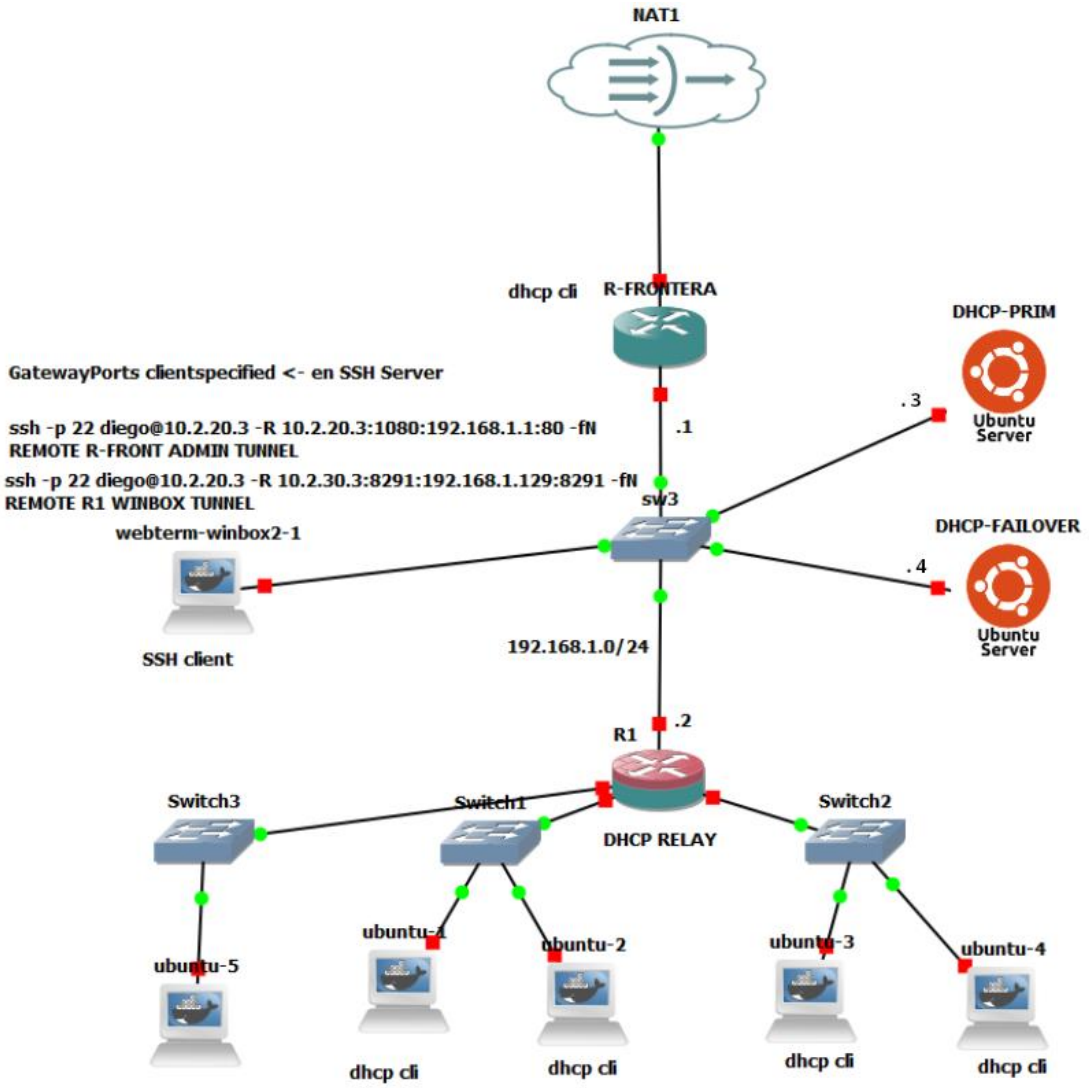
1.Nombre:.Cristóbal Suárez Abad..... Fecha: ……

SERVICIO DHCP

NOMBRE DEL PROYECTO EN GNS3: **SRI_Recuperacion_DHCP_Cristobal_Suarez**

Crea el siguiente diagrama en GNS3:



El direccionamiento de las LANs será:

- LAN Switch1 10.10.X*2.0/23
- LAN Switch2 10.12.X.32/27
- LAN Switch3 10.8.X.96/28

donde X es vuestro número de lista.

1. **(0.5 pts.) Establece la configuración de red necesaria en los servidores DHCP y las rutas estáticas necesarias para conectar con las redes bajo R1.**

MODIFICAMOS EL ARCHIVO: nano /etc/netplan/50-cloud-init.yaml

Servidor DHCP 1:

```
network:
  version: 2
  ethernets:
    ens3:
#     match:
#       macaddress: 0c:3a:40:aa:00:00
#     set-name: ens3
#     dhcp4: false
      addresses:
       - 192.168.1.3/24
      routes:
       - to: 0.0.0.0/0
         via: 192.168.1.1
       - to: 10.10.14.0/23
         via: 192.168.1.2
       - to: 10.12.7.32/27
         via: 192.168.1.2
       - to: 10.8.7.96/28
         via: 192.168.1.2
      nameservers:
       addresses:
         - 172.16.200.1
         - 8.8.8.8
```

```
root@ubuntu-cloud:/home/ubuntu# cat /etc/netplan/50-cloud-init.yaml
network:
  version: 2
  ethernets:
    ens3:
#       match:
#         macaddress: 0c:3a:40:aa:00:00
#       set-name: ens3
#       dhcp4: false
      addresses:
        - 192.168.1.3/24
      routes:
        - to: 0.0.0.0/0
          via: 192.168.1.1
        - to: 10.10.14.0/23
          via: 192.168.1.2
        - to: 10.12.7.32/27
          via: 192.168.1.2
        - to: 10.8.7.96/28
          via: 192.168.1.2
      nameservers:
        addresses:
          - 172.16.200.1
          - 8.8.8.8
root@ubuntu-cloud:/home/ubuntu#
```

SERVIDOR DHCP 2:

```
network:
  version: 2
  ethernets:
    ens3:
#       match:
#         macaddress: 0c:3a:40:aa:00:00
#       set-name: ens3
#       dhcp4: false
      addresses:
        - 192.168.1.4/24
      routes:
        - to: 0.0.0.0/0
          via: 192.168.1.1
        - to: 10.10.14.0/23
          via: 192.168.1.2
        - to: 10.12.7.32/27
          via: 192.168.1.2
```

