



Homework 4: Basic Network Management

Môn học: Hệ điều hành Linux và Ứng dụng

CS11117 - 22MMT

Sinh viên: Nguyễn Hồ Đăng Duy - 22127085

Table of Contents

- Step 1: Network Interfaces & IP Addresses
 - Step 2: DNS Configuration
 - Step 3: Hostname Setup
 - Step 4: Check Open Ports and Services
 - Step 5: Test Connectivity
 - Step 6: Test Web Requests
 - Step 7: Use nmcli to View Connections
 - Step 8: Modify a Connection via nmcli
 - Step 9: Use nmtui for Configuration
 - Step 10: Enable/Disable Network Connection
-

Note:

This exercise was performed on two machines: Ubuntu 24.04 and a Kali Linux virtual machine on Windows VirtualBox, so the screenshots may differ slightly in some tasks.

Step 1: Network Interfaces & IP Addresses

1. Display all network interfaces and their IP addresses

```
ip a
```

Explanation:

- `ip a` (short for `ip address`) shows all network interfaces along with their current IP address configurations.

Screenshot:

```
o > ~/Documents/CS11117_Linux > P main
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp1s0f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether c4:c6:e6:df:c8:84 brd ff:ff:ff:ff:ff:ff
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 8c:3b:4a:59:89:de brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.236/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp2s0
            valid_lft 213sec preferred_lft 213sec
        inet6 fe80::5246:d08e:1524:ec39/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
4: br-9cfcae9cd9fe: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 2e:38:ab:f0:1b:e6 brd ff:ff:ff:ff:ff:ff
        inet 172.21.0.1/16 brd 172.21.255.255 scope global br-9cfcae9cd9fe
            valid_lft forever preferred_lft forever
        inet6 fe80::2c38:abff:fef0:1be6/64 scope link
            valid_lft forever preferred_lft forever
5: br-b7f909555d4c: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 1a:01:1b:a9:fb:96 brd ff:ff:ff:ff:ff:ff
        inet 172.20.0.1/16 brd 172.20.255.255 scope global br-b7f909555d4c
            valid_lft forever preferred_lft forever
        inet6 fe80::1801:1bff:fea9:fb96/64 scope link
            valid_lft forever preferred_lft forever
6: br-b86997ec4fd3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 36:ac:4f:cf:3a:63 brd ff:ff:ff:ff:ff:ff
        inet 172.28.0.1/16 brd 172.28.255.255 scope global br-b86997ec4fd3
            valid_lft forever preferred_lft forever
7: br-d63615e004f9: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 4a:2e:54:97:b6:4a brd ff:ff:ff:ff:ff:ff
        inet 172.22.0.1/16 brd 172.22.255.255 scope global br-d63615e004f9
            valid_lft forever preferred_lft forever
```

2. Identify the system's main IP address and interface name

```
ip route | grep default
```

Explanation:

- This command shows the default route, including the primary network interface and gateway used for internet access.

Screenshot:

```
o > ~/Documents/CS11117_Linux > P main !1
ip route | grep default
default via 192.168.1.1 dev wlp2s0 proto dhcp src 192.168.1.236 metric 600
```

Step 2: DNS Configuration**1. View DNS configuration file**

```
cat /etc/resolv.conf
```

Explanation:

- Shows current DNS server settings. It's dynamically updated by your network manager.

Screenshot:

```
(denver㉿kali)-[~]
$ cat /etc/resolv.conf
# Generated by NetworkManager

(denver㉿kali)-[~]
$
```

2. Temporarily set DNS to Google's 8.8.8.8

```
sudo nano /etc/resolv.conf
```

Then add line:

```
nameserver 8.8.8.8
```

Explanation:

- Manually sets a DNS server (temporary until reboot or DHCP renewal).

Screenshot:

```
File Actions Edit View Help
GNU nano 8.3
# Generated by NetworkManager
nameserver 8.8.8.8
```

3. Test DNS resolution

```
dig google.com
# or
nslookup google.com
```

Explanation:

- Checks if DNS resolution is working by translating a domain name to an IP address.

