Chapter 6:
Configuring Networking

Learning objectives

Upon completing this chapter, the learner should be able to:

- Configure networking and hostname resolution statically or dynamically
- Validate network address configuration, routing, ports, services
- Understand DNS resolving and troubleshooting
- Understand Network Time Protocol NTP
- Configure a system to use time service
- Configure NTP server and client

Key terms

NIC

ip

ipv4

ipv6

protocol

port

subnet mask

DNS

DHCP

connection

interface

FQDN

routing

hostname

NTP

NTP server

NTP primary server

NTP secondary server

NTP peer

NTP client

Table of content

1. Setting up networking

Networking is an important foundation because without it, you aren't able to communicate with the outside world or the rest of your network, or share files with your users.

1.1. Network interfaces

Network Interface Cards are the hardware adapters that provide one or more Ethernet ports for network connectivity, and they are abbreviated as NICs. They may also be referred to as network adapters and individual ports as network interfaces or simply interfaces, two or more interfaces can be configured to provide bonding and teaming for redundancy and faster throughput.

1.2. Working on network configuration files

As a Linux server administrator, you need to manage network addresses and network interfaces, the network addresses can be assigned in two ways:

- **Fixed IP addresses:** Useful for servers that always need to be available at the same IP address.
- **Dynamically assigned IP addresses:** Useful for end users devices, and for instances in a cloud environment, to dynamically assign IP addresses, a Dynamic Host Configuration Protocol (DHCP) server is usually used.

Configuration files for all interfaces are stored at a central location, which is the /etc/sysconfig/network-scripts directory.

The /etc/sysconfig/network-scripts directory

This directory contains important files that deal with networking, where each network interface has its own config which stores IP assignments and other relevant parameters for the interface, and the system reads this file and applies the settings at the time the interface is activated.

```
root@server1:~# ls /etc/sysconfig/network-scripts/
ifcfg-ens33     ifdown-bnep     ifdown-ipv6     ifdown-ppp
ifdown-Team        ifup             ifup-eth    ifup-isdn
ifup-post     ifup-sit        ifup-tunnel        network-
functions
ifcfg-lo        ifdown-eth     ifdown-isdn   ifdown-routes
ifdown-TeamPort    ifup-aliases    ifup-ippp    ifup-plip
ifup-ppp      ifup-Team       ifup-wireless      network-
functions-ipv6
ifdown             ifdown-ippp   ifdown-post    ifdown-sit
ifdown-tunnel      ifup-bnep       ifup-ipv6   ifup-plusb
ifup-routes  ifup-TeamPort  init.ipv6-global
root@server1:~#
```

Every connection that you create is stored as interface file, names begin with ifcfg- and is followed by the name of the interface by which the system recognizes it:

```
root@server1:~# cat /etc/sysconfig/network-scripts/ifcfg-
ens33
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="dhcp"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="57552a43-79b4-44c4-bd13-95df76edb304"
DEVICE="ens33"
ONBOOT="yes"
root@server1:~#
```

There are many important elements to note in this file:

- The TYPE option determines the interface type and here it's Ethernet
- The DEVICE option, which specifies which interface you are working with. Here, the ens33 interface is specified
- This interface uses the DHCP protocol defined using the BOOTPROTO option, which means that the IP address is leased from a DHCP server on the network (automatically during boot)
- Note: its value will be “static” in the case of fixed IP addresses
- ONBOOT: this option tells the NIC to start up when the system starts automatically

Normally, there should be no need to modify these configuration files manually, but after making changes to the configuration file, use the nmcli con reload command to activate the new configuration, see 1.7 Configuring the Network with nmcli.

1.3. Validating network address configuration

To verify the configuration of the network address, you need to use the ip utility. The ip utility is a modern utility that can consider advanced networking features that have been introduced recently, with it many aspects of networking can be monitored:

- Use ip addr to configure and monitor network addresses
- Use ip route to configure and monitor routing information
- Use ip link to configure and monitor network link state

To show current network settings, you can use the ip addr show command (which can be abbreviated as ip a s or even as ip a). The result of the ip addr show command looks as in Listing:

```
[root@server1:~]# ip a
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 :ens33:   <BROADCAST,MULTICAST,UP,LOWER_UP>   mtu   1500
qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:5a:82:8f brd ff:ff:ff:ff:ff:ff
    inet  192.168.114.148/24  brd  192.168.114.255  scope
global noprefixroute dynamic ens33
       valid_lft 1675sec preferred_lft 1675sec
    inet6    fe80::e926:49bc:1dc9:2b55/64    scope    link
noprefixroute
       valid_lft forever preferred_lft forever
```

Beside the LOOPBACK status that appeared in the command on the first block of info, the command shows the following items about its current status:

- **Current state:** The line contains the NIC name ens33 in our case you can see the text state UP, which shows that this network card is currently up and available
- **MAC address configuration:** This is the unique MAC address that is set for every network card. You can see the MAC address itself (00:0c:29:5a:82:8f), as well as the corresponding broadcast address
- **IPv4 configuration:** The line contains the inet phrase, it shows the IP address that is currently set (192.168.114.148), as well as the subnet mask that is used. You can also see the broadcast address that is used for this network configuration
- **IPv6 configuration:** The line contains the inet6 phrase, this line shows the current IPv6 address and its configuration

If you are just interested in the link state of the network interfaces, you can use the ip link show command, this repeats the link state information of the ip addr show command.

```
[root@server1:~]# ip link show
1 :lo:  <LOOPBACK,UP,LOWER_UP>  mtu  65536  qdisc  noqueue
state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2 :ens33:   <BROADCAST,MULTICAST,UP,LOWER_UP>   mtu   1500
qdisc pfifo_fast state UP mode DEFAULT group default qlen
1000
    link/ether 00:0c:29:5a:82:8f brd ff:ff:ff:ff:ff:ff
```

In case the ip link show command shows the current link state as down, you can temporarily bring it up again by using ip link set dev devicename up (for example, ip link set dev eno16777736 up).

Exercise: Validating network configuration

1. Open a root shell.
2. Type ip –s link. This shows all existing network connections, in addition to statistics about the number of packets that have been sent and associated error messages.
3. Type ip addr show. You'll see the current address assignments for network interfaces on your server.

1.3.1. Validating routing

One important aspect of networking is routing, on every network that needs to communicate with nodes on other networks, routing is a requirement. Every network has, at least, a default router (also called the default gateway) that is set, and you can see which router is used as the default router by using the command ip route show. You should always perform one quick check to verify that your router is set correctly, where the default router at all times must be on the same network as the local IP address that your network card is using.

```
[root@server1:~]# ip route show
default via 192.168.114.2 dev ens33 proto dhcp metric 100
192.168.114.0/24  dev  ens33  proto  kernel  scope  link  src
192.168.114.148 metric 100
```

1.3.2. Validating the availability of ports and services

Network problems can be related to the local IP and router settings but can also be related to network ports that are not available on your server or on a remote server. To verify availability of ports on your server, you can use the netstat command, or the newer ss command, which provides the same functionality. Example shows how to verify network settings, by typing ss -lt, you'll see all listening TCP ports on the local system:

```
[root@server1 ~]# ss -lt
State Recv-Q Send-Q Local Address:Port Peer Address:Port
LISTEN 0 100 127.0.0.1:smtp*:*
LISTEN 0 128 *:56601*:*
LISTEN 0 128 .0.0.1:x11-ssh-offse*:*
LISTEN 0 128 *:sunrpc*:*
LISTEN 0 128 *:ssh*:*
LISTEN 0 128 127.0.0.1:ipp*:*
LISTEN 0 100 ::1:smtp*=:::
LISTEN 0 128 :1:x11-ssh-offset* :::
LISTEN 0 128 :::sunrpc*:::
LISTEN 0 128 :::34449*:::
LISTEN 0 128 :::ssh*:::
LISTEN 0 128 ::1:ipp :::*
```

Notice where the port is listening on, some ports are only listening on the IPv4 loopback address 127.0.0.1 or the IPv6 loopback address::1, which means that they are locally accessible only. Other ports are listening on *, which stands for all IPv4 addresses, or on:::*, which represents all ports on all IPv6 addresses.

Exercise: Verifying network settings

1. Open a root shell to your server and type ip addr show. This shows the current network configuration. Note the IPv4 address that is used. Notice the network device names that are used; you need these later in this exercise.

2. Type ip route show to verify routing configuration.

3. If your computer is connected to the Internet, you can now use the ping command to verify the connection to the Internet is working properly. Type ping –c 4 8.8.8.8, for instance, to send four packets to IP address 8.8.8.8. If your Internet connection is up and running, you should get "echo reply" answers.

4. Type ip addr add 10.0.0.10/24 dev <yourdevicename>.

5. Type ip addr show. You'll see the newly set IP address, in addition to the IP address that was already in use.

6. Type ifconfig. Notice that you do not see the newly set IP address (and there are no options with the ifconfig command that allow you to see it). This is one example why you should not use the ifconfig command anymore.

7. Type ss –tul. You'll now see a list of all UDP and TCP ports that are listening on your server.

1.4. Configuring network

Networking on RHEL7 is managed by the NetworkManager service, you can use the systemctl status NetworkManager command to verify its current status. When NetworkManager comes up, it reads the network card configuration scripts, which are in /etc/sysconfig/network-scripts and have a name that starts with ifcfg and is followed by the name of the network card.

