

Network Management using

‘ nmcli ’

LinuXpert Systems

WhatsApp: 98841 65649, 90808 01382

E-mail: baskar@linuxpert.in

TABLE OF CONTENTS

1. NetworkManager and the ‘**nmcli**’ Mental Model 2
2. Installing, Verifying and Getting Help 3
3. General Status, Connectivity and Radio Control 5
4. Discovering Devices 7
5. Discovering Connection Profiles 8
6. Output Formatting for Humans and Scripts 9
7. Wired Ethernet: Automatic DHCP 10
8. Wired Ethernet: Static IPv4 11
9. Modifying Existing Ethernet Profiles 12
10. Wired Ethernet: DNS Configuration 13
11. Wired Ethernet: Autoconnect and Profile Priority 14
12. Wired Ethernet: MAC, MTU and Link Properties 15
13. Wired Ethernet: Multiple Addresses 16
14. Wired Ethernet: Static Routes 17
15. IPv4 Route Metrics and Default Route Control 18
16. IPv6: Automatic Configuration 19
17. IPv6: Static Addresses, DNS and Routes 20
18. Wi-Fi Discovery and Scanning 21
19. Wi-Fi: Connect to WPA/WPA2/WPA3-Personal 22
20. Wi-Fi: Open Networks, Hidden SSIDs and BSSID 23
21. Wi-Fi: Band, Channel and Power-Management Considerations .. 24
22. Wi-Fi: WPA-Enterprise / 802.1X Concepts 25
23. Wi-Fi: WPA-Enterprise Certificate-Based TLS 26
24. Wi-Fi: Disconnect, Reconnect and Forget 27
25. Wi-Fi: Roaming, Priority and Automatic Connection 28
26. Wi-Fi: Hotspot / Shared Connection 29
27. Wi-Fi Security Profiles and Secrets 30
28. DHCP Lease and Renewal Troubleshooting 31

29. Device Reapply, Disconnect and Reconnect 32
30. Device Management: Managed/Unmanaged State 33
31. VLAN Configuration 34
32. Linux Bridges 35
33. Bonding / Link Aggregation 36
34. Teaming, Tunnels and Other Connection Types 37
35. WireGuard Profiles 38
36. Connection Clone, Import, Export and Migration 39
37. Connection Editing with **nmcli con edit** 40
38. Connection Profile Inspection in Depth 41
39. Common **nmcli** Errors and Their Meaning 42
40. Logs and Troubleshooting Workflow 43
41. A Complete Wired Troubleshooting Lab 44
42. A Complete Wireless Troubleshooting Lab 45
43. Scripting **nmcli** Safely 47
44. Passwords, Secrets and Automation 48
45. Remote Server Safety and Recovery 49
46. Advanced Profile Properties Worth Knowing 50
47. Capstone Exercise: Build a Dual Wired / Wireless Lab 51
48. '**nmcli**' Command Cheat Sheet 52
49. Reference, Further Reading and Final Checklist 53

‘nmcli’ for Wired and Wireless Network Configuration

A comprehensive, practical guide for Linux students, administrators and trainers

This guide teaches NetworkManager's command-line client, **nmcli**, from first inspection through wired Ethernet, Wi-Fi, IPv4/IPv6, DNS, routing, VLANs, bridges, bonds, VPN-related profiles, scripting, troubleshooting and recovery. It emphasizes reusable connection profiles rather than one-off interface changes.

What you will learn

- How NetworkManager represents devices, connection profiles and active connections.
- How to inspect and configure Ethernet and Wi-Fi without a graphical desktop.
- How to build DHCP and static IPv4/IPv6 configurations.
- How to manage DNS, routes, metrics, autoconnect and profile priority.
- How to automate nmcli safely in shell scripts and training laboratories.

Scope

Examples target modern NetworkManager releases. Individual properties can vary by NetworkManager version, distribution, driver and connection type.

Important: Commands that change networking normally require root privileges. Always keep an alternative management path when changing a remote server's active network connection.

1. NetworkManager and the ‘nmcli’ Mental Model

NetworkManager manages network devices and connection profiles. A profile contains the intended configuration; activation applies that profile to a device. One device can have multiple profiles, but normally only one profile is active on a device at a time.

Concept	Meaning	Typical command
Device	Physical or virtual network interface	nmcli device status
Connection profile	Saved configuration	nmcli connection show
Active connection	Profile currently applied	nmcli connection show --active
AP	Wi-Fi access point	nmcli device wifi list
NetworkManager state	Overall manager status	nmcli general status

Why profiles matter

Profiles make it possible to keep DHCP, static, office, lab, VPN and Wi-Fi configurations separately and activate the appropriate one.

Aliases

nmcli accepts abbreviations when they are unambiguous. For teaching, the full forms such as 'connection', 'device', 'show', and 'modify' are clearer.

Command examples

Overall status

```
nmcli general status
```

Shows NetworkManager's overall state.

Devices

```
nmcli device status
```

Shows device, type, state and connection.

Profiles

```
nmcli connection show
```

Lists saved profiles.

---

2. Installing, Verifying and Getting Help

On most NetworkManager-based distributions **nmcli** is installed with NetworkManager. Before teaching configuration, verify both the client and daemon.

Distribution package examples

- Debian/Ubuntu: **apt install network-manager**
- Fedora/RHEL-family: **dnf install NetworkManager**
- Arch-family: **pacman -S networkmanager**

Lab preparation

If another networking service is controlling the same interface, decide which service owns it before testing **nmcli**. Mixed ownership can produce confusing results.

Command examples

Version

```
nmcli --version
```

Records the nmcli/NetworkManager version used in a lab.

Help

```
nmcli --help
nmcli connection --help
nmcli device --help
```

Use command-specific help when syntax differs between versions.

