Linux Networking

GAC

How to verify Network connectivty

- Ping is the command to verify network connectivity
- Ping 192.168.0.10

Configuration NIC IP address

- NIC: Network Interface Card
- Use “ifconfig” command to determine IP address, interface devices, and change NIC configuration
- Any device use symbol to determine
  - eth0: Ethernet device number 0
  - eth1: ethernet device number 1
  - lo : local loopback device
  - Wlan0 : Wireless lan 0

How to plumb N/W card

- #ifconfig eth0 plumb
- #ifconfig eth0 unplumb

To bring N/W card up/down(active/inactive)

- #ifconfig eth0 up/down
- #ifup eth0
- #ifdown eth0

Determining NIC IP Address

```
[root@tmp]# ifconfig -a

eth0 Link encap:Ethernet HWaddr 00:08:C7:10:74:A8
BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:11 Base address:0x1820

lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:787 errors:0 dropped:0 overruns:0 frame:0
TX packets:787 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:82644 (80.7 Kb) TX bytes:82644 (80.7 Kb)
```

Changing IP Address

- We could give this eth0 interface an IP address using the ifconfig command.

```
[root@]# ifconfig eth0 192.168.0.10 netmask 255.255.255.0 up
```

- The "up" at the end of the command activates the interface.

Permanent IP configuration

- Fedora Linux also makes life a little easier with interface configuration files located in the **/etc/sysconfig/network-scripts** directory.
- Interface eth0 has a file called ifcfg-eth0, eth1 uses ifcfg-eth1, and so on.
- Admin can place your IP address information in these files

File formats for network-scripts

```
root@network-scripts]# less ifcfg-eth0

DEVICE=eth0
IPADDR=192.168.1.100
NETMASK=255.255.255.0

BOOTPROTO=static
ONBOOT=yes
#
# The following settings are optional
#

BROADCAST=192.168.1.255
NETWORK=192.168.1.0

[root@network-scripts]#
```

Getting the IP Address Using DHCP

```
[root@tmp]# cd /etc/sysconfig/network-scripts

[root@network-scripts]# less ifcfg-eth0


DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes


[root@network-scripts]#
```

Activate config change

- After change the values in the configuration files for the NIC you have to deactivate and activate it for the modifications to take effect.
- The ifdown and ifup commands can be used to do this:

```
[root@network-scripts]# ifdown eth0
[root@network-scripts]# ifup eth0
```

```
[root@tmp]# ifconfig –a

wlan0 Link encap:Ethernet HWaddr 00:06:25:09:6A:B5
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:47379 errors:0 dropped:0 overruns:0 frame:0
TX packets:107900 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:4676853 (4.4 Mb) TX bytes:43209032 (41.2 Mb)
Interrupt:11 Memory:c887a000-c887b000

wlan0:0 Link encap:Ethernet HWaddr 00:06:25:09:6A:B5
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Interrupt:11 Memory:c887a000-c887b000
```

How to View Current Routing Table

- The netstat -nr command will provide the contents of the touting table.
- Networks with a gateway of 0.0.0.0 are usually directly connected to the interface.
- No gateway is needed to reach your own directly connected interface, so a gateway address of 0.0.0.0 seems appropriate.
- The route with a destination address of 0.0.0.0 is your default gateway
- **Flags** U- means up, indicating that this is an active line. G – means this line uses a Gateway.

#natstat –nr command

```
[root@tmp]# netstat -nr

Kernel IP routing table
Destination     Gateway         Genmask         Flags MSS Window irtt Iface
172.16.68.64    172.16.69.193   255.255.255.224 UG    40     0   0    eth1
172.16.11.96    172.16.69.193   255.255.255.224 UG    40     0   0    eth1
172.16.68.32    172.16.69.193   255.255.255.224 UG    40     0   0    eth1
172.16.67.0     172.16.67.135   255.255.255.224 UG     40    0   0    eth0
172.16.69.192   0.0.0.0      255.255.255.192      U      40    0   0    eth1
172.16.67.128   0.0.0.0      255.255.255.128      U     40     0   0    eth0
172.160.0       172.16.67.135   255.255.0.0             UG    40     0   0    eth0
172.16.0.0      172.16.67.131   255.240.0.0             UG    40     0   0    eth0
127.0.0.0   0.0.0.0      255.0.0.0               U     40     0   0    lo
0.0.0.0     172.16.69.193 0.0.0.0         UG    40 0   0    eth1
[root@tmp]#
```

