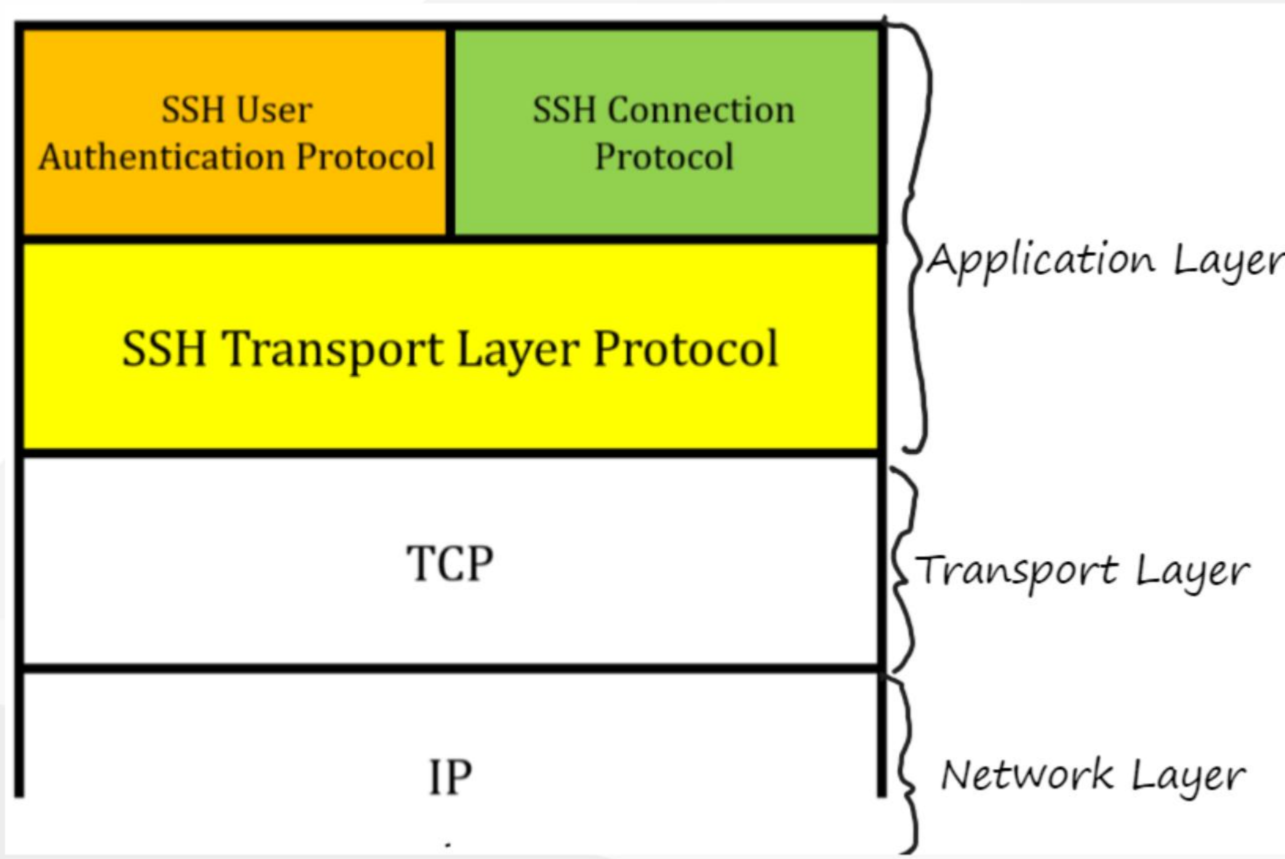
How SSH Works

- Client-Server model
- SSH operates over TCP, typically using port 22
- Components:
    - SSH client
    - SSH server