NetworkManager service

```
systemctl status NetworkManager
systemctl is-active NetworkManager
```

The service name is normally NetworkManager.

Man pages

```
man nmcli
man nmcli-examples
man nm-settings-nmcli
```

The settings manual is especially useful for discovering properties.

Important: Do not blindly disable another networking stack on a production system. First determine how that distribution and machine are intended to manage networking.

3. General Status, Connectivity and Radio Control

The general, networking and radio objects provide quick high-level control.

Connectivity is not a ping

The connectivity state is NetworkManager's assessment and should not replace application-level tests such as ping, DNS queries or HTTPS tests.

Training exercise

Turn Wi-Fi off, observe device state, turn it back on, then compare 'nmcli device status' before and after.

Command examples

General state

```
nmcli general status
nmcli -t -f RUNNING general
```

The terse field form is convenient for scripts.

Connectivity

```
nmcli networking connectivity
nmcli networking connectivity check
```

NetworkManager can report none, portal, limited, full or unknown.

NetworkManager networking switch

```
nmcli networking off
nmcli networking on
```

Disabling networking deactivates NetworkManager-managed interfaces.

Wi-Fi radio

```
nmcli radio wifi
nmcli radio wifi off
nmcli radio wifi on
```

Controls NetworkManager's Wi-Fi radio switch.

All radios

```
nmcli radio all
```

Shows radio states; 'all on'/'all off' can be used when appropriate.

4. Discovering Devices

Device inspection is the first step in both wired and wireless configuration.

Typical device type	Example name	Common role
ethernet	enp3s0	Wired LAN
wifi	wlp2s0	Wireless LAN
bridge	br0	Layer-2 bridge
bond	bond0	Link aggregation/redundancy
vlan	enp3s0.20	802.1Q VLAN interface

Command examples

Compact list

```
nmcli device status
```

Shows DEVICE, TYPE, STATE and CONNECTION.

All device details

```
nmcli device show
```

Displays properties for all devices.

One interface

```
nmcli device show enp3s0
```

Replace the interface name with the local Ethernet device.

Wi-Fi device

```
nmcli device show wlp2s0
```

Displays driver, firmware, hardware address and active IP information when available.

Only Ethernet

```
nmcli device status | grep ethernet
```

Useful for quick inventory, though structured output is better for scripts.

5. Discovering Connection Profiles

A connection profile is the saved configuration that NetworkManager activates.

Profile versus device

Deleting a profile does not necessarily delete a hardware device. Conversely, disconnecting a device does not delete its saved profile.

Duplicate names

Use UUIDs when a name is ambiguous.

Command examples

List profiles

```
nmcli connection show
```

Lists NAME, UUID, TYPE and DEVICE.

Active only

```
nmcli connection show --active
```

Shows profiles currently active.

Detailed profile

```
nmcli connection show "Wired connection 1"
```

Quote names containing spaces.

Selected fields

```
nmcli -f name,uuid,type,device connection show
```

Field selection reduces output.

Multiline readable output

```
nmcli -p -m multiline connection show "Wired connection 1"
```

Useful for classroom demonstrations.

6. Output Formatting for Humans and Scripts

nmcli supports table, multiline and terse output. Scripts should avoid scraping pretty tables when a stable field-oriented output is available.

Shell safety

Prefer readarray/mapfile or carefully quoted variables when consuming command output. SSIDs and connection names may contain spaces or punctuation.

JSON

nmcli itself is not a general JSON API. If your automation needs structured machine-readable data, use terse output carefully or use NetworkManager's D-Bus/libnm interfaces.

Command examples

Pretty output

```
nmcli -p device show enp3s0
```

Improves readability.

Multiline

```
nmcli -m multiline -f all device show enp3s0
```

One property per line.

Terse

```
nmcli -t -f DEVICE,TYPE,STATE device
```

Designed for machine processing.

Custom separator

```
nmcli -t -d '|' -f NAME,UUID,TYPE connection show
```

Choose a separator that does not collide with expected values.

No headings

```
nmcli -t -f NAME connection show
```

Convenient when feeding a loop.

7. Wired Ethernet: Automatic DHCP

The simplest Ethernet profile obtains an address, gateway and DNS settings automatically from DHCP.

DHCP concept

The DHCP server may supply address, prefix, default route, DNS servers and other options.

Device naming

Interface names differ by machine. Always discover the actual interface first.

Command examples

Activate an existing profile

```
nmcli connection up "Wired connection 1"
```

Activates the saved profile on its suitable device.

Create DHCP profile

```
sudo nmcli connection add type ethernet ifname enp3s0 con-name
Lab-Ethernet ipv4.method auto
```

Creates a persistent profile.

Bring it up

```
sudo nmcli connection up Lab-Ethernet
```

Activates the profile.

Inspect IP

```
nmcli device show enp3s0
ip addr show enp3s0
ip route
```

Use nmcli plus standard Linux tools to verify results.

8. Wired Ethernet: Static Ipv4

A static profile stores the IPv4 address, prefix and gateway instead of requesting them through DHCP.

Prefix notation

192.168.10.50/24 means address 192.168.10.50 with a 24-bit network prefix, equivalent to 255.255.255.0.

Gateway

The default gateway is normally the router that provides access beyond the local subnet.

Command examples

Create static profile

```
sudo nmcli con add type ethernet ifname enp3s0 con-name Static-LAN ipv4.method manual ipv4.addresses 192.168.10.50/24 ipv4.gateway 192.168.10.1
```

Creates a basic static IPv4 profile.

Add DNS

```
sudo nmcli con mod Static-LAN ipv4.dns "192.168.10.1 1.1.1.1"
```

DNS values are profile properties.

Activate

```
sudo nmcli con up Static-LAN
```

Applies the complete profile.

