

RHSA (Admin 1) Red Hat Enterprise Linux 8 Essentials

Prepared by Eng. Taghreed Elsayed

Lecture 7

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

MANAGING RED HAT ENTERPRISE LINUX NETWORKING



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

IPV4 NETWORKING TCP/IP standards follow a 5-layers network model

- Application Each application has specifications for communication so that clients and servers may communicate across platforms. Common protocols include SSH (remote login), HTTPS (secure web), NFS or CIFS (file sharing), and SMTP (electronic mail delivery).
- Transport Transport protocols are **TCP and UDP**.
- TCP is a reliable connection-oriented communication, while UDP is a connectionless datagram protocol. Application protocols use TCP or UDP ports.
- A list of well-known and registered ports can be found in the /etc/services file.
- When a packet is sent on the network, the combination of the service port and IP address forms a socket. Each packet has a source socket and a destination socket. This information can be used when monitoring and filtering.



- Internet The Internet, or network layer, carries data from the source host to the destination host. Each host has an IP address and a prefix used to determine network addresses. Routers are used to connect networks.
- **ICMP** is a control protocol at this layer. Instead of ports, it has types. The ping utility is an example of using ICMP packets to test connectivity.
- ping sends an **ICMP ECHO_REQUEST** packet. A successful ping receives an **ICMP ECHO_REPLY** acknowledgment.
- An unsuccessful ping may receive **ICMP** error messages such as "destination unreachable" or may not receive any response.
- Link The link, or media access, layer provides the connection to physical media. The most common types of networks are **wired Ethernet (802.3)** and wireless WLAN (802.11). Each physical device has a hardware address (**MAC**) which is used to identify the destination of packets on the local network segment.



IP Address:

192.168.5.3 = 11000000 . 10101000 . 00000101 . 00000011

Prefix: /24

Netmask:

255.255.255.0 = 11111111 . 11111111 . 11111111 . 00000000

11000000 . 10101000 . 00000101 . 00000011

Network

Host

❑ IPv4 addresses

- ❑ An IPv4 address is a 32-bit number, normally expressed in decimal as four octets ranging in value from 0 to 255, separated by dots.
- ❑ The address is divided into two parts: the network part and the host part.
- ❑ All hosts on the same subnet, which can talk to each other directly without a router, have the same network part; the network part identifies the subnet.
- ❑ No two hosts on the same subnet can have the same host part; the host part identifies a particular host on a subnet.
- ❑ The lowest possible address on a subnet (host part is all zeros in binary) is sometimes called the network address.



- ❑ The highest possible address on a subnet (host part is all ones in binary) is used for broadcast messages in IPv4, and is called the broadcast address.
- ❑ Network masks are expressed in two forms. The older syntax for a netmask which uses 24 bits for the network part would read 255.255.255.0. A newer syntax, called CIDR notation, would specify a network prefix of /24. Both forms convey the same information; namely, how many leading bits in the IP address contribute to its network address.



Calculating the network address for 192.168.1.107/24

Host addr	192.168.1.107	11000000.10101000.00000001.01101011
Network prefix	/24 (255.255.255.0)	11111111.11111111.11111111.00000000
Network addr	192.168.1.0	11000000.10101000.00000001.00000000
Broadcast addr	192.168.1.255	11000000.10101000.00000001.11111111



Calculating the network address for 172.16.181.23/19

Host addr	172.168.181.23	10101100.10101000.10110101.00010111
Network prefix	/19 (255.255.224.0)	11111111.11111111.11100000.00000000
Network addr	172.168.160.0	10101100.10101000.10100000.00000000
Broadcast addr	172.168.191.255	10101100.10101000.10111111.11111111



❑ Names and IP addresses

- ❖ The IP protocol uses addresses to communicate, but human beings would rather work with names than long and hard-to-remember strings of numbers.
- ❖ DNS, the Domain Name System, is a distributed network of servers that maps host names to IP addresses.
- ❖ In order for name service to work, the host needs to be pointed at a nameserver. This nameserver does not need to be on the same subnet; it just needs to be reachable by the host.
- ❖ DHCP or static network configuration Many systems are configured to obtain network settings automatically at boot time.



