



DHCP Server Configuration Using Ansible

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

Session - 39 Agenda:

1. DHCP Server Configuration Using Ansible

```
[vikasnehra@ansible-server ~]$ ping node1
```

```
PING node1.nehraclasses.local (192.168.229.129) 56(84) bytes of data.  
64 bytes from node1.nehraclasses.local (192.168.229.129): icmp_seq=1 ttl=64 time=1.61 ms  
64 bytes from node1.nehraclasses.local (192.168.229.129): icmp_seq=2 ttl=64 time=0.529 ms  
^C  
--- node1.nehraclasses.local ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1002ms  
rtt min/avg/max/mdev = 0.529/1.071/1.614/0.542 ms
```

```
[vikasnehra@ansible-server ~]$ vim dhcp-server.yml
```

```
---
```

```
- name: DHCP Server Configuration Playbook
  hosts: node1
  become: true
  tasks:
    - name: Setting up the static hostname in the machine.
      hostname:
        name: dhcp-server.nehraclasses.local
        use: systemd

    - name: Installing DHCP packages in the machine.
      dnf:
        name: dhcp*
        state: latest

    - name: Copying the configuration file into the /etc/dhcp directory using jinja2 template.
      template:
        src: dhcp.conf.j2
        dest: /etc/dhcp/dhcpd.conf
        force: true

    - name: Allowing DHCP traffic in the firewall.
      firewalld:
        service: dhcp
        zone: public
        permanent: true
        immediate: true
        state: enabled

    - name: Starting & enabling the DHCPD service.
      service:
        name: dhcpcd
        state: started
        enabled: yes
```

```
...
```



DHCP Server Configuration Using Ansible

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

```
[vikasnehra@ansible-server ~]$ vim dhcp.conf.j2
# dhcpd.conf
#
# Sample configuration file for ISC dhcpcd
#
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# Use this to enable / disable dynamic dns updates globally.
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 10.152.187.0 netmask 255.255.255.0 {
}
# This is a very basic subnet declaration.

subnet 10.254.239.0 netmask 255.255.255.224 {
    range 10.254.239.10 10.254.239.20;
    option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

subnet 10.254.239.32 netmask 255.255.255.224 {
    range dynamic-bootp 10.254.239.40 10.254.239.60;
    option broadcast-address 10.254.239.31;
    option routers rtr-239-32-1.example.org;
}

# A slightly different configuration for an internal subnet.
subnet 192.168.229.0 netmask 255.255.255.0 {
    range 192.168.229.110 192.168.229.113;
    # option domain-name-servers ns1.internal.example.org;
```



DHCP Server Configuration Using Ansible

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

```
# option domain-name "internal.example.org";
# option routers 10.5.5.1;
# option broadcast-address 10.5.5.31;
  default-lease-time 600;
  max-lease-time 7200;
}

# Hosts which require special configuration options can be listed in
# host statements. If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.
```

```
host passacaglia {
  hardware ethernet 0:0:c0:5d:bd:95;
  filename "vmunix.passacaglia";
  server-name "toccata.example.com";
}
```

```
# Fixed IP addresses can also be specified for hosts. These addresses
# should not also be listed as being available for dynamic assignment.
# Hosts for which fixed IP addresses have been specified can boot using
# BOOTP or DHCP. Hosts for which no fixed address is specified can only
# be booted with DHCP, unless there is an address range on the subnet
# to which a BOOTP client is connected which has the dynamic-bootp flag
# set.
```

```
host fantasia {
  hardware ethernet 08:00:07:26:c0:a5;
  fixed-address fantasia.example.com;
}
```

```
# You can declare a class of clients and then do address allocation
# based on that. The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.
```

```
class "foo" {
  match if substring(option vendor-class-identifier, 0, 4) = "SUNW";
}
```

```
shared-network 224-29 {
  subnet 10.17.224.0 netmask 255.255.255.0 {
    option routers rtr-224.example.org;
  }
  subnet 10.0.29.0 netmask 255.255.255.0 {
    option routers rtr-29.example.org;
  }
  pool {
    allow members of "foo";
    range 10.17.224.10 10.17.224.250;
  }
}
```



DHCP Server Configuration Using Ansible

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

```
pool {  
    deny members of "foo";  
    range 10.0.29.10 10.0.29.230;  
}  
}
```

```
[vikasnehra@ansible-server ~]$ ansible-galaxy collection install ansible.posix
```

```
[vikasnehra@ansible-server ~]$ ansible-playbook dhcp-server.yml
```

```
[vikasnehra@ansible-server ~]$ ansible node1 -m command -a 'sudo systemctl status dhcpcd'
```

Now, goto virtual network adapter setting in the VMWare and click on change settings to change disable the inbuild DHCP Server of VMWare.

Now, go to any of the virtual machines in the same subnet of your node machine (DHCP Server: node1 in our case) in the VMWare and add a network interface (NIC).

```
[vikasnehra@ansible-server ~]$ logout
```

```
[root@ansible-server ~]# nmcli d s  
DEVICE  TYPE      STATE      CONNECTION  
ens160  ethernet  connected  ens160  
ens224  ethernet  disconnected --  
lo     loopback  unmanaged  --
```

```
[root@ansible-server ~]# nmcli c s  
NAME      UUID           TYPE      DEVICE  
ens160   7e4ed711-1dd0-3299-b2e7-7de82b3280c1  ethernet  ens160
```

```
[root@ansible-server ~]# nmcli connection add type ethernet con-name ens224 ifname ens224  
ipv4.method auto connection.autoconnect true  
Connection 'ens224' (c5c31c84-73bb-47d2-ab0a-847661332299) successfully added.
```

```
[root@ansible-server ~]# ifconfig  
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.229.110 netmask 255.255.255.0 broadcast 192.168.229.255
```

The DHCP Server is working perfectly as expected.

Thank You