Verify

```
nmcli -f ipv4 connection show Static-LAN
ip addr show enp3s0
ip route
```

Verify both profile intent and running state.

9. Modifying Existing Ethernet Profiles

Use 'connection modify' to change a saved profile. Changes normally become active after reactivation.

Command examples

Change address

```
sudo nmcli con mod Static-LAN ipv4.addresses 192.168.10.60/24
```

Edits the profile.

Change gateway

```
sudo nmcli con mod Static-LAN ipv4.gateway 192.168.10.254
```

Changes the default gateway.

Change DNS

```
sudo nmcli con mod Static-LAN ipv4.dns "1.1.1.1 8.8.8.8"
```

Sets DNS servers.

Disable IPv4

```
sudo nmcli con mod Static-LAN ipv4.method disabled
```

Useful for IPv6-only or specialized links.

Reactivate

```
sudo nmcli con down Static-LAN
sudo nmcli con up Static-LAN
```

Applies the modified profile cleanly.

Important: On a remote system, changing the active connection can immediately terminate your SSH session. Plan rollback and console access first.

10. Wired Ethernet: DNS Configuration

NetworkManager can obtain DNS automatically or use servers specified in a connection profile.

DNS is not routing

An IP address can be reachable even when DNS is broken. Test a numeric address and a hostname separately.

Multiple connections

DNS behavior can depend on active profiles and the resolver service. Do not assume /etc/resolv.conf alone tells the whole story.

Command examples

Automatic DNS

```
sudo nmcli con mod Lab-Ethernet ipv4.ignore-auto-dns no
```

Accept DNS supplied by DHCP.

Static DNS

```
sudo nmcli con mod Lab-Ethernet ipv4.dns "192.168.1.1 1.1.1.1"
```

Set DNS servers for the profile.

Search domains

```
sudo nmcli con mod Lab-Ethernet ipv4.dns-search "example.edu"
```

Adds a DNS search domain.

Verify profile

```
nmcli -f ipv4.dns,ipv4.dns-search con show Lab-Ethernet
```

Shows stored DNS properties.

Runtime resolver

```
resolvectl status
```

On systemd-resolved systems, inspect the resolver state separately.

11. Wired Ethernet: Autoconnect and Profile Priority

Autoconnect controls whether NetworkManager automatically activates a profile when a suitable device becomes available.

Command examples

Enable autoconnect

```
sudo nmcli con mod Lab-Ethernet connection.autoconnect yes
```

Profile can activate automatically.

Disable autoconnect

```
sudo nmcli con mod Lab-Ethernet connection.autoconnect no
```

Prevents automatic activation.

Priority

```
sudo nmcli con mod Lab-Ethernet connection.autoconnect-priority
100
```

Higher priority is preferred among eligible profiles.

Inspect

```
nmcli -f name,connection.autoconnect,connection.autoconnect-
priority con show
```

Compare profile behavior.

Manual activation

```
nmcli con up Lab-Ethernet ifname enp3s0
```

Explicitly select the device.

12. Wired Ethernet: MAC, MTU and Link Properties

Some environments require a specific MAC address, MTU or Ethernet link setting. These are advanced profile properties and hardware-dependent.

Command examples

Inspect Ethernet settings

```
nmcli -f 802-3-ethernet con show Lab-Ethernet
```

Displays Ethernet-specific profile properties.

Set cloned MAC

```
sudo nmcli con mod Lab-Ethernet 802-3-ethernet.cloned-mac-
address 02:11:22:33:44:55
```

Requests a specific MAC identity.

Set MTU

```
sudo nmcli con mod Lab-Ethernet 802-3-ethernet.mtu 1500
```

Sets the profile MTU.

Inspect physical link

```
ethtool enp3s0
```

Use ethtool for negotiated speed, duplex and link state.

Important: MAC randomization/cloning and MTU changes can affect DHCP reservations, switches, security controls and tunneling. Test in a controlled lab.

13. Wired Ethernet: Multiple Addresses

A profile can contain more than one IPv4 address. This is useful for servers hosting services on multiple subnets or legacy applications.

Routing caution

Multiple addresses do not automatically mean multiple default gateways. Default-route selection needs deliberate routing configuration.

Primary address

NetworkManager documents address ordering; the first address is the primary address in the setting list.

Command examples

Two addresses at creation

```
sudo nmcli con add type ethernet ifname enp3s0 con-name Multi-IP
ipv4.method manual ipv4.addresses
"192.168.10.20/24,192.168.20.20/24" ipv4.gateway 192.168.10.1
```

Stores multiple addresses in one profile.

Replace address list

```
sudo nmcli con mod Multi-IP ipv4.addresses
"192.168.10.20/24,192.168.20.20/24"
```

Replaces the existing list.

Inspect

```
nmcli -f ipv4.addresses,ipv4.gateway con show Multi-IP
ip addr show enp3s0
```

Verify both saved and active configuration.

14. Wired Ethernet: Static Routes

Static routes tell NetworkManager how to reach specific networks through the connection.

Command examples

Add route

```
sudo nmcli con mod Static-LAN +ipv4.routes "10.20.0.0/16
192.168.10.254"
```

Adds a route to 10.20.0.0/16 through the specified next hop.

Multiple routes

```
sudo nmcli con mod Static-LAN +ipv4.routes "10.30.0.0/16
192.168.10.254,10.40.0.0/16 192.168.10.253"
```

Adds multiple routes.

Inspect profile routes

```
nmcli -f ipv4.routes con show Static-LAN
```

Shows stored routes.

Inspect active routes

```
ip route
```

Shows the kernel routing table.

Important: Route syntax and additional attributes can differ by NetworkManager version. Use 'nm-settings-nmcli' and 'nmcli connection modify --help' for the exact release installed.

