



# Configuring Network Access on RHEL/CentOS 7.x Using nmcli

---

This appendix contains the following section:

- [Configuring Network Access on RHEL/CentOS 7.x Using nmcli, on page 1](#)

## Configuring Network Access on RHEL/CentOS 7.x Using nmcli

The **NetworkManager** command-line tool (**nmcli**) provides a command line way to configure networking by controlling NetworkManager. This section provides only an overview with some examples to help you learn how to use nmcli to configure network access on the virtual appliance.

In a departure from previous approaches to network interface configuration, NetworkManager deals with both connections and interfaces (also known as devices). Connections are configured with IP addresses, gateways, DNS servers, and then applied to interfaces (devices). This is a critical change from the past way of configuring network access on CentOS Linux.

There are two nmcli commands that are of general usefulness:

- The **nmcli d** command lists all available network interfaces (devices).
- The **nmcli c** command lists all available configurations.

Use the above two commands frequently as you learn to use nmcli.

Follow the steps below to configure an IP address for an interface on your virtual appliance. Typically, these commands are typed directly into the console of the virtual appliance. If you are already connected through the network (for example, by **ssh**), then making changes to the network interface configuration can be problematic, as you may also lose network connectivity (and thereby your ability to issue nmcli commands) at any point in the process.

---

### Step 1

Make sure that the interface does not block nmcli. The **nmcli d** command lists the existing interfaces. If the interface you want to configure is listed as **unmanaged**, then NetworkManager has been explicitly blocked from configuring this interface. Until you remove this blockage, no nmcli command will have any effect on this interface. Note that you may not need to perform this procedure unless the interface is listed as **unmanaged**. Follow the steps below to allow it to be managed by NetworkManager:

- a) Remove the line `NM_CONTROLLED=no` from the file `/etc/sysconfig/network-scripts/ifcfg-interface`, where *interface* is the interface name listed in the `nmcli d` command. If there is no file with this name, then you do not need to perform this procedure.
- b) Tell the NetworkManager to read the configuration files again by using the following command:

```
nmcli connection reload
```

**Note** Manual changes to any `ifcfg` file will not be noticed by NetworkManager until the `nmcli connection reload` command is issued.

**Step 2** Make sure that there is no current configuration for the interface that you want to configure. If you want the configuration that you create to be the default for the interface and there are multiple configurations associated with an interface, it may lead to confusion when the system reboots. The `nmcli c` command lists the existing configurations. If you see any existing configurations, examine them to see if they apply to the interface you want to configure. An easy way to do this is to use the following command:

```
nmcli con show config | grep interface
```

If you see any output, you should remove the configuration `config` using the following command:

```
nmcli con delete config
```

**Note** There is often a configuration called "Wired connection 1" which needs to be deleted.

**Step 3** Create the configuration and associate it with the interface (device) in one command. This command only creates the configuration and associates it with the interface, it does not apply it to the interface.

```
nmcli con add type ethernet con-name config ifname interface ip4 ip/netmaskwidth gw4 gateway
```

where *config* is the name of the configuration, which can be anything (including the name of the interface), *interface* is the name of the interface (device), *ip* is the IPv4 address, *netmaskwidth* is the network mask width, and *gateway* is the IPv4 gateway address.

For example (type all in one line):

```
nmcli con add type ethernet con-name my-office ifname ens160 ip4 10.10.24.25/24 gw4 10.10.20.174
```

**Step 4** Add the DNS server to the configuration for the interface (device):

```
nmcli con mod config ipv4.dns dnsip
```

where *dnsip* is the IPv4 address of the DNS server and *config* is the name of the configuration.

For example:

```
nmcli con mod my-office ipv4.dns 72.63.128.140
```

You can add two DNS addresses as given below:

```
nmcli con mod my-office ipv4.dns "72.63.128.140 72.63.111.120"
```

**Note** This will replace any previously set DNS servers. To add to an previously set DNS entry, use the `+` before `ipv4.dns` as shown below:

```
nmcli con mod test-lab +ipv4.dns "72.63.128.140 72.63.111.120"
```

**Step 5** Apply the configuration to the interface, which will bring up the interface if it was not already running:

```
nmcli con up config
```

where *config* is the name of the configuration.

**Step 6** Use the following command to examine information about a connection:

```
nmcli -p con show config
```

This will typically scroll off of the console screen, leaving the beginning unreadable. To allow you to move back and forth and examine the output easily, use the following command:

```
nmcli -p con show config | less
```

From this, you can see the entire configuration. You can modify things in the configuration using the following command:

```
nmcli con mod config something.other new-value
```

For example:

```
nmcli con mod my-office wifi-min.key-cntl wpa-psk
```

### Step 7

Use the **set-hostname** command to set the hostname for the system:

```
hostnamectl set-hostname hostname.domain
```

**Note** This must be done before registering the local to the regional. Otherwise, an error will result about "localhost" already existing.

where *hostname* is the hostname you want to use and *domain* is the domain name, ending with .com, .org, and so on. It is important to include the domain name (along with the .com, .org, or whatever ending is appropriate), since this is used as the default for DNS lookups.

For example:

```
hostnamectl set-hostname my-server.gooddomain.com
```

### Step 8

After you configure the networking, you **must** restart Cisco Prime Network Registrar for the interfaces to be properly discovered by Cisco Prime Network Registrar. Use the following commands to restart:

- For local cluster:

```
# systemctl restart nwreglocal
```

- For regional cluster:

```
# systemctl restart nwregregional
```

If you fail to restart, it will result in a misconfigured registration at the regional.

---

To develop a complete understanding of the usage of nmcli, search the Internet for online resources on nmcli and CentOS 7.7.

