Create NIC Channel Bonding in RedHat/CentOS/Fedora

by [Narad Shrestha](https://www.tecmint.com/author/navin/) | Published: July 25, 2012 | Last Updated: March 17, 2015

 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/)

Channel Bonding enables two or more network interfaces to act as one, simultaneously increasing the bandwidth and providing redundancy. This is a great way of achieving redundancy to a server. If one physical NIC is down or unplugged, it will automatically move resource to other NIC card. Channel bonding will work with the help of bonding driver in kernel. This post guides you through how to create NIC / Channel Bonding in RedHat, CentOS and Fedora Linux.

[](https://www.tecmint.com/wp-content/uploads/2012/07/Channel-Bonding-in-Linux.jpg)

*Create Channel Bonding in Linux*

How to Create NIC Channel Bonding in RedHat, CentOS and Fedora

Step 1: Creating Bonding Channel

As a root, create a new file name bonding.conf in the /etc/modprobe.d/ directory. Name can be anything you like as long as it ends with a .conf extension. Insert the following line in this new file.

alias bond0 bonding

For each configured channel bonding interface, there must be a corresponding entry in your new /etc/modprobe.d/bonding.conf file.

Step 2: Creating Channle Bonding Interface

To create a channel bonding interface, create a file in the /etc/sysconfig/network-scripts/ directory called ifcfg-bond0. The following is a sample channel bonding configuration file. (Note : IP Address may differ in your environment.)

# vi /etc/sysconfig/network-scripts/ifcfg-bond0

DEVICE=bond0

IPADDR=192.168.1.8

NETMASK=255.255.255.0

ONBOOT=yes

BOOTPROTO=none

USERCTL=no

Step 3: Configuring Channel Bonding Interface

After the channel bonding interface is created, the network interfaces to be bound together must be configured by adding the MASTER and SLAVE directives to their configuration files. The configuration files for each of the channel-bonded interfaces can be nearly identical. For example, if two Ethernet interfaces are being channel bonded, both eth0 and eth1 may look like the following example. Edit physical interface card details as under.

For eth0

# vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

USERCTL=no

ONBOOT=yes

**MASTER=bond0**

SLAVE=yes

BOOTPROTO=none

For eth1

# vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1

USERCTL=no

ONBOOT=yes

**MASTER=bond0**

SLAVE=yes

BOOTPROTO=none

The above directives are self explanatory for many system admin, however, let me explain to newbie.

1. DEVICE: Indicates what is the device name
2. USERCTL: Indicates that can user control this device(here its no)
3. ONBOOT:  Indicates that at the boot time do this device should be up?
4. MASTER: Is this device has master? Then what it is(here its bond0)
5. SLAVE: Is this device acting as slave?
6. BOOTPROTO: What about getting IP Address from DHCP? It’s set to none which indicate it’s a static IP)

Step 4: Restarting Network Service