Red Hat
Enterprise Linux

Advanced Networking with NetworkManager in RHEL8 for Servers

Stop touching ifcfg-eth0!

Marc Skinner
Principal Solution Architect

Red Hat

Agenda

- What / How
- Tooling
- Getting information
- Simple modifications
- Up / Down control
- Teaming
- Bridges
- VLANs
- Infiniband – IPoIB
- Examples

What is NetworkManager?

- A networking service manager
  - Manage, configure all things networking
- Introduced in RHEL6 … ho-hum
- Optional in RHEL7 … getting better
- Default in RHEL8 … I love it

What are the benefits of NetworkManager?

- Provides an API through D-BUS
  - Multiple application integration
  - Provides consistency for graphical desktop environments
- Front end to configuration files

How to use NetworkManager?

- Install
- Enable
- Start
- Configure
  - Command line
  - UI, both text and graphical

NetworkManager Tooling

- nmcli
  - Command-line
- nmtui
  - Text User Interface
- nm-connection-editor / control-center
  - Graphical User Interface
- Cockpit
  - Web administrative portal
- Ansible Role

NetworkManager Command-line

- nmcli and nmtui
  - nmcli is included in “NetworkManager”
  - nmtui is included in “NetworkManager-tui”

```
# yum –y install NetworkManager NetworkManager–tui
# systemctl enable NetworkManager
# systemclt start NetworkManager
```

- Optional – add bash completion if not installed!

```
# yum –y install bash–completion
```

Getting Information

- Is the device connected?
  # nmcli device status

```
[mskinner@p1 ~]$ nmcli dev stat
DEVICE      TYPE      STATE         CONNECTION
ens1u1      ethernet  connected     ens1u1
virbr0      bridge    connected     virbr0
enp0s31f6   ethernet  unavailable   --
lo          loopback  unmanaged     --
virbr0-nic  tun       unmanaged     --
```

- Connect a device
  # nmcli device enp3s0 connect

Getting Information

- Get connection information

```
[root@rhel8-latest ~]# nmcli connection
NAME    UUID                                  TYPE      DEVICE
enp1s0  1539f72f-269e-43f0-8539-7933690da435  ethernet  enp1s0
[root@rhel8-latest ~]#
```

```
[root@kvmu ~]# nmcli connection
NAME        UUID                                  TYPE      DEVICE
Bridge kvm  6c97e217-58ad-b10f-5b30-9aad04cf8be3  bridge    kvm
enp3s0      63aa2036-8665-f54d-9a92-c3035bad03f7  ethernet  enp2s0f0
enp4s0      b325fd44-30b3-c744-3fc9-e154b78e8c82  ethernet  enp2s0f1
Team team0  702de3eb-2e80-897c-fd52-cd0494dd8123  team      team0
vnet0       8b947e97-7f89-46a8-ad43-cbb760bb6835  tun       vnet0
vnet1       9a54ee56-4403-44d3-929f-b9a653975fd4  tun       vnet1
[root@kvmu ~]#
```