The local configuration files indicate that DHCP should be used and a separate client service queries the network for a server and obtains a lease for network settings

- ❖ If a DHCP server is not available, the system must use a static configuration where the network settings are read from a local configuration file.
- ❖ The correct network settings are obtained from the network administrator or architecture team to ensure there are no conflicts with other systems. Since DHCP uses the hardware address to track assignments, only one address may be assigned per interface with DHCP.
- ❖ Multiple static addresses may be assigned to a single interface. This practice is common in systems hosting services for multiple clients, such as HTTP IP-based hosting. Red Hat Enterprise Linux interfaces typically have an IPv4 address and an IPv6 local-link address, but may have more addresses assigned.



NOTE

Network interface naming can be overridden. If the administrator has installed and enabled the **biosdevname** package or set customized **udev** device naming rules, those settings will override the default naming scheme. Depending on support for **biosdevname** in the system BIOS, names such as **em1**, **em2**, etc. may be used for on-board network cards (corresponding to their names on the chassis). PCI(e) cards are represented with **pYpX** (e.g., **p4p1**), where **Y** is the PCI slot number and **X** is the number for the port on that specific card.

VALIDATING NETWORK CONFIGURATION

❑ DISPLAYING IP ADDRESSES

- ❑ The `/sbin/ip` command is used to show device and address information.

```
[student@desktopX ~]$ ip addr show eth0  
2: eth0: <BROADCAST,MULTICAST, ①UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP  
qlen 1000  
    ②link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff  
    ③inet 172.25.0.10/24 brd ④172.25.0.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    ⑤inet6 fe80::5054:ff:fe00:b/64 scope link  
        valid_lft forever preferred_lft forever
```



- 1 An active interface has the status of **UP**.
- 2 The link line specifies the hardware (MAC) address of the device.
- 3 The inet line shows the IPv4 address and prefix.
- 4 The broadcast address, scope, and device name are also on this line.
- 5 The inet6 line shows IPv6 information.

TROUBLESHOOTING ROUTING The `/sbin/ip` command is also used to show routing information.

```
[student@desktopX ~]$ ip route
default via 172.25.0.254 dev eth0 proto static metric 1024
172.25.X.0/24 dev eth0 proto kernel scope link src 172.25.X.10
10.0.0.0/8 dev eth1 proto kernel scope link src 10.0.0.11
```

- ❑ The ping command is used to test connectivity. The command will continue to run until Ctrl+C is pressed unless options are given to limit the number of packets sent.

```
[student@desktopX ~]$
ping -c3 172.25.X.254
```



- ❑ To trace the path to a remote host, use either `traceroute` or `tracpath`.
- ❖ Both commands can be used to trace a path with UDP packets; however, many networks block UDP and ICMP traffic.
- The `traceroute` command has options to trace the path with UDP (default), ICMP (-I), or TCP (-T) packets, but may not be installed by default.

```
[student@desktopX ~]$ tracpath access.redhat.com
...
4: 71-32-28-145.rcmt.qwest.net 48.853ms asymm 5
5: dcp-brdr-04.inet.qwest.net 100.732ms asymm 7
6: 206.111.0.153.ptr.us.xo.net 96.245ms asymm 7
7: 207.88.14.162.ptr.us.xo.net 85.270ms asymm 8
8: ae1d0.cir1.atlanta6-ga.us.xo.net 64.160ms asymm 7
9: 216.156.108.98.ptr.us.xo.net 108.652ms
10: bu-ether13.atlngamq46w-bcr00.tbone.rr.com 107.286ms asymm 12
...
```



EXAMINING NETWORK CONFIGURATION

1. Display the current IP address and netmask for all interfaces.

```
[student@serverX ~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:00:00:0b brd ff:ff:ff:ff:ff:ff
    inet 172.25.X.11/24 brd 172.25.X.255 scope global dynamic eth0
        valid_lft 12704sec preferred_lft 12704sec
    inet6 fe80::5054:ff:fe00:b/64 scope link
        valid_lft forever preferred_lft forever
```



