Version : **2024.01**

Last update : 2024/11/29 08:45

RH12413 - Network Management

Contents

Understanding IPv4

TCP headers

The TCP header is encoded on 4 bytes, i.e. 32 bits:

1st byte	2nd byte	3rd byte	4th byte
Source port		Destination port	
Sequence number			
Acknowledgement number			
Offset	Flags	Window	
Checksum		Urgent Pointer	
Options			Padding
Data			

You'll notice that port numbers are encoded on 16 bits. This information allows us to calculate the maximum number of ports in IPv4, i.e. 2^{16} ports or 65,535.

The **Offset** contains the size of the header.

The **Flags** are :

- URG - If the value is 1, the urgent pointer is used. The sequence number and the urgent pointer indicate a specific byte.
- ACK - If the value is 1, the packet is an acknowledgement.
- PSH - If the value is 1, the data is immediately presented to the application.
- RST - If the value is 1, there is a problem with the communication and the connection is reset.
- SYN - If the value is 1, the packet is a synchronisation packet
- FIN - If the value is 1, the packet indicates the end of the connection.

The **Window** is coded on 16 bits. The Window is a data element linked to the data forwarding operation known as the **sliding window**. Since it would be impossible, for performance reasons, to wait for each packet sent to be acknowledged, the sender sends packets in groups. The size of this group is called the Window. In the event of a problem receiving part of the Window, the whole Window is resent.

The **Checksum** is a way of calculating whether the packet is complete.

The **Padding** is a field that can be filled with zero values so that the size of the header is a multiple of 32.

UDP headers

The UDP header is encoded on 4 bytes, i.e. 32 bits:

1st byte	**2nd byte**	**3rd byte**	**4th byte**
Source port		Destination port	
Length		Checksum	
Data			

Fragmentation and Re-encapsulation

The maximum size of a TCP packet, including the header, is **65,535 bytes**. However, each network is qualified by its MTU (Maximum Tranfer Unit). This is the maximum packet size allowed. The unit is in **bytes**. For an Ethernet network its value is 1500. When a packet must be sent over a network with an MTU smaller than its own size, the packet must be **fragmented**. On leaving the network, the packet is reconstituted. This reconstitution is called **re-encapsulation**.

Addressing

IP addressing requires each device on the network to have a unique 4-byte IP address, i.e. 32 bits in the format XXX.XXX.XXX.XXX. In this way the total number of addresses is 2^{32} = 4.3 Billion.

IP addresses are divided into 5 classes, from A to E. The 4 bytes of classes A to C are divided into two parts, a part called the **Net ID** which identifies the network and a part called the **Host ID** which identifies the host:

	1st byte	**2nd byte**	**3rd byte**	**4th byte**
A	Net ID	Host ID		
B	Net ID		Host ID	

	1st byte	2nd byte	3rd byte	4th byte
C	Net ID			Host ID
D	Multicast			
E	Reserved			

The allocation of a class depends on the number of hosts to be connected. Each class is identified by a **Class ID** consisting of 1 to 3 bits:

Class	Class ID bits	Class ID value	Network ID bits	No. of networks	Host ID bits	No. of addresses	Start byte
A	1	0	7	2^7=128	24	2^{24}=16 777 216	1 - 126
B	2	10	14	2^{14}=16 834	16	2^{16}=65 535	128 - 191
C	3	110	21	2^{21}=2 097 152	8	2^8=256	192 - 223

Network 127. is reserved. It is called **loopback** and identifies the local machine.

In each class, certain addresses are reserved for private use:

Class	Start IP	End IP
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

There are special addresses which cannot be used to identify a host:

Special address	Description
169.254.0.0 to 169.254.255.255	Automatic Private IP Addressing from Microsoft
Host on current network	All Net ID bits set to 0
Network Address	All Host ID bits set to 0
Broadcast address	All Host ID bits set to 1

The network address identifies the **segment** of the entire network, while the broadcast address identifies all the hosts on the network segment.

To better understand the network address and the broadcast address, let's take the case of the 192.168.10.1 address in class C:

	1st byte	2nd byte	3rd byte	4th byte
	Net ID			Host ID
IP address	192	168	10	1
Binary	11000000	10101000	000001010	00000001
Calculation of the network address				
Binary	11000000	10101000	000001010	**00000000**
Network address	192	168	10	0
Calculation of the broadcast address				
Binary	11000000	10101000	000001010	**11111111**
Broadcast address	192	168	10	255

Subnet Masks

Like the IP address, the subnet mask has 4 bytes or 32 bits. Subnet masks are used to identify the Net ID and Host ID:

Class	Mask	CIDR notation
A	255.0.0.0	/8
B	255.255.0.0	/16
C	255.255.255.0	/24

CIDR stands for **Classless InterDomain Routing**. The term CIDR notation corresponds to the number of bits of a value of 1 in the subnet mask.

When a host wishes to transmit, it first identifies its own network address using an AND calculation applied to its own address and subnet mask, which stipulates :

- 1 x 1 = 1
- 0 x 1 = 0
- 1 x 0 = 0
- 0 x 0 = 0

Let's take the case of the IP address 192.168.10.1 with a mask of 255.255.255.0:

	1st byte	2nd byte	3rd byte	4th byte

	1st byte	2nd byte	3rd byte	4th byte
IP address	192	168	10	1
Binary	11000000	10101000	000001010	00000001
Subnet mask				
Binary	11111111	11111111	11111111	00000000
AND Calculation	11000000	10101000	00001010	00000000
Network address	192	168	10	0

This host is trying to communicate with a host having an IP address of 192.168.10.10. It therefore performs the same calculation by applying **its own subnet mask** to the IP address of the destination host:

	1st byte	2nd byte	3rd byte	4th byte
IP address	192	168	10	10
Binary	11000000	10101000	000001010	00001010
Subnet mask				
Binary	11111111	11111111	11111111	00000000
AND Calculation	11000000	10101000	00001010	00000000
Network address	192	168	10	0

Since the network address is identical in both cases, the sending host assumes that the destination host is on its network and sends the packets directly to the network.

The sending host is now trying to communicate with a host with an IP address of 192.168.2.1. It therefore performs the same calculation by applying **its own subnet mask** to the IP address of the destination host:

	1st byte	2nd byte	3rd byte	4th byte
IP address	192	168	2	1
Binary	11000000	10101000	00000010	00000001
Subnet mask				
Binary	11111111	11111111	11111111	00000000
AND Calculation	11000000	10101000	00000010	00000000
Network address	192	168	2	0

In this case, the sending host finds that the destination network 192.168.2.0 is not identical to its own network 192.168.10.0. It therefore sends the packets to the default gateway.

VLSM

Since the stock of networks available under IPv4 is almost exhausted, a solution has had to be found to create subnets while waiting for the introduction of IPv6. This solution is called VLSM or Variable Length Subnet Masks. VLSM expresses subnet masks in CIDR format.

The principle is simple. In order to create different networks from a network address of a given class, the number of hosts needs to be reduced. In this way, the ‘freed’ bits of the Host ID can be used to identify the sub-networks.

To illustrate this, let's take the example of a 192.168.1.0 network. On this network, we can put 2^8-2 or 254 hosts between 192.168.1.1 and 192.168.1.254.

Suppose we want to divide our network into 2 sub-networks. To code 2 sub-networks, we need to free 2 bits of the Host ID. The two freed bits will have the following binary values:

- 00
- 01
- 10
- 11

The binary values of the fourth byte of our subnet addresses will therefore be :

- 192.168.1.00XXXXXX
- 192.168.1.01XXXXXX
- 192.168.1.10XXXXXX
- 192.168.1.11XXXXXX

where the XXXXXX represent the bits we reserve to describe the hosts in each of the subnets.

We cannot use the following two subnets:

- 192.168.1.00XXXXXX

- 192.168.1.11XXXXXX

because these correspond to the beginnings of the network address 192.168.1.0 and the broadcast address 192.168.1.255.

We can use the following two subnets:

- 192.168.1.01XXXXXX
- 192.168.1.10XXXXXX

For the first subnet, the network address and broadcast address are :

Subnet #1	192	168	1	01XXXXXX
Calculation of the network address				
Binary	11000000	10101000	00000001	01**000000**
Network address	192	168	1	**64**
Calculation of the broadcast address				
Binary	11000000	10101000	00000001	01**111111**
Broadcast address	192	168	1	**127**

- The network's CIDR address is therefore 192.168.1.64/26 because the Net ID is coded on 24+2 bits.
- The subnet mask is therefore 11111111.11111111.11111111.11000000 or 255.255.255.192.
- We can have 2^6-2 or 62 hosts.
- The valid range of IP addresses is 192.168.1.65 to 192.168.1.126.

For the second subnet, the network address and broadcast address are :

Subnet #2	192	168	1	10XXXXXX
Calculation of the network address				
Binary	11000000	10101000	00000001	10**000000**
Network address	192	168	1	**128**
Calculation of the broadcast address				
Binary	11000000	10101000	00000001	10**111111**
Broadcast address	192	168	1	**191**

- The network's CIDR address is therefore 192.168.1.128/26 because the Net ID is coded on 24+2 bits.
- The subnet mask is therefore 11111111.11111111.11111111.11000000 or 255.255.255.192.
- We can have 2^6-2 or 62 hosts.
- The valid range of IP addresses is 192.168.1.129 to 192.168.1.190.

The value separating the subnets is 64. This value has the name **increment**.

Ports and Sockets

TCP uses port numbers on the transport layer to ensure that data reaches the applications it is intended for. Port numbers are divided into three groups:

- **Well Known Ports**
  - From 1 to 1023
- Registered Ports
  - From 1024 to 49151
- **Dynamic** and/or **Private Ports**.
  - From 49152 to 65535

The **IP number:port number** pair is called a **socket**.

The /etc/services File

The most commonly used ports are detailed in the **/etc/services** file:

```
[root@redhat9 ~]# more /etc/services
# /etc/services:
# $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
#
# Network services, Internet style
# IANA services version: last updated 2016-07-08
#
```

```
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#       http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name  port/protocol  [aliases ...]   [# comment]

tcpmux          1/tcp                           # TCP port service multiplexer
tcpmux          1/udp                           # TCP port service multiplexer
rje             5/tcp                           # Remote Job Entry
rje             5/udp                           # Remote Job Entry
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
daytime         13/tcp
daytime         13/udp
qotd            17/tcp          quote
qotd            17/udp          quote
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp-data        20/udp
```

```
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # The Secure Shell (SSH) Protocol
ssh             22/udp                          # The Secure Shell (SSH) Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
lmtp            24/tcp                          # LMTP Mail Delivery
lmtp            24/udp                          # LMTP Mail Delivery
smtp            25/tcp          mail
smtp            25/udp          mail
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/tcp          resource        # resource location
--More--(0%)
[q]
```

Note that ports are listed in pairs:

- the TCP port
- the UDP port

The most complete list can be found at **https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml**.

To find out which sockets are open on your computer, enter the following command:

```
[root@redhat9 ~]# netstat -an | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:15023         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
```

```
tcp        0      0 10.0.2.102:22           10.0.2.1:59570          ESTABLISHED
tcp6       0      0 ::1:15023               :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::23                   :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 ::1:55794               ::1:22                  ESTABLISHED
tcp6       0      0 ::1:22                  ::1:55794               ESTABLISHED
udp        0      0 127.0.0.1:323           0.0.0.0:*
udp        0      0 0.0.0.0:42140           0.0.0.0:*
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp6       0      0 ::1:323                 :::*
udp6       0      0 :::58479                :::*
udp6       0      0 :::5353                 :::*
raw6       0      0 :::58                   :::*                    7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     22852    /var/run/lsm/ipc/sim
unix  2      [ ACC ]     STREAM     LISTENING     22853    /var/run/lsm/ipc/simc
unix  2      [ ACC ]     STREAM     LISTENING     23975    @/tmp/.ICE-unix/1794
unix  2      [ ACC ]     STREAM     LISTENING     23732    @/tmp/dbus-40kNqTSK
unix  2      [ ]         DGRAM                    40557    /run/user/1000/systemd/notify
unix  2      [ ]         DGRAM                    23906    /run/user/42/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     40560    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     23909    /run/user/42/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     40570    /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     23930    /run/user/42/bus
unix  2      [ ACC ]     STREAM     LISTENING     40572    /run/user/1000/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     23932    /run/user/42/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     40574    /run/user/1000/pipewire-0
unix  2      [ ACC ]     STREAM     LISTENING     23934    /run/user/42/pipewire-0
unix  3      [ ]         DGRAM      CONNECTED     2446     /run/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     40576    /run/user/1000/pipewire-0-manager
unix  2      [ ACC ]     STREAM     LISTENING     23936    /run/user/42/pipewire-0-manager
```

```
unix  2      [ ACC ]     STREAM     LISTENING     2451     /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     2452     /run/systemd/io.system.ManagedOOM
unix  2      [ ]         DGRAM      CONNECTED     25598    /run/chrony/chronyd.sock
unix  18     [ ]         DGRAM      CONNECTED     2459     /run/systemd/journal/dev-log
unix  9      [ ]         DGRAM      CONNECTED     2461     /run/systemd/journal/socket
unix  2      [ ACC ]     STREAM     LISTENING     22376    /run/user/42/wayland-0
unix  2      [ ACC ]     STREAM     LISTENING     2463     /run/systemd/journal/stdout
unix  2      [ ACC ]     STREAM     LISTENING     23731    @/tmp/dbus-peR2NXOg
unix  2      [ ACC ]     STREAM     LISTENING     22391    /tmp/dbus-ApzcsH4y3k
unix  2      [ ACC ]     STREAM     LISTENING     23976    /tmp/.ICE-unix/1794
unix  2      [ ACC ]     STREAM     LISTENING     22756    @ISCSID_UIP_ABSTRACT_NAMESPACE
unix  2      [ ACC ]     STREAM     LISTENING     31411    /etc/httpd/run/cgisock.1083
unix  2      [ ACC ]     STREAM     LISTENING     25894    @/var/lib/gdm/.cache/ibus/dbus-hSex1tg6
unix  2      [ ACC ]     STREAM     LISTENING     22373    /tmp/.X11-unix/X1024
unix  2      [ ACC ]     STREAM     LISTENING     22375    /tmp/.X11-unix/X1025
unix  2      [ ACC ]     STREAM     LISTENING     23734    @/tmp/dbus-TkF9Vnen
--More--
[q]
```