How to Change Default Gateway

```
[root@tmp]# route add default gw 192.168.1.0 eth0
```

- In this case, make sure that the router/firewall with IP address 192.168.1.0 is connected to the same network as interface wlan0
- Once done, you'll need to update **"/etc/sysconfig/network"** file to reflect the change. This file is used to configure your default gateway each time Linux boots.

```
NETWORKING=yes
HOSTNAME=REDHAT
GATEWAY=192.168.1.0
```

How to Delete a Route

```
[root@tmp]# route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 wlan0
```

Linux router

- Router/firewall appliances that provide basic Internet connectivity for a small office or home network are becoming more affordable every day
- when budgets are tight you might want to consider modifying an existing Linux server to be a router

How to find the hostname ?

- #hostname
- #uname –a
- **To change the servername(temporary)**
- #hostname <newname>
- #hostname
- **To make Hostname Permanent**
- /etc/sysconfig/network
- /etc/hosts

Configuring /etc/hosts File

- The /etc/hosts file is just a list of IP addresses and their corresponding server names.
- Your server will typically check this file before referencing DNS. If the name is found with a corresponding IP address then DNS won't be queried at all.
- Unfortunately, if the IP address for that host changes, you also have to also update the file. This may not be much of a concern for a single server, but can become laborious if it has to be done companywide.
- Use a centralized DNS server to handle most of the rest. Sometimes you might not be the one managing the DNS server, and in such cases it may be easier to add a quick /etc/hosts file entry till the centralized change can be made.

/etc/hosts

```
192.168.1.101  smallfry
```

- You can also add aliases to the end of the line which enable you to refer to the server using other names. Here we have set it up so that smallfry can also be accessed using the names tiny and littleguy.

```
192.168.1.101  smallfry  tiny  littleguy
```

/etc/hosts

- You should never have an IP address more than once in this file because Linux will use only the values in the first entry it finds.

```
192.168.1.101  smallfry    # (Wrong)
192.168.1.101  tiny        # (Wrong)
192.168.1.101  littleguy   # (Wrong)
```

Simple Network Troubleshooting

Sources of Network Slowness

- NIC duplex and speed incompatibilities
- Network congestion
- Poor routing
- Bad cabling
- Electrical interference
- An overloaded server at the remote end of the connection
- Misconfigured DNS

Sources of a Lack of Connectivity

- All sources of slowness can become so severe that connectivity is lost. Additional sources of disconnections are:
  - Power failures
  - The remote server or an application on the remote server being shut down.

Doing Basic Cable and Link Tests

- Server won't be able to communicate with any other device on network unless the NIC's "link" light is on. This indicates that the connection between server and the switch/router is functioning correctly.
- In most cases a lack of link is due to the wrong cable type being used. There are two types of Ethernet cables crossover and straight-through. Always make sure you are using the correct type.

Other sources of link failure

- Other sources of link failure include:
  - The cables are bad.
  - The switch or router to which the server is connected is powered down.
  - The cables aren't plugged in properly.
  - If you have an extensive network, investment in a battery-operated cable tester for basic connectivity testing is invaluable. More sophisticated models in the market will be able to tell you the approximate location of a cable break and whether an Ethernet cable is too long to be used

Viewing Activated Interfaces

- The **ifconfig command** without any arguments gives all the active interfaces on the system.
- Interfaces will not appear if they are shut down.
- The **ifconfig -a command** provides all the network interfaces, whether they are functional or not.
- Interfaces that are shut down by the systems administrator or are nonfunctional will not show an IP address line and the word **UP** will not show in the second line of the output