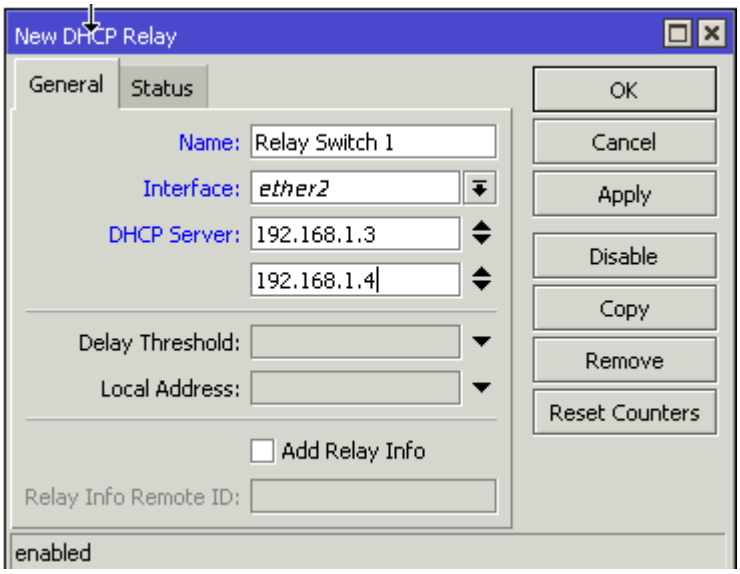
- to: 10.8.7.96/28
via: 192.168.1.2
nameservers:
addresses:
- 172.16.200.1
- 8.8.8.8

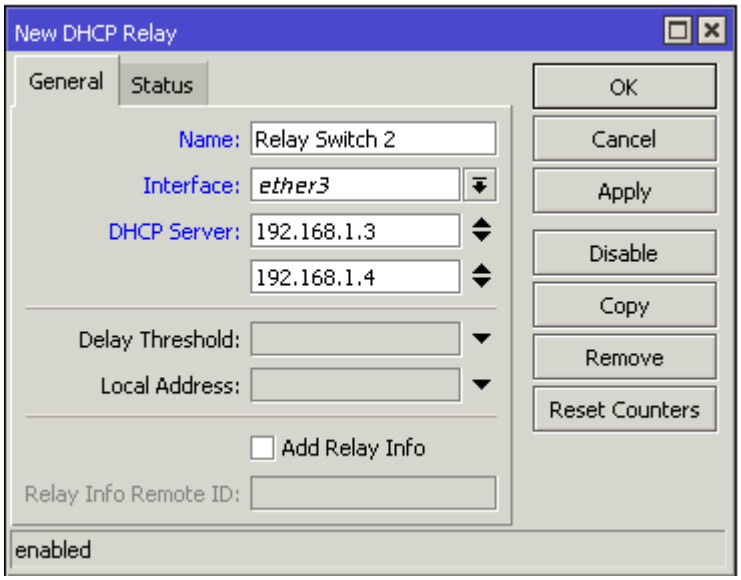
```
network:
  version: 2
  ethernets:
    ens3:
#       match:
#         macaddress: 0c:3a:40:aa:00:00
#       set-name: ens3
#       dhcp4: false
      addresses:
        - 192.168.1.4/24
      routes:
        - to: 0.0.0.0/0
          via: 192.168.1.1
        - to: 10.10.14.0/23
          via: 192.168.1.2
        - to: 10.12.7.32/27
          via: 192.168.1.2
        - to: 10.8.7.96/28
          via: 192.168.1.2
      nameservers:
        addresses:
          - 172.16.200.1
          - 8.8.8.8
root@ubuntu-cloud:/home/ubuntu#
```

Una vez terminada la configuración:
netplan apply

2. **(1 pts) Realiza la configuración de red necesaria en R1 para que las peticiones DHCP de las LAN de SW1 y SW2 alcancen el servidor DHCP.**

Establecemos el servicio de DHCP Relay en el Router R1 (Mikrotik).





3. **(3 pts.) Instala el servicio DHCP en los servidores DHCP y configura lo necesario para que se reparta entre ambos (50%/50%) la configuración de red a los equipos de las LAN SW1 y SW2, con las siguientes especificaciones:**
    i. Asignar un rango que deje 4 ips para uso como estáticas o reservas.
    ii. Duración de la concesión: 18 horas
    iii. Servidores DNS: 9.9.9.9 y 7.7.5.5
    iv. Asigna una puerta de enlace lógica válida
    v. Dominio “make.it”
    vi. Ambos servidores serán autoritarios.

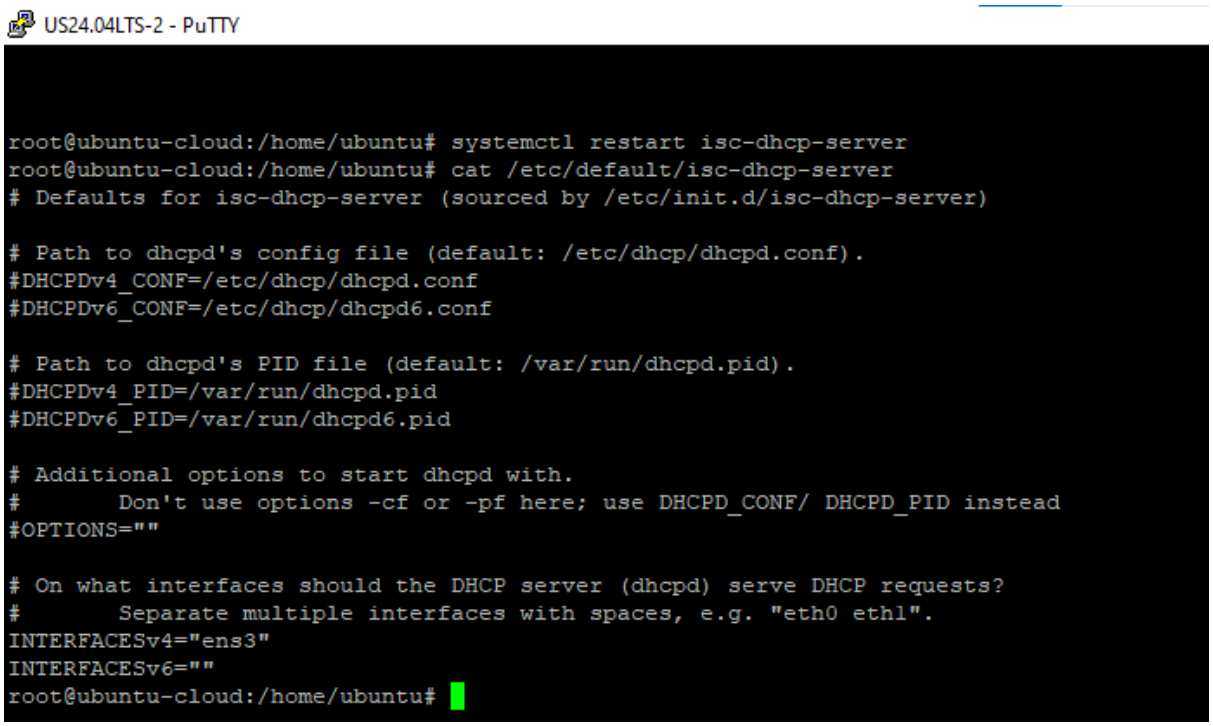
Ya hemos instalado isc-dhcp-server en ambos servidores:
Modificamos en **ambos** para poner el nombre de la interfaz donde pondremos el servicio:

Servidor DHCP 1:

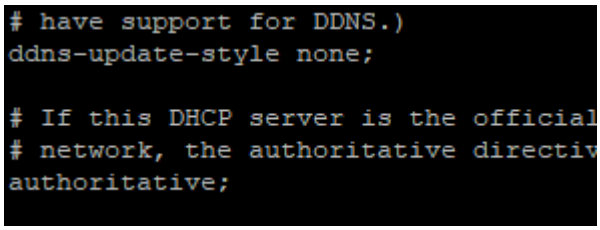
US24.04LTS-1 - PuTTY

```
      nameservers:
        addresses:
          - 172.16.200.1
          - 8.8.8.8
root@ubuntu-cloud:/home/ubuntu# 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="ens3"
INTERFACESv6=""
root@ubuntu-cloud:/home/ubuntu#
```

Servidor DHCP 2:



Ahora modificamos: **/etc/dhcp/dhcpd.conf**
Debemos descomentar:
authoritative;
ddns-update-style none;
DHCP 1 y 2:



Ahora hay una parte que es individual para cada uno:
DHCP Server 1:
#CONFIGURACIÓN DE ((DHCP-PRIM))

failover peer "FAILOVER" {
  primary;
  address 192.168.1.3;
  port 647;
  peer address 192.168.1.4;

```
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
  mclt 1800;
  split 128;
}
```

```
#CONFIGURACIÓN DE ((DHCP-PRIM))

failover peer "FAILOVER" {
  primary;
  address 192.168.1.3;
  port 647;
  peer address 192.168.1.4;
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
  mclt 1800;
  split 128;
}
```

SERVIDOR DHCP 2:

```
#CONFIGURACIÓN DE ((DHCP-SECOND))

failover peer "FAILOVER" {
  secondary;
  address 192.168.1.4;
  port 647;
  peer address 192.168.1.3;
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
}
```

```
#CONFIGURACIÓN DE ((DHCP-SECOND))

failover peer "FAILOVER" {
  secondary;
  address 192.168.1.4;
  port 647;
  peer address 192.168.1.3;
  peer port 647;
  max-unacked-updates 10;
  max-response-delay 30;
  load balance max seconds 3;
}
```

Y una parte que es común para ambos servidores:

##Configuración común

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  not authoritative;
}


#LAN Switch1 10.10.14.0/23
#LAN Switch2 10.12.7.32/27

# SubNet 10.10.14.0/23
subnet 10.10.14.0 netmask 255.255.254.0 {
  option domain-name-servers 9.9.9.9, 7.7.5.5;
  option domain-name "make.it";
  option routers 10.10.14.1;
  pool {
    failover peer "FAILOVER";
    range 10.10.14.5 10.10.15.254;
    default-lease-time 64800;
    max-lease-time 86000;
    host megalol-01 {
      hardware ethernet 02:42:d2:70:03:00;
      fixed-address 10.10.14.4;
    }
  }
```

```
}

# SubNet 10.12.7.32/27
subnet 10.12.7.32 netmask 255.255.255.224 {
  option domain-name-servers 9.9.9.9, 7.7.5.5;
  option domain-name "make.it";
  option routers 10.12.7.33;
  pool {
    failover peer "FAILOVER";
    range 10.12.7.39 10.12.7.62;
    default-lease-time 64800;
    max-lease-time 86000;
    host megalol-02 {
      hardware ethernet 02:42:97:2b:f7:00;
      fixed-address 10.12.7.40;
    }
  }
}
```

```
##Configuración común


subnet 192.168.1.0 netmask 255.255.255.0 {
  not authoritative;
}
```

```
# SubNet 10.10.14.0/23
subnet 10.10.14.0 netmask 255.255.254.0 {
  option domain-name-servers 9.9.9.9, 7.7.5.5;
  option domain-name "make.it";
  option routers 10.10.14.1;
  pool {
    failover peer "FAILOVER";
    range 10.10.14.5 10.10.15.254;
    default-lease-time 64800;
    max-lease-time 86000;
    host megalol-01 {
      hardware ethernet 02:42:d2:70:03:00;
      fixed-address 10.10.14.4;
    }
  }
}
```

