Setting Up and Managing Network Connections Using NetworkManager

NetworkManager is a dynamic network control and configuration utility used to oversee connections and devices in SUSE Linux Enterprise Server for NVIDIA. Through the `nmcli` command-line interface, administrators can efficiently configure, monitor and troubleshoot Ethernet, Wi-Fi and VPN connections. It provides seamless network switching, automated connection management and advanced redundancy configurations like bonding and bridging.

Prerequisites

- Basic understanding of networking and IP addressing

Publication Date: 12 Nov 2026

Contents

1 The NetworkManager concept 3

2 Installing and Configuring NetworkManager 3

3 Managing the NetworkManager daemon 5

4 Creating network connections 6

5 Modifying network connections 24

6 Establishing and terminating network connections 26

7 Monitoring network connections 27

8 NetworkManager logging 28

9 The **nmcli** command reference 30

10 Troubleshooting 36

11 Legal Notice 39

A GNU Free Documentation License 40

1 The NetworkManager concept

NetworkManager is a tool that enables managing network connections and devices. NetworkManager allows you to create, configure and manage network connections and devices.

On SUSE Linux Enterprise Server for NVIDIA, NetworkManager has been designed to be fully automatic by default. It is enabled by default and is shipped with all necessary service unit files for managing primary network connection and other network interfaces. NetworkManager supports state-of-the-art encryption types and standards for network connections, including connections to 802.1X protected networks. 802.1X is the IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control.

You can switch between wired or wireless networks seamlessly, as NetworkManager automatically connects to known wireless networks and manages several network connections in parallel. You can also manually switch between available networks.

NetworkManager typically consists of the following parts:

- NetworkManager daemon—you can interact with the daemon using standard `systemd` commands
- the **`nmcli`** command-line interface
- the ncurses interface `nmtui`
- NetworkManager libraries
- configuration files.

2 Installing and Configuring NetworkManager

On SUSE Linux Enterprise Server for NVIDIA, NetworkManager is installed and enabled by default and thus runs out of the box. Usually, you do not have to reinstall it or change the configuration, but if such an action is needed, these sections provide guidance.

2.1 Installing NetworkManager

You can install NetworkManager using **`zypper`**. Once you install, you can enable NetworkManager to start automatically at boot time.

Install NetworkManager:

```
> sudo zypper install NetworkManager
```

Enable NetworkManager:

```
> sudo systemctl enable NetworkManager
```

Once NetworkManager is enabled, the change persists across reboots.

2.2 Configuring NetworkManager behavior

The behavior of NetworkManager is defined in its central configuration file `/usr/lib/NetworkManager/NetworkManager.conf`. The system runs with the built-in defaults and the `NetworkManager.conf` file is not mandatory. To modify the behavior of NetworkManager, create spinoff files in the directory `/etc/NetworkManager/conf.d`.

`NetworkManager.conf` is the primary location for viewing the default behavior and settings of NetworkManager, including logging, connection management, and network device handling.

The file consists of sections of key-value pairs. Each key-value pair must belong to a section. A section starts with a name enclosed in `[]`. Lines beginning with a `#` are considered comments. A common configuration includes the `[main]` section with the `plugins` value, `[logging]`, and `[connectivity]`:

```
[main] ❶
plugins=keyfile ❷
dhcp=dhclient

[connectivity] ❸
uri=http://name.org

[logging] ❹
level=INFO
domains=ALL
```

❶ Controls the general settings for NetworkManager.

❷ Manages how the connection profiles are stored. The [keyfile] plug-in supports all the connection types and capabilities of NetworkManager.

❸ Defines connection defaults and options and specifies the URI to check the network connection.

❹ Manages the NetworkManager logging levels and domains.

PROCEDURE 1: CONFIGURING NETWORKMANAGER.CONF

1. Stop the NetworkManager service:

```
> sudo  systemctl stop network
```

2. Create spinoff files in the `/etc/NetworkManager/conf.d/` if they do not exist.
3. To prevent NetworkManager from managing specific interfaces such as Ethernet ports, add them to `/etc/NetworkManager/conf.d/unmanaged-devices.conf`.
4. Alternatively, back up the existing configuration files before modifying them.
5. Modify the spinoff files to suit your needs.
6. Save the changes.
7. Start NetworkManager:

```
> sudo systemctl start network
```

8. View the existing configuration and settings:

```
> sudo  nmcli connection show
```

3 Managing the NetworkManager daemon

The NetworkManager daemon is a standard `systemd` service so you can use the **`systemctl`** command to manage the daemon.

You can use any of the following commands to manage the NetworkManager daemon:

Checking status

To check if the NetworkManager daemon is running and thus the network should be active:

```
> systemctl status network
```

Restarting the daemon

For example, in cases of networking problem, you can try to restart the NetworkManager Daemon by using the following command:

```
> sudo systemctl restart network
```

Stopping the daemon

You can stop NetworkManager during network configuration changes or to switch to manual control. You can also stop the service for troubleshooting or debugging network issues. Stopping NetworkManager prevents it from automatically managing the interface or changing your manual configuration. You can also stop NetworkManager to manually address specific requirements for network interfaces.

To stop the NetworkManager daemon:

```
> sudo systemctl stop network
```

Starting the NetworkManager daemon

If you stopped the daemon, you need to start it again to manage all network connections:

```
> sudo systemctl start NetworkManager
```

4 Creating network connections

You can create a network connection profile using the **`nmcli`** command.

NetworkManager stores all network configurations as a connection profile, which is a collection of data that describes how to create or connect to a network. These connection profiles are stored as files in the `/etc/NetworkManager/system-connections/` directory by default. Each network connection profile (Wi-Fi, Ethernet, VPN) is represented by a separate file in this directory.

A connection is an instance of a particular connection profile that is active when a particular device uses the connection. The device may have more than one connection profile configured. The other connections can be used to fast switch from one connection to another. For example, if the active connection is not available, NetworkManager tries to connect the device to another configured connection.

The NetworkManager daemon manages network connections. You can interact with it using a command-line interface: **`nmcli`** or ncurses interface: **`nmtui`**.

4.1 Creating an Ethernet connection

Configure the Ethernet connection using the **`nmcli`** command and proceed as follows:

1. List available devices to get the exact device name:

```
> nmcli device
```

2. View the list of connections to make sure the profile name you want to use is not already taken:

```
nmcli connection show
```

The NetworkManager creates a profile for each Network Interface Controller (NIC). To connect the NIC to networks with different settings, you must create separate profiles for each network.