Getting Information

- Show all configurable options 120+ by default
- Use -f for filtering by field

nmcli -f “field” con show enp1s0

field = connection, ipv4, bridge, etc

```
[root@rhel8-latest ~]# nmcli connection show enp1s0
connection.id:                          enp1s0
connection.uuid:                        1539f72f-269e-43f0-8539-7933690da435
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              enp1s0
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1591372725
connection.read-only:                   no
connection.permissions:                 --
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.autoconnect-slaves:          -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.wait-device-timeout:         -1
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          no
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist:   --
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:        --
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:            --
802-3-ethernet.wake-on-lan:             default
802-3-ethernet.wake-on-lan-password:    --
ipv4.method:                            manual
ipv4.dns:                               192.168.40.15
ipv4.dns-search:                        rhlab.skinnerlabs.com
```

```
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         192.168.40.85/24
ipv4.gateway:                           192.168.40.10
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
ipv4.dhcp-client-id:                    --
ipv4.dhcp-iaid:                         --
ipv4.dhcp-timeout:                      0 (default)
ipv4.dhcp-send-hostname:                yes
ipv4.dhcp-hostname:                     --
ipv4.dhcp-fqdn:                         --
ipv4.dhcp-hostname-flags:               0x0 (none)
ipv4.never-default:                     no
ipv4.may-fail:                          no
ipv4.dad-timeout:                       -1 (default)
ipv6.method:                            auto
ipv6.dns:                               --
ipv6.dns-search:                        --
ipv6.dns-options:                       --
ipv6.dns-priority:                      0
ipv6.addresses:                         --
ipv6.gateway:                           --
ipv6.routes:                            --
ipv6.route-metric:                      -1
ipv6.route-table:                       0 (unspec)
ipv6.routing-rules:                     --
ipv6.ignore-auto-routes:                no
ipv6.ignore-auto-dns:                   no
ipv6.never-default:                     no
ipv6.may-fail:                          yes
ipv6.ip6-privacy:                       -1 (unknown)
ipv6.addr-gen-mode:                     stable-privacy
ipv6.ra-timeout:                        0 (default)
ipv6.dhcp-duid:                         --
ipv6.dhcp-iaid:                         --
ipv6.dhcp-timeout:                      0 (default)
```

```
ipv6.dhcp-send-hostname:                yes
ipv6.dhcp-hostname:                     --
ipv6.dhcp-hostname-flags:               0x0 (none)
ipv6.token:                             --
proxy.method:                           none
proxy.browser-only:                     no
proxy.pac-url:                          --
proxy.pac-script:                       --
GENERAL.NAME:                           enp1s0
GENERAL.UUID:                           1539f72f-269e-43f0-8539-7933690da435
GENERAL.DEVICES:                        enp1s0
GENERAL.IP-IFACE:                       enp1s0
GENERAL.STATE:                          activated
GENERAL.DEFAULT:                        yes
GENERAL.DEFAULT6:                       no
GENERAL.SPEC-OBJECT:                    --
GENERAL.VPN:                            no
GENERAL.DBUS-PATH:                      /org/freedesktop/NetworkManager/ActiveConnection/1
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/Settings/1
GENERAL.ZONE:                           --
GENERAL.MASTER-PATH:                    --
IP4.ADDRESS[1]:                         192.168.40.85/24
IP4.GATEWAY:                            192.168.40.10
IP4.ROUTE[1]:                           dst = 192.168.40.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 192.168.40.10, mt = 100
IP4.DNS[1]:                             192.168.40.15
IP6.ADDRESS[1]:                         fe80::fbb7:42f6:8ffe:e4f9/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh = ::, mt = 256, table=255
[root@rhel8-latest ~]#
```

Getting Information

- Show device details

```
[root@rhel8-latest ~]# nmcli device show enp1s0
GENERAL.DEVICE:                         enp1s0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         52:54:00:FE:A4:72
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     enp1s0
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/1
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.40.85/24
IP4.GATEWAY:                            192.168.40.10
IP4.ROUTE[1]:                           dst = 192.168.40.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 192.168.40.10, mt = 100
IP4.DNS[1]:                             192.168.40.15
IP6.ADDRESS[1]:                         fe80::fbb7:42f6:8ffe:e4f9/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh = ::, mt = 256, table=255
[root@rhel8-latest ~]#
```

Adding a device

- Add a new device (35 types)

```
[root@p1 ~]# nmcli con add type
6lowpan           bluetooth        cdma             infiniband       ovs-bridge       team-slave       wifi
802-11-olpc-mesh  bond             dummy            ip-tunnel        ovs-interface    tun              wifi-p2p
802-11-wireless   bond-slave       ethernet         macsec           ovs-port         vlan             wimax
802-3-ethernet    bridge           generic          macvlan          pppoe            vpn              wireguard
adsl              bridge-slave     gsm              olpc-mesh        team             vxlan            wpan
```

Simple Modifications

- Rename an interface connection name

```
# nmcli con modify Wired\ connection\ 3 con-name ens3
```

- Modify IPv4 address

```
# nmcli con modify ens3 ipv4.addresses 192.168.40.81/24
```

- Modify IPv4 gateway

```
# nmcli con modify ens3 ipv4.gateway 192.168.40.10
```

