Networking continued
ip addr show --info on all interfaces
ip -s link show eno1 --- get packet stats on an int. Int now not like eth0 but (more like bsd?) dependent on HW

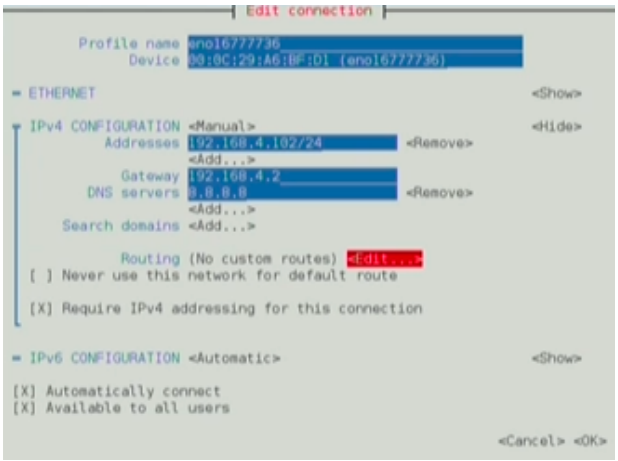
/etc/sysconfig/network-scripts
Recommended: check out scripts in /etc/sysconfig/network-scripts/ifcfg-*
They are fairly readable and there is one for each connection/ interface

Hostname is in /etc/hostname
hostnamectl set-hostname vm0.example.com -- to set the hostname
hostnamectl status -- to request the hostname
DNS resolver is pushed from /etc/sysconfig/network-scripts/ifcfg-* to /etc/resolv.conf

IPv6
ip a -------lists... same as -a? ip link show
nmcli con add con-name blah type ethernet iframe eth0 ip6 2001:db8:0:10::d000:310/64 gw6 2001:db8:0:10::1 ipv4 192.168.1.5 gw4 192.168.1.1
Remember, new connections use IP4 or IP6 with NO "v" - modifying (using a "+") DOES use a "v":
nmcli con mod blah +ipv6.dns 2001:4860:4860::8888

Setting up routing
Rather than jump into ip route commands, this goes to nmtui
NMTUI opens with edit and activate (a connection) options and "set system hostname"

Routing - "edit a connection"
Under all of the standard IPv4 options is "Routing" with an edit hyperlink
It will say "no custom routes" and give you an "add" hyperlink
It asks for destination/prefix, next hop, and metric. click ok and exit
ip route show won't show it unless you reopen nmtui and click "activate connection"
It will show up in /etc/sysconfig/network-scripts/ When you open it, all it has is this:
ADDRESS0:10.0.0.0
NETMASK0:255.255.255.0
GATEWAY0: 192.168.4.4



Virtual Bridges
eno1 to virbr0 bridges vnet0 and vnet1 virtual network interfaces, and each of those has one separate host attached
Remember that these "2 networks" are merely switched - the explanation made it sound like 2 subnets
These will be messing with vish constructs:
virsh list --all --get a list of the VM runnings
brctl show -- we see virbr0, whether stp is enabled, the bridgeid, the interfaces that are up

ip link show --baremetal lo and eno1 listed, and then virbr, vnet0 and vnet1
on the vnet0 and 1 these values are added: pfifo_fast master vibr0 state UNKNOWN mode default qlen 500
on virbr0, noqueue state UP mode DEFAULT
on eno1 - pfifo_fast state UP mode DEFAULT
All of these have the standard other stuff: <multicast, broadcast, up, lower_up>, mtu 1500 qdisk link/ether (mac address) brd ff:ff:ff:ff:ff:ff

The RHCE requirement is knowing how to set up a bridge
yum install -y bridge-utils --- You need to install this, and you have to know the current interfaces:
nmcli dev show -- Get the id of the interface, eno0166777723 for our ethernet
nmcli dev disconnect eno0166777723
nmcli con add type bridge-slave con-name br0-port1 ifname eno0166777723 master br0
-- do this to add the interface we disconnected to the bridge- It will complain the bridge isn't there, we make it below
-- you repeat this for the other interfaces that will be part of this bridge, br0
-- if you only have one interface, thats fine, two you can do load balancing etc for HP
nmcli con add type bridge con-name br0 ifname br0 -- defines the bridge
brctl show --will show your bridges
-- interesting to note, the bridge id is 8000.000000000000
Here is what we get when we check out the files in /etc/sysconfig/network-scripts/
Scared you won't remember these? **Man pages! See nmcli-examples(5) page (is at the bottom of "man nmcli")**

```
[root@server2 network-scripts]# cat ifcfg-br0
DEVICE=br0
STP=yes
BRIDGING_OPTS=priority=32768
TYPE=Bridge
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=br0
UUID=da404f2b-f967-4d06-8bb5-bca4bfca4ee0
ONBOOT=yes
[root@server2 network-scripts]# cat ifcfg-br0-port1
TYPE=Ethernet
NAME=br0-port1
UUID=91ead09a-0bfb-46da-856b-e7b10ad723d2
DEVICE=eno16777736
ONBOOT=yes
BRIDGE=br0
[root@server2 network-scripts]#
```

The nmcli-examples man page has everything you need to type, as long as you know what you are doing!!
TeamD
Network Link Aggregation - Bonding is deprecated. Teaming takes it's place.
Modes "runners": broadcast, roundrobin, activebackup (one line active, the other a backup), loadbalance, and LACP.