15. IPv4 Route Metrics and Default Route Control

When multiple interfaces can reach networks, route metrics help determine preferred paths.

Practical example

Use Ethernet as the preferred Internet path and a secondary connection only for a lab subnet. Give the lab profile a suitable route and avoid installing a competing default route.

Command examples

Connection route metric

```
sudo nmcli con mod Lab-Ethernet ipv4.route-metric 100
```

Sets a route metric for IPv4 routes from the profile.

Avoid default route

```
sudo nmcli con mod Lab-Ethernet ipv4.never-default yes
```

Prevents this profile from becoming the default route.

Allow default route

```
sudo nmcli con mod Lab-Ethernet ipv4.never-default no
```

Allows default-route installation.

Inspect

```
nmcli -f ipv4.never-default,ipv4.route-metric con show Lab-
Ethernet
ip route
```

Compare profile and kernel state.

16. IPv6: Automatic Configuration

IPv6 can use Router Advertisements and DHCPv6 depending on the network. NetworkManager's ipv6.method is commonly 'auto' for automatic configuration.

Command examples

Enable IPv6 auto

```
sudo nmcli con mod Lab-Ethernet ipv6.method auto
```

Uses automatic IPv6 configuration.

Inspect

```
nmcli -f ipv6 con show Lab-Ethernet
ip -6 addr show enp3s0
ip -6 route
```

Check profile and running IPv6 state.

Disable IPv4 but keep IPv6

```
sudo nmcli con mod IPv6-LAN ipv4.method disabled ipv6.method
auto
```

Useful for IPv6-only lab exercises.

17. IPv6: Static Addresses, DNS and Routes

Static IPv6 configuration follows the same profile model as IPv4, but uses IPv6 addresses and routing.

Command examples

Static IPv6

```
sudo nmcli con mod IPv6-LAN ipv6.method manual ipv6.addresses
2001:db8:10::20/64 ipv6.gateway 2001:db8:10::1
```

Uses documentation-prefix addresses for examples.

IPv6 DNS

```
sudo nmcli con mod IPv6-LAN ipv6.dns "2001:4860:4860::8888
2606:4700:4700::1111"
```

Stores IPv6 DNS servers.

IPv6 route

```
sudo nmcli con mod IPv6-LAN +ipv6.routes "2001:db8:20::/64
2001:db8:10::1"
```

Adds a static IPv6 route.

Verify

```
ip -6 addr
ip -6 route
```

Check the active kernel state.

Important: 2001:db8::/32 is reserved for documentation and should not be used for Internet connectivity.

18. Wi-Fi Discovery and Scanning

nmcli can request a Wi-Fi scan and display visible access points. Results depend on the adapter, driver, regulatory domain and radio state.

Hidden SSIDs

A hidden SSID may not appear normally. NetworkManager can still connect when the correct SSID is supplied and the profile is configured appropriately.

Signal

The signal percentage is a convenient display metric; for deeper RF analysis, use iw and driver-specific tools.

Command examples

List APs

```
nmcli device wifi list
```

Displays SSID, mode, channel, rate, signal, bars and security.

Rescan

```
nmcli device wifi rescan
```

Requests a fresh scan.

List a particular device

```
nmcli device wifi list ifname wlp2s0
```

Restricts the scan/listing to a Wi-Fi device.

All fields

```
nmcli -f all device wifi list
```

Useful for discovering additional columns.

19. Wi-Fi: Connect to WPA/WPA2/WPA3-Personal

For ordinary password-protected Wi-Fi, nmcli can create and activate a profile in one command.

Command examples

Connect by SSID

```
nmcli device wifi connect "Campus-WiFi" password
'CorrectHorseBatteryStaple'
```

Creates or activates a Wi-Fi profile.

Prompt for password

```
nmcli --ask device wifi connect "Campus-WiFi"
```

Avoids placing the password in shell history.

Specify device

```
nmcli device wifi connect "Campus-WiFi" password 'example-
password' ifname wlp2s0
```

Selects the Wi-Fi interface.

Inspect profile

```
nmcli connection show "Campus-WiFi"
```

Displays stored settings and connection properties.

Important: Avoid exposing real passwords in scripts, screenshots, shell history or shared training material.

20. Wi-Fi: Open Networks, Hidden SSIDs and BSSID

Different Wi-Fi deployments need different profile properties.

Command examples

Open network

```
sudo nmcli con add type wifi ifname wlp2s0 con-name Open-Lab
ssid Open-Lab
```

Creates an open Wi-Fi profile; security is absent.

Hidden SSID

```
sudo nmcli con add type wifi ifname wlp2s0 con-name Hidden-Lab
ssid Hidden-Lab wifi.hidden yes
```

Marks the SSID as hidden.

Lock to BSSID

```
sudo nmcli con mod Hidden-Lab 802-11-wireless.bssid
AA:BB:CC:DD:EE:FF
```

Restricts association to one AP when supported.

Activate

```
nmcli con up Hidden-Lab
```

Activates the profile.

Important: BSSID locking can prevent roaming and is driver-dependent. It is useful for controlled labs but often undesirable on mobile clients.

21. Wi-Fi: Band, Channel and Power-Management Considerations

Wi-Fi profiles can include radio-related preferences. Exact capabilities depend on NetworkManager and the Wi-Fi driver.

Command examples

Inspect Wi-Fi profile

```
nmcli -f 802-11-wireless con show Campus-WiFi
```

Shows wireless properties.

Prefer 5 GHz band

```
sudo nmcli con mod Campus-WiFi 802-11-wireless.band a
```

Locks the profile to the 5 GHz class represented by 'a' in NetworkManager's setting.

Remove band lock

```
sudo nmcli con mod Campus-WiFi 802-11-wireless.band ''
```

Returns the property to its default/unset state.

