Project DHCP Server

DHCP PRACTICE A – COMPLETE DOCUMENTATION
Author: Álvaro Mateos Gálvez & Raúl Ibáñez Sola
Date: October 2025
System: Debian + Vagrant

1. OBJECTIVE

The goal of this practice is to learn how to configure a DHCP (Dynamic Host Configuration Protocol) server on a Linux machine (Debian) and verify that client machines automatically obtain their network configuration (IP address, gateway, DNS, etc.) from it.

We will use **Vagrant** to automate the creation of virtual machines, and **VirtualBox** as the virtualization software.

At the end of the practice:

- Client c1 will obtain an IP dynamically.
- Client c2 will have a fixed IP assigned based on its MAC address.

2. WHAT IS DHCP?

DHCP is a network protocol used to automatically assign IP addresses to devices in a network. This avoids having to manually configure each computer.

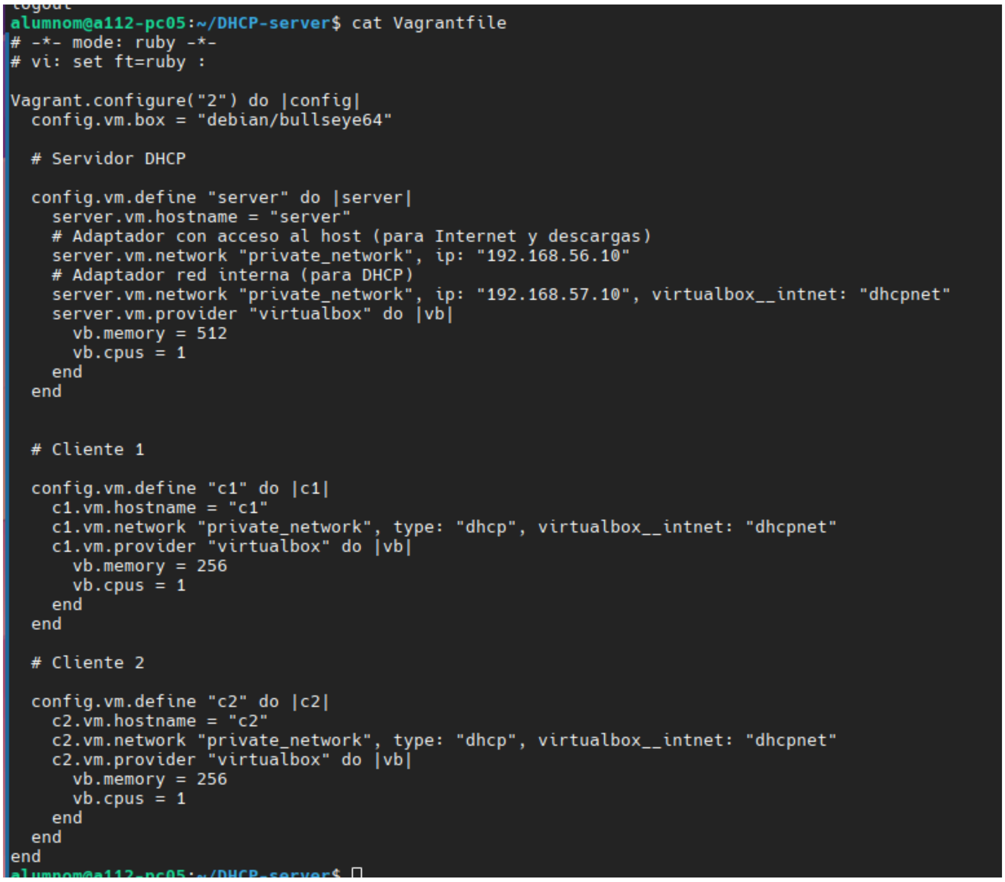
Basic DHCP steps:

1. The client sends a “DHCPDISCOVER” message to find a DHCP server.
2. The server responds with a “DHCPOFFER” (an available IP).
3. The client requests that IP with “DHCPREQUEST”.
4. The server confirms it with “DHCPACK”.

3. INITIAL SETUP WITH VAGRANT

We use Vagrant to easily create and manage virtual machines from a configuration file called Vagrantfile.

To start, create a folder (for example: dhcp-practice-a) and inside it, create the file "Vagrantfile"
with the following content:



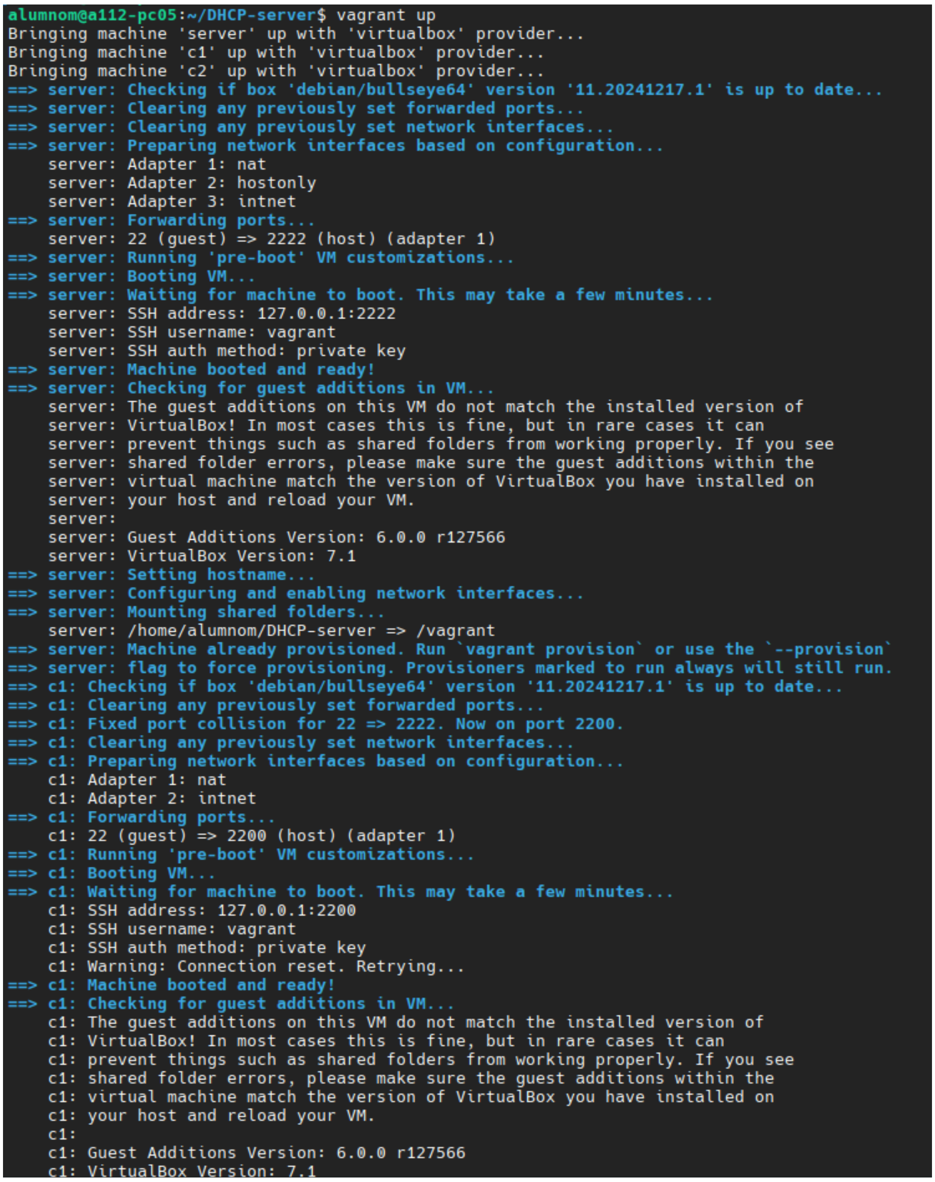
```
logout
alumnom@a112-pc05:~/DHCP-server$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.box = "debian/bullseye64"

  # Servidor DHCP

  config.vm.define "server" do |server|
    server.vm.hostname = "server"
    # Adaptador con acceso al host (para Internet y descargas)
    server.vm.network "private_network", ip: "192.168.56.10"
    # Adaptador red interna (para DHCP)
    server.vm.network "private_network", ip: "192.168.57.10", virtualbox__intnet: "dhcpnet"
    server.vm.provider "virtualbox" do |vb|
      vb.memory = 512
      vb.cpus = 1
    end
  end


  # Cliente 1

  config.vm.define "c1" do |c1|
    c1.vm.hostname = "c1"
    c1.vm.network "private_network", type: "dhcp", virtualbox__intnet: "dhcpnet"
    c1.vm.provider "virtualbox" do |vb|
      vb.memory = 256
      vb.cpus = 1
    end
  end

  # Cliente 2

  config.vm.define "c2" do |c2|
    c2.vm.hostname = "c2"
    c2.vm.network "private_network", type: "dhcp", virtualbox__intnet: "dhcpnet"
    c2.vm.provider "virtualbox" do |vb|
      vb.memory = 256
      vb.cpus = 1
    end
  end
end
alumnom@a112-pc05:~/DHCP-server$
```

Then, open a terminal in that folder and run:

```
vagrant up
```

This command tells Vagrant to download the Debian image (if it’s not already cached) and create three machines:

- server (DHCP server)
- c1 (client)
- c2 (client)



```
alumnom@a112-pc05:~/DHCP-server$ vagrant up
Bringing machine 'server' up with 'virtualbox' provider...
Bringing machine 'c1' up with 'virtualbox' provider...
Bringing machine 'c2' up with 'virtualbox' provider...
==> server: Checking if box 'debian/bullseye64' version '11.20241217.1' is up to date...
==> server: Clearing any previously set forwarded ports...
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
    server: Adapter 1: nat
    server: Adapter 2: hostonly
    server: Adapter 3: intnet
==> server: Forwarding ports...
    server: 22 (guest) => 2222 (host) (adapter 1)
==> server: Running 'pre-boot' VM customizations...
==> server: Booting VM...
==> server: Waiting for machine to boot. This may take a few minutes...
    server: SSH address: 127.0.0.1:2222
    server: SSH username: vagrant
    server: SSH auth method: private key
==> server: Machine booted and ready!
==> server: Checking for guest additions in VM...
    server: The guest additions on this VM do not match the installed version of
    server: VirtualBox! In most cases this is fine, but in rare cases it can
    server: prevent things such as shared folders from working properly. If you see
    server: shared folder errors, please make sure the guest additions within the
    server: virtual machine match the version of VirtualBox you have installed on
    server: your host and reload your VM.
    server:
    server: Guest Additions Version: 6.0.0 r127566
    server: VirtualBox Version: 7.1
==> server: Setting hostname...
==> server: Configuring and enabling network interfaces...
==> server: Mounting shared folders...
    server: /home/alumnom/DHCP-server => /vagrant
==> server: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> server: flag to force provisioning. Provisioners marked to run always will still run.
==> c1: Checking if box 'debian/bullseye64' version '11.20241217.1' is up to date...
==> c1: Clearing any previously set forwarded ports...
==> c1: Fixed port collision for 22 => 2222. Now on port 2200.
==> c1: Clearing any previously set network interfaces...
==> c1: Preparing network interfaces based on configuration...
    c1: Adapter 1: nat
    c1: Adapter 2: intnet
==> c1: Forwarding ports...
    c1: 22 (guest) => 2200 (host) (adapter 1)
==> c1: Running 'pre-boot' VM customizations...
==> c1: Booting VM...
==> c1: Waiting for machine to boot. This may take a few minutes...
    c1: SSH address: 127.0.0.1:2200
    c1: SSH username: vagrant
    c1: SSH auth method: private key
    c1: Warning: Connection reset. Retrying...
==> c1: Machine booted and ready!
==> c1: Checking for guest additions in VM...
    c1: The guest additions on this VM do not match the installed version of
    c1: VirtualBox! In most cases this is fine, but in rare cases it can
    c1: prevent things such as shared folders from working properly. If you see
    c1: shared folder errors, please make sure the guest additions within the
    c1: virtual machine match the version of VirtualBox you have installed on
    c1: your host and reload your VM.
    c1:
    c1: Guest Additions Version: 6.0.0 r127566
    c1: VirtualBox Version: 7.1
```

After the machines start, verify that they exist with:

```
vagrant status
```

4. SERVER CONFIGURATION

Access the server machine:

```
vagrant ssh server
```

Update packages and install the DHCP service:

```
sudo apt update
sudo apt install isc-dhcp-server -y
```

4.1. INTERFACE SELECTION

We need to tell the DHCP service which network card (interface) it should use to listen for client requests.

List the interfaces with:

```
ip a
```

You should see something similar to:

- eth0 → NAT (Internet access)
- eth1 → 192.168.56.10 (host-only network)
- eth2 → 192.168.57.10 (internal network for DHCP)

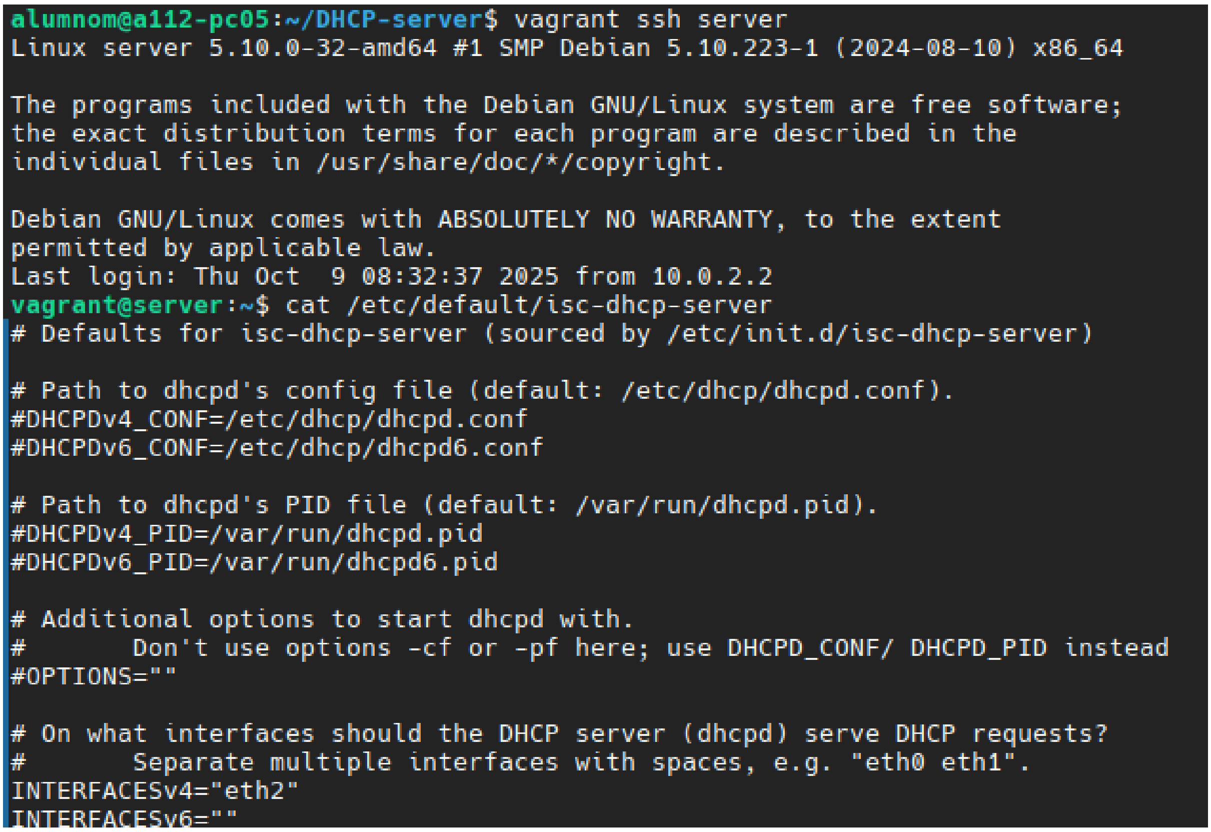
In this case, the DHCP server must listen on **eth2**.