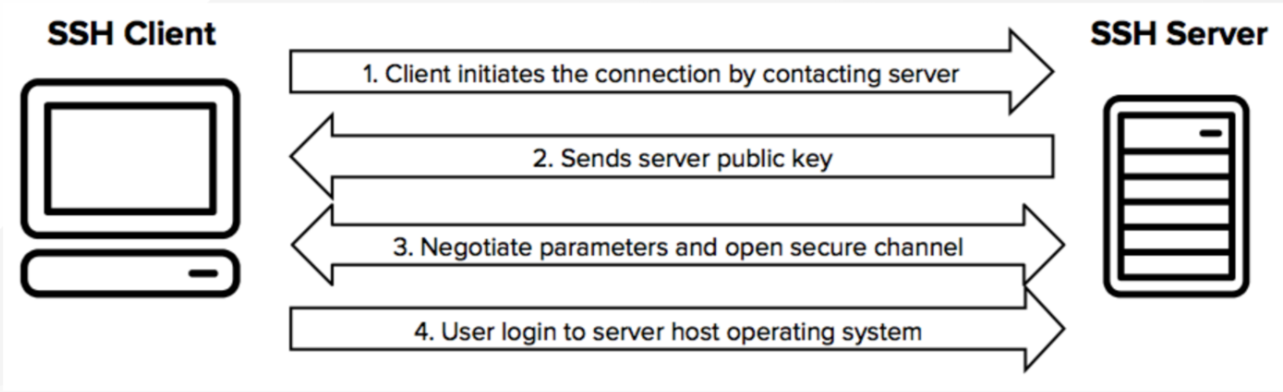
SSH Protocol

- Transport layer Protocol: Handles encryption
- User authentication protocol: Verifies the client's identity
- Connection protocol: Manages channels for services like shell, SFTP, port forwarding



How SSH Works - Encryption

- SSH uses **public-key cryptography** for authentication and **encryption**
  - **Encryption**: SSH encrypts data transmitted over a network, making it unreadable to unauthorised entities
  - **Authentication**: SSH verifies the identity of both the client and the server, including password or public-key authentication


SSH Client
SSH Server
1. Client initiates the connection by contacting server
2. Sends server public key
3. Negotiate parameters and open secure channel
4. User login to server host operating system

SSH Authentication Protocols

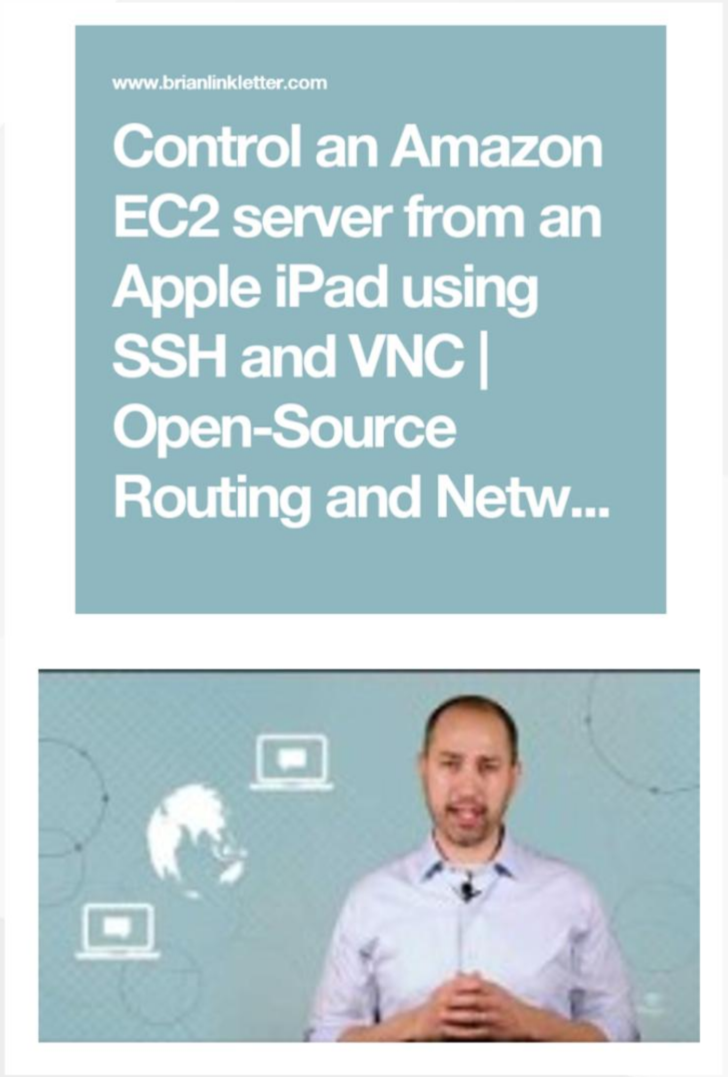
- Password-based authentication
  - You start an SSH connection using a command like ssh user@hostname, the SSH client sends a connection request to the server
  - The server responds by asking for the user's identity, which, in this case, is verified with a password.
- Public key authentication:
  - SSH uses a **key pair**: a public key and a private key.
  - The **public key** is stored on the server; the **private key** stays with the client.
  - When connecting, the server verifies the private key without transmitting it, ensuring security.