Viewing Activated Interfaces

- Shutdown interface

```
wlan0    Link encap:Ethernet  HWaddr 00:06:25:09:6A:D7
         BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:2924 errors:0 dropped:0 overruns:0 frame:0
         TX packets:2287 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:100
         RX bytes:180948 (176.7 Kb)  TX bytes:166377 (162.4 Kb)
         Interrupt:10 Memory:c88b5000-c88b6000
```

- Active interface

```
wlan0    Link encap:Ethernet  HWaddr 00:06:25:09:6A:D7
         inet addr:216.10.119.243  Bcast:216.10.119.255
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:2924 errors:0 dropped:0 overruns:0 frame:0
         TX packets:2295 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:100
         RX bytes:180948 (176.7 Kb)  TX bytes:166521 (162.6 Kb)
         Interrupt:10 Memory:c88b5000-c88b6000
```

Using mii-tool

- The “mii-tool” command is the original Linux tools for setting the speed and duplex of NIC card.
- It is destined to be deprecated and replaced by the newer ethtool command, but many older NICs support only mii-tool.
- Issuing the command without any arguments gives a brief status report

```
[root@rose ~]# mii-tool
eth0: negotiated 100baseTx-FD, link ok
eth1: negotiated 100baseTx-FD, link ok
[root@rose ~]#
```

#mii-tool -v

- By using the verbose mode -v switch can get much more information. In this case, negotiation was OK, with the NIC selecting 100Mbps, full duplex mode (FD):

```
[root@rose ~]# mii-tool -v
eth0: negotiated 100baseTx-FD, link ok
  product info: vendor 00:00:00, model 0 rev 0
  basic mode:  autonegotiation enabled
  basic status: autonegotiation complete, link ok
  capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising:  100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
eth1: negotiated 100baseTx-FD, link ok
  product info: Intel 82555 rev 4
  basic mode:  autonegotiation enabled
  basic status: autonegotiation complete, link ok
  capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising:  100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
  link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
[root@rose ~]#
```

Using ethtool

- The ethtool command is slated to be the replacement for mii-tool in the near future and tends to be supported by newer NIC cards.
- The command provides the status of the interface you provide as its argument
  - #ethtool eth0

ethtool example

```
[root@rose ~]# ethtool eth1
Settings for eth1:
        Supported ports: [ TP MII ]
        Supported link modes:  10baseT/Half 10baseT/Full
                100baseT/Half 100baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                100baseT/Half 100baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: g
        Current message level: 0x00000007 (7)
        Link detected: yes
[root@rose ~]#
```

Setting NIC's Speed Parameters with ethtool

- Unlike mii-tool, ethtool settings can be permanently set as part of the interface's configuration script with the ETHTOOL_OPTS variable.
- In example, the settings will be set to 100 Mbps, full duplex with no chance for auto-negotiation on the next reboot:

```
#
# File: /etc/sysconfig/network-script/ifcfg-eth0
#
DEVICE=eth0
IPADDR=192.168.1.100
NETMASK=255.255.255.0
BOOTPROTO=static
ONBOOT=yes
ETHTOOL_OPTS="speed 100 duplex full autoneg off"
```

Viewing network error

Possible Causes of Ethernet Errors

- Collisions: Signifies when the NIC card detects itself and another server on the LAN attempting data transmissions at the same time. Collisions can be expected as a normal part of Ethernet operation and are typically below 0.1% of all frames sent. Higher error rates are likely to be caused by faulty NIC cards or poorly terminated cables.
- Single Collisions: The Ethernet frame went through after only one collision
- Multiple Collisions: The NIC had to attempt multiple times before successfully sending the frame due to collisions.

Viewing network error

Possible Causes of Ethernet Errors

- CRC Errors: Frames were sent but were corrupted in transit. The presence of CRC errors, but not many collisions usually is an indication of electrical noise.
  - Make sure that you are using the correct type of cable, that the cabling is undamaged and that the connectors are securely fastened.
- Frame Errors: An incorrect CRC and a non-integer number of bytes are received. This is usually the result of collisions or a bad Ethernet device.

