Networking

Create connectivity between 2 or more devices to transfer data.

Network Components:

- **Switch:** Connects computers and servers within a local network so they can communicate and share data.
- **Router:** Connects different networks (for example, your home network to the internet) and directs data to its destination.
- **Firewall:** Protects your network by allowing or blocking specific types of traffic, based on rules (e.g., allow SSH port 22 and HTTP port 80, but block NFS port 2049).
- **Load Balancer:** Distributes incoming requests or traffic across multiple servers so no single server is overloaded. For example, if 10 requests come in, 5 go to server1 and 5 to server2; the load balancer handles this split.
- **Ethernet Card (NIC):** Enables computers and servers to physically connect to any type of network, either by cable or wirelessly.

These components are essential for designing a reliable, scalable, and secure network.

Types of Networks:

1. **LAN (Local Area Network):** Connects devices in a small area like a home or office building.
   Example: All computers and printers in your office are connected to the same network, allowing easy sharing of files and resources.

2. **MAN (Metropolitan Area Network):** Connects multiple LANs across a city.
   Example: Different branch offices of a company in the same city connected to share data and resources.

3. **WAN (Wide Area Network):** Connects networks across large geographical areas, including countries and continents.
   Example: The Internet, connecting millions of networks worldwide.

IP ADDRESS:

IP stands for **Internet Protocol**. An IP address is a **unique identifier** assigned to every device on a network.

A **Static IP** address is like your home's permanent phone number—it never changes. It is assigned to a device manually and stays the same all the time. This is useful for things like hosting a website or connecting to a server because the address is always the same, so others can find it easily.

A **Dynamic IP (DHCP - Dynamic Host Configuration Protocol)** address is like a hotel room number that changes each time you check in. It is assigned automatically by the network and can change whenever you connect to the internet or restart your device. This is common for regular internet users because it saves resources and is managed automatically.

Example of static IP config: IP - 192.168.10.50, Netmask - 255.255.255.0, Gateway - 192.168.10.1
Example of DHCP dynamic IP: IP assigned automatically like 192.168.10.120 (might differ on reboot)

IPv4 and **IPv6** are two versions of internet addresses.

- **IPv4** uses **32-bit** addresses (like 192.168.1.1) and can support about 4 billion devices. It's the older system.
- **IPv6** uses **128-bit** addresses (like 2001:db8:85a3::8a2e:370:7334) and supports almost unlimited devices. It is newer and better.

IANA (Internet Assigned Numbers Authority) is the global organization responsible for coordinating the allocation of IP addresses, domain names, and

other internet protocol resources. It ensures that public and private IP address ranges are distributed fairly and uniquely across the Internet in coordination with regional internet registries and the global community.

Class	IP Range	Type	Private Range	Purpose
A	1.0.0.0 – 126.255.255.255	Network	10.0.0.0 – 10.255.255.255	Large networks, ISPs, big companies
	127.0.0.0 – 127.255.255.255	Loopback	N/A	Loopback, internal host communication
B	128.0.0.0 – 191.255.255.255	Network	172.16.0.0 – 172.31.255.255	Medium-size enterprise networks
C	192.0.0.0 – 223.255.255.255	Network	192.168.0.0 – 192.168.255.255	Small business/home networks
D	224.0.0.0 – 239.255.255.255	Multicast	N/A	Multicasting (streaming, TV, etc.)
E	240.0.0.0 – 255.255.255.255	Reserved/R&D(Research and Development)	N/A	Research and experimental use

Loopback:

- The loopback IP address is used for internal communication within the same system, allowing services on the device to communicate with each other without sending data outside.
- It is meant only for internal testing and communication and cannot be used to communicate with devices outside the system.