Screenshot:

```
(denver㉿kali)-[~]
$ dig google.com
;; UDP setup with 8.8.8.8#53(8.8.8.8) for google.com failed: network unreachable.
;; no servers could be reached
;; UDP setup with 8.8.8.8#53(8.8.8.8) for google.com failed: network unreachable.
;; no servers could be reached
;; UDP setup with 8.8.8.8#53(8.8.8.8) for google.com failed: network unreachable.
;; no servers could be reached

(denver㉿kali)-[~]
$ nslookup google.com
;; UDP setup with 8.8.8.8#53(8.8.8.8) for google.com failed: network unreachable.
;; no servers could be reached
;; UDP setup with 8.8.8.8#53(8.8.8.8) for google.com failed: network unreachable.
;; no servers could be reached
;; UDP setup with 8.8.8.8#53(8.8.8.8) for google.com failed: network unreachable.
;; no servers could be reached

(denver㉿kali)-[~]
$ 
```

Step 3: Hostname Setup

1. Set hostname to **mystudent-pc**

```
sudo hostnamectl set-hostname mystudent-pc
```

Explanation:

- This sets a permanent system hostname using **systemd**.

Screenshot:

```
(denver㉿kali)-[~]
$ hostname
kali

(denver㉿kali)-[~]
$ sudo hostnamectl set-hostname mystudent-pc

(denver㉿kali)-[~]
$ 
```

2. Confirm new hostname

```
hostname
```

Explanation:

- Displays the current hostname to verify the change was applied.

Screenshot:

The screenshot shows a terminal session on a Kali Linux system. It consists of four lines of text:

- Line 1: The user runs the command `hostname`, which outputs the current hostname: `denver`.
- Line 2: The user runs the command `sudo hostnamectl set-hostname mystudent-pc` to change the hostname.
- Line 3: The user runs the command `hostname` again to verify the change, which now outputs `mystudent-pc`.
- Line 4: The user exits the terminal.

Step 4: Check Open Ports and Services

1. Show listening ports

```
ss -tuln
```

Explanation:

- Lists all TCP/UDP listening ports without resolving service names.

Screenshot:

The screenshot shows the output of the `ss -tuln` command. The table lists various network connections, including listening ports for both TCP and UDP protocols. The columns are:

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	127.0.0.54:53	0.0.0.0:*	
udp	UNCONN	0	0	127.0.0.53x1:53	0.0.0.0:*	
udp	UNCONN	0	0	0.0.0.0:53632	0.0.0.0:*	
udp	UNCONN	0	0	0.0.0.0:5353	0.0.0.0:*	
udp	UNCONN	0	0	0.0.0.0:40659	0.0.0.0:*	
udp	UNCONN	0	0	[::]:54210	[::]:*	
udp	UNCONN	0	0	[::]:5353	[::]:*	
tcp	LISTEN	0	4096	127.0.0.54:53	0.0.0.0:*	
tcp	LISTEN	0	4096	127.0.0.53x1:53	0.0.0.0:*	
tcp	LISTEN	0	4096	0.0.0.0:13337	0.0.0.0:*	
tcp	LISTEN	0	4096	127.0.0.1:631	0.0.0.0:*	
tcp	LISTEN	0	100	0.0.0.0:25	0.0.0.0:*	
tcp	LISTEN	0	4096	0.0.0.0:5005	0.0.0.0:*	
tcp	LISTEN	0	4096	0.0.0.0:8092	0.0.0.0:*	
tcp	LISTEN	0	4096	0.0.0.0:8090	0.0.0.0:*	
tcp	LISTEN	0	4096	[::]:13337	[::]:*	
tcp	LISTEN	0	511	*:80	*:*	
tcp	LISTEN	0	100	[::]:25	[::]:*	
tcp	LISTEN	0	4096	[::]:5005	[::]:*	
tcp	LISTEN	0	4096	[::]:631	[::]:*	
tcp	LISTEN	0	4096	[::]:8092	[::]:*	
tcp	LISTEN	0	4096	[::]:8090	[::]:*	