SSH Availability

- SSH is present on most Operating Systems
  - MAC, Linux, Windows 10+
- SSH widely used in Cloud Computing

  Access remote VMs in the cloud using SSH
  - You will see it again in Dev Ops

    https://www.youtube.com/watch?v=Atbl7D_yPug


www.brianlinkletter.com
Control an Amazon EC2 server from an Apple iPad using SSH and VNC | Open-Source Routing and Netw...

Advantages of Using SSH

- **Security**: SSH encrypts data to protect it from eavesdropping.
- **Authentication**: Uses key-based authentication, reducing reliance on passwords.
- **Convenience**: Provides command-line access and allows file transfers.
- **Port Forwarding**: Enables secure tunneling for other applications (e.g., databases).

Setting Up SSH on Raspberry Pi

- Do it when you install the OS (see Lab)
- If using RPi as Desktop:
  - Open **Raspberry Pi Configuration** and enable SSH in the **Interfaces** tab.
  - Alternatively, use the command: `sudo systemctl enable ssh` and `sudo systemctl start ssh`.
- Note the IP address of your Raspberry Pi, which you'll need to connect via SSH.

Connecting to Raspberry Pi via SSH (Command Line)

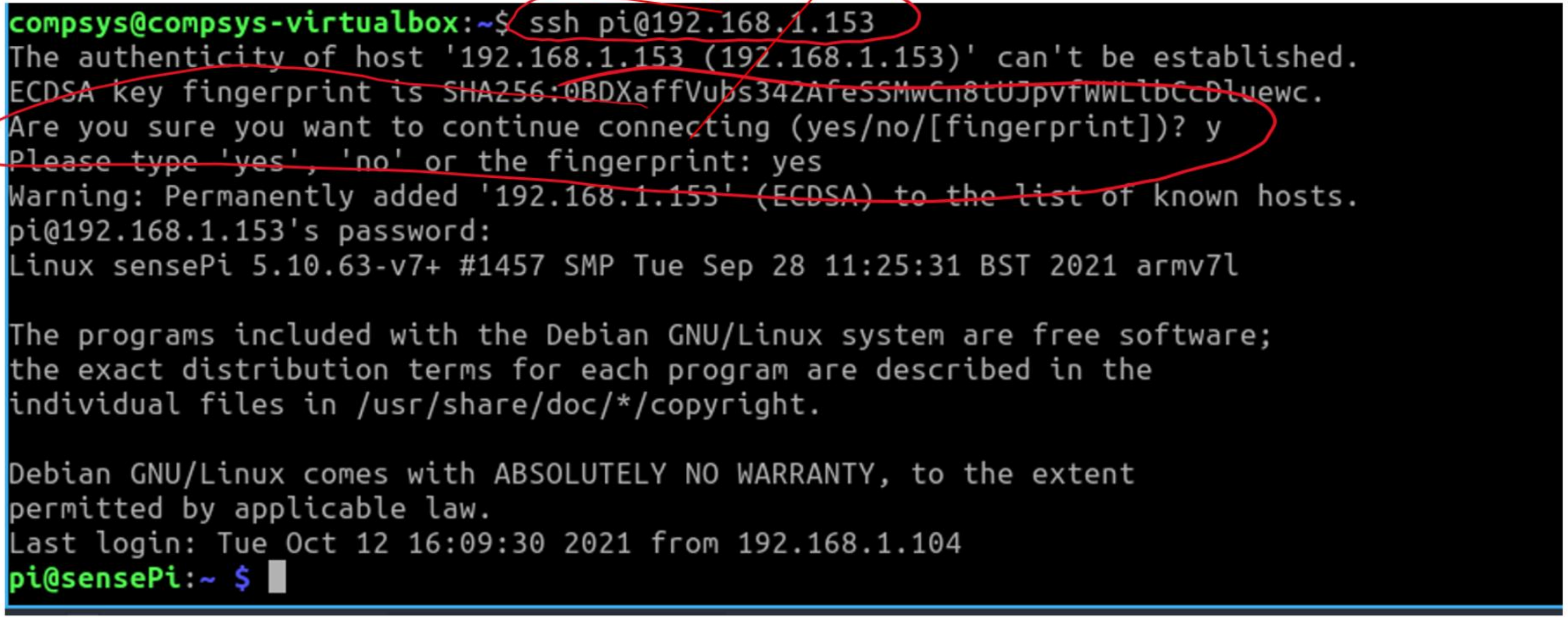
- Open the terminal on your computer.
- Use the SSH command: `ssh pi@<RaspberryPi_IP>`, replacing `<RaspberryPi_IP>` with your Pi's IP address.
- Accept the SSH key fingerprint the first time you connect.
- Enter your **password** when prompted. You are now connected to the Raspberry Pi.
- You can also use host name instead of IP address: ssh frank@raspberrypi.local or user@someDomainName
  - Hostname resolved to IP address using DNS/mDNS – more later