Edit the file:

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

Set this line:

```
INTERFACESv4="eth2"
```



```
alumnom@a112-pc05:~/DHCP-server$ vagrant ssh server
Linux server 5.10.0-32-amd64 #1 SMP Debian 5.10.223-1 (2024-08-10) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  9 08:32:37 2025 from 10.0.2.2
vagrant@server:~$ cat /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="eth2"
INTERFACESv6=""
```

4.2. DHCP CONFIGURATION FILE

Now we configure the DHCP server to define the IP range and network parameters.

First, make a backup of the configuration file:

```
sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.bak
```

Then edit it:

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

Add the following configuration:

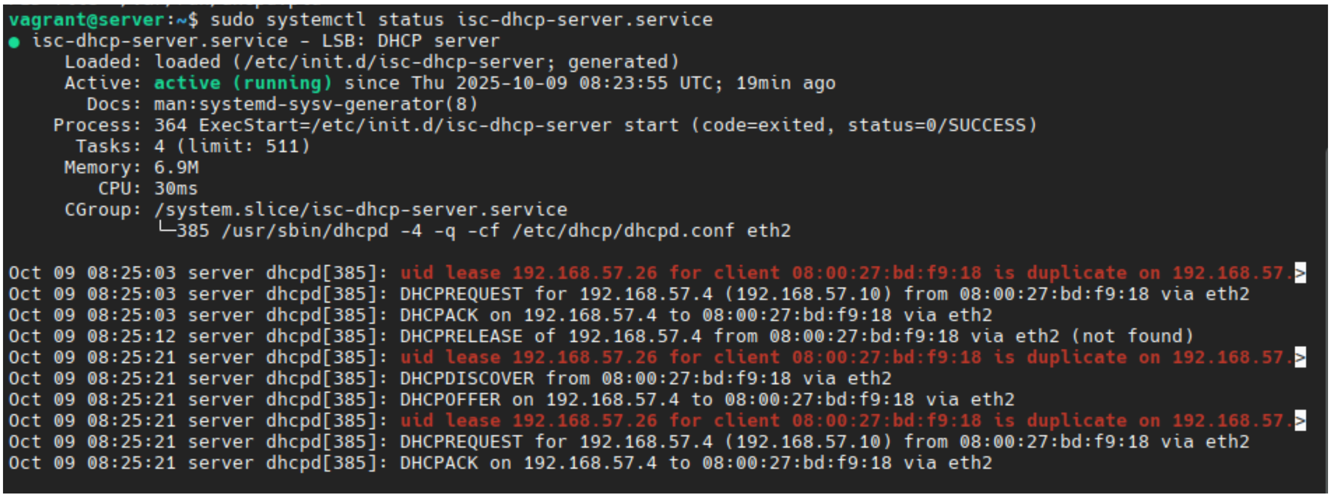
```
default-lease-time 86400;
max-lease-time 691200;

subnet 192.168.57.0 netmask 255.255.255.0 {
range 192.168.57.25 192.168.57.50;
option broadcast-address 192.168.57.255;
option routers 192.168.57.10;
option domain-name-servers 8.8.8.8, 4.4.4.4;
option domain-name "micasa.es";
 }
```

Explanation:

- `range` → IP addresses that will be assigned dynamically.
- `routers` → default gateway.
- `domain-name-servers` → DNS servers.
- `domain-name` → the local network name for clients.