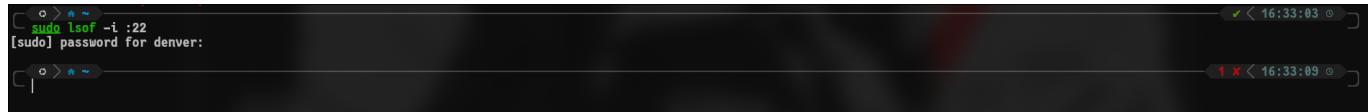
2. Identify which service is using port 22

```
sudo lsof -i :22
```

Explanation:

- Finds the process using port 22

Screenshot:



```
sudo lsof -i :22
```

Step 5: Test Connectivity

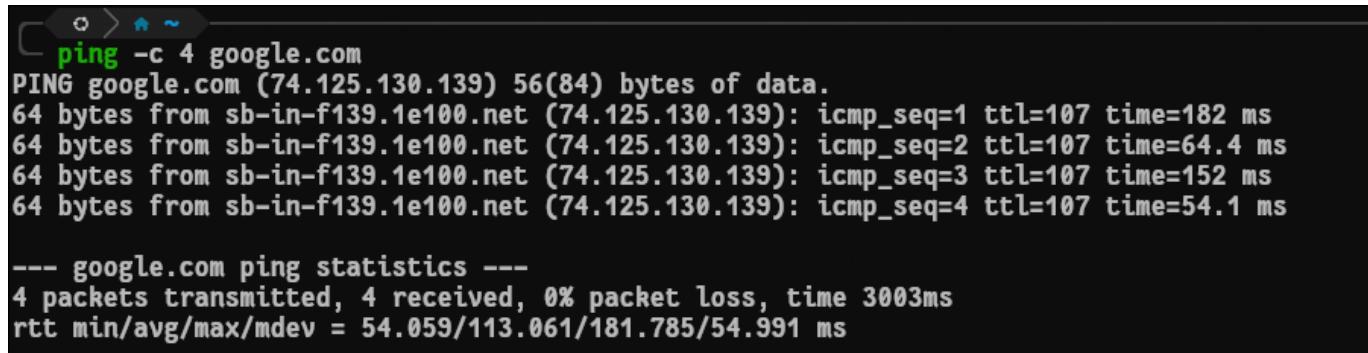
1. Ping to check if reachable

```
ping -c 4 google.com
```

Explanation:

- Sends 4 ICMP packets to test if Google is reachable.

Screenshot:



```
ping -c 4 google.com
PING google.com (74.125.130.139) 56(84) bytes of data.
64 bytes from sb-in-f139.1e100.net (74.125.130.139): icmp_seq=1 ttl=107 time=182 ms
64 bytes from sb-in-f139.1e100.net (74.125.130.139): icmp_seq=2 ttl=107 time=64.4 ms
64 bytes from sb-in-f139.1e100.net (74.125.130.139): icmp_seq=3 ttl=107 time=152 ms
64 bytes from sb-in-f139.1e100.net (74.125.130.139): icmp_seq=4 ttl=107 time=54.1 ms
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 54.059/113.061/181.785/54.991 ms
```

2. Trace route to Google

```
traceroute google.com
```

Explanation:

- Displays the path (hops) that packets take to reach Google.