Viewing network error

Possible Causes of Ethernet Errors

- FIFO and Overrun Errors: The number of times that the NIC was unable of handing data to its memory buffers because the data rate the capabilities of the hardware. This is usually a sign of excessive traffic.
- Length Errors: The received frame length was less than or exceeded the Ethernet standard. This is most frequently due to incompatible duplex settings.
- Carrier Errors: Errors are caused by the NIC card losing its link connection to the hub or switch. Check for faulty cabling or faulty interfaces on the NIC and networking equipment.

“ifconfig” error output

- The ifconfig command shows the number of overrun, carrier, dropped packet and frame errors.

```
eth1    Link encap:Ethernet  HWaddr 00:D0:B7:17:33:7D
        inet addr:172.27.21.199  Bcast:172.27.21.255  Mask:255.255.255.0
        inet6 addr: fe80::2d0:b7ff:fe17:337d/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:2153169 errors:0 dropped:0 overruns:0 frame:0
        TX packets:312348 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:260613351 (248.5 MiB) TX bytes:363578058 (346.7 MiB)
```

“netstat” error output

- The netstat command is very versatile and can provide a limited report when used with the -i switch. This is useful for systems where mii-tool or ethtool are not available.

```
[root@rose ~]# netstat -i
Kernel Interface table
Iface       MTU  Met       RX-OK       RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0    1500 0            10313242     0      0   6                13684527  0   0   0 BMRU
eth1    1500 0            2153176       0      0   0          312348  0   0   0 BMRU
lo          164360        17407           0      0   0          17407   0   0   0 LRU
[root@rose ~]#
```

Using ping to Test Network Connectivity

- The Linux ping command will send continuous pings, once a second, until stopped with a Ctrl-C. Here is an example of a successful ping to the server bigboy at 192.168.1.100

```
[root@smallfry tmp]# ping 192.168.1.101

PING 192.168.1.101 (192.168.1.101) from 192.168.1.100 : 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=128 time=3.95 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=128 time=7.07 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=128 time=4.46 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=128 time=4.31 ms

--- 192.168.1.101 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 3026ms
rtt min/avg/max/mdev = 3.950/4.948/7.072/1.242 ms

[root@smallfry tmp]#
```

Using ping to Test Network Connectivity

Most servers will respond to a ping query it becomes a very handy tool. A lack of response could be due to:

- A server with that IP address doesn't exist
- The server has been configured not to respond to pings
- You have incorrect routing. Check the routes and subnet masks on both the local and remote servers and all routers in between.
- Either the source or destination device having an incorrect IP address or subnet mask.
- Server may be down

Example:budcispapp101v down from Openview, Opsware, and Spectrum

```
[root@budcispapp101v ~]# netsoswbb heartbeat:Sat Apr 16 05:35:45 CEST 2016
tat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
3.215.64.0      0.0.0.0         255.255.252.0   U         0 0          0 eth2
3.215.144.0     0.0.0.0         255.255.252.0   U         0 0          0 eth0
2.5.0.0         0.0.0.0         255.255.0.0     U         0 0          0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 eth2
0.0.0.0         3.215.144.1     0.0.0.0         UG        0 0          0 eth0
[root@budcispapp101v ~]# ping 3.215.144.1
PING 3.215.144.1 (3.215.144.1) 56(84) bytes of data.
64 bytes from 3.215.144.1: icmp_seq=13 ttl=255 time=61.9 ms

--- 3.215.144.1 ping statistics ---
16 packets transmitted, 1 received, 93% packet loss, time 15030ms
rtt min/avg/max/mdev = 61.921/61.921/61.921/0.000 ms
```

```
bash-3.00# ping budcispapp101v.corporate.ge.com
PING budcispapp101v.corporate.ge.com (3.215.146.15) 56(84) bytes of data.
64 bytes from budcispapp101v.corporate.ge.com (3.215.146.15): icmp_seq=24 ttl=56 time=427 ms

--- budcispapp101v.corporate.ge.com ping statistics ---
29 packets transmitted, 1 received, 96% packet loss, time 28040ms
rtt min/avg/max/mdev = 427.130/427.130/427.130/0.000 ms, pipe 2
bash-3.00# ping 3.215.146.15
bash-3.00# ssh budcispapp101v.corporate.ge.com
ssh: connect to host budcispapp101v.corporate.ge.com port 22: Connection refused
bash-3.00#
```

