PROJECT1:

DNS and DHCP Server Administration in Linux

26.09.2024

—

Prepared By:

K. Dushyant Reddy

Guided By:

Zakir Hussain

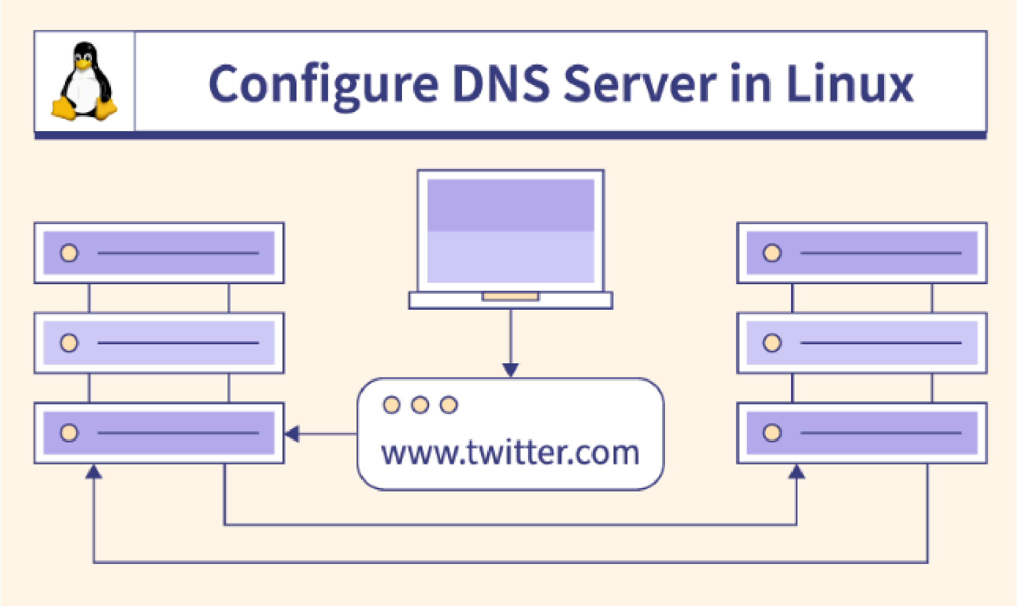
INDEX

OBJECTIVE

This project focuses on setting up two essential services—DNS (Domain Name System) and DHCP (Dynamic Host Configuration Protocol)—on a CentOS Linux system.

What Do These Services Do?

1. **Domain Name System (DNS)**: DNS acts like a phone book for the internet. When you type a website's name (like www.cloudush.com), DNS translates that name into the corresponding IP address so your device can find it.

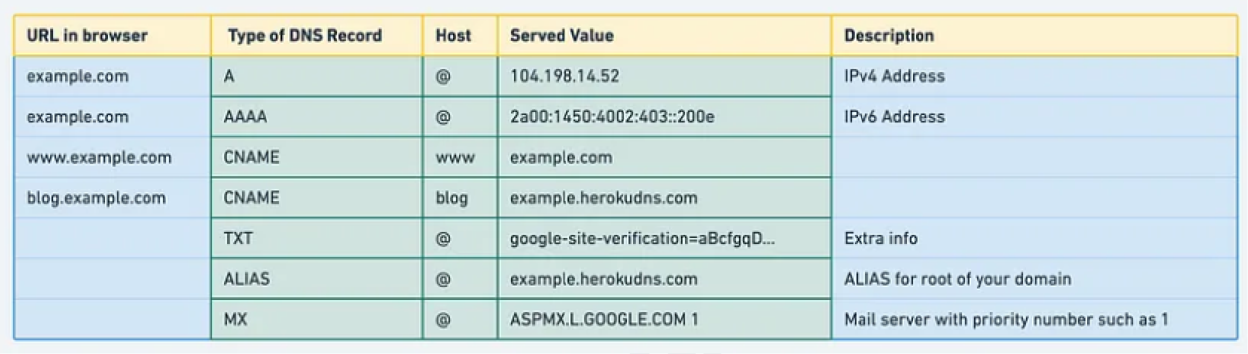


Structure:

- **Hierarchy** - DNS has a hierarchical structure, starting from the root domain (represented by a dot) and branching out to top-level domains (TLDs) like .com, .org, and .net. Each level is separated by a dot.
- **Domain Registration** - To have a domain name, you need to register it through a domain registrar. This ties your domain to an IP address through DNS records.

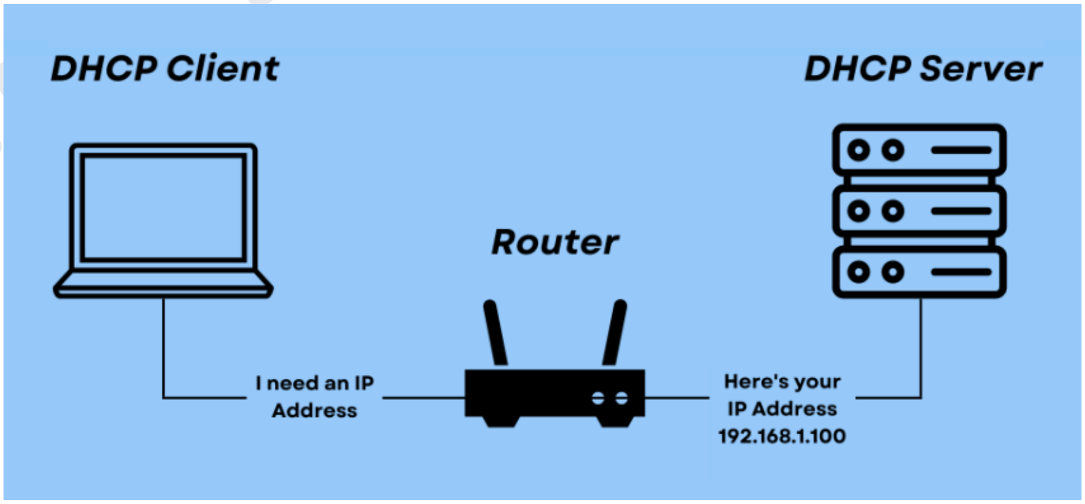
DNS Records:

- **A Record** - Maps a domain name to an IPv4 address.
- **AAAA Record** - Maps a domain name to an IPv6 address.
- **CNAME Record** - Alias of one domain name to another (e.g., www to example.com).
- **MX Record** - Specifies the mail servers for a domain, helping with email routing.
- **NS Record** - Indicates which name servers are authoritative for the domain



URL in browser	Type of DNS Record	Host	Served Value	Description
example.com	A	@	104.198.14.52	IPv4 Address
example.com	AAAA	@	2a00:1450:4002:403::200e	IPv6 Address
www.example.com	CNAME	www	example.com	
blog.example.com	CNAME	blog	example.herokudns.com	
	TXT	@	google-site-verification=aBcfgqD...	Extra info
	ALIAS	@	example.herokudns.com	ALIAS for root of your domain
	MX	@	ASPMX.L.GOOGLE.COM 1	Mail server with priority number such as 1

2. **Dynamic Host Configuration Protocol (DHCP)**: Think of DHCP as the system that automatically assigns IP addresses to devices on your network. Instead of manually configuring each device, DHCP makes it easy by dynamically giving them the right addresses.

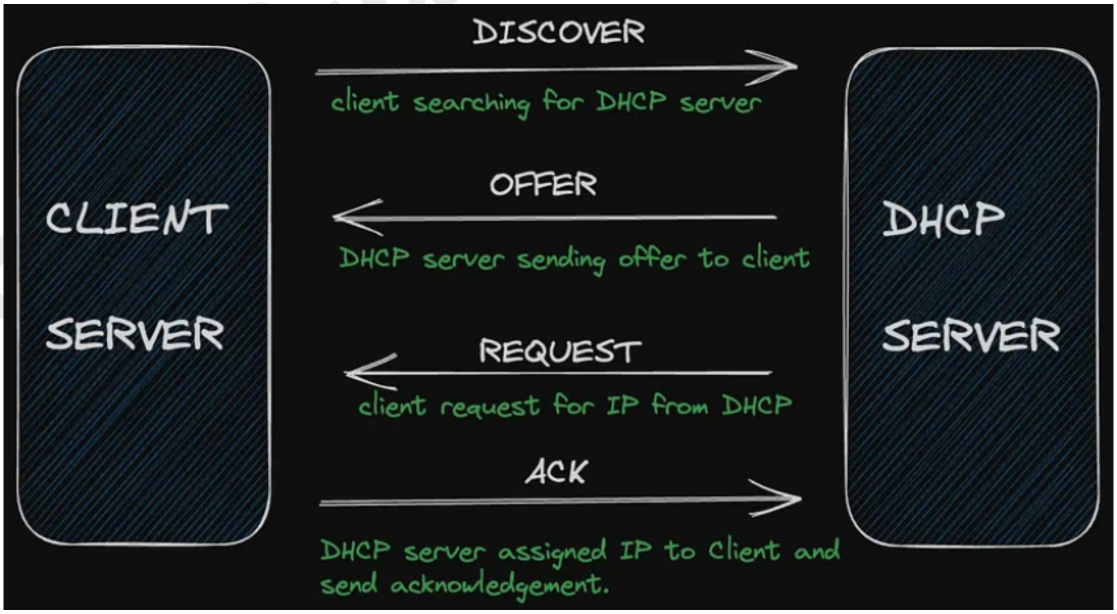


Components:

- **DHCP Server** - This is the device (often a router or dedicated server) that manages the DHCP process. It holds the IP address pool and configuration settings.
- **DHCP Client** - Any device (like computers, smartphones, printers) that requests configuration information from the DHCP server.
- **IP Address Pool** - A range of IP addresses available for assignment to clients.

DHCP Lease Process:

- **DHCP Discover** - The client broadcasts a message (DHCP Discover) to the local network to find available DHCP servers.
- **DHCP Offer** - Each DHCP server that receives the Discover message responds with a DHCP Offer message. This message includes an available IP address and other configuration details (like subnet mask, gateway, DNS servers).
- **DHCP Request** - The client selects one of the Offers and broadcasts a DHCP Request message back to the chosen server, indicating that it wants to accept the offered IP address.
- **DHCP Acknowledgment (ACK)** - The DHCP server responds with a DHCP Acknowledgment message, confirming the IP address lease. This message may also include additional configuration options.


CLIENT
SERVER
DISCOVER
client searching for DHCP server
OFFER
DHCP server sending offer to client
REQUEST
client request for IP from DHCP
ACK
DHCP server assigned IP to Client and send acknowledgement.
DHCP
SERVER

REAL-TIME SCENARIO

Company Background:

ABCD Marketing Company is a mid-sized marketing firm founded in 2010 by a group of entrepreneurs with a passion for innovative marketing strategies. The company has grown rapidly over the years, with a current workforce of 200 employees across three offices in the United States. ABCD Marketing Company specializes in providing digital marketing services, including social media management, content creation, and search engine optimization (SEO) to a diverse range of clients. However, the rapid growth and lack of centralized network management have led to a complex and inefficient network infrastructure.

Current Scenario:

ABCD Marketing Company is currently facing a critical issue with its network infrastructure. The company's network administrator is struggling to manage the IP addresses and domain name resolution for internal services and devices. The current setup is decentralized, with multiple DHCP servers and DNS servers scattered across the three office locations. This has resulted in:

- IP address conflicts and duplication
- Inefficient domain name resolution, leading to slow network performance
- Difficulty in tracking and managing devices on the network
- Inability to implement a centralized network management system

Problem Statement:

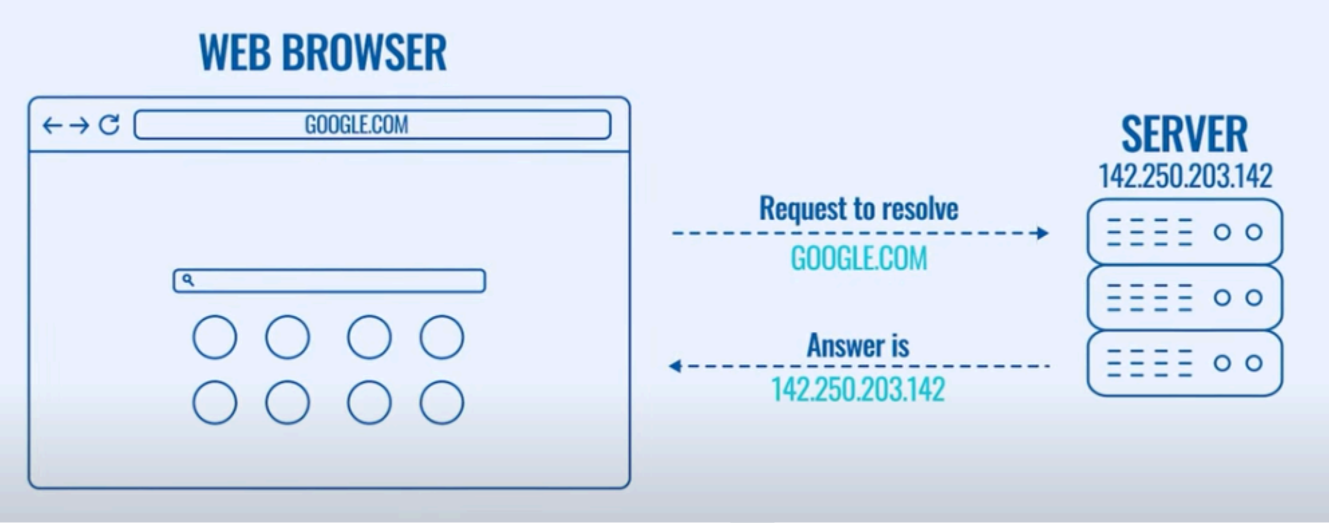
The network administrator at ABCD Marketing Company needs to design and implement a centralized solution for managing IP addresses and ensuring efficient domain name resolution for internal services and devices. The solution must be scalable, secure, and easy to manage, with the ability to integrate with existing network infrastructure.

Requirements:

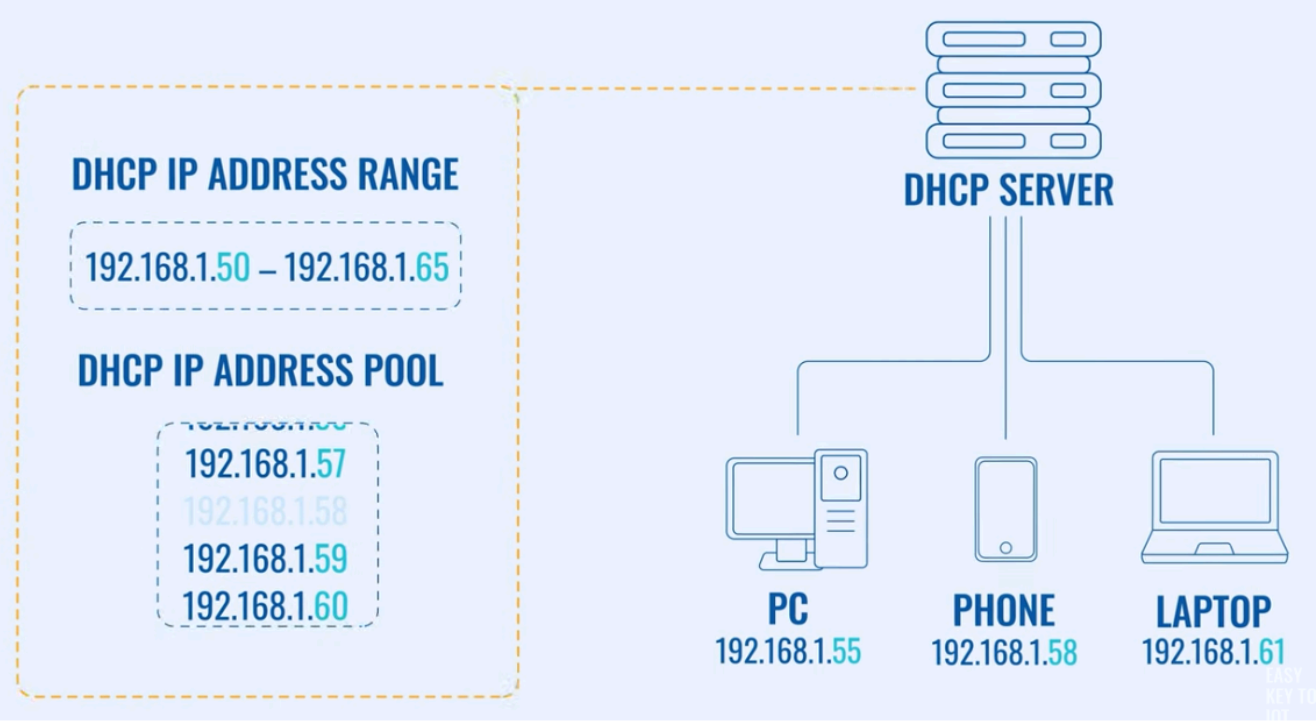
- Implement a centralized DNS solution to manage domain name resolution for internal services and devices
- Configure a DHCP server to assign IP addresses dynamically to devices on the network
- Ensure efficient network performance and minimize IP address conflicts
- Implement a centralized network management system to track and manage devices on the network
- Ensure scalability and security of the solution to support future growth and expansion

By addressing this, ABCD Marketing Company can improve its network infrastructure, increase efficiency, and reduce costs associated with network management and maintenance

ARCHITECTURE DIAGRAM



Domain Name System Server Configuration



Dynamic Host Configuration Protocol Server Configuration

IMPLEMENTATION

1. DNS Server Configuration

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`.

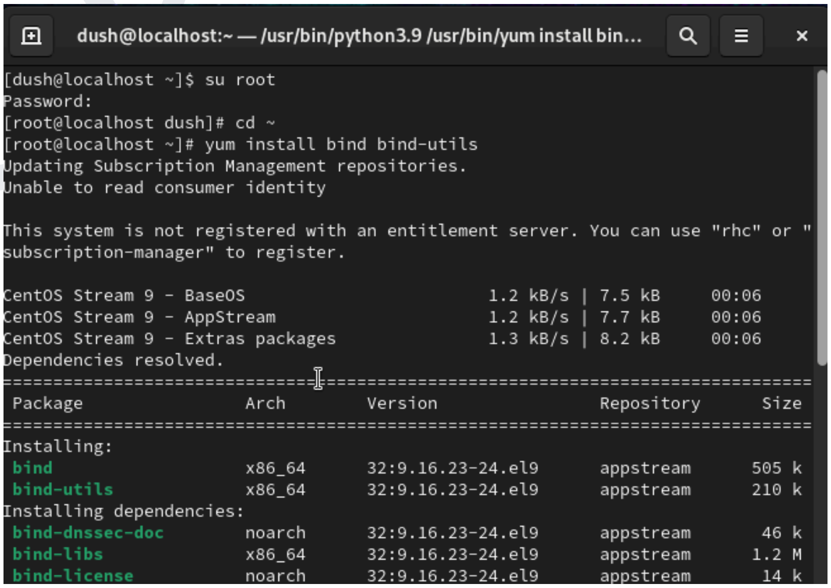
Steps

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 yum install bind bind-utils
```


dush@localhost:~ — /usr/bin/python3.9 /usr/bin/yum install bin…

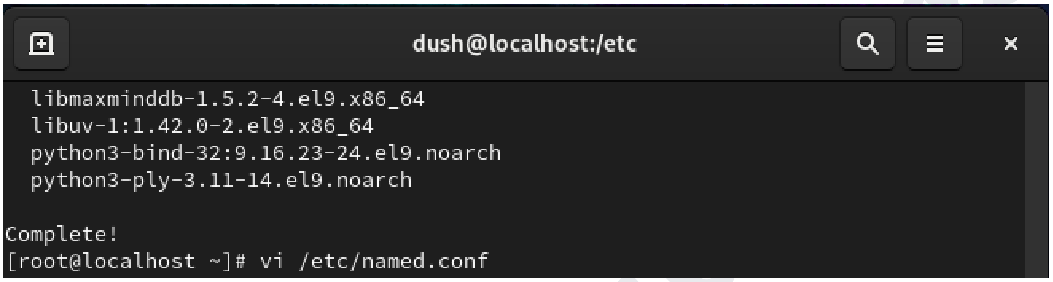
```
[dush@localhost ~]$ su root
Password:
[root@localhost dush]# cd ~
[root@localhost ~]# yum install bind bind-utils
Updating Subscription Management repositories.
Unable to read consumer identity

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

CentOS Stream 9 - BaseOS                        1.2 kB/s | 7.5 kB     00:06
CentOS Stream 9 - AppStream                     1.2 kB/s | 7.7 kB     00:06
CentOS Stream 9 - Extras packages               1.3 kB/s | 8.2 kB     00:06
Dependencies resolved.
================================================================================
 Package               Arch         Version                Repository      Size
================================================================================
Installing:
 bind                  x86_64       32:9.16.23-24.el9      appstream      505 k
 bind-utils            x86_64       32:9.16.23-24.el9      appstream      210 k
Installing dependencies:
 bind-dnssec-doc       noarch       32:9.16.23-24.el9      appstream       46 k
 bind-libs             x86_64       32:9.16.23-24.el9      appstream      1.2 M
 bind-license          noarch       32:9.16.23-24.el9      appstream       14 k
```

Step 2. Configuring BIND as a DNS Server

Once BIND is installed, the next step is to configure it.
The main configuration file for BIND is located at /etc/named.conf.

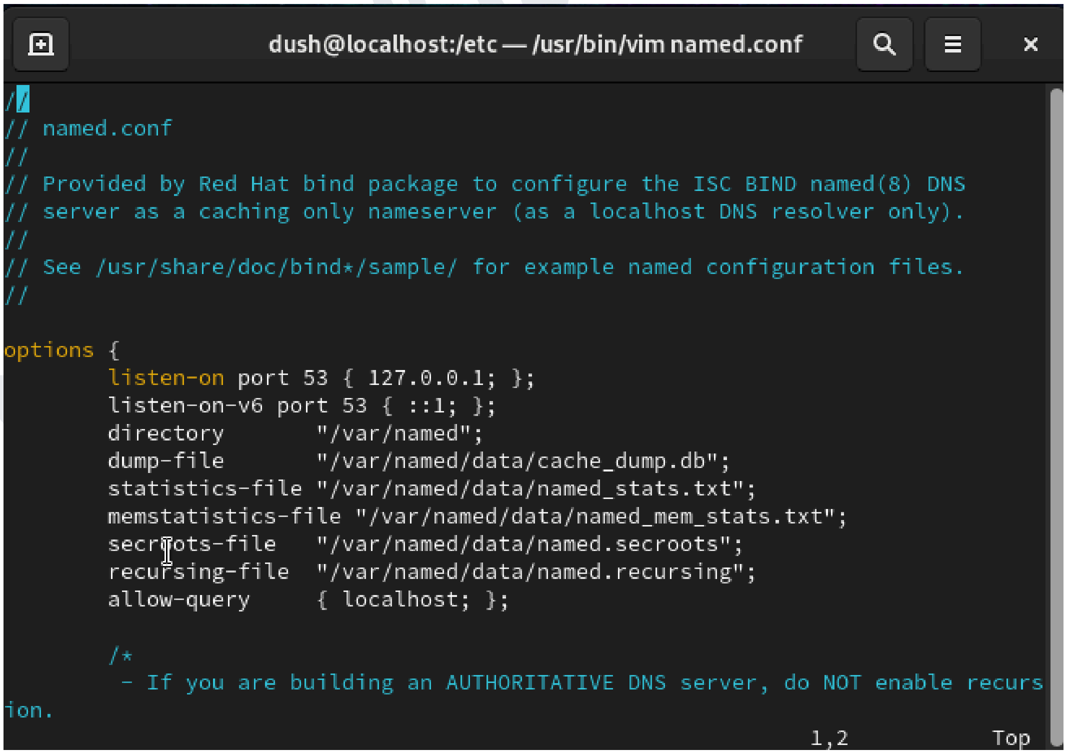
Syntax:

```
sudo vi /etc/named.conf
```

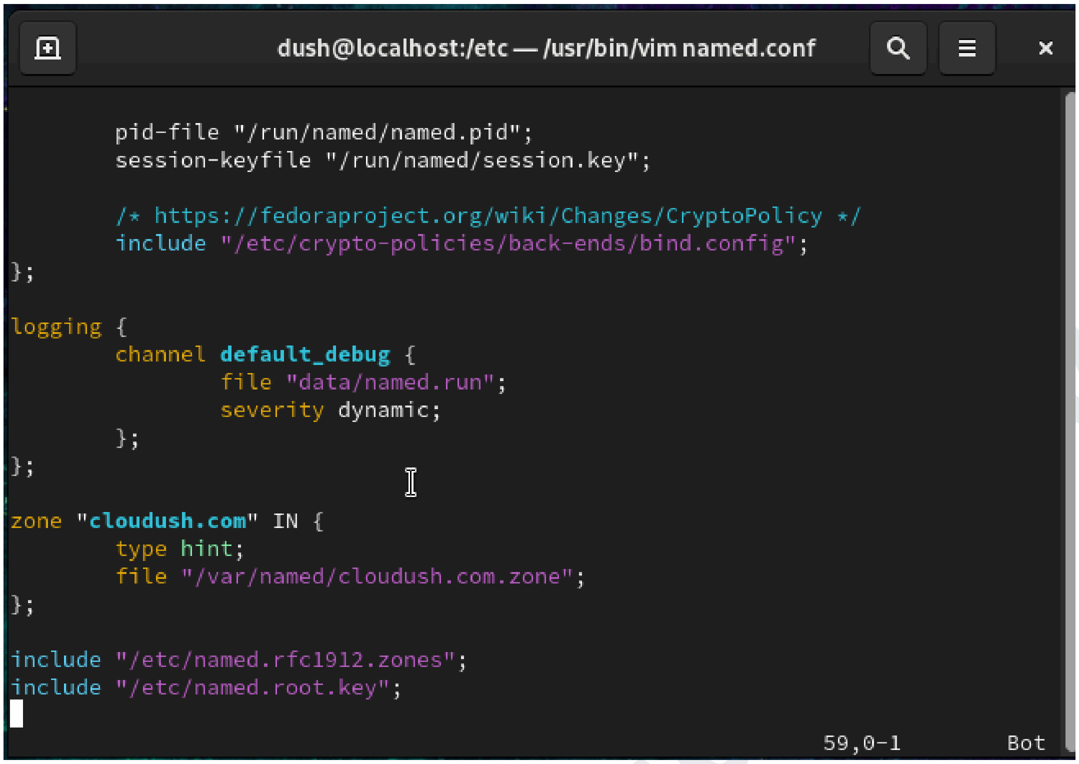


```
  libmaxminddb-1.5.2-4.el9.x86_64
  libuv-1:1.42.0-2.el9.x86_64
  python3-bind-32:9.16.23-24.el9.noarch
  python3-ply-3.11-14.el9.noarch

Complete!
[root@localhost ~]# vi /etc/named.conf
```

Here's a sample named.conf configuration:



```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { 127.0.0.1; };
        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     { localhost; };

        /*
         - If you are building an AUTHORITATIVE DNS server, do NOT enable recurs
ion.
                                                              1,2           Top
```



```
        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 "cloudush.com" IN {
        type hint;
        file "/var/named/cloudush.com.zone";
};

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

                                                              59,0-1        Bot
```

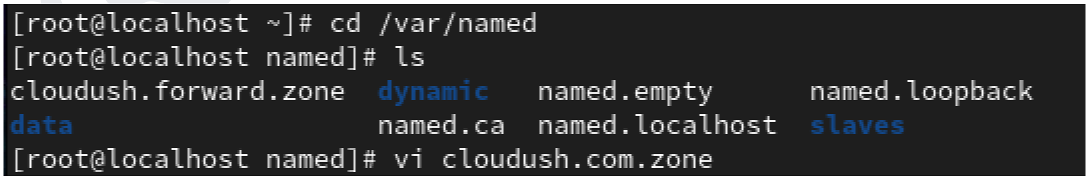
Step 3. Creating the Zone File

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

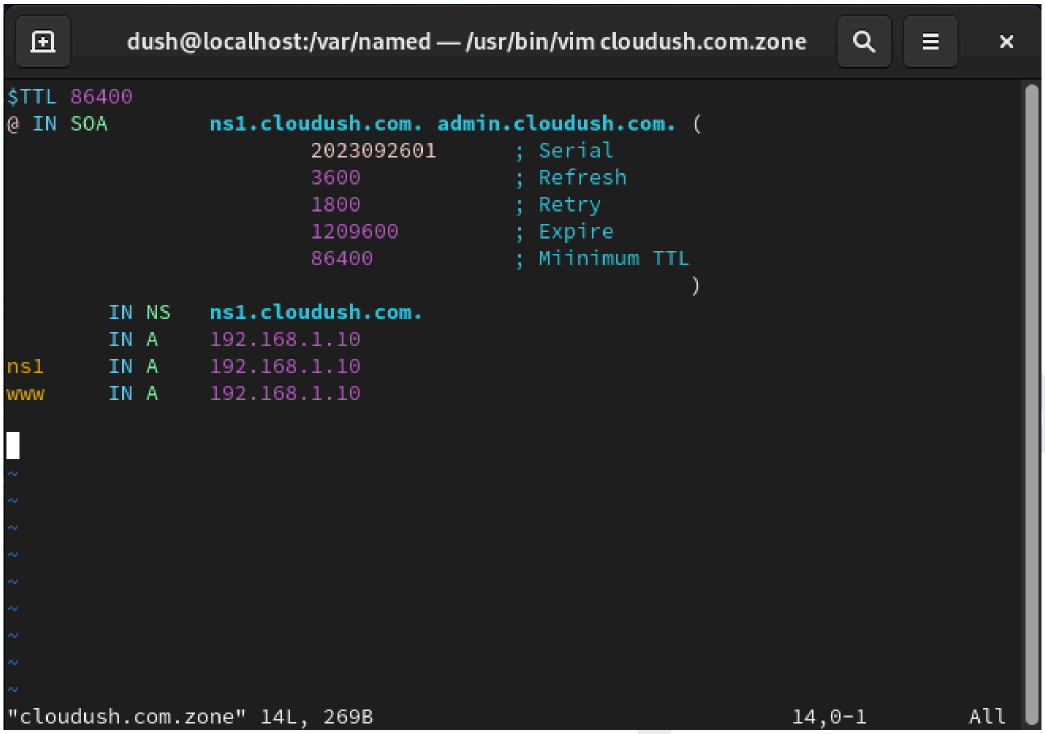
Syntax:

```
    sudo vi /var/named/cloudush.com.zone
```

Example zone file:



```
[root@localhost ~]# cd /var/named
[root@localhost named]# ls
cloudush.forward.zone  dynamic   named.empty      named.loopback
data                   named.ca  named.localhost  slaves
[root@localhost named]# vi cloudush.com.zone
```



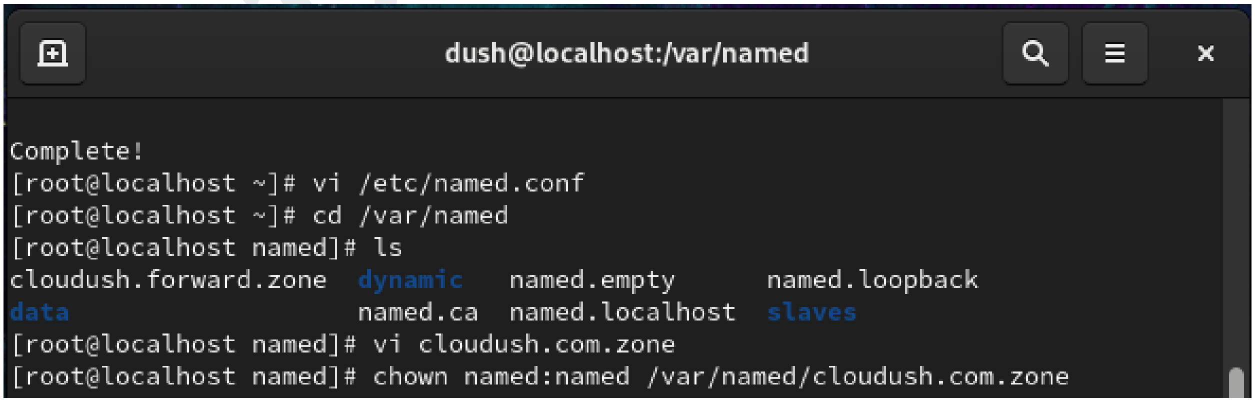
```
$TTL 86400
@ IN SOA        ns1.cloudush.com. admin.cloudush.com. (
                        2023092601      ; Serial
                        3600            ; Refresh
                        1800            ; Retry
                        1209600         ; Expire
                        86400           ; Miinimum TTL
                                                      )
        IN NS   ns1.cloudush.com.
        IN A    192.168.1.10
ns1     IN A    192.168.1.10
www     IN A    192.168.1.10

~
~
~
~
~
~
~
~
~
"cloudush.com.zone" 14L, 269B                               14,0-1        All
```

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/cloudush.com.zone
```



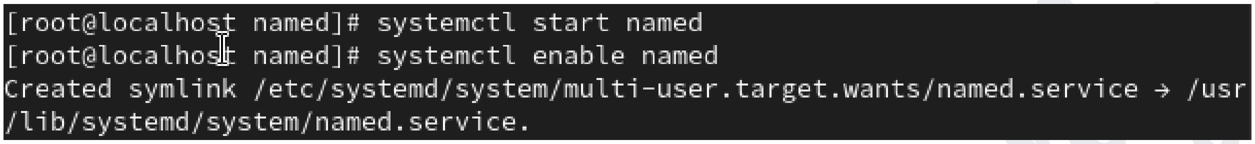
```
Complete!
[root@localhost ~]# vi /etc/named.conf
[root@localhost ~]# cd /var/named
[root@localhost named]# ls
cloudush.forward.zone  dynamic   named.empty      named.loopback
data                   named.ca  named.localhost  slaves
[root@localhost named]# vi cloudush.com.zone
[root@localhost named]# chown named:named /var/named/cloudush.com.zone
```

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
```



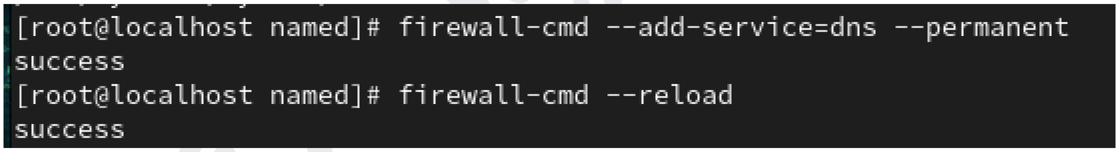
```
[root@localhost named]# systemctl start named
[root@localhost named]# systemctl enable named
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr
/lib/systemd/system/named.service.
```

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
```



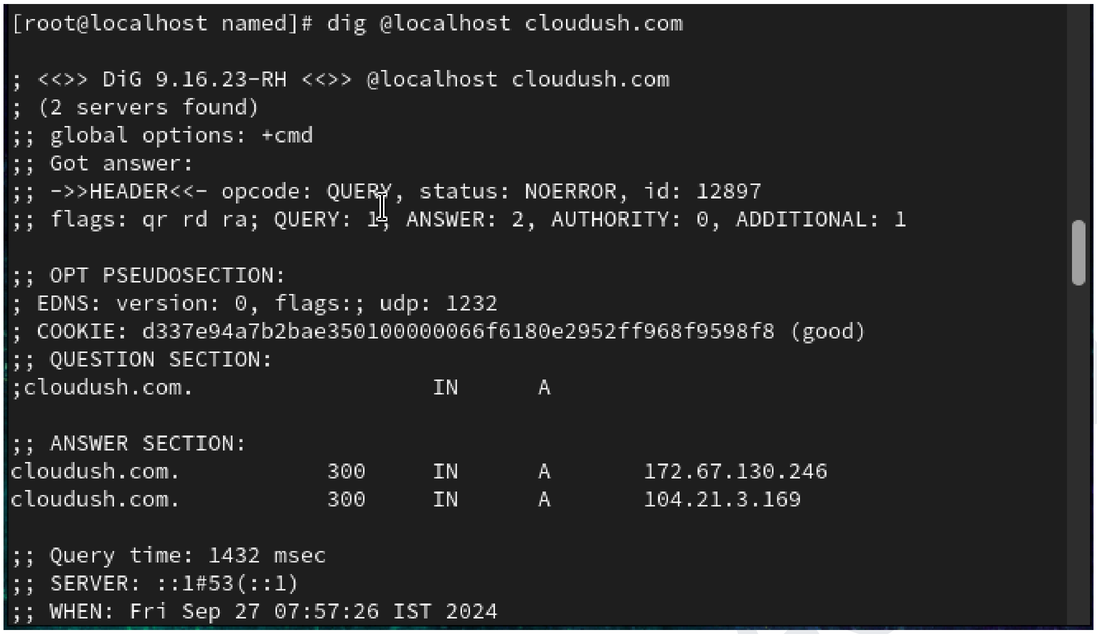
```
[root@localhost named]# firewall-cmd --add-service=dns --permanent
success
[root@localhost named]# firewall-cmd --reload
success
```

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 cloudush.com
```



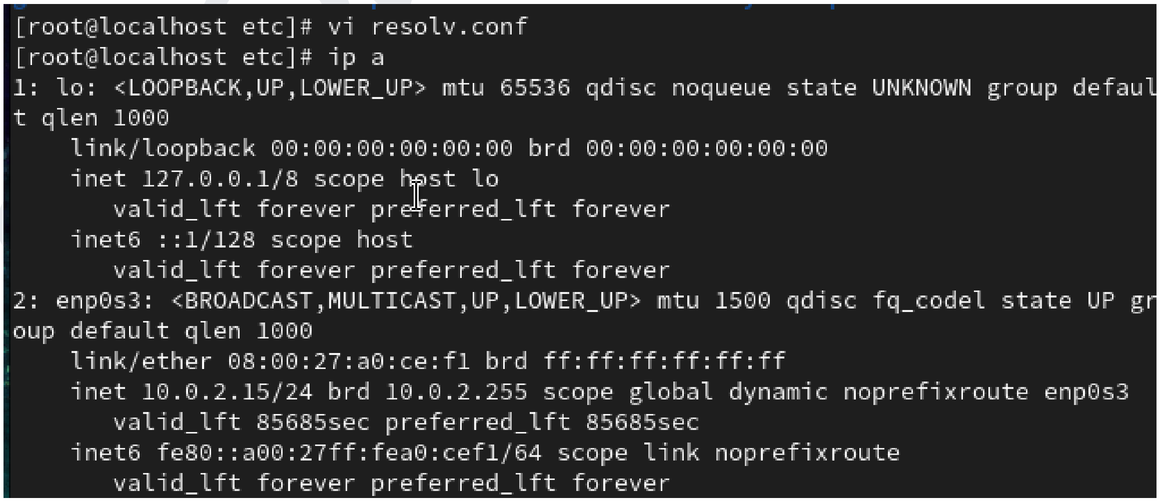
```
[root@localhost named]# dig @localhost cloudush.com

; <<>> DiG 9.16.23-RH <<>> @localhost cloudush.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12897
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: d337e94a7b2bae350100000066f6180e2952ff968f9598f8 (good)
;; QUESTION SECTION:
;cloudush.com.                  IN      A

;; ANSWER SECTION:
cloudush.com.           300     IN      A       172.67.130.246
cloudush.com.           300     IN      A       104.21.3.169

;; Query time: 1432 msec
;; SERVER: ::1#53(::1)
;; WHEN: Fri Sep 27 07:57:26 IST 2024
```

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 vi /etc/resolv.conf
```



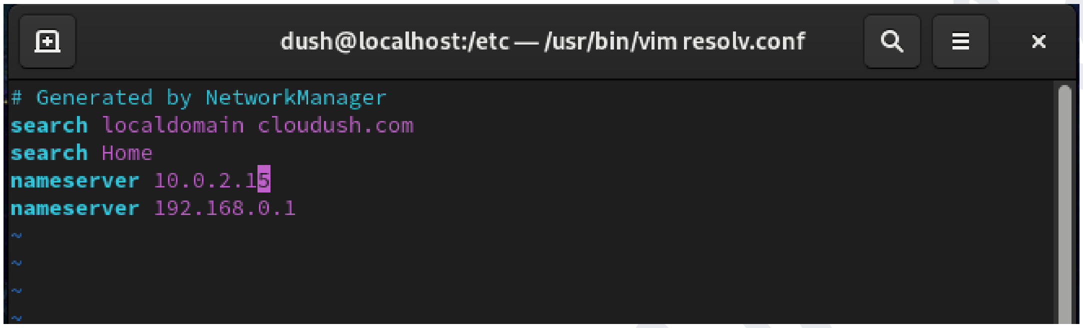
```
[root@localhost etc]# vi resolv.conf
[root@localhost etc]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:a0:ce:f1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85685sec preferred_lft 85685sec
    inet6 fe80::a00:27ff:fea0:cef1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

[b] Add the DNS server IP address

In the `resolv.conf` file, add the IP address of your DNS server (replace `10.0.2.15` with the actual IP of your DNS server):

Syntax:

```
nameserver 10.0.2.15
```



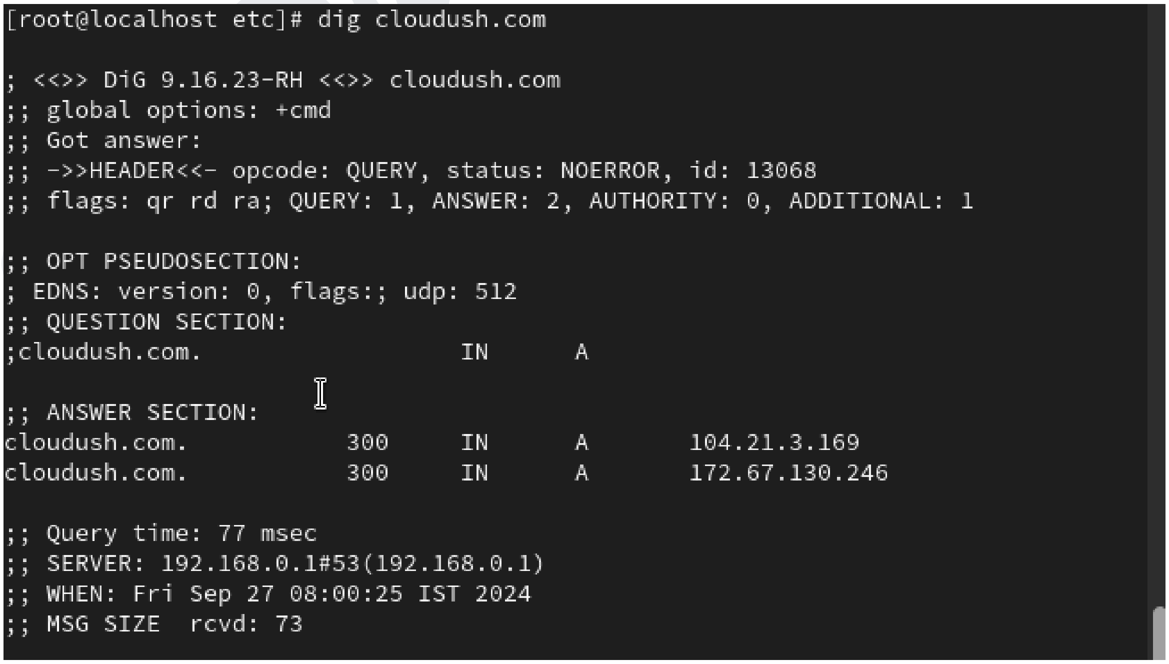
```
# Generated by NetworkManager
search localdomain cloudush.com
search Home
nameserver 10.0.2.15
nameserver 192.168.0.1
~
~
~
~
```

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 cloudush.com
```



```
[root@localhost etc]# dig cloudush.com

; <<>> DiG 9.16.23-RH <<>> cloudush.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13068
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;cloudush.com.                  IN      A

;; ANSWER SECTION:
cloudush.com.           300     IN      A       104.21.3.169
cloudush.com.           300     IN      A       172.67.130.246

;; Query time: 77 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Fri Sep 27 08:00:25 IST 2024
;; MSG SIZE  rcvd: 73
```

2. DHCP Server Configuration

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.

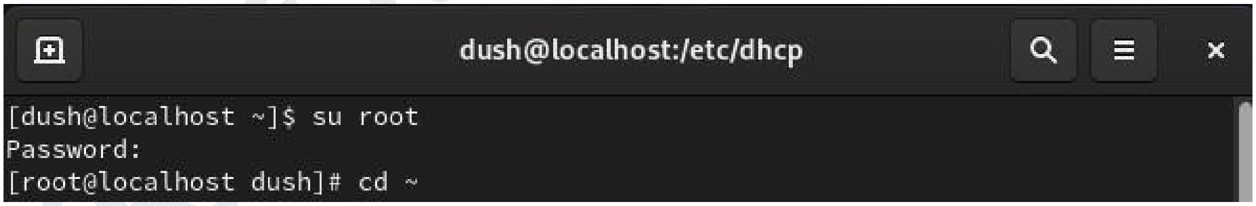
Steps

Step 1: Login into Root account

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

Syntax:

```
su root
```



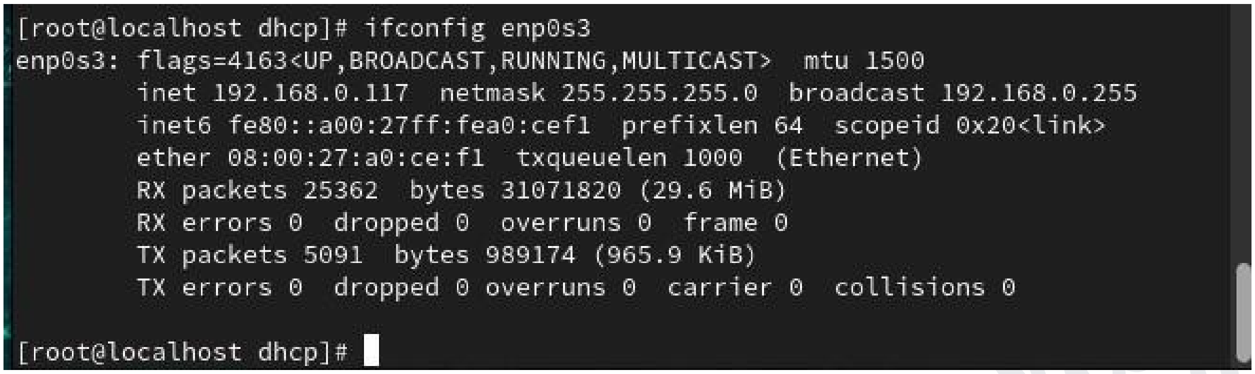
```
[dush@localhost ~]$ su root
Password:
[root@localhost dush]# cd ~
```

Step 2: Configure IP

Check the IP and other network details of wired connection.

Syntax:

```
ifconfig enp0s3
```



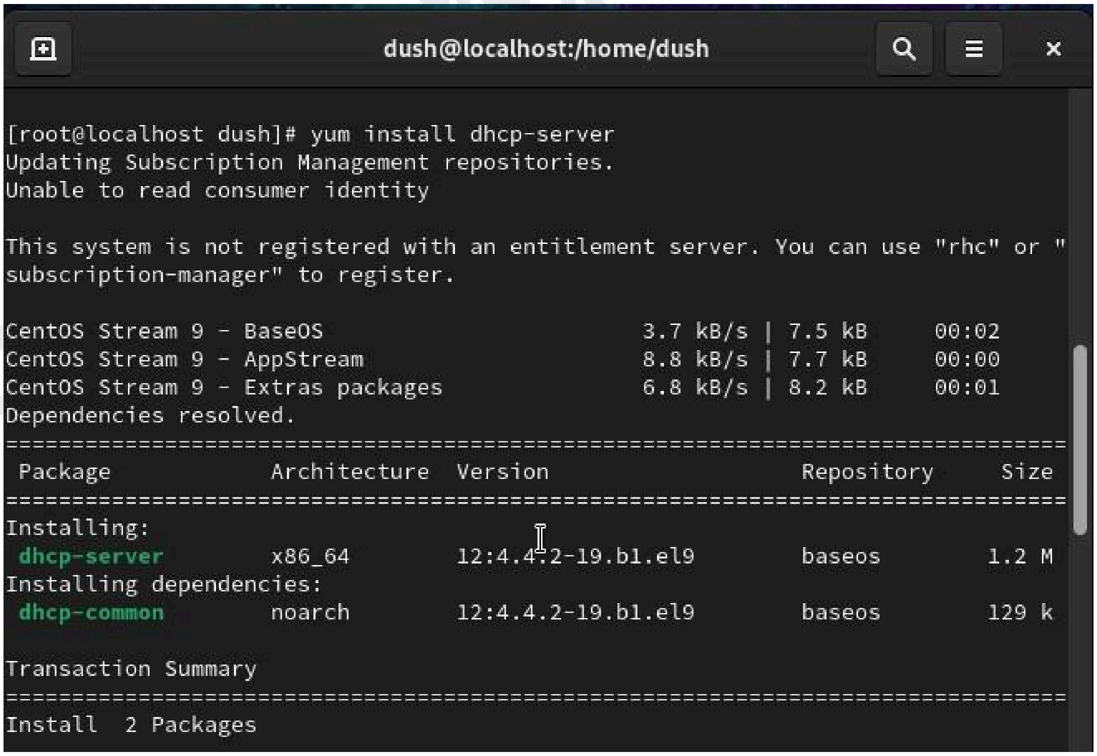
```
[root@localhost dhcp]# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.117  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fea0:cef1  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a0:ce:f1  txqueuelen 1000  (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
```

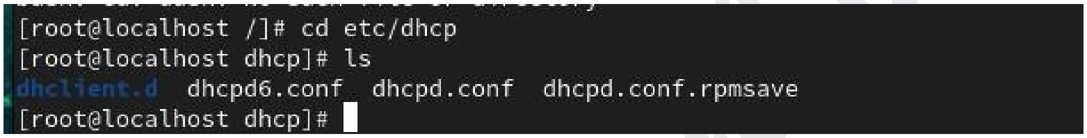
Step 4: Configure DHCP Server

1. Edit the DHCP Configuration File

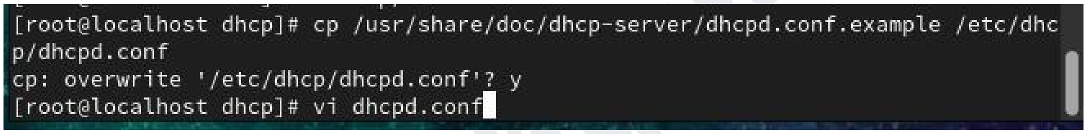
The main DHCP configuration file is `/etc/dhcp/dhcpd.conf`. Edit this file to define the network ranges and settings for your DHCP clients. Before that, copy the `dhcpd.conf.example` file to `dhcpd.conf` file for sample configuration.

Syntax:

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

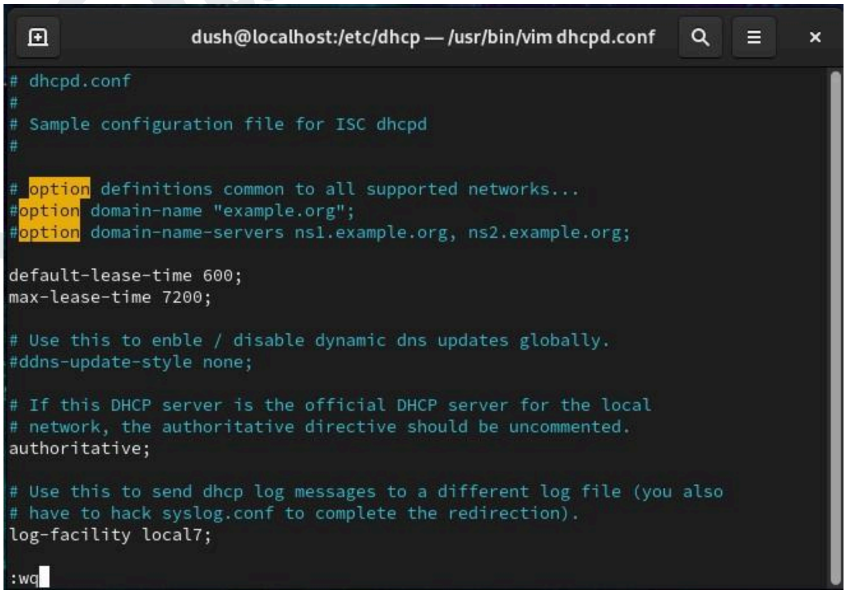


```
[root@localhost /]# cd etc/dhcp
[root@localhost dhcp]# ls
dhclient.d  dhcpd6.conf  dhcpd.conf  dhcpd.conf.rpmsave
[root@localhost dhcp]# 
```

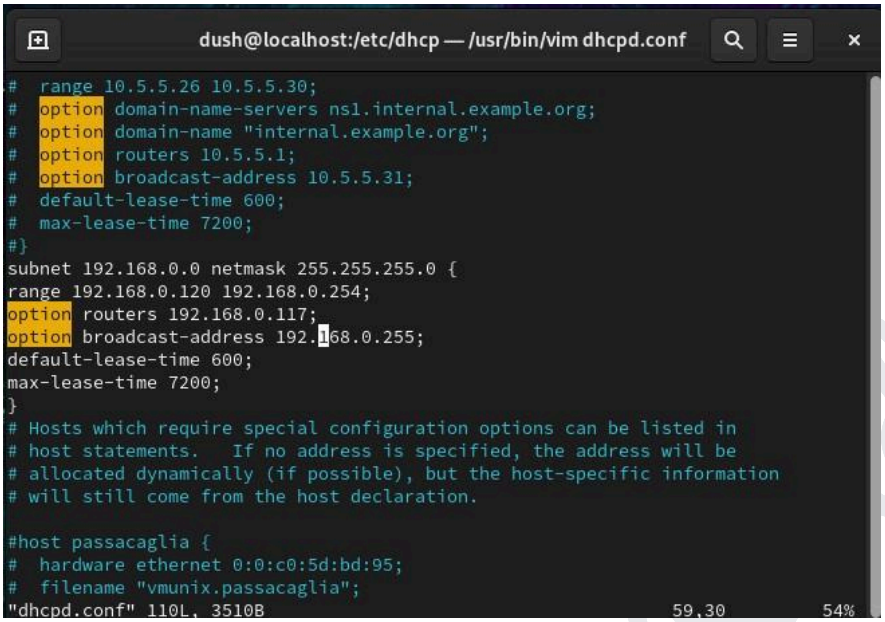


```
[root@localhost dhcp]# cp /usr/share/doc/dhcp-server/dhcpd.conf.example /etc/dhc
p/dhcpd.conf
cp: overwrite '/etc/dhcp/dhcpd.conf'? y
[root@localhost dhcp]# vi dhcpd.conf
```

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
```



dush@localhost:/etc/dhcp — /usr/bin/vim dhcpd.conf

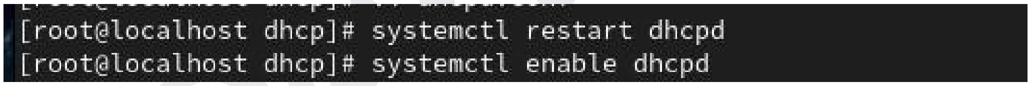
```
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}
subnet 192.168.0.0 netmask 255.255.255.0 {
range 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:

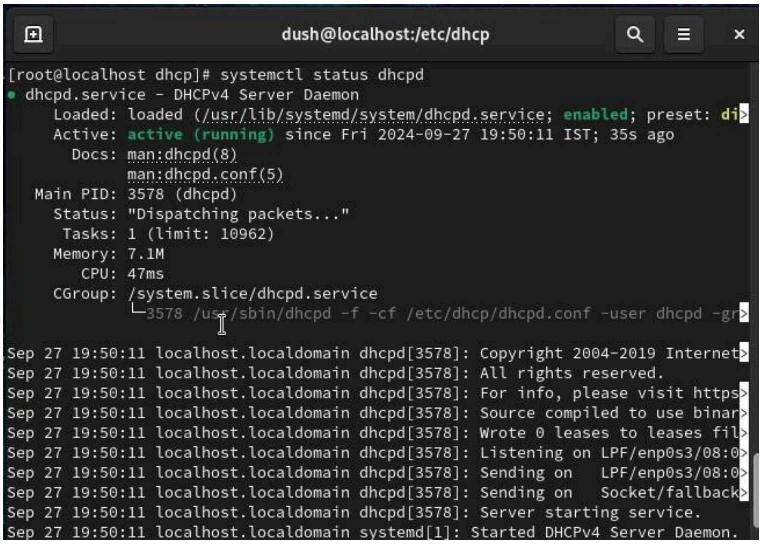
Syntax:

```
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:



dush@localhost:/etc/dhcp

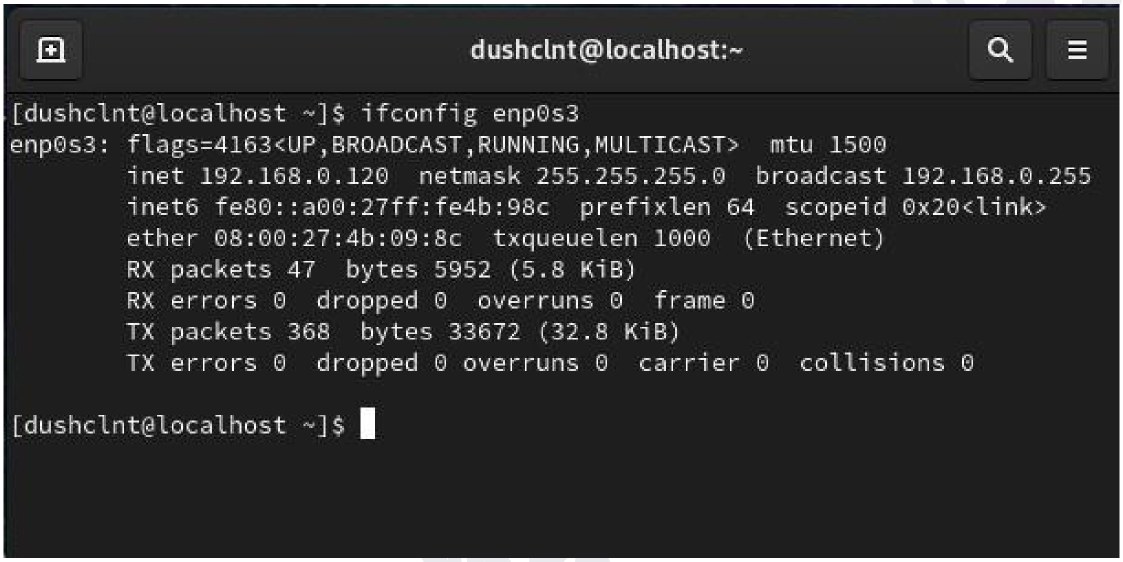
```
[root@localhost dhcp]# systemctl status dhcpd
● dhcpd.service - DHCPv4 Server Daemon
     Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; preset: di>
     Active: active (running) since Fri 2024-09-27 19:50:11 IST; 35s ago
       Docs: man:dhcpd(8)
             man:dhcpd.conf(5)
   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.

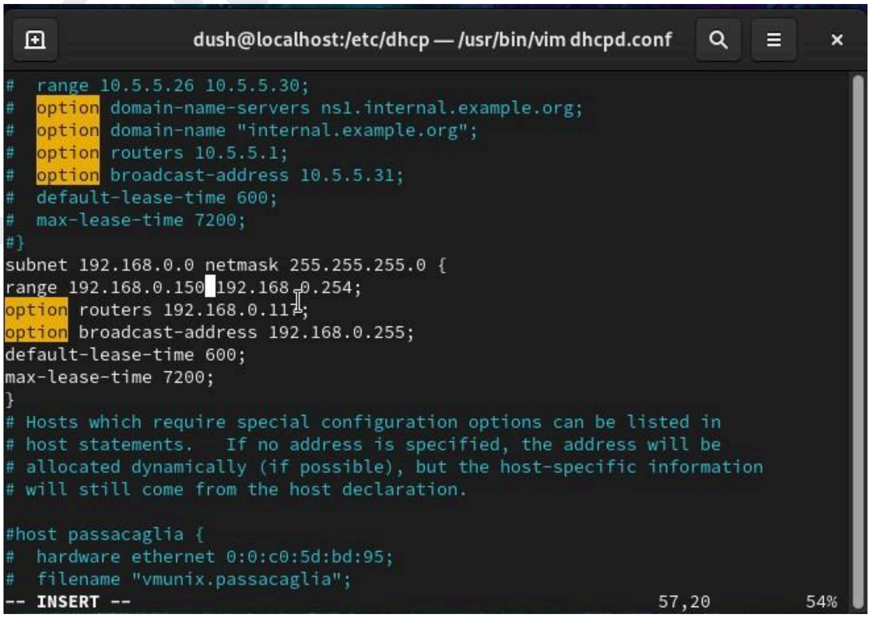
Syntax:

```
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.



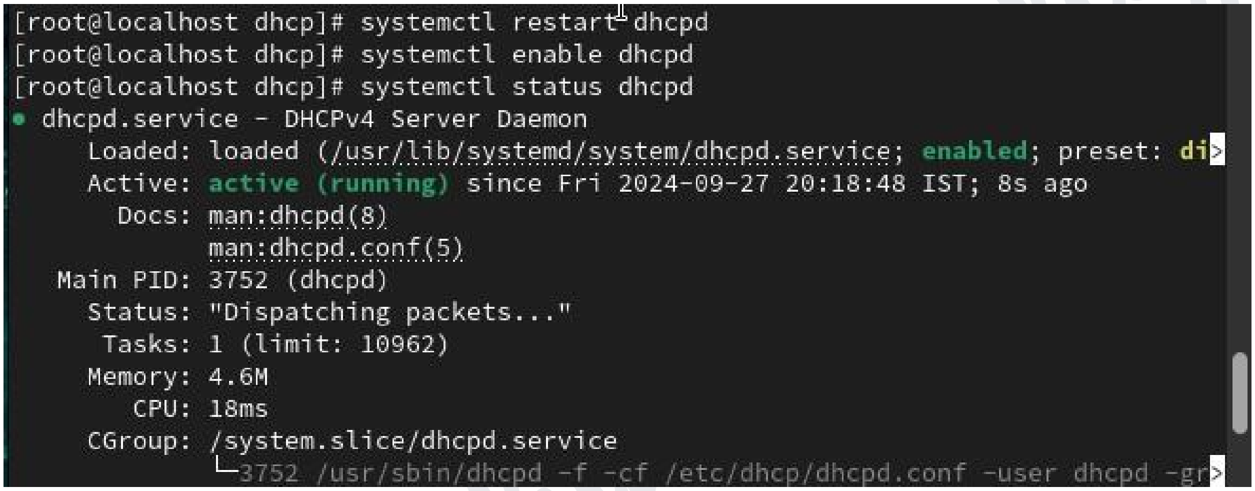
```
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}
subnet 192.168.0.0 netmask 255.255.255.0 {
range 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
```



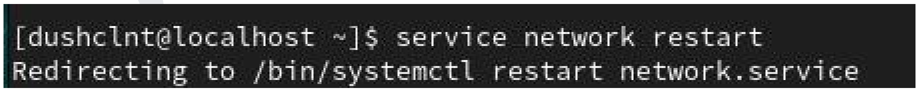
```
[root@localhost dhcp]# systemctl restart dhcpd
[root@localhost dhcp]# systemctl enable dhcpd
[root@localhost dhcp]# systemctl status dhcpd
● dhcpd.service - DHCPv4 Server Daemon
     Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; preset: di>
     Active: active (running) since Fri 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

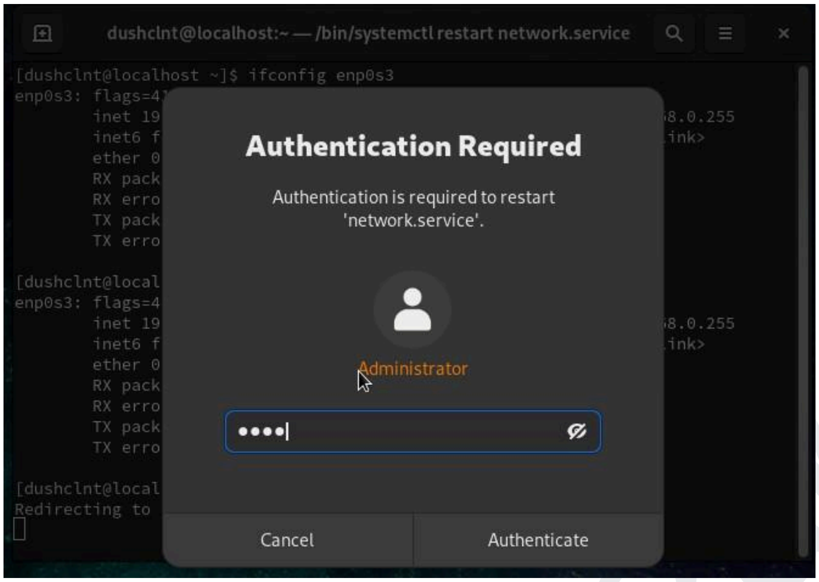
Similarly, Restart the network service to auto assign new DHCP within the range allocated and authenticate to restart it.

Syntax:

```
service network restart
```



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



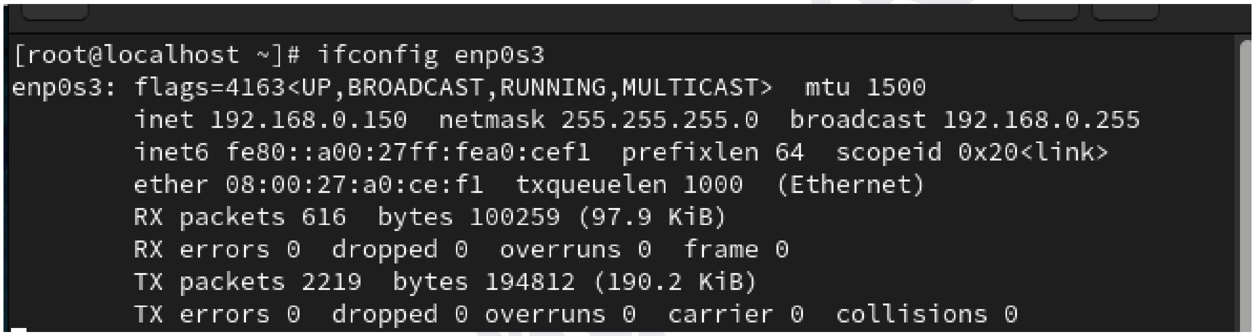
dushclnt@localhost:~ — /bin/systemctl restart network.service

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

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

[dushclnt@local
Redirecting to
```

Authentication Required

Authentication is required to restart 'network.service'.

Administrator
••••
Cancel
Authenticate



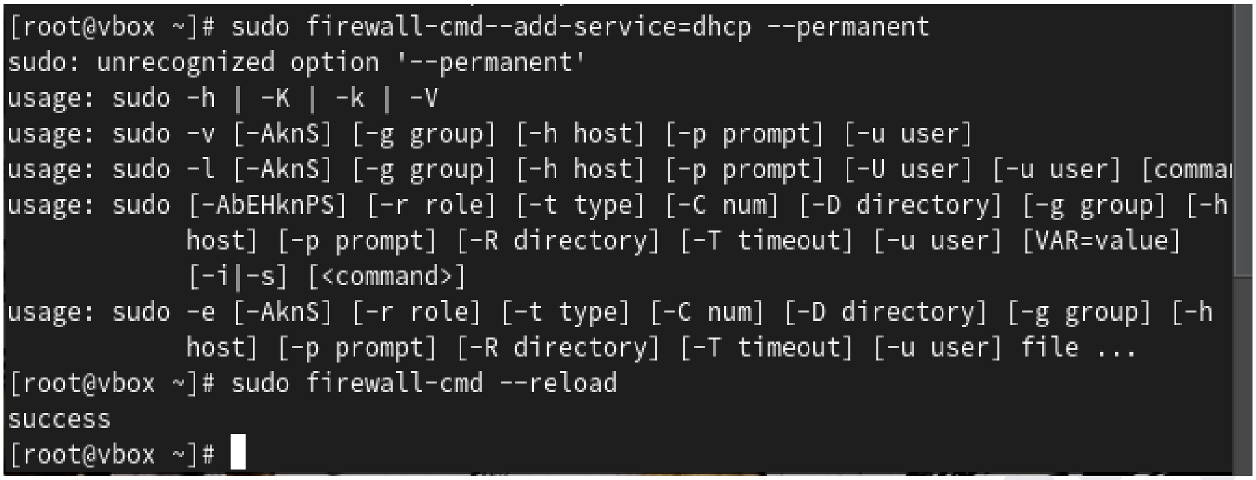
```
[root@localhost ~]# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.150  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fea0:cef1  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a0:ce:f1  txqueuelen 1000  (Ethernet)
        RX packets 616  bytes 100259 (97.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2219  bytes 194812 (190.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

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:

```
firewall-cmd --add-service=dhcp --permanent
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] [comma
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.

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.