Screenshot:

```
traceroute google.com
traceroute to google.com (74.125.130.100), 30 hops max, 60 byte packets
 1 _gateway (192.168.1.1)  17.341 ms  17.260 ms  21.890 ms
 2 125.235.249.189.adsl.viettel.vn (125.235.249.189)  22.633 ms  44.855 ms  44.778 ms
 3 10.255.38.205 (10.255.38.205)  44.737 ms  44.703 ms  10.255.38.199 (10.255.38.199)  44.662 ms
 4 localhost (27.68.237.186)  44.629 ms  44.596 ms localhost (27.68.237.130)  44.564 ms
 5 * localhost (27.68.237.139)  44.605 ms *
 6 localhost (27.68.250.170)  44.542 ms localhost (27.68.250.208)  2092.099 ms  2092.033 ms
 7 * 142.250.172.204 (142.250.172.204)  150.213 ms *
 8 * * *
 9 142.251.52.48 (142.251.52.48)  150.774 ms 142.251.49.190 (142.251.49.190)  150.742 ms 209.85.245.50 (209.85.245.50)  149.984 ms
10 192.178.109.208 (192.178.109.208)  149.951 ms  149.916 ms 142.250.60.240 (142.250.60.240)  149.879 ms
11 142.251.230.213 (142.251.230.213)  150.574 ms 72.14.235.60 (72.14.235.60)  150.496 ms 66.249.95.248 (66.249.95.248)  100.333 ms
12 142.251.230.230 (142.251.230.230)  100.261 ms 74.125.252.254 (74.125.252.254)  100.792 ms 142.251.230.236 (142.251.230.236)  100.749 ms
13 216.239.35.147 (216.239.35.147)  100.718 ms 216.239.35.173 (216.239.35.173)  100.687 ms 216.239.35.165 (216.239.35.165)  100.657 ms
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * sb-in-f100.1e100.net (74.125.130.100)  125.939 ms *
```

Step 6: Test Web Requests

1. Download homepage using curl or wget

```
curl http://example.com
# or
wget http://example.com
```

Screenshot:

```

[ ~ > ~/Documents/CS11117_Linux > = P main !2 ?5
curl http://example.com
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
body {
    background-color: #f0f0f2;
    margin: 0;
    padding: 0;
    font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}

div {
    width: 600px;
    margin: 5em auto;
    padding: 2em;
    background-color: #fdfdff;
    border-radius: 0.5em;
    box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
}

a:link, a:visited {
    color: #38488f;
    text-decoration: none;
}
@media (max-width: 700px) {
    div {
        margin: 0 auto;
        width: auto;
    }
}
</style>
</head>

<body>
<div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.</p>
    <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>

```

```

[ ~ > ~/Documents/CS11117_Linux > = P main !2 ?5
wget http://example.com
--2025-06-15 16:40:33-- http://example.com/
Resolving example.com (example.com)... 23.215.0.136, 96.7.128.175, 23.192.228.80, ...
Connecting to example.com (example.com)|23.215.0.136|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1256 (1.2K) [text/html]
Saving to: 'index.html'

index.html          100%[=====] 1.23K --.-KB/s   in 0s
2025-06-15 16:40:34 (59.0 MB/s) - 'index.html' saved [1256/1256]

```

2. Explain the difference between `curl` and `wget`

Feature	<code>curl</code>	<code>wget</code>
Main Purpose	Transfers data to or from a server using protocols	Downloads files from the web (HTTP, HTTPS, FTP, etc.)
Default Behavior	Prints output to the terminal (<code>stdout</code>)	Automatically saves the file to disk
POST/Upload Support	Strong support (POST, PUT, custom headers, etc.)	Limited; mostly used for downloading files (GET only)
Recursive Download	✗ Not supported	<input checked="" type="checkbox"/> Supported (can download entire websites recursively)
Output Formatting	Highly customizable using flags (<code>-o</code> , <code>-</code> , <code>H</code> , <code>-i</code>)	Less customizable, but straightforward for downloading

Step 7: Use nmcli to View Connections

1. Show all network connections

```
nmcli connection show
```

Explanation:

- Lists all saved network connections.

Screenshot:

NAME	UUID	TYPE	DEVICE
Yourvibes Thinking Lab	295c3ccc-6c1f-4941-95c9-5a26818a358f	wifi	wlp2s0
br-86af80836b48	637bfe95-b934-42c7-9d42-098367710305	bridge	br-86af80836b48
br-9cfcae9cd9fe	5112887a-e58c-4cde-9da4-5909e6602a07	bridge	br-9cfcae9cd9fe
br-b3f7bcf268a8	e8486eaf-e7a5-4277-a618-79bef520f8fa	bridge	br-b3f7bcf268a8
br-b7f909555d4c	1bb77a43-a39c-4ab2-8e10-c3f72d55a3c0	bridge	br-b7f909555d4c
br-fa9853f017bc	c85056eb-5fe9-4f95-92bd-89ed1b8a8bab	bridge	br-fa9853f017bc
lo	7aea1653-bc35-4fb3-861d-b88ac42ca47f	loopback	lo
br-0b31e96f66e1	2a14a53f-6e4f-4147-bb7a-54ae55eba14c	bridge	br-0b31e96f66e1
br-226917c26716	d586a2de-bb6d-499f-b84d-34c1b769be4b	bridge	br-226917c26716
br-4dcf6c5165bb	375bfe4b-a256-4923-a11f-7a5ac2e26d10	bridge	br-4dcf6c5165bb
br-6afd25ff3b16	9426d138-9016-486a-b086-3f3c1ad0e96e	bridge	br-6afd25ff3b16
br-7650805b7c9a	66e1f1ab-a38c-42b7-878f-45171b0e9732	bridge	br-7650805b7c9a
br-b86997ec4fd3	70f324b9-eb26-46b7-81d2-34089fb3feed	bridge	br-b86997ec4fd3
br-c2c7a59f534a	ef09e36c-35ae-43c8-9425-5cb4c1308b7c	bridge	br-c2c7a59f534a
br-d63615e004f9	a92f117b-ecfb-4e07-85f6-0274aac36f7f	bridge	br-d63615e004f9
docker0	4e36f19f-6722-43be-8e8e-b1833c3ba95b	bridge	docker0
Duy_5G	9c33951d-b126-43a4-9a26-ec5b1a8da6de	wifi	--
HCMUS-Phonghoc	838950c4-294a-4fc4-8fdbd-3f16f248b89b	wifi	--
HCMUS-Public	201c5d3d-fdba-4122-8c4f-8f919ffee8fb	wifi	--
thuyencafe	186938af-d3b4-441d-81bc-43326d518968	wifi	--
Wired connection 1	62f3372d-d32b-389f-9be2-b148aa7b09a6	ethernet	--
Yourvibes Thinking Lab 5G	51b8395b-87e1-42e6-9a32-5a0d2d5b49c3	wifi	--

2. Show currently active connection

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

Explanation:

- Displays which connections are currently active.

Screenshot:

NAME	UUID	TYPE	DEVICE
Yourvibes Thinking Lab	295c3ccc-6c1f-4941-95c9-5a26818a358f	wifi	wlp2s0
br-86af80836b48	637bfe95-b934-42c7-9d42-098367710305	bridge	br-86af80836b48
br-9cfcae9cd9fe	5112887a-e58c-4cde-9da4-5909e6602a07	bridge	br-9cfcae9cd9fe
br-b3f7bcf268a8	e8486eaf-e7a5-4277-a618-79bef520f8fa	bridge	br-b3f7bcf268a8
br-b7f909555d4c	1bb77a43-a39c-4ab2-8e10-c3f72d55a3c0	bridge	br-b7f909555d4c
br-fa9853f017bc	c85056eb-5fe9-4f95-92bd-89ed1b8a8bab	bridge	br-fa9853f017bc
lo	7aea1653-bc35-4fb3-861d-b88ac42ca47f	loopback	lo
br-0b31e96f66e1	2a14a53f-6e4f-4147-bb7a-54ae55eba14c	bridge	br-0b31e96f66e1
br-226917c26716	d586a2de-bb6d-499f-b84d-34c1b769be4b	bridge	br-226917c26716
br-4dcf6c5165bh	375bfe4b-a256-4923-a11f-7a5ac2e26d10	bridge	br-4dcf6c5165bh
br-6afd25ff3b16	9426d138-9016-486a-b086-3f3c1ad0e96e	bridge	br-6afd25ff3b16
br-7650805b7c9a	66e1f1ab-a38c-42b7-878f-45171b0e9732	bridge	br-7650805b7c9a
br-b86997ec4fd3	70f324b9-eb26-46b7-81d2-34089fb3feed	bridge	br-b86997ec4fd3
br-c2c7a59f534a	ef09e36c-35ae-43c8-9425-5cb4c1308b7c	bridge	br-c2c7a59f534a
br-d63615e004f9	a92f117b-ecfb-4e07-85f6-0274aac36f7f	bridge	br-d63615e004f9
docker0	4e36f19f-6722-43be-8e8e-b1833c3ba95b	bridge	docker0