There are Four Steps to Network Teaming
- Create team interface
  - **nmcli con add type team con-name team0 ifname team0 config '{"runner": {"name": "loadbalance"}}'**
- Determine network configuration
  - **nmcli con mod team0 ipv4.addresses 10.0.0.10/24**
  - **nmcli con mod team0 ipv4.method manual**
- Assign the port interfaces
  - **nmcli con add type team-slave ifname eth0 master team0 con-name team0-eth0**
  - **nmcli con add type team-slave ifname eth1 master team0 con-name team0-eth1**
- Bring team and port interfaces up/down
  - **nmcli con up team0**
  - **nmcli dev dis eth0; nmcli dev dis eth1**
- Verify: **teamdctl team0 state**

Note: Above, the order in "Assign port interfaces" is wrong (see example) con-name team0-eth0 goes before ifname!

Creating a Bridge Based on Network Teams

- Doesn't work with NetworkManager enabled
- Modify the team configuration file ifcfg-team0 and add BRIDGE=brteam0 to tell it to connect to the bridge device
- Make sure no IP configuration remains in the ifcfg-team0-port files
- Manually create a bridge file:

```
DEVICE=brteam0
TYPE=Bridge
IPADDR0=192.168.122.100
PREFIX0=24
```

- Check **man 5 nmcli-examples**!

Command line example- complete teamd and link aggregation setup using teaming

```
[root@server2 ~]# nmcli con add type team con-name team0 ifname team0 config '{"runner": {"name": "loadbalance"}}'
Connection 'team0' (1c62729f-8685-4174-ae56-8e4b7daec063) successfully added.
[root@server2 ~]# nmcli con mod team0 ipv4.addresses 10.0.0.10/24
[root@server2 ~]# nmcli con mod team0 ipv4.method manual
[root@server2 ~]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:a6:bf:d1 brd ff:ff:ff:ff:ff:ff
3: eno33554992: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:a6:bf:db brd ff:ff:ff:ff:ff:ff
4: eno50332216: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN mode DEFAULT qlen 1000
    link/ether 00:0c:29:a6:bf:e5 brd ff:ff:ff:ff:ff:ff
5: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
    link/ether 00:0c:29:a6:bf:d1 brd ff:ff:ff:ff:ff:ff
6: team0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT
    link/ether 3a:f6:cb:f4:ce:5c brd ff:ff:ff:ff:ff:ff
[root@server2 ~]# nmcli con add type team-slave ifname eno33554992 master team0 con-name team0-eno33554992
Error: Unexpected argument 'con-name'
[root@server2 ~]# man nmcli-examples
[root@server2 ~]# nmcli con add type team-slave con-name eno33554992 ifname eno33554992 master team0
Connection 'eno33554992' (4bcecd02-4031-44e7-8812-51efeba4e69b) successfully added.
[root@server2 ~]# nmcli con add type team-slave con-name  eno50332216 ifname  eno50332216 master team0
Connection 'eno50332216' (c7b310c5-411b-477b-856e-d14025878a16) successfully added.
[root@server2 ~]# nmcli con up team0
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@server2 ~]# nmcli dev dis  eno50332216
Error: Device 'eno50332216' (/org/freedesktop/NetworkManager/Devices/4) disconnecting failed: This device is not active
[root@server2 ~]# nmcli dev dis eno33554992
[root@server2 ~]# teamdctl team0 state
setup:
  runner: loadbalance
[root@server2 ~]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eno16777736: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master br0 state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:a6:bf:d1 brd ff:ff:ff:ff:ff:ff
3: eno33554992: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:a6:bf:db brd ff:ff:ff:ff:ff:ff
4: eno50332216: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN mode DEFAULT qlen 1000
    link/ether 00:0c:29:a6:bf:e5 brd ff:ff:ff:ff:ff:ff
5: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
    link/ether 00:0c:29:a6:bf:d1 brd ff:ff:ff:ff:ff:ff
6: team0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT
    link/ether 3a:f6:cb:f4:ce:5c brd ff:ff:ff:ff:ff:ff
[root@server2 ~]#
```

Exercise 6

- Shut down your virtual machine. If possible, at this point make a snapshot of the virtual machine to easily restore the original configuration after the exercise
- Add a new virtual network card to the virtual machine
- Add a network team interface and call it team0 and assign both network interfaces to the team
- Put the team driver in a network bridge

- ip link
- nmcli dev dis eth0; nmcli dev dis eth1
- nmcli con add type team con-name team0 ifname team0 config '{"runner": {"name": "activebackup"}}'
- nmcli con add type team-slave con-name team0-port1 ifname eth0 master team0; repeat for eth1
- teamdctl team0 state
- At this point the team works. To put it in a bridge we need to disable NetworkManager and disable the team0 driver as the bridge will take the configuration instead of the team driver!
- nmcli dev dis team0
- systemctl stop NetworkManager; systemctl disable NetworkManager
- yum install bridge-utils(!!)
- vim /etc/sysconfig/network-scripts/ifcfg-team0; add BRIDGE=brteam0
- Remove the IP configuration from the ifcfg-team0-port files
- Create the ifcfg-brteam0 file with following contents:
  DEVICE=brteam0
  ONBOOT=yes
  TYPE=Bridge
  IPADDR0=192.168.122.100
  PREFIX0=24
- systemctl restart network