3. Create a new connection profile:

```
> sudo nmcli connection add con-name CONNECTION_NAME ifname DEVICE_NAME type
 ethernet
```

4. View the existing network settings of the new connection profile:

```
> nmcli connection show CONNECTION_NAME
```

5. Configure the connection profile.
The generic command syntax is the following:

```
> sudo nmcli connection modify CONNECTION_NAME SETTING VALUE
```

For example, automatic IP address assignment through DHCP or SLAAC is enabled by default. You may want to configure a static IP address:

- for IPv4

```
> sudo nmcli connection modify CONNECTION_NAME ipv4.method manual
 ipv4.addresses 192.0.2.1/24 ipv4.gateway 192.0.2.254 ipv4.dns 192.0.2.200
 ipv4.dns-search example.com
```

- for IPv6

```
> sudo nmcli connection modify INTERNAL-LAN ipv6.method manual ipv6.addresses
 2001:db8:1::fffe/64 ipv6.gateway 2001:db8:1::fffe ipv6.dns 2001:db8:1::ffbb
 ipv6.dns-search example.com
```

6. Configure the system to handle network-dependent services. To prevent services from failing or experiencing race conditions if they are activated before the static IP address is fully assigned by NetworkManager, choose one of the following methods:

- a. Enable `NetworkManager-wait-online.service` to force the system to wait until network interfaces are up before proceeding with the boot target:

```
> sudo systemctl enable NetworkManager-wait-online.service
```

b. Individual systemd unit overrides: For each service binding to a specific static IP, manually create a systemd unit override that enforces proper startup ordering and behavior:

```
/etc/systemd/system/sshd.service.d/override.conf
[Unit]
After=network-online.target
[Service]
Restart=always
```

- a. Centralized Non-Local Binding (Alternative): Instead of managing overrides per service, you can allow applications to safely bind to IPv4 and IPv6 addresses that are not yet fully initialized on the system. Create a custom sysctl file at `/etc/sysctl.d/90-nonlocal_bind.conf` containing the following lines:

```
net.ipv4.ip_nonlocal_bind = 1
net.ipv6.ip_nonlocal_bind = 1
```

b. Apply the sysctl configuration immediately by running:

```
> sudo sysctl --system
```



Note

Non-local binding should not be enabled by default unless required, as it may cause unexpected behavior or errors in certain strict or legacy applications.

7. Activate the profile:

```
> sudo nmcli connection up CONNECTION_NAME
```

8. Verify the configurations:

a. Verify the IP settings of the NIC:

```
> ip address show HOSTNAME
```

b. Verify the default IPv4 gateway:

```
> ip route show default
```

c. Verify the default IPv6 gateway:

```
> ip -6 route show default
```

d. View the DNS settings:

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

4.2 Creating a Wi-Fi connection profile

You can connect to a Wi-Fi using the **`nmcli`** command. The NetworkManager creates a new connection profile when you connect to a Wi-Fi for the first time. You can configure the profile after connecting to the Wi-Fi.

1. Enable the Wi-Fi radio:

```
> sudo nmcli radio wifi on
```

2. View the list of available devices:

```
> sudo nmcli device
```

3. Connect to Wi-Fi:

```
> sudo nmcli device wifi connect WI-FI_CONNECTION_NAME
    PASSWORD
```

4. View the existing network settings of the new connection profile:

```
> sudo nmcli connection show CONNECTION_NAME
```

5. Configure the connection profile as needed, using the command:

```
> sudo nmcli connection modify CONNECTION_NAME
  SETTING VALUE
```

For example, to configure a static IPv4:

```
> sudo nmcli connection modify WI-FI_CONNECTION_NAME ipv4.method manual
 ipv4.addresses IP_ADDRESS/SUBNET_MASK
```

To configure a static IPv6 address

```
> sudo nmcli connection modify WI-FI_CONNECTION_NAME ipv6.method manual
 ipv6.addresses IP_ADDRESS/SUBNET_MASK
```

6. Restart the connection:

```
> sudo nmcli connection up WI-FI_CONNECTION_NAME
```

7. Verify the connection using the command: .

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

The list of available active connections appears.

4.3 Creating a network bond

A network bond combines physical and virtual network interfaces and provides a logical interface. You can create network bonds on Ethernet devices, Virtual LANs, and so on. Network bonding can increase bandwidth and/or provide redundancy.

REQUIREMENTS

- Network connection: two or more physical or virtual networks installed on the server
- Existing network interfaces to include in the bonded device
- Switch support, depending on the bonding mode
- To use Ethernet devices for network bonding, you must install the physical or virtual Ethernet devices on the server.
- When using network teams, bridges or VLAN devices as ports for a bond, you can create them either during the bond creation process or beforehand.

RESTRICTIONS

- Do not split bonds over multiple switches.
  In most hardware setups, all network interfaces in a bonded device must be connected to the same switch. For more information, consult your switch vendor documentation.
- IBM POWER: Bonding modes 5 and 6 (`balance-tlb` and `balance-alb`) unsupported by ibmveth.

The bonding drivers in `tlb` or `alb` modes send Ethernet Loopback packets with both the source and destination MAC addresses listed as the Virtual Ethernet MAC address. These packets are not supported by POWER firmware. Therefore, bonding modes 5 and 6 are unsupported by ibmveth.

- Bonding and virtualization: Bonded devices are made up of multiple network interfaces. In most configurations, you should only configure bonding in the host. Virtual interfaces to guests are then created as a bridge with the bonded devices, simplifying guest creation and deployment.
  It is possible, but not recommended, to configure bonding in a guest. When configuring bonding in a guest, you must assign multiple interfaces to the guest and configure the host without bonding. You must also be careful to configure the host and its network bridges so that you do not mix bonding in the host and guests.

BONDING MODES

The following bonding modes are available:

- (*0*) *balance-rr*
  Packets are transmitted in round-robin fashion from the first to the last available interface. Provides fault tolerance and load balancing. Requires switch support. Certain switches might fail with this mode.
- (*1*) *active-backup*
  Only one network interface is active. If it fails, a different interface becomes active. Provides fault tolerance. This is the default mode. No specific switch support is required.
- (*2*) *balance-xor*
  Traffic is split between all available interfaces based on the number of interfaces included in the bonded device. Provides fault tolerance and load balancing. Requires switch support. Certain switches might fail with this mode.
- (*3*) *broadcast*
  All traffic is broadcast on all interfaces. Provides fault tolerance. Requires switch support. Certain switches might fail with this mode. If possible, use mode *1* instead, or use this mode to provide sniffing capability by connecting each member of the bond to a different switch or device.
- (*4*) *802.3ad*
  Also called *LACP*. All interfaces in the LACP group must share the same speed and duplex settings, and must be connected to the same switch. Provides fault tolerance and load balancing. Requires **`ethtool`** support in the interface drivers, and a switch that supports and is configured for IEEE 802.3ad Dynamic link aggregation. If your switch supports it, this is the preferred mode.
- (*5*) *balance-tlb*

Adaptive transmit load balancing. Provides fault tolerance and load balancing. Requires **`ethtool`** support in the interface drivers. No specific switch support is required, but certain switches might fail with this mode.