Step 8: Modify a Connection via nmcli

1. Set static IP address

List available connections

```
nmcli connection show
```

We'll get the connection name is: **Wired connection 1**

NAME	UUID	TYPE	DEVICE
Wired connection 1	d062b852-feba-4cc8-abb8-d0196967999a	ethernet	eth0
lo	51d6d29e-d8b6-43f6-af3d-eb300f02067e	loopback	lo

Assign a static IP address

```
sudo nmcli connection modify "Wired connection 1" ipv4.method manual
ipv4.addresses 192.168.1.100/24 ipv4.gateway 192.168.1.1 ipv4.dns 8.8.8.8
```

- Replace the IP/gateway if needed based on your VM's network.
- **192.168.1.100/24** = static IP + subnet mask
- **192.168.1.1** = gateway
- **8.8.8.8** = DNSAssigns a static IP temporarily to **eth0**.

```
(denver@kali)-[~]
$ sudo nmcli connection modify "Wired connection 1" ipv4.method manual ipv4.addresses 192.168.1.100/24 ipv4.gateway 192.168.1.1 ipv4.dns 8.8.8.8
[sudo] password for denver:
[denver@kali)-[~]
$
```

Apply changes (restart the connection)

```
sudo nmcli connection down "Wired connection 1"
sudo nmcli connection up "Wired connection 1"
```

```
(denver@kali)-[~]
$ sudo nmcli connection down "Wired connection 1"
Connection 'Wired connection 1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)

(denver@kali)-[~]
$ sudo nmcli connection up "Wired connection 1"
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)

(denver@kali)-[~]
$
```

And we can verify with:

```
ip a
```

```
(denver@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1c:f8:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1c:f826/64 scope link
        valid_lft forever preferred_lft forever
```

2. Restore the Connection to Use DHCP

Revert to DHCP

```
sudo nmcli connection modify "Wired connection 1" ipv4.method auto
```

- Restores automatic IP assignment (DHCP).

```
(denver㉿kali)-[~]
$ sudo nmcli connection modify "Wired connection 1" ipv4.method auto
```

Reactivate the connection

```
sudo nmcli connection down "Wired connection 1"
sudo nmcli connection up "Wired connection 1"
```

```
(denver㉿kali)-[~]
$ sudo nmcli connection down "Wired connection 1"
Connection 'Wired connection 1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/12)

(denver㉿kali)-[~]
$ sudo nmcli connection up "Wired connection 1"
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/13)
```

And we can verify with:

```
ip a
```

```
(denver㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1c:f8:26 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.100/24 brd 192.168.1.255 scope global noprefixroute eth0
            valid_lft forever preferred_lft forever
            inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
                valid_lft 86357sec preferred_lft 86357sec
            inet6 fd00::e362:bdb6:f83a:b5b9/64 scope global temporary dynamic
                valid_lft 86359sec preferred_lft 14359sec
            inet6 fd00::a00:27ff:fe1c:f826/64 scope global dynamic mngtmpaddr noprefixroute
                valid_lft 86359sec preferred_lft 14359sec
            inet6 fe80::a00:27ff:fe1c:f826/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
```