Power saving

```
nmcli -f 802-11-wireless.powersave con show Campus-WiFi
```

Inspect the profile's power-save setting; supported values depend on NetworkManager.

Important: Do not teach channel selection as a universal client-side setting. Channel behavior differs between infrastructure mode, ad-hoc mode, drivers and regulatory requirements.

22. Wi-Fi: WPA-Enterprise / 802.1X Concepts

Enterprise Wi-Fi uses 802.1X/EAP rather than a shared personal password. NetworkManager exposes EAP-related settings through the 802-1x setting.

Security warning

Enterprise authentication is policy-sensitive. Certificate validation should not be disabled merely to make a connection work.

EAP methods

PEAP, TTLS, TLS and other methods require different properties and credentials. Use the institution's wireless documentation.

Command examples

Inspect 802.1X settings

```
nmcli -f 802-1x con show Enterprise-WiFi
```

Shows the profile's authentication properties.

PEAP/MSCHAPv2 example

```
sudo nmcli con add type wifi ifname wlp2s0 con-name Corp-WiFi
ssid Corp-WiFi wifi-sec.key-mgmt wpa-eap 802-1x.eap peap 802-
1x.phase2-auth mschapv2 802-1x.identity student@example.edu
```

Illustrative profile construction; add the organization's certificate/credential requirements.

Prompt for secret

```
nmcli --ask con up Corp-WiFi
```

Lets NetworkManager request needed secrets interactively.

23. Wi-Fi: WPA-Enterprise Certificate-Based TLS

EAP-TLS uses client and server certificates and is common in managed institutional networks.

Command examples

Conceptual profile

```
sudo nmcli con add type wifi ifname wlp2s0 con-name EAP-TLS ssid EAP-TLS wifi-sec.key-mgmt wpa-eap 802-1x.eap tls 802-1x.identity user@example.edu
```

This establishes the core profile; certificate/key paths and server identity must be supplied according to the deployment.

Inspect

```
nmcli -f 802-1x con show EAP-TLS
```

Check stored EAP properties.

Activate

```
nmcli con up EAP-TLS
```

NetworkManager performs the EAP exchange.

Important: Never copy private keys or institutional credentials into public course notes. Use placeholder paths and credentials in training documents.

24. Wi-Fi: Disconnect, Reconnect and Forget

Profiles can be deactivated, reactivated, edited or deleted.

Command examples

Disconnect

```
nmcli connection down "Campus-WiFi"
```

Deactivates the profile.

Reconnect

```
nmcli connection up "Campus-WiFi"
```

Reactivates it.

Delete profile

```
sudo nmcli connection delete "Campus-WiFi"
```

Removes the saved profile; it does not physically remove the Wi-Fi adapter.

Delete by UUID

```
sudo nmcli connection delete uuid 11111111-2222-3333-4444-
555555555555
```

Useful when names are duplicated.

Check

```
nmcli connection show
```

Confirm the profile is gone.

25. Wi-Fi: Roaming, Priority and Automatic Connection

A laptop may have several Wi-Fi profiles saved. Autoconnect and priority influence which profile is selected.

Roaming

Do not confuse profile priority with AP roaming. A Wi-Fi profile can normally follow an SSID across compatible APs; BSSID locking changes that behavior.

Multiple adapters

Use ifname on activation when you need to force a profile onto a particular Wi-Fi interface.

Command examples

Enable autoconnect

```
sudo nmcli con mod Campus-WiFi connection.autoconnect yes
```

Prefer campus

```
sudo nmcli con mod Campus-WiFi connection.autoconnect-priority
100
```

Disable guest auto-connect

```
sudo nmcli con mod Guest-WiFi connection.autoconnect no
```

Check

```
nmcli -f NAME,connection.autoconnect,connection.autoconnect-
priority con show
```

26. Wi-Fi: Hotspot / Shared Connection

NetworkManager can create a shared connection so a computer can provide network access to clients through a supported interface. Exact Wi-Fi AP capabilities depend on the hardware and driver.

Command examples

Create shared Wi-Fi

```
nmcli device wifi hotspot ifname wlp2s0 con-name Lab-Hotspot
ssid Lab-Hotspot password 'TrainingPassword123'
```

Creates/activates a hotspot using NetworkManager's shared mode.

Inspect

```
nmcli connection show Lab-Hotspot
```

Review the resulting profile.

Stop

```
nmcli connection down Lab-Hotspot
```

Deactivates the hotspot.

Delete

```
sudo nmcli connection delete Lab-Hotspot
```

Removes the saved profile.

Important: Use strong temporary credentials in real deployments. Shared mode can provide DHCP, DNS forwarding and NAT; it is not equivalent to a transparent bridge.

27. Wi-Fi Security Profiles and Secrets

Wi-Fi credentials are stored as NetworkManager connection secrets according to profile and secret flags. nmcli can prompt for secrets or display properties without necessarily revealing them.

History safety

Prefer '--ask' or a protected passwd-file mechanism over putting passwords directly in a command line.

Permissions

Who can read or modify a profile depends on NetworkManager's permissions and the distribution's configuration.

Command examples

Show security setting

```
nmcli -f 802-11-wireless-security con show Campus-WiFi
```

Ask for missing secrets

```
nmcli --ask con up Campus-WiFi
```

Narrow inspection

```
nmcli -f NAME,TYPE,DEVICE con show
```

28. DHCP Lease and Renewal Troubleshooting

When a DHCP connection fails, inspect both NetworkManager state and the underlying interface.