- (*6*) *balance-alb*
  Adaptive load balancing. Provides fault tolerance and load balancing. Requires **`ethtool`** support in the interface drivers. No specific switch support is required, but certain switches might fail with this mode.

Consult your hardware manual to check which modes your switch supports.

For a more detailed description of the modes, see https://www.kernel.org/doc/Documentation/networking/bonding.txt.

PROCEDURE 2: CREATING A NETWORK BOND

1. Create a bond interface:

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

   A network bond, *`NETWORK_BOND`* that uses `active-backup` mode is created.

2. View the list of network interfaces:

```
> nmcli device status

   DEVICE         TYPE      STATE                   CONNECTION
   wlan0          wifi      connected               Vision
   virbr0         bridge    connected (externally)  virbr0
   p2p-dev-wlan0  wifi-p2p  disconnected            --
   eth0           ethernet  unavailable             --
   lo             loopback  unmanaged               --
```

   The list of available network interfaces appears. You can also add devices that are not configured to the bond. In the given list, `p2p-dev-wlan0` is not configured, while `virbr0` is configured and has a connection profile.

3. To configure `p2p-dev-wlan0` as a port, create a connection profile:

```
> sudo nmcli connection add type wifi-p2p slave-type bond con-name bond0-port1
 ifname p2p-dev-wlan0 master bond0
```

   This creates a new profile for `p2p-dev-wlan0` and adds it to the `bond0` connection. The name of the bond is `bond0`.

4. To assign `virbr0` to a bond:

```
> sudo nmcli connection modify virbr0 master bond0
```

   The connection profile for `virbr0` is added to the `bond0` connection.

5. Activate the connection:

```
> sudo nmcli connection up virbr0
```

6. Configure the IPv4 settings:

   - To use `bond0` as a port for other devices:

```
> sudo nmcli connection modify bond0 ipv4.method disabled
```

   - To use DHCP, no configuration is required.
   - To configure a static IPv4 address, network mask, default gateway and DNS server to the `bond0` connection:

```
> sudo nmcli connection modify bond0 ipv4.addresses '192.0.2.1/24'
 ipv4.gateway '192.0.2.254' ipv4.dns '192.0.2.253' ipv4.dns-search
 'example.com' ipv4.method manual
```

7. Configure the IPv6 settings:

   - To use this bond device as a port of other devices:

```
> sudo nmcli connection modify bond0 ipv6.method disabled
```

   - To use stateless address autoconfiguration (SLAAC), no action is required.
   - To set a static IPv6 address, network mask, default gateway and DNS server to the `bond0` connection:

```
> sudo nmcli connection modify bond0 ipv6.addresses '2001:db8:1::1/64'
 ipv6.gateway '2001:db8:1::fffe' ipv6.dns '2001:db8:1::fffd' ipv6.dns-search
 'example.com' ipv6.method manual
```

8. Activate the connection:

```
> sudo nmcli connection up bond0
```

9. View and verify the connections:

```
> nmcli device
```

The list of connections appears.

4.4 Configuring a network team

Network teaming combines two or more network interfaces into a single teamed device to increase bandwidth and/or provide redundancy. The behavior of the teamed device is configured using teaming modes. Network teaming can increase bandwidth and/or provide redundancy.

REQUIREMENTS

- Network connection
- Existing network interfaces to include in the teamed device
- Switch support in kernel, depending on the teaming mode
- The package `libteam-tools` is installed
- Install the `teamd` and `NetworkManager-team` packages:

```
> sudo  zypper install teamd
> sudo  zypper install NetworkManager-team
```

The `teamd`, `NetworkManager-team` and `libteam-tools` packages are deprecated in SUSE Linux Enterprise Server for NVIDIA 16.1 as the upstream activities are deprecated. You must migrate existing team interfaces to bonding configurations managed through NetworkManager. NetworkManager bonding supports many bonding scenarios, including LACP.

- Install two or more physical or virtual devices on the server:
- To use Ethernet devices as ports of the team, the physical or virtual Ethernet devices must be installed on the server and connected to a switch.
- To use bond, bridge or VLAN devices as ports of the team, create them in advance or when you create the team.

RESTRICTIONS

- Do not split teams over multiple switches.

In most hardware setups, all network interfaces in a teamed device must be connected to the same switch. For more information, consult your switch vendor documentation.

- Teaming and virtualization:
  Teamed devices are made up of multiple network interfaces. In most configurations you should only configure teaming in the host. Virtual interfaces to guests are then created as a bridge with the teamed devices, simplifying guest creation and deployment.
  It is possible, but not recommended, to configure teaming in a guest. When configuring teaming in a guest, you must assign multiple interfaces to the guest and configure the host without teaming. You must also be careful to configure the host and its network bridges so that you do not mix teaming in the host and guests.

TEAMING MODES

The following teaming modes are available:

- *broadcast*
  All traffic is broadcast on all interfaces. Provides fault tolerance. Requires switch support.
- *roundrobin*
  Packets are transmitted in round-robin fashion from the first to the last available interface. Provides fault tolerance and load balancing. Requires switch support.
- *activebackup*
  Only one network interface is active. If it fails, a different interface becomes active. Provides fault tolerance.
- *loadbalance*
  The teamed device transmits packets via all its interfaces, performing load balancing (passive or active) with a use of hash functions. For passive load balancing, only the BPF hash function is used. For active load balancing, the runner finds the best balance by moving hashes between available interfaces. Provides fault tolerance and load balancing. No specific switch support is required.
- *lacp*
  All interfaces in the LACP group must share the same speed and duplex settings, and must be connected to the same switch. Provides fault tolerance and load balancing. Requires **`ethtool`** support in the interface drivers, and a switch that supports and is configured for IEEE 802.3ad Dynamic link aggregation. If your switch supports it, this is the preferred mode.

Consult your hardware manual to check which modes your switch supports.

