**\*\*\*\*\*\*\*\*\*\*\*Ethernet Bonding in RHEL 7\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

RHEL 7 takes onboard naming convention of Ethernet interfaces default

Check if Bonding Kernel Module is installed

**modprobe --first-time bonding**

To display information about the module, issue the following command:

**modinfo bonding**

Create a Channel Bonding Interface

To create a channel bonding interface, create a file in the /etc/sysconfig/network-scripts/ directory called ifcfg-bond*N*, replacing *N* with the number for the interface, such as 0. The contents of the file can be based on a configuration file for whatever type of interface is getting bonded, such as an Ethernet interface. The essential differences are that the DEVICE directive is bond*N*, replacing *N* with the number for the interface, and TYPE=Bond. In addition, set BONDING\_MASTER=yes.

Vi /etc/sysconfig/network-scripts/ifcfg-bond0

DEVICE=bond0

TYPE=bond

BOOTPROTO=none

NAME=bond0

ONBOOT=yes

IPADDR=10.128.96.X

NETMASK=255.255.255.0

GATEWAY=10.128.96.X

#PREFIX=24

**BONDING\_MASTER=yes**

BONDING\_OPTS="mode=4 miimon=100 xmit\_hash\_policy=1"

################################################################

**Creating SLAVE Interfaces**

################################################################

Vi /etc/sysconfig/network-scripts/ifcfg-ens192

DEVICE=ens192

TYPE=Ethernet

BOOTPROTO=none

NAME=bond0-slave

ONBOOT=yes

MASTER=bond0

SLAVE=yes

################################################################

vi /etc/sysconfig/network-scripts/ifcfg-ens224

DEVICE=ens224

TYPE=Ethernet

BOOTPROTO=none

NAME=bond0-slave

ONBOOT=yes

MASTER=bond0

SLAVE=yes

Important

Parameters for the bonding kernel module must be specified as a space-separated list in the BONDING\_OPTS="*bonding parameters*" directive in the ifcfg-bond*N* interface file.

Do *not* specify options for the bonding device in /etc/modprobe.d/*bonding*.conf, or in the deprecated /etc/modprobe.conf file.

🡪To make NetworkManager aware of the changes, issue a command for every changed interface as root:

**nmcli con reload**

🡪to iptables stop /disable at startup

**service iptables stop**

**systemctl disable iptables**

🡪to check list of services at starttup

**systemctl list-units**

**systemctl list-unit-files**

#####################################################

systemctl stop NetworkMaanager.service

**to disable NetworkManager at startup**

systemctl disable NetworkMaanager.service

service network restart

To view the status of the bond interface, issue the following command:

# Ip link show

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Done**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* Mode 0 – Round-robin (RR) policy: Packet transmissions is done in a sequential order. For ex, the first packet is transmitted using the interface and second packet is done through next interface and so on. This mode provides load balancing and fault tolerance.
* Mode 1 – active-backup policy: Packet transmission happens only through the active interface and the traffic is routed through the backup interface only when the active interface went down.
* Mode 2 – XOR policy: This mode balances outgoing traffic across the active ports. In this mode the same slave is used for transmission for each destination MAC address. This provides load balancing and fault tolerance.
* Mode 3 – broadcast policy: This transmits everything on all slave interfaces. This mode provides fault tolerance.
* Mode 4( 802.3ad ): This mode is used to create the link aggregation groups that share the same speed and duplex settings.
* Mode 5( balance-tlb ): Adaptive transmit load balancing. The outgoing traffic is distributed based on the current load on each slave. Incoming traffic is received by the current slave. If the current slave fails, then another slave takes over the incoming traffic based on MAC address.
* Mode 6( balance-alb ): Adaptive load balancing. In this, the incoming traffic is handled to use load balancing across all the slaves. The receive load balancing is done through ARP negotiation.