2. Display the statistics for the eth0 interface.

```
[student@serverX ~]$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
  DEFAULT qlen 1000
    link/ether 52:54:00:00:00:0b brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    418398      4588      0        0          0        0
    TX: bytes  packets  errors  dropped carrier collsns
    360733      1730      0        0          0        0
```

3. Display the routing information.

```
[student@serverX ~]$ ip route
default via 172.25.X.254 dev eth0  proto static  metric 1024
172.25.X.0/24 dev eth0  proto kernel  scope link  src 172.25.X.11
```



4. Verify that the router is accessible.

```
[student@serverX ~]$ ping -c3 172.25.X.254

[student@serverX ~]$ ip route
default via 172.25.X.254 dev eth0 proto static metric 1024
172.25.X.0/24 dev eth0 proto kernel scope link src 172.25.X.11
PING 172.25.X.254 (172.25.X.254) 56(84) bytes of data.
 64 bytes from 172.25.X.254: icmp_seq=1 ttl=64 time=0.489 ms
 64 bytes from 172.25.X.254: icmp_seq=2 ttl=64 time=0.510 ms
 64 bytes from 172.25.X.254: icmp_seq=3 ttl=64 time=0.458 ms

--- 172.25.X.254 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss, time 1999ms
 rtt min/avg/max/mdev = 0.458/0.485/0.510/0.033 ms
```



5. Show all the hops between the local system and classroom.example.com.

```
[student@serverX ~]$ tracepath classroom.example.com
 1:  classroom.example.com                0.522ms !H
    Resume: pmtu 65535
```

6. Display the listening TCP sockets on the local system.

```
[student@serverX ~]$ ss -lt
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	128	*:55630	*:*
LISTEN	0	128	*:sunrpc	*:*
LISTEN	0	128	*:ssh	*:*
LISTEN	0	100	127.0.0.1:smtp	*:*
LISTEN	0	128	:::sunrpc	:::*
LISTEN	0	128	:::ssh	:::*
LISTEN	0	128	:::33079	:::*
LISTEN	0	100	:::1:smtp	:::*



CONFIGURING NETWORKING WITH nmcli

- ❑ TO manage network settings and devices with nmcli and NetworkManager.
- ❑ **NetworkManager** is a daemon that monitors and manages network settings. In addition to the daemon, there is a GNOME Notification Area applet that provides network status information. Command-line and graphical tools talk to NetworkManager and save configuration files in the / etc/sysconfig/network-scripts directory.

- ❑ The **nmcli** command can be used to show device status and details.

```
[root@desktopX ~]# nmcli dev status
DEVICE   TYPE      STATE      CONNECTION
eth0     ethernet  connected  static-eth0
wlp3s0   wifi      connected  guest
lo       loopback  unmanaged  --
[root@desktopX ~]# nmcli dev show eth0
GENERAL.DEVICE:                eth0
GENERAL.TYPE:                  ethernet
GENERAL.HWADDR:                52:54:00:00:00:0A
GENERAL.MTU:                   1500
GENERAL.STATE:                 100 (connected)
GENERAL.CONNECTION:            static-eth0
GENERAL.CON-PATH:              /org/freedesktop/NetworkManager/
                                ActiveConnection/1
WIRED-PROPERTIES.CARRIER:     on
IP4.ADDRESS[1]:                ip = 172.25.X.10/24, gw = 172.25.X.254
IP4.DNS[1]:                    172.25.254.254
IP6.ADDRESS[1]:                ip = fe80::5054:ff:fe00:b/64, gw = ::
```

nmcli commands

COMMAND	USE
nmcli dev status	List all devices.
nmcli con show	List all connections.
nmcli con up "<ID>"	Activate a connection.
nmcli con down "<ID>"	Deactivate a connection. The connection will restart if autoconnect is yes.
nmcli dev dis <DEV>	Bring down an interface and temporarily disable autoconnect.
nmcli net off	Disable all managed interfaces.
nmcli con add ...	Add a new connection.
nmcli con mod "<ID>" ...	Modify a connection.
nmcli con del "<ID>"	Delete a connection.

CONFIGURING HOST NAMES AND NAME RESOLUTION

- ❑ CHANGING THE SYSTEM HOST NAME The hostname command displays or temporarily modifies the system's fully qualified host name.