- Modify IPv4 method

```
# nmcli con modify ens3 ipv4.method static
```

Simple Modifications

- Modify IPv4 DNS

```
# nmcli con modify ens3 ipv4.dns
192.168.40.15,192.168.40.16
```

- Modify IPv4 DNS-options

```
# nmcli con modify ens3 ipv4.dns-options rotate,timeout:1
```

- Modify IPv4 DNS-search

```
# nmcli con modify ens3 ipv4.dns-search
"rhlab.skinnerlabs.com,i.skinnerlabs.com"
```

- Modify MTU

```
# nmcli con modify ens3 802-3-ethernet.mtu 9000
```

Up / Down Control

- Bring connection down
  ```
  # nmcli con down enp3s0
  ```
- Bring connection up
  ```
  # nmcli con up enp3s0
  ```
- Reload all connections
  ```
  # nmcli con reload
  ```

Teaming

- Teaming is the replacement for Bonding
  - # bond2team
- Teaming allows for N+1 NICs to be configured as a logical device with specific benefits based on the runner selected
  - broadcast (all ports)
  - roundrobin (all ports in turn)
  - activebackup (one port until failure)
  - loadbalance (all ports with a hash)
  - random (all ports randomly)
  - lacp (802.3ad LACP – requires LACP switch)

Teaming

- Create a LACP TEAM called team1 using two NICs

```
# nmcli con add type team ifname team1 con-name team1
# nmcli con modify team1 team.config '{"runner": {"name":
"lacp", "active": true, "fast_rate": true, "tx_hash":
["ipv4","tcp","udp"]}, "link_watch": {"name": "ethtool"},
"tx_balancer": { "name": "basic"}}'
# nmcli con add type ethernet con-name team1-enp10s0
ifname enp10s0 master team1
# nmcli con add type ethernet con-name team1-enp9s0 ifname
enp9s0 master team1
```

Teaming

- View teaming port status

```
# teamnl team1 ports
 2: enp10s0: up 1000Mbit FD
 3: enp9s0: up 1000Mbit FD
```

Teaming

- View teaming status/configuration

```
# teamdctl team1 state view
setup:
  runner: lacp
ports:
  enp10s0
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
    runner:
      aggregator ID: 2,
Selected
      selected: yes
      state: current
enp9s0
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
    runner:
      aggregator ID: 2, Selected
      selected: yes
      state: current
runner:
  active: yes
  fast rate: yes
```

Teaming

- View teaming configuration– lots of data

```
# teamdctl team1 state dump
# teamdctl team1 config dump
```

Bridges

- Bridge mode turns a NIC into a layer 2 switch
- Can enable/disable STP (Spanning Tree Protocol)
- Needed for multiple virtual machines
- No NAT

- Optional Bridge tooling
  - RHEL 6/7

    ```
    # yum install bridge-utils
    ```

  - RHEL 8

    ```
    # ip bridge
    ```

Bridges

- Create bridge named KVM with IP addressed assigned

```
# nmcli con add type bridge ifname kvm con-name kvm
ipv4.address 192.168.33.12/24 ipv4.method static
ipv4.gateway 192.168.33.2 ipv4.dns
"192.168.33.44,192.168.33.50,192.168.33.15" ipv4.dns-
options "rotate,timeout:1" ipv4.dns-search
"ib.skinnerlabs.com,i.skinnerlabs.com"
# nmcli con add type bridge-slave ifname enp3s0 master kvm
```

Bridges

- Create bridge named TEST with NO addressed assigned

```
# nmcli con modify enp4s0f0 ipv4.method disabled
ipv6.method ignore
# nmcli con add type bridge ifname test con-name test
# nmcli con add type bridge-slave ifname enp4s0f0 master
test
```

- Turn off STP - on by default

```
# nmcli con modify test bridge.stp no
```

Bridges

- Show bridge configuration

```
# nmcli -f bridge con show test
bridge.mac-address:                     --
bridge.stp:                             no
bridge.priority:                        32768
bridge.forward-delay:                   15
bridge.hello-time:                      2
bridge.max-age:                         20
bridge.ageing-time:                     300
bridge.group-forward-mask:              0
bridge.multicast-snooping:              yes
bridge.vlan-filtering:                  no
bridge.vlan-default-pvid:               1
bridge.vlans:                           --
```

