MANAGING NETWORKING

Objectives

- Manage network settings and devices using nmcli.
- Modify network settings by editing configuration files.
- Configure a server's static host name and its name resolution, and test the results.

CONFIGURING NETWORKING FROM THE COMMAND LINE

1. DESCRIBING NETWORK CONCEPTS

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

- A device is a network interface.
- A connection is a collection of settings that can be configured for a device.
- Only one connection can be active for any one device at a time. Multiple connections may exist for use by different devices or to allow a configuration to be altered for the same device. If you need to temporarily change networking settings, instead of changing the configuration of a connection, you can change which connection is active for a device. For example, a device for a wireless network interface on a laptop might use different connections for the wireless network at a work site and for the wireless network at home.
- Each connection has a name or ID that identifies it.
- The nmcli utility is used to create and edit connection files from the command line.

2. VIEWING NETWORK INFORMATION

The nmcli dev status command displays the status of all network devices:

```
[user@host ~]$ nmcli dev status

DEVICE   TYPE       STATE          CONNECTION

eno1     ethernet   connected      eno1

ens3     ethernet   connected      static-ens3

eno2     ethernet   disconnected   --

lo       loopback   unmanaged      --
```

The **nmcli con** show command displays a list of all connections. To list only the active connections, add the --active option.

```
[user@host ~]$ nmcli con show
```

```
NAME           UUID                                  TYPE            DEVICE
eno2           ff9f7d69-db83-4fed-9f32-939f8b5f81cd  802-3-ethernet  --
static-ens3    72ca57a2-f780-40da-b146-99f71c431e2b  802-3-ethernet  ens3
eno1           87b53c56-1f5d-4a29-a869-8a7bdaf56dfa  802-3-ethernet  eno1
```

[user@host ~]$ nmcli con show --active

```
NAME           UUID                                  TYPE            DEVICE
static-ens3    72ca57a2-f780-40da-b146-99f71c431e2b  802-3-ethernet  ens3
eno1           87b53c56-1f5d-4a29-a869-8a7bdaf56dfa  802-3-ethernet  eno1
```

3. ADDING A NETWORK CONNECTION

The **nmcli con add** command is used to add new network connections. The following example **nmcli con add** commands assume that the name of the network connection being added is not already in use.

The following command adds a new connection named eno2 for the interface eno2, which gets IPv4 networking information using DHCP and autoconnects on startup. It also gets IPv6 networking settings by listening for router advertisements on the local link. The name of the configuration file is based on the value of the con-name option, eno2, and is saved to the **/etc/sysconfig/network-scripts/ifcfg-eno2** file.

[root@host ~]# nmcli con add con-name eno2 type ethernet ifname eno2

The next example creates an eno2 connection for the eno2 device with a static IPv4 address, using the IPv4 address and network prefix 192.168.0.5/24 and default gateway 192.168.0.254, but still autoconnects at startup and saves its configuration into the same file. Due to screen size limitations, terminate the first line with a shell \ escape and complete the command on the next line.

**[root@host ~]# nmcli con add con-name eno2 type ethernet ifname eno2 **

```
> ip4 192.168.0.5/24 gw4 192.168.0.254
```

This final example creates an eno2 connection for the eno2 device with static IPv6 and IPv4 addresses, using the IPv6 address and network prefix 2001:db8:0:1::c000:207/64 and default IPv6 gateway 2001:db8:0:1::1, and the IPv4 address and network prefix 192.0.2.7/24 and default IPv4 gateway 192.0.2.1, but still autoconnects at startup and saves its configuration into /etc/sysconfig/network-scripts/ifcfg-eno2. Due to screen size limitations, terminate the first line with a shell \ escape and complete the command on the next line.

```
[root@host ~]# nmcli con add con-name eno2 type ethernet ifname eno2 \
> ip6 2001:db8:0:1::c000:207/64 gw6 2001:db8:0:1::1 ip4 192.0.2.7/24 gw4 192.0.2.1
```

4. CONTROLLING NETWORK CONNECTIONS

The **nmcli con up name** command activates the connection name on the network interface it is bound to. Note that the command takes the name of a connection, not the name of the network interface. Remember that the nmcli con show command displays the names of all available connections.

```
[root@host ~]# nmcli con up static-ens3
```

The nmcli dev disconnect device command disconnects the network interface device and brings it down. This command can be abbreviated nmcli dev dis device:

```
[root@host ~]# nmcli dev dis ens3
```

5. MODIFYING NETWORK CONNECTION SETTINGS

NetworkManager connections have two kinds of settings. There are static connection properties, configured by the administrator and stored in the configuration files in **/etc/sysconfig/network-scripts/ifcfg-*.** There may also be active connection data, which the connection gets from a DHCP server and which are not stored persistently.