```
# SubNet 10.12.7.32/27
subnet 10.12.7.32 netmask 255.255.255.224 {
  option domain-name-servers 9.9.9.9, 7.7.5.5;
  option domain-name "make.it";
  option routers 10.12.7.33;
  pool {
    failover peer "FAILOVER";
    range 10.12.7.39 10.12.7.62;
    default-lease-time 64800;
    max-lease-time 86000;
    host megalol-02 {
      hardware ethernet 02:42:97:2b:f7:00;
      fixed-address 10.12.7.40;
    }
  }
}
```

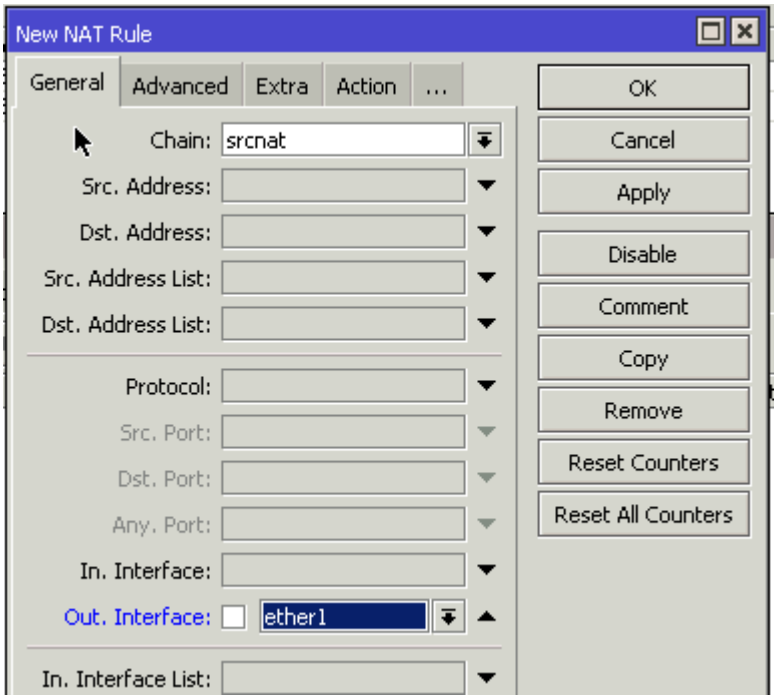
Una vez terminada la configuración:

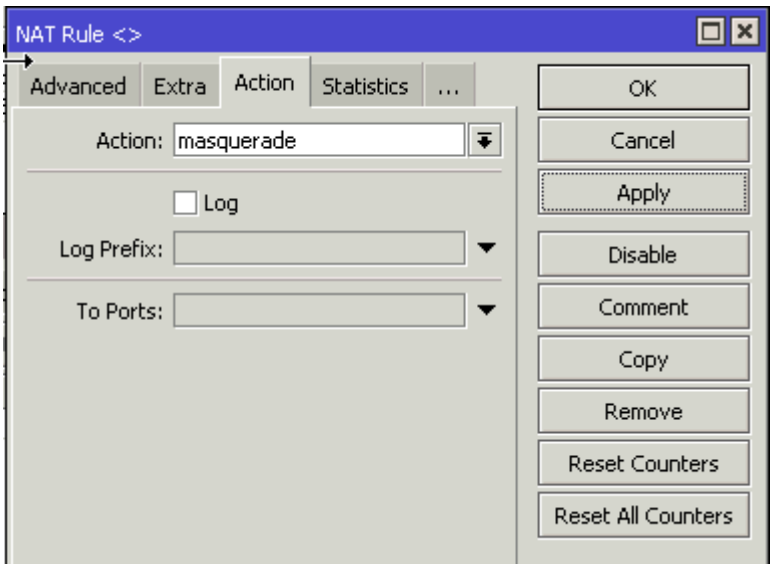
systemctl restart isc-dhcp-server

systemctl status isc-dhcp-server

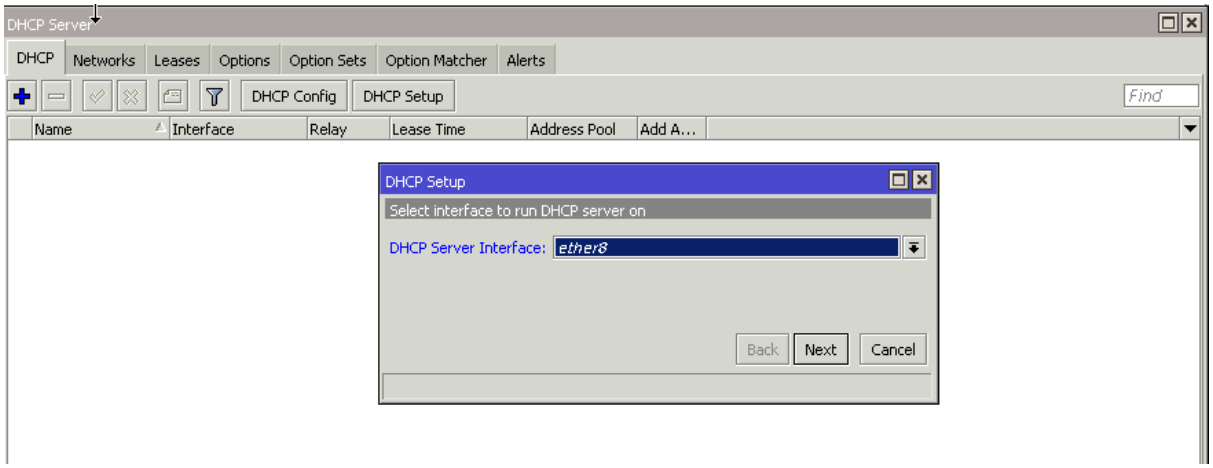
```
root@ubuntu-cloud:/home/ubuntu# systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled; >
     Active: active (running) since Wed 2025-11-19 19:08:13 UTC; 3min 26s ago
       Docs: man:dhcpd(8)
   Main PID: 2238 (dhcpd)
      Tasks: 1 (limit: 1112)
     Memory: 4.3M (peak: 4.8M)
        CPU: 184ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─2238 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/d>