Bridges

- Show bridge status – brief

```
# brctl show test
bridge name     bridge id               STP enabled     interfaces
test            8000.001b21514010       no              enp4s0f0
```

VLANs

- VLAN = Virtual Local Area Network
- Isolated Layer 2
- Allows for multiple isolated networks to share the same physical medium
- VLANs use VLAN IDs 0-4095
- VLAN ID 1 is default

VLANs

- Add VLAN 32 to enp3s0 with IP address

```
# nmcli connection add type vlan con-name enp3s0.32 ifname
enp3s0.32 id 32 dev enp3s0 ip4 192.168.32.11/24

# ip add | grep enp3s0.32
21: enp3s0.32@enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP>
mtu 1500 qdisc noqueue state UP group default qlen 1000
    inet 192.168.32.11/24 brd 192.168.32.255 scope global
noprefixroute enp3s0.32
```

- Add VLAN **60** to existing team10 with no IP address

```
# nmcli connection add type vlan con-name team10.60 ifname
team10.60 id 60 dev team10 ipv4.method disabled
ipv6.method ignore
```

Infiniband / IPoIB (InternetProtcol over Infiniband)

- Infiniband is specialized low latency, high performance networking gear
- Typically run in native RDMA (Remote Direct Memory Access) mode
  - Applications must be able to understand RDMA
- Can run in IPoIB mode (more compatible)
  - Applications can use traditional IP

Infiniband - IPoIB

- Create IPoIB on Infiniband NIC with MTU 65520

```
# nmcli connection delete ib0
# nmcli connection add type infiniband con-name ib0 ifname
ib0 transport-mode connected mtu 65520
# nmcli connection modify ib0 ipv4.addresses
192.168.103.50/24
# nmcli connection modify ib0 ipv4.method static
```

Infiniband – IpoIB

- Validate IP Information

```
# ip addr │ grep ib0
8: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc
pfifo_fast state UP group default qlen 256
    inet 192.168.103.12/24 brd 192.168.103.255 scope
global noprefixroute ib0
```

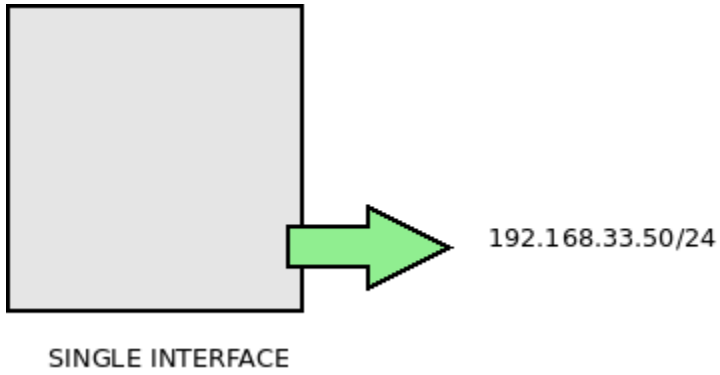
Infiniband – IpoIB – Tooling

- Status tooling

```
# yum install infiniband-diags
# ibstatus
Infiniband device 'mthca0' port 1 status:
        default gid:
fe80:0000:0000:0000:0002:c902:002a:ad39
        base lid:        0x4
        sm lid:          0x1
        state:           4: ACTIVE
        phys state:      5: LinkUp
        rate:            20 Gb/sec (4X DDR)
        link_layer:      InfiniBand
```