To list the current settings for a connection, run the nmcli con show name command, where name is the name of the connection. Settings in lowercase are static properties that the administrator can change. Settings in all caps are active settings in temporary use for this instance of the connection.

```
[root@host ~]# nmcli con show static-ens3
connection.id:                          static-ens3
connection.uuid:                        87b53c56-1f5d-4a29-a869-8a7bdaf56dfa
connection.interface-name:              --
connection.type:                        802-3-ethernet
connection.autoconnect:                 yes
connection.timestamp:                   1401803453
connection.read-only:                   no
connection.permissions:
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.secondaries:
connection.gateway-ping-timeout:        0
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          yes
802-3-ethernet.mac-address:             CA:9D:E9:2A:CE:F0
802-3-ethernet.cloned-mac-address:      --
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:
ipv4.method:                            manual
ipv4.dns:                               192.168.0.254
ipv4.dns-search:                        example.com
ipv4.addresses:                         { ip = 192.168.0.2/24, gw = 192.168.0.254 }
```

The **nmcli con mod name** command is used to change the settings for a connection. These changes are also saved in the /etc/sysconfig/network-scripts/ifcfg-name file for the connection. Available settings are documented in the nm-settings(5) man page.

To set the IPv4 address to 192.0.2.2/24 and default gateway to 192.0.2.254 for the connection static-ens3:

```
[root@host ~]# nmcli con mod static-ens3 ipv4.addresses "192.0.2.2/24 192.0.2.254"
```

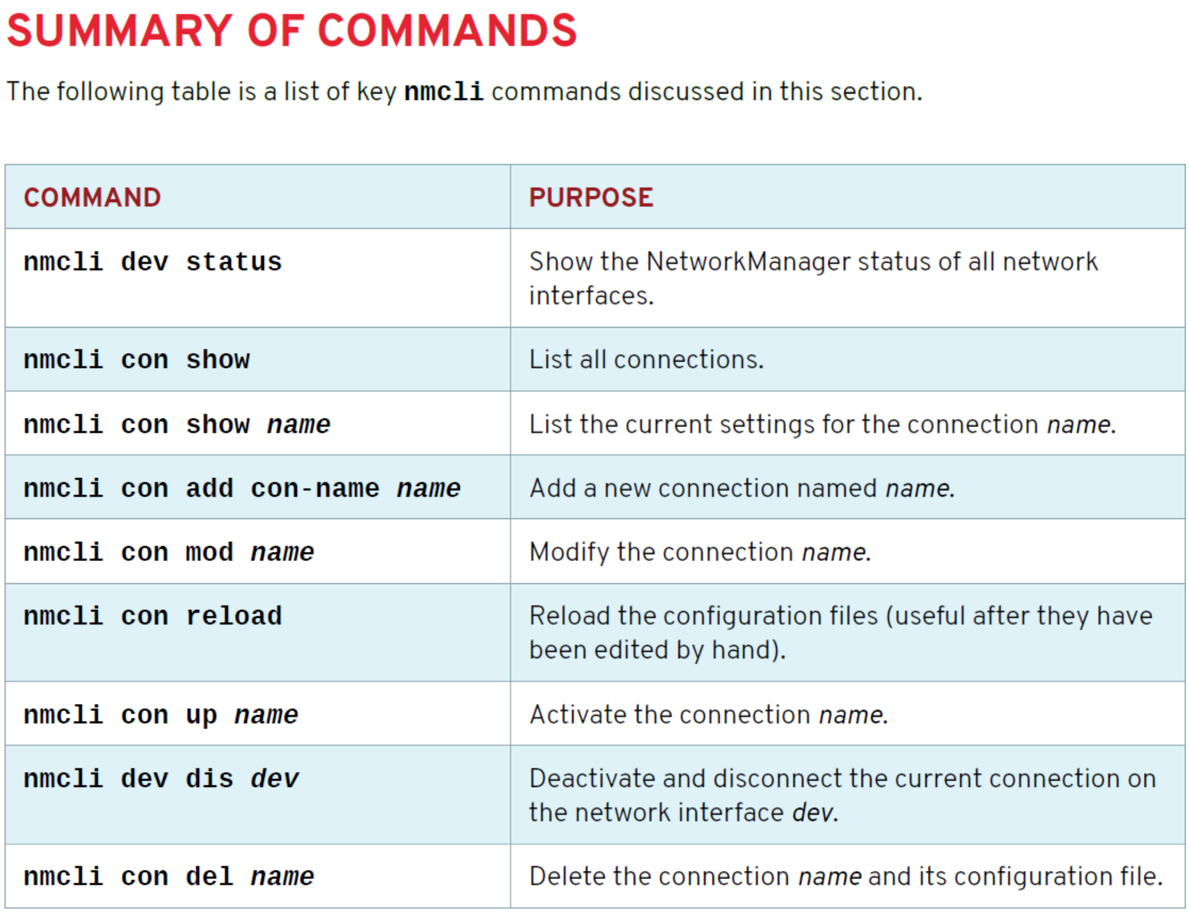
To set the IPv6 address to 2001:db8:0:1::a00:1/64 and default gateway to 2001:db8:0:1::1 for the connection static-ens3:

```
[root@host ~]# nmcli con mod static-ens3 ipv6.address "2001:db8:0:1::a00:1/64
2001:db8:0:1::1"
```