Nov 19 19:08:13 ubuntu-cloud dhcpd[2238]: failover peer FAILOVER: Both servers >
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: peer FAILOVER: disconnected
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: failover peer FAILOVER: I move from n>
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: failover peer FAILOVER: peer moves fr>
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: failover peer FAILOVER: I move from c>
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: failover peer FAILOVER: Both servers >
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: balancing pool 5dd08ebe65a0 10.12.7.3>
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: balanced pool 5dd08ebe65a0 10.12.7.32>
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: balancing pool 5dd08ebe1c10 10.10.14.>
Nov 19 19:08:15 ubuntu-cloud dhcpd[2238]: balanced pool 5dd08ebe1c10 10.10.14.0>
lines 1-21/21 (END)
```

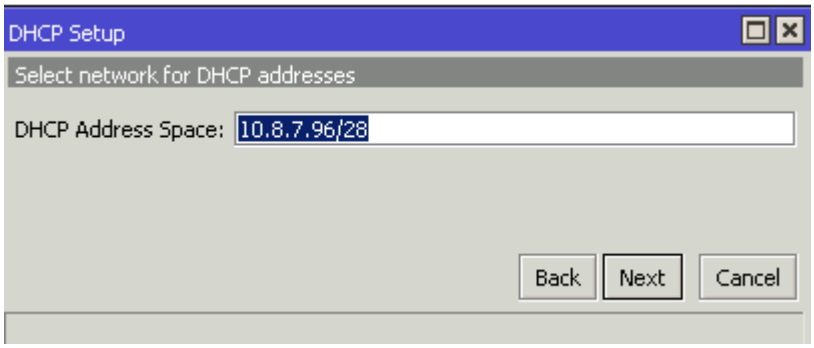
En el Router Relay Mikrotik establecemos una regla NAT:

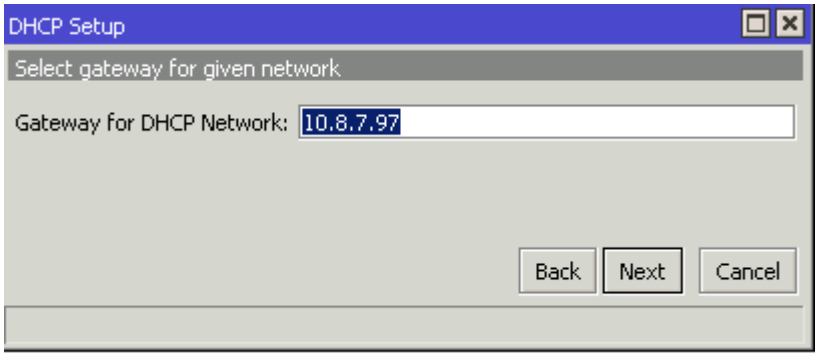


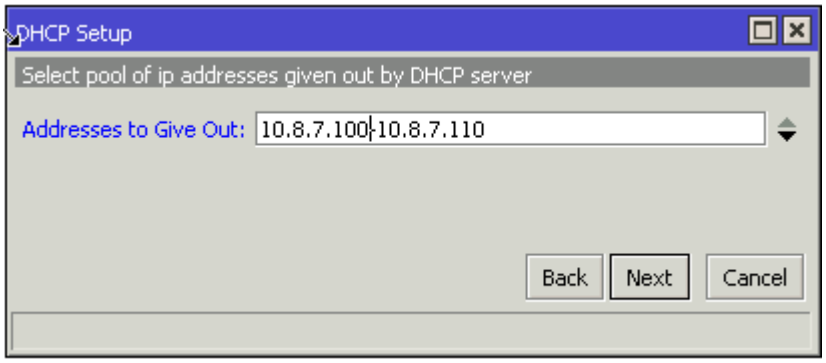


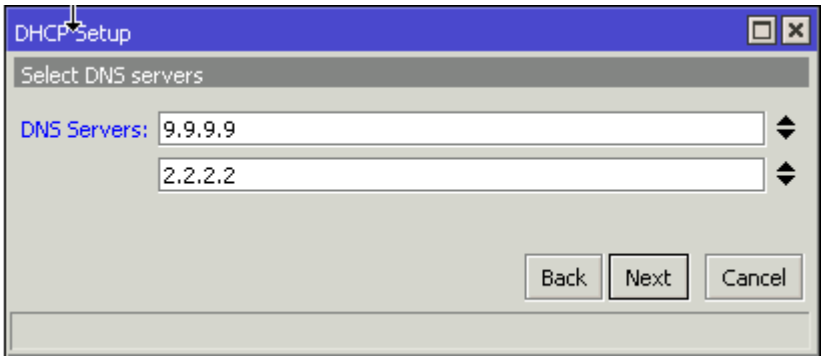
4. **(2 pts.) Configura lo necesario para que R1 sea el servidor DHCP de la LAN del SW3.**
    i. Asignar un rango que deje 3 ips para uso como estáticas o reservas.
    ii. Duración de la concesión: 9 horas
    iii. Servidores DNS: 9.9.9.9 y 2.2.2.2
    iv. Asigna una puerta de enlace lógica válida
    v. Dominio “agar.io”

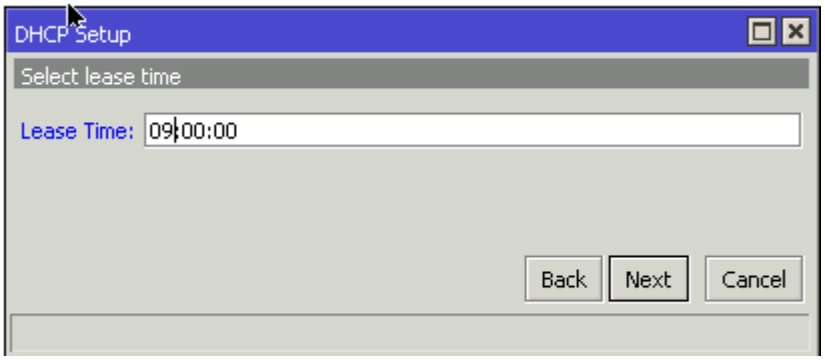



DHCP Setup
Select network for DHCP addresses
DHCP Address Space: 10.8.7.96/28
Back
Next
Cancel

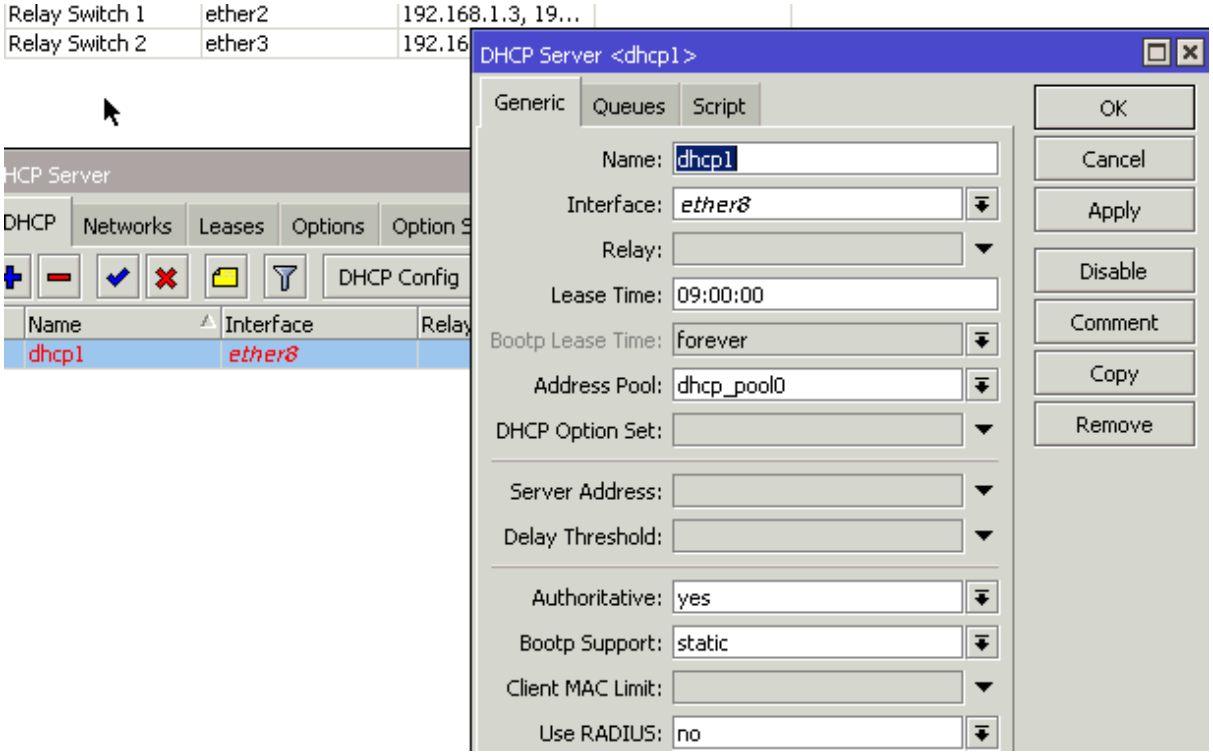

DHCP Setup
Select gateway for given network
Gateway for DHCP Network: 10.8.7.97
Back
Next
Cancel


DHCP Setup
Select pool of ip addresses given out by DHCP server
Addresses to Give Out: 10.8.7.100-10.8.7.110
Back
Next
Cancel

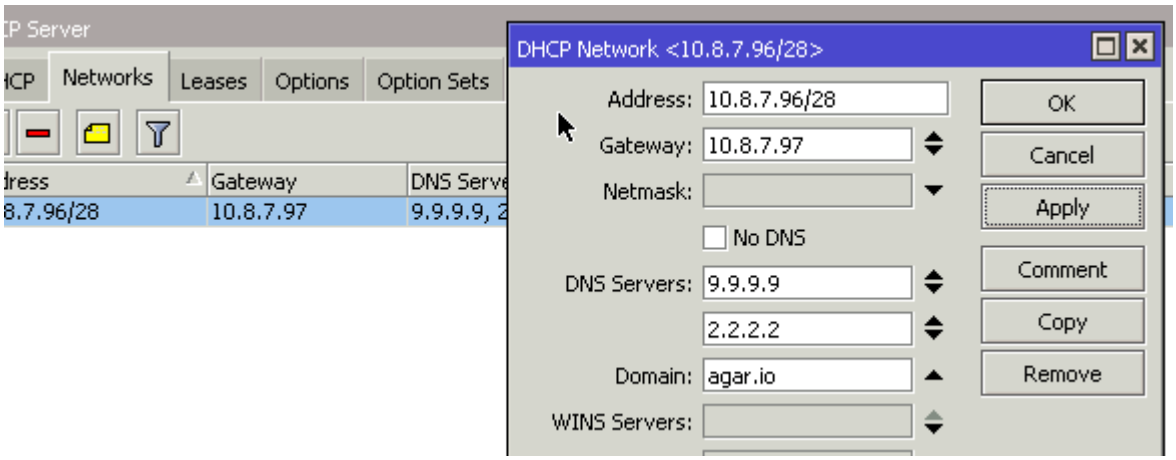

DHCP Setup
Select DNS servers
DNS Servers: 9.9.9.9
2.2.2.2
Back
Next
Cancel


DHCP Setup
Select lease time
Lease Time: 09:00:00
Back
Next
Cancel

CREADO:


Relay Switch 1 ether2 192.168.1.3, 19...
Relay Switch 2 ether3 192.16
HCP Server
DHCP
Networks
Leases
Options
Option S
DHCP Config
Name
Interface
Relay
dhcp1
ether8
DHCP Server <dhcp1>
Generic
Queues
Script
OK
Cancel
Apply
Disable
Comment
Copy
Remove
Name: dhcp1
Interface: ether8
Relay:
Lease Time: 09:00:00
Bootp Lease Time: forever
Address Pool: dhcp_pool0
DHCP Option Set:
Server Address:
Delay Threshold:
Authoritative: yes
Bootp Support: static
Client MAC Limit:
Use RADIUS: no

Esto se hace una vez se ha terminado el asistente: **poner el nombre del dominio**.



5. **(1 pts.) Enciende los clientes y <u>verifica en los servidores</u> qué direcciones están asignadas dinámicamente en este instante. Usa busybox y muestra la configuración de red incluidos dns y sufijo de búsqueda. Comprueba que hay conectividad con los servidores DHCP.**

SERVIDOR DHCP 1:

Usamos: **dhcp-lease-list --parsable**