Restart and verify the service:

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



```
vagrant@server:~$ sudo systemctl status isc-dhcp-server.service
● isc-dhcp-server.service - LSB: DHCP server
     Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
     Active: active (running) since Thu 2025-10-09 08:23:55 UTC; 19min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 364 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
      Tasks: 4 (limit: 511)
     Memory: 6.9M
        CPU: 30ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─385 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf eth2

Oct 09 08:25:03 server dhcpd[385]: uid lease 192.168.57.26 for client 08:00:27:bd:f9:18 is duplicate on 192.168.57.>
Oct 09 08:25:03 server dhcpd[385]: DHCPREQUEST for 192.168.57.4 (192.168.57.10) from 08:00:27:bd:f9:18 via eth2
Oct 09 08:25:03 server dhcpd[385]: DHCPACK on 192.168.57.4 to 08:00:27:bd:f9:18 via eth2
Oct 09 08:25:12 server dhcpd[385]: DHCPRELEASE of 192.168.57.4 from 08:00:27:bd:f9:18 via eth2 (not found)
Oct 09 08:25:21 server dhcpd[385]: uid lease 192.168.57.26 for client 08:00:27:bd:f9:18 is duplicate on 192.168.57.>
Oct 09 08:25:21 server dhcpd[385]: DHCPDISCOVER from 08:00:27:bd:f9:18 via eth2
Oct 09 08:25:21 server dhcpd[385]: DHCPOFFER on 192.168.57.4 to 08:00:27:bd:f9:18 via eth2
Oct 09 08:25:21 server dhcpd[385]: uid lease 192.168.57.26 for client 08:00:27:bd:f9:18 is duplicate on 192.168.57.>
Oct 09 08:25:21 server dhcpd[385]: DHCPREQUEST for 192.168.57.4 (192.168.57.10) from 08:00:27:bd:f9:18 via eth2
Oct 09 08:25:21 server dhcpd[385]: DHCPACK on 192.168.57.4 to 08:00:27:bd:f9:18 via eth2
```

5. CLIENT CONFIGURATION (c1 AND c2)

Access client c1:

```
vagrant ssh c1
```

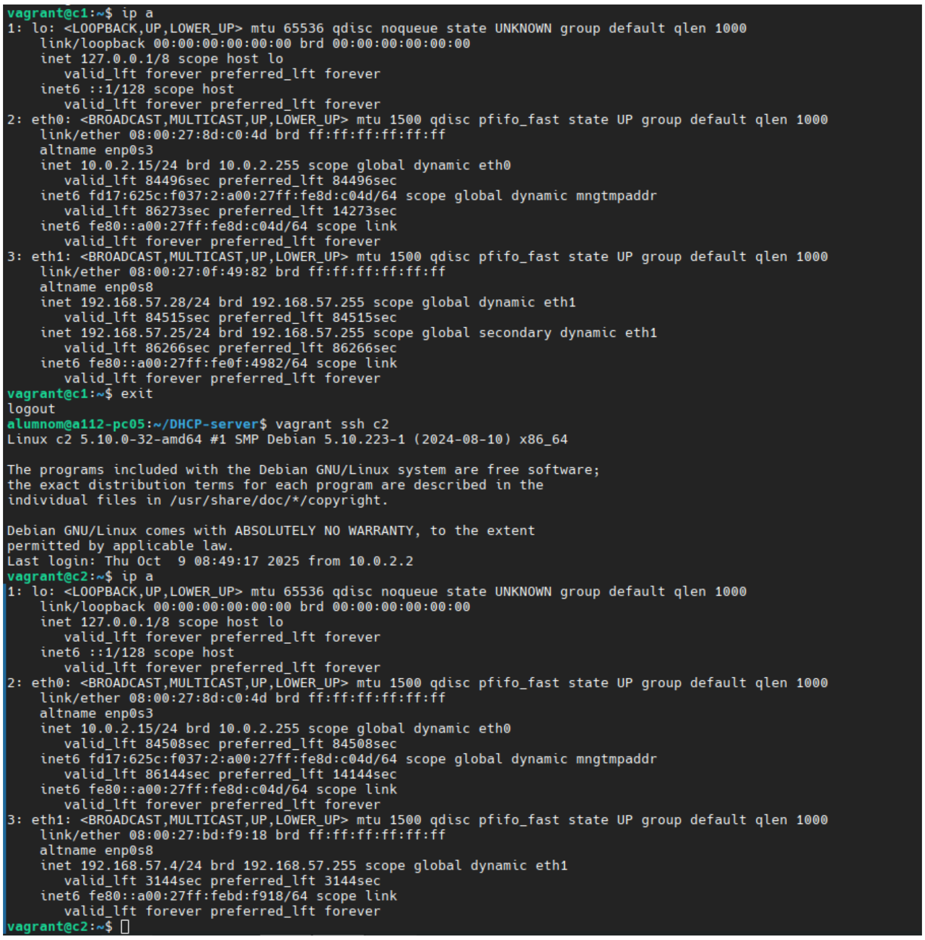
Request an IP from the DHCP server manually:

```
sudo dhclient -v
ip a
```

You should see an IP between 192.168.57.25 and 192.168.57.50.