When working with network configuration, you should know the difference between a device and a connection:

- A device is a network interface card
- A connection is the configuration that is used on a device

In RHEL7, you can create multiple connections for a device, this can make sense on mobile computers, to make a difference between settings that are used while connected to the home network and settings that are needed to the corporate network.

Switching between connections on devices is something that is common on end-user computers, and not so common on servers. To manage the network connections that you want to assign to devices, you use the nmtui or the nmcli command.

1.4.1. Configuring the network with nmcli

Everything you do with the ip command is none persistent, though, if you want to make your configuration persistent, use nmtui or nmcli. A good start is to use nmcli to show all connections, this shows active and inactive connections, you can easily see the difference because inactive connections are not currently assigned to a device.

```
[root@server1:~]# nmcli con show
NAME    UUID                                  TYPE
DEVICE
ens33    57552a43-79b4-44c4-bd13-95df76edb304    ethernet
ens33
```

After finding the name of the connection, you can use nmcli con show followed by the name of the connection to see all properties of the connection:

```
[root@server1:~]# nmcli con show ens33
connection.id:                          ens33
connection.uuid:                            57552a43-79b4-
44c4-bd13-95df76edb304
connection.stable-id--                    :
connection.type:                        802-3-ethernet
connection.interface-name:              ens33
connection.autoconnect:                 yes
...
ipv4.method:                            auto
ipv4.dns--                                :
ipv4.dns-search--                         :
ipv4.dns-options""                        :
ipv4.dns-priority:                      0
ipv4.addresses--                          :
ipv4.gateway--                            :
ipv4.routes--                             :
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules--                      :
ipv4.ignore-auto-routes:                no
```

You can also use nmcli to show an overview of currently configured devices and the status of these devices. Type, for instance, the nmcli dev status command to show a list of all devices:

```
[root@server1:~]# nmcli dev status
DEVICE TYPE      STATE      CONNECTION
ens33   ethernet connected ens33
lo      loopback unmanaged --
```

And nmcli dev show <devicename> to show settings for a specific device:

```
[root@server1: ~]# nmcli dev show ens33
GENERAL.DEVICE:                         ens33
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:0C:29:5A:82:8F
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens33
GENERAL.CON-PATH:
/org/freedesktop/NetworkManager/ActiveConnection/1
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:
192.168.114.148/24
IP4.GATEWAY:                            192.168.114.2
IP4.ROUTE[1]:                             dst = 0.0.0.0/0,
nh = 192.168.114.2, mt = 100
IP4.ROUTE[2]:                                          dst  =
192.168.114.0/24, nh = 0.0.0.0, mt = 100
IP4.DNS[1]:                             192.168.114.2
IP4.DOMAIN[1]:                          localdomain
IP6.ADDRESS[1]:
fe80::e926:49bc:1dc9:2b55/64
IP6.GATEWAY--                            :
IP6.ROUTE[1]:                             dst = fe80::/64,
nh = ::, mt = 100
IP6.ROUTE[2]:                              dst = ff00::/8,
nh = ::, mt = 256, table=255
```

Exercise: Managing network connections with nmcli

1. Create a new network connection using nmcli con add con-name “dhcp” type ethernet ifname eth0
2. Create a connection with the name static to define a static IP address and gateway: nmcli con add con-name “static” ifname eth0 autoconnect no type ethernet ip4 10.0.0.10/24 gw4 10.0.0.1. The gateway might not exist in your configuration, but that does not matter. (Make sure to change the ifname eth0 into the interface name that matches your hardware!)
3. Type nmcli con show to show the connections, and use nmcli con up “static” to activate the static connection. Switch back to the DHCP connection using nmcli con up “dhcp”

Exercise: Changing connection parameters with nmcli

In this exercise, you created network connections using nmcli con add. You can also change current connection properties by using nmcli con mod.

1. Make sure that the static connection does not connect automatically by using nmcli con mod “static” connection autoconnect no.

2. Add a DHCP server to the static connection by using nmcli con mod “static” ipv4.dns 10.0.0.10. Notice that while adding a network connection you used ip4, but while modifying parameters for an existing connection, you’ll often use ipv4 instead. This is not a typo, it is just an inconsistency in the command.

3. To add a second item for the same parameters, use a + sign. Test this by adding a second DNS server, using nmcli con mod “static” + ipv4.dns 8.8.8.8.