To find out which applications have opened a port on the computer, enter the following command:

```
[root@redhat9 ~]# netstat -anp | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      5583/sshd: /usr/sbi
tcp        0      0 127.0.0.1:15023         0.0.0.0:*               LISTEN      5603/ssh
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      878/cupsd
tcp        0      0 10.0.2.102:22           10.0.2.1:59570          ESTABLISHED 4306/sshd: trainee
tcp6       0      0 ::1:15023               :::*                    LISTEN      5603/ssh
tcp6       0      0 :::22                   :::*                    LISTEN      5583/sshd: /usr/sbi
tcp6       0      0 :::23                   :::*                    LISTEN      1/systemd
tcp6       0      0 :::80                   :::*                    LISTEN      1083/httpd
tcp6       0      0 ::1:631                 :::*                    LISTEN      878/cupsd
tcp6       0      0 ::1:55794               ::1:22                  ESTABLISHED 5603/ssh
```

```
tcp6       0      0 ::1:22                  ::1:55794               ESTABLISHED 5596/sshd: trainee
udp        0      0 127.0.0.1:323           0.0.0.0:*                           2675/chronyd
udp        0      0 0.0.0.0:42140           0.0.0.0:*                           752/avahi-daemon: r
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           752/avahi-daemon: r
udp6       0      0 ::1:323                 :::*                                2675/chronyd
udp6       0      0 :::58479                :::*                                752/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                752/avahi-daemon: r
raw6       0      0 :::58                   :::*                    7           4456/NetworkManager
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   PID/Program name     Path
unix  2      [ ACC ]     STREAM     LISTENING     22852    755/lsmd             /var/run/lsm/ipc/sim
unix  2      [ ACC ]     STREAM     LISTENING     22853    755/lsmd             /var/run/lsm/ipc/simc
unix  2      [ ACC ]     STREAM     LISTENING     23975    1794/gnome-session-  @/tmp/.ICE-unix/1794
unix  2      [ ACC ]     STREAM     LISTENING     23732    1140/gdm             @/tmp/dbus-40kNqTSK
unix  2      [ ]         DGRAM                    40557    4129/systemd         /run/user/1000/systemd/notify
unix  2      [ ]         DGRAM                    23906    1340/systemd         /run/user/42/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     40560    4129/systemd         /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     23909    1340/systemd         /run/user/42/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     40570    4129/systemd         /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     23930    1340/systemd         /run/user/42/bus
unix  2      [ ACC ]     STREAM     LISTENING     40572    4129/systemd         /run/user/1000/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     23932    1340/systemd         /run/user/42/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     40574    4129/systemd         /run/user/1000/pipewire-0
unix  2      [ ACC ]     STREAM     LISTENING     23934    1340/systemd         /run/user/42/pipewire-0
unix  3      [ ]         DGRAM      CONNECTED     2446     1/systemd            /run/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     40576    4129/systemd         /run/user/1000/pipewire-0-manager
unix  2      [ ACC ]     STREAM     LISTENING     23936    1340/systemd         /run/user/42/pipewire-0-manager
unix  2      [ ACC ]     STREAM     LISTENING     2451     1/systemd
/run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     2452     1/systemd            /run/systemd/io.system.ManagedOOM
unix  2      [ ]         DGRAM      CONNECTED     25598    2675/chronyd         /run/chrony/chronyd.sock
unix  18     [ ]         DGRAM      CONNECTED     2459     1/systemd            /run/systemd/journal/dev-log
unix  9      [ ]         DGRAM      CONNECTED     2461     1/systemd            /run/systemd/journal/socket
unix  2      [ ACC ]     STREAM     LISTENING     22376    1802/gnome-shell     /run/user/42/wayland-0
```

```
unix  2      [ ACC ]     STREAM     LISTENING     2463     1/systemd            /run/systemd/journal/stdout
unix  2      [ ACC ]     STREAM     LISTENING     23731    1140/gdm             @/tmp/dbus-peR2NXOg
unix  2      [ ACC ]     STREAM     LISTENING     22391    1825/dbus-daemon     /tmp/dbus-ApzcsH4y3k
unix  2      [ ACC ]     STREAM     LISTENING     23976    1794/gnome-session-  /tmp/.ICE-unix/1794
unix  2      [ ACC ]     STREAM     LISTENING     22756    1/systemd            @ISCSID_UIP_ABSTRACT_NAMESPACE
unix  2      [ ACC ]     STREAM     LISTENING     31411    3121/httpd           /etc/httpd/run/cgisock.1083
unix  2      [ ACC ]     STREAM     LISTENING     25894    2130/ibus-daemon     @/var/lib/gdm/.cache/ibus/dbus-
hSex1tg6
unix  2      [ ACC ]     STREAM     LISTENING     22373    1831/Xwayland        /tmp/.X11-unix/X1024
unix  2      [ ACC ]     STREAM     LISTENING     22375    1802/gnome-shell     /tmp/.X11-unix/X1025
unix  2      [ ACC ]     STREAM     LISTENING     23734    1140/gdm             @/tmp/dbus-TkF9Vnen
--More--
[q]
```

The **ss** command can also be used to find out the list of open sockets:

```
[root@redhat9 ~]# ss -ta
State                  Recv-Q                 Send-Q                                  Local Address:Port
Peer Address:Port                 Process
LISTEN                 0                      128                                        0.0.0.0:ssh
0.0.0.0:*
LISTEN                 0                      128                                     127.0.0.1:15023
0.0.0.0:*
LISTEN                 0                      4096                                    127.0.0.1:ipp
0.0.0.0:*
ESTAB                  0                      0                                     10.0.2.102:ssh
10.0.2.1:59570
LISTEN                 0                      128                                        [::1]:15023
[::]:*
LISTEN                 0                      128                                         [::]:ssh
[::]:*
LISTEN                 0                      4096                                           *:telnet
*:*
LISTEN                 0                      511                                            *:http
```

```
*:*
LISTEN                  0                       4096                                    [::1]:ipp
[::]:*
ESTAB                   0                       0                                       [::1]:55794
[::1]:ssh
ESTAB                   0                       0                                       [::1]:ssh
[::1]:55794
```

Ethernet Address Resolution

Each protocol can be encapsulated in an Ethernet **frame**. When the frame has to be transported from the sender to the recipient, the sender must know the Ethernet address of the recipient. The Ethernet address is also known as the **Physical** or **MAC** address.

To find out the recipient's Ethernet address, the sender uses the **ARP** protocol. The information received is stored in a table. To view this information, use the following command:

```
[root@redhat9 ~]# arp -a
_gateway (10.0.2.1) at 92:8f:ca:52:ce:96 [ether] on ens18
```

The command line switches for this command are :

```
[root@redhat9 ~]# arp --help
Usage:
  arp [-vn]  [<HW>] [-i <if>] [-a] [<hostname>]             <-Display ARP cache
  arp [-v]          [-i <if>] -d  <host> [pub]               <-Delete ARP entry
  arp [-vnD] [<HW>] [-i <if>] -f  [<filename>]            <-Add entry from file
  arp [-v]   [<HW>] [-i <if>] -s  <host> <hwaddr> [temp]            <-Add entry
  arp [-v]   [<HW>] [-i <if>] -Ds <host> <if> [netmask <nm>] pub          <-''-

        -a                      display (all) hosts in alternative (BSD) style
        -e                      display (all) hosts in default (Linux) style
        -s, --set               set a new ARP entry
        -d, --delete            delete a specified entry
```

```
        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        -i, --device             specify network interface (e.g. eth0)
        -D, --use-device         read <hwaddr> from given device
        -A, -p, --protocol       specify protocol family
        -f, --file               read new entries from file or from /etc/ethers

  <HW>=Use '-H <hw>' to specify hardware address type. Default: ether
  List of possible hardware types (which support ARP):
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
    dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) x25 (generic X.25) infiniband (InfiniBand)
    eui64 (Generic EUI-64)
```

Understanding IPv6

Overview

IPv6 can be used in parallel with IPv4 in a dual-stack model. In this configuration, a network interface can have one or more IPv6 addresses as well as IPv4 addresses. RHEL 9 operates in dual-stack mode by default.

IPv6 addresses

An IPv6 address is a 128-bit number, normally expressed as eight groups of four hexadecimal **nibbles** (half-bytes) separated by a colon. Each nibble represents four bits of the IPv6 address, so that each group represents 16 bits of the IPv6 address.

```
2001:0db8:0000:0010:0000:0000:0000:0001
```

To make it easier to write IPv6 addresses, it is not necessary to write zeros at the start of a group separated by a colon. However, at least one

hexadecimal digit must be written in each group separated by a colon:

```
2001:db8:0:10:0:0:0:1
```

Under these rules, 2001:db8::0010:0:0:0:1 would be a less practical way of writing the address in the example, but it is a valid representation of the same address.

Tips for writing readable addresses in a consistent way are:

- Remove leading zeros in a group.
- Use : : to shorten the address as much as possible.
- If an address contains two consecutive groups of zeros of the same length, it is preferable to shorten the leftmost groups of zeros to : : and the rightmost groups to :0 : for each group.
- Although permitted, do not use :: to shorten a group of zeros. Instead, use :0 : and keep : : for consecutive groups of zeros.
- Always use lower case letters for hexadecimal numbers from a to f.

For example :

```
2001:db8:0:10::1
```

Lastly, an IPv6 socket must contain the characters [ ] around the IPv6 address:

```
[2001:db8:0:10::1]:80
```

A normal IPv6 unicast address is divided into two parts:

- The network prefix,
    - The prefix identifies the subnet.
- The interface identifier,
    - Two network interfaces on the same subnet cannot have the same identifier,
    - An interface identifier identifies a particular interface on the subnet.

Subnet Masks

Unlike IPv4, IPv6 has a standard subnet mask of /64, which is used for almost all normal addresses. This means that a single subnet can contain as many hosts as required. Typically, the network provider will assign a shorter prefix to an organisation, for example, /48. This leaves the rest of the network free to assign subnets (always of length /64) from the prefix assigned. For a /48 prefix, there are 16 bits left for subnets, i.e. 65536 subnets.

For example, in the case of the address **2001:0db8:0000:0001:0000:0000:0000:0001**, expressed as **2001:0db8:0:1/64**, the NetID part is **2001:0db8:0000:0001** and the HostID part is **0000:0000:0000:0001**.

Looking at the NetID, the **2001:0db8:0000** part, expressed as **2001:db8::/48** represents the allocation provided, while **0001/16** represents the subnet.

