# **How to Setup and Configure Network Bonding or Teaming in RHEL/CentOS 7 – Part 11**

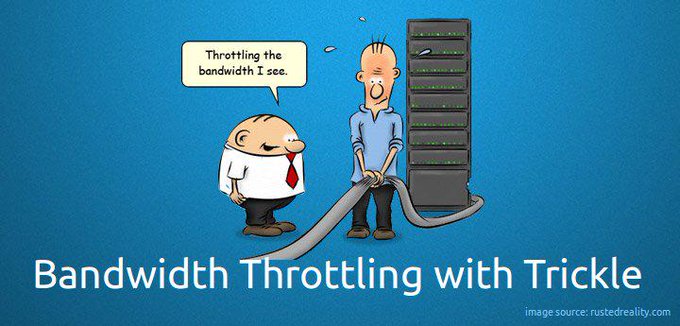
by [Gabriel Cánepa](https://www.tecmint.com/author/gacanepa/) | Published: February 13, 2016 | Last Updated: April 18, 2018

 Download Your Free eBooks NOW - [10 Free Linux eBooks for Administrators](https://www.tecmint.com/10-useful-free-linux-ebooks-for-newbies-and-administrators/) | [4 Free Shell Scripting eBooks](https://www.tecmint.com/free-linux-shell-scripting-books/)

When a system administrator wants to increase the bandwidth available and provide redundancy and load balancing for data transfers, a kernel feature known as network **bonding** allows to get the job done in a cost-effective way.

Read more about how to increase or bandwidth throttling in Linux

[View image on Twitter](https://twitter.com/tecmint/status/644355842305687552/photo/1)

[](https://twitter.com/tecmint/status/644355842305687552/photo/1)

[https://pbs.twimg.com/profile_images/482496825115422720/8EcVTA3M_normal.jpeg](https://twitter.com/tecmint)

**[Linux Inside: The Ideal Blog for Sysadmins & Geeks](https://twitter.com/tecmint)**[@tecmint](https://twitter.com/tecmint)

How to Limit the Network Bandwidth Used by Applications in a Linux with Trickle[http://www.tecmint.com/manage-and-limit-downloadupload-bandwidth-with-trickle-in-linux/ …](http://t.co/It2ccJeAih)  
Via [@tecmint](https://twitter.com/tecmint)

[8](https://twitter.com/intent/like?tweet_id=644355842305687552" \o "Like)

[9:12 AM - Sep 17, 2015](https://twitter.com/tecmint/status/644355842305687552)

[See Linux Inside: The Ideal Blog for Sysadmins & Geeks's other Tweets](https://twitter.com/tecmint" \o "View Linux Inside: The Ideal Blog for Sysadmins & Geeks's profile on Twitter)

[Twitter Ads info and privacy](https://support.twitter.com/articles/20175256)

In simple words, bonding means aggregating two or more physical network interfaces (called slaves) into a single, logical one (called master). If a specific NIC (Network Interface Card) experiences a problem, communications are not affected significantly as long as the other(s) remain active.

Read more about network bonding in Linux systems here:

1. [Network Teaming or NiC Bondin in RHEL/CentOS 6/5](https://www.tecmint.com/ethernet-channel-bonding-aka-nic-teaming-on-linux-systems/)
2. [Network NIC Bonding or Teaming on Debian based Systems](https://www.tecmint.com/network-nic-bonding-teaming-in-debian-linux/)
3. [How to Configure Network Bonding or Teaming in Ubuntu](https://www.tecmint.com/configure-network-bonding-teaming-in-ubuntu/)

### Enabling and Configuring Network Bonding or Teaming

By default, the bonding kernel module is not enabled. Thus, we will need to load it and ensure it is persistent across boots. When used with the **--first-time** option, **modprobe** will alert us if loading the module fails:

# modprobe --first-time bonding

The above command will load the bonding module for the current session. In order to ensure persistency, create a **.conf** file inside **/etc/modules-load.d** with a descriptive name, such as **/etc/modules-load.d/bonding.conf**:

# echo "# Load the bonding kernel module at boot" > /etc/modules-load.d/bonding.conf

# echo "bonding" >> /etc/modules-load.d/bonding.conf

Now reboot your server and once it restarts, make sure the bonding module is loaded automatically, as seen in **Fig. 1**:

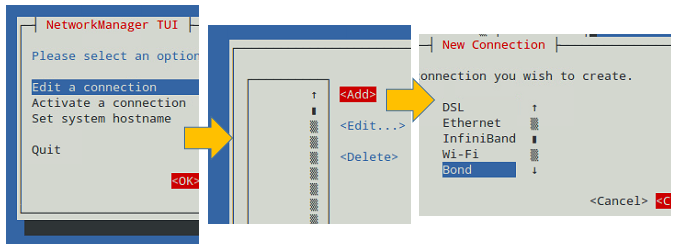
[Check Network Bonding Module Loaded in Kernel](https://www.tecmint.com/wp-content/uploads/2016/02/Check-Network-Bonding-Module.png)

*Check Network Bonding Module Loaded in Kernel*

In this article we will use 3 interfaces (**enp0s3**, **enp0s8**, and **enp0s9**) to create a bond, named conveniently **bond0**.

To create **bond0**, we can either use **nmtui**, the text interface for controlling **NetworkManager**. When invoked without arguments from the command line, **nmtui** brings up a text interface that allows you to edit an existing connection, activate a connection, or set the system hostname.

Choose **Edit connection** –> **Add** –> **Bond** as illustrated in **Fig. 2**:

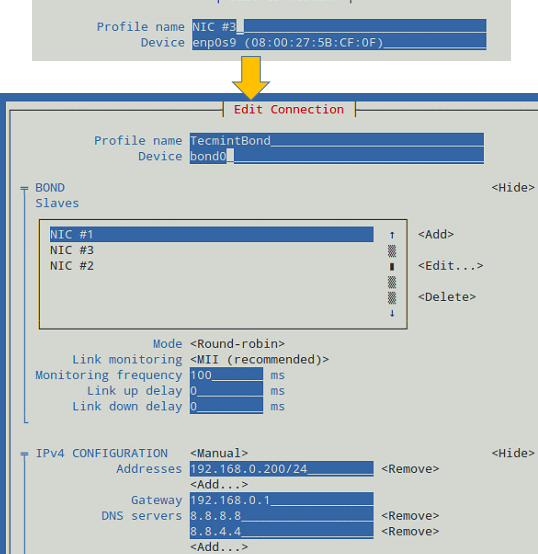
[](https://www.tecmint.com/wp-content/uploads/2016/02/Create-Network-Bonding.png)

*Create Network Bonding Channel*

In the **Edit Connection** screen, add the slave interfaces (**enp0s3**, **enp0s8**, and **enp0s9** in our case) and give them a descriptive (Profile) name (for example, **NIC #1**, **NIC #2**, and **NIC #3**, respectively).

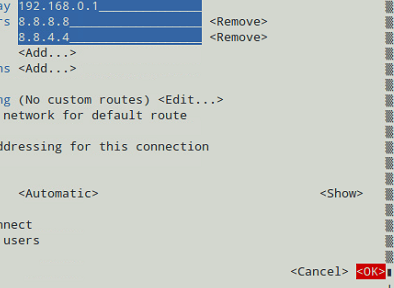
In addition, you will need to set a name and device for the bond (**TecmintBond** and **bond0** in Fig. 3, respectively) and an IP address for **bond0**, enter a gateway address, and the IPs of DNS servers.

**Note** that you do not need to enter the MAC address of each interface since **nmtui** will do that for you. You can leave all other settings as default. See Fig. 3 for more details.

[](https://www.tecmint.com/wp-content/uploads/2016/02/Network-Bonding-Teaming-Configuration.png)

*Network Bonding Teaming Configuration*

When you’re done, go to the bottom of the screen and choose **OK** (see Fig. 4):

[](https://www.tecmint.com/wp-content/uploads/2016/02/Configuration-of-bond0.png)

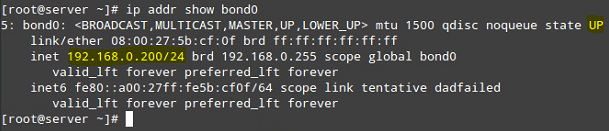
*Configuration of bond0*

And you’re done. Now you can exit the text interface and return to the command line, where you will enable the newly created interface using [ip command](https://www.tecmint.com/ip-command-examples/" \t "_blank):

# ip link set dev bond0 up

After that, you can see that **bond0** is UP and is assigned **192.168.0.200**, as seen in Fig. 5:

# ip addr show bond0

[](https://www.tecmint.com/wp-content/uploads/2016/02/Check-Network-Bond-Interface.png)

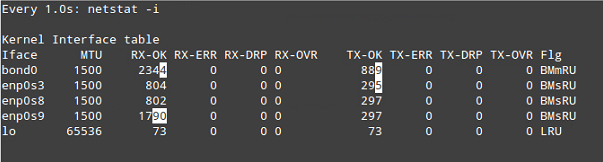
*Check Network Bond Interface Status*

### Testing Network Bonding or Teaming in Linux

To verify that **bond0** actually works, you can either ping its IP address from another machine, or what’s even better, watch the kernel interface table in real time (well, the refresh time in seconds is given by the **-n** option) to see how network traffic is distributed between the three network interfaces, as shown in Fig. 6.