Repeat with c2:

```
vagrant ssh c2
sudo dhclient -v
ip a
```



```
vagrant@c1:~$ ip a
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
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:8d:c0:4d brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 84496sec preferred_lft 84496sec
    inet6 fd17:625c:f037:2:a00:27ff:fe8d:c04d/64 scope global dynamic mngtmpaddr
       valid_lft 86273sec preferred_lft 14273sec
    inet6 fe80::a00:27ff:fe8d:c04d/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:0f:49:82 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.57.28/24 brd 192.168.57.255 scope global dynamic eth1
       valid_lft 84515sec preferred_lft 84515sec
    inet 192.168.57.25/24 brd 192.168.57.255 scope global secondary dynamic eth1
       valid_lft 86266sec preferred_lft 86266sec
    inet6 fe80::a00:27ff:fe0f:4982/64 scope link
       valid_lft forever preferred_lft forever
vagrant@c1:~$ exit
logout
alumnom@a112-pc05:~/DHCP-server$ vagrant ssh c2
Linux c2 5.10.0-32-amd64 #1 SMP Debian 5.10.223-1 (2024-08-10) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  9 08:49:17 2025 from 10.0.2.2
vagrant@c2:~$ ip a
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
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:8d:c0:4d brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 84508sec preferred_lft 84508sec
    inet6 fd17:625c:f037:2:a00:27ff:fe8d:c04d/64 scope global dynamic mngtmpaddr
       valid_lft 86144sec preferred_lft 14144sec
    inet6 fe80::a00:27ff:fe8d:c04d/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:bd:f9:18 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.57.4/24 brd 192.168.57.255 scope global dynamic eth1
       valid_lft 3144sec preferred_lft 3144sec
    inet6 fe80::a00:27ff:febd:f918/64 scope link
       valid_lft forever preferred_lft forever
vagrant@c2:~$
```

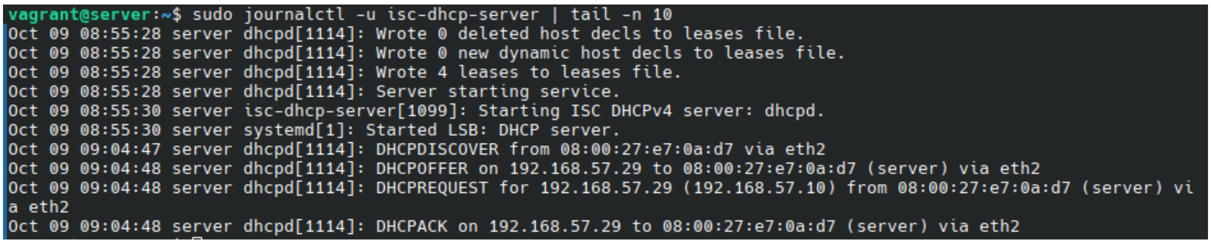
6. CHECKING LOGS AND LEASES

On the server, check the DHCP logs to see communication with clients:

```
sudo journalctl -u isc-dhcp-server | tail -n 10
```

You should see messages like:

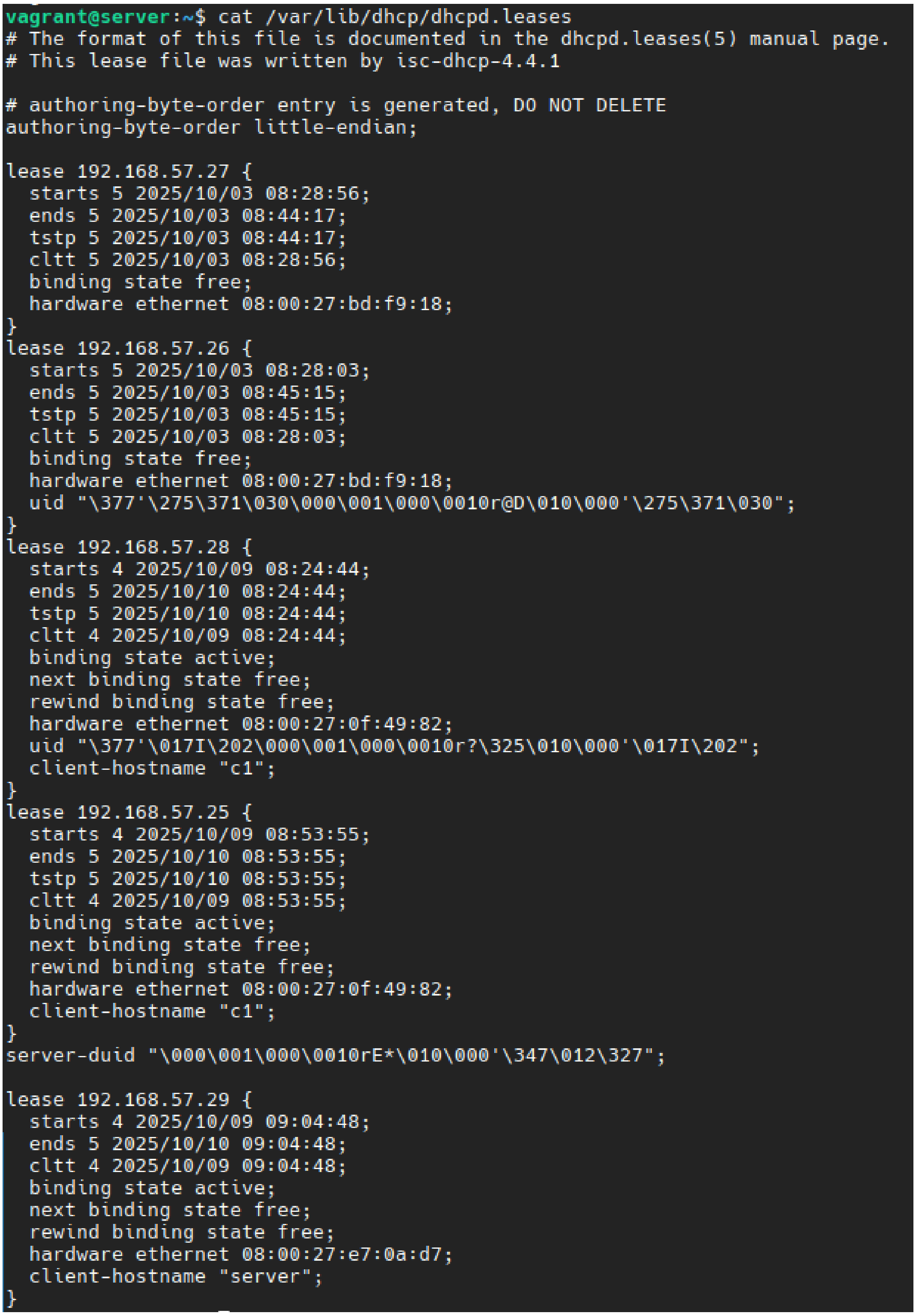
- DHCPDISCOVER
- DHCPOFFER
- DHCPREQUEST
- DHCPACK



```
vagrant@server:~$ sudo journalctl -u isc-dhcp-server | tail -n 10
Oct 09 08:55:28 server dhcpd[1114]: Wrote 0 deleted host decls to leases file.
Oct 09 08:55:28 server dhcpd[1114]: Wrote 0 new dynamic host decls to leases file.
Oct 09 08:55:28 server dhcpd[1114]: Wrote 4 leases to leases file.
Oct 09 08:55:28 server dhcpd[1114]: Server starting service.
Oct 09 08:55:30 server isc-dhcp-server[1099]: Starting ISC DHCPv4 server: dhcpd.
Oct 09 08:55:30 server systemd[1]: Started LSB: DHCP server.
Oct 09 09:04:47 server dhcpd[1114]: DHCPDISCOVER from 08:00:27:e7:0a:d7 via eth2
Oct 09 09:04:48 server dhcpd[1114]: DHCPOFFER on 192.168.57.29 to 08:00:27:e7:0a:d7 (server) via eth2
Oct 09 09:04:48 server dhcpd[1114]: DHCPREQUEST for 192.168.57.29 (192.168.57.10) from 08:00:27:e7:0a:d7 (server) vi
a eth2
Oct 09 09:04:48 server dhcpd[1114]: DHCPACK on 192.168.57.29 to 08:00:27:e7:0a:d7 (server) via eth2
```

Check the leases file (where the server stores assigned IPs):

cat /var/lib/dhcp/dhcpd.leases



```
vagrant@server:~$ 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-4.4.1

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

lease 192.168.57.27 {
  starts 5 2025/10/03 08:28:56;
  ends 5 2025/10/03 08:44:17;
  tstp 5 2025/10/03 08:44:17;
  cltt 5 2025/10/03 08:28:56;
  binding state free;
  hardware ethernet 08:00:27:bd:f9:18;
}
lease 192.168.57.26 {
  starts 5 2025/10/03 08:28:03;
  ends 5 2025/10/03 08:45:15;
  tstp 5 2025/10/03 08:45:15;
  cltt 5 2025/10/03 08:28:03;
  binding state free;
  hardware ethernet 08:00:27:bd:f9:18;
  uid "\377'\275\371\030\000\001\000\0010r@D\010\000'\275\371\030";
}
lease 192.168.57.28 {
  starts 4 2025/10/09 08:24:44;
  ends 5 2025/10/10 08:24:44;
  tstp 5 2025/10/10 08:24:44;
  cltt 4 2025/10/09 08:24:44;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:0f:49:82;
  uid "\377'\017I\202\000\001\000\0010r?\325\010\000'\017I\202";
  client-hostname "c1";
}
lease 192.168.57.25 {
  starts 4 2025/10/09 08:53:55;
  ends 5 2025/10/10 08:53:55;
  tstp 5 2025/10/10 08:53:55;
  cltt 4 2025/10/09 08:53:55;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:0f:49:82;
  client-hostname "c1";
}
server-duid "\000\001\000\0010rE*\010\000'\347\012\327";

lease 192.168.57.29 {
  starts 4 2025/10/09 09:04:48;
  ends 5 2025/10/10 09:04:48;
  cltt 4 2025/10/09 09:04:48;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:e7:0a:d7;
  client-hostname "server";
}
```