PROCEDURE 3: CREATING A NETWORK TEAM

1. Create a team interface:

```
> sudo nmcli connection add type team con-name CONNECTION_NAME ifname TEAM_NAME
 team.runner RUNNER-TYPE
```

   For example, to create a network team `team0` with the `activebackup` runner, run:

```
> sudo nmcli connection add type team con-name team0 ifname team0 team.runner
 active.backup
```

2. View the list of network interfaces:

```
> nmcli device status
```

```
DEVICE         TYPE      STATE                   CONNECTION
    wlan0          wifi      connected               Vision
    virbr0         bridge    connected (externally)  virbr0
    p2p-dev-wlan0  wifi-p2p  disconnected            --
    eth0         ethernet  unavailable             --
    lo             loopback  unmanaged               --
```

   You can add the listed devices to the team. The following examples use `p2p-dev-wlan0` and `virbr0`. Note that `p2p-dev-wlan0` is not configured, while `virbr0` has a connection profile.

3. Configure port interfaces to the team:

```
> sudo nmcli connection add type wifi-p2p slave-type team con-name team0-port1
 ifname p2p-dev-wlan0 master team0
```

   A new profile is created for `p2p-dev-wlan0` and added to the `team0` connection.

4. Assign the existing connection to the team:

```
> sudo nmcli connection modify bond0 master team0
```

   The connection profile for `bond0` is added to the `team0` connection.

5. Activate the connection:

```
> sudo nmcli connection up bond0
```

6. Configure the IPv4 settings:

    - To use the team device as a port of other devices:

      ```
      > sudo nmcli connection modify team0 ipv4.method disabled
      ```

    - To use DHCP, no configuration is required.

    - To configure a static IPv4 address, network mask, default gateway and DNS server to the `bond0` connection, run the command:

      ```
      > sudo nmcli connection modify team0 ipv4.addresses '192.0.2.1/24'
       ipv4.gateway '192.0.2.254' ipv4.dns '192.0.2.253' ipv4.dns-search
       'example.com' ipv4.method manual
      ```

7. Configure the IPv6 settings:

    - To use this team device as a port of other devices:

      ```
      > sudo nmcli connection modify team0 ipv6.method disabled
      ```

    - To use stateless address autoconfiguration (SLAAC), no action is required.

    - To set a static IPv6 address, network mask, default gateway and DNS server to the `team0` connection:

      ```
      > sudo nmcli connection modify team0 ipv6.addresses '2001:db8:1::1/64'
       ipv6.gateway '2001:db8:1::fffe' ipv6.dns '2001:db8:1::fffd' ipv6.dns-search
       'example.com' ipv6.method manual
      ```

8. Activate the connection:

   ```
   > sudo nmcli connection up team0
   ```

9. To view the status of the team:

   ```
   > sudo teamdctl team0 state
   ```

4.5 Configuring a network bridge

A network bridge is a device that facilitates communication between two or more network segments, creating a single network from multiple segments

To configure a network bridge, ensure the following:

- Install two or more physical or virtual devices on the server.
- To use Ethernet devices as ports for the bridge, ensure the server has physical or virtual Ethernet devices installed and connected to a switch.
- When using team, bond or VLAN devices as ports for the bridge, you can create these devices either during bridge creation or beforehand.

PROCEDURE 4: CREATING A NETWORK BRIDGE

1. Create a bridge interface:

```
> sudo nmcli connection add type bridge con-name CONNECTION_NAME ifname BRIDGE_NAME
```

   For example, create a bridge `bridge0` by running the command:

```
> sudo nmcli connection add type bridge con-name bridge0 ifname bridge0
```

2. View the list of network interfaces to verify that `bridge0` is created:

```
> nmcli device status

    DEVICE         TYPE      STATE                                  CONNECTION
    wlan0               wifi      connected                              Vision
    virbr0              bridge    connected (externally)                 virbr0

    p2p-dev-wlan0       wifi-p2p  disconnected                           --

    eth0                ethernet  unavailable                            --

    lo                  loopback  unmanaged                              --
    bridge0             bridge    connecting (getting IP configuration)  brdige0
```

   The bridge is in the state *getting IP configuration*, because you have not assigned interfaces to it yet. Later you will assign the interfaces: `p2p-dev-wlan0` (not configured) and `virbr0` (configured with a connection profile).

3. Add interfaces to the bridge:

```
> sudo nmcli connection add type wifi-p2p slave-type bridge con-name bridge0-port1
 ifname p2p-dev-wlan0 master bridge0
```

   This creates a new profile for p2p-dev-wlan0 and adds it to the `bridge0` connection.

4. To assign an existing connection to the bridge:

```
> sudo nmcli connection modify bond0 master bridge0
```

   This adds the connection profile for `bond0` to the `bridge0` connection.

5. Restart the connection:

```
> sudo nmcli connection up bond0
```

6. Configure the IPv4 settings:

   - To use the bridge device as a port of other devices:

```
> sudo nmcli connection modify bridge0 ipv4.method disabled
```

   - To use DHCP, no configuration is required.
   - To configure a static IPv4 address, network mask, default gateway and DNS server to the `bridge0` connection:

```
> sudo nmcli connection modify bridge0 ipv4.addresses '192.0.2.1/24'
 ipv4.gateway '192.0.2.254' ipv4.dns '192.0.2.253' ipv4.dns-search
 'example.com' ipv4.method manual
```

7. Configure the IPv6 settings:

   - To use this bridge device as a port of other devices:

```
> sudo nmcli connection modify bridge0 ipv6.method disabled
```

   - To use stateless address autoconfiguration (SLAAC), no action is required.
   - To set a static IPv6 address, network mask, default gateway and DNS server to the `bridge0` connection:

```
> sudo nmcli connection modify bridge0 ipv6.addresses '2001:db8:1::1/64'
 ipv6.gateway '2001:db8:1::fffe' ipv6.dns '2001:db8:1::fffd' ipv6.dns-search
 'example.com' ipv6.method manual
```

8. Activate the connection:

```
> sudo nmcli connection up bridge0
```

9. Verify the connection:

```
> nmcli device
```

When you activate any port of the connection, NetworkManager also activates the bridge, but not the other ports of it.
Enable all ports automatically when the bridge is enabled:

```
> sudo nmcli connection modify bridge0 connection.autoconnect-slaves 1
```

10. View the link status of Ethernet devices that are ports of a specific bridge.

```
> sudo ip link show master bridge0
```

11. View the status of Ethernet devices that are ports of any bridge device.

```
> sudo bridge link show
```

4.6 Configuring a VPN connection

A VPN (Virtual Private Network) connection is a secure, encrypted tunnel between your device and another network over the Internet.

You can configure a VPN connection using the **`nmcli`** command.