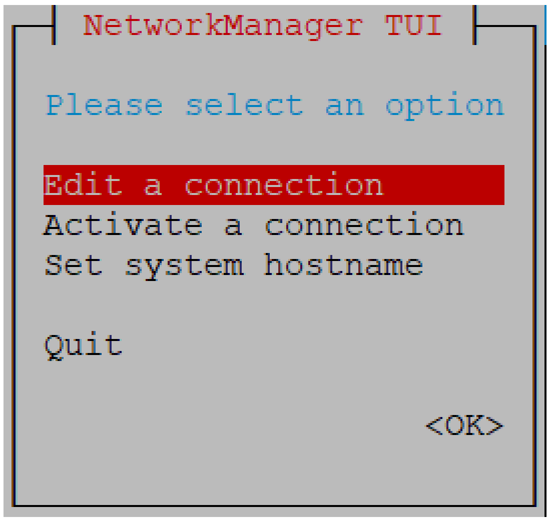
4. Using nmcli con mod, you can also change parameters such as the existing IP address. Try this by using nmcli con mod “static” ipv4.addresses “10.0.0.20/24” 10.0.0.100.

5. And to add a second IP address you use the + sign again: nmcli con mod “static” +ipv4.addresses 10.20.30.40/16.

6. After changing connection properties, you need to activate them. To do that, you can use nmcli con up “static”.

1.4.2. Configuring the network with nmtui

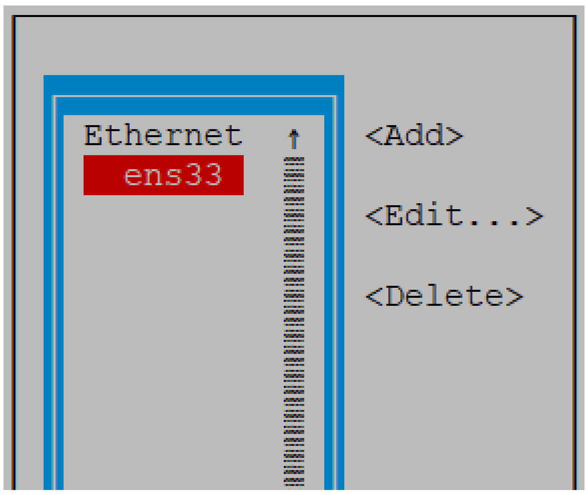
If you do not like the complicated syntax of the nmcli command line, you might like nmtui. This is a text user interface that allows you to create network connections easily, it consists of three menu options:

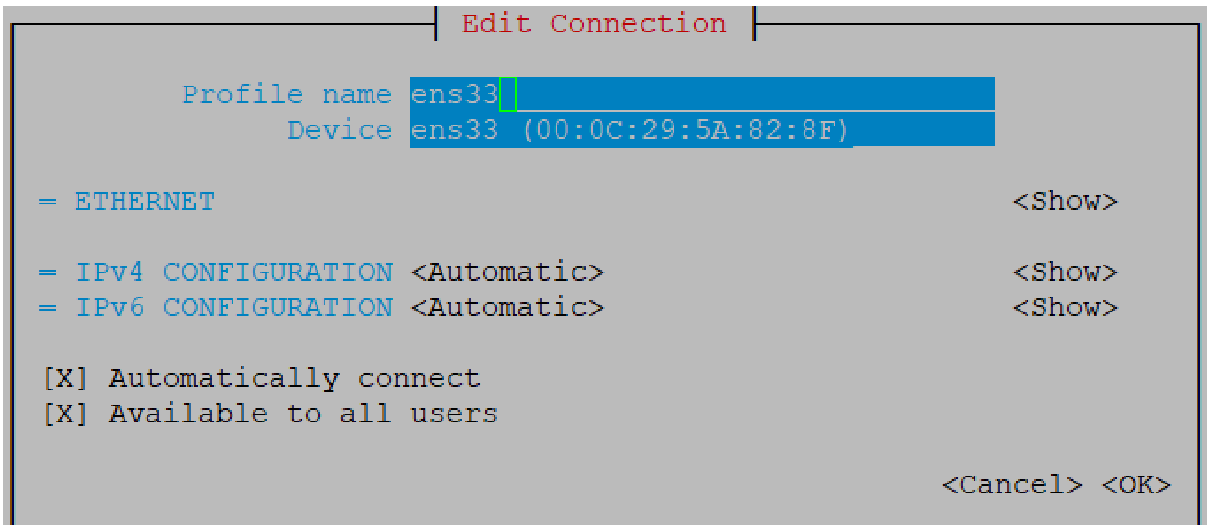


- **Edit a Connection:** Use this option to create new connections or edit existing connections
- **Activate a Connection:** Use this to (re) activate a connection
- **Set System Hostname:** Use this to set the hostname of your computer

The option to edit a connection offers almost all the features that you might ever need to do while working on network connections. You can use it to add any type of connection, not just Ethernet connections, but also advanced connection types such as network bridges and teamed network drivers are supported.