```
[root@desktopX ~]# hostname  
desktopX.example.com
```

CONFIGURING NAME RESOLUTION

- ❑ The stub resolver is used to convert host names to IP addresses or the reverse. The contents of the file /etc/hosts are checked first.



```
[root@desktopX ~]# cat /etc/hosts
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomain4
::1           localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.254.254 classroom.example.com
172.25.254.254 content.example.com
```

- The `getent hosts hostname` command can be used to test host name resolution with the `/etc/hosts` file.

❑ you will configure the system host name and name resolution.

- ▶ 1. View the current host name settings.
 - 1.1. Display the current host name.

```
[student@serverX ~]$ hostname  
serverX.example.com
```



1.2. Display the host name status.

```
[student@serverX ~]$ hostnamectl status
  Static hostname: n/a
Transient hostname: serverX.example.com
    Icon name: computer
    Chassis: n/a
  Machine ID: 9f6fb63045a845d79e5e870b914c61c9
    Boot ID: d4ec3a2e8d3c48749aa82738c0ea946a
Operating System: Red Hat Enterprise Linux Server 7.0 (Maipo)
  CPE OS Name: cpe:/o:redhat:enterprise_linux:7.0:GA:server
    Kernel: Linux 3.10.0-97.el7.x86_64
Architecture: x86_64
```



- ▶ 2. Set a static host name to match the current transient host name.
 - 2.1. Change the host name and host name configuration file. Replace the *X* with your station number and match the output of the previous step.

```
[student@serverX ~]$  
sudo hostnamectl set-hostname serverX.example.com
```

- 2.2. View the configuration file providing the host name at network start.

```
[student@serverX ~]$ cat /etc/hostname
```

```
serverX.example.com
```



2.3. Display the host name status.

```
[student@serverX ~]$ hostnamectl status
  Static hostname: serverX.example.com
        Icon name: computer
        Chassis: n/a
        Machine ID: 9f6fb63045a845d79e5e870b914c61c9
        Boot ID: d4ec3a2e8d3c48749aa82738c0ea946a
  Operating System: Red Hat Enterprise Linux Server 7.0 (Maipo)
        CPE OS Name: cpe:/o:redhat:enterprise_linux:7.0:GA:server
        Kernel: Linux 3.10.0-97.el7.x86_64
  Architecture: x86_64
```



▶ **3.** Temporarily change the host name.

3.1. Change the host name.

```
[student@serverX ~]$  
sudo hostname testname
```

3.2. Display the current host name.

```
[student@serverX ~]$ hostname  
testname
```

3.3. View the configuration file providing the host name at network start.

```
[student@serverX ~]$ cat /etc/hostname  
serverX.example.com
```



3.4. Reboot the system.

```
[student@serverX ~]$  
reboot
```

3.5. Display the current host name.

```
[student@serverX ~]$ hostname  
serverX.example.com
```

► 4. Add a local nickname for the classroom server.

4.1. Look up the IP address of the classroom.example.com.

```
[student@serverX ~]$ host classroom.example.com
```



classroom.example.com has address 172.25.254.254

- 4.2. Modify **/etc/hosts** so that the name **class** has the IP address 172.25.254.254 and can be used to communicate with classroom.example.com.

```
[student@serverX ~]$ sudo vim /etc/hosts[student@serverX ~]$ cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.254.254 classroom.example.com class
172.25.254.254 content.example.com
```

- 4.3. Look up the IP address of the class.

```
[student@serverX ~]$ host class
Host class not found: 2(SERVFAIL)
[student@serverX ~]$ getent hosts class
172.25.254.254    classroom.example.com class
```



4.4. Ping class.

```
[student@serverX ~]$ ping -c3 class
PING classroom.example.com (172.25.254.254) 56(84) bytes of data.
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=1 ttl=64 time=0.397
ms
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=2 ttl=64 time=0.447
ms
64 bytes from classroom.example.com (172.25.254.254): icmp_seq=3 ttl=64 time=0.470
ms

--- classroom.example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.397/0.438/0.470/0.030 ms
```



Chapter 011

MANAGING RED HAT ENTERPRISE LINUX NETWORKING





جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

