



### Show Wi-Fi iface & Status

<b><code>nmcli connection show --active</code></b>	(Note: two dashes) Simple, human readable
<b><code>nmcli device show &lt;iface&gt;</code></b>	All details (IP, gateway, DNS, etc.)
<b><code>iw dev &lt;iface&gt; link</code></b> Example: <code>iw dev wlan0 link</code>	Signal, frequency, TX rate, etc.
<b><code>iwconfig</code></b> Example: <code>iwconfig</code> Example: <code>iwconfig &lt;iface&gt;</code>	Legacy method to quickly see the current status. If no iface is defined, then all ifaces will be displayed.
<b><code>ip link dev &lt;iface&gt;</code></b> Example: <code>ip link dev wlan0</code>	Displays detailed information about the named network iface
<b><code>ip link show &lt;iface&gt;</code></b> Example: <code>ip link show wlan0</code>	
<b><code>sudo rfkill list</code></b>	Shows whether Wi-Fi/Bluetooth are hardware or software blocked

### Scan for Wi-Fi Networks

<b><code>nmcli dev wifi list</code></b>	Lists available Wi-Fi networks that is colour coded for easier viewing.
<b><code>nmcli dev wifi list</code></b>	Forces NetworkManager to rescan Wi-Fi
<b><code>iw &lt;iface&gt; scan</code></b> Example: <code>iw wlan0 scan</code>	Full raw scan of nearby Wi-Fi networks.
<b><code>iw wlan0 scan   grep -e SSID -e signal -e freq</code></b>	Filters scan output for readability

### Checking regulatory domain

<b><code>iw reg get</code></b>	Lists what regulatory region is set
<b><code>sudo iw reg set &lt;regularory domain&gt;</code></b> Example: <code>sudo iw reg set CA</code> CA = Canada	Changes the regulatory domain

### Turn Wi-Fi radio on/off

<b><code>sudo nmcli radio wifi off/on</code></b>	Turns off or on all Wi-Fi ifaces (Airplane mode)
<b><code>nmcli device disconnect &lt;iface&gt;</code></b>	Only disconnects specific iface
<b><code>nmcli device connect &lt;iface&gt;</code></b>	Connects/reconnects specific iface
<b><code>sudo ip link set &lt;iface&gt; down/up</code></b> Example: <code>sudo ip link set wlan0 down. sudo ip link set wlan0 up</code>	Full raw scan of nearby Wi-Fi networks.
<b><code>sudo rfkill block wifi</code></b>	Soft-blocks (disables) all Wi-Fi radios at the kernel level.
<b><code>sudo rfkill unblock wifi</code></b>	Soft-unblocks (enables) all Wi-Fi radios at the kernel level.
<b><code>sudo rfkill block all</code></b>	Soft-blocks all radios (Wi-Fi + Bluetooth).
<b><code>sudo rfkill unblock all</code></b>	Soft-unblocks all radios. (Wi-Fi + Bluetooth)

### Monitor Mode (analysis)

<b><u>Setting into monitor mode:</u></b> <b><code>sudo ip link set &lt;iface&gt; down</code></b> <b><code>sudo iw &lt;iface&gt; set monitor</code></b> <b><code>sudo ip link set &lt;iface&gt; up</code></b>	
<b><u>Returning to managed mode:</u></b> <b><code>sudo ip link set &lt;iface&gt; down</code></b> <b><code>sudo iw &lt;iface&gt; set type managed</code></b> <b><code>sudo ip link set &lt;iface&gt; up</code></b>	
<b><u>Setting into monitor mode:</u></b> <b><code>sudo airmon-ng start &lt;iface&gt;</code></b>	Must have aircrack-ng installed. airmon-ng will change the interface name to <iface>mon. When returning to managed mode, the <iface> needs to be the mon interface. Example: wlan0mon
<b><u>Returning to managed mode:</u></b> <b><code>Sudo airmon-ng stop &lt;iface&gt;</code></b>	
<b><u>Setting into monitor mode:</u></b> <b><code>sudo ip link set &lt;iface&gt; down</code></b> <b><code>sudo iw config &lt;iface&gt; mode monitor</code></b> <b><code>sudo ip link set &lt;iface&gt; up</code></b>	
<b><u>Returning to managed mode:</u></b> <b><code>sudo ip link set &lt;iface&gt; down</code></b> <b><code>sudo iw &lt;iface&gt; set type managed</code></b> <b><code>sudo ip link set &lt;iface&gt; up</code></b>	



### Connect to a personal network (WPA2 and above)

```
nmcli device wifi connect "SSID_Name" password "your_password" iface <iface>
```

Broadcasted SSID

```
nmcli device wifi connect "SSID" password "pass" hidden yes
```

Hidden SSID

### Connect to an enterprise network (WPA2 and above)

```
nmcli device wifi connect "SSID" password "yourpassword" wifi-sec.key-mgmt  
wpa-eap 802-1x.eap peap 802-1x.phase2-auth mschapv2 802-1x.identity  
"yourusername@domain.com"
```

PEAP + MSCHAPV2

```
nmcli device wifi connect "SSID" wifi-sec.key-mgmt wpa-eap 802-1x.eap ttls pap  
802-1x.identity "username" 802-1x.password "password"
```

TTLS + PAP

```
nmcli device wifi connect "SSID" wifi-sec.key-mgmt wpa-eap 802-1x.eap tls 802-  
1x.identity "username" 802-1x.client-cert ~/client.crt 802-1x.private-key  
~/client.key 802-1x.private-key-password "keypass"
```

TLS (Client Certificate)

```
nmcli dev wifi con "eduroam" wifi-sec.key-mgmt wpa-eap 802-1x.eap peap 802-  
1x.phase2-auth mschapv2 802-1x.identity "user@university.edu" 802-  
1x.anonymous-identity "anonymous@university.edu" 802-1x.password  
"yourpassword"
```

PEAP with anonymous identity. Example: eduroam.

### Disconnect or forget a network

```
nmcli device disconnect <iface>
```

Disconnects the Wi-Fi interface from whatever network it's connected to.

```
nmcli connection down id "SSID_NAME"
```

Disconnects only that specific connection, without affecting others.

```
nmcli connection | grep wifi | awk '{print $1}' | xargs -n 1 nmcli connection delete  
id
```

Deletes all saved Wi-Fi connections. Use carefully — it's irreversible via CLI

#### Notes:

- **Disconnect vs Forget:**
- disconnect → stops current connection, but profile is saved.
- delete / forget → removes saved credentials so the system won't auto-connect.
- If you have **multiple interfaces**, specify the interface for disconnecting:
- For **Enterprise networks**, forgetting the profile also removes certificates or saved credentials stored by NetworkManager.