EXAMPLES

Single NIC with IP Address

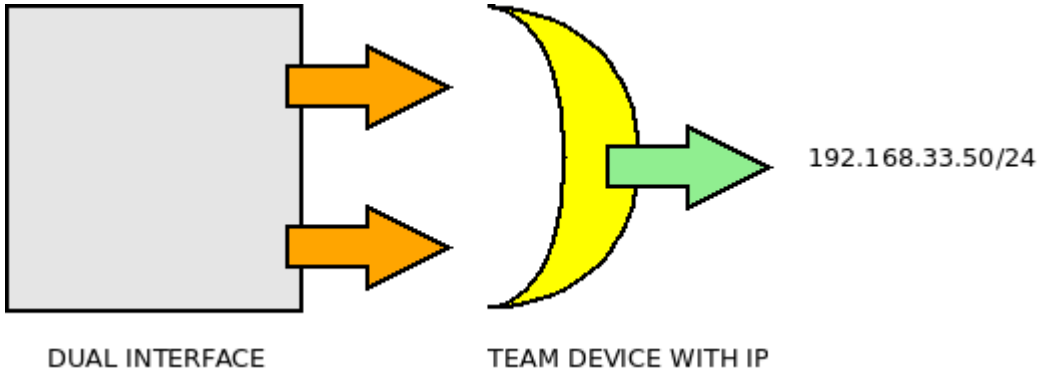


Single NIC with IP Address

- Add IP/GW/DNS/Options to NIC: ens3

```
# nmcli con modify ens3 ipv4.addresses 192.168.33.50/24
# nmcli con modify ens3 ipv4.gateway 192.168.33.1
# nmcli con modify ens3 ipv4.dns 192.168.33.15
# nmcli con modify ens3 ipv4.method static
# nmcli con modify ens3 ipv4.dns-options rotate,timeout:1
# nmcli con modify ens3 ipv4.dns-search
"ib.skinnerlabs.com,i.skinnerlabs.com"
# nmcli con modify ens3 ipv6.method ignore
```

Dual NIC with LACP TEAM with IP Address



Dual NIC with LACP TEAM with IP Address

- Disable NICs so they don't try to get IP information

```
# nmcli con modify enp6s0f0 ipv4.method disabled
ipv6.method ignore
# nmcli con modify enp6s0f1 ipv4.method disabled
ipv6.method ignore
```

- Create TEAM

```
# nmcli con add type team ifname team10 con-name team10
```

- Create TEAM runner

```
# nmcli con modify team10 team.config '{"runner": {"name":
"lacp", "active": true, "fast_rate": true, "tx_hash":
["ipv4","tcp","udp"]}, "link_watch": {"name": "ethtool"},
"tx_balancer": { "name": "basic"}}'
```

Dual NIC with LACP TEAM with IP Address

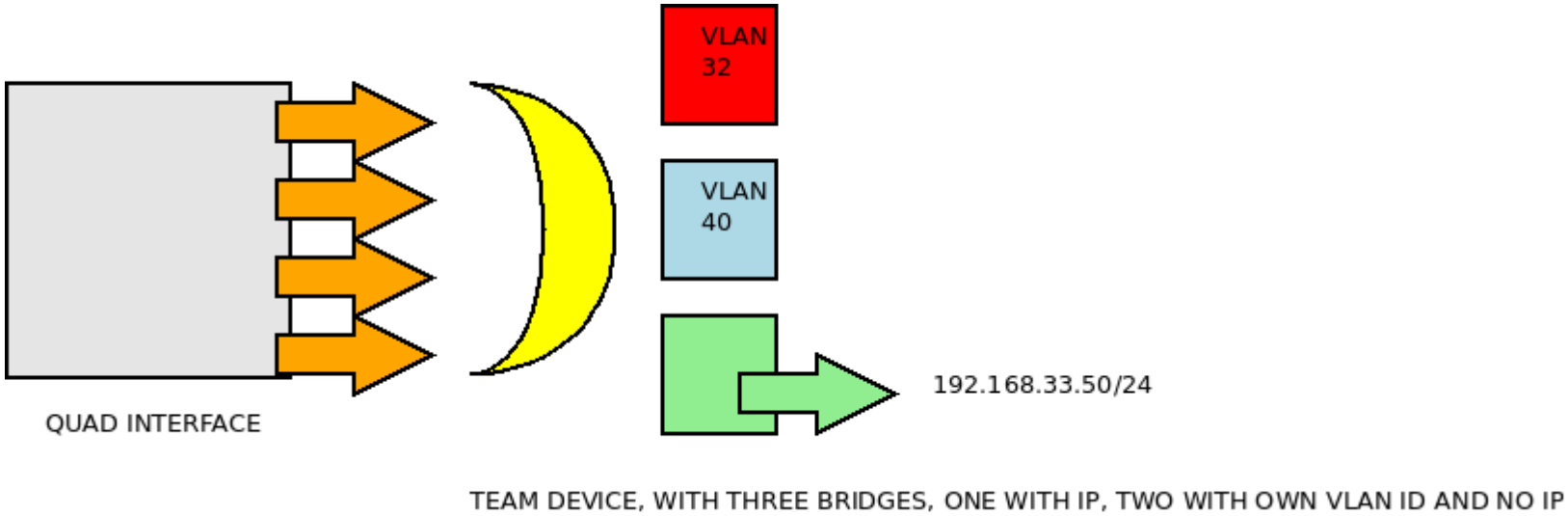
- Attach NICs to TEAM

```
# nmcli con add type ethernet con-name team10-enp6s0f0
ifname enp6s0f0 master team10
# nmcli con add type ethernet con-name team10-enp6s0f1
ifname enp6s0f1 master team10
```