PROCEDURE 5: INSTALLING AND CONFIGURING A VPN CONNECTION

1. Install OpenVPN:

```
> sudo  zypper install networkmanager-openvpn
```

2. Create a VPN connection:

```
> sudo nmcli connection add type vpn con-name MyOpenVPN ifname -- vpn-type openvpn
```

3. Configure the settings:

```
> sudo nmcli connection modify MyOpenVPN vpn.data "remote=VPN-SERVER-
ADDRESS,username=YOUR-USERNAME"
```

```
> sudo nmcli connection modify MyOpenVPN vpn.secrets "password=YOUR-PASSWORD"
```

4. Configure DNS settings:

```
> sudo nmcli connection modify MyOpenVPN ipv4.dns "8.8.8.8 8.8.4.4"
```

5. Add routes:

```
> sudo nmcli connection modify MyOpenVPN ipv4.routes "192.168.1.0/24 192.168.1.1"
```

6. Activate the VPN connections:

```
> sudo nmcli connection up MyOpenVPN
```

7. Verify if the configured connection is active:

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

4.7 Convert teamd configuration to NetworkManager

In SUSE Linux Enterprise Server for NVIDIA16.1, **`libteam`** and **`teamd`** are deprecated and hence Network Load Balancing configurations that relied on **`libteam`** in the past must be converted to a NetworkManager based bonding interface.

PROCEDURE 6: ACTIVE LOAD BALANCING WITH LACP

1. A sample `team0` interface configuration (`/etc/sysconfig/network/ifcfg-team0`) is mentioned below based on which the commands are described.

```
BOOTPROTO="static"
STARTMODE="onboot"

TEAM_RUNNER="lacp"
TEAM_LACP_TX_HASH="l4,l3,eth,vlan"
TEAM_LACP_TX_BALANCER_NAME="basic"
TEAM_LACP_TX_BALANCER_INTERVAL="100"
TEAM_LACP_ACTIVE=true
TEAM_LACP_SELECT_POLICY="lacp_prio"
TEAM_LW_NAME="ethtool"
TEAM_LW_ETHTOOL_DELAY_UP="10"
TEAM_LW_ETHTOOL_DELAY_DOWN="10"
IPADDR="10.1.1.1"
NETMASK="255.255.255.0"
IPADDR_0="2001:db8::1"
PREFIXLEN_0="64"
TEAM_PORT_DEVICE_0="eth0"
```

```
TEAM_PORT_DEVICE_1="eth3"
```

This configuration describes a network bond using active transmit load balancing. Any load balancing that happens on incoming traffic is performed by the switch.

2. Run the following **nmcli** commands to configure a bonding interface (bond0), and add the Ethernet ports to it:

```
> sudo  nmcli connection add type bond bond-options "...." \
ifname bond0 con-name bond0 \
ipv4.method manual ipv4.address 10.1.1.1/24 \
ipv6.method manual ipv6.address 2001:db8::1/64
> sudo  nmcli connection add type ethernet ifname eth0 con-name eth0 master bond0
> sudo  nmcli connection add type ethernet ifname eth3 con-name eth3 master bond0
> sudo  nmcli connection modify bond0 connection.autoconnect-slaves yes
```

3. For the specific `ifcfg-team0` example shown above, you can translate the LACP settings into options for the bond device shown as "..." as given below:

   - **TX_BALANCER_NAME=basic** indicates active load balancing. There is no other balancer in `teamd`, so this is actually redundant and can be ignored.
   - The protocol chosen for managing the bond needs to be provided to the kernel via the mode option. The **TEAM_RUNNER** variable above specifies LACP, the Link Aggregation Protocol. This is defined by IEEE 802.3ad, and that's also the name by which it is referred to in the kernel bond documentation. So the corresponding option is **mode=802.3ad**
   - The **TX_HASH** setting describes which headers are used to distribute outgoing packets to links. The bonding equivalent to the above settings is **xmit_hash_policy=layer3+4**
   - The **TX_BALANCER_INTERVAL** describes how frequently the status of the Ethernet ports is inspected (in milliseconds). The corresponding option is **miimon=100**.
   - The **LW_NAME** variable specifies the link watch method. The bonding driver can check the availability of a link using one of two methods; one, by using some generic ioctls, or by using the NIC driver's internal **netif_carrier_ok** logic. When **LW_NAME** is set to **ethtool**, choose **use_carrier=0**; otherwise, choose **use_carrier=1**.
     When ethtool is used, the corresponding DELAY values can be translated to bonding options updelay; and downdelay, respectively.
     So with all of the above together, the first **nmcli** command would use bond-options **mode=802.3ad,miimon=100,xmit_hash_policy=layer3+4,use_carrier=0,updelay=10,downdelay=10**.

The resulting configuration is persistent, it activates whenever the system reboots.

PROCEDURE 7: ACTIVE-PASSIVE SETUPS USING MII MONITORING

1. This is another way of using is active-backup, where two or more Ethernet ports are used for redundancy in the face of network failures. Usually, each NIC is connected to a different switch in order to establish a separate network path. When the active link is failing, the connection will automatically fail over to the secondary port.
   A typical ifcfg file for a teaming based active-backup setup would look like this (excluding the BOOTPROTO, IPADDR settings and so on from the example given above).

```
TEAM_RUNNER="activebackup"
TEAM_LW_NAME="ethtool"
TEAM_PORT_DEVICE_0="eth0"
TEAM_PORT_DEVICE_1="eth3"
```

2. Run the following command to migrate configurations to NetworkManager based bonding connections.

   - The mode parameter needs to be set as **`mode=active-backup`**.
   - The link watch method chosen here is ethtool, which translated to **`miimon=100 use_carrier=0`**.

PROCEDURE 8: ACTIVE-PASSIVE SETUPS USING ARP FOR LINK MONITORING

1. The activebackup runner in teamd supports other link watch methods apart from ethtool; in particular, ARP pings. A typical configuration snippet would look like this:

```
TEAM_RUNNER="activebackup"
TEAM_LW_NAME="arp_ping"
TEAM_LW_ARP_PING_TARGET=10.1.2.3
TEAM_LW_ARP_PING_INTERVAL=60
```

2. This is translated to bond-options **`mode=active-backup arp_ip_target=10.1.2.3 arp_interval=60`**.
   Other link watch types supported by teams are not supported by the kernel bonding driver. In particular, if you are using **`nsna_ping`**, you need to change to using ARP pings.

PROCEDURE 9: MANAGING UNUSED INTERFACES

By default, NetworkManager manages all network interfaces available on the system. This may sometimes cause issues in some configurations. In this case, you can configure NetworkManager to not manage the NIC by adding them to **`/etc/NetworkManager/conf.d/unmanaged-devices.conf`**.