How to assign Static IP address for ethernet card in 8.8 version:

Step 1: Open the network interface configuration file for editing

command: **vi /etc/sysconfig/network-scripts/ifcfg-ens192**

Step 2: Find the line with BOOTPROTO=dhcp and change it to:

BOOTPROTO=static

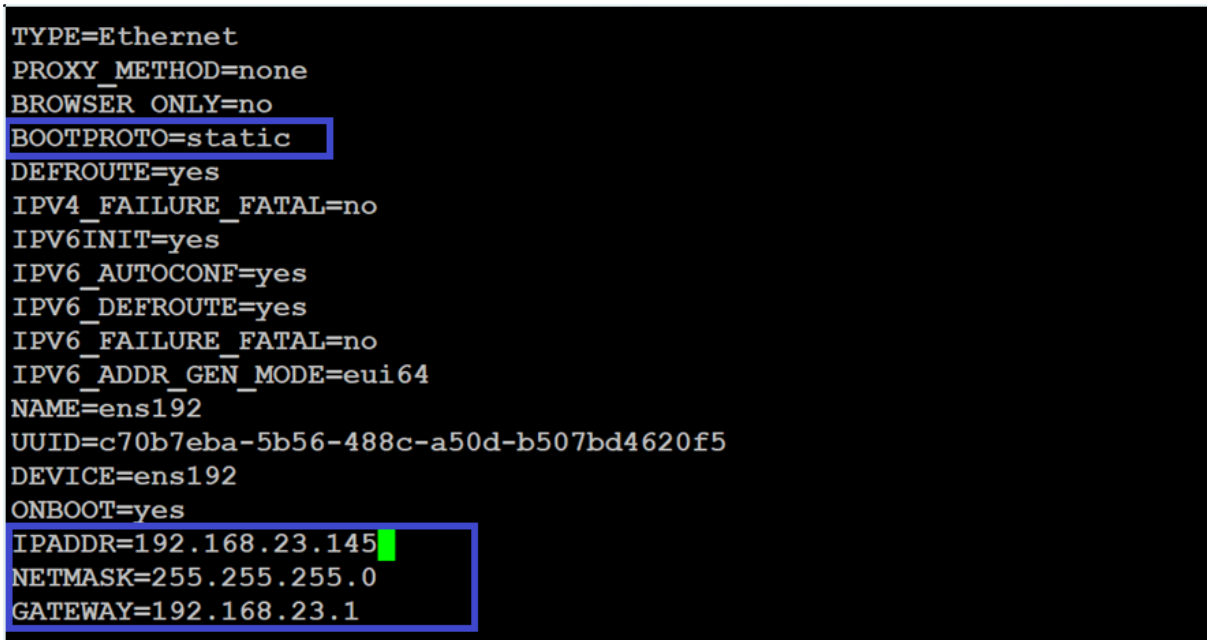
Step 3: At the end of the file, add the following lines:

IPADDR=192.168.23.150

NETMASK=255.255.255.0

GATEWAY=192.168.23.1

Save and exit the file in **vi** by pressing **Esc**, then typing **:wq** and pressing Enter.



How to assign addition IP Address to existing ethernet card:

At the end of the file, add the following lines:

IPADDR1=192.168.23.151

NETMASK1=255.255.255.0

GATEWAY1=192.168.23.1

PREFIX1=24

Step 4: Restart NetworkManager service:

command: **systemctl restart NetworkManager**

Step 5: Bring the interface down and then up using **nmcli**:

commands:

nmcli connection down ens192

nmcli connection up ens192

Step 6: Check the IP

Command:

ip a

nmcli device show ens192

Steps to assign a static IP to Ethernet card (ens192) manually in 9.5 version:

Step 1: Edit the NetworkManager connection file for your interface (ens192):

Command: **vi /etc/NetworkManager/system-connections/ens192.nmconnection**

Inside the file, locate or add the [ipv4] section and configure it as below:

```
[ipv4]
address1=192.168.23.146/24,192.168.23.1
may-fail=false
method=manual
```

- **address1=192.168.23.146/24,192.168.23.1** means IP address is 192.168.23.146 with subnet mask /24 and default gateway is 192.168.23.1
- **method=manual** indicates static IP configuration.

How to assign addition IP Address to existing ethernet card:

vi /etc/NetworkManager/system-connections/ens192.nmconnection

[ipv4]

address1=192.168.23.145/24,192.168.23.1

address2=192.168.23.150/24,192.168.23.1

may-fail=false

method=manual

Step 2: Restart NetworkManager service:

command: **systemctl restart NetworkManager**

Step 3: Bring the interface down and then up using **nmcli**:

commands:

nmcli connection down ens192

nmcli connection up ens192

Step 4: Check the IP

Command:

ip a

nmcli device show ens192

nmcli conn show

NIC Bonding:

- **NIC Bonding** is the process of combining two or more physical or virtual network interfaces into a single logical interface. This bonded interface increases network bandwidth and provides redundancy or fault tolerance.
- This ensures that if one NIC fails, network traffic automatically switches to another NIC, maintaining network connectivity without interruption.
- It allows multiple NICs to work together as one for better performance and high availability.