Reserved IPv6 Addresses

IPv6 addresses reserved for a specific use are :

Address	Description
::1/128	The loopback address similar to 127.0.0.1/8
::	Global listening address similar to 0.0.0.0
::/0	Default route similar to 0.0.0.0/0
2000::/3	This address space concerns the network addresses allocated by IANA, ranging from 2000::/16 to 3fff::/16
fd00::/8	From RFC 4193, this is similar to RFC 1918, i.e. a private address space.
fe80::/10	Link-local addresses
ff00::/8	Multicast addresses

Link-local Addresses

Each interface on a network is automatically configured with a Link-local address:

```
[root@redhat9 ~]# ifconfig
ens18: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

```
        inet 10.0.2.102  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::2da3:cf78:c904:b9b9  prefixlen 64  scopeid 0x20<link>
        ether 92:86:d7:66:e7:5a  txqueuelen 1000  (Ethernet)
        RX packets 21754  bytes 51437196 (49.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14363  bytes 1838520 (1.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 944  bytes 110925 (108.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 944  bytes 110925 (108.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

To test the connectivity of the Link-local address, use the **ping6** command:

```
[root@redhat9 ~]# ping6 -c4 fe80::2da3:cf78:c904:b9b9%ens18
PING fe80::2da3:cf78:c904:b9b9%ens18(fe80::2da3:cf78:c904:b9b9%ens18) 56 data bytes
64 bytes from fe80::2da3:cf78:c904:b9b9%ens18: icmp_seq=1 ttl=64 time=0.111 ms
64 bytes from fe80::2da3:cf78:c904:b9b9%ens18: icmp_seq=2 ttl=64 time=0.107 ms
64 bytes from fe80::2da3:cf78:c904:b9b9%ens18: icmp_seq=3 ttl=64 time=0.116 ms
64 bytes from fe80::2da3:cf78:c904:b9b9%ens18: icmp_seq=4 ttl=64 time=0.145 ms

--- fe80::2da3:cf78:c904:b9b9%ens18 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.107/0.119/0.145/0.014 ms
```

Important: Note that at the end of the address, you must add the **scope** which is represented by the **%** character followed by the name of the interface.

DHCPv6

DHCPv6 does not work in the same way as DHCPv4 because there are no broadcast addresses under IPv6.

In summary, the host sends a DHCPv6 request to the multicast address **all-dhcp-servers**, **ff02::1:2** on port **547/udp**. In return, the host receives the requested information on the **546/udp** port of its Link-local address.

Configuring the Network

RHEL 9 uses **Network Manager** to manage the network. Network Manager has two components:

- a service that manages network connections and reports their status,
- front-ends that use an API to configure the service.

> **Important**: Note that with this version of NetworkManager, IPv6 is enabled by default.

The NetworkManager service must always be started:

```
[root@redhat9 ~]# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
     Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-09-28 15:37:17 CEST; 20h ago
       Docs: man:NetworkManager(8)
   Main PID: 857 (NetworkManager)
      Tasks: 3 (limit: 48800)
     Memory: 12.0M
        CPU: 1.834s
     CGroup: /system.slice/NetworkManager.service
             └─857 /usr/sbin/NetworkManager --no-daemon
```

```
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.6471] device (ens18): state
change: config -> ip-config (reason 'none', sys-iface-state: 'managed')
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.6481] policy: set 'ens18' (ens18)
as default for IPv4 routing and DNS
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.7310] device (ens18): state
change: ip-config -> ip-check (reason 'none', sys-iface-state: 'managed')
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.7327] device (ens18): state
change: ip-check -> secondaries (reason 'none', sys-iface-state: 'managed')
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.7328] device (ens18): state
change: secondaries -> activated (reason 'none', sys-iface-state: 'managed')
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.7331] manager: NetworkManager
state is now CONNECTED_SITE
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.7333] device (ens18): Activation:
successful, device activated.
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.7338] manager: NetworkManager
state is now CONNECTED_GLOBAL
Sep 28 15:37:18 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530638.7340] manager: startup complete
Sep 28 15:37:29 redhat9.ittraining.loc NetworkManager[857]: <info>  [1727530649.0717] agent-manager:
agent[3c1f9786a5e709c2,:1.25/org.gnome.Shell.NetworkAgent/42]: agent registered
```

The nmcli command

The **nmcli** (Network Manager Command Line Interface) command is used to configure NetworkManager.

Command line switches and sub-commands can be viewed using the following command:

```
[root@redhat9 ~]# nmcli help
Usage: nmcli [OPTIONS] OBJECT { COMMAND | help }

OPTIONS
  -a, --ask                                ask for missing parameters
  -c, --colors auto|yes|no                 whether to use colors in output
  -e, --escape yes|no                      escape columns separators in values
```

```
  -f, --fields <field,...>|all|common      specify fields to output
  -g, --get-values <field,...>|all|common  shortcut for -m tabular -t -f
  -h, --help                               print this help
  -m, --mode tabular|multiline             output mode
  -o, --overview                           overview mode
  -p, --pretty                             pretty output
  -s, --show-secrets                       allow displaying passwords
  -t, --terse                              terse output
  -v, --version                            show program version
  -w, --wait <seconds>                     set timeout waiting for finishing operations

OBJECT
  g[eneral]       NetworkManager's general status and operations
  n[etworking]    overall networking control
  r[adio]         NetworkManager radio switches
  c[onnection]    NetworkManager's connections
  d[evice]        devices managed by NetworkManager
  a[gent]         NetworkManager secret agent or polkit agent
  m[onitor]       monitor NetworkManager changes
```

LAB #1 - Network Configuration

1.1 - Connections and Profiles

NetworkManager includes the notion of **connections** or **profiles** allowing different configurations depending on the location. To view the current connections, use the **nmcli c** command with the **show** sub-command:

```
[root@redhat9 ~]# nmcli c show
NAME   UUID                                  TYPE      DEVICE
ens18  ea4c8254-6236-3130-8323-8b3f71d807a1  ethernet  ens18
lo     8df82174-1d45-4506-9248-6bfcd2d20240  loopback  lo
```

Now create a another IP profile attached to the **ens18** device:

```
[root@redhat9 ~]# nmcli connection add con-name ip_fixed ifname ens18 type ethernet ip4 10.0.2.102/24 gw4
10.0.2.1
Connection 'ip_fixed' (b3d51921-4deb-4975-ad52-f31993b2af0c) successfully added.
```

Note its presence :

```
[root@redhat9 ~]# nmcli c show
NAME     UUID                                  TYPE      DEVICE
ens18    ea4c8254-6236-3130-8323-8b3f71d807a1  ethernet  ens18
lo       8df82174-1d45-4506-9248-6bfcd2d20240  loopback  lo
ip_fixed b3d51921-4deb-4975-ad52-f31993b2af0c  ethernet  --
```

Note that the output does not indicate that the **ip_fixed** profile is associated with the **ens18** device because the **ip_fixed** profile is not enabled:

```
[root@redhat9 ~]# nmcli d show
GENERAL.DEVICE:                         ens18
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         92:86:D7:66:E7:5A
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens18
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/2
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         10.0.2.101/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP4.DNS[1]:                             8.8.8.8
IP6.ADDRESS[1]:                         fe80::9086:d7ff:fe66:e75a/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 1024
```

```
GENERAL.DEVICE:                         lo
GENERAL.TYPE:                           loopback
GENERAL.HWADDR:                         00:00:00:00:00:00
GENERAL.MTU:                            65536
GENERAL.STATE:                          100 (connected (externally))
GENERAL.CONNECTION:                     lo
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/1
IP4.ADDRESS[1]:                         127.0.0.1/8
IP4.GATEWAY:                            --
IP6.ADDRESS[1]:                         ::1/128
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = ::1/128, nh = ::, mt = 256
```

To activate the ip_fixed profile, use the following command :

```
[root@redhat9 ~]# nmcli connection up ip_fixed
```

Note that your terminal is blocked because the IP address has changed.

> **To do** - Return to the Guacamole home page and reconnect to the VM as a trainee using the **RedHat9_10.0.2.102_SSH** connection.

The ip_fixed profile is now enabled while the ens18 profile has been disabled:

```
[root@redhat9 ~]# nmcli c show
NAME     UUID                                  TYPE      DEVICE
ip_fixed b3d51921-4deb-4975-ad52-f31993b2af0c  ethernet  ens18
lo       8df82174-1d45-4506-9248-6bfcd2d20240  loopback  lo
ens18    ea4c8254-6236-3130-8323-8b3f71d807a1  ethernet  --
[root@redhat9 ~]# nmcli d show
GENERAL.DEVICE:                         ens18
```

```
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         92:86:D7:66:E7:5A
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ip_fixed
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/3
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         10.0.2.102/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP6.ADDRESS[1]:                         fe80::2da3:cf78:c904:b9b9/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 1024

GENERAL.DEVICE:                         lo
GENERAL.TYPE:                           loopback
GENERAL.HWADDR:                         00:00:00:00:00:00
GENERAL.MTU:                            65536
GENERAL.STATE:                          100 (connected (externally))
GENERAL.CONNECTION:                     lo
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/1
IP4.ADDRESS[1]:                         127.0.0.1/8
IP4.GATEWAY:                            --
IP6.ADDRESS[1]:                         ::1/128
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = ::1/128, nh = ::, mt = 256
```

To view the **ens18** profile parameters, use the following command :

```
[root@redhat9 ~]# nmcli -p connection show ens18
===============================================================================
                      Connection profile details (ens18)
===============================================================================
```

```
connection.id:                          ens18
connection.uuid:                        ea4c8254-6236-3130-8323-8b3f71d807a1
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              ens18
connection.autoconnect:                 yes
connection.autoconnect-priority:        -999
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1727605468
connection.permissions:                 --
connection.zone:                        --
connection.controller:                  --
connection.master:                      --
connection.slave-type:                  --
connection.port-type:                   --
connection.autoconnect-slaves:          -1 (default)
connection.autoconnect-ports:           -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.dns-over-tls:                -1 (default)
connection.mptcp-flags:                 0x0 (default)
connection.wait-device-timeout:         -1
connection.wait-activation-delay:       -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:    --
802-3-ethernet.accept-all-mac-addresses:-1 (default)
------------------------------------------------------------------------------
ipv4.method:                            manual
ipv4.dns:                               8.8.8.8
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         10.0.2.101/24
lines 1-55
[q]
```

Similarly, to view the **ip_fixed** profile parameters, use the following command:

```
[root@redhat9 ~]# nmcli -p connection show ip_fixed
==============================================================================
                    Connection profile details (ip_fixed)
==============================================================================
connection.id:                          ip_fixed
connection.uuid:                        b3d51921-4deb-4975-ad52-f31993b2af0c
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              ens18
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
```

```
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1727605469
connection.permissions:                 --
connection.zone:                        --
connection.controller:                  --
connection.master:                      --
connection.slave-type:                  --
connection.port-type:                   --
connection.autoconnect-slaves:          -1 (default)
connection.autoconnect-ports:           -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.dns-over-tls:                -1 (default)
connection.mptcp-flags:                 0x0 (default)
connection.wait-device-timeout:         -1
connection.wait-activation-delay:       -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:    --
802-3-ethernet.accept-all-mac-addresses:-1 (default)
------------------------------------------------------------------------------
ipv4.method:                            manual
ipv4.dns:                               --
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         10.0.2.102/24
lines 1-55
[q]
```

To view the list of profiles associated with a device, use the following command :

```
[root@redhat9 ~]# nmcli -f CONNECTIONS device show ens18
CONNECTIONS.AVAILABLE-CONNECTION-PATHS:
/org/freedesktop/NetworkManager/Settings/1,/org/freedesktop/NetworkManager/Settings/3
CONNECTIONS.AVAILABLE-CONNECTIONS[1]:   ea4c8254-6236-3130-8323-8b3f71d807a1 | ens18
CONNECTIONS.AVAILABLE-CONNECTIONS[2]:   b3d51921-4deb-4975-ad52-f31993b2af0c | ip_fixed
```

The configuration files for the **ens18** device can be found in the **/etc/NetworkManager/system-connections** directory:

```
[root@redhat9 ~]# ls -l /etc/NetworkManager/system-connections
total 8
-rw-------. 1 root root 253 Oct 19  2023 ens18.nmconnection
-rw-------. 1 root root 218 Sep 29 12:21 ip_fixed.nmconnection
```

1.2 - Name resolution

A study of the **/etc/NetworkManager/system-connections/ip_fixed.nmconnection** file shows that there are no directives concerning DNS :

```
[root@redhat9 ~]# cat /etc/NetworkManager/system-connections/ip_fixed.nmconnection
[connection]
id=ip_fixed
uuid=b3d51921-4deb-4975-ad52-f31993b2af0c
type=ethernet
interface-name=ens18

[ethernet]

[ipv4]
address1=10.0.2.102/24,10.0.2.1
method=manual

[ipv6]
addr-gen-mode=default
method=auto

[proxy]
```

Name resolution is therefore inactive:

```
[root@redhat9 ~]# ping www.free.fr
ping: www.free.fr: Name or service not known
```

Modify the configuration of the **ip_fixed** profile:

```
[root@redhat9 ~]# nmcli connection mod ip_fixed ipv4.dns 8.8.8.8
```

