REVATURE

DNS and DHCP
Server Administration

P 1

Guided by -

Zakir Hussain

Created By-

Mainak Roy

Index :

Overview :

This project aims to set up and configure both DNS (Domain Name System) and DHCP (Dynamic Host Configuration Protocol) services on a CentOS Linux system. These two services work in tandem to ensure smooth network management, with DHCP dynamically assigning IP addresses and DNS resolving hostnames to IP addresses.

DNS Server Configuration:

- Install and configure a DNS server (using BIND) on CentOS.
- Set up a forward lookup zone to resolve domain names to IP addresses.
- Configure a reverse lookup zone to resolve IP addresses to domain names.
- Verify DNS functionality using tools like `nslookup` and `dig`.

DHCP Server Configuration:

- Install and configure a DHCP server (using ISC DHCP) to dynamically assign IP addresses to clients.
- Define the IP address range (pool) that the DHCP server will allocate.
- Set up custom options like defining the gateway, DNS servers, and lease time for clients.
- Ensure DHCP clients receive correct IP and network settings dynamically.

Implementation :

1. Setting Up a DNS Server:

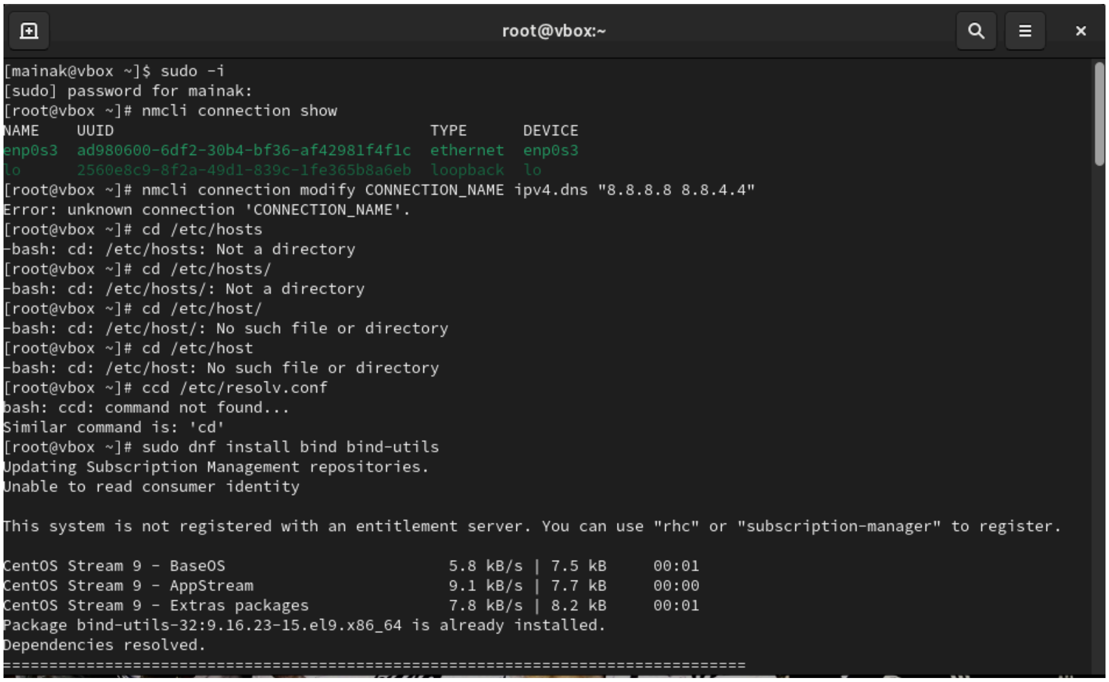
A DNS configuration project on CentOS 9 typically involves setting up a DNS server (like `BIND`) and configuring DNS resolution for the server and clients. Below is a step-by-step guide, along with the relevant commands, to set up a DNS server using `BIND` and client-side DNS configuration on CentOS 9.

Step 1. Installing BIND (DNS Server)

First, you'll need to install the `BIND` package on your CentOS 9 system. BIND (Berkeley Internet Name Domain) is the most widely used DNS server.

Syntax :

sudo dnf install bind bind-utils



```
[mainak@vbox ~]$ sudo -i
[sudo] password for mainak:
[root@vbox ~]# nmcli connection show
NAME    UUID                                  TYPE      DEVICE
enp0s3  ad980600-6df2-30b4-bf36-af42981f4f1c  ethernet  enp0s3
lo      2560e8c9-8f2a-49d1-839c-1fe365b8a6eb  loopback  lo
[root@vbox ~]# nmcli connection modify CONNECTION_NAME ipv4.dns "8.8.8.8 8.8.4.4"
Error: unknown connection 'CONNECTION_NAME'.
[root@vbox ~]# cd /etc/hosts
-bash: cd: /etc/hosts: Not a directory
[root@vbox ~]# cd /etc/hosts/
-bash: cd: /etc/hosts/: Not a directory
[root@vbox ~]# cd /etc/host/
-bash: cd: /etc/host/: No such file or directory
[root@vbox ~]# cd /etc/host
-bash: cd: /etc/host: No such file or directory
[root@vbox ~]# ccd /etc/resolv.conf
bash: ccd: command not found...
Similar command is: 'cd'
[root@vbox ~]# sudo dnf install bind bind-utils
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

CentOS Stream 9 - BaseOS                        5.8 kB/s | 7.5 kB     00:01
CentOS Stream 9 - AppStream                     9.1 kB/s | 7.7 kB     00:00
CentOS Stream 9 - Extras packages               7.8 kB/s | 8.2 kB     00:01
Package bind-utils-32:9.16.23-15.el9.x86_64 is already installed.
Dependencies resolved.
================================================================================
```

Step 2. Configuring BIND as a DNS Server

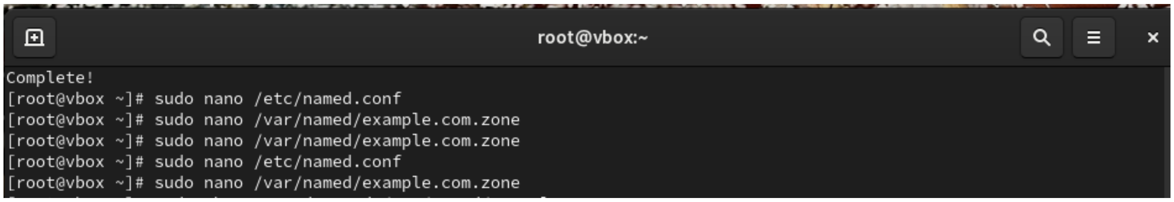
Once BIND is installed, the next step is to configure it.

[a] Editing the main configuration file-

The main configuration file for BIND is located at /etc/named.conf.