When you select the option: Edit Connection, you get access to a rich interface that allows you to edit most properties of network connections.





```
                    Edit Connection

        Profile name ens33
              Device ens33 (00:0C:29:5A:82:8F)

  ═ ETHERNET                                          <Show>

  ═ IPv4 CONFIGURATION <Automatic>                     <Show>
  ═ IPv6 CONFIGURATION <Automatic>                     <Show>

  [X] Automatically connect
  [X] Available to all users

                                               <Cancel> <OK>
```

After editing the connection, you need to deactivate it and activate it again, this should work automatically, but the fact is it does not.

1.5. IPv6 addresses

An IPv6 address may look like fe80::225:90ff: fe23:8998. In an IPv6 address, there may be components that have leading 0s. Multiple sequences of 0000 can be written as::, as you can see in this IPv6 address example. By using IPv6 addresses, billions of nodes can be addressed, which is why IPv6 addresses are slowly being introduced in current network configurations.

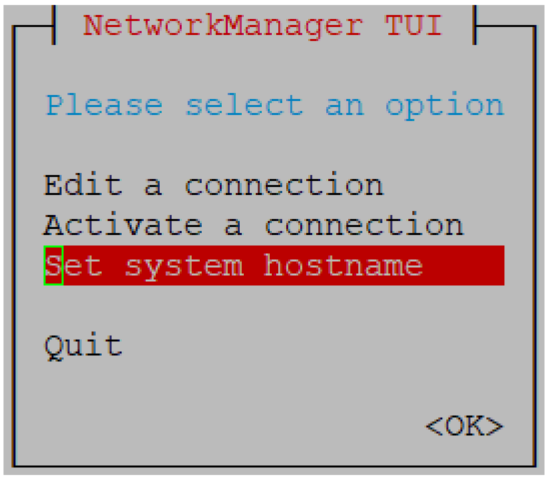
2. Hostname and name resolution

As an administrator, it is important that you know how to set the hostname, you also need to make sure that hosts can contact one another based on hostnames by setting up hostname resolution.

2.1. Hostnames

Because hostnames are used to access servers and the services they're offering, it is important to know how to set the system hostname. A hostname typically consists of different parts, these are the name of the host and the DNS domain in which the host resides. These two parts make up for the Fully Qualified Domain Name (FQDN), which looks like server1.localdomain. There are different ways to change the hostname:

2.1.1. Use nmtui tool



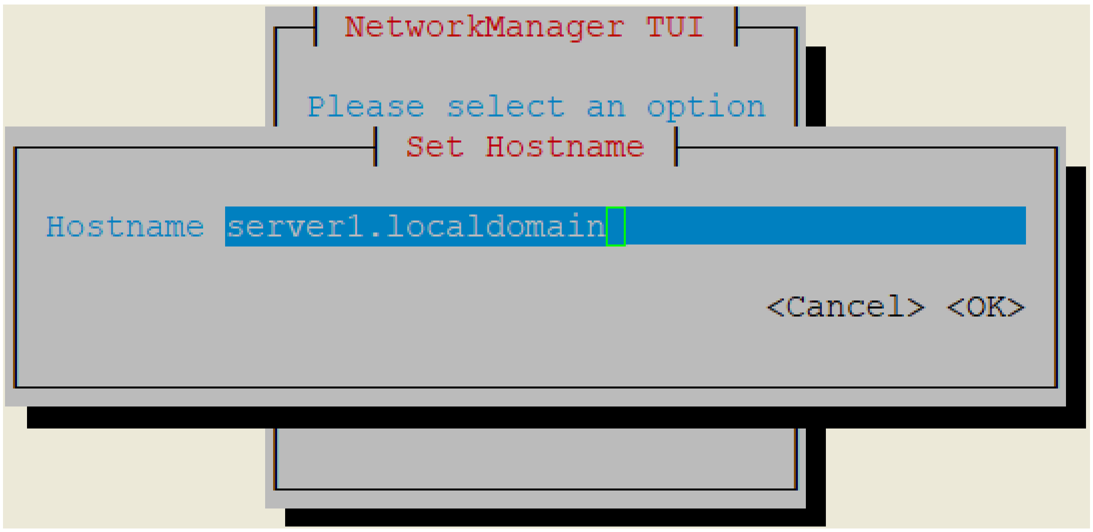
```
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

Quit

                         <OK>
```



```
NetworkManager TUI

Please select an option
Set Hostname

Hostname server1.localdomain

                         <Cancel> <OK>
```