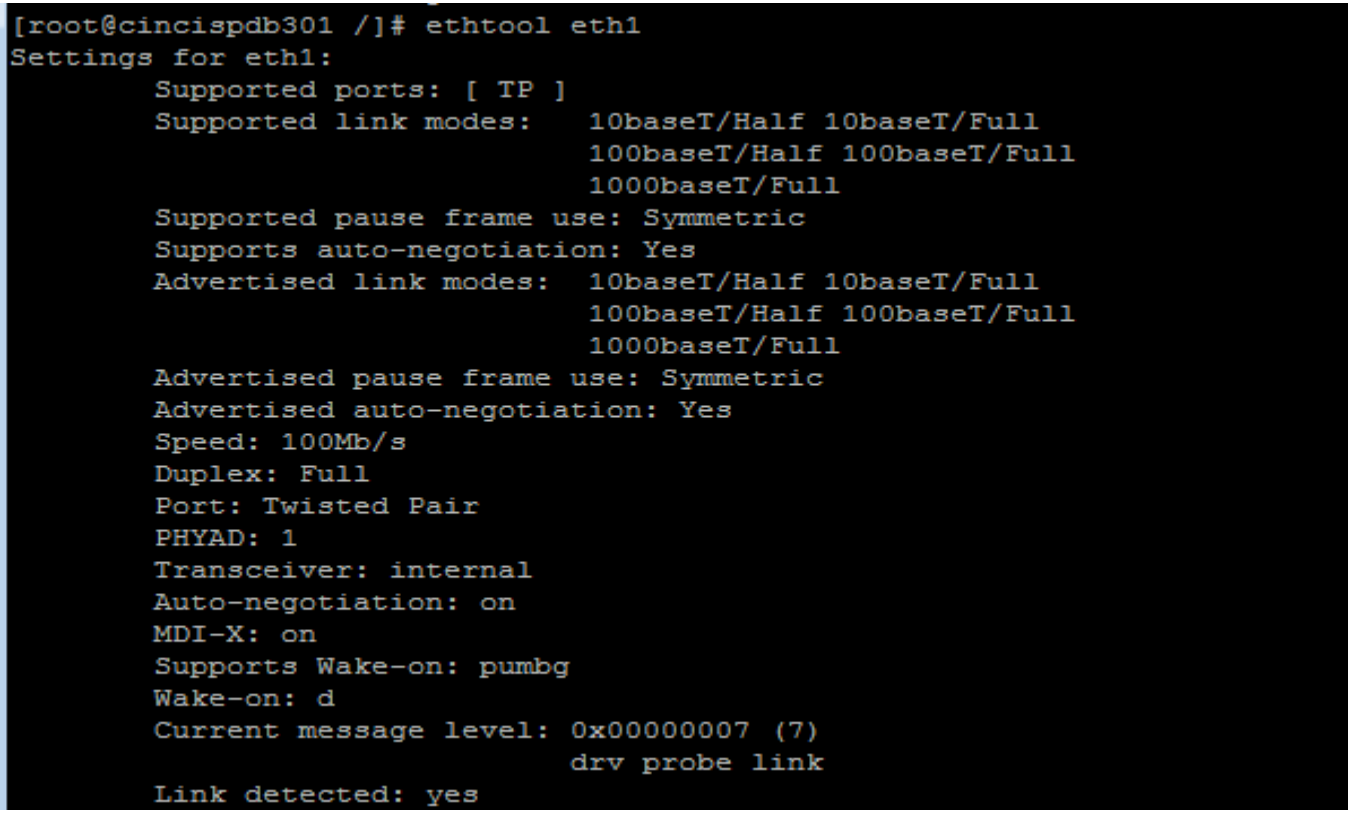
NIC CARD REPLACEMENT

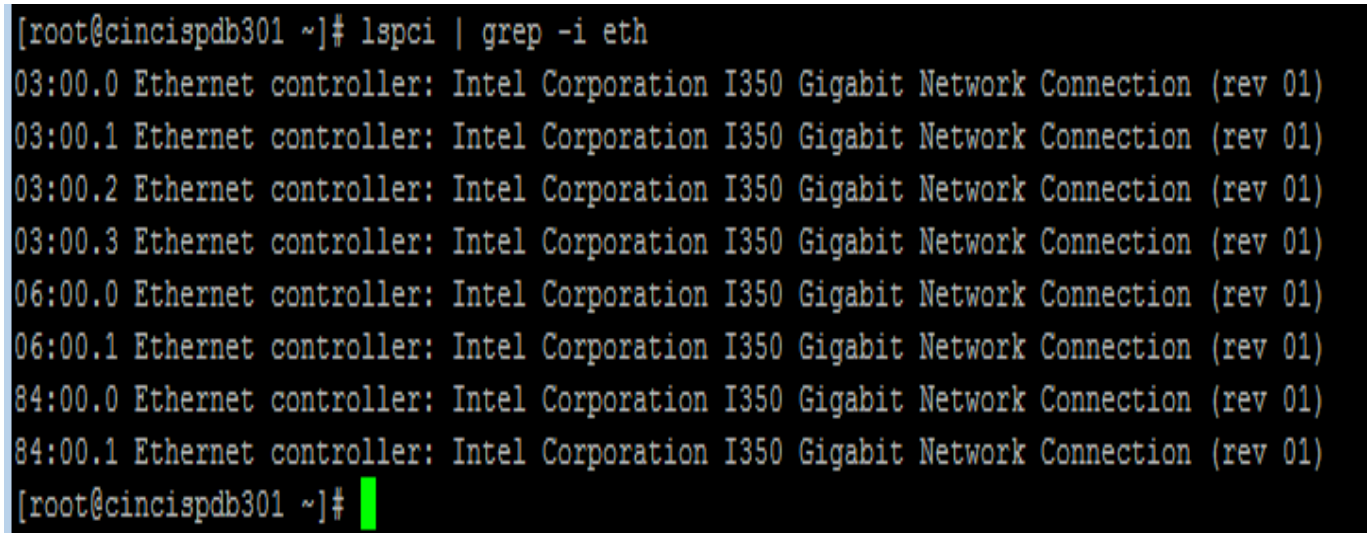
- dmesg | grep –I NIC
- Ethtool eth1
- Netstat –nrv
- Netstat –I
- Cat /proc/net/bonding/bond0
- Arp

```
[root@cincispdb301 /]# mv /etc/udev/rules.d/70-persistent-net.rules /etc/udev/rules.d/70-persistent-net.rules.16042016^C
[root@cincispdb301 /]#
[root@cincispdb301 /]# cat /etc/udev/rules.d/70-persistent-net.rules
# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="8c:dc:d4:ae:f8:c0", ATTR{type}=="1", KERNEL=="eth*", NAM

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="8c:dc:d4:ae:f8:c1", ATTR{type}=="1", KERNEL=="eth*", NAM

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="8c:dc:d4:ae:f8:c2", ATTR{type}=="1", KERNEL=="eth*", NAM

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="8c:dc:d4:ae:f8:c3", ATTR{type}=="1", KERNEL=="eth*", NAM

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="38:ea:a7:15:c4:9c", ATTR{type}=="1", KERNEL=="eth*", NAM

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="38:ea:a7:15:c4:9d", ATTR{type}=="1", KERNEL=="eth*", NAM

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="38:ea:a7:15:c6:16", ATTR{type}=="1", KERNEL=="eth*", NAM

# PCI device 0x8086:0x1521 (igb)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="38:ea:a7:15:c6:17", ATTR{type}=="1", KERNEL=="eth*", NAM
```

```
[root@cincispdb301 /]# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
169.254.47.235           ether   8c:dc:d4:af:33:3d   C                     eth1
169.254.12.76            ether   8c:dc:d4:af:30:e9   C                     eth1
169.254.130.81           ether   38:ea:a7:15:98:e9   C                     eth5
rsdwp005.corporate.ge.c  ether   8c:dc:d4:af:30:e8   C                     bond0
169.254.133.83           ether   38:ea:a7:15:96:77   C                     eth5