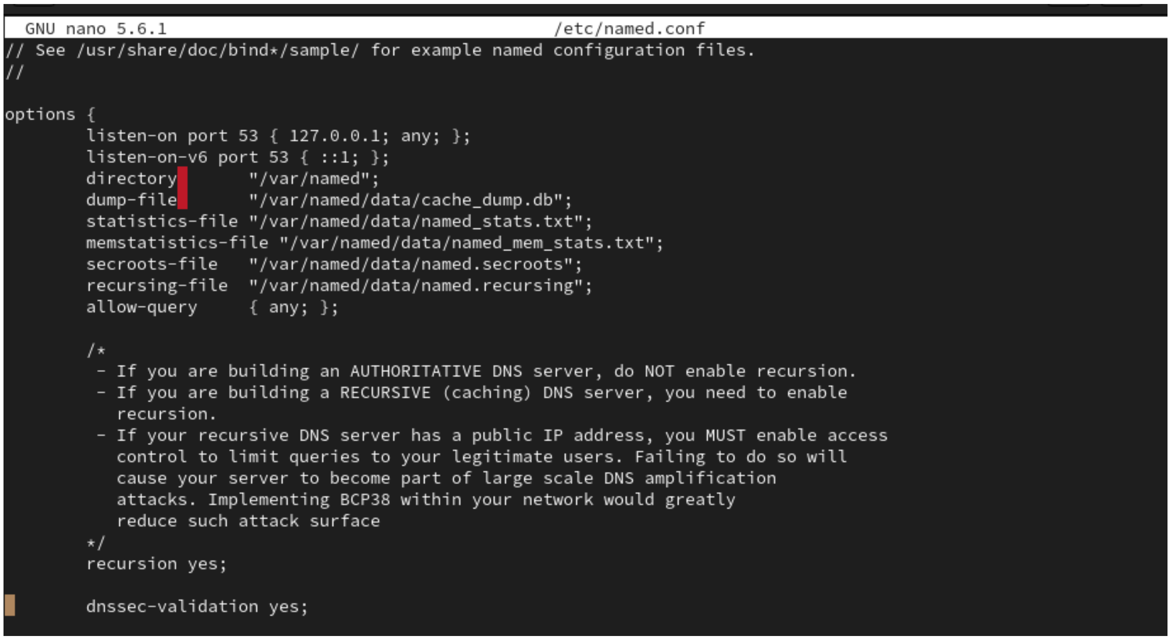
Syntax:

sudo nano /etc/named.conf

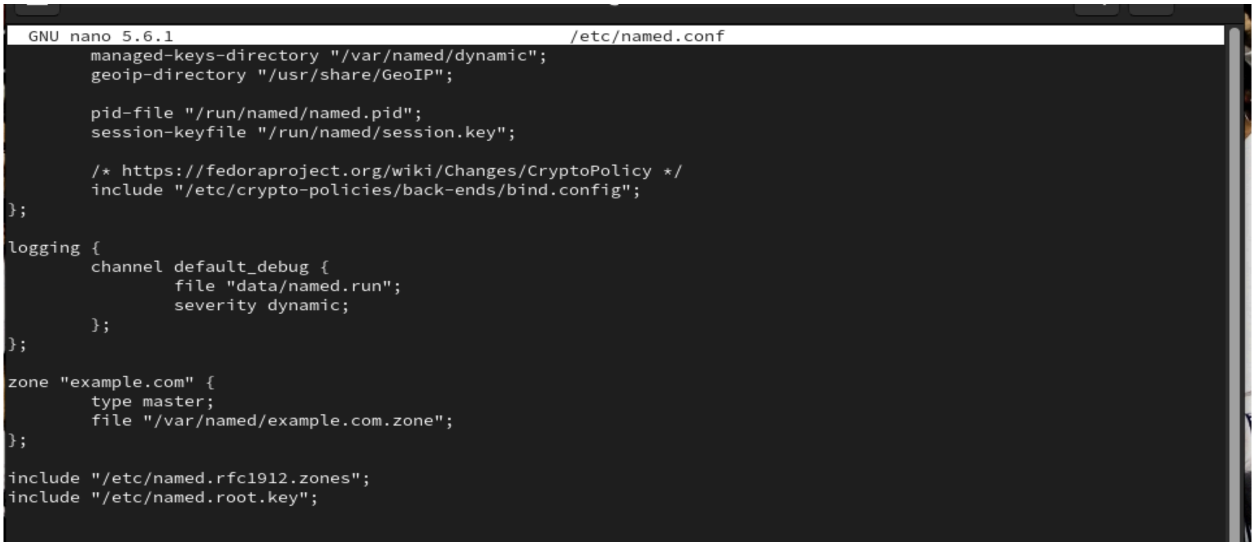


```
Complete!
[root@vbox ~]# sudo nano /etc/named.conf
[root@vbox ~]# sudo nano /var/named/example.com.zone
[root@vbox ~]# sudo nano /var/named/example.com.zone
[root@vbox ~]# sudo nano /etc/named.conf
[root@vbox ~]# sudo nano /var/named/example.com.zone
```

Here's a sample named.conf configuration:



```
  GNU nano 5.6.1                                       /etc/named.conf
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { 127.0.0.1; any; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { any; };

        /*
         - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
         - If you are building a RECURSIVE (caching) DNS server, you need to enable
           recursion.
         - If your recursive DNS server has a public IP address, you MUST enable access
           control to limit queries to your legitimate users. Failing to do so will
           cause your server to become part of large scale DNS amplification
           attacks. Implementing BCP38 within your network would greatly
           reduce such attack surface
        */
        recursion yes;

        dnssec-validation yes;
```



```
 GNU nano 5.6.1                                    /etc/named.conf
        managed-keys-directory "/var/named/dynamic";
        geoip-directory "/usr/share/GeoIP";

        pid-file "/run/named/named.pid";
        session-keyfile "/run/named/session.key";

        /* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
        include "/etc/crypto-policies/back-ends/bind.config";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "example.com" {
        type master;
        file "/var/named/example.com.zone";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

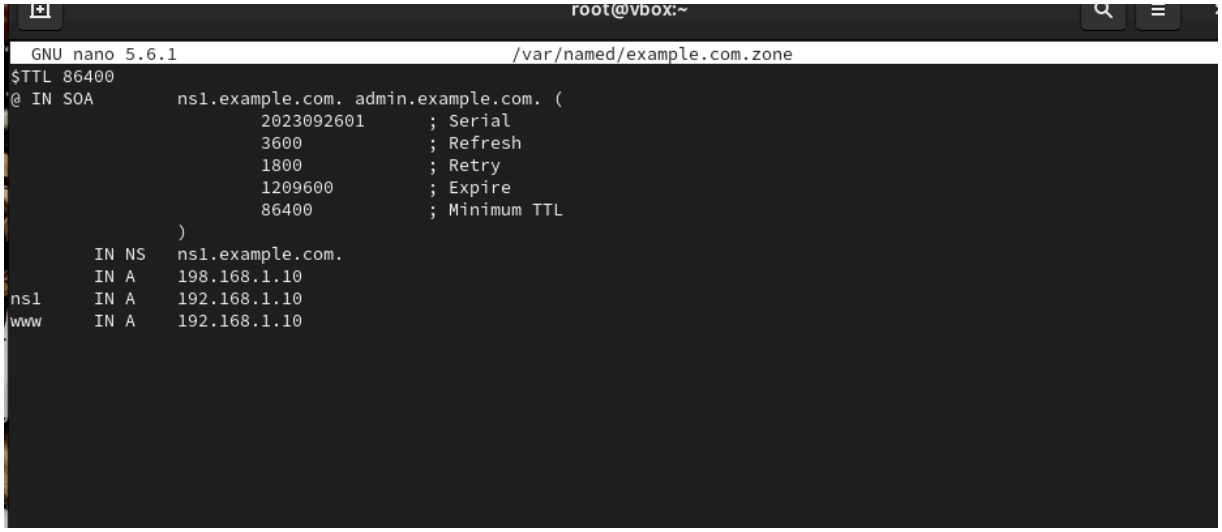
Step 3. Creating the Zone File

Next, create the zone file that contains DNS records for the domain `example.com`.

Syntax:

    sudo nano /var/named/example.com.zone

Example zone file:



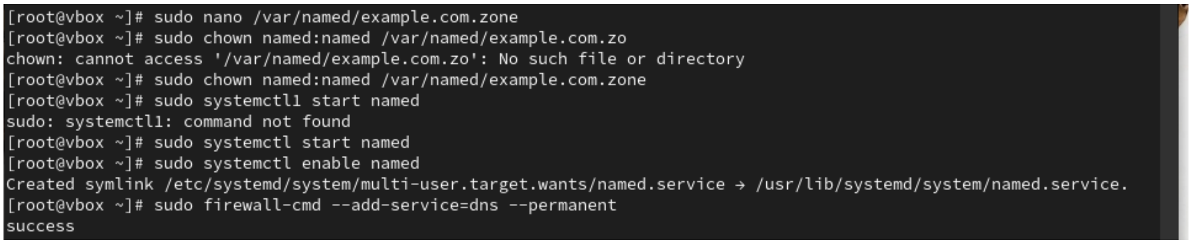
```
 GNU nano 5.6.1                               /var/named/example.com.zone
$TTL 86400
@ IN SOA        ns1.example.com. admin.example.com. (
                        2023092601      ; Serial
                        3600            ; Refresh
                        1800            ; Retry
                        1209600         ; Expire
                        86400           ; Minimum TTL
                )
        IN NS   ns1.example.com.
        IN A    198.168.1.10
ns1     IN A    192.168.1.10
www     IN A    192.168.1.10
```

Step 4. Set Correct File Permissions

BIND runs under the named user, so you need to set the appropriate permissions for the zone files.

Syntax:

```
sudo chown named:named /var/named/example.com.zone
```



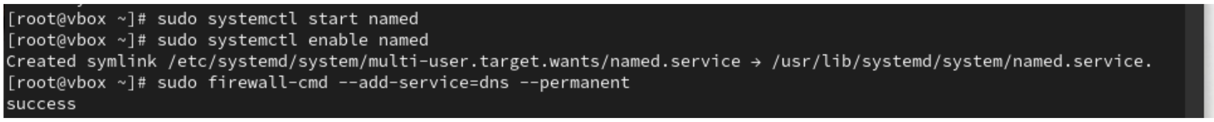
```
[root@vbox ~]# sudo nano /var/named/example.com.zone
[root@vbox ~]# sudo chown named:named /var/named/example.com.zo
chown: cannot access '/var/named/example.com.zo': No such file or directory
[root@vbox ~]# sudo chown named:named /var/named/example.com.zone
[root@vbox ~]# sudo systemctl1 start named
sudo: systemctl1: command not found
[root@vbox ~]# sudo systemctl start named
[root@vbox ~]# sudo systemctl enable named
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
[root@vbox ~]# sudo firewall-cmd --add-service=dns --permanent
success
```

Step 5. Starting and Enabling BIND Service

Now, start the named service and enable it to start automatically on boot.

Syntax:

```
sudo systemctl start named
sudo systemctl enable named
```



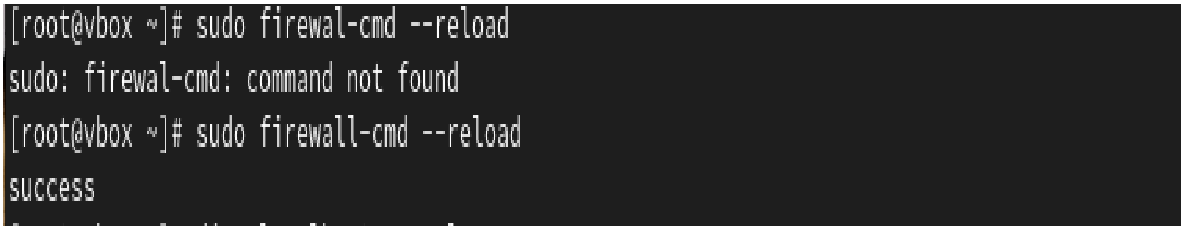
```
[root@vbox ~]# sudo systemctl start named
[root@vbox ~]# sudo systemctl enable named
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
[root@vbox ~]# sudo firewall-cmd --add-service=dns --permanent
success
```

Step 6. Configuring Firewall for DNS

To allow DNS queries to pass through the firewall, you need to allow port 53 (DNS).

Syntax:

```
sudo firewall-cmd --add-service=dns --permanent
sudo firewall-cmd --reload
```



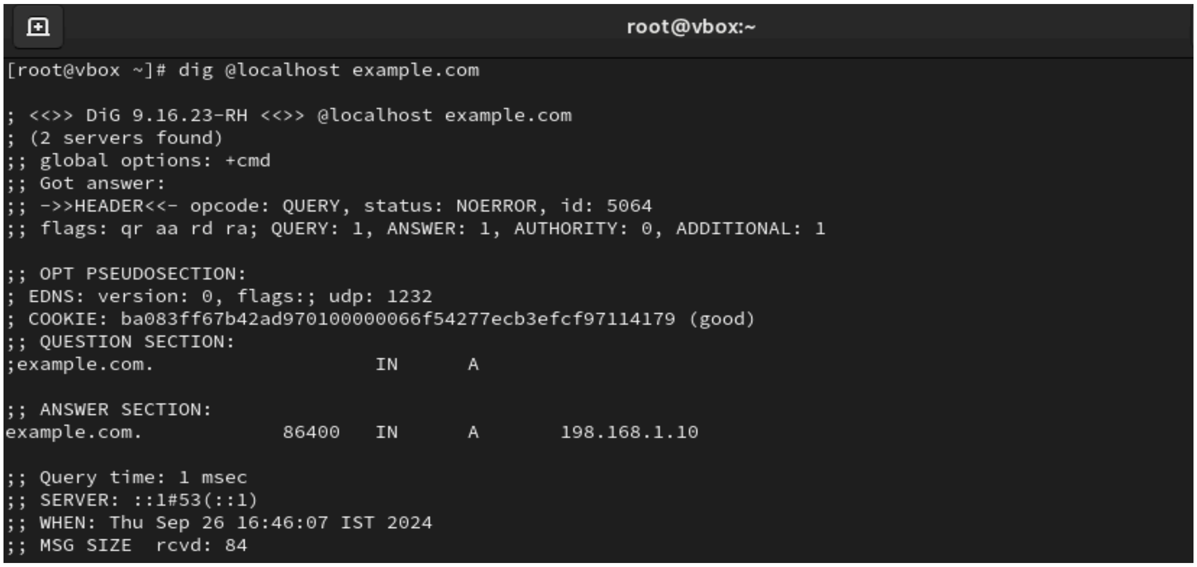
```
[root@vbox ~]# sudo firewal-cmd --reload
sudo: firewal-cmd: command not found
[root@vbox ~]# sudo firewall-cmd --reload
success
```

Step 7. Testing the DNS Server

To ensure your DNS server is working correctly, use the `dig` or `nslookup` commands to query the DNS server.

Syntax:

```
dig @localhost example.com
```



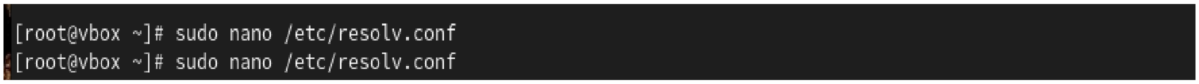
```
[root@vbox ~]# dig @localhost example.com

; <<>> DiG 9.16.23-RH <<>> @localhost example.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5064
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: ba083ff67b42ad970100000066f54277ecb3efcf97114179 (good)
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            86400   IN      A       198.168.1.10

;; Query time: 1 msec
;; SERVER: ::1#53(::1)
;; WHEN: Thu Sep 26 16:46:07 IST 2024
;; MSG SIZE  rcvd: 84
```

Step 8. Configuring a Client to Use the New DNS Server

On a client machine, you need to configure it to use the new DNS server.
[a] Modify the `/etc/resolv.conf` file

Syntax:

```
sudo nano /etc/resolv.conf
```



```
[root@vbox ~]# sudo nano /etc/resolv.conf
[root@vbox ~]# sudo nano /etc/resolv.conf
```

[b] Add the DNS server IP address

In the `resolv.conf` file, add the IP address of your DNS server (replace `192.168.1.10` with the actual IP of your DNS server):

Syntax:

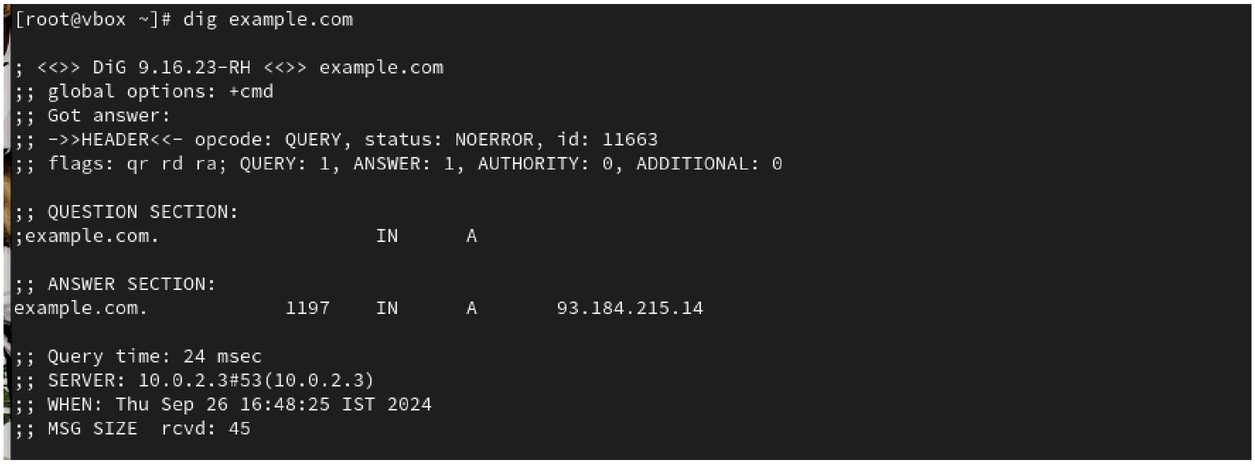
  nameserver 192.168.1.10

Step 9. Verifying DNS Resolution from Client

Once you've configured the DNS server, you can verify that the client is using it by using `dig` or `nslookup`.

Syntax:

  dig example.com



```
[root@vbox ~]# dig example.com

; <<>> DiG 9.16.23-RH <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11663
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            1197    IN      A       93.184.215.14

;; Query time: 24 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Thu Sep 26 16:48:25 IST 2024
;; MSG SIZE  rcvd: 45
```

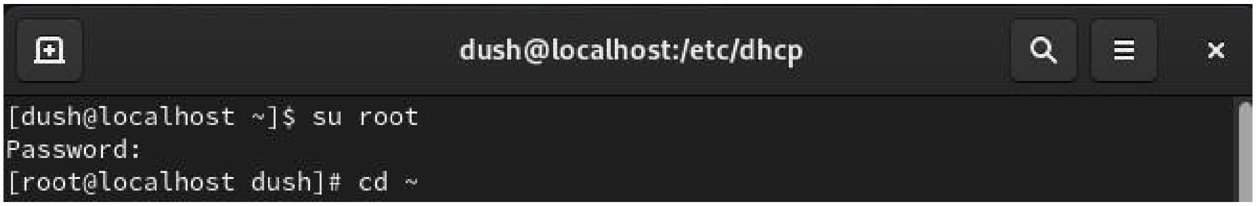
2. Step-by-Step DHCP Configuration on CentOS

Step 1: Login into Root account

Open the terminal and login into the root account by using password .

Syntax:

```
su root
```



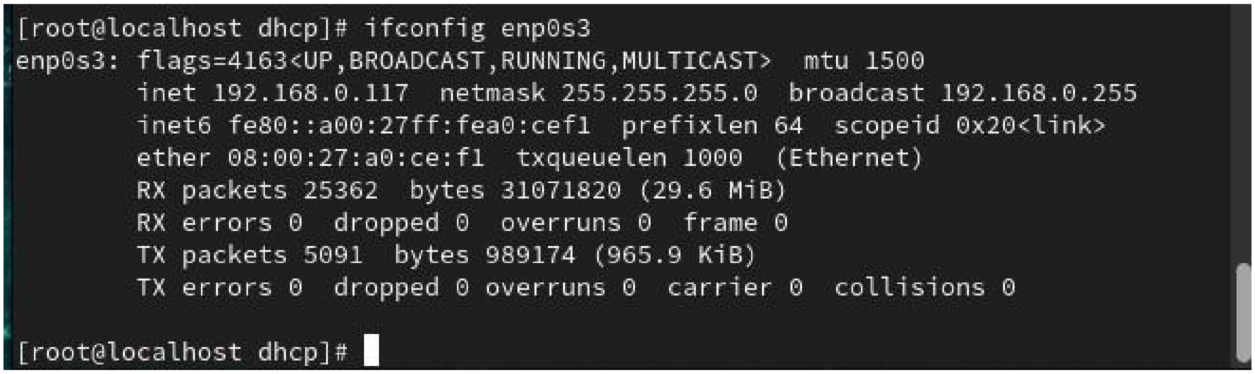
```
[dush@localhost ~]$ su root
Password:
[root@localhost dush]# cd ~
```

Step 2: Configure IP

Check the IP and other network details of wired connection.

Syntax:

```
ifconfig enp0s3
```



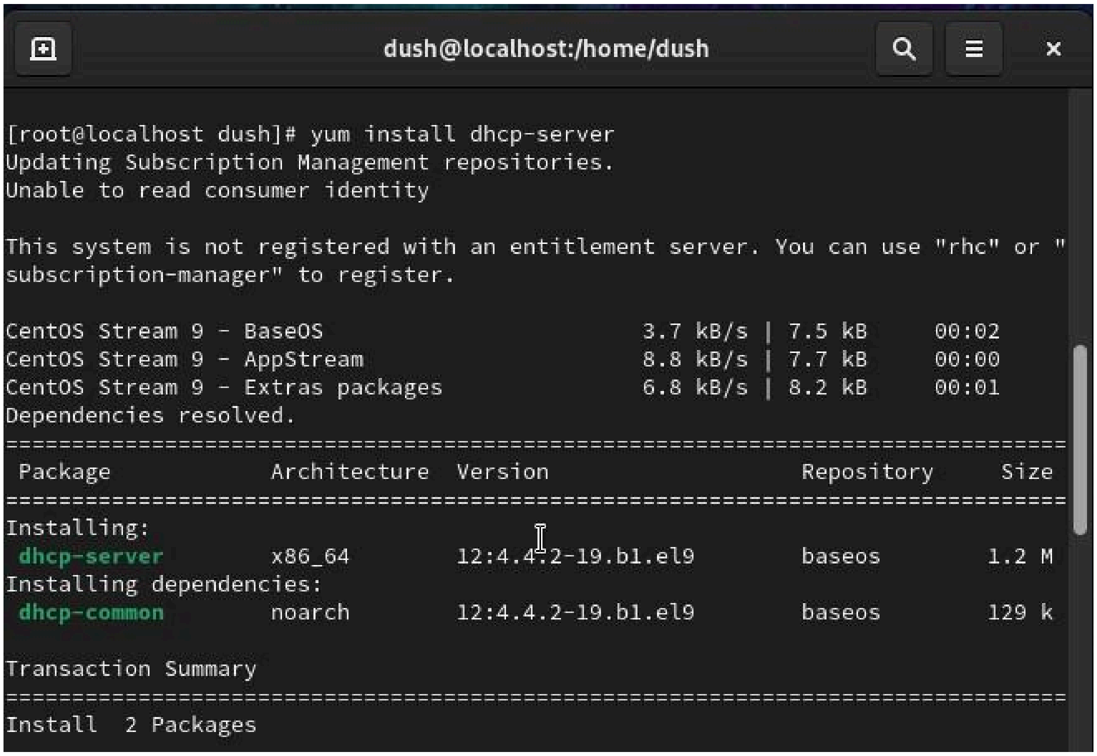
```
[root@localhost dhcp]# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.117  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fea0:cef1  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a0:ce:f1  txqueuelen 1000  (Ethernet)
        RX packets 25362  bytes 31071820 (29.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5091  bytes 989174 (965.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost dhcp]# 
```

Step 3: Install Necessary Packages

Use the yum package manager to install BIND (for DNS) and the DHCP server.

Syntax:

```
yum install dhcp-server -y
```



```
[root@localhost dush]# yum install dhcp-server
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "
subscription-manager" to register.

CentOS Stream 9 - BaseOS                         3.7 kB/s | 7.5 kB     00:02
CentOS Stream 9 - AppStream                      8.8 kB/s | 7.7 kB     00:00
CentOS Stream 9 - Extras packages                6.8 kB/s | 8.2 kB     00:01
Dependencies resolved.
================================================================================
 Package            Architecture  Version                  Repository     Size
================================================================================
Installing:
 dhcp-server        x86_64        12:4.4.2-19.b1.el9       baseos        1.2 M
Installing dependencies:
 dhcp-common        noarch        12:4.4.2-19.b1.el9       baseos        129 k

Transaction Summary
================================================================================
Install  2 Packages
```

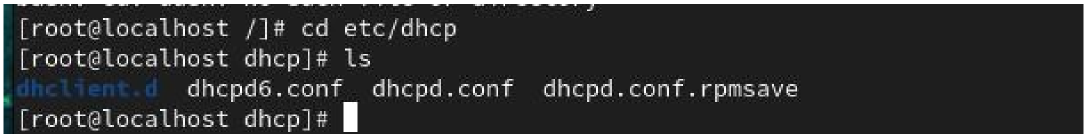
Step 4: Configure DHCP Server

1. Edit the DHCP Configuration File

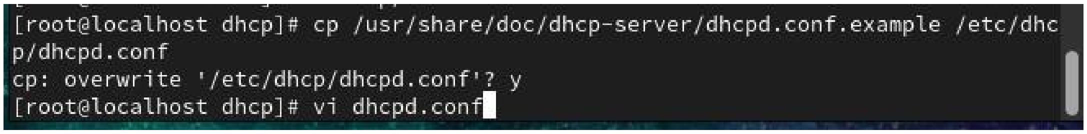
The main DHCP configuration file is /etc/dhcp/dhcpd.conf. Edit this file to define the network ranges and settings for your DHCP clients. Before that, copy the dhcpd.conf.example file to dhcpd.conf file for sample configuration.

Syntax:

```
cd /etc/dhcp
ls
vi /etc/dhcp/dhcpd.conf
```

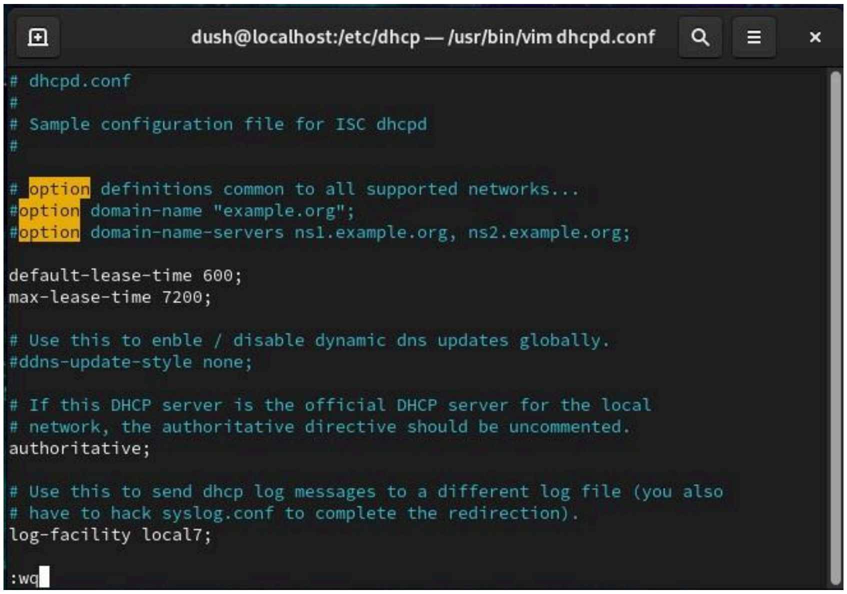


```
[root@localhost /]# cd etc/dhcp
[root@localhost dhcp]# ls
dhclient.d  dhcpd6.conf  dhcpd.conf  dhcpd.conf.rpmsave
[root@localhost dhcp]# 
```

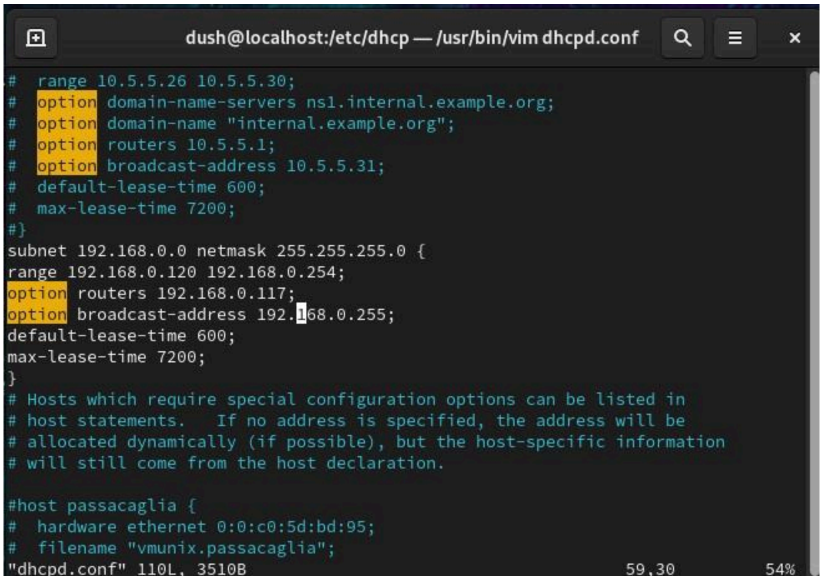


```
[root@localhost dhcp]# cp /usr/share/doc/dhcp-server/dhcpd.conf.example /etc/dhc
p/dhcpd.conf
cp: overwrite '/etc/dhcp/dhcpd.conf'? y
[root@localhost dhcp]# vi dhcpd.conf
```

2. Add Configuration to Assign IP Addresses



```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# 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 enble / 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;

:wq
```



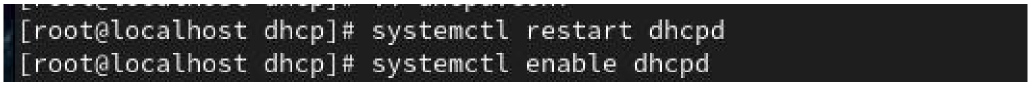
```
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.120 192.168.0.254;
option routers 192.168.0.117;
option broadcast-address 192.168.0.255;
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";
"dhcpd.conf" 110L, 3510B                               59,30          54%
```

3. Start the DHCP Service

Enable and start the DHCP service:

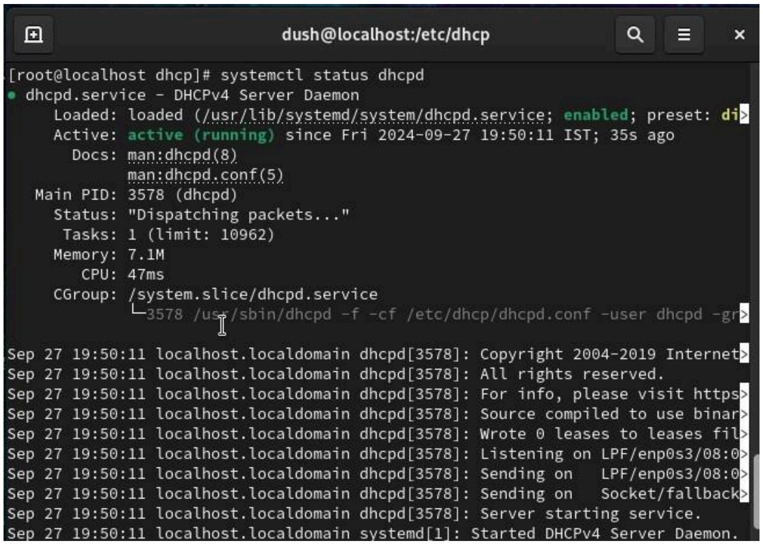
Syntax:

```
systemctl start dhcpd
systemctl enable dhcpd
```



```
[root@localhost dhcp]# systemctl restart dhcpd
[root@localhost dhcp]# systemctl enable dhcpd
```

Check the status to ensure it's running without errors:



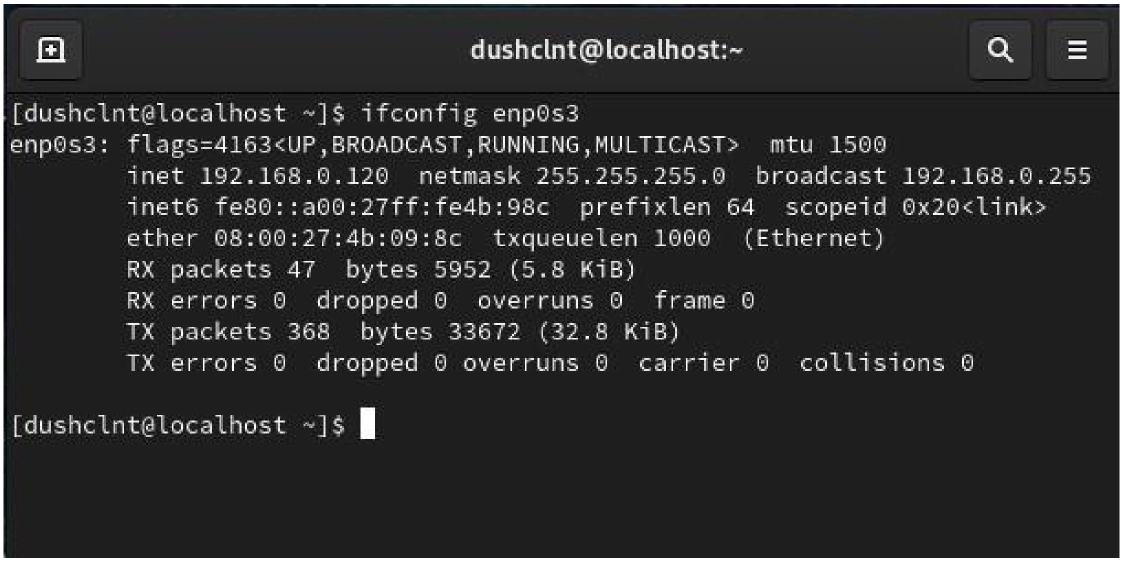
```
[root@localhost dhcp]# systemctl status dhcpd
● dhcpd.service - DHCPv4 Server Daemon
     Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; preset: di>
     Active: active (running) since Fri 2024-09-27 19:50:11 IST; 35s ago
       Docs: man:dhcpd(8)
             man:dhcpd.conf(5)
   Main PID: 3578 (dhcpd)
     Status: "Dispatching packets..."
      Tasks: 1 (limit: 10962)
     Memory: 7.1M
        CPU: 47ms
     CGroup: /system.slice/dhcpd.service
             └─3578 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -gr>

Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: Copyright 2004-2019 Internet>
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: All rights reserved.
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: For info, please visit https>
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: Source compiled to use binar>
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: Wrote 0 leases to leases fil>
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: Listening on LPF/enp0s3/08:0>
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: Sending on   LPF/enp0s3/08:0>
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: Sending on   Socket/fallback>
Sep 27 19:50:11 localhost.localdomain dhcpd[3578]: Server starting service.
Sep 27 19:50:11 localhost.localdomain systemd[1]: Started DHCPv4 Server Daemon.
```

Step 5: Configure ClientOS

Check the IP and other network details of wired connection in ClientOS.

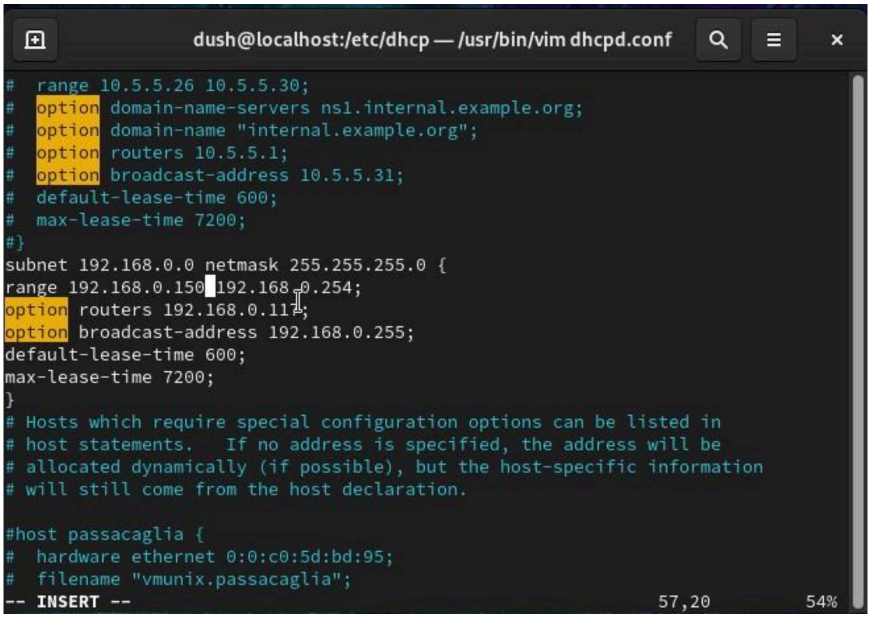
Syntax:

```
ifconfig enp0s3
```



```
[dushclnt@localhost ~]$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.120  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe4b:98c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:4b:09:8c  txqueuelen 1000  (Ethernet)
        RX packets 47  bytes 5952 (5.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 368  bytes 33672 (32.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[dushclnt@localhost ~]$
```

Now, reconfigure the DHCP file is `/etc/dhcp/dhcpd.conf`. Edit this file to test the network ranges and settings for your DHCP clients.



```
#   range 10.5.5.26 10.5.5.30;
#   option domain-name-servers ns1.internal.example.org;
#   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;
#}
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.150 192.168.0.254;
option routers 192.168.0.117;
option broadcast-address 192.168.0.255;
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";
-- INSERT --                                           57,20         54%
```

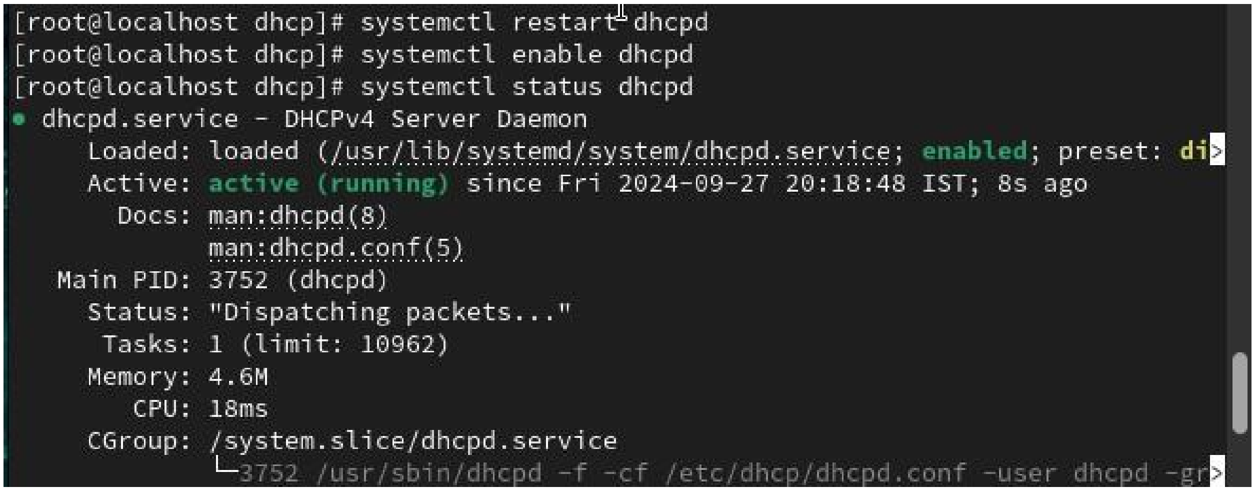
Step 6: Configure ClientOS

Restart the dhcpd and enable the systemctl of it.

Syntax:

```
    systemctl start dhcpd
```

```
systemctl enable dhcpd
```



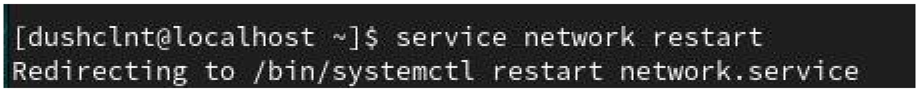
```
[root@localhost dhcp]# systemctl restart dhcpd
[root@localhost dhcp]# systemctl enable dhcpd
[root@localhost dhcp]# systemctl status dhcpd
● dhcpd.service - DHCPv4 Server Daemon
     Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; preset: di>
     Active: active (running) since Fri 2024-09-27 20:18:48 IST; 8s ago
       Docs: man:dhcpd(8)
             man:dhcpd.conf(5)
   Main PID: 3752 (dhcpd)
     Status: "Dispatching packets..."
      Tasks: 1 (limit: 10962)
     Memory: 4.6M
        CPU: 18ms
     CGroup: /system.slice/dhcpd.service
             └─3752 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -gr>
```

Step 7: Configure ClientOS

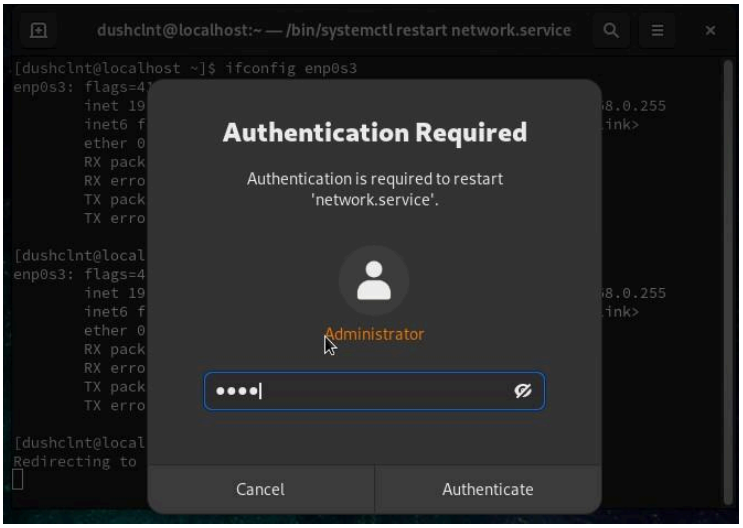
Similarly, Restart the network service to auto assign new DHCP within the range allocated and authenticate to restart it.

Syntax:

```
service network restart
```



```
[dushclnt@localhost ~]$ service network restart
Redirecting to /bin/systemctl restart network.service
```



dushclnt@localhost:~ — /bin/systemctl restart network.service

```
[dushclnt@localhost ~]$ ifconfig enp0s3
enp0s3: flags=4                                     8.0.255
        inet 19                                     ink>
        inet6 f
        ether 0
        RX pack
        RX erro
        TX pack
        TX erro

[dushclnt@local
enp0s3: flags=4                                     8.0.255
        inet 19                                     ink>
        inet6 f
        ether 0
        RX pack
        RX erro
        TX pack
        TX erro

[dushclnt@local
Redirecting to
```

Authentication Required

Authentication is required to restart 'network.service'.

Administrator

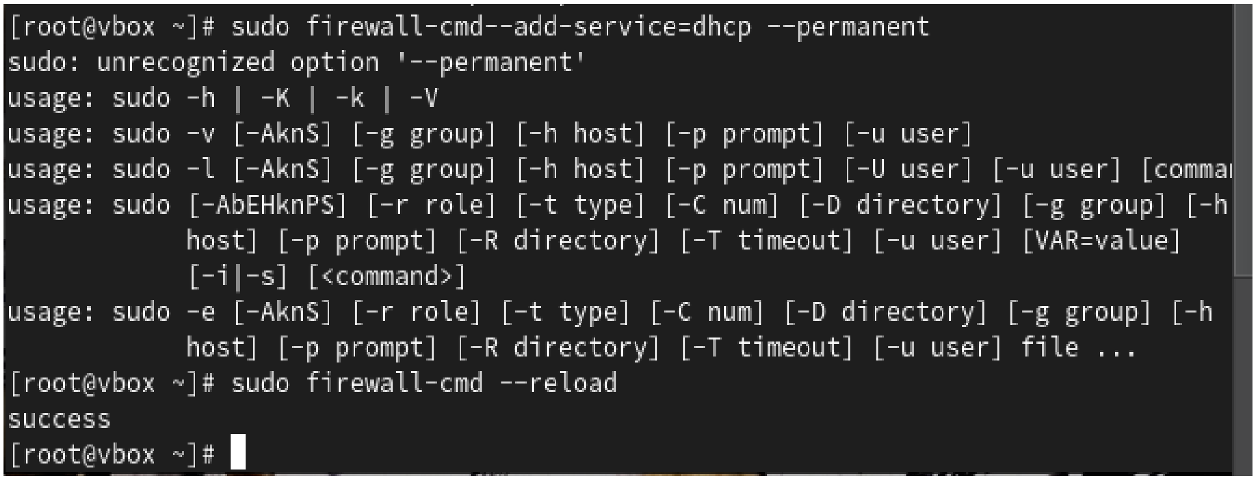
••••

Step 8. Configure the Firewall

If the firewall is enabled on your CentOS 9 system, you need to allow DHCP traffic through the firewall (DHCP uses UDP ports 67 and 68).

Syntax:

```
sudo firewall-cmd --add-service=dhcp --permanent
sudo firewall-cmd --reload
```



```
[root@vbox ~]# sudo firewall-cmd--add-service=dhcp --permanent
sudo: unrecognized option '--permanent'
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [comman
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-D directory] [-g group] [-h
            host] [-p prompt] [-R directory] [-T timeout] [-u user] [VAR=value]
            [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-D directory] [-g group] [-h
            host] [-p prompt] [-R directory] [-T timeout] [-u user] file ...
[root@vbox ~]# sudo firewall-cmd --reload
success
[root@vbox ~]#
```

By following these steps, you can configure a DHCP server on CentOS 9 to dynamically assign IP addresses to clients in your network.

Project Output :

In this project, you have configured two critical network services — DNS (Domain Name System) and DHCP (Dynamic Host Configuration Protocol) — on a CentOS system. These services are essential for the smooth operation of any network, facilitating automated IP address allocation and resolving domain names to their respective IP addresses.

DNS Overview and Configuration Recap

DNS plays a fundamental role in translating human-readable domain names into IP addresses, enabling users to access network resources via easy-to-remember names (e.g., `www.example.com`). We configured the DNS server using BIND (Berkeley Internet Name Domain), one of the most widely used DNS server implementations in Linux environments.

- BIND was installed using `yum`, and the primary configuration file, `/etc/named.conf`, was modified to define forward and reverse lookup zones.
- The forward lookup zone mapped domain names to IP addresses, while the reverse lookup zone performed the reverse operation (IP addresses to domain names).
- We created zone files to store these mappings and verified the DNS setup using tools like `dig` and `nslookup`.

Through these steps, you established a working DNS server that allows both forward and reverse lookups within the network. The DNS server now provides critical services, resolving domain names requested by clients into their associated IP addresses.

DHCP Overview and Configuration Recap

The DHCP server automates the assignment of IP addresses to clients on the network, making network management more efficient and reducing the potential for errors that come with manual IP assignment. Using dhcpd, we configured a DHCP server to dynamically allocate IP addresses from a specified range to devices on the network.

- **DHCP configuration** was performed through the `/etc/dhcp/dhcpd.conf` file, where we defined the IP address range (scope), default gateway, subnet mask, and DNS servers.
- We ensured the DHCP server was listening on the correct network interface, and then enabled and started the service. The DHCP server successfully assigned IP addresses to clients within the predefined range.

By automating IP assignment, the DHCP server eliminates the need for static IP configurations on individual clients. This dynamic setup simplifies network scalability and enhances manageability.

Integration of DNS and DHCP

In a fully functional network, DNS and DHCP often work together. By pointing DHCP clients to the DNS server, all network devices can communicate with each other using human-readable domain names rather than numeric IP addresses. Optionally, you can configure Dynamic DNS (DDNS) updates, where the DHCP server automatically registers new devices with the DNS server. This ensures seamless domain name resolution for devices receiving their IP addresses dynamically.

Challenges and Considerations

Throughout the project, several critical aspects of system administration and networking were addressed:

1. **Network Interface Configuration**: Ensuring the server's network interface was configured with the correct IP address and gateway.
2. **Security and Permissions**: Adjusting file permissions for BIND zone files to allow the DNS service to function correctly.
3. **Service Management**: Utilizing `systemctl` to manage services and ensure they start at boot.

4. **Troubleshooting**: Diagnosing issues using tools like `journalctl`, `systemctl status`, and `dig` to resolve errors related to DNS or DHCP services.

Real-World Applications

This project closely mirrors real-world tasks performed by system administrators in corporate and data center environments. Configuring DNS and DHCP is essential for managing internal networks, especially in scenarios involving:

- **Corporate networks**: Where devices need automated IP address assignment and internal domain name resolution.
- **Virtualized environments**: Where virtual machines are frequently created and destroyed, requiring dynamic network configurations.
- **Home networks**: Providing network connectivity to multiple devices with minimal manual configuration.

By understanding how to configure and manage these services, you gain essential skills for maintaining network infrastructure in a Linux-based environment.

Summary of Commands :

DNS :

Install BIND:

```
sudo dnf install bind bind-utils
```

1.

Edit BIND configuration:

```
sudo nano /etc/named.conf
```

2.

Create zone file:

```
sudo nano /var/named/example.com.zone
```

3.

Set correct permissions:

```
sudo chown named:named /var/named/example.com.zone
```

4.

Start and enable BIND:

```
sudo systemctl start named

sudo systemctl enable named
```

5.

Allow DNS through firewall:

```
sudo firewall-cmd --add-service=dns --permanent

sudo firewall-cmd --reload
```

6.

Test DNS server:

```
dig @localhost example.com
```

7.

Configure client DNS:

```
sudo nano /etc/resolv.conf
```

8.

Verify DNS resolution from client:

```
dig example.com
```

This complete set of steps and commands will help you configure a basic DNS server and client setup on CentOS 9.

DHCP:

Install the DHCP server:

```
sudo dnf install dhcp-server
```

1.

Edit the DHCP configuration:

```
sudo nano /etc/dhcp/dhcpd.conf
```

2.

Specify the network interface for DHCP:

```
sudo nano /etc/sysconfig/dhcpd
```

3.

Start and enable the DHCP service:

```
sudo systemctl start dhcpd

sudo systemctl enable dhcpd
```

4.

Allow DHCP traffic through the firewall:

```
sudo firewall-cmd --add-service=dhcp --permanent

sudo firewall-cmd --reload
```

5.

Verify DHCP server status:

```
sudo systemctl status dhcpd
```

6.

Check DHCP leases:

```
cat /var/lib/dhcpd/dhcpd.leases
```

By following these steps, you can configure a DHCP server on CentOS 9 to dynamically assign IP addresses to clients in your network.

Conclusion :

Successfully completing the DNS and DHCP configuration project on CentOS has provided you with a deep understanding of two of the most important network services. The ability to configure DNS with BIND and set up a DHCP server using `dhcpd` is a valuable skill for system administrators. This project enables you to implement a robust, scalable, and automated IP addressing and domain name resolution system in any network environment.

With the integration of DNS and DHCP services, your CentOS system is now capable of automating network operations, enhancing network efficiency, and reducing administrative overhead. These are key steps toward building and maintaining a resilient and efficient network infrastructure.