```
To get manufacturer names please download http://standards-oui.ieee.org/oui.txt to /usr/local/etc/oui.txt
MAC 02:42:49:f2:e6:00 IP 10.10.15.1 HOSTNAME UbuntuDockerGuest-1 BEGIN 2025-11-19 18:32:02 END 2025-11-20 12:32:02 MANUFACTURER -NA-
MAC 02:42:4b:ef:05:00 IP 10.12.7.51 HOSTNAME UbuntuDockerGuest-2 BEGIN 2025-11-19 18:32:13 END 2025-11-20 06:32:13 MANUFACTURER -NA-
root@ubuntu-cloud:/home/ubuntu#
```

SERVIDOR DHCP 2:

```
root@ubuntu-cloud:/home/ubuntu# dhcp-lease-list --parsable
To get manufacturer names please download http://standards-oui.ieee.org/oui.txt to /usr/local/etc/oui.txt
MAC 02:42:49:f2:e6:00 IP 10.10.15.1 HOSTNAME UbuntuDockerGuest-1 BEGIN 2025-11-19 18:32:02 END 2025-11-20 12:32:02 MANUFACTURER -NA-
MAC 02:42:4b:ef:05:00 IP 10.12.7.51 HOSTNAME UbuntuDockerGuest-2 BEGIN 2025-11-19 18:32:13 END 2025-11-20 06:32:13 MANUFACTURER -NA-
root@ubuntu-cloud:/home/ubuntu#
```

Cliente Switch 1:

Al iniciar ya nos avisa que se nos concede una IP y el tiempo.

```
root@UbuntuDockerGuest-1: ~
UbuntuDockerGuest-1 console is now available... Press RETURN to get started.
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 10.10.15.1
udhcpc: lease of 10.10.15.1 obtained, lease time 64232
root@UbuntuDockerGuest-1:~#
```

Resultado de “ip a”.