Using Secure Shell

At command line, connect to remote server using ssh command with user name and address(either IP or Domain Address) of remote server:

ssh UserName@SSHserver.com
ssh pi@10.1.1.234

If it's the first connection between the host and the server, you will be prompted with the remote host's public key fingerprint and prompted to connect, despite there having been no prior connection:



```
compsys@compsys-virtualbox:~$ ssh pi@192.168.1.153
The authenticity of host '192.168.1.153 (192.168.1.153)' can't be established.
ECDSA key fingerprint is SHA256:0BDXaffVubs342AfeSSMwCn8tUJpvfWWLlbCcDluewc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.153' (ECDSA) to the list of known hosts.
pi@192.168.1.153's password:
Linux sensePi 5.10.63-v7+ #1457 SMP Tue Sep 28 11:25:31 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 12 16:09:30 2021 from 192.168.1.104
pi@sensePi:~ $
```

SSH Best Practices

- Use **SSH keys** instead of passwords for enhanced security
  - OK to use passwords for now
- Change the default SSH port from **22** to reduce unauthorised access attempts.
- Regularly update SSH software on both client and server.
- Disable root login over SSH to prevent potential security breaches.

From DNS to mDNS

- DNS translates domain names to IP addresses (e.g., example.com → 93.184.216.34)
  – Covered in previous weeks
- Requires a DNS server and usually works across the Internet or large networks
- But what about local networks with no DNS server?

Multicast DNS (mDNS) Overview

- mDNS = “Multicast DNS”
- Works on local networks (LANs) without a DNS server
- Devices broadcast queries to find others using a special multicast address
- Defined in **RFC 6762**
  - **Request for Comments used to** describe, propose, or standardise aspects of the internet and computer networking.

How mDNS Works

- Uses UDP port 5353
- Multicast IP address: 224.0.0.251 (IPv4)
- Devices send queries like 'Who has myprinter.local?'
- The device owning that name responds with its IP address
- **Domain names always end in .local**

mDNS Examples and Uses

- Apple’s Bonjour and Linux’s Avahi use mDNS
- Smart home and IoT devices (e.g., raspberrypi.local)
- Network printers, media servers, and file shares
- Simplifies discovery and setup — no need to know IP addresses

CLI Network Configuration on Linux – Overview

Content distilled from Red Hat Netowrking Guide – Enterprise Linux

Understanding Network Configuration

- Network interfaces identified as **eth0**, **wlan0**, etc.
- Configuration controls how the device connects to a network
- Can be managed via GUI, NetworkManager, or CLI tools
- **Command line offers speed, flexibility, and automation**

Viewing Network Interfaces

- **ip link show** — list all interfaces
- **ip addr show** — show assigned IP addresses
- **ip route** — display routing table
- **ping 8.8.8.8** or **ping raspberrypi.local** — test connectivity

Temporary Network Configuration

- Assign IP: `sudo ip addr add 192.168.1.10/24 dev eth0`
- Bring interface up/down: `sudo ip link set eth0 up / down`
- Add route: `sudo ip route add default via 192.168.1.1`
- **Temporary only — lost after reboot**

Managing Network with **nmcli**

- show interfaces : `nmcli device status`
- list saved connections: `nmcli connection show`
- scan for networks: `nmcli device wifi list`
- Connect : `sudo nmcli device wifi connect "SSID" password "mypassword"`
- Modify: `sudo nmcli connection modify <name> ipv4.addresses 192.168.1.20/24`

Network Troubleshooting Tools

- Test connectivity and DNS: ping, traceroute, dig, nslookup
- verbose logs: nmcli general logging level DEBUG
- view service logs: journalctl -u NetworkManager
- reset connections: sudo systemctl restart NetworkManager

Lots more detail at Red Hat Networking Guide for Enterprise Linux:
https://docs.redhat.com/en/documentation/red_hat_ent

End of Presentation