Overview of NIC Bonding

- Requires minimum 2 Ethernet interfaces (e.g., ens192 and ens224)
- Bonding modes commonly used:
    - **mode 0 (round-robin):** Load balancing and fault tolerance (requires switch support)
    - **mode 1 (active-backup):** Only one NIC active, backup on failure (no load balancing)
    - **mode 6 (adaptive load balancing):** Load balancing without switch requirement
- You configure slaves (NICs) attached to a master bond interface (bond0)

Configure Round-Robin Bonding as bond0 using nmtui:

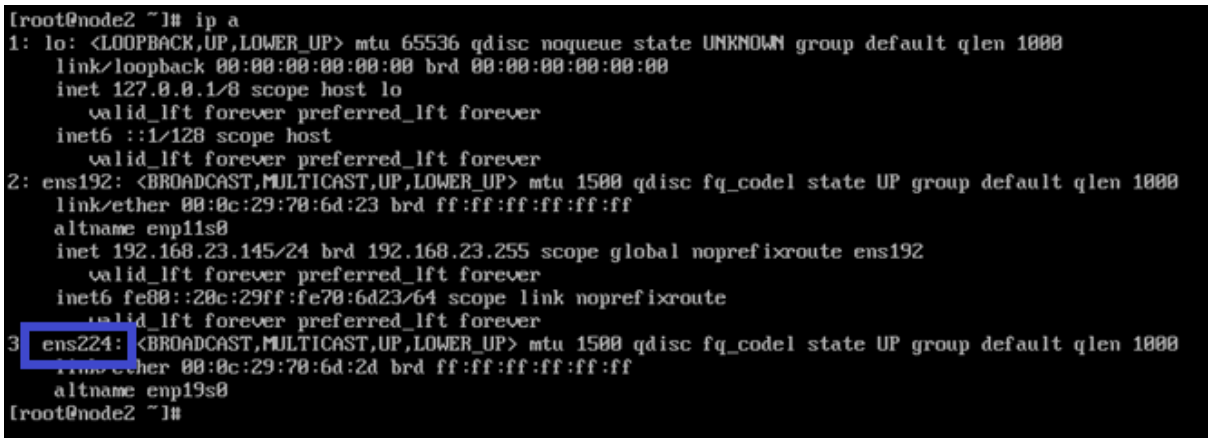
Step 1: Assign Additional ethernet to the server