- To ignore eth1 completely,including across reboots, add the following content to **`/etc/NetworkManager/conf.d/unmanaged-devices.conf`**:

```
[keyfile]
unmanaged-devices=interface-name:eth1
```

5 Modifying network connections

You can use the **`nmcli connection modify`** command to modify network connections. The generic syntax of the command follows:

```
> sudo nmcli connection modify
      CONNECTION-NAME PROPERTY VALUE
```

To obtain the value of *`CONNECTION-NAME`*, list connections using the command: **`nmcli connection show`**. Available properties and their possible values are described in the following section.

5.1 Connections attributes

This section lists and describes attributes you can modify on connections:

TABLE 1: CONNECTIONS DETAILS

Property	Description	Values
ipv4.method	It defines how the interface obtains and handles its IPv4 address configuration	• `auto`—the default value used to allocate IP

Property	Description	Values
		addresses dynamically using DHCP • `manual`—to configure a static IP address You must set `ipv4.address1` if you set `ipv4.method` as `manual`. • `link-local`—to use IPv4 link-local addressing only (169.254.0.0/16) • `shared`—to share the connection with other computers • `disabled`—to disable IPv4.
ipv4.dns	A space separated list of DNS IP addresses	For example, "8.8.8.8 8.8.4.4"
ipv4.gateway	The property is the router address that your system uses to reach networks beyond your local network	An IP address of the gateway
connection.id	It renames the connection	a string representing the new connection name

Property	Description	Values
802-11-wireless.ssid	The property renames a Wi-Fi network	A string representing the new Wi-Fi SSID
connection.autoconnect	Toggles on/off automatic connection when the device is online	`on` or `off`

6 Establishing and terminating network connections

Enable the network connection to access and connect devices and resources.

6.1 Enabling network connections

Particular connections may be disabled, or all of the connections may be disabled. To activate all connections, run the command:

```
> sudo nmcli networking on
```

Bear in mind that the command does not activate manually disabled connections. To activate such a connection, proceed as described in the following procedure:

PROCEDURE 10: ACTIVATING AN EXISTING CONNECTION

1. View the list of existing connections:

   ```
   > sudo nmcli connection show
   ```

2. Enable a connection using the name or UUID:

   ```
   > sudo nmcli connection up uuid CONNECTION-UUID
   ```

3. Check the connection status:

   ```
   > sudo nmcli connection show --active
   ```

6.2 Disabling network connections

You can temporarily disconnect your system from external networks using the **nmcli** command to disable your network connection.

To disable a particular connection, proceed as described in the following procedure:

1. View the list of active connections:

```
> sudo nmcli connection show --active

NAME                UUID                                  TYPE      DEVICE
Wired connection 2  7b0c32ee-851e-3015-a658-f4372b426273  ethernet  enp6s0f3u1u4
lo                  8041ed2a-e4ae-4bf7-a0db-d4b513b9d745  loopback  lo
virbr0              1bec1271-4bb6-46a7-a50a-4b329fa318d3  bridge    virbr0
vnet0               ac196c81-0505-49f0-9328-b78e1746b4a9  tun       vnet0
```

2. Terminate a specific connection:

```
> sudo nmcli connection down CONNECTION-NAME/CONNECTION-UUID
```

For example:

```
> sudo nmcli connection down virbr0
```

To disable **all** connections temporarily, run:

```
> sudo nmcli networking off
```



Note: Temporal changes only

In both cases, the changes persist only till the system reboots. After reboot, NetworkManager and connections are active again.

7 Monitoring network connections

Use the **nmcli** command to view the status, activity and details of network connections managed by NetworkManager.

The following list provides commands for basic monitoring of network connections.

View the list of active connections

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

View the status of NetworkManager

```
> nmcli monitor
```

The real-time updates about the network states and connections appear.

View details of a specific network connection

```
> nmcli connection monitor CONNECTION-NAME
```

Each time the connection changes, NetworkManager prints a line.

Monitor the status of network devices

```
> nmcli device monitor
```

The list of all network devices with device name, type, state and connection name appears.

View the signal strength of Wi-Fi connections

```
> nmcli device wifi list
```

The list of available Wi-Fi networks with SSID, signal strength (in %) and security type appears.

8 NetworkManager logging

NetworkManager activities are logged by the `journald` system logging mechanism. The NetworkManager logs are saved in `/var/log/syslog`, and you can access the details using the **`journalctl`** command.

The type of logged NetworkManager activities differs according to the current logging level. Available levels are described below:

- *ERR*—logs only error messages. For example, connection failures.
- *WARN*—logs warnings and errors. For example, authentication issues.
- *INFO*—logs informational messages. That is the default level for all logging domains.
- *DEBUG*—logs detailed debugging information. For example, detailed DHCP negotiations
- *TRACE*—logs very detailed, usually unimportant events. For example, packet-level details.

To check the current logging level, run:

```
> nmcli general logging
INFO    PLATFORM,RFKILL,ETHER,WIFI,BT,MB,DHCP4,DHCP6,PPP,IP4,IP6,AUTOIP4,DNS,VPN,SHARING,
SUPPLICANT,AGENTS,SETTINGS,SUSPEND,CORE,DEVICE,OLPC,INFINIBAND,FIREWALL,ADSL,BOND,VLAN,
BRIDGE,TEAM,CONCHECK,DCB,DISPATCH,AUDIT,SYSTEMD,PROXY
```

The output shows that the logging level is INFO for all domains—that is the default setting. You can modify logging levels on particular domains and then the command outputs only the modified domains.

To change the logging level on all domains, run:

```
> sudo general logging level LEVEL domains ALL
```

For example, to revert changes to the default setting:

```
> sudo nmcli general logging level INFO domains ALL
```

To change a logging level on particular domains, for example, to set `DEBUG` on `DNS` and `FIREWALL`, run:

```
> sudo nmcli general logging level DEBUG domains FIREWALL,DNS
```

The following list provides commands to manage NetworkManager logs using **`journald`**.

MANAGING NETWORKMANAGER LOGS

Viewing logs

To view the NetworkManager logs

```
> sudo journalctl -u NetworkManager
```

To view the NetworkManager logs in real time:

```
> sudo journalctl -u NetworkManager -f
```

To view only specific logs, use **`grep`** to filter the **`journalctl`** output. For example, for log related to DHCP, run:

```
> sudo journalctl -u NetworkManager | grep DHCP
```

Saving logs

To save NetworkManager logs to a file, for example, to `networkmanager.log`:

```
> sudo journalctl -u NetworkManager > networkmanager.log
```

To save the NetworkManager logs of a specific time to a file, for example, to `networkmanager_timerange.log`:

```
> sudo journalctl -u NetworkManager --since "YYYY-MM-DD HH:MM:SS" --until "YYYY-MM-DD HH:MM:SS" > networkmanager_timerange.log
```

To save the NetworkManager logs with real-time monitoring and save them as and when generated:

```
> sudo journalctl -u NetworkManager -f >> live_networkmanager.log
```

9 The **nmcli** command reference

This section provides a summary of options and subcommands of the **nmcli** command you can use to interact with NetworkManager daemon to manage the network.