```
root@UbuntuDockerGuest-1:~# hostname
UbuntuDockerGuest-1
root@UbuntuDockerGuest-1:~# b ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue 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
1661: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
    link/ether 02:42:49:f2:e6:00 brd ff:ff:ff:ff:ff:ff
    inet 10.10.15.1/23 scope global eth0
       valid_lft forever preferred_lft forever
root@UbuntuDockerGuest-1:~#
```

El gateway:

```
root@UbuntuDockerGuest-1:~# b ip route
default via 10.10.14.1 dev eth0  metric 1861
10.10.14.0/23 dev eth0 scope link  src 10.10.15.1
root@UbuntuDockerGuest-1:~#
```

Servidor DNS:

```
root@UbuntuDockerGuest-1:~# cat /etc/resolv.conf
search make.it
nameserver 9.9.9.9
nameserver 7.7.5.5
root@UbuntuDockerGuest-1:~#
```

¿Tenemos Internet? Claro que si.

```
root@UbuntuDockerGuest-1:~# ping x.uk
PING x.uk (185.249.71.213) 56(84) bytes of data.
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=1 ttl=45 time=39.2 ms
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=2 ttl=45 time=38.7 ms
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=3 ttl=45 time=38.8 ms
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=4 ttl=45 time=38.8 ms
^C
--- x.uk ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 38.696/38.876/39.222/0.205 ms
root@UbuntuDockerGuest-1:~#
```

Conectividad con los servidores DHCP:

```
root@UbuntuDockerGuest-1:~# ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=63 time=5.63 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=63 time=4.66 ms
^C
--- 192.168.1.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 4.656/5.141/5.627/0.485 ms
root@UbuntuDockerGuest-1:~# ping 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=1 ttl=63 time=4.96 ms
64 bytes from 192.168.1.4: icmp_seq=2 ttl=63 time=4.20 ms
^C
--- 192.168.1.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 4.201/4.579/4.958/0.378 ms
root@UbuntuDockerGuest-1:~#
```

Cliente Switch 2:

root@UbuntuDockerGuest-2: ~

```
UbuntuDockerGuest-2 console is now available... Press RETURN to get started.
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 10.12.7.51
udhcpc: lease of 10.12.7.51 obtained, lease time 42637
root@UbuntuDockerGuest-2:~#
```

```
UbuntuDockerGuest-2
root@UbuntuDockerGuest-2:~# b ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue 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
1662: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
    link/ether 02:42:4b:ef:05:00 brd ff:ff:ff:ff:ff:ff
    inet 10.12.7.51/27 scope global eth0
       valid_lft forever preferred_lft forever
root@UbuntuDockerGuest-2:~#
```

```
root@UbuntuDockerGuest-2:~# b ip route
default via 10.12.7.33 dev eth0  metric 1862
10.12.7.32/27 dev eth0 scope link  src 10.12.7.51
root@UbuntuDockerGuest-2:~#
```

```
root@UbuntuDockerGuest-2:~# cat /etc/resolv.conf
search make.it
nameserver 9.9.9.9
nameserver 7.7.5.5
root@UbuntuDockerGuest-2:~#
```

```
root@UbuntuDockerGuest-2:~# ping x.uk
PING x.uk (185.249.71.213) 56(84) bytes of data.
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=1 ttl=45 time=38.6 ms
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=2 ttl=45 time=37.9 ms
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=3 ttl=45 time=38.4 ms
^C
--- x.uk ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 37.871/38.291/38.636/0.316 ms
root@UbuntuDockerGuest-2:~#
```

root@UbuntuDockerGuest-2: ~

```
root@UbuntuDockerGuest-2:~# ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=63 time=5.56 ms
^C
--- 192.168.1.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.555/5.555/5.555/0.000 ms
root@UbuntuDockerGuest-2:~# ping 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=1 ttl=63 time=5.41 ms
^C
--- 192.168.1.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.414/5.414/5.414/0.000 ms
root@UbuntuDockerGuest-2:~#
```

Cliente Switch 3:

root@UbuntuDockerGuest-3: ~

```
UbuntuDockerGuest-3 console is now available... Press RETURN to get started.
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 10.8.7.110
udhcpc: lease of 10.8.7.110 obtained, lease time 32400
root@UbuntuDockerGuest-3:~#
```

```
root@UbuntuDockerGuest-3:~# b ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue 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
1666: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
    link/ether 02:42:76:60:eb:00 brd ff:ff:ff:ff:ff:ff
    inet 10.8.7.110/28 scope global eth0
       valid_lft forever preferred_lft forever
root@UbuntuDockerGuest-3:~#
```