2.1.2. Use hostnamectl

To configure the hostname with hostnamectl, you can use a command like hostnamectl set-hostname myhost.example.com, after setting the hostname, you can use hostnamectl status to show the current hostname, this shows not only information about the hostname but also information about the Linux kernel, virtualization type, and much more.

```
root@server1:~# hostnamectl
   Static hostname: server1.localdomain
         Icon name: computer-vm
           Chassis: vm
        Machine ID: d408f7cf3bf34049ad45e6a698787781
           Boot ID: 206322be2538420fa6675b4329ad853c
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-1062.4.3.el7.x86_64
      Architecture: x86-64
root@server1:~#          hostnamectl          set-hostname
server1.localdomain.com
```

2.1.3. use the /etc/hostname file

Applying the commands above actually affecting the content of the /etc/hostname, so editing the contents of this file will change the hostname on the system.

```
[root@server1:~]# cat /etc/hostname
localserver.localdomain.com
```

2.2. DNS resolving

Instead of working with IP addresses, you can use hostnames (the name of the system. The lookup and translation of an IP address to a hostname happen automatically via a DNS server on your network, just using an /etc/hosts file is not enough for name resolution if you want to be able to communicate with other hosts on the Internet, you should use DNS, too.

It is recommended to always set up at least two DNS name servers to be contacted, if the first name server does not answer, the second name server is contacted. To specify which DNS name servers you want to use, you have a few different options:

- Use nmtui to set the DNS name servers
- Set the DNS1 and DNS2 in the ifcfg network connection configuration file in /etc/sysconfig/network - scripts
- Use nmcli con mod <connection-id> [+]ipv4.dns <ip-of-dns>

Notice that if your computer is configured to get the network configuration from a DHCP server, the DNS server is also set via the DHCP server. If you do not want this to happen, you have two options:

- Edit the ifcfg configuration file to include the option PEER DNS=no
- Use nmcli con mod <con-name> ipv4.ignore-auto-dns yes

To verify host name resolution, you can use the getent hosts <servername> command, this searches in both /etc/hosts and DNS to resolve the hostname that has been specified.

```
[root@server1:~]# getent hosts server1
67.227.226.241  localdomain.com server1.localdomain.com
[root@server1:~]#
```

2.2.1. The /etc/hosts file

To set host name resolution, DNS is typically used. Apart from DNS, you can configure host name resolution in the /etc/hosts file, all hostname-IP address definitions as set in /etc/hosts will be applied before the hostname in DNS is used.

Setting up /etc/hosts file is easy; just make sure that it contains rows of the syntax: <ip> <fqdn> <alias>.

- The first column <ip> has the IP address of the specific host
- The second column <fqdn> specifies the hostname, where the hostname can be provided as a short name (like server1), or an FQDN
- The third column <alias>, is optional and is used when a host has more than one name, like a short name and a fully qualified DNS name. In that case, the second column must contain the FQDN, and the third column can contain the alias, listing below shows a hostname configuration example:

```
[root@server1 ~]# cat /etc/hosts
127.0.0.1 localhost    localhost.localdomain    localhost4
localhost4.localdomain4
::1      localhost      localhost.localdomain     localhost6
localhost6.localdomain6
```

By default, there is nothing in this hosts file other than a localhost entry (localhost is the local loopback). When you run a small environment, using host files is a feasible task, and you would need to make entries in the hosts file on every server (remember the hosts file is local to the server). The logical solution for large number of servers is to let the system query a DNS server instead of the local hosts file.

2.2.2. The /etc/resolv.conf file

Contains the IP address of the DNS server, this allows your systems to rely on DNS queries instead of a local lookup in the hosts file, see the content of the /etc/resolv.conf file:

```
[root@server1 ~]# cat /etc/resolv.conf
 #Generated by NetworkManager
search localdomain
nameserver 192.168.114.2
```

This output shows that the server you are on (server1) sends all DNS queries to the DNS server 192.168.114.2, although you don't have a DNS server installed, this example shows how to troubleshoot incorrect settings on clients, you could also have a /etc/resolv.conf file that looks like this:

```
[root@server1 ~]# cat /etc/resolv.conf
search example.com
nameserver 192.168.1.1
nameserver 172.168.1.1
```

The search option is the name of the domain that will be searched by default.

