Linux administration avancée

04/01/2022

Étudiants:
CONSTANCY Denis
KANDEL Mavel
GEORGE Mukilventhan

Intervenant:
M.DEBLANGY

Sommaire

SERVEUR DHCP RÉEL

Mini serveur DHCP

SERVEUR DHCP RÉEL

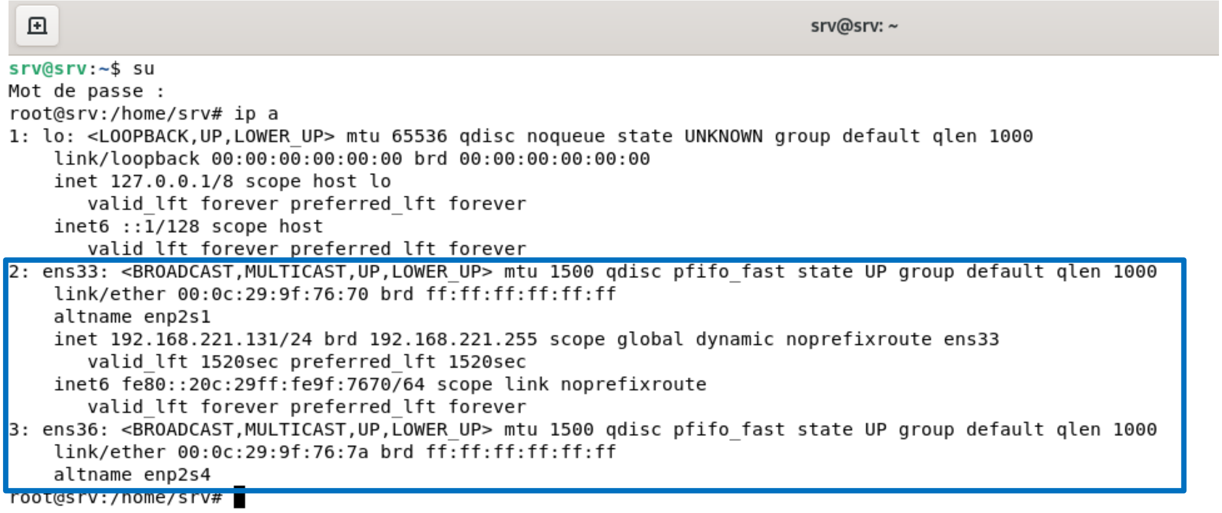
1. Prérequis

1 Machine Linux Serveur
2 Machines Linux Clients

Nom machines	Réseau 1 (NAT)	Réseau 2 (LAN)
Serveur	192.168.221.131	10.0.0.1
Client 1		10.0.0.10
Client 2		10.0.0.11

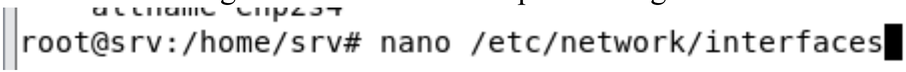
2. Configuration du serveur DHCP

Tout d'abord nous allons ajouter une carte réseau en LAN.
Nous avons 2 cartes ens33 et ens36.