Command examples

Connection state

```
nmcli device status
nmcli general status
```

Profile method

```
nmcli -f ipv4.method,ipv4.addresses,ipv4.gateway con show Lab-Ethernet
```

Active IP

```
nmcli -f IP4.ADDRESS,IP4.GATEWAY,IP4.DNS device show enp3s0
```

Reapply

```
sudo nmcli device reapply enp3s0
```

Cycle connection

```
sudo nmcli con down Lab-Ethernet
sudo nmcli con up Lab-Ethernet
```

29. Device Reapply, Disconnect and Reconnect

nmcli separates device operations from connection-profile operations.

Key distinction

A device command acts on the interface; a connection command acts on the saved profile or its activation.

Training exercise

Make a DHCP profile, activate it, disconnect the device, then reconnect it and observe the state transitions.

Command examples

Disconnect device

```
sudo nmcli device disconnect enp3s0
```

Connect device

```
sudo nmcli device connect enp3s0
```

Reapply profile

```
sudo nmcli device reapply enp3s0
```

Down profile

```
nmcli con down Lab-Ethernet
```

Up profile

```
nmcli con up Lab-Ethernet ifname enp3s0
```

30. Device Management: Managed/Unmanaged State

NetworkManager may be configured to manage or ignore a device. This matters when troubleshooting an interface that nmcli sees but does not control.

Command examples

Check device

```
nmcli device status
nmcli device show enp3s0
```

Permissions

```
nmcli general permissions
```

Print configuration

```
NetworkManager –print-config
```

Logs

```
journalctl -u NetworkManager -b –no-pager
```

Important: Exact managed/unmanaged policy depends on NetworkManager configuration and distribution integration. Avoid editing configuration files until you understand which component generated them.

31. VLAN Configuration

A VLAN profile creates a tagged 802.1Q interface associated with a parent Ethernet device.

Command examples

Create VLAN

```
sudo nmcli con add type vlan ifname enp3s0.20 dev enp3s0 id 20
con-name VLAN20
```

DHCP on VLAN

```
sudo nmcli con mod VLAN20 ipv4.method auto
```

Static VLAN

```
sudo nmcli con mod VLAN20 ipv4.method manual ipv4.addresses
192.168.20.10/24 ipv4.gateway 192.168.20.1
```

Activate

```
sudo nmcli con up VLAN20
```

Important: The physical switch port must be configured consistently as an access/trunk port. nmcli cannot correct a switch-side VLAN mismatch.

32. Linux Bridges

A bridge connects multiple Layer-2 interfaces. NetworkManager can create and manage bridge devices and bridge ports.

Command examples

Create bridge

```
sudo nmcli con add type bridge ifname br0 con-name br0
```

Add Ethernet port

```
sudo nmcli con add type ethernet ifname enp3s0 master br0 con-
name br0-port-enp3s0
```

DHCP on bridge

```
sudo nmcli con mod br0 ipv4.method auto
```

Activate

```
sudo nmcli con up br0
```

Important: Bridge design is common with virtualization. Moving an active remote server's IP configuration from an Ethernet device to a bridge can interrupt connectivity.

33. Bonding / Link Aggregation

Bonding combines multiple physical links under a logical bond interface. NetworkManager can create bond profiles and bond-port profiles.

Command examples

Create bond

```
sudo nmcli con add type bond ifname bond0 con-name bond0
bond.options "mode=active-backup"
```

Add port 1

```
sudo nmcli con add type ethernet ifname enp3s0 master bond0 con-
name bond0-port1
```

Add port 2

```
sudo nmcli con add type ethernet ifname enp4s0 master bond0 con-
name bond0-port2
```

Static IP

```
sudo nmcli con mod bond0 ipv4.method manual ipv4.addresses
192.168.50.10/24 ipv4.gateway 192.168.50.1
```

Inspect

```
nmcli con show bond0
cat /proc/net/bonding/bond0
```

Important: Bonding modes require compatible switch configuration in some modes. Active-backup is often simpler for redundancy, while other modes may require switch-side aggregation.

34. Teaming, Tunnels and Other Connection Types

NetworkManager supports many connection types beyond ordinary Ethernet and Wi-Fi. Not every type is appropriate for every distribution or release.

Type	Example	Typical purpose
ethernet	802-3-ethernet	Physical LAN
wifi	802-11-wireless	Wireless LAN
vlan	vlan	802.1Q segmentation
bridge	bridge	Layer-2 aggregation
bond	bond	Link redundancy/aggregation
wireguard	wireguard	VPN tunnel
macvlan	macvlan	Virtual L2 interface
vxlan	vxlan	Overlay networking
dummy	dummy	Software-only interface
tun	tun	TUN interface

Discovery

Use 'nmcli connection add --help' and the current NetworkManager reference for the exact syntax supported by your release.

Teaching principle

Master Ethernet/Wi-Fi first; introduce virtual networking only after students understand addresses, routes, gateways and interfaces.

35. WireGuard Profiles

Modern NetworkManager versions can manage WireGuard profiles. WireGuard configuration is sensitive and should be built from trusted keys and endpoint information.

Command examples

Check support

```
nmcli connection add type wireguard –help
```

Create profile

```
sudo nmcli con add type wireguard ifname wg0 con-name wg0
```

Inspect

```
nmcli -f wireguard con show wg0
```

IP configuration

```
sudo nmcli con mod wg0 ipv4.method manual ipv4.addresses
10.200.0.2/24
```

Important: Do not put real private keys in classroom documents. Use placeholders and obtain the exact peer configuration from the VPN administrator.

36. Connection Clone, Import, Export and Migration

NetworkManager can clone profiles and, depending on profile type and version, import/export connection configurations.

Command examples

Clone

```
nmcli connection clone "Lab-Ethernet" "Lab-Ethernet-Clone"
```

List profiles

```
nmcli connection show
```

Reload profiles

```
sudo nmcli connection reload
```

Load a profile

```
sudo nmcli connection load /path/to/profile.nmconnection
```

Import help

```
nmcli connection import –help
```

Important: Export formats and supported VPN types are version/plugin dependent. Always consult the installed nmcli help before automating imports.