A look at the **/etc/NetworkManager/system-connections/ip_fixed.nmconnection** file shows that the DNS server directive has been added:

```
[root@redhat9 ~]# cat /etc/NetworkManager/system-connections/ip_fixed.nmconnection
[connection]
id=ip_fixed
```

```
uuid=b3d51921-4deb-4975-ad52-f31993b2af0c
type=ethernet
interface-name=ens18
timestamp=1727605469

[ethernet]

[ipv4]
address1=10.0.2.102/24,10.0.2.1
dns=8.8.8.8;
method=manual

[ipv6]
addr-gen-mode=default
method=auto

[proxy]
```

For the DNS server change to take effect, restart the NetworkManager service:

```
[root@redhat9 ~]# systemctl restart NetworkManager.service

[root@redhat9 ~]# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
     Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-29 12:36:35 CEST; 11s ago
       Docs: man:NetworkManager(8)
   Main PID: 4456 (NetworkManager)
      Tasks: 4 (limit: 48800)
     Memory: 5.5M
        CPU: 82ms
     CGroup: /system.slice/NetworkManager.service
             └─4456 /usr/sbin/NetworkManager --no-daemon
```

```
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5342] device (lo): state change:
secondaries -> activated (reason 'none', sys-iface-state: 'external')
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5347] device (lo): Activation:
successful, device activated.
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5360] device (ens18): state
change: ip-config -> ip-check (reason 'none', sys-iface-state: 'assume')
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5654] device (ens18): state
change: ip-check -> secondaries (reason 'none', sys-iface-state: 'assume')
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5660] device (ens18): state
change: secondaries -> activated (reason 'none', sys-iface-state: 'assume')
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5663] manager: NetworkManager
state is now CONNECTED_SITE
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5666] device (ens18):
Activation: successful, device activated.
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5671] manager: NetworkManager
state is now CONNECTED_GLOBAL
Sep 29 12:36:35 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606195.5673] manager: startup complete
Sep 29 12:36:36 redhat9.ittraining.loc NetworkManager[4456]: <info>  [1727606196.0852] agent-manager:
agent[923443df876692f7,:1.25/org.gnome.Shell.NetworkAgent/42]: agent registered
```

Check that the **/etc/resolv.conf** file has been modified by NetworkManager :

```
[root@redhat9 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search ittraining.loc
nameserver 8.8.8.8
```

Lastly, check the name resolution:

```
[root@redhat9 ~]# ping www.free.fr
PING www.free.fr (212.27.48.10) 56(84) bytes of data.
64 bytes from www.free.fr (212.27.48.10): icmp_seq=1 ttl=47 time=89.2 ms
64 bytes from www.free.fr (212.27.48.10): icmp_seq=2 ttl=47 time=89.2 ms
64 bytes from www.free.fr (212.27.48.10): icmp_seq=3 ttl=47 time=89.3 ms
```

```
^C
--- www.free.fr ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3005ms
rtt min/avg/max/mdev = 89.153/89.186/89.252/0.046 ms
```

Important : Note that there is a graphical front-end, **nmtui**, for configuring NetworkManager.

1.3 - Adding a Second IP Address to a Profile

To add a second IP address to a profile under RHEL 9, use the following command:

```
[root@redhat9 ~]# nmcli connection mod ip_fixed +ipv4.addresses 192.168.1.2/24
```

Reload the profile configuration:

```
[root@redhat9 ~]# nmcli con up ip_fixed
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)
```

Then enter the following command:

```
[root@redhat9 ~]# nmcli connection show ip_fixed
connection.id:                          ip_fixed
connection.uuid:                        b3d51921-4deb-4975-ad52-f31993b2af0c
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              ens18
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
```

```
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1727606325
connection.permissions:                 --
connection.zone:                        --
connection.controller:                  --
connection.master:                      --
connection.slave-type:                  --
connection.port-type:                   --
connection.autoconnect-slaves:          -1 (default)
connection.autoconnect-ports:           -1 (default)
connection.secondaries:                 --
connection.gateway-ping-timeout:        0
connection.metered:                     unknown
connection.lldp:                        default
connection.mdns:                        -1 (default)
connection.llmnr:                       -1 (default)
connection.dns-over-tls:                -1 (default)
connection.mptcp-flags:                 0x0 (default)
connection.wait-device-timeout:         -1
connection.wait-activation-delay:       -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:    --
802-3-ethernet.accept-all-mac-addresses:-1 (default)
ipv4.method:                            manual
ipv4.dns:                               8.8.8.8
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         10.0.2.102/24, 192.168.1.2/24
ipv4.gateway:                           10.0.2.1
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.replace-local-rule:                -1 (default)
ipv4.ignore-auto-routes:                no
ipv4.ignore-auto-dns:                   no
ipv4.dhcp-client-id:                    --
ipv4.dhcp-iaid:                         --
ipv4.dhcp-dscp:                         --
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:                          yes
ipv4.required-timeout:                  -1 (default)
ipv4.dad-timeout:                       -1 (default)
ipv4.dhcp-vendor-class-identifier:      --
ipv4.link-local:                        0 (default)
ipv4.dhcp-reject-servers:               --
ipv4.auto-route-ext-gw:                 -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.replace-local-rule:                -1 (default)
ipv6.ignore-auto-routes:                no
ipv6.ignore-auto-dns:                   no
ipv6.never-default:                     no
ipv6.may-fail:                          yes
ipv6.required-timeout:                  -1 (default)
ipv6.ip6-privacy:                       -1 (unknown)
ipv6.addr-gen-mode:                     default
ipv6.ra-timeout:                        0 (default)
ipv6.mtu:                               auto
ipv6.dhcp-pd-hint:                      --
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.auto-route-ext-gw:                 -1 (default)
ipv6.token:                             --
proxy.method:                           none
proxy.browser-only:                     no
proxy.pac-url:                          --
proxy.pac-script:                       --
GENERAL.NAME:                           ip_fixed
```

```
GENERAL.UUID:                           b3d51921-4deb-4975-ad52-f31993b2af0c
GENERAL.DEVICES:                        ens18
GENERAL.IP-IFACE:                       ens18
GENERAL.STATE:                          activated
GENERAL.DEFAULT:                        yes
GENERAL.DEFAULT6:                       no
GENERAL.SPEC-OBJECT:                    --
GENERAL.VPN:                            no
GENERAL.DBUS-PATH:                      /org/freedesktop/NetworkManager/ActiveConnection/3
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/Settings/3
GENERAL.ZONE:                           --
GENERAL.MASTER-PATH:                    --
IP4.ADDRESS[1]:                         192.168.1.2/24
IP4.ADDRESS[2]:                         10.0.2.102/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 192.168.1.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[3]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP4.DNS[1]:                             8.8.8.8
IP6.ADDRESS[1]:                         fe80::2da3:cf78:c904:b9b9/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 1024
lines 77-131/131 (END)
[q]
```

Important : Note the addition of the secondary address to the **ipv4.addresses:** and the addition of the **IP4.ADDRESS[2]:** line.

Now look at the contents of the **/etc/NetworkManager/system-connections/ip_fixed.nmconnection** file:

```
[root@redhat9 ~]# cat /etc/NetworkManager/system-connections/ip_fixed.nmconnection
```

```
[connection]
id=ip_fixed
uuid=b3d51921-4deb-4975-ad52-f31993b2af0c
type=ethernet
interface-name=ens18
timestamp=1727606195

[ethernet]

[ipv4]
address1=10.0.2.102/24,10.0.2.1
address2=192.168.1.2/24
dns=8.8.8.8;
method=manual

[ipv6]
addr-gen-mode=default
method=auto

[proxy]
```

Important: Note the addition of the line **address2=192.168.1.2/24**.

1.4 - The hostname Command

The hostname modification procedure is simplified and takes effect immediately:

```
[root@redhat9 ~]# hostname
redhat9.ittraining.loc
```

```
[root@redhat9 ~]# cat /etc/hostname
redhat9.ittraining.loc

[root@redhat9 ~]# nmcli general hostname redhat.ittraining.loc

[root@redhat9 ~]# cat /etc/hostname
redhat.ittraining.loc

[root@redhat9 ~]# nmcli general hostname redhat9.ittraining.loc

[root@redhat9 ~]# cat /etc/hostname
redhat9.ittraining.loc

[root@redhat9 ~]# hostname
redhat9.ittraining.loc
```

1.5 - The ip Command

Under RHEL 9 the **ip** command is preferred to the ifconfig command:

```
[root@redhat9 ~]# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 92:86:d7:66:e7:5a brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.0.2.102/24 brd 10.0.2.255 scope global noprefixroute ens18
       valid_lft forever preferred_lft forever
    inet 192.168.1.2/24 brd 192.168.1.255 scope global noprefixroute ens18
```

```
       valid_lft forever preferred_lft forever
    inet6 fe80::2da3:cf78:c904:b9b9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Command Line Switches

The command line switches for this command are :

```
[root@redhat9 ~]# ip --help
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { address | addrlabel | amt | fou | help | ila | ioam | l2tp |
                   link | macsec | maddress | monitor | mptcp | mroute | mrule |
                   neighbor | neighbour | netconf | netns | nexthop | ntable |
                   ntbl | route | rule | sr | tap | tcpmetrics |
                   token | tunnel | tuntap | vrf | xfrm }
       OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                    -h[uman-readable] | -iec | -j[son] | -p[retty] |
                    -f[amily] { inet | inet6 | mpls | bridge | link } |
                    -4 | -6 | -M | -B | -0 |
                    -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                    -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                    -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                    -c[olor]}
```

1.6 - Manually Enabling/Disabling an Interface

There are two commands for manually deactivating and activating a network interface:

```
# nmcli device disconnect enp0s3
# nmcli device connect enp0s3
```

Important: Please do **NOT** execute these two commands.

1.7 - Static Routing

The ip Command

Under RHEL 9, to delete the route to the 192.168.1.0 network, use the ip command and not the route command:

```
[root@redhat9 ~]# ip route
default via 10.0.2.1 dev ens18 proto static metric 100
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.102 metric 100
192.168.1.0/24 dev ens18 proto kernel scope link src 192.168.1.2 metric 100

[root@redhat9 ~]# ip route del 192.168.1.0/24 via 0.0.0.0

[root@redhat9 ~]# ip route
default via 10.0.2.1 dev ens18 proto static metric 100
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.102 metric 100
```

To add the route to the 192.168.1.0 network :

```
[root@redhat9 ~]# ip route add 192.168.1.0/24 via 10.0.2.1

[root@redhat9 ~]# ip route
default via 10.0.2.1 dev ens18 proto static metric 100
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.102 metric 100
192.168.1.0/24 via 10.0.2.1 dev ens18
```

Important - The command used to add a default gateway takes the following form **ip route add default via *address ip***.

Enable Routing on the Server

To enable IPv4 routing on the server, packet retransmission must be enabled:

```
[root@redhat9 ~]# cat /proc/sys/net/ipv4/ip_forward
0
[root@redhat9 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward

[root@redhat9 ~]# cat /proc/sys/net/ipv4/ip_forward
1
```

To enable IPv6 routing on the server, you need to enable packet retransmission:

```
[root@redhat9 ~]# cat /proc/sys/net/ipv6/conf/all/forwarding
0

[root@redhat9 ~]# echo ‘1’ > /proc/sys/net/ipv6/conf/all/forwarding

[root@redhat9 ~]# cat /proc/sys/net/ipv6/conf/all/forwarding
1
```

LAB #2 - Network Diagnostics

2.1 - ping

To test the accessibility of a machine, you need to use the **ping** command:

```
[root@redhat9 ~]# ping -c4 10.0.2.1
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data.
64 bytes from 10.0.2.1: icmp_seq=1 ttl=64 time=0.157 ms
64 bytes from 10.0.2.1: icmp_seq=2 ttl=64 time=0.130 ms
64 bytes from 10.0.2.1: icmp_seq=3 ttl=64 time=0.212 ms
64 bytes from 10.0.2.1: icmp_seq=4 ttl=64 time=0.222 ms

--- 10.0.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3104ms
rtt min/avg/max/mdev = 0.130/0.180/0.222/0.038 ms
```

Command Line Switches for the ping command

The command line switches for this command are :

```
[root@redhat9 ~]# ping --help
ping: invalid option -- '-'

Usage
  ping [options] <destination>

Options:
  <destination>      dns name or ip address
  -a                 use audible ping
  -A                 use adaptive ping
  -B                 sticky source address
  -c <count>         stop after <count> replies
  -D                 print timestamps
```

```
  -d                 use SO_DEBUG socket option
  -f                 flood ping
  -h                 print help and exit
  -I <interface>     either interface name or address
  -i <interval>      seconds between sending each packet
  -L                 suppress loopback of multicast packets
  -l <preload>       send <preload> number of packages while waiting replies
  -m <mark>          tag the packets going out
  -M <pmtud opt>     define mtu discovery, can be one of <do|dont|want>
  -n                 no dns name resolution
  -O                 report outstanding replies
  -p <pattern>       contents of padding byte
  -q                 quiet output
  -Q <tclass>        use quality of service <tclass> bits
  -s <size>          use <size> as number of data bytes to be sent
  -S <size>          use <size> as SO_SNDBUF socket option value
  -t <ttl>           define time to live
  -U                 print user-to-user latency
  -v                 verbose output
  -V                 print version and exit
  -w <deadline>      reply wait <deadline> in seconds
  -W <timeout>       time to wait for response