2.2.3. The /etc/nsswitch.conf file

All right, so you have configured a hosts file and a resolv.conf file. Now, which one gets queried first? Naturally, there is a third file that contains a single line defining the search order. This is configured as a default in the hosts line in /etc/nsswitch.conf:

```
# cat /etc/nsswitch.conf | grep hosts
hosts: files dns
```

This nsswitch file is the default—and with good reason. Sometimes you want a specific system to look up a specific host. Instead of making a DNS entry where multiple systems would see it, you could define an entry in a system's local /etc/hosts file. Because you usually don't have anything in your hosts file, however, the system defaults to the DNS server next. Testing that name resolution can be done in one of two ways:

- **First,** you could use the ping command to ping a host by its hostname to see if it responds, ping server1.localdomain.com
- **Second**, you could use the nslookup command to query the IP address associated with a hostname

The dig command can be used to gain a larger look at IP addresses and hostnames, and a good methodology would be to follow the order presented here when troubleshooting:

- Is the hostname set correctly?
- In what order are the files queried?
- Are the settings for the hosts file and resolv conf file correct?

3. Network time protocol NTP

3.1. Understanding local time

When a Linux server boots, the hardware clock, also referred to as real–time clock, is read. This clock typically resides in the computer hardware. Generally, it is an integrated circuit on the system board that is completely independent of the current state of the operating system and keeps running even when the computer is shut down. From the hardware clock, the system gets its initial time setting.

System time is a time maintained by the operating system, once the system has booted, its clock is completely independent of the hardware clock. Therefore, when system time is changed, the new system time is not automatically synchronized with the hardware clock.

Local time is the actual time in the current time zone, so it is a good idea to use time from a more reliable source, and there are two available solutions:

1. hardware clock.
2. configure your server to use Network Time Protocol (NTP).

3.2. Using network time protocol

NTP is a method of maintaining system time that is provided through NTP servers on the Internet. It is an easy solution to provide an accurate time to servers, because most servers are connected to the Internet anyway.

Setting up a server to use NTP time on RHEL7 is easy if the server is already connected to the Internet. If this is the case, the /etc/chrony.conf file is configured with a standard list of NTP servers on the Internet that should be contacted. The only thing the administrator has to do is switch on NTP, by using timedatectl set-ntp.

3.3. Understanding NTP

To synchronize time, the Network Time Protocol is used, before getting into detail about how to set up time synchronization, it makes sense knowing a bit about how NTP works.

To synchronize time in large network environments, you need a reliable clock. On networks that are connected to the Internet, reliable time is fetched from Internet servers that are connected to an external clock, this external clock is also referred to as the reference clock. Different models of hardware clocks are available, of which atomic clocks are the most reliable time sources.

3.4. NTP roles

A role is a function that a system performs from an NTP standpoint, a system can be configured to assume one or more of the following roles:

1. **Primary NTP Server:** A primary NTP server gets time from one of the time sources mentioned above, and provides time to one or more secondary servers or clients, or both. It can also be configured to broadcast time to secondary servers and clients.

2. **Secondary NTP Server:** A secondary NTP server receives time from a primary server or directly from one of the time sources mentioned above. It can be used to provide time to a set of clients to offload the primary, or for redundancy. The presence of a secondary server on the network is optional, although highly recommended. A secondary NTP server can also be configured to broadcast time to clients and peers.

3. **NTP Peer:** An NTP peer provides time to an NTP server and receives time from it. All peers work at the same stratum level, and all of them are considered equally reliable. Both primary and secondary servers can be peers of each other.

4. **NTP Client:** An NTP client receives time from either a primary or a secondary server.

3.5. Configure NTP client

In order to establish the binding with a remote time server for time synchronization, install the ntp software on the system, along with the GUI tool: yum –y install ntp system-config-date.

By default, the NTP client functionality is pre-configured in the /etc/ntp.conf file, which is the primary configuration file for NTP in RHEL7. You simply need to enable and start the service, there are four default remote time servers listed in the /etc/ntp.conf file, and they can be viewed as follows:

```
 #grep ^server /etc/ntp.conf
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
```

After the installation of the software is complete, run the following pair of commands to enable and start the service:

```
# systemctl enable ntpd
ln        -s        '/usr/lib/systemd/system/ntpd.service'
'/etc/systemd/system/multi-
user.target.wants/ntpd.service'
# systemctl start ntpd
```

The NTP daemon is now started, execute the ntpq command with the –p switch to determine the association of your system with a timeserver.

Exercise: Use pre–defined NTP polling client

This exercise should be done on server2. By default, the NTP software comes pre–configured for use as an NTP client. The configuration file, /etc/ntp.conf, already has four public NTP server entries. In this exercise, you will activate the NTP service and check to ensure that it is functional.

1. Install the NTP software (if it is not already installed).
2. Ensure that the following public NTP server entries are defined in the /etc/ntp.conf file.
3. Enable the ntpd daemon to autostart at reboots.
4. Start the NTP service and check its status.
5. Check whether the system is bound to the NTP servers: The above output indicates that the ntpd daemon on server2 is currently bound to an NTP server ntp3.torix.ca. Details for each column in this output are provided later in this chapter.
6. Check the status of NTP client.