The **-d** option is used to highlight changes when they occur:

# watch -d -n1 netstat -i

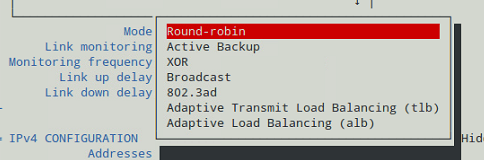
[](https://www.tecmint.com/wp-content/uploads/2016/02/Check-Kernel-Interface-Table.png)

*Check Kernel Interface Table*

It is important to note that there are several bonding modes, each with its distinguishing characteristics. They are documented in section 4.5 of the [Red Hat Enterprise Linux 7 Network Administration](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/sec-Using_Channel_Bonding.html#s3-modules-bonding-directives) guide. Depending on your needs, you will choose one or the other.

In our current setup, we chose the **Round-robin** mode (see Fig. 3), which ensures packets are transmitted beginning with the first slave in sequential order, ending with the last slave, and starting with the first again.

The Round-robin alternative is also called **mode 0**, and provides load balancing and fault tolerance. To change the bonding mode, you can use **nmtui** as explained before (see also Fig. 7):

[](https://www.tecmint.com/wp-content/uploads/2016/02/Changing-Bonding-Mode-Using-nmtui.png)

*Changing Bonding Mode Using nmtui*

If we change it to **Active Backup**, we will be prompted to choose a slave that will the only one active interface at a given time. If such card fails, one of the remaining slaves will take its place and becomes active.

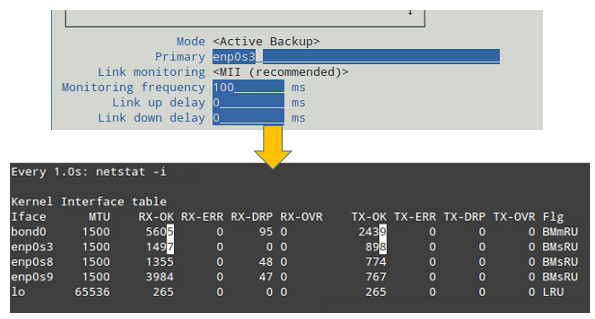
Let’s choose **enp0s3** to be the primary slave, bring **bond0** down and up again, restart the network, and display the kernel interface table (see Fig. 8).

Note how data transfers (**TX-OK** and **RX-OK**) are now being made over **enp0s3** only:

# ip link set dev bond0 down

# ip link set dev bond0 up

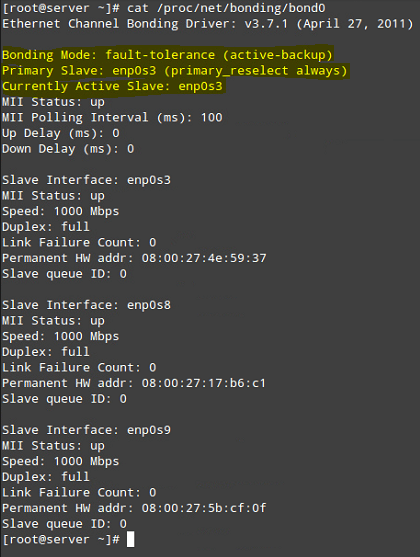
# systemctl restart network

[](https://www.tecmint.com/wp-content/uploads/2016/02/Bond-Acting-in-Active-Backup-Mode.png)

*Bond Acting in Active Backup Mode*

Alternatively, you can view the bond as the kernel sees it (see Fig. 9):

# cat /proc/net/bonding/bond0

[](https://www.tecmint.com/wp-content/uploads/2016/02/Check-Network-Bond-as-Kernel.png)

*Check Network Bond as Kernel*

### Summary

In this chapter we have discussed how to set up and configure bonding in **Red Hat Enterprise Linux 7** (also works on **CentOS 7** and **Fedora 22+**) in order to increase bandwidth along with load balancing and redundancy for data transfers.

As you take the time to explore other bonding modes, you will come to master the concepts and practice related with this topic of the certification.