IPv4 options:
  -4                 use IPv4
  -b                 allow pinging broadcast
  -R                 record route
  -T <timestamp>     define timestamp, can be one of <tsonly|tsandaddr|tsprespec>

IPv6 options:
  -6                 use IPv6
  -F <flowlabel>     define flow label, default is random
  -N <nodeinfo opt>  use icmp6 node info query, try <help> as argument
```

```
For more details see ping(8).
```

2.2 - netstat -i

To view network statistics, use the **netstat** command:

```
[root@redhat9 ~]# netstat -i
Kernel Interface table
Iface             MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
ens18            1500    18785      0      0 0         12157      0      0      0 BMRU
lo              65536      105      0      0 0           105      0      0      0 LRU
```

Command Line Switches

The command line switches for this command are :

```
[root@redhat9 ~]# netstat --help
usage: netstat [-vWeenNcCF] [<Af>] -r         netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -I[<Iface>] | [-veenNac] -i | [-cnNe] -M | -s [-6tuw] } [delay]

        -r, --route              display routing table
        -I, --interfaces=<Iface> display interface table for <Iface>
        -i, --interfaces         display interface table
        -g, --groups             display multicast group memberships
        -s, --statistics         display networking statistics (like SNMP)
        -M, --masquerade         display masqueraded connections

        -v, --verbose            be verbose
        -W, --wide               don't truncate IP addresses
        -n, --numeric            don't resolve names
```

```
       --numeric-hosts          don't resolve host names
       --numeric-ports          don't resolve port names
       --numeric-users          don't resolve user names
       -N, --symbolic           resolve hardware names
       -e, --extend             display other/more information
       -p, --programs           display PID/Program name for sockets
       -o, --timers             display timers
       -c, --continuous         continuous listing

       -l, --listening          display listening server sockets
       -a, --all                display all sockets (default: connected)
       -F, --fib                display Forwarding Information Base (default)
       -C, --cache              display routing cache instead of FIB
       -Z, --context            display SELinux security context for sockets

  <Socket>={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw}
           {-x|--unix} --ax25 --ipx --netrom
  <AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
```

2.3 - traceroute

The ping command is the basis of the **traceroute** command. This command is used to find out the route taken to reach a given host:

```
[root@redhat9 ~]# traceroute www.ittraining.team
bash: traceroute: command not found...
Install package 'traceroute' to provide command 'traceroute'? [N/y] y


 * Waiting in queue...
```

```
 * Loading list of packages....
The following packages have to be installed:
 traceroute-3:2.1.0-18.el9.x86_64       Traces the route taken by packets over an IPv4/IPv6 network
Proceed with changes? [N/y] y



 * Waiting in queue...
 * Waiting for authentication...
 * Waiting in queue...
 * Downloading packages...
 * Requesting data...
 * Testing changes...
 * Installing packages...
traceroute to www.ittraining.team (136.143.190.199), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.1)  0.437 ms  0.396 ms  0.380 ms
 2  51.79.19.252 (51.79.19.252)  0.554 ms  0.689 ms  0.818 ms
 3  10.161.82.50 (10.161.82.50)  0.372 ms 10.161.82.52 (10.161.82.52)  0.444 ms 10.161.82.50 (10.161.82.50)
0.372 ms
 4  10.34.98.50 (10.34.98.50)  0.483 ms  0.580 ms  0.973 ms
 5  10.74.8.92 (10.74.8.92)  0.232 ms 10.74.8.90 (10.74.8.90)  9.825 ms 10.74.8.94 (10.74.8.94)  0.206 ms
 6  10.95.81.10 (10.95.81.10)  0.712 ms 10.95.81.8 (10.95.81.8)  1.103 ms 10.95.81.10 (10.95.81.10)  1.435 ms
 7  be101.chi-ch2-sbb1-8k.il.us (198.27.73.207)  17.425 ms be101.chi-ch2-sbb2-8k.il.us (192.99.146.141)  17.089
ms be101.chi-ch2-sbb1-8k.il.us (198.27.73.207)  17.055 ms
 8  * * *
 9  10.200.1.1 (10.200.1.1)  66.593 ms  68.592 ms *
10  * 10.200.1.1 (10.200.1.1)  68.518 ms *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
```

```
19  * * *
20  * * *
21  * *^C
```

Command Line Switches

The command line switches for this command are :

```
[root@redhat9 ~]# traceroute --help
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ] [ -
p port ] [ -t tos ] [ -l flow_label ] [ -w MAX,HERE,NEAR ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [ --
fwmark=num ] host [ packetlen ]
Options:
  -4                          Use IPv4
  -6                          Use IPv6
  -d  --debug                 Enable socket level debugging
  -F  --dont-fragment         Do not fragment packets
  -f first_ttl  --first=first_ttl
                              Start from the first_ttl hop (instead from 1)
  -g gate,...  --gateway=gate,...
                              Route packets through the specified gateway
                              (maximum 8 for IPv4 and 127 for IPv6)
  -I  --icmp                  Use ICMP ECHO for tracerouting
  -T  --tcp                   Use TCP SYN for tracerouting (default port is 80)
  -i device  --interface=device
                              Specify a network interface to operate with
  -m max_ttl  --max-hops=max_ttl
                              Set the max number of hops (max TTL to be
                              reached). Default is 30
  -N squeries  --sim-queries=squeries
                              Set the number of probes to be tried
                              simultaneously (default is 16)
```

```
-n                           Do not resolve IP addresses to their domain names
-p port  --port=port         Set the destination port to use. It is either
                             initial udp port value for "default" method
                             (incremented by each probe, default is 33434), or
                             initial seq for "icmp" (incremented as well,
                             default from 1), or some constant destination
                             port for other methods (with default of 80 for
                             "tcp", 53 for "udp", etc.)
-t tos  --tos=tos            Set the TOS (IPv4 type of service) or TC (IPv6
                             traffic class) value for outgoing packets
-l flow_label  --flowlabel=flow_label
                             Use specified flow_label for IPv6 packets
-w MAX,HERE,NEAR  --wait=MAX,HERE,NEAR
                             Wait for a probe no more than HERE (default 3)
                             times longer than a response from the same hop,
                             or no more than NEAR (default 10) times than some
                             next hop, or MAX (default 5.0) seconds (float
                             point values allowed too)
-q nqueries  --queries=nqueries
                             Set the number of probes per each hop. Default is
                             3
-r                           Bypass the normal routing and send directly to a
                             host on an attached network
-s src_addr  --source=src_addr
                             Use source src_addr for outgoing packets
-z sendwait  --sendwait=sendwait
                             Minimal time interval between probes (default 0).
                             If the value is more than 10, then it specifies a
                             number in milliseconds, else it is a number of
                             seconds (float point values allowed too)
-e  --extensions             Show ICMP extensions (if present), including MPLS
-A  --as-path-lookups        Perform AS path lookups in routing registries and
                             print results directly after the corresponding
                             addresses
```

```
  -M name  --module=name      Use specified module (either builtin or external)
                              for traceroute operations. Most methods have
                              their shortcuts (`-I' means `-M icmp' etc.)
  -O OPTS,...  --options=OPTS,...
                              Use module-specific option OPTS for the
                              traceroute module. Several OPTS allowed,
                              separated by comma. If OPTS is "help", print info
                              about available options
  --sport=num                 Use source port num for outgoing packets. Implies
                              `-N 1'
  --fwmark=num                Set firewall mark for outgoing packets
  -U  --udp                   Use UDP to particular port for tracerouting
                              (instead of increasing the port per each probe),
                              default port is 53
  -UL                         Use UDPLITE for tracerouting (default dest port
                              is 53)
  -D  --dccp                  Use DCCP Request for tracerouting (default port
                              is 33434)
  -P prot  --protocol=prot    Use raw packet of protocol prot for tracerouting
  --mtu                       Discover MTU along the path being traced. Implies
                              `-F -N 1'
  --back                      Guess the number of hops in the backward path and
                              print if it differs
  -V  --version               Print version info and exit
  --help                      Read this help and exit

Arguments:
+     host          The host to traceroute to
      packetlen     The full packet length (default is the length of an IP
                    header plus 40). Can be ignored or increased to a minimal
                    allowed value
```

2.4 - tracepath

Another command used to find out the route taken to access a given site is **tracepath** :

```
[root@redhat9 ~]# tracepath www.ittraining.team
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              0.199ms
 1:  _gateway                                              0.138ms
 2:  51.79.19.252                                          0.651ms
 3:  10.161.82.52                                          0.704ms
 4:  10.34.98.56                                           0.628ms
 5:  10.74.8.88                                            0.301ms
 6:  10.95.81.8                                           36.365ms
 7:  be101.chi-ch2-sbb2-8k.il.us                          17.152ms
 8:  be101.chi-ch2-sbb2-8k.il.us                          17.442ms asymm  7
 9:  10.200.1.1                                           69.375ms
10:  10.200.1.1                                           68.618ms asymm  9
11:  no reply
12:  no reply
13:  no reply
14:  no reply
^C
```

Command Line Switches

The command line switches for this command are :

```
[root@redhat9 ~]# tracepath --help
tracepath: invalid option -- '-'

Usage
  tracepath [options] <destination>
```

```
Options:
  -4             use IPv4
  -6             use IPv6
  -b             print both name and ip
  -l <length>    use packet <length>
  -m <hops>      use maximum <hops>
  -n             no dns name resolution
  -p <port>      use destination <port>
  -V             print version and exit
  <destination>  dns name or ip address

For more details see tracepath(8).
```

LAB #3 - Remote Connections

3.1 - Telnet

Important - If the **telnet** command is not installed, install it using the **dnf install telnet** command as root.

The **telnet** command is used to establish a remote connection with a telnet server:

```
# telnet ip_number
```

Important - The telnet service amounts to a redirection of the standard input and output channels. Note that the connection is **not** secure. To close the connection, enter the **exit** command. The telnet command does not offer file transfer services. For this, use the **ftp** command.

Command Line Switches

The command line switches for this command are :

```
[root@redhat9 ~]# telnet --help
telnet: invalid option -- '-'
Usage: telnet [-4] [-6] [-8] [-E] [-L] [-S tos] [-a] [-c] [-d] [-e char] [-l user]
        [-n tracefile] [-b hostalias ] [-r]
 [host-name [port]]
```

3.2 - wget

The **wget** command is used to retrieve a file via http, https or ftp :

```
[root@redhat9 ~]# wget
https://www.dropbox.com/scl/fi/c0cbo91y2i7qwjexeldgt/wget_file.txt?rlkey=g8fgje9z8oeqgb4nd2g7x3wkx
--2024-09-29 13:56:10--
https://www.dropbox.com/scl/fi/c0cbo91y2i7qwjexeldgt/wget_file.txt?rlkey=g8fgje9z8oeqgb4nd2g7x3wkx
Resolving www.dropbox.com (www.dropbox.com)... 162.125.11.18, 2620:100:6050:18::a27d:b12
Connecting to www.dropbox.com (www.dropbox.com)|162.125.11.18|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location:
https://uc22f408c6cacfcc03fb4dbb269d.dl.dropboxusercontent.com/cd/0/inline/CbfNI4-pa7NlBqzKN3_KWHUoXFn8kcMiV99ekp
nDl2iVRanx9yx3YdpDcC5FHk8MJqHFfDnbPFeENko4TWJUAKMwZJ82s18b69iKmgbpMCFyd5oGQHLZs6uB3xy0_nmJl59Ru2MjCGeyEo9ikQ3Uaaq
Y/file# [following]
--2024-09-29 13:56:11--
https://uc22f408c6cacfcc03fb4dbb269d.dl.dropboxusercontent.com/cd/0/inline/CbfNI4-pa7NlBqzKN3_KWHUoXFn8kcMiV99ekp
nDl2iVRanx9yx3YdpDcC5FHk8MJqHFfDnbPFeENko4TWJUAKMwZJ82s18b69iKmgbpMCFyd5oGQHLZs6uB3xy0_nmJl59Ru2MjCGeyEo9ikQ3Uaaq
Y/file
Resolving uc22f408c6cacfcc03fb4dbb269d.dl.dropboxusercontent.com
(uc22f408c6cacfcc03fb4dbb269d.dl.dropboxusercontent.com)... 162.125.11.15, 2620:100:6050:15::a27d:b0f
Connecting to uc22f408c6cacfcc03fb4dbb269d.dl.dropboxusercontent.com
```

```
(uc22f408c6cacfcc03fb4dbb269d.dl.dropboxusercontent.com)|162.125.11.15|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46 [text/plain]
Saving to: ‘wget_file.txt?rlkey=g8fgje9z8oeqgb4nd2g7x3wkx’

wget_file.txt?rlkey=g8fgje9z8oeqgb4nd2g7x3wkx
100%[=========================================================================================================
========>]      46  --.-KB/s    in 0s

2024-09-29 13:56:11 (40.9 MB/s) - ‘wget_file.txt?rlkey=g8fgje9z8oeqgb4nd2g7x3wkx’ saved [46/46]

[root@redhat9 ~]# cat wget_file.txt?rlkey=g8fgje9z8oeqgb4nd2g7x3wkx
This is a file retrieved by the wget command.
```

Command Line Switches

The command line switches for this command are :

```
[root@redhat9 ~]# wget --help
GNU Wget 1.21.1, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...

Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version                   display the version of Wget and exit
  -h,  --help                      print this help
  -b,  --background                go to background after startup
  -e,  --execute=COMMAND           execute a `.wgetrc'-style command

Logging and input file:
  -o,  --output-file=FILE          log messages to FILE
  -a,  --append-output=FILE        append messages to FILE
```

```
  -d,  --debug                     print lots of debugging information
  -q,  --quiet                     quiet (no output)
  -v,  --verbose                   be verbose (this is the default)
  -nv, --no-verbose                turn off verboseness, without being quiet
       --report-speed=TYPE         output bandwidth as TYPE.  TYPE can be bits
  -i,  --input-file=FILE           download URLs found in local or external FILE
  -F,  --force-html                treat input file as HTML
  -B,  --base=URL                  resolves HTML input-file links (-i -F)
                                     relative to URL
       --config=FILE               specify config file to use
       --no-config                 do not read any config file
       --rejected-log=FILE         log reasons for URL rejection to FILE

Download:
  -t,  --tries=NUMBER              set number of retries to NUMBER (0 unlimits)
       --retry-connrefused         retry even if connection is refused
       --retry-on-http-error=ERRORS    comma-separated list of HTTP errors to retry
  -O,  --output-document=FILE      write documents to FILE
  -nc, --no-clobber                skip downloads that would download to
                                     existing files (overwriting them)
       --no-netrc                  don't try to obtain credentials from .netrc
  -c,  --continue                  resume getting a partially-downloaded file
       --start-pos=OFFSET          start downloading from zero-based position OFFSET
       --progress=TYPE             select progress gauge type
       --show-progress             display the progress bar in any verbosity mode
  -N,  --timestamping              don't re-retrieve files unless newer than
                                     local
       --no-if-modified-since      don't use conditional if-modified-since get
                                     requests in timestamping mode
       --no-use-server-timestamps  don't set the local file's timestamp by
                                     the one on the server
  -S,  --server-response           print server response
       --spider                    don't download anything
  -T,  --timeout=SECONDS           set all timeout values to SECONDS
```

```
       --dns-timeout=SECS           set the DNS lookup timeout to SECS
       --connect-timeout=SECS       set the connect timeout to SECS
       --read-timeout=SECS          set the read timeout to SECS
  -w,  --wait=SECONDS               wait SECONDS between retrievals
                                      (applies if more then 1 URL is to be retrieved)
       --waitretry=SECONDS          wait 1..SECONDS between retries of a retrieval
                                      (applies if more then 1 URL is to be retrieved)
       --random-wait                wait from 0.5*WAIT...1.5*WAIT secs between retrievals
                                      (applies if more then 1 URL is to be retrieved)
       --no-proxy                   explicitly turn off proxy
  -Q,  --quota=NUMBER               set retrieval quota to NUMBER
       --bind-address=ADDRESS       bind to ADDRESS (hostname or IP) on local host
       --limit-rate=RATE            limit download rate to RATE
       --no-dns-cache               disable caching DNS lookups
       --restrict-file-names=OS     restrict chars in file names to ones OS allows
       --ignore-case                ignore case when matching files/directories
  -4,  --inet4-only                 connect only to IPv4 addresses
  -6,  --inet6-only                 connect only to IPv6 addresses
       --prefer-family=FAMILY       connect first to addresses of specified family,
                                      one of IPv6, IPv4, or none
       --user=USER                  set both ftp and http user to USER
       --password=PASS              set both ftp and http password to PASS
       --ask-password               prompt for passwords
       --use-askpass=COMMAND        specify credential handler for requesting
                                      username and password.  If no COMMAND is
                                      specified the WGET_ASKPASS or the SSH_ASKPASS
                                      environment variable is used.
       --no-iri                     turn off IRI support
       --local-encoding=ENC         use ENC as the local encoding for IRIs
       --remote-encoding=ENC        use ENC as the default remote encoding
       --unlink                     remove file before clobber
       --xattr                      turn on storage of metadata in extended file attributes

Directories:
```

```
  -nd, --no-directories            don't create directories
  -x,  --force-directories         force creation of directories
  -nH, --no-host-directories       don't create host directories
       --protocol-directories      use protocol name in directories
  -P,  --directory-prefix=PREFIX   save files to PREFIX/..
       --cut-dirs=NUMBER           ignore NUMBER remote directory components

HTTP options:
       --http-user=USER            set http user to USER
       --http-password=PASS        set http password to PASS
       --no-cache                  disallow server-cached data
       --default-page=NAME         change the default page name (normally
                                     this is 'index.html'.)
  -E,  --adjust-extension          save HTML/CSS documents with proper extensions
       --ignore-length             ignore 'Content-Length' header field
       --header=STRING             insert STRING among the headers
       --compression=TYPE          choose compression, one of auto, gzip and none. (default: none)
       --max-redirect              maximum redirections allowed per page
       --proxy-user=USER           set USER as proxy username
       --proxy-password=PASS       set PASS as proxy password
       --referer=URL               include 'Referer: URL' header in HTTP request
       --save-headers              save the HTTP headers to file
  -U,  --user-agent=AGENT          identify as AGENT instead of Wget/VERSION
       --no-http-keep-alive        disable HTTP keep-alive (persistent connections)
       --no-cookies                don't use cookies
       --load-cookies=FILE         load cookies from FILE before session
       --save-cookies=FILE         save cookies to FILE after session
       --keep-session-cookies      load and save session (non-permanent) cookies
       --post-data=STRING          use the POST method; send STRING as the data
       --post-file=FILE            use the POST method; send contents of FILE
       --method=HTTPMethod         use method "HTTPMethod" in the request
       --body-data=STRING          send STRING as data. --method MUST be set
       --body-file=FILE            send contents of FILE. --method MUST be set
       --content-disposition       honor the Content-Disposition header when
```

```
                                  choosing local file names (EXPERIMENTAL)
       --content-on-error         output the received content on server errors
       --auth-no-challenge        send Basic HTTP authentication information
                                  without first waiting for the server's
                                  challenge

HTTPS (SSL/TLS) options:
       --secure-protocol=PR       choose secure protocol, one of auto, SSLv2,
                                  SSLv3, TLSv1, TLSv1_1, TLSv1_2 and PFS
       --https-only               only follow secure HTTPS links
       --no-check-certificate     don't validate the server's certificate
       --certificate=FILE         client certificate file
       --certificate-type=TYPE    client certificate type, PEM or DER
       --private-key=FILE         private key file
       --private-key-type=TYPE    private key type, PEM or DER
       --ca-certificate=FILE      file with the bundle of CAs
       --ca-directory=DIR         directory where hash list of CAs is stored
       --crl-file=FILE            file with bundle of CRLs
       --pinnedpubkey=FILE/HASHES Public key (PEM/DER) file, or any number
                                  of base64 encoded sha256 hashes preceded by
                                  'sha256//' and separated by ';', to verify
                                  peer against

       --ciphers=STR          Set the priority string (GnuTLS) or cipher list string (OpenSSL) directly.
                                  Use with care. This option overrides --secure-protocol.
                                  The format and syntax of this string depend on the specific SSL/TLS engine.
HSTS options:
       --no-hsts                  disable HSTS
       --hsts-file                path of HSTS database (will override default)

FTP options:
       --ftp-user=USER            set ftp user to USER
       --ftp-password=PASS        set ftp password to PASS
       --no-remove-listing        don't remove '.listing' files
```

```
       --no-glob                  turn off FTP file name globbing
       --no-passive-ftp           disable the "passive" transfer mode
       --preserve-permissions     preserve remote file permissions
       --retr-symlinks            when recursing, get linked-to files (not dir)

FTPS options:
       --ftps-implicit               use implicit FTPS (default port is 990)
       --ftps-resume-ssl             resume the SSL/TLS session started in the control connection when
                                       opening a data connection
       --ftps-clear-data-connection  cipher the control channel only; all the data will be in plaintext
       --ftps-fallback-to-ftp        fall back to FTP if FTPS is not supported in the target server
WARC options:
       --warc-file=FILENAME       save request/response data to a .warc.gz file
       --warc-header=STRING       insert STRING into the warcinfo record
       --warc-max-size=NUMBER     set maximum size of WARC files to NUMBER
       --warc-cdx                 write CDX index files
       --warc-dedup=FILENAME      do not store records listed in this CDX file
       --no-warc-compression      do not compress WARC files with GZIP
       --no-warc-digests          do not calculate SHA1 digests
       --no-warc-keep-log         do not store the log file in a WARC record
       --warc-tempdir=DIRECTORY   location for temporary files created by the
                                    WARC writer

Recursive download:
  -r,  --recursive                specify recursive download
  -l,  --level=NUMBER             maximum recursion depth (inf or 0 for infinite)
       --delete-after             delete files locally after downloading them
  -k,  --convert-links            make links in downloaded HTML or CSS point to
                                    local files
       --convert-file-only        convert the file part of the URLs only (usually known as the basename)
       --backups=N                before writing file X, rotate up to N backup files
  -K,  --backup-converted         before converting file X, back up as X.orig
  -m,  --mirror                   shortcut for -N -r -l inf --no-remove-listing
  -p,  --page-requisites          get all images, etc. needed to display HTML page
```

```
       --strict-comments           turn on strict (SGML) handling of HTML comments

Recursive accept/reject:
  -A,  --accept=LIST               comma-separated list of accepted extensions
  -R,  --reject=LIST               comma-separated list of rejected extensions
       --accept-regex=REGEX        regex matching accepted URLs
       --reject-regex=REGEX        regex matching rejected URLs
       --regex-type=TYPE           regex type (posix|pcre)
  -D,  --domains=LIST              comma-separated list of accepted domains
       --exclude-domains=LIST      comma-separated list of rejected domains
       --follow-ftp                follow FTP links from HTML documents
       --follow-tags=LIST          comma-separated list of followed HTML tags
       --ignore-tags=LIST          comma-separated list of ignored HTML tags
  -H,  --span-hosts                go to foreign hosts when recursive
  -L,  --relative                  follow relative links only
  -I,  --include-directories=LIST  list of allowed directories
       --trust-server-names        use the name specified by the redirection
                                     URL's last component
  -X,  --exclude-directories=LIST  list of excluded directories
  -np, --no-parent                 don't ascend to the parent directory

Email bug reports, questions, discussions to <bug-wget@gnu.org>
and/or open issues at https://savannah.gnu.org/bugs/?func=additem&group=wget.
```

3.3 - ftp

> **Important** - If the **ftp** command is not installed, install it using the **dnf install ftp** command as root.

The **ftp** command is used to transfer files. Once connected, use the **help** command to display the list of available commands:

```
[root@redhat9 ~]# ftp
ftp> help
Commands may be abbreviated.  Commands are:

!               debug           mdir            sendport        site
$               dir             mget            put             size
account         disconnect      mkdir           pwd             status
append          exit            mls             quit            struct
ascii           form            mode            quote           system
bell            get             modtime         recv            sunique
binary          glob            mput            reget           tenex
bye             hash            newer           rstatus         tick
case            help            nmap            rhelp           trace
cd              idle            nlist           rename          type
cdup            image           ntrans          reset           user
chmod           lcd             open            restart         umask
close           ls              prompt          rmdir           verbose
cr              macdef          passive         runique         ?
delete          mdelete         proxy           send
ftp>
```

The **!** character is used to execute a command on the client machine

```
ftp> !pwd
/root
```

To transfer a file to the server, use the **put** command:

```
ftp> put local_file_name remote_file_name
```

You can also transfer several files at once using the **mput** command. In this case, enter the following command:

```
ftp> mput name*.*
```

To transfer a file from the server, use the **get** command:

```
ftp> get filename
```

You can also transfer several files at once using the **mget** command (see the **mput** command above).

To delete a file on the server, use the **del** command:

```
ftp> del filename
```

To close the session, use the **quit** command:

```
ftp> quit
[root@redhat9 ~]#
```

3.4 - SSH

Overview

The **ssh** command is the successor and replacement for the **rlogin** command. It is used to establish secure connections with a remote machine. SSH has five players:

- the **SSH server**
  - The sshd daemon, which handles client authentication and authorisation,
- The **SSH client**.
  - ssh or scp, which connects and talks to the server,
- The **session**, which represents the current connection and starts immediately after successful authentication,
- The **keys**
  - **Asymmetric** and persistent user key pairs which ensure a user's identity and which are stored on the hard disk,
  - **Asymmetric and persistent** host key guaranteeing the identity of the server and stored on the hard disk.
  - **Temporary asymmetric server key** used by the SSH1 protocol to encrypt the session key,
  - **Symmetric session key** which is generated at random and is used to encrypt the communication between the client and the server. It is

destroyed at the end of the session. SSH-1 uses a single key, while SSH-2 uses one key for each direction of communication,
- The **known hosts database** which stores the keys of previous connections.

SSH works as follows to set up a secure channel:

- The client contacts the server on port 22,
- The client and server exchange their versions of SSH. If the two versions do not match, one of them terminates the process,
- The SSH server identifies itself to the client by providing :
  - Its host key,
  - Its server key,
  - A random sequence of eight bytes to be included in future client responses,
  - A list of encryption, compression and authentication methods,
- The client and server produce an identical identifier, a 128-bit MD5 hash containing the host key, the server key and the random sequence,
- The client generates its symmetrical session key and encrypts it twice, once with the server's host key and the second time with the server key. The client sends this key to the server along with the random sequence and a choice of supported algorithms,
- The server decrypts the session key,
- The client and server set up the secure channel.

SSH-1

SSH-1 uses an RSA1 key pair. It ensures data integrity by means of a **Cyclic Redundancy Check** (CRC) and is a so-called **monolithic** block.

In order to identify itself, the client tries each of the following six methods:

- **Kerberos**,
- **Rhosts**,
- **RhostsRSA**,
- Using **asymmetric keys**,
- **TIS**,
- Using a **password**.

SSH-2

SSH-2 uses **DSA** or **RSA**. It ensures data integrity using the **HMAC** algorithm. SSH-2 is organised into three **layers**:

- **SSH-TRANS** - Transport Layer Protocol,
- **SSH-AUTH** - Authentication Protocol,
- **SSH-CONN** - Connection Protocol.

SSH-2 differs from SSH-1 mainly in the authentication phase.

There are three authentication methods:

- Using **asymmetric keys**,
- **RhostsRSA**,
- Using a **password**.

Command Line Switches

The command line switches for this command are :

```
[root@redhat9 ~]# ssh --help
unknown option -- -
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

Password Authentication

The user provides the ssh client with a password. The ssh client transmits it securely to the ssh server and then the server checks the password and accepts it or not.

Advantages:

- No asymmetric key configuration required.

Disadvantages:

- The user must provide a login and password for each connection,
- Less secure than asymmetric keys.

Asymmetric Key Authentication

- The **client** sends the server an asymmetric key authentication request containing the key module to be used,
- The **server** looks for a match for this module in the **~/.ssh/authorized_keys** authorised keys file,
  - If no match is found, the server terminates communication,
  - If a match is found, the server generates a 256-bit random string called a **challenge** and encrypts it with the client's **public key**,
- The **client** receives the challenge and decrypts it using the private part of its key. It combines the challenge with the session identifier and encrypts the result. It then sends the encrypted result to the server.
- The **server** generates the same hash and compares it with the one received from the client. If the two hashes are identical, authentication is successful.

Server Configuration

The server is configured in the **/etc/ssh/sshd_config** file:

```
[root@redhat9 ~]# cat /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $
```

```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO
```

```
# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes
```

```
# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
#GSSAPIEnablek5users no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
# WARNING: 'UsePAM no' is not supported in RHEL and may cause several
# problems.
#UsePAM no

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
```

```
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem       sftp    /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
```

To secure the ssh server, add or modify the following directives :

```
AllowGroups adm
Banner /etc/issue.net
HostbasedAuthentication no
IgnoreRhosts yes
LoginGraceTime 60
```

```
LogLevel INFO
PermitEmptyPasswords no
PermitRootLogin no
PrintLastLog yes
Protocol 2
StrictModes yes
X11Forwarding no
```

Your file will look like this:

```
[root@redhat9 tmp]# vi sshd_config

[root@redhat9 tmp]# cat sshd_config
AllowGroups adm
Banner /etc/issue.net
HostbasedAuthentication no
IgnoreRhosts yes
LoginGraceTime 60
LogLevel INFO
PermitEmptyPasswords no
PermitRootLogin no
PrintLastLog yes
Protocol 2
StrictModes yes
X11Forwarding no
Include /etc/ssh/sshd_config.d/*.conf
AuthorizedKeysFile      .ssh/authorized_keys
Subsystem       sftp    /usr/libexec/openssh/sftp-server
```

Rename the **/etc/ssh/sshd_config** file to **/etc/ssh/sshd_config.old** :

```
[root@redhat9 tmp]# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old
```

Copy the file **/tmp/sshd_config** to **/etc/ssh/** :

```
[root@redhat9 tmp]# cp /tmp/sshd_config /etc/ssh
cp: overwrite ‘/etc/ssh/sshd_config’? y
```

Restart the sshd service:

```
[root@redhat9 tmp]# systemctl restart sshd

[root@redhat9 tmp]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-29 14:06:49 CEST; 9s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 5560 (sshd)
      Tasks: 1 (limit: 48800)
     Memory: 1.4M
        CPU: 13ms
     CGroup: /system.slice/sshd.service
             └─5560 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 29 14:06:49 redhat9.ittraining.loc systemd[1]: Starting OpenSSH server daemon...
Sep 29 14:06:49 redhat9.ittraining.loc sshd[5560]: Server listening on 0.0.0.0 port 22.
Sep 29 14:06:49 redhat9.ittraining.loc sshd[5560]: Server listening on :: port 22.
Sep 29 14:06:49 redhat9.ittraining.loc systemd[1]: Started OpenSSH server daemon.
```

Put the **trainee** user in the **adm** group:

```
[root@redhat9 tmp]# groups trainee
trainee : trainee

[root@redhat9 tmp]# usermod -aG adm trainee

[root@redhat9 tmp]# groups trainee
```

```
trainee: trainee adm
```

To generate the server keys, enter the following command as **root**. Note that the passphrase must be **empty**.

```
[root@redhat9 tmp]# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa): /etc/ssh/ssh_host_dsa_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_dsa_key
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub
The key fingerprint is:
SHA256:VWOX1JHwvNZHo8kC6OLgKXypZI97W56JBiKwc9YjFpk root@redhat9.ittraining.loc
The key's randomart image is:
+---[DSA 1024]----+
|            +o++o|
|         .  o o+..|
|   o   . ..    +.|
|. E   .  .. . o.+|
|.. + . .S  . + oo|
|*.* B .     . . .|
|oX.* +           |
|o *o+ o          |
| +++.+           |
+----[SHA256]-----+
```

In the same way, it is possible to generate keys in the **RSA**, **ECDSA** and **ED25519** formats:

```
[root@redhat9 tmp]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /etc/ssh/ssh_host_rsa_key
/etc/ssh/ssh_host_rsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_rsa_key
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub
The key fingerprint is:
SHA256:SjQBVxLP5mtWmsHUWN0j1FvgokkEmKbEftzWc1ZlrAM root@redhat9.ittraining.loc
The key's randomart image is:
+---[RSA 3072]----+
|   ...==o.oo.o.+o|
|    o.+= =  E.=.o|
|   o +o.*.o .+.= |
|    o.o*o.ooo.+  |
|     ...S ++   . |
|     . . *       |
|      . *        |
|       o         |
|                 |
+----[SHA256]-----+

[root@redhat9 tmp]# ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/root/.ssh/id_ecdsa): /etc/ssh/ssh_host_ecdsa_key
/etc/ssh/ssh_host_ecdsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_ecdsa_key
Your public key has been saved in /etc/ssh/ssh_host_ecdsa_key.pub
The key fingerprint is:
SHA256:auaDgRcvbqpC5SkGadt7AizXVta2A8w5hYwVgUPXNlc root@redhat9.ittraining.loc
The key's randomart image is:
+---[ECDSA 256]---+
|   ..===   .E    |
|    +.o = .      |
| .   + = o       |
```

```
|o. .. O o        |
|+ =o.= +S.       |
|.Bo+* ..o        |
|+.o+.++  .       |
|.  o+=.          |
|o..oo ..         |
+----[SHA256]-----+

[root@redhat9 tmp]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): /etc/ssh/ssh_host_ed25519_key
/etc/ssh/ssh_host_ed25519_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_ed25519_key
Your public key has been saved in /etc/ssh/ssh_host_ed25519_key.pub
The key fingerprint is:
SHA256:UwUjA5Ln700GWzhNxvIvdNafLOhD1/hdrMigOvARQwI root@redhat9.ittraining.loc
The key's randomart image is:
+--[ED25519 256]--+
|  E.....o.o..    |
|    o.o .o+o     |
|     =   B.  .   |
|      + +.= o .  |
|       +S* +. +o.|
|    . . o.=o.+ ++|
|     o o =o+..o.o|
|      o o .oo ...|
|      .o    .    |
+----[SHA256]-----+
```

Generated public keys have the **.pub** extension. Private keys do not have a file extension:

```
[root@redhat9 tmp]# ls /etc/ssh
moduli      ssh_config.d  sshd_config.d    ssh_host_dsa_key      ssh_host_ecdsa_key      ssh_host_ed25519_key
ssh_host_rsa_key
ssh_config  sshd_config   sshd_config.old  ssh_host_dsa_key.pub  ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub
ssh_host_rsa_key.pub
```

Now restart the sshd service:

```
[root@redhat9 tmp]# systemctl restart sshd.service

[root@redhat9 tmp]# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-09-29 14:14:14 CEST; 13s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 5583 (sshd)
      Tasks: 1 (limit: 48800)
     Memory: 1.3M
        CPU: 12ms
     CGroup: /system.slice/sshd.service
             └─5583 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 29 14:14:14 redhat9.ittraining.loc systemd[1]: sshd.service: Deactivated successfully.
Sep 29 14:14:14 redhat9.ittraining.loc sshd[5583]: Server listening on 0.0.0.0 port 22.
Sep 29 14:14:14 redhat9.ittraining.loc systemd[1]: Stopped OpenSSH server daemon.
Sep 29 14:14:14 redhat9.ittraining.loc sshd[5583]: Server listening on :: port 22.
Sep 29 14:14:14 redhat9.ittraining.loc systemd[1]: Starting OpenSSH server daemon...
Sep 29 14:14:14 redhat9.ittraining.loc systemd[1]: Started OpenSSH server daemon.
```

Client Configuration

Now enter the following commands as **trainee** :

Important - When generating keys, the passphrase must be **empty**.

```
[root@redhat9 tmp]# exit
logout

[trainee@redhat9 ~]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_dsa):
Created directory '/home/trainee/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_dsa
Your public key has been saved in /home/trainee/.ssh/id_dsa.pub
The key fingerprint is:
SHA256:TT4VSKqep7i/5t6FoPSQ88LmTo3eLRBBFxz3Cz2Kxi4 trainee@redhat9.ittraining.loc
The key's randomart image is:
+---[DSA 1024]----+
|   ...+o.....    |
|    .... +.  .   |
|     .  o = .    |
|    .o o * +     |
|    =.* S =      |
|   o.% o . .     |
|    E.B o .      |
|   = =o= .       |
|   .**Boo        |
+----[SHA256]-----+

[trainee@redhat9 ~]$ ssh-keygen -t rsa
```

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_rsa
Your public key has been saved in /home/trainee/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:75jUyGd9zw6fA5Z2KIZAacp+0b5KLeWpOONItt/4Z9k trainee@redhat9.ittraining.loc
The key's randomart image is:
+---[RSA 3072]----+
|        .        |
|       +         |
|    . + .        |
|     o o .       |
|    .   S..   o  |
|     . o+*.+ * . |
|   o  .o++X = =  |
|  o oo=.o@ E . *.|
|   oo=+=* .    o*|
+----[SHA256]-----+

[trainee@redhat9 ~]$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_ecdsa
Your public key has been saved in /home/trainee/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:xjl9G3ycaF3s9cuyCqHhPvUWyN1qzBGxJtZwFAFeRvw trainee@redhat9.ittraining.loc
The key's randomart image is:
+---[ECDSA 256]---+
|         .=Bo    |
|        ..o+   . |
```

```
|         .+ +   +|
|       . = * E +o|
|       .S.* O = o|
|      ..o=.* =. .|
|       o..+ =. o |
|      ..  .B  o  |
|       .. o...   |
+----[SHA256]-----+