3.6. Configure NTP server and polling client

We will set up server1 as an NTP server and sync time to its local clock and provide time to clients on the network. Then we will:

1. Install the NTP software on server1:

```
# yum –y install ntp
```

2. Comment out all server directives from the /etc/ntp.conf file and add a new one with IP 127.127.1.0:

```
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server 127.127.1.0
```

3. Enable the NTP service to autostart at reboots:

```
# systemctl enable ntpd
ln -s '/usr/lib/systemd/system/ntpd.service' '/etc/systemd/system/multi-user.target.wants/ntpd.service'
```

4. Open UDP port 123 in the firewall persistently and load the new rule:

```
# firewall-cmd --permanent --add-service ntp
# firewall-cmd --reload
```

5. Start the ntpd service and check its status

6. Disable the server directives in the /etc/ntp.conf file on server2 and add the following to use server1 as the timeserver:

```
#server 0.rhel.pool.ntp.org iburst
#server 1.rhel.pool.ntp.org iburst
#server 2.rhel.pool.ntp.org iburst
#server 3.rhel.pool.ntp.org iburst
server server1.example.com
```

7. Restart ntpd on server2 and check the status of binding with the ntpq command.

Quiz

Chapter review questions

1. The NTP client is pre–configured when the ntp software is installed on the system. We just need to start the service to begin synchronizing the clock.

   a. True

   b. False

2. Which command may we use to display the hardware address of a network interface?

   a. ip

   b. ip link show

   c. ipconfig

3. In which directory does the Network Manager store the interface configuration files?

   a. /etc/sysconfig/network

   b. /etc/network

   c. /etc/sysconfig/network–scripts

4. What is the purpose of the ONBOOT directive in the interface configuration file?

   a. activate this interface on boot

   b. add the interface to grub

   c. configure interface to use dhcp on boot

5. The /etc/hosts file maintains the hostname to hardware address mappings.

   a. True

   b. False

6. What would the ip addr command produce?

   a. information about interfaces including IP and hardware address
   b. ip address information
   c. number of interfaces

7. Which of the following would be the default network interface name on a RHEL7 system?

   a. p6p1
   b. ens33
   c. eno1677783
   d. e0

8. What protocol and port does NTP use?

   a. UDP port 321
   b. UDP port 123
   c. TCP port 123

9. NTP server can be configured to broadcast time on the network

   a. True
   b. False

10. The difference between an NTP peer and an NTP client.

   a. NTP peer synchronize time while an NTP client generate time
   b. NTP client provides time while an NTP peer obtain time from it
   c. NTP peer provides time while an NTP client obtain time from it

11. Which configuration file contains the default list of NTP servers that should be contacted on RHEL7?

   a. /etc/ntp/ntp.conf
   b. /etc/ntp.conf
   c. /etc/chrony/chrony.conf
   d. /etc/chrony.conf

12. Which of the following is not a private IP address?

a. 10.10.10.10
b. 169.254.11.23
c. 172.19.18.17
d. 192.168.192.192

13. Which command shows the recommended way to display information about the network interface as well as its IP configuration?

a. ifconfig –all
b. ipconfig
c. ip link show
d. ip addr show

14. Which statement about NetworkManager is not true?

a. It is safe to disable NetworkManager and work with the network service instead
b. NetworkManager manages network connections that are applied to network interfaces
c. NetworkManager has a text based user interface with the name nmtui
d. NetworkManager is the default service to manage networking in RHEL7

15. Which of the following is not a recommended way to specify which DNS servers to use?

a. Edit /etc/resolv.conf
b. Set the DNS options in /etc/sysconfig/network–scripts/ifcfg–<ID>
c. Set the DNS server names using nmcli
d. Use nmtui to set the DNS server names

16. In which configuration file would you set the hostname?

a. /etc/sysconfig/networking
b. /etc/sysconfig/hostname
c. /etc/hostname
d. /etc/defaults/hostname

17. Which service manages networking in RHEL7?

a. networking
b. NetworkManager
c. ipnet

18. Which command manages networking in RHEL7?

a. nmcli con reload
b. service net restart
c. systemctl network start

19. Which service is used as the default service to synchronize time on RHEL7?

a. timedatectl
b. ntpd
c. hwclock
d. chronyd

20. Which command shows current routing configuration?

a. nmcli
b. ifconfig
c. ip route show

Answers to chapter Review Questions:

1. a
2. b
3. c
4. a
5. b
6. a
7. b
8. b
9. a
10. c
11. d
12. c
13. d
14. a
15. b
16. c
17. b
18. a
19. b
20. c