```
root@UbuntuDockerGuest-3:~# b ip route
default via 10.8.7.97 dev eth0  metric 1866
10.8.7.96/28 dev eth0 scope link  src 10.8.7.110
root@UbuntuDockerGuest-3:~#
```

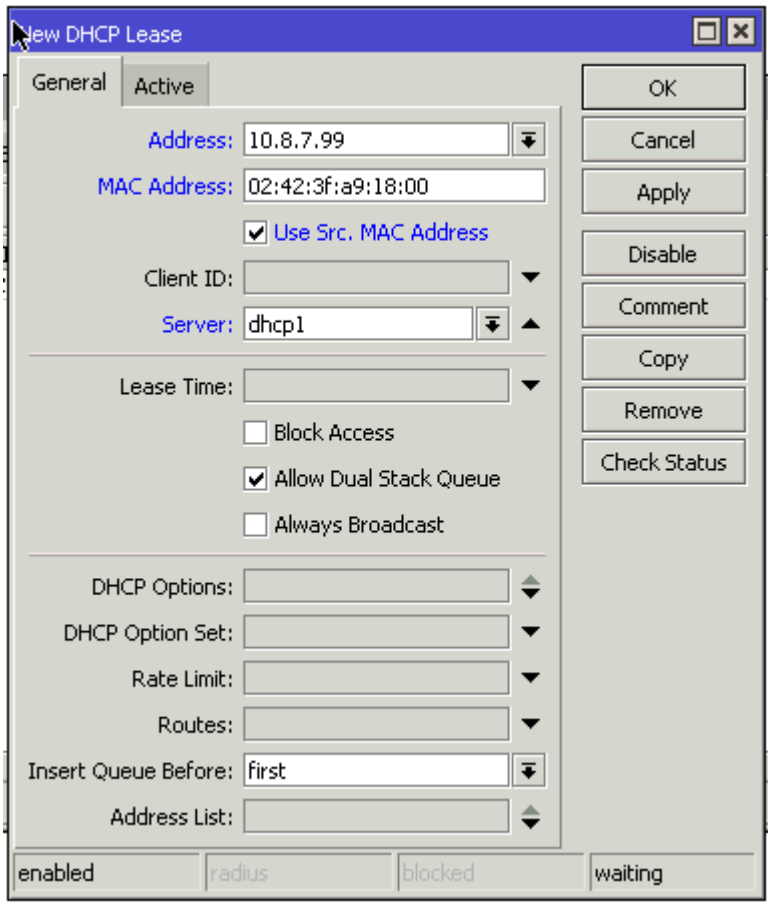
```
root@UbuntuDockerGuest-3:~# cat /etc/resolv.conf
search agar.io
nameserver 9.9.9.9
nameserver 2.2.2.2
root@UbuntuDockerGuest-3:~#
```

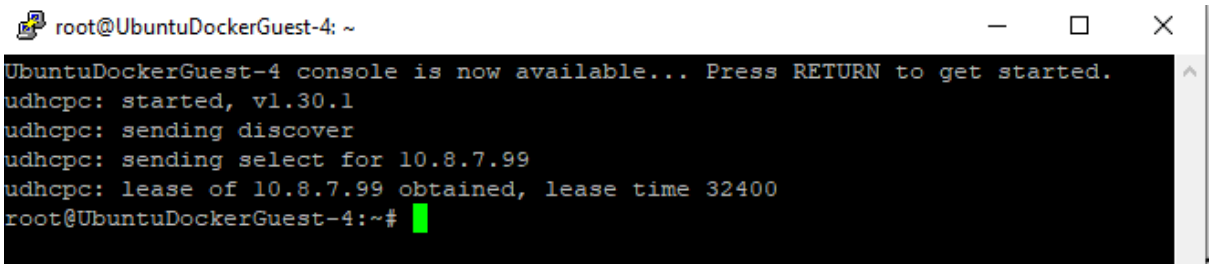
```
root@UbuntuDockerGuest-3:~# ping x.uk
PING x.uk (185.249.71.213) 56(84) bytes of data.
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=1 ttl=45 time=37.9 ms
64 bytes from 185.249.71.213 (185.249.71.213): icmp_seq=2 ttl=45 time=38.1 ms
^C
--- x.uk ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 37.926/37.994/38.062/0.068 ms
root@UbuntuDockerGuest-3:~#
```

```
root@UbuntuDockerGuest-3:~# ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=63 time=4.37 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=63 time=3.83 ms
^C
--- 192.168.1.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 3.833/4.102/4.372/0.269 ms
root@UbuntuDockerGuest-3:~# ping 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=1 ttl=63 time=6.46 ms
64 bytes from 192.168.1.4: icmp_seq=2 ttl=63 time=4.65 ms
64 bytes from 192.168.1.4: icmp_seq=3 ttl=63 time=4.37 ms
^C
--- 192.168.1.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 4.374/5.162/6.461/0.925 ms
root@UbuntuDockerGuest-3:~#
```

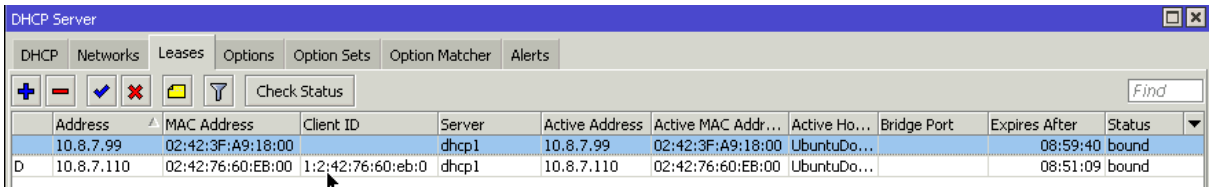
6. **(1 pts.) Realiza una reserva para cada LAN bajo R1 y demuestra que los clientes las obtienen.**

RESERVA EN LAN SWITCH 3:

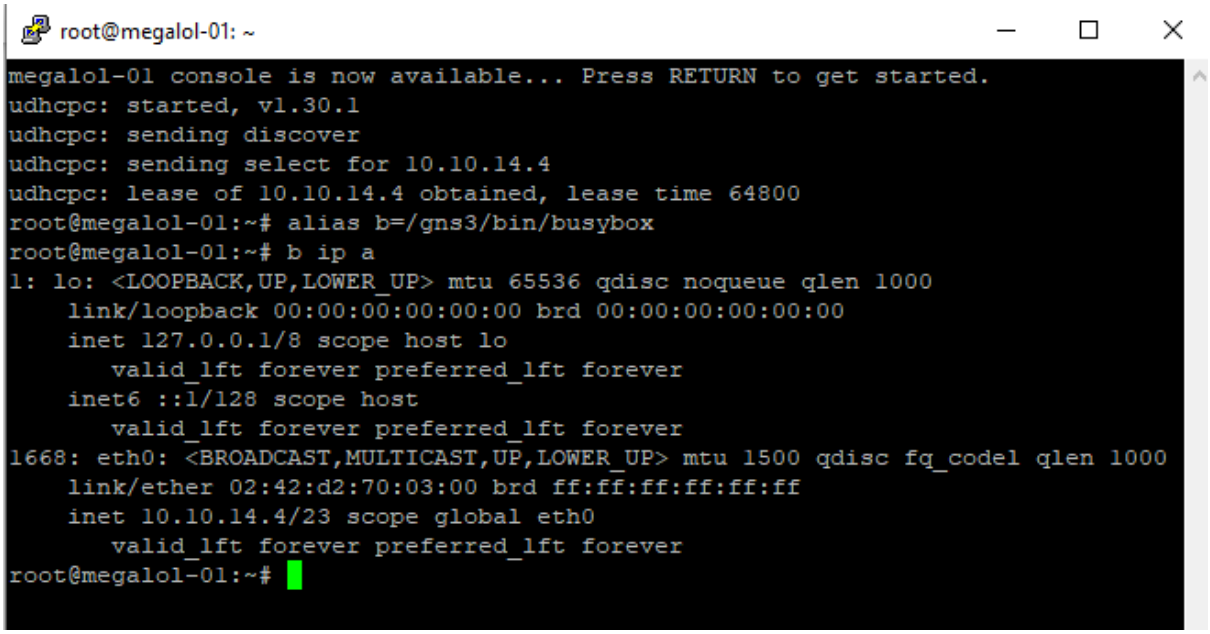




Un equipo con Reserva y el otro por DHCP:



Reserva Switch 1:
Hostname: megalol-01
Mac: 02:42:d2:70:03:00



Reserva Switch 2:
Hostname: megalol-02
Mac: 02:42:97:2b:f7:00