6. DELETING A NETWORK CONNECTION

The nmcli con del name command deletes the connection named name from the system, disconnecting it from the device and removing the file **/etc/sysconfig/network-scripts/ifcfg-name.**

```
[root@host ~]# nmcli con del static-ens3
```



SUMMARY OF COMMANDS

The following table is a list of key **`nmcli`** commands discussed in this section.

COMMAND	PURPOSE
`nmcli dev status`	Show the NetworkManager status of all network interfaces.
`nmcli con show`	List all connections.
`nmcli con show` ***`name`***	List the current settings for the connection *name*.
`nmcli con add con-name` ***`name`***	Add a new connection named *name*.
`nmcli con mod` ***`name`***	Modify the connection *name*.
`nmcli con reload`	Reload the configuration files (useful after they have been edited by hand).
`nmcli con up` ***`name`***	Activate the connection *name*.
`nmcli dev dis` ***`dev`***	Deactivate and disconnect the current connection on the network interface *dev*.
`nmcli con del` ***`name`***	Delete the connection *name* and its configuration file.

CONFIGURING HOST NAMES AND NAME RESOLUTION

1. CHANGING THE HOST NAME

The hostname command displays or temporarily modifies the system's fully qualified host name.

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

A static host name may be specified in the /etc/hostname file. The hostnamectl command is used to modify this file and may be used to view the status of the system's fully qualified host name. If this file does not exist, the host name is set by a reverse DNS query once the interface has an IP address assigned.

```
[root@host ~]# hostnamectl set-hostname host@example.com
[root@host ~]# hostnamectl status
Static hostname: host.example.com
Icon name: computer-vm
Chassis: vm
Machine ID: 73ab164e278e48be9bf80e80714a8cd5
Boot ID: 6b1cbc4177164ef58c0e9ed4adb2904f
Virtualization: kvm
Operating System: Red Hat Enterprise Linux 8.0 beta (Ootpa)
CPE OS Name: cpe:/o:redhat:enterprise_linux:8.0:beta
Kernel: Linux 4.18.0-60.el8.x86_64
Architecture: x86-64
[root@host ~]# cat /etc/hostname
host@example.com
```

2. CONFIGURING HOST NAME RESOLUTION

The stub resolver is used to convert host names to IP addresses or the reverse. It determines where to look based on the configuration of the **/etc/nsswitch.conf** file. By default, the contents of the **/etc/hosts** file are checked first.

```
[root@host ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
172.25.254.254 classroom.example.com
172.25.254.254 content.example.com
```

The **getent hosts hostname** command can be used to test host name resolution using the **/etc/hosts** file.

If an entry is not found in the **/etc/hosts** file, by default the stub resolver tries to look up the hostname by using a DNS nameserver. The **/etc/resolv.conf** file controls how this query is performed:

- search: a list of domain names to try with a short host name. Both this and domain should not be set in the same file; if they are, the last instance wins. See resolv.conf(5) for details.
- nameserver: the IP address of a nameserver to query. Up to three nameserver directives may be given to provide backups if one is down.

```
[root@host ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 172.25.254.254
```

NetworkManager updates the **/etc/resolv.conf** file using DNS settings in the connection configuration files. Use the nmcli to modify the connections.

```
[root@host ~]# nmcli con mod ID ipv4.dns IP
[root@host ~]# nmcli con down ID
[root@host ~]# nmcli con up ID
[root@host ~]# cat /etc/sysconfig/network-scripts/ifcfg-ID
...output omitted...
DNS1=8.8.8.8
...output omitted...
```

The default behavior of nmcli con mod ID ipv4.dns IP is to replace any previous DNS settings with the new IP list provided. A + or - symbol in front of the ipv4.dns argument adds or removes an individual entry.

```
[root@host ~]# nmcli con mod ID +ipv4.dns IP
```

To add the DNS server with IPv6 IP address 2001:4860:4860::8888 to the list of nameservers to use with the connection static-ens3:

```
[root@host ~]# nmcli con mod static-ens3 +ipv6.dns 2001:4860:4860::8888
```

Testing DNS Name Resolution

The host HOSTNAME command can be used to test DNS server connectivity.

```
[root@host ~]# host classroom.example.com
classroom.example.com has address 172.25.254.254
[root@host ~]# host 172.25.254.254
254.254.25.172.in-addr.arpa domain name pointer classroom.example.com.
```