37. Connection Editing with nmcli con edit

Besides one-shot 'connection modify' commands, nmcli has an interactive connection editor. It is useful for exploring settings and properties.

Why teach the editor ?

It helps students discover properties without memorizing every property name and is useful for diagnosing profile structure.

Why not rely on it alone?

For scripts and repeatable labs, explicit nmcli commands are easier to document and automate.

Command examples

Open editor

```
nmcli connection edit "Lab-Ethernet"
```

Create new editor session

```
nmcli connection edit
```

Show settings

```
print
```

Describe a property

```
describe ipv4.addresses
```

Save and quit

```
save
quit
```

38. Connection Profile Inspection in Depth

The connection profile is a database-like object with settings such as connection, ethernet, wifi, wifi-security, ipv4, ipv6 and 802-1x.

Command examples

Everything

```
nmcli -p -m multiline -f all con show Lab-Ethernet
```

IPv4 only

```
nmcli -f ipv4 con show Lab-Ethernet
```

IPv6 only

```
nmcli -f ipv6 con show Lab-Ethernet
```

Wi-Fi security

```
nmcli -f 802-11-wireless,802-11-wireless-security con show
Campus-WiFi
```

Important: Some secret values may be hidden or require privileges. Do not treat redacted output as evidence that no secret exists.

39. Common ‘nmcli’ Errors and Their Meaning

The fastest troubleshooting improvement is learning to classify errors rather than repeatedly retrying commands.

Symptom	Likely category	First checks
NetworkManager not running	Service	systemctl status NetworkManager
Device unavailable	Hardware/driver/radio	nmcli device; rfkill; dmesg
No suitable connection	Profile eligibility	nmcli con show; autoconnect; ifname
Wi-Fi SSID absent	RF/scan/regulatory	nmcli dev wifi rescan; rfkill; iw
DHCP timeout	L2/DHCP/VLAN	link, switch, VLAN, DHCP server
Can ping IP but not hostname	DNS	resolvectl status; nmcli device show
Connected but no Internet	Route/firewall/upstream	ip route; ping gateway; connectivity
SSH lost after change	Expected consequence	Console/out-of-band recovery

First principle

Determine whether the failure is device, link, authentication, IP configuration, routing, DNS or application-level connectivity.

40. Logs and Troubleshooting Workflow

NetworkManager logs often reveal why a profile failed to activate.

Recommended sequence

1) radio/device → 2) profile → 3) activation → 4) address → 5) route → 6) DNS →

7) application.

Evidence

Capture command output before changing settings so you can compare the original and modified states.

Command examples

Service log

```
journalctl -u NetworkManager -b –no-pager
```

Follow logs

```
journalctl -u NetworkManager -f
```

Recent warnings

```
journalctl -u NetworkManager -b -p warning –no-pager
```

Kernel messages

```
dmesg | grep -Ei 'firmware|wifi|wlan|eth|network'
```

Network state

```
nmcli general status
nmcli device status
nmcli connection show --active
```

41. A Complete Wired Troubleshooting Lab

This lab gives students a repeatable procedure for diagnosing an Ethernet connection.

Step 1 - inventory

- nmcli device status
- nmcli device show enp3s0
- ethtool enp3s0

Step 2 - profile

- nmcli connection show
- nmcli -f ipv4,ipv6 connection show Lab-Ethernet

Step 3 - activation

- nmcli connection up Lab-Ethernet ifname enp3s0

Step 4 - addressing

- ip addr show enp3s0
- ip route
- resolvectl status

Step 5 - tests

- ping -c 4 <gateway>
- ping -c 4 1.1.1.1
- ping -c 4 example.com

Step 6 - logs

- journalctl -u NetworkManager -b –no-pager

Important: Replace placeholders with addresses appropriate to the lab network. Do not use an address that conflicts with another host.

42. A Complete Wireless Troubleshooting Lab

This lab separates Wi-Fi radio/link problems from IP-layer problems.

Step 1 - hardware

- nmcli device status
- lspci -nnk | grep -A3 -i network
- lsusb

Step 2 - radio

- nmcli radio wifi
- rfkill list

Step 3 - scan

- nmcli device wifi rescan
- nmcli device wifi list

Step 4 - connect

- nmcli --ask device wifi connect "Lab-WiFi"

Step 5 - verify link

- nmcli device status
- nmcli device show wlp2s0

Step 6 - verify IP

- ip addr show wlp2s0
- ip route

Step 7 - verify DNS

- resolvectl status
- getent hosts example.com

Step 8 - logs

- journalctl -u NetworkManager -b --no-pager
- dmesg | grep -i firmware

43. Scripting nmcli Safely

nmcli is well suited to provisioning and classroom automation, but scripts must handle quoting, failures, credentials and remote-session risk.

Command examples

Strict shell starter

```
set -euo pipefail
nmcli -t -f DEVICE,STATE device
```

Check NetworkManager

```
if ! nmcli -t -f RUNNING general | grep -qx running; then
  echo 'NetworkManager is not running' >&2
  exit 1
fi
```

Check a profile

```
if nmcli -t -f NAME con show | grep -Fxq 'Lab-Ethernet'; then
  echo 'Profile exists'
fi
```

Activate

```
nmcli con up 'Lab-Ethernet' ifname enp3s0
```

Important: Never assume nmcli output formatting is unchanged forever. Pin versions in automated environments and use documented terse fields.

44. Passwords, Secrets and Automation

Network credentials are the most common reason a technically correct nmcli script becomes a security problem.