- Add IP/DNS

```
# nmcli con modify team10 ipv4.addresses 192.168.33.50/24
# nmcli con modify team10 ipv4.dns 192.168.33.15
# nmcli con modify team10 ipv4.method static
```

Quad NIC with LACP TEAM with three Bridges, three VLANs, one with IP address



Quad NIC with LACP TEAM with three Bridges, three VLANs, one with IP address

- Disable NICs so they don't try to get IP information

```
# nmcli con modify enp4s0f0 ipv4.method disabled
ipv6.method ignore
# nmcli con modify enp4s0f1 ipv4.method disabled
ipv6.method ignore
# nmcli con modify enp5s0f0 ipv4.method disabled
ipv6.method ignore
# nmcli con modify enp5s0f1 ipv4.method disabled
ipv6.method ignore
```

Quad NIC with LACP TEAM with three Bridges, three VLANs, one with IP address

- Create Bridge with IP (default VLAN)
  ```
  # nmcli con add type bridge ifname kvm con-name kvm
  ipv4.address 192.168.33.50/24 ipv4.method static
  ipv4.gateway 192.168.33.1 ipv4.dns
  "192.168.33.15,192.168.33.16"
  ```
- Create TEAM and attach Bridge
  ```
  # nmcli con add type team ifname team1 con-name team1
  master kvm
  ```
- Create TEAM runner
  ```
  # nmcli con modify team1 team.config '{"runner": {"name":
  "lacp", "active": true, "fast_rate": true, "tx_hash":
  ["ipv4","tcp","udp"]}, "link_watch": {"name": "ethtool"},
  "tx_balancer": { "name": "basic"}}'
  ```

Quad NIC with LACP TEAM with three Bridges, three VLANs, one with IP address

- Attach NICs to TEAM

```
# nmcli con add type ethernet con-name team1-enp4s0f0
ifname enp4s0f0 master team1
# nmcli con add type ethernet con-name team1-enp4s0f1
ifname enp4s0f1 master team1
# nmcli con add type ethernet con-name team1-enp5s0f0
ifname enp5s0f0 master team1
# nmcli con add type ethernet con-name team1-enp5s0f1
ifname enp5s0f1 master team1
```

Quad NIC with LACP TEAM with three Bridges, three VLANs, one with IP address

- Add bridge **dmz**, then create vlan **32** and associate with bridge

```
# nmcli con add type bridge ifname dmz con-name dmz
ipv4.method disabled ipv6.method ignore
# nmcli con add type vlan con-name team1.32 dev team1 id
32 master dmz
```

- Add bridge **rhlab**, then create vlan **40** and associate with bridge

```
# nmcli con add type bridge ifname rhlab con-name rhlab
ipv4.method disabled ipv6.method ignore
# nmcli con add type vlan con-name team1.40 dev team1 id
40 master rhlab
```

Cleanup

- Delete / Cleanup an interface

```
# nmcli connection del team1 team2 team3
# nmcli connection del enp3s0
# nmcli connection del uuid 954559e9-5f8c-4f9b-b2bc-
36ff23f18d4a
```

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/
RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat