



Instituto Politécnico Nacional
Escuela Superior de Cómputo



PSA 4.SERVIDOR DE DIRECCIONES IP, DHCP

Grupo: 7CV1

Alumnos:

- **Díaz Ortiz Brandon Aldair**
- **García Cárdenas Ángel Alberto**
- **Gutiérrez Pérez Lizbeth Alejandra**
 - **Nava Izquierdo César**

Unidad de Aprendizaje:

- **ADMINISTRACIÓN DE SERVICIOS EN RED**

Profesor:

- **MANUEL ALEJANDRO SOTO RAMOS**

Fecha:13/12/2024

Índice

Índice	2
Introducción	3
Desarrollo	4
1. Instalar el Paquete isc-dhcp-server	4
2. Configurar IP Estática en el Servidor	4
3. Configuración del Servidor DHCP	6
4. Iniciar y Verificar el Servicio	9
5. Pruebas de Funcionamiento	10
6. Construir Herramienta de Monitoreo	11
Conclusión	14

Introducción

En el siguiente reporte se realizará los siguientes puntos en un nuevo servidor:

Instalar el Paquete isc-dhcp-server: Comprenderán cómo instalar el software necesario para configurar un servidor DHCP en Debian.

1. Configurar el Servidor DHCP
2. Configurarán la interfaz de red para que el servidor DHCP escuche en la red correcta.
3. Administrar el Servicio DHCP
4. Iniciarán el servicio DHCP y lo configurarán para que se ejecute bajo las siguientes condiciones.
 1. Red clase A. Máximo 15 usuarios en un segmento, con un conjunto de MAC-ADD conocidas.
 2. Red clase B. con un máximo de 20 usuarios, tiempo de arrendamiento de 2 minutos y máximo 10 minutos
 3. Red clase B. Con usuarios asignados a una IP única con vbase en la MAC-ADDR.
 4. Red Clase C. 20 usuarios máximo, únicamente usuarios con sistema operativo Windows
 5. Red clase C. 20 Usuarios máximo, únicamente dispositivos móviles.
 6. Red Clase C. 30 usuarios máximo, únicamente dispositivos con sistema operativo tipo unix
 7. En todos los casos mencionados se debe configurar (IP, máscara de red, puerta de enlace predeterminada, servidor de nombre de dominio)
5. Verificar el estado del servicio para asegurarse de que esté funcionando correctamente.
6. Utilizar archivos de registro y comandos de diagnóstico para identificar y resolver problemas comunes en la configuración del servidor DHCP.
7. Realizar Pruebas de Funcionamiento en cada caso (Prueba de conexión, bitácora y MACADD de dispositivo)
8. Conectar clientes a la red y verificarán que reciban direcciones IP dentro del rango configurado.
9. Forzar la renovación de direcciones IP en clientes para comprobar el correcto funcionamiento del servidor DHCP.
10. Construir una herramienta (cualquier lenguaje de programación) que permita mostrar los siguientes datos:

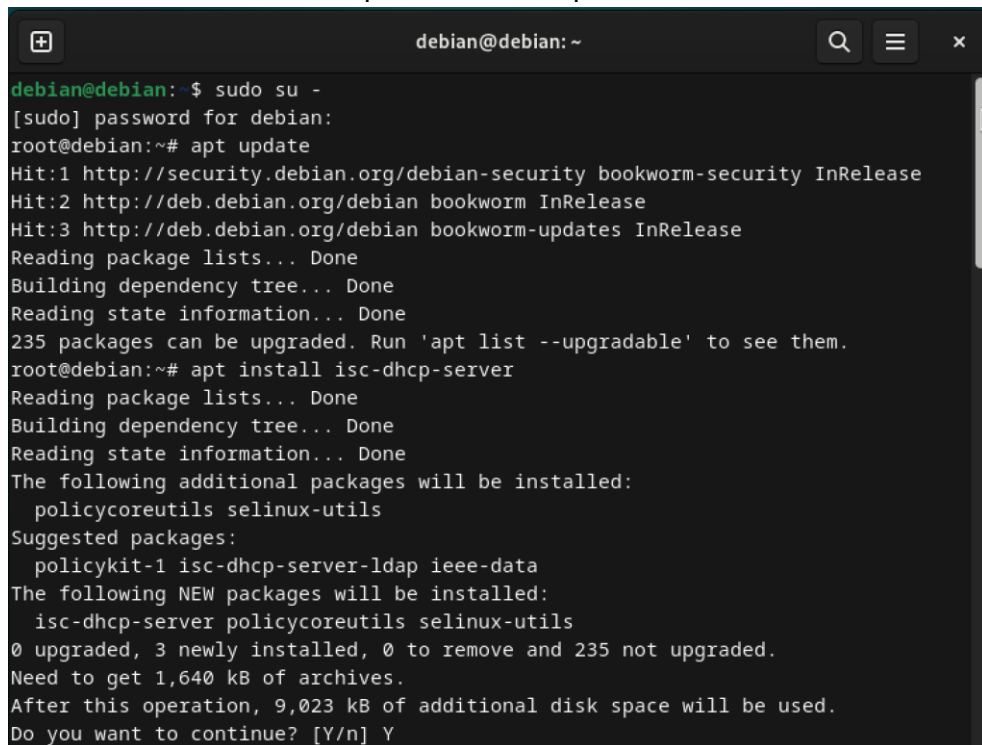
1. Clase de Red
2. Número de ip disponibles/utilizadas
3. Ip asignada- MAcADDR asociada-Tiempo de arrendamiento restante antes de su caducidad.
4. Estado del arrendamiento

Desarrollo

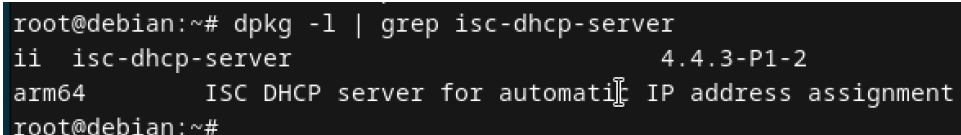
1. Instalar el Paquete isc-dhcp-server

1. Cambiar a usuario administrador
sudo su -
2. Actualiza los repositorios e instala el servidor DHCP:
apt update

apt install isc-dhcp-server

A terminal window titled 'debian@debian: ~' showing the installation process. The user runs 'sudo su -' to become root. Then 'apt update' is run, showing updates for bookworm-security, bookworm, and bookworm-updates. Next, 'apt install isc-dhcp-server' is run. The terminal shows that 3 new packages will be installed: isc-dhcp-server, policycoreutils, and selinux-utils. The total size of the packages is 1,640 kB. The user confirms the installation with 'Y'.

3. Verifica la instalación:

A terminal window showing the command 'dpkg -l | grep isc-dhcp-server' being run. The output shows the installed package 'isc-dhcp-server' with version '4.4.3-P1-2' and architecture 'arm64'. The description 'ISC DHCP server for automatic IP address assignment' is also visible.

2. Configurar IP Estática en el Servidor

Edita el archivo `/etc/network/interfaces`:

`sudo nano /etc/network/interfaces`

Configura la interfaz de red principal con una IP estática.

Ejemplo:

```
auto enp0s3
```

```
iface enp0s3 inet static
    address 192.168.0.50
    netmask 255.255.255.0
    gateway 192.168.0.1
    dns-nameservers 8.8.8.8 8.8.4.4
```

Si necesitas configurar más subredes, añade interfaces virtuales. Ejemplo:

```
auto enp0s3:1
```

```
iface enp0s3:1 inet static
    address 10.0.0.1
    netmask 255.0.0.0
```

```
auto enp0s3:2
```

```
iface enp0s3:2 inet static
    address 172.16.0.1
    netmask 255.255.0.0
```

```
auto enp0s3:3
```

```
iface enp0s3:3 inet static
    address 192.168.1.1
    netmask 255.255.255.0
```

```
GNU nano 7.2 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto enp0s1
iface enp0s1 inet static
    address 192.168.100.50
    netmask 255.255.255.0
    gateway 192.168.100.1
    dns-nameservers 8.8.8.8 8.8.4.4

auto enp0s1:1
iface enp0s1:1 inet static
    address 10.0.0.1
    netmask 255.0.0.0

auto enp0s1:2
iface enp0s1:2 inet static
    address 172.16.0.1
    netmask 255.255.0.0

auto enp0s1:3
iface enp0s1:3 inet static
    address 192.168.1.1
    netmask 255.255.255.0

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

Reinicia la red:

`sudo systemctl restart networking`

```
root@debian:~# systemctl restart networking
root@debian:~#
```

Verifica que las interfaces estén configuradas correctamente:

`ip addr`

```
root@debian:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether de:33:c4:0b:12:0a brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.50/24 brd 192.168.100.255 scope global enp0s1
        valid_lft forever preferred_lft forever
root@debian:~#
```

3. Configuración del Servidor DHCP

Edita el archivo de configuración del servidor DHCP:

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

Configura las redes clase A, B y C según los requisitos:

Red Clase A

```
subnet 10.0.0.0 netmask 255.0.0.0 {

    range 10.0.0.10 10.0.0.25;
    option routers 10.0.0.1;
    option subnet-mask 255.0.0.0;
    option domain-name-servers 8.8.8.8, 8.8.4.4;

    host device1 {
        hardware ethernet 00:11:22:33:44:55;
        fixed-address 10.0.0.10;
    }

    host device2 {
        hardware ethernet 66:77:88:99:AA:BB;
        fixed-address 10.0.0.11;
    }
}
```

```
}  
}
```

Red Clase B (Arrendamiento Temporal)

```
subnet 172.16.0.0 netmask 255.255.0.0 {  
    range 172.16.0.10 172.16.0.30;  
    option routers 172.16.0.1;  
    option subnet-mask 255.255.0.0;  
    option domain-name-servers 8.8.8.8, 8.8.4.4;  
    default-lease-time 120;  
    max-lease-time 600;  
}
```

Red Clase C (Windows, móviles y Unix)

plaintext

Copiar código

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.10 192.168.1.30;  
    option routers 192.168.1.1;  
    option subnet-mask 255.255.255.0;  
    option domain-name-servers 8.8.8.8, 8.8.4.4;  
  
    if substring(option vendor-class-identifier, 0, 8) = "MSFT 5.0" {  
        allow booting;  
    } else {  
        deny booting;  
    }  
}  
  
subnet 192.168.2.0 netmask 255.255.255.0 {  
    range 192.168.2.10 192.168.2.30;  
    option routers 192.168.2.1;  
    option subnet-mask 255.255.255.0;  
    option domain-name-servers 8.8.8.8, 8.8.4.4;  
}  
  
subnet 192.168.3.0 netmask 255.255.255.0 {  
    range 192.168.3.10 192.168.3.40;  
    option routers 192.168.3.1;  
    option subnet-mask 255.255.255.0;  
    option domain-name-servers 8.8.8.8, 8.8.4.4;  
  
}
```

```

#clase A
subnet 10.0.0.0 netmask 255.0.0.0 {
    range 10.0.0.10 10.0.0.25;
    option routers 10.0.0.1;
    option subnet-mask 255.0.0.0;
    option domain-name-servers 8.8.8.8, 8.8.4.4;

    host device1 {
        hardware ethernet 00:11:22:33:44:55;
        fixed-address 10.0.0.10;
    }

    host device2 {
        hardware ethernet 66:77:88:99:AA:BB;
        fixed-address 10.0.0.11;
    }
}

#clase B
subnet 172.16.0.0 netmask 255.255.0.0 {
    range 172.16.0.10 172.16.0.30;
    option routers 172.16.0.1;
    option subnet-mask 255.255.0.0;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    default-lease-time 120;
    max-lease-time 600;
}

#clase C
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.30;
}

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

Guarda y cierra el archivo.

Configura las interfaces para el servidor DHCP:

`sudo nano /etc/default/isc-dhcp-server`

Agrega las interfaces donde el servidor escuchará:

`INTERFACESv4="enp0s3"`


```
GNU nano 7.2 /etc/default/isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s1"
INTERFACESv6=""

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
               ^_          ^_           ^_           ^_           ^_ Location
               ^_          ^_           ^_           ^_           ^_ Go To Line
               ^_          ^_           ^_           ^_           ^_ M-U Undo
               ^_          ^_           ^_           ^_           ^_ M-E Redo
```

4. Iniciar y Verificar el Servicio

Reinicia el servicio:

```
sudo systemctl restart isc-dhcp-server
```

Verifica su estado:

```
sudo systemctl status isc-dhcp-server
```

Consulta los logs:

```
sudo journalctl -xeu isc-dhcp-server
```

```
root@debian:~# systemctl restart isc-dhcp-server
root@debian:~# systemctl status isc-dhcp-server
● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Thu 2025-01-16 10:51:36 PST; 3s ago
     Docs: man:systemd-sys-generator(8)
  Process: 2080 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, sta>
    Tasks: 1 (limit: 4568)
   Memory: 4.7M
      CPU: 58ms
   CGroup: /system.slice/isc-dhcp-server.service
           └─2093 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf enp0s1 enp0s>

Jan 16 10:51:34 debian dhcpd[2093]: DHCPDISCOVER from cc:08:fa:63:0f:fb (Air-de>
Jan 16 10:51:35 debian dhcpd[2093]: DHCPREQUEST for 172.16.0.28 (172.16.0.1) fr>
Jan 16 10:51:35 debian dhcpd[2093]: DHCPREQUEST for 172.16.0.28 (172.16.0.1) fr>
Jan 16 10:51:35 debian dhcpd[2093]: DHCPACK on 172.16.0.28 to cc:08:fa:63:0f:fb>
Jan 16 10:51:35 debian dhcpd[2093]: DHCPREQUEST for 172.16.0.28 (172.16.0.1) fr>
Jan 16 10:51:35 debian dhcpd[2093]: DHCPREQUEST for 172.16.0.28 (172.16.0.1) fr>
Jan 16 10:51:35 debian dhcpd[2093]: DHCPPOFFER on 10.0.0.20 to cc:08:fa:63:0f:fb>
Jan 16 10:51:35 debian dhcpd[2093]: DHCPPOFFER on 192.168.100.70 to cc:08:fa:63:>
Jan 16 10:51:36 debian isc-dhcp-server[2080]: Starting ISC DHCPv4 server: dhcpd.
Jan 16 10:51:36 debian systemd[1]: Started isc-dhcp-server.service - LSB: DHCP >

lines 1-21/21 (END)
```

5. Pruebas de Funcionamiento

1. Conecta dispositivos a cada red (clase A, B y C).

Verifica que los dispositivos reciban direcciones IP correctas:

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

```
root@debian:~# cat /var/lib/dhcp/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.4.3-P1

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

lease 192.168.100.67 {
    starts 4 2025/01/16 00:21:41;
    ends 4 2025/01/16 00:23:41;
    tstp 4 2025/01/16 00:23:41;
    cltt 4 2025/01/16 00:21:41;
    binding state free;
    hardware ethernet ee:37:c6:c9:6c:3e;
    uid "\001\3567\306\311l>";
    set vendor-class-identifier = "android-dhcp-13";
    client-hostname "realme-GT-Master-Edition";
}

lease 192.168.100.79 {
    starts 4 2025/01/16 00:21:43;
    ends 4 2025/01/16 00:23:43;
    tstp 4 2025/01/16 00:23:43;
    cltt 4 2025/01/16 00:21:43;
    binding state free;
    hardware ethernet 82:c3:a5:55:a1:d1;
    uid "\001\202\303\245U\241\321";
    set vendor-class-identifier = "android-dhcp-14";
```

Esta es una herramienta que

Fuerza la renovación de IP en los dispositivos:

En

Linux:

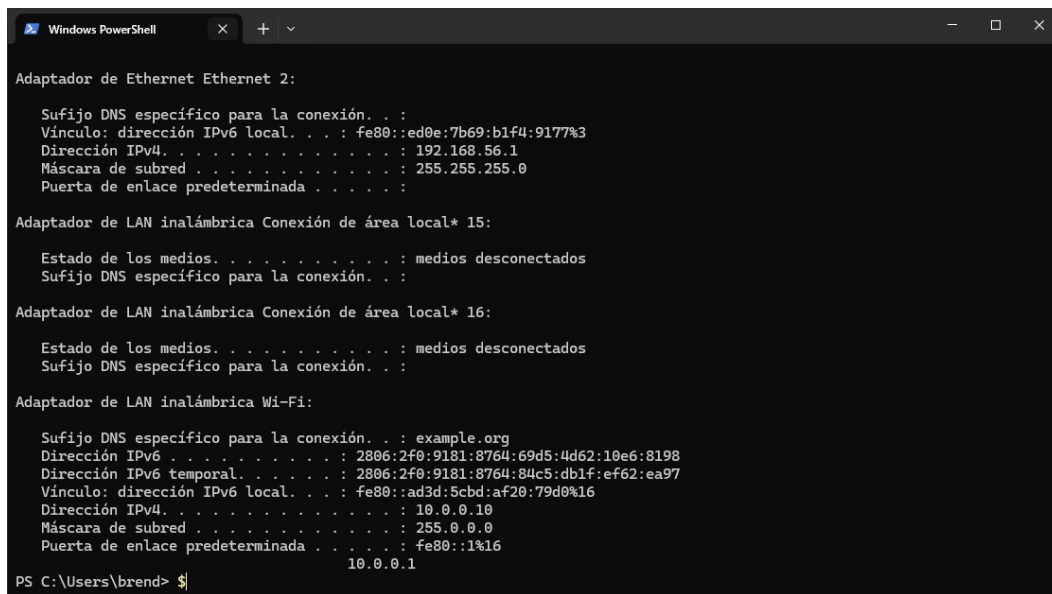
```
sudo dhclient -r && sudo dhclient
```

En

Windows:

```
ipconfig /release
```

```
ipconfig /renew
```



```
Windows PowerShell

Adaptador de Ethernet Ethernet 2:

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::ed0e:7b69:b1f4:9177%3
Dirección IPv4. . . . . : 192.168.56.1
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada. . . . . :

Adaptador de LAN inalámbrica Conexión de área local* 15:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 16:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . . : example.org
Dirección IPv6. . . . . : 2806:2f0:9181:8764:69d5:4d62:10e6:8198
Dirección IPv6 temporal. . . . . : 2806:2f0:9181:8764:84c5:db1f:ef62:ea97
Vínculo: dirección IPv6 local. . . : fe80::ad3d:5cbd:af20:79d0%16
Dirección IPv4. . . . . : 10.0.0.10
Máscara de subred. . . . . : 255.0.0.0
Puerta de enlace predeterminada. . . . . : fe80::1%16
10.0.0.1

PS C:\Users\brend> $|
```

6. Construir Herramienta de Monitoreo

Usa Python para crear una herramienta que analice los arrendamientos:

python

Copiar código

```
import re
```

```
def parse_leases(file_path):
```

```
    leases = []
```

```
    with open(file_path, 'r') as file:
```

```
        data = file.read()
```

```

    matches
    re.findall(r"lease\s([\d\.]+)\s\{.*?hardware\s(ethernet\s([\w:]+);.*?ends\s\d+\s\d+\V\d+\V\d+\s(\d+:\d+:\d+);", data, re.DOTALL)

    for match in matches:

        leases.append({

            "IP": match[0],

            "MAC": match[1],

            "Lease End": match[2]

        })

    return leases

leases = parse_leases("/var/lib/dhcp/dhcpd.leases")

for lease in leases:

    print(f"IP: {lease['IP']}, MAC: {lease['MAC']}, Lease Ends: {lease['Lease End']}")

```

```

root@debian:~# python3 monitero.py
=== Class C ===
Total IPs: 32
Used IPs: 2
Available IPs: 30
Leases:
IP: 192.168.100.67, MAC: ee:37:c6:c9:6c:3e, Lease Ends: 00:23:43, Time Remaining: 12:55:43.216093, State: free
IP: 192.168.100.68, MAC: b8:1e:a4:fa:e2:b3, Lease Ends: 00:23:56, Time Remaining: 12:55:56.212940, State: free
IP: 192.168.100.89, MAC: 44:b2:95:ed:79:92, Lease Ends: 00:26:34, Time Remaining: 12:58:34.212905, State: free
IP: 192.168.100.98, MAC: d6:bf:20:21:4a:cb, Lease Ends: 00:26:37, Time Remaining: 12:58:37.212895, State: free
IP: 192.168.100.74, MAC: 42:6b:9f:ca:aa:c2, Lease Ends: 00:26:51, Time Remaining: 12:58:51.212887, State: free
IP: 192.168.100.94, MAC: 4a:0b:8f:75:84:44, Lease Ends: 00:27:12, Time Remaining: 12:59:12.212880, State: free
IP: 192.168.100.63, MAC: be:38:d1:54:b5:20, Lease Ends: 00:27:19, Time Remaining: 12:59:19.212868, State: free
IP: 192.168.100.82, MAC: ce:de:4d:92:8b:f3, Lease Ends: 00:27:30, Time Remaining: 12:59:30.212859, State: free
IP: 192.168.100.97, MAC: 02:94:cd:10:0c:2c, Lease Ends: 00:27:33, Time Remaining: 12:59:33.212849, State: free
IP: 192.168.100.62, MAC: 52:e8:cf:ba:8c:ad, Lease Ends: 00:27:53, Time Remaining: 12:59:53.212837, State: free
IP: 192.168.100.100, MAC: 72:91:b6:05:10:ba, Lease Ends: 00:28:02, Time Remaining: 13:00:02.212830, State: free
IP: 192.168.100.61, MAC: de:c6:77:00:54:2f, Lease Ends: 00:28:07, Time Remaining: 13:00:07.212809, State: free
IP: 192.168.100.87, MAC: ea:e8:6f:cb:23:45, Lease Ends: 00:28:25, Time Remaining: 13:00:25.212801, State: free
IP: 192.168.100.88, MAC: 1e:8b:2e:20:4c:c2, Lease Ends: 00:28:32, Time Remaining: 13:00:32.212795, State: free
IP: 192.168.100.83, MAC: 16:b5:68:8f:70:5e, Lease Ends: 00:28:34, Time Remaining: 13:00:34.212790, State: free
IP: 192.168.100.84, MAC: a2:f2:90:4a:02:3d, Lease Ends: 00:28:35, Time Remaining: 13:00:35.212785, State: free
IP: 192.168.100.91, MAC: ac:74:b1:0e:58:80, Lease Ends: 00:28:38, Time Remaining: 13:00:38.212779, State: free
IP: 192.168.100.76, MAC: 5a:9a:b6:b4:8e:1a, Lease Ends: 00:31:47, Time Remaining: 13:03:47.212774, State: free
IP: 192.168.100.77, MAC: 5a:53:9b:87:8d:5f, Lease Ends: 17:31:57, Time Remaining: 6:03:57.212768, State: free
IP: 192.168.100.64, MAC: fa:80:a7:19:13:0a, Lease Ends: 18:53:29, Time Remaining: 7:25:29.212763, State: active
IP: 192.168.100.66, MAC: 16:62:e1:28:26:f5, Lease Ends: 23:58:59, Time Remaining: 12:30:59.212757, State: free
IP: 192.168.1.10, MAC: b0:3c:dc:e9:2d:b6, Lease Ends: 00:01:35, Time Remaining: 12:33:35.212752, State: free
IP: 192.168.1.13, MAC: 0c:96:e6:c3:d7:c9, Lease Ends: 00:05:13, Time Remaining: 12:37:13.212746, State: free
IP: 192.168.1.14, MAC: dc:21:5c:d2:81:74, Lease Ends: 00:06:23, Time Remaining: 12:38:23.212741, State: free
IP: 192.168.1.27, MAC: d0:12:55:99:e6:13, Lease Ends: 00:07:04, Time Remaining: 12:39:04.212736, State: free
IP: 192.168.1.21, MAC: 4c:5f:70:35:79:0b, Lease Ends: 00:07:15, Time Remaining: 12:39:15.212730, State: free
IP: 192.168.1.16, MAC: 94:08:53:96:66:1d, Lease Ends: 00:08:27, Time Remaining: 12:40:27.212725, State: free

```

```

IP: 172.16.0.27, MAC: 08:00:27:63:87:a0, Lease Ends: 02:20:21, Time Remaining: 14:52:21.212683, State: free
IP: 172.16.0.14, MAC: 56:ae:86:33:8b:bb, Lease Ends: 02:20:23, Time Remaining: 14:52:23.212677, State: free
IP: 172.16.0.15, MAC: ba:35:9b:47:fa:7c, Lease Ends: 02:20:43, Time Remaining: 14:52:43.212672, State: free
IP: 172.16.0.23, MAC: de:c6:d7:00:54:2f, Lease Ends: 02:20:57, Time Remaining: 14:52:57.212667, State: free
IP: 172.16.0.12, MAC: 4e:4b:bf:90:ca:66, Lease Ends: 02:21:17, Time Remaining: 14:53:17.212662, State: free
IP: 172.16.0.13, MAC: 50:98:39:a1:36:42, Lease Ends: 02:22:49, Time Remaining: 14:54:49.212656, State: free
IP: 172.16.0.30, MAC: de:33:c4:0b:12:0a, Lease Ends: 06:54:23, Time Remaining: 19:26:23.212651, State: free
IP: 172.16.0.28, MAC: cc:08:fa:63:0f:fb, Lease Ends: 02:19:49, Time Remaining: 14:51:49.212646, State: free
IP: 172.16.0.18, MAC: a2:59:07:0f:a4:6a, Lease Ends: 19:07:43, Time Remaining: 7:39:43.212591, State: active
IP: 172.16.0.28, MAC: cc:08:fa:63:0f:fb, Lease Ends: 19:13:41, Time Remaining: 7:45:41.212586, State: active
IP: 172.16.0.28, MAC: cc:08:fa:63:0f:fb, Lease Ends: 19:16:54, Time Remaining: 7:48:54.212581, State: active
IP: 172.16.0.28, MAC: cc:08:fa:63:0f:fb, Lease Ends: 19:21:48, Time Remaining: 7:53:48.212576, State: active
IP: 172.16.0.28, MAC: cc:08:fa:63:0f:fb, Lease Ends: 19:27:14, Time Remaining: 7:59:14.212571, State: active
IP: 172.16.0.28, MAC: cc:08:fa:63:0f:fb, Lease Ends: 19:34:05, Time Remaining: 8:06:05.212566, State: active
IP: 172.16.0.18, MAC: a2:59:07:0f:a4:6a, Lease Ends: 21:24:40, Time Remaining: 9:56:40.212561, State: active

=== Class A ===
Total IPs: 8
Used IPs: 1
Available IPs: 7
Leases:
IP: 10.0.0.23, MAC: 2e:16:a7:e2:eb:ed, Lease Ends: 02:19:59, Time Remaining: 14:51:59.212640, State: free
IP: 10.0.0.16, MAC: fa:19:6f:17:6a:58, Lease Ends: 02:20:13, Time Remaining: 14:52:13.212635, State: free
IP: 10.0.0.14, MAC: 42:40:73:e0:a8:71, Lease Ends: 02:20:17, Time Remaining: 14:52:17.212630, State: free
IP: 10.0.0.18, MAC: 7e:c3:21:06:d9:a2, Lease Ends: 02:20:21, Time Remaining: 14:52:21.212625, State: free
IP: 10.0.0.10, MAC: 08:00:27:63:87:a0, Lease Ends: 02:20:22, Time Remaining: 14:52:22.212620, State: free
IP: 10.0.0.12, MAC: 8a:00:e7:89:61:87, Lease Ends: 02:24:55, Time Remaining: 14:56:55.212614, State: free
IP: 10.0.0.15, MAC: f6:ab:de:23:c9:c8, Lease Ends: 04:14:42, Time Remaining: 16:46:42.212609, State: free
IP: 10.0.0.20, MAC: cc:08:fa:63:0f:fb, Lease Ends: 18:56:35, Time Remaining: 7:28:35.212602, State: active