The **nmcli** command has the following generic syntax:

```
# nmcli OPTIONS SUBCOMMAND SUBCOMMAND_ARGUMENTS
```

where *OPTIONS* are described in *Section 9.1, "The **nmcli** command options"* and *SUBCOMMAND* can be any of the following:

connection
: enables you to configure your network connection. For details, refer to *Section 9.2, "The **connection** subcommand"*.

device
: used for network device management. For details, refer to *Section 9.3, "The **device** subcommand"*.

general
: shows status and permissions. For details refer to *Section 9.4, "The **general** subcommand"*.

monitor
: monitors activity of NetworkManager and watches for changes in the state of connectivity and devices. This subcommand does not take any arguments.

networking
: queries the networking status. For details, refer to *Section 9.5, "The **networking** subcommand"*.

9.1 The **nmcli** command options

Besides the subcommands and their arguments, the **nmcli** command can take the following options:

`-a|--ask`
: The command stops its run to ask for any missing arguments, for example, for a password to connect to a network.

`-c|--color {yes|no|auto}`

controls the color output: `yes` to enable the colors, `no` to disable them, and `auto` creates color output only when the standard output is directed to a terminal.

`-m|--mode {tabular|multiline}`

switches between `tabular` (each line describes a single entry, columns define particular properties of the entry) and `multiline` (each entry comprises more lines, each property is on its own line). `tabular` is the default value.

`-h|--help`

prints help.

`-w|--wait seconds`

sets a time-out period for which to wait for NetworkManager to finish operations. Using this option is recommended for commands that might take longer to complete, for example, connection activation.

9.2 The **`connection`** subcommand

The **`connection`** command enables you to manage connections or view any information about particular connections. The **`nmcli connection`** provides the following commands to manage your network connections:

`show`

to list connections:

```
> nmcli connection show
```

You can also use this command to show details about a specified connection:

```
> nmcli connection show CONNECTION_ID
```

where *`CONNECTION_ID`* is any of the identifiers: *a connection name, UUID or a path*

`up`

to activate the provided connection. Use the command to reload a connection. Also run this command after you perform any change to the connection.

```
> sudo nmcli connection up [--active] [CONNECTION_ID]
```

When `--active` is specified, only the active profiles are displayed. The default is to display both active connections and static configuration.

down

to deactivate a connection.

```
> sudo nmcli connection down CONNECTION_ID
```

where: *CONNECTION_ID* is any of the identifiers: *a connection name, UUID or a path*
If you deactivate the connection, it will not reconnect later even if it has the `autoconnect` flag.

modify

to change or delete a property of a connection.

```
> sudo nmcli connection modify CONNECTION_ID SETTING.PROPERTY PROPERTY_VALUE
```

where:

- *CONNECTION_ID* is any of the identifiers: *a connection name, UUID, or a path*
- *SETTING.PROPERTY* is the name of the property, for example, `ipv4.addresses`
- *PROPERTY_VALUE* is the desired value of *SETTING.PROPERTY*

The following example deactivates the `autoconnect` option on the `ethernet1` connection:

```
> sudo nmcli connection modify ethernet1 connection.autoconnect no
```

add

to add a connection with the provided details. The command syntax is similar to the **modify** command:

```
> sudo nmcli connection add CONNECTION_ID save YES|
NO SETTING.PROPERTY PROPERTY_VALUE
```

You should at least specify a `connection.type` or use `type`. The following example adds an Ethernet connection tied to the `eth0` interface with DHCP and disables the connection's `autoconnect` flag:

```
> sudo nmcli connection add type ethernet autoconnect no ifname eth0
```

edit

to edit an existing connection using an interactive editor.

```
> sudo nmcli connection edit CONNECTION_ID
```

clone

to clone an existing connection. The minimal syntax follows:

```
> sudo nmcli connection clone CONNECTION_ID NEW_NAME
```

where *CONNECTION_ID* is the connection to be cloned.

delete

to delete an existing connection:

```
> sudo nmcli connection delete CONNECTION_ID
```

monitor

to monitor the provided connection. Each time the connection changes, NetworkManager prints a line.

```
> sudo nmcli connection monitor CONNECTION_ID
```

reload

to reload all connection files from the disk. As NetworkManager does not monitor changes performed to the connection files, you need to use this command whenever you make changes to the files. This command does not take any further subcommands.

load

to load/reload a particular connection file, run:

```
> sudo nmcli connection load CONNECTION_FILE
```

For details about the above-mentioned commands, refer to the **nmcli** documentation (https://networkmanager.dev/docs/api/latest/nmcli.html).

9.3 The **device** subcommand

The **device** subcommand enables you to show and manage network interfaces. The **nmcli device** command recognizes the following commands:

status

to print the status of all devices.

```
> nmcli device status
```

show

shows detailed information about a device. If no device is specified, all devices are displayed.

```
> mcli device show [DEVICE_NAME]
```

connect

to connect a device. NetworkManager tries to find a suitable connection to activate. If there is no compatible connection, a new profile is created.

```
> sudo nmcli device connect DEVICE_NAME
```

modify

performs temporary changes to the configuration that is active on the particular device. The changes are not stored in the connection profile.

```
> sudo nmcli device modify DEVICE_NAME [+|-] SETTING.PROPERTY VALUE
```

For possible *SETTING.PROPERTY* values, refer to *nm-settings-nmcli(5)*.