Right Click on VM

Settings

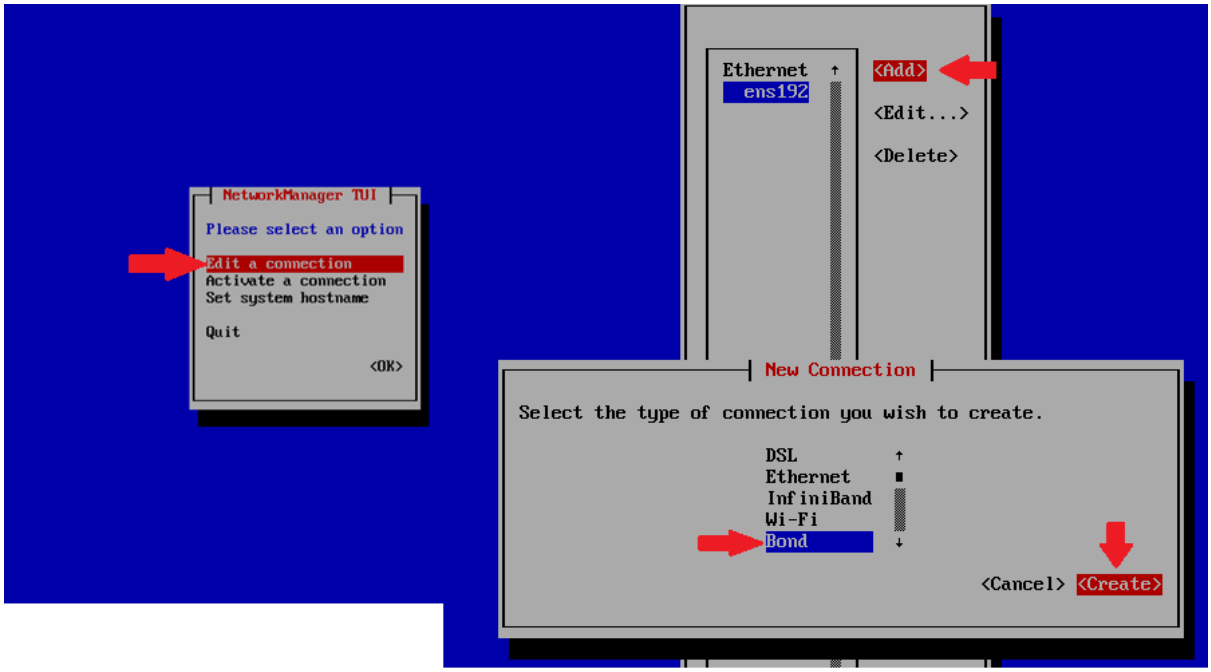
Add Network Adapter

Finish



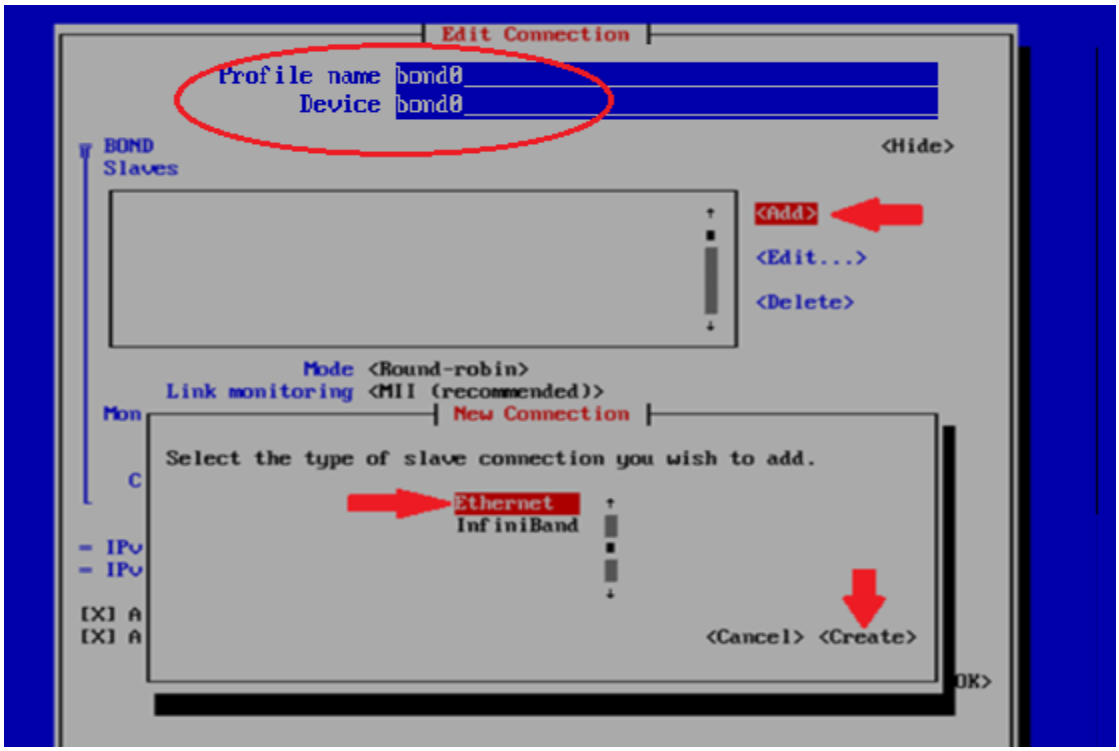
Step 2: Using "**nmtui**" command for easily configuring and managing network connections on Linux systems without a graphical interface.