isscinecnas.corporate.g  ether   02:07:43:08:b7:1c   C                     eth6
3.24.148.12              ether   02:07:43:09:40:54   C                     eth6
cincispdb302.corporate.  ether   8c:dc:d4:af:33:3c   C                     bond0
giscinnbumstr1_b         ether   00:00:0c:9f:fe:d9   C                     eth2
cincispdb303-priv0.corp  ether   8c:dc:d4:af:30:e9   C                     eth1
cincispdb302-priv1.corp  ether   38:ea:a7:15:98:e9   C                     eth5
eddwp005.corporate.ge.c  ether   8c:dc:d4:af:33:3c   C                     bond0
cincispdb303-priv1.corp  ether   38:ea:a7:15:96:77   C                     eth5
cincispdb302-priv0.corp  ether   8c:dc:d4:af:33:3d   C                     eth1
cincispdb301-vip.corpor          (incomplete)                              bond0
3.24.148.159             ether   02:15:17:d3:27:48   C                     eth6
snulbxuscingh03.gdn.ge.  ether   00:00:0c:07:ac:07   C                     bond0
bodwp005.corporate.ge.c  ether   8c:dc:d4:af:33:3c   C                     bond0
cincispdb303.corporate.  ether   8c:dc:d4:af:30:e8   C                     bond0
[root@cincispdb301 /]# ifconfig | grep -i addr
```

```
[root@cincispdb301 /]# dmesg | grep -i NIC
igb: eth4 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth2 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth5 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth6 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth1 NIC Link is Up 100 Mbps Full Duplex, Flow Control: None
igb: eth1 NIC Link is Down
igb: eth1 NIC Link is Up 100 Mbps Full Duplex, Flow Control: None
igb: eth1 NIC Link is Up 100 Mbps Full Duplex, Flow Control: None