root@debian:~#

```

Este programa es una herramienta desarrollada en Python cuyo propósito es analizar y monitorear el estado de las direcciones IP asignadas por un servidor DHCP, utilizando el archivo de arrendamientos (dhcpd.leases) generado por dicho servidor. Este archivo contiene información detallada sobre las direcciones IP otorgadas a los dispositivos conectados a la red.

El programa se encarga de procesar este archivo para clasificar las direcciones IP según las clases de red (A, B y C). Dentro de cada clase, calcula cuántas direcciones están actualmente asignadas (ocupadas), cuántas están disponibles y cuántas existen en total. Adicionalmente, obtiene información detallada de cada arrendamiento, incluyendo la dirección IP asignada, la dirección MAC asociada al dispositivo, la hora en que expira el arrendamiento y el tiempo restante antes de la expiración.

El proceso inicia leyendo el archivo de arrendamientos, del cual se extraen datos clave mediante el uso de expresiones regulares. Luego, las direcciones IP se organizan y analizan según su clase de red, lo que permite una visualización clara y segmentada de los recursos disponibles. Además, el programa calcula el tiempo restante para cada arrendamiento comparando la hora actual del sistema con la hora de expiración registrada.

Finalmente, genera un reporte detallado por cada clase de red. Este informe muestra estadísticas generales, como el número total de direcciones, las ocupadas y las disponibles, y detalla los arrendamientos activos y sus respectivos tiempos restantes.

Este programa es una herramienta valiosa para los administradores de red, ya que facilita el monitoreo de las direcciones IP utilizadas en la red, la gestión de recursos disponibles y la identificación de posibles problemas como la falta de direcciones IP o la presencia de

dispositivos no autorizados. Su diseño permite visualizar el estado de la red de forma clara y precisa, proporcionando información esencial para una gestión eficiente.

Conclusión

La práctica realizada permitió explorar y aplicar los conceptos fundamentales relacionados con la configuración y administración de servidores DHCP, así como el desarrollo de herramientas de monitoreo para evaluar su desempeño. A través de esta experiencia, se lograron configurar redes de diferentes clases (A, B y C) con reglas específicas, garantizando la asignación eficiente de direcciones IP y el cumplimiento de requerimientos particulares, como la asignación de IPs estáticas basadas en direcciones MAC y la imposición de restricciones por tipo de dispositivo o sistema operativo.

Además, se analizó en detalle el archivo de arrendamientos del servidor DHCP (`dhcpd.leases`), lo que permitió interpretar los estados de las direcciones IP (activas, libres) y sus implicaciones en la disponibilidad de recursos de red. Este análisis fue clave para identificar y resolver problemas relacionados con la superposición de redes, conflictos de arrendamientos y la correcta implementación de las políticas establecidas.

Como parte de la práctica, se desarrolló una herramienta en Python capaz de monitorear en tiempo real el estado del servidor DHCP. Esta herramienta presentó de manera clara y estructurada información relevante, como las direcciones IP asignadas, las direcciones MAC correspondientes, el tiempo restante de arrendamiento y el número de direcciones disponibles y utilizadas. Este desarrollo no solo facilitó la supervisión del servidor, sino que también destacó la importancia de contar con soluciones personalizadas para la gestión proactiva de redes.

La práctica permitió, además, comprender la relevancia de un servidor DHCP bien configurado para el óptimo funcionamiento de una red, asegurando una asignación ordenada de recursos y evitando problemas de saturación o conflictos. En conjunto, las tareas realizadas demostraron cómo la combinación de una correcta configuración, análisis detallado y herramientas de monitoreo puede garantizar la estabilidad, eficiencia y seguridad en una infraestructura de red.

En conclusión, esta experiencia no solo consolidó conocimientos teóricos, sino que también evidenció su aplicabilidad en escenarios reales, proporcionando una base sólida para enfrentar retos más complejos en la administración de redes.