7. FIXED IP FOR C2

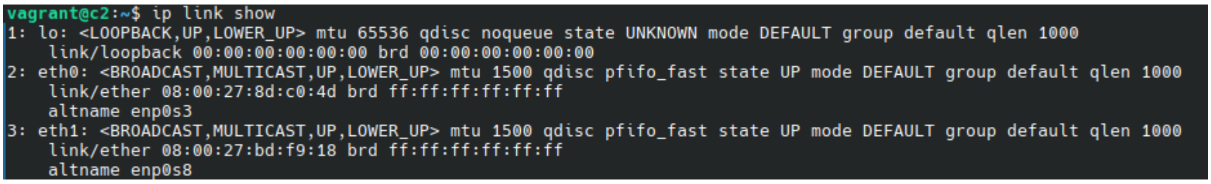
Now we'll assign a fixed IP address to c2 based on its MAC address.

First, get c2's MAC address:

```
vagrant ssh c2
ip link show
```

Look for interface eth1 (the internal one). You'll see something like:

```
link/ether 08:00:27:bd:f9:18
```



```
vagrant@c2:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:8d:c0:4d brd ff:ff:ff:ff:ff:ff
    altname enp0s3
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:bd:f9:18 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
```

Go back to the server and edit the DHCP configuration file again:

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

At the end of the file, add:

```
host c2 {
        hardware ethernet 08:00:27:bd:f9:18;
        fixed-address 192.168.57.4;
        default-lease-time 3600;
        option domain-name-servers 1.1.1.1;
}
```

Explanation:

- `hardware ethernet` → MAC address of c2.
- `fixed-address` → IP that will always be assigned to that MAC.
- `default-lease-time` → duration of the lease (1 hour = 3600 seconds).

Validate configuration:

```
sudo dhcpd -t
```

Restart the DHCP service:

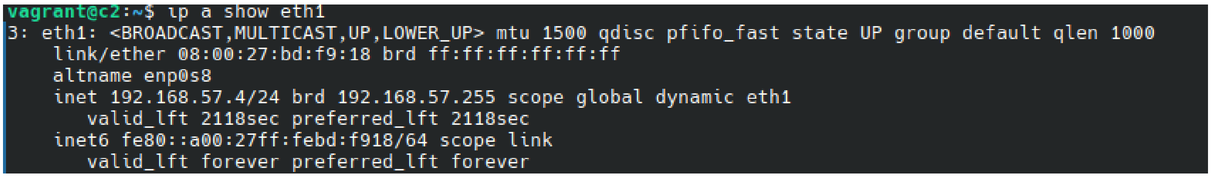
```
sudo systemctl restart isc-dhcp-server
```

On c2, release and renew the IP:

```
sudo dhclient -r eth1
sudo dhclient -v eth1
```

Check that c2 now has the fixed IP:

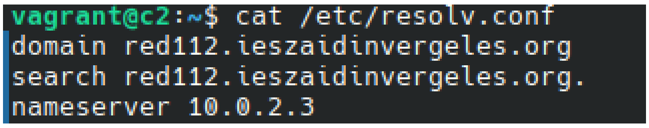
```
ip a show eth1
```



```
vagrant@c2:~$ ip a show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:bd:f9:18 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.57.4/24 brd 192.168.57.255 scope global dynamic eth1
       valid_lft 2118sec preferred_lft 2118sec
    inet6 fe80::a00:27ff:febd:f918/64 scope link
       valid_lft forever preferred_lft forever
```

Also verify the DNS configuration:

```
cat /etc/resolv.conf
```



```
vagrant@c2:~$ cat /etc/resolv.conf
domain red112.ieszaidinvergeles.org
search red112.ieszaidinvergeles.org.
nameserver 10.0.2.3
```

8. CONCLUSIÓN

This practice demonstrated how to deploy and configure a DHCP server using Debian and Vagrant.

- The server successfully provided automatic IPs to multiple clients in the same internal network.
- The configuration of a fixed IP based on MAC address ensures a consistent IP assignment for specific devices.
- All tasks were performed in an isolated virtual environment, making it ideal for testing without affecting the real network.

9. SUMMARY OF USEFUL COMMANDS

Check network interfaces:

```
ip a
```

Edit files:

```
sudo nano [file]
```

Validate DHCP configuration:

```
sudo dhcpd -t
```

Restart DHCP service:

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

View logs:

```
sudo journalctl -u isc-dhcp-server
```

View DHCP leases:

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

Renew IP on clients:

```
sudo dhclient -r
sudo dhclient -v
```