igb: eth5 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth4 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth2 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth5 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth6 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth1 NIC Link is Up 100 Mbps Full Duplex, Flow Control: None
igb: eth1 NIC Link is Down
igb: eth1 NIC Link is Up 100 Mbps Full Duplex, Flow Control: None
```


[root@cincispdb301 /]# ethtool eth1
Settings for eth1:
Supported ports: [TP]
Supported link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Supported pause frame use: Symmetric
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Advertised pause frame use: Symmetric
Advertised auto-negotiation: Yes
Speed: 100Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: on
Supports Wake-on: pumbg
Wake-on: d
Current message level: 0x00000007 (7)
drv probe link
Link detected: yes


[root@cincispdb301 ~]# lspci | grep -i eth
03:00.0 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
03:00.1 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
03:00.2 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
03:00.3 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
06:00.0 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
06:00.1 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
84:00.0 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
84:00.1 Ethernet controller: Intel Corporation I350 Gigabit Network Connection (rev 01)
[root@cincispdb301 ~]#

After adding new NIC card & reboot

```
[root@cincispdb301 ~]# dmesg | grep -i NIC
igb: eth4 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth5 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth6 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth2 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
igb: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[root@cincispdb301 ~]#
```

References

- http://www.linuxhomenetworking.com/
- http://en.wikipedia.org/wiki/Main_Page