Navigate to the "**Edit a connection**" tab, and also navigate to ok, and press the enter key. Navigate to **Add,** and hit the enter key and choose **bond.**

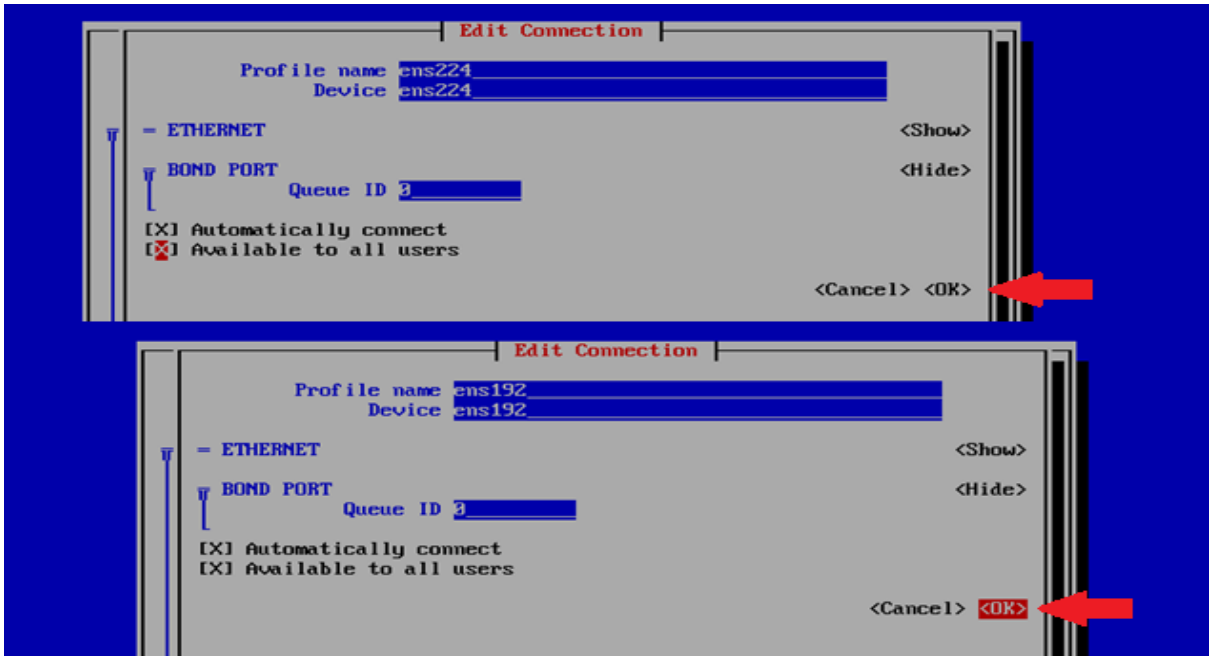


Step 3: Give the profile name and the Device name of the bond interface.

Select the type of slave connection, also navigate to create, and hit the enter key. Our type of slave connection is **Ethernet**.



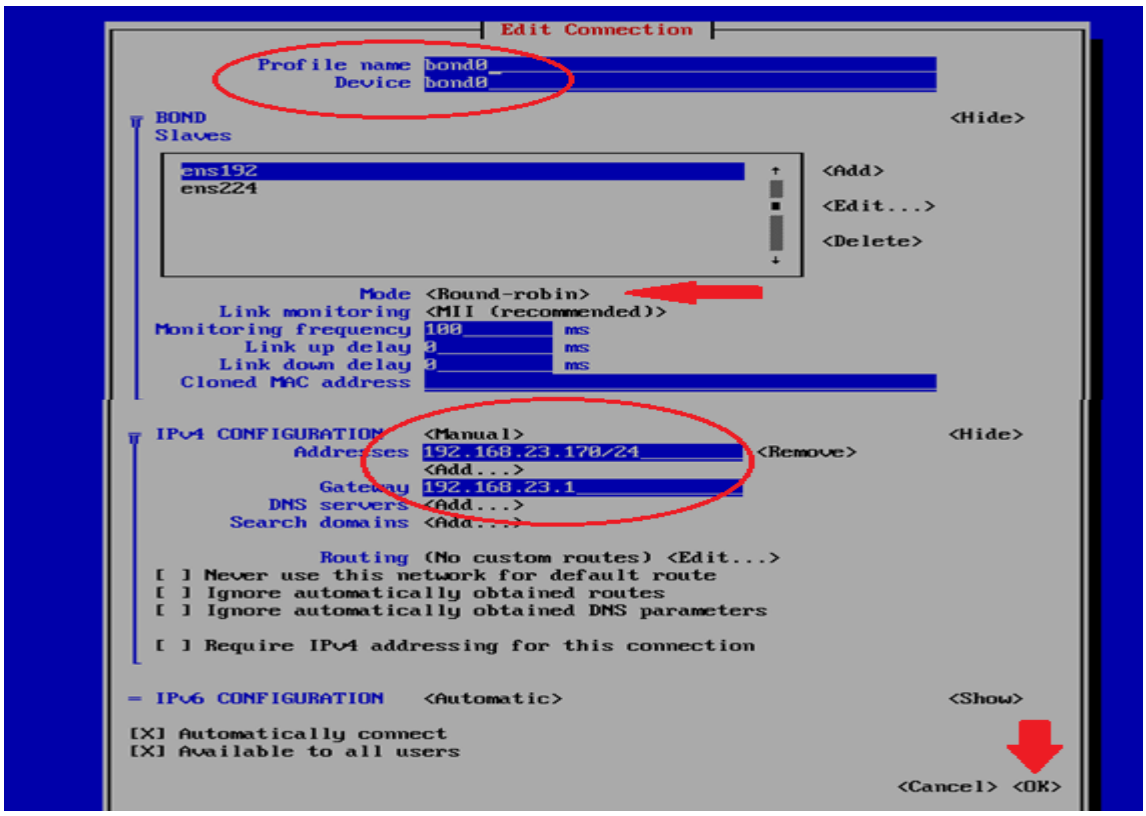
Step 4: Give the profile names, and the device names for the slaves.



Step 5:

Navigate to edit, and press enter, to add the mode "**Round-robin**".

Navigate to **IPV4 configuration**, change to manual, and include the IP. Navigate to ok, and press the enter key.



Step 6: Restart Network Manager Service

Command: systemctl restart NetworkManager

Step 7: Restart the NIC Interfaces

Bring down and bring up network connections to apply changes

nmcli conn down ens192 && nmcli conn up ens192

nmcli conn down ens224 && nmcli conn up ens224

nmcli conn down bond0 && nmcli conn up bond0

nmcli conn down ens192

```
[root@node2 ~]# 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
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master bond0 state UP group default qlen 1000
    link/ether 00:0c:29:70:6d:23 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
3: ens224: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master bond0 state UP group default qlen 1000
    link/ether 00:0c:29:70:6d:23 brd ff:ff:ff:ff:ff:ff permaddr 00:0c:29:70:6d:2d
    altname enp19s0
7: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:0c:29:70:6d:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.23.170/24 brd 192.168.23.255 scope global noprefixroute bond0
       valid_lft forever preferred_lft forever
    inet6 fe80::64c1:585d:fb80:3f78/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@node2 ~]#
```

Step 8: Verify the bonding state

Command: **cat /proc/net/bonding/bond0**

The bond interface can also be seen in /sys/class/net

Configure Round-Robin Bonding as bond0 using nmcli:

Step 1: Create the bond0 interface with round-robin mode

Command**: nmcli con add type bond con-name bond0 ifname bond0 bond.options "mode=balance-rr,miimon=100"**

Step 2: Add slave interfaces (e.g., ens192 and ens224) to bond0

Commands:

nmcli con add type bond-slave con-name ens33 ifname ens192 master bond0

nmcli con add type bond-slave con-name ens38 ifname ens224 master bond0

Step 3: Configure IP addresses and network settings on bond0

Commands:

nmcli conn modify bond0 ipv4.addresses 192.168.23.171/24

nmcli conn modify bond0 ipv4.gateway 192.168.23.1

nmcli conn modify bond0 ipv4.method manual

Step 4: Restart the NetworkManager

Command: **systemctl restart NetworkManager**

Step 5: Restart the slave interfaces and bond0 to apply changes

Commands:

nmcli conn down ens192 && nmcli conn up ens192

nmcli conn down ens224 && nmcli conn up ens224

nmcli conn down bond0 && nmcli conn up bond0

nmcli conn down ens192

Step 6: Verify the bond0 interface and IP address

Command:

ip a

nmcli conn show

Step 7: Verify the bonding state

Command: **cat /proc/net/bonding/bond0**

The bond interface can also be seen in /sys/class/net