Avoid

- Hard-coding Wi-Fi passwords in scripts committed to Git.
- Putting secrets into shared classroom screenshots.
- Passing secrets on command lines when shell history or process inspection may expose them.

Prefer

- ınteractive '--ask' during manual setup.
- Protected secret storage and NetworkManager's secret handling.
- A controlled password file where supported by the exact nmcli operation.
- Short-lived lab credentials that are rotated after training.

Command examples

Interactive Wi-Fi

```
nmcli --ask device wifi connect "Training-WiFi"
```

Narrow inspection

```
nmcli -f NAME,TYPE,DEVICE con show
```

45. Remote Server Safety and Recovery

Network changes on remote systems are uniquely risky because the command can destroy the management path.

Before changing a remote server

- Confirm console/out-of-band access.
- Record the current IP, gateway, DNS and active profile.
- Know the rollback command before making the change.
- Prefer staged changes and a maintenance window.

Capture baseline

nmcli device status

nmcli con show --active

ip addr

ip route

resolvectl status

Rollback principle

Keep the old profile until the new path is confirmed. Do not delete the only known-good profile first.

Command examples

Reactivate known-good profile

```
nmcli con show
nmcli con up 'Known-Good-Profile'
```

46. Advanced Profile Properties Worth Knowing

The full NetworkManager settings reference is extensive. These properties are especially useful for advanced administration.

Setting	Examples of useful properties
connection	id, uuid, type, interface-name, autoconnect, autoconnect-priority, metered
802-3-ethernet	mac-address, cloned-mac-address, mtu
802-11-wireless	ssid, mode, band, bssid, hidden, powersave
802-11-wireless-security	key-mgmt, auth-alg, psk
802-1x	eap, identity, phase2-auth, ca-cert, client-cert
ipv4	method, addresses, gateway, dns, dns-search, routes, route-metric, never-default
ipv6	method, addresses, gateway, dns, routes, route-metric, never-default

Discover rather than memorize

Use 'nmcli connection show <profile>', 'nmcli connection edit', and the NetworkManager settings reference to discover valid properties for the installed release.

47. Capstone Exercise: Build a Dual Wired / Wireless Lab

This capstone asks students to configure a machine with Ethernet for the lab LAN and Wi-Fi for a separate network.

Requirements

- Ethernet: DHCP profile named Lab-Ethernet.
- Wi-Fi: connect to Training-WiFi using an interactive password prompt.
- Ethernet should have a static alternative profile named Server-LAN.
- Wi-Fi should not automatically connect to an unwanted guest SSID.
- Document the final addresses, routes, DNS and active profiles.

Suggested commands

- nmcli device status
- nmcli device wifi list
- sudo nmcli con add type ethernet ifname enp3s0 con-name Lab-Ethernet ipv4.method auto
- nmcli --ask device wifi connect Training-WiFi
- sudo nmcli con mod Guest-WiFi connection.autoconnect no
- nmcli con show --active
- ip route
- resolvectl status

Assessment

Students should explain which component is responsible for device state, Wi-Fi association, IP assignment, routing and DNS.

48. ‘nmcli’ Command Cheat Sheet

Task	Command
NetworkManager status	nmcli general status
Connectivity	nmcli networking connectivity
Wi-Fi radio	nmcli radio wifi
Device list	nmcli device status
Device details	nmcli device show IFACE
Profiles	nmcli connection show
Active profiles	nmcli connection show --active
Wi-Fi scan	nmcli device wifi list
Wi-Fi rescan	nmcli device wifi rescan
Wi-Fi connect	nmcli --ask device wifi connect SSID
Connection up	nmcli con up PROFILE
Connection down	nmcli con down PROFILE
Create Ethernet DHCP	nmcli con add type ethernet ifname IFACE con-name NAME ipv4.method auto
Static IPv4	nmcli con mod NAME ipv4.method manual ipv4.addresses A.B.C.D/24 ipv4.gateway A.B.C.1
DNS	nmcli con mod NAME ipv4.dns '1.1.1.1 8.8.8.8'
Static route	+ipv4.routes 'NET/PREFIX NEXT-HOP'
Autoconnect	nmcli con mod NAME connection.autoconnect yes
Delete profile	sudo nmcli con delete NAME
Reload profiles	sudo nmcli con reload
Monitor	nmcli monitor

Important:

Treat this page as a quick reference, not a replacement for the installed version's help and manual pages.

49. Reference, Further Reading and Final Checklist

The authoritative NetworkManager documentation should be the final reference when a property or syntax differs between releases.

Official references

- NetworkManager **nmcli** reference manual:
  https://networkmanager.dev/docs/api/latest/nmcli.html
- NetworkManager **settings** reference:
  https://www.networkmanager.dev/docs/api/latest/nm-settings-nmcli.html
- NetworkManager **manual** pages:
  https://www.networkmanager.dev/docs/man-pages/
- NetworkManager **nmcli** examples:
  https://www.networkmanager.dev/docs/api/1.34/nmcli-examples.html

---

Final student checklist

- I can identify NetworkManager, devices, profiles and active connections.
- I can configure DHCP and static IPv4 on Ethernet.
- I can configure DNS and routes.
- I can inspect and modify IPv6 settings.
- I can scan for Wi-Fi and connect securely.
- I understand Wi-Fi personal versus enterprise authentication at a conceptual level.
- I can use nmcli output fields in scripts.
- I can troubleshoot device, link, IP, route and DNS failures separately.
- I understand the danger of changing an active remote connection.

Core principle

Learn the model, not just the commands:

DEVICE → PROFILE → ACTIVATION → ADDRESS → ROUTE → DNS → APPLICATION.

Important: This guide intentionally uses documentation-safe example addresses, placeholder SSIDs and placeholder credentials. Adapt commands to the actual interface names and network design of the laboratory.