Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 37 Agenda:
1. Manage Networking Using Ansible

Network Management in Linux:
NetworkManager is a service and set of tools designed specifically to make it easier to manage the networking configuration on Linux systems and is the default network management service on RHEL 9. In addition to a service that runs in the background, NetworkManager also includes the following tools:

nmcli – A tool for working with NetworkManager via the command line. This tool is useful when access to a graphical environment is unavailable and can also be used within scripts to make network configuration changes.

nmtui – A basic text-based user interface for managing NetworkManager. This tool can be run within any terminal window and allows changes to be made by making menu selections and entering data. While helpful in performing basic tasks, nmtui lacks many of the features provided by the nmcli tool.

nm-connection-editor – A complete graphical management tool providing access to most NetworkManager configuration options.

GNOME Settings – The Network screen of the GNOME desktop Settings application allows basic network management tasks to be performed.

Cockpit Network Settings – The Network screen of the Cockpit web interface allows a range of network management tasks to be performed.

Installing and Enabling NetworkManager:
NetworkManager should be installed by default for most RHEL 9 installations. Use the rpm command to find out if it needs to be installed:
 # rpm -q NetworkManager

If necessary, install the package as follows:
 # dnf install NetworkManager -y

Once the package is installed, the NetworkManager daemon will need to be enabled so that it starts each time the system boots:
 # systemctl enable NetworkManager

Finally, start the service running and check the status to verify that the launch was successful:
 # systemctl start NetworkManager
 # systemctl status NetworkManager

Creating Connection & Assigning IPv4 Address on the NIC:
Verify the available devices (NICs), connections and their allocated IP addresses.
 # nmcli dev status
 # nmcli connection show
 # ip a
 # ifconfig -a

Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

In RHEL 9, you can find the network connection file in the following location.

 # cd /etc/NetworkManager/system-connections/ # For RHEL 9 Only
 # ls -lh
 # cat ens160.nmconnections

To add a new connection, we would require a new device (NIC). This can be done by going in the settings of your virtual machine. And again, check the devices.

 # nmcli dev status
 # nmcli connection show
 # ip a
 # ifconfig -a

Add a new connection for the newly added NIC using nmcli command.

 # nmcli connection add type ethernet con-name ens224 ifname ens224 ipv4.addresses 192.168.229.145/24 ipv4.dns 8.8.8.8 gw4 192.168.229.128 ipv4.method manual connection.autoconnect yes

Verify the same using below commands.

 # ls -lh /etc/NetworkManager/system-connections/
 # cat /etc/NetworkManager/system-connections/ens224.nmconnection
 # ip a
 # nmcli d s
 # nmcli c s

To make changes in the available connections we can use the nmcli connection modify command.

 # nmcli connection modify ens224 connection.autoconnect no
OR
 # nmcli connection modify ens224 ipv4.addresses 192.168.229.149/24 ipv4.method manual
 # cat /etc/NetworkManager/system-connections/ens224.nmconnection
 # ip a
OR
 # ifconfig -a
 # ping 192.168.229.149
 # ping 192.168.229.145

Apply the changes by reloading the connection profile.

 # nmcli dev reapply ens224
 # ping 192.168.229.149
 # ping 192.168.229.145

OR you can bring down and start the connection again.

 # nmcli connection modify ens224 ipv4.addresses 192.168.229.145/24 ipv4.method manual
 # nmcli connection down ens224
 # ifconfig -a
 # nmcli connection up ens224
 # ifconfig -a
 # ping 192.168.229.149
 # ping 192.168.229.145

To bring down/up the interface (device), we can execute the below commands which require

Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

NetworkManager-initscripts packages.

 # ifdown ens224
 # nmcli dev status
 # nmcli connection show
 # ip a
 # ifconfig -a
 # ifup ens224
 # nmcli dev status
 # nmcli connection show
 # ip a
 # ifconfig -a

Apply the dynamic IP address using DHCP.

 # nmcli connection delete ens224
 # nmcli connection add type ethernet con-name ens224 ifname ens224 ipv4.method auto connection.autoconnect yes
 # nmcli device reapply ens224
 # ls -lh /etc/NetworkManager/system-connections/
 # cat /etc/NetworkManager/system-connections/ens224.nmconnection
 # ip a
 # nmcli d s
 # nmcli c s

Similarly, we can use nmtui tool for managing the interfaces as well as the connections.

 # nmtui

Similarly, we can use the GUI utility or the cockpit feature for managing the interfaces as well as the network connections in RHEL 9.

Manage Networking Using Ansible:

We can manage networking on the managed nodes by using:

1. Ansible Ad-Hoc Commands
2. Ansible Playbooks

1. Ansible Ad-Hoc Commands:

We can use command and nmcli modules to manage the networking in the managed nodes.

Command Module:

$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'

Add a new NIC on the managed node and verify the same.

$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'

Add a new connection and verify the same.

$ ansible node1 -m command -a 'nmcli connection add type ethernet con-name ens224 ifname ens224 ipv4.addresses 192.168.229.140/24 ipv4.dns 8.8.8.8 gw4 192.168.229.128 ipv4.method

manual connection.autoconnect yes' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.140

Bring down the connection and verify the same.
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'nmcli con down ens224' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.140

Bring up the connection and verify the same.
$ ansible node1 -m command -a 'nmcli con up ens224' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ping 192.168.229.140

Remove the connection.
$ ansible node1 -m command -a 'nmcli con delete ens224' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.140

NMCLI Module:
To use nmcli module, we have to install the community.general collection from the ansible galaxy.
$ ansible-galaxy collection install community.general
$ ansible node1 -m nmcli -a 'conn_name=ens224 ifname=ens224 type=ethernet ip4=192.168.229.150/24 gw4=192.168.229.128 state=present' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.150

Modify the existing connection.
$ ansible node1 -m nmcli -a 'conn_name=ens224 ifname=ens224 type=ethernet ip4=192.168.229.146/24 gw4=192.168.229.128 dns4=8.8.8.8 state=present autoconnect=true' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'

Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

$ ping 192.168.229.146
$ ansible node1 -m command -a 'nmcli dev reapply ens224' -b
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ping 192.168.229.146

Remove the existing connection.
$ ansible node1 -m nmcli -a 'conn_name=ens224 ifname=ens224 type=ethernet ip4=192.168.229.146/24 gw4=192.168.229.128 dns4=8.8.8.8 state=absent autoconnect=true' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.146

2. Ansible Playbooks:
We can use nmcli module in the ansible playbooks to manage the networking in the managed nodes.
$ vim nmcli.yml

```
---
- name: manage networking
  hosts: node1
  become: true
  tasks:
    - name: Add an Ethernet connection with static IP configuration
      community.general.nmcli:
        conn_name: ens224
        ifname: ens224
        type: ethernet
        ip4: 192.168.229.150/24
        gw4: 192.168.229.128
        dns4: 8.8.8.8
        state: present
        autoconnect: true
...
```

$ ansible-playbook nmcli.yml
$ ansible node1 -m command -a 'nmcli dev reapply ens224' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.150

$ vim nmcli2.yml

Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
---
- name: manage networking
  hosts: node1
  become: true
  tasks:
    - name: remove an Ethernet connection with static IP configuration
      community.general.nmcli:
        conn_name: ens224
        ifname: ens224
        type: ethernet
        ip4: 192.168.229.150/24
        gw4: 192.168.229.128
        dns4: 8.8.8.8
        state: absent
        autoconnect: true
...

$ ansible-playbook nmcli2.yml
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.150

$ vim nmcli3.yml
---
- name: manage networking
  hosts: node1
  become: true
  tasks:
    - name: Add an Ethernet connection with static IPv4, 2 DNSs & IPv6 configuration
      community.general.nmcli:
        conn_name: ens224
        ifname: ens224
        type: ethernet
        ip4: 192.168.229.150/24
        gw4: 192.168.229.128
        dns4:
          - 8.8.8.8
          - 192.168.1.1
        ip6: 2001:db8::cafe
        gw6: 2001:db8::1
        state: present
        autoconnect: true
...

$ ansible-playbook nmcli3.yml
$ ansible node1 -m command -a 'nmcli dev reapply ens224' -b
```

Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.150

$ vim nmcli4.yml
---
- name: manage networking
  hosts: node1
  become: true
  tasks:
    - name: Modify the existing connection for adding 2 additional IPv4 Address
      community.general.nmcli:
        conn_name: ens224
        ifname: ens224
        type: ethernet
        ip4:
          - 192.168.229.150/24
          - 192.168.229.144/24
          - 192.168.229.142/24
        gw4: 192.168.229.128
        dns4:
          - 8.8.8.8
          - 192.168.1.1
        state: present
        autoconnect: true
...

$ ansible-playbook nmcli4.yml
$ ansible node1 -m command -a 'nmcli dev reapply ens224' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.150
$ ping 192.168.229.144
$ ping 192.168.229.142

$ vim nmcli5.yml
---
- name: manage networking
  hosts: node1
  become: true
  tasks:
    - name: Delete the existing connection for adding 2 additional IPv4 Addresses
```

Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
    community.general.nmcli:
      conn_name: ens224
      ifname: ens224
      type: ethernet
      state: absent
...

$ ansible-playbook nmcli5.yml
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.150
$ ping 192.168.229.144
$ ping 192.168.229.142

$ vim nmcli6.yml
---
- name: manage networking
  hosts: node1
  become: true
  tasks:
    - name: Add an Ethernet connection with dynamic IPv4 (DHCP) configuration
      community.general.nmcli:
        conn_name: ens224
        ifname: ens224
        type: ethernet
        method4: auto
        state: present
        autoconnect: true
...

$ ansible-playbook nmcli6.yml
$ ansible node1 -m command -a 'nmcli dev reapply ens224' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
$ ping 192.168.229.137

$ ansible-playbook nmcli5.yml
$ vim nmcli7.yml
---
- name: manage networking
  hosts: node1
  become: true
```

Manage Networking Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
  tasks:
    - name: Add an Ethernet connection with dynamic IPv6 (DHCP) configuration
      community.general.nmcli:
        conn_name: ens224
        ifname: ens224
        type: ethernet
        method6: auto
        state: present
        autoconnect: true
...

$ ansible-playbook nmcli7.yml
$ ansible node1 -m command -a 'nmcli dev reapply ens224' -b
$ ansible node1 -m command -a 'ls -lh /etc/NetworkManager/system-connections/'
$ ansible node1 -m command -a 'cat /etc/NetworkManager/system-connections/ens224.nmconnection' -b
$ ansible node1 -m command -a 'ifconfig -a'
$ ansible node1 -m command -a 'nmcli d s'
$ ansible node1 -m command -a 'nmcli c s'
```

Thank You