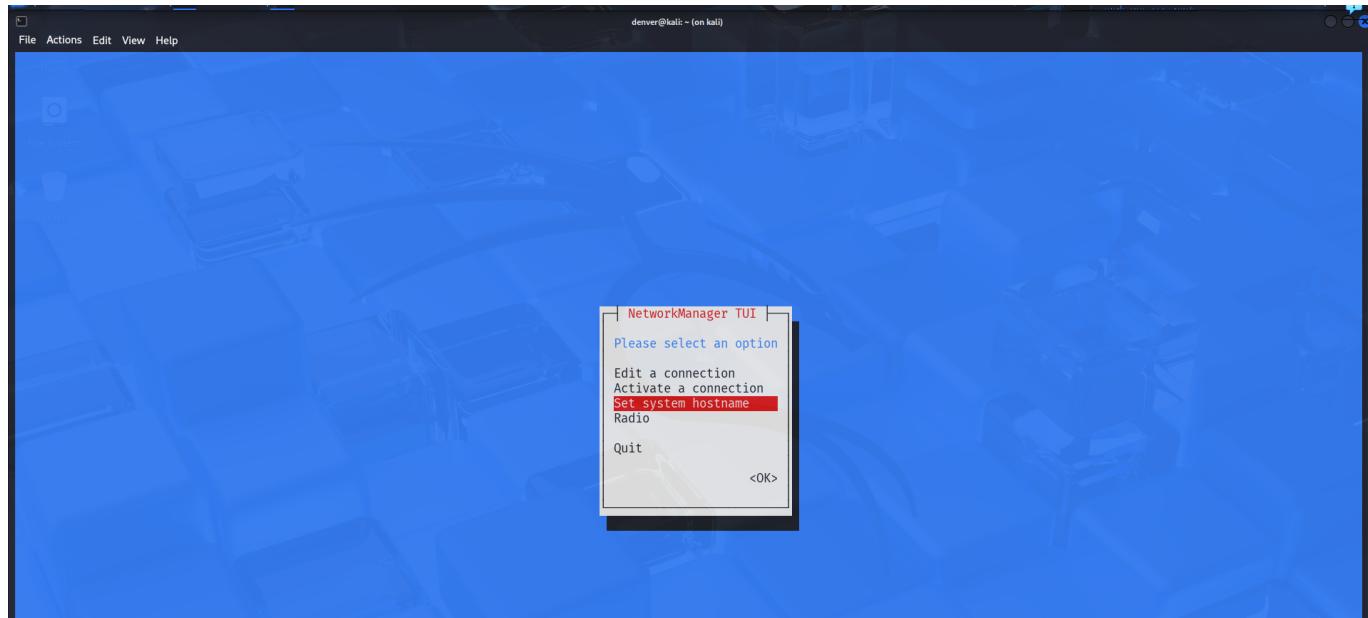
Step 9: Use nmtui for Configuration

1. Launch nmtui (graphical)

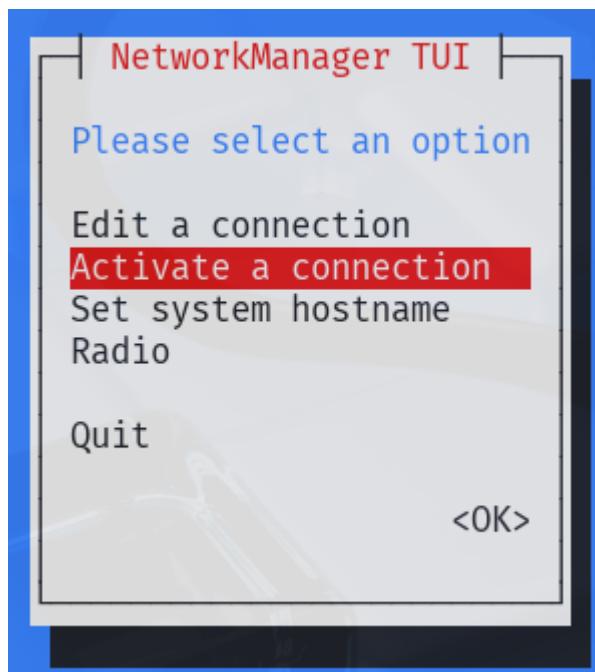
```
nmtui
```

Explanation:

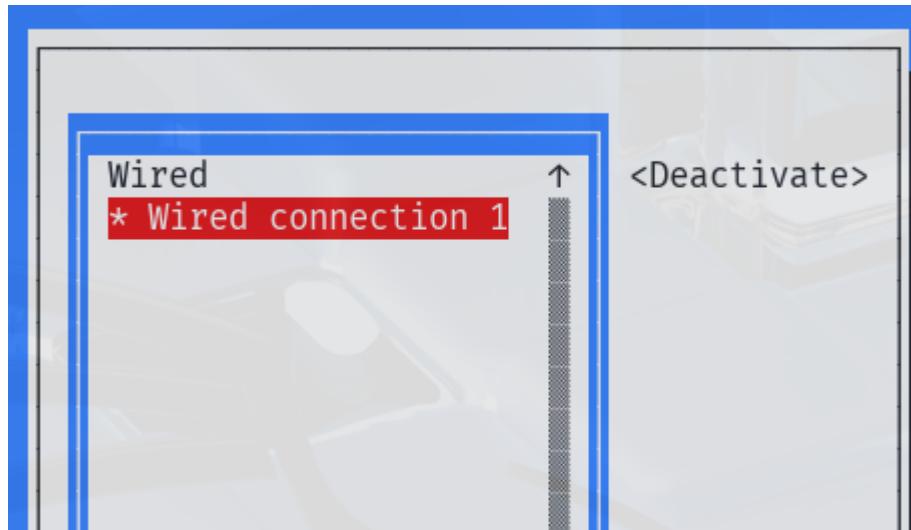
- Opens a terminal UI to manage connections.

Screenshot:**2. Describe the steps for:****Activating / Deactivating a Network Connection:**

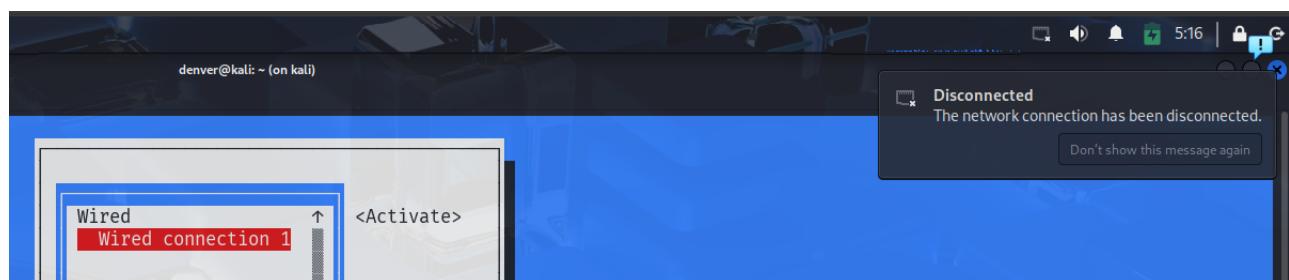
- In the main menu, select: **Activate a connection**



- A list of available connections will appear.



- Use the arrow keys to select the desired connection (e.g., Wired connection 1).
- Press Enter to:
 - Deactivate the connection if it's active.



- Activate the connection if it's inactive.



Step 10: Enable/Disable Network Connection

1. Disable connection

```
sudo nmcli connection down "Wired connection 1"
```

Explanation:

- Temporarily disables the selected network connection.

Screenshot:

```
(denver㉿kali)-[~]
└─$ sudo nmcli connection down "Wired connection 1"
[sudo] password for denver:
Connection 'Wired connection 1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/2)
└─$
```

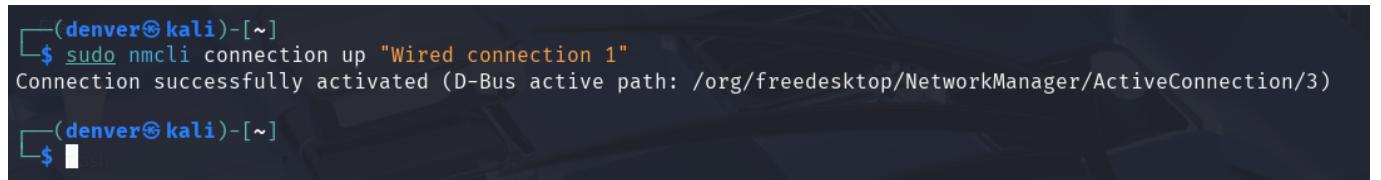
2. Re-enable connection

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

Explanation:

- Re-enables and reconnects to the network.

Screenshot:



```
(denver㉿kali)-[~]
$ sudo nmcli connection up "Wired connection 1"
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)

(denver㉿kali)-[~]
$
```