[trainee@redhat9 ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_ed25519
Your public key has been saved in /home/trainee/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:qtFzx7O0swNGsRoP9Cy+cISURCvWYpANMn1WtljZ8WQ trainee@redhat9.ittraining.loc
The key's randomart image is:
+--[ED25519 256]--+
|== ooo+o..E      |
|ooo.==o.o+       |
|  =++o.o o.      |
| o o. = =        |
|     o BS        |
|    ..+.+.       |
|    .o+o..=      |
|     o.o oo+     |
|    .     ++     |
+----[SHA256]-----+
```

The generated keys will be placed in the **~/.ssh/** directory:

```
[trainee@redhat9 ~]$ ls .ssh
```

```
id_dsa id_dsa.pub id_ecdsa id_ecdsa.pub id_ed25519 id_ed25519.pub id_rsa id_rsa.pub
```

SSH Tunnels

The SSH protocol can be used to secure protocols such as telnet, pop3 and so on. You can create an SSH *tunnel* through which unsecured protocol communications pass.

The command to create an ssh tunnel takes the following form:

```
ssh -N -f account@host -Local_port:localhost:remote_port
```

In your case, you are going to create a tunnel in your own vm between port 15023 and port 23 :

```
[trainee@redhat9 ~]$ ssh -N -f trainee@localhost -L15023:localhost:23
The authenticity of host 'localhost (::1)' can't be established.
ED25519 key fingerprint is SHA256:UwUjA5Ln700GWzhNxvIvdNafLOhD1/hdrMigOvARQwI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.
\S
Kernel \r on an \m
trainee@localhost's password: trainee
```

Now install the telnet server:

```
[trainee@redhat9 ~]$ su -
Password:
[root@redhat9 ~]# dnf install telnet-server
Updating Subscription Management repositories.
Last metadata expiration check: 3:26:47 ago on Sun 29 Sep 2024 10:52:14 AM CEST.
Dependencies resolved.
================================================================================
```

```
================================================================================
 Package                                 Architecture                    Version
Repository                                                    Size
================================================================================
================================================================================
Installing:
 telnet-server                           x86_64                          1:0.17-85.el9
rhel-9-for-x86_64-appstream-rpms                              41 k

Transaction Summary
================================================================================
================================================================================
Install  1 Package

Total download size: 41 k
Installed size: 58 k
Is this ok [y/N]: y
Downloading Packages:
telnet-server-0.17-85.el9.x86_64.rpm
145 kB/s |  41 kB     00:00
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Total
144 kB/s |  41 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
1/1
  Installing       : telnet-server-1:0.17-85.el9.x86_64
1/1
  Running scriptlet: telnet-server-1:0.17-85.el9.x86_64
```

```
1/1
  Verifying        : telnet-server-1:0.17-85.el9.x86_64
1/1
Installed products updated.

Installed:
  telnet-server-1:0.17-85.el9.x86_64

Complete!
```

Telnet is neither started nor activated. It must be started and activated:

```
[root@redhat9 ~]# systemctl status telnet.socket
○ telnet.socket - Telnet Server Activation Socket
     Loaded: loaded (/usr/lib/systemd/system/telnet.socket; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:telnetd(8)
     Listen: [::]:23 (Stream)
   Accepted: 0; Connected: 0;

[root@redhat9 ~]# systemctl start telnet.socket

[root@redhat9 ~]# systemctl status telnet.socket
● telnet.socket - Telnet Server Activation Socket
     Loaded: loaded (/usr/lib/systemd/system/telnet.socket; disabled; preset: disabled)
     Active: active (listening) since Sun 2024-09-29 14:19:51 CEST; 13s ago
      Until: Sun 2024-09-29 14:19:51 CEST; 13s ago
       Docs: man:telnetd(8)
     Listen: [::]:23 (Stream)
   Accepted: 0; Connected: 0;
      Tasks: 0 (limit: 48800)
     Memory: 8.0K
        CPU: 700us
     CGroup: /system.slice/telnet.socket
```

```
Sep 29 14:19:51 redhat9.ittraining.loc systemd[1]: Listening on Telnet Server Activation Socket.
[root@redhat9 ~]# systemctl enable telnet.socket
Created symlink /etc/systemd/system/sockets.target.wants/telnet.socket → /usr/lib/systemd/system/telnet.socket.
```

Then connect via telnet to port 15023:

```
[root@redhat9 ~]# telnet localhost 15023
Trying ::1...
Connected to localhost.
Escape character is '^]'.

Kernel 5.14.0-427.37.1.el9_4.x86_64 on an x86_64
redhat9 login: trainee
Password: trainee
Last login: Sun Sep 29 12:26:00 from 10.0.2.1

[trainee@redhat9 ~]$ whoami
trainee

[trainee@redhat9 ~]$ pwd
/home/trainee
```

Important - Note that your telnet communication goes through the SSH tunnel.

3.5 - SCP

Overview

The **scp** command is the successor and replacement for the **rcp** command in the **remote** command family. It is used to make secure transfers from a

remote machine:

```
$ scp account@ip(machine_name):/remote_path/remote_file /local_path/local_file
```

or to a remote machine :

```
$ scp /local_path/local_file account@ip(hostname):/remote_path/remote_file
```

Usage

We are now going to use **scp** to download a file on the «server» :

Create the file **/home/trainee/scp_test** :

```
[trainee@redhat9 ~]$ touch scp-test

[trainee@redhat9 ~]$ exit
logout
Connection closed by foreign host.

[root@redhat9 ~]#
```

Retrieve the **scp_test** file using scp :

```
[root@redhat9 ~]# scp trainee@127.0.0.1:/home/trainee/scp-test .
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:UwUjA5Ln700GWzhNxvIvdNafLOhD1/hdrMigOvARQwI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '127.0.0.1' (ED25519) to the list of known hosts.
\S
Kernel \r on an \m
```

```
trainee@127.0.0.1's password: trainee

[root@redhat9 ~]# ls -l
total 2042944
-rw-------. 1 root    root          1226 Oct 19  2023  anaconda-ks.cfg
-rw-r--r--. 1 trainee trainee 2091941797 Oct 19  2023  ansible-automation-platform-setup-bundle-2.4-2.2-
x86_64.tar.gz
-rw-r--r--. 1 root    root            64 Sep 27 08:24  device.map
-rw-------. 1 root    root          7118 Sep 27 08:24  grub.cfg
drwxr-xr-x. 3 root    root            21 Oct 19  2023  home
-rw-r--r--. 1 root    root            98 Sep 27 08:23  montages.list
-rw-r--r--. 1 root    root          2109 Sep 25 16:20  passwd
-rw-r--r--. 1 root    root             0 Sep 29 14:24  scp-test
-rw-r--r--. 1 root    root           457 Sep 27 08:22  structure.list
-rw-r--r--. 1 root    root            46 Sep 29 13:56 'wget_file.txt?rlkey=g8fgje9z8oeqgb4nd2g7x3wkx'
```

3.6 - Setting up Asymmetric Keys

We now need to connect to the «server» using ssh and check that the ~/.ssh directory is present:

```
[root@redhat9 ~]# ssh -l trainee 127.0.0.1
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Sun Sep 29 14:21:21 2024 from localhost

[trainee@redhat9 ~]$ ls -la | grep .ssh
-rw-------.  1 trainee trainee   20 Sep 25 15:18 .lesshst
drwx------.  2 trainee trainee  188 Sep 29 14:18 .ssh
```

Important - If the remote .ssh folder does not exist in the user's home directory, it must be created with 700 file permissions. In your case, since your machine is the server **and** the client, the /home/trainee/.ssh folder **already** exists.

Next, you need to transfer the local file **.ssh/id_ecdsa.pub** from the «client» to the «server» by renaming it to **authorized_keys** :

```
[trainee@redhat9 ~]$ exit
logout
Connection to 127.0.0.1 closed.

[root@redhat9 ~]# exit
logout

[trainee@redhat9 ~]$ scp .ssh/id_ecdsa.pub trainee@127.0.0.1:/home/trainee/.ssh/authorized_keys
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:UwUjA5Ln700GWzhNxvIvdNafLOhD1/hdrMigOvARQwI.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: localhost
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '127.0.0.1' (ED25519) to the list of known hosts.
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
id_ecdsa.pub
100%  192   427.3KB/s   00:00
```

Connect via ssh :

```
[trainee@redhat9 ~]$ ssh -l trainee localhost
\S
Kernel \r on an \m
```

```
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Sun Sep 29 14:26:20 2024 from 127.0.0.1
[trainee@redhat9 ~]$
```

Important - When connecting to the server, the authentication procedure uses the asymmetric key pair in ecdsa format and no password is required.

Now insert the remaining public keys in the .ssh/authorized_keys file:

```
[trainee@redhat9 ~]$ cd .ssh

[trainee@redhat9 .ssh]$ ls
authorized_keys  id_dsa  id_dsa.pub  id_ecdsa  id_ecdsa.pub  id_ed25519  id_ed25519.pub  id_rsa  id_rsa.pub
known_hosts  known_hosts.old

[trainee@redhat9 .ssh]$ cat authorized_keys
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGJIMNJ1m+xIpYzfYwK7VpdCI9inhQx3wptO+z4Xsl3XYcb+WIXsEsJpKSyQn
Ov98HmfZVJWcqXaSBkE5mskFGI= trainee@redhat9.ittraining.loc

[trainee@redhat9 .ssh]$ cat id_rsa.pub >> authorized_keys

[trainee@redhat9 .ssh]$ cat id_dsa.pub >> authorized_keys

[trainee@redhat9 .ssh]$ cat id_ed25519.pub >> authorized_keys

[trainee@redhat9 .ssh]$ cat authorized_keys
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGJIMNJ1m+xIpYzfYwK7VpdCI9inhQx3wptO+z4Xsl3XYcb+WIXsEsJpKSyQn
Ov98HmfZVJWcqXaSBkE5mskFGI= trainee@redhat9.ittraining.loc
```

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDSBgNBWkHU0UWXablPEQLRXjcdG7WYN0651bSh122wAMpMo5j3Jjlxm+tCKILpU5kjmpLIC+YCVw3nNr2e8
RxIucxfEy1NIXrrCS0Ps3t5zsdsta/Z716BvX0x4tRuKqLPOMmwLWyvjYG4ixnECNSJTnHOmj122wxpdBX6HWJpJ+61NIvOnI+fAGTU6nPmb0jkan
NT/HZ27eDlqo88gNAXrRSt/Uc1bglNue5NGKn2TcuUEIau0tmZpguAcWfrESl4SK7bg0c+IJZHZw8T8Fkpo44T4oWj03/x+uWjpphOfFPux6nAiF4
+bRxFZnoUMvhe/WGX13Yxf0w3mymRyZXeSDyb0KF32Vy9hVh+0nVAIRtzwCc82aFtVdlJi9wW54tUwUlMkVFjVylJRNif37FwRd65BNp2eoN7v2IT
dZ+uWLGs3HJZKp+xLTHMZ3CZMkLyNsbdwyD5VAsD3dFB49W2voSC9DN5xlUWPp4m7TbNifa8b7nuQKP8p2FaVdr3YWc=
trainee@redhat9.ittraining.loc
ssh-dss
AAAAB3NzaC1kc3MAAACBAOlvM/inrDQNvmCMcWH4eTNud6egbiG4XysF++2If9Dx89mW8RWZmlvyiV8wRf71UyHHdiiBc/3SkYrka04l65F3GYH8B
Q0jiaf4WpzTvn1uuEjS2kO3+vrXW1JxYlWiOYyb44eufnrxK+qia6FkhC6Wmn6xnibmbwkBeXTXOQp7AAAAFQCYu7xH3JreaiS4bYuZ1b3MV4IAmQ
AAAIEApEwhXE8jn2SWk/tpQKkX5dATCa1K5T+XMEzunjeEr+w1F1tapptOFaaujmZeelNgBYT4LauNYRu5TuXmoDpOp2q1puQKmGjW3b6bQRN0Pal
o/rcPlI6NN3Efm242vhspWEl4fjYFoVxPfaG0ysTwjOmgM4TamcxgrYDclDc0hNUAAACBAJrqqa12g422N6YRw3CXbyMwSv2xagX09YjwvsbDBMyC
JtqoDg+6YavISLU3VQYJ+FmzBzObS2lkzk0yGMgqKOmnRIPaPi3HmpSPXp7828BU4lTsN4yv6zp4C1MIazvnE2rqBIVy1ZhCt9ADiCHZrRY2M/Czl
jfUhi/LinnznFVs trainee@redhat9.ittraining.loc
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKzhdX1reo1vStZd/lfAQ1Yjz7eo0qj7ir/f6jyGp4iG trainee@redhat9.ittraining.loc

[trainee@redhat9 .ssh]$ exit
logout
Connection to localhost closed.
[trainee@redhat9 ~]$
```