The example below starts the IPv4 shared connection sharing on the device `con1`.

```
> sudo nmcli dev modify con1 ipv4.method shared
```

disconnect

disconnects a device and prevents the device from automatically activating further connections without manual intervention.

```
> sudo nmcli device disconnect DEVICE_NAME
```

delete

to delete the interface from the system. You can use the command to delete only software devices like bonds and bridges. You cannot delete hardware devices with this command.

```
> sudo nmcli device delete DEVICE_NAME
```

wifi

lists all available access points.

```
> nmcli device wifi
```

wifi connect

connects to a Wi-Fi network specified by its SSID or BSSID. The command takes the following options:

- `password` - password for secured networks
- `ifname` - interface used for activation
- `name` - you can give the connection a name

```
> sudo nmcli device wifi connect SSID [password PASSWORD_VALUE]
 [ifname INTERFACE_NAME]
```

To connect to a Wi-Fi *GUESTWiFi* with a password `pass$word2#@@`, run:

```
> sudo nmcli device wifi connect GUESTWiFi password pass$word2#@@
```

9.4 The **general** subcommand

You can use this command to view NetworkManager status and permissions, and change the host name and logging level. The **`nmcli general`** recognizes the following commands:

`status`

displays the overall status of NetworkManager. Whenever you do not specify a command to the **`nmcli general`** command, status is used by default.

```
> nmcli general status
```

`hostname`

if you do not provide a new host name as an argument, the current host name is displayed. If you specify a new host name, the value is used to set a new host name.

```
> sudo nmcli general hostname [HOSTNAME]
```

For example, to set `MyHostname`, run:

```
> sudo nmcli general hostname MyHostname
```

`permissions`

shows your permission for NetworkManager operations like enabling or disabling networking, modifying connections, etc.

```
> nmcli general permissions
```

`logging`

shows and changes NetworkManager logging levels and domains. Without any arguments, the command displays current logging levels and domains.

```
> sudo nmcli general logging [level LEVEL domains DOMAIN]
```

`LEVEL` is any of the values: `OFF`, `ERR`, `WARN`, `INFO`, `DEBUG`, or `TRACE`.
`DOMAIN` is a list of values that can be as follows: `PLATFORM, RFKILL, ETHER, WIFI, BT, MB, DHCP4, DHCP6, PPP, WIFI_SCAN, IP4, IP6, AUTOIP4, DNS, VPN, SHARING, SUPPLICANT, AGENTS, SETTINGS, SUSPEND, CORE, DEVICE, OLPC, WIMAX, INFINIBAND, FIREWALL, ADSL, BOND, VLAN, BRIDGE, DBUS_PROPS, TEAM, CONCHECK, DCB, DISPATCH, AUDIT, SYSTEMD, VPN_PLUGIN, PROXY`.

9.5 The `networking` subcommand

The subcommand enables you to query the status of the network. Also, by using this command, you can enable or disable networking. The **`nmcli networking`** command takes the following commands:

`on/off`

enables or disables networking. The **`off`** command deactivates all interfaces managed by NetworkManager.

```
> sudo nmcli networking on
```

connectivity

displays the network connectivity state. If `check` is used, NetworkManager performs a new check of the state. Otherwise, the last detected state is displayed.

```
> nmcli networking connectivity
```

Possible states are the following:

- *none* - the host is not connected to any network.
- *portal* - the host is behind a captive portal and cannot reach the full Internet.
- *limited* - the host is connected to a network, but it has no access to the Internet.
- *full* - the host is connected to a network and has full access to the Internet.
- *unknown* - NetworkManager could not determine the network state.

10 Troubleshooting

Learn how to debug and troubleshoot NetworkManager installation and configuration issues.

10.1 Network is not running

If the network is not working, this may be caused by NetworkManager itself. To check that, proceed as follows:

1. Check that NetworkManager is enabled and active:

```
> sudo systemctl status network
```

2. If NetworkManager is disabled, enable it:

```
> sudo systemctl enable NetworkManager
```

3. If NetworkManager is inactive, restart it:

```
> sudo restart NetworkManager
```

10.2 Wi-Fi connectivity issue

If you are experiencing problems with Wi-Fi connectivity, proceed as described below:

1. View the list of Wi-Fi connections:

```
> sudo nmcli device wifi list
```

2. If a particular device is listed, make sure its Wi-Fi connection is active:

```
> sudo nmcli connection show --active
```

3. If the Wi-Fi connection is not listed, verify the Wi-Fi status:

```
> sudo nmcli device status
```

   a. If the status is `disconnected`, activate the connection.

```
> sudo nmcli connection up CONNECTION_NAME
```

   b. If the status is `unavailable`, restart NetworkManager:

```
> sudo systemctl restart NetworkManager
```

4. Inspect the NetworkManager logs for error messages.

```
> sudo journalctl -u NetworkManager -n 100
```

5. The Wi-Fi device may be blocked:

   a. Check if the device is blocked:

```
> sudo rfkill list

phy0: Wireless LAN
```

```
Soft blocked: yes
Hard blocked: no
```

b. Unblock the device:

```
> sudo rfkill unblock all
```

6. IPv4 static address may be configured incorrectly, to check that, reset to use DHCP:

```
> sudo nmcli connection modify SSID ipv4.method auto
```

7. Try to activate the connection again:

```
> sudo nmcli connection up SSID
```

8. Try to reconnect to the Wi-Fi network:

```
> sudo nmcli device wifi connect SSID password PASSWORD
```

10.3 Network bonding issues

To troubleshoot network bonding issues, verify connection status, check the bonding interface's status, and restart network services using the **`nmcli`** command.

PROCEDURE 11: NETWORK BONDING TROUBLESHOOTING

1. List and view status of connections:

```
nmcli connection status
```

2. Check the status of network devices, including the bonded interface.

```
nmcli device status
```

3. Check the bonding interface status in `/proc/net/bonding/bond0`
   This file provides information about the bonding mode, active slaves, and other relevant details.

4. If the connections are inactive, activate the connections:

```
nmcli con up connectionname
```

5. Modify the connection if required.

```
nmcli con edit connectionname
```

6. Restart the connection.

```
> sudo  nmcli connection up bond0
```

10.4 Network teaming issues

Verify the error messages related to network teaming in `/var/log/messages`.

You can troubleshoot network teaming issues by analyzing the devices and team connection details, enabling the device if it is disabled, and modifying the bonding mode if required. After making any changes, reload the network team connection and restart NetworkManager.

PROCEDURE 12: NETWORK TEAMING TROUBLESHOOTING

1. View the list of devices:

```
nmcli device
```

2. View the team connection details:

```
nmcli connection show teamname
```

3. Enable interface:

```
nmcli connection modify teamname bond.options "mode=active-backup"
```

4. Reload the network team connection:

```
nmcli connection reload teamname
```

5. Restart the NetworkManager.

```
> sudo systemctl restart NetworkManager.service
```

11 Legal Notice

Copyright© 2006– 2026 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

For SUSE trademarks, see https://www.suse.com/company/legal/. All other third-party trademarks are the property of their respective owners. Trademark symbols (®, ™ etc.) denote trademarks of SUSE and its affiliates. Asterisks (*) denote third-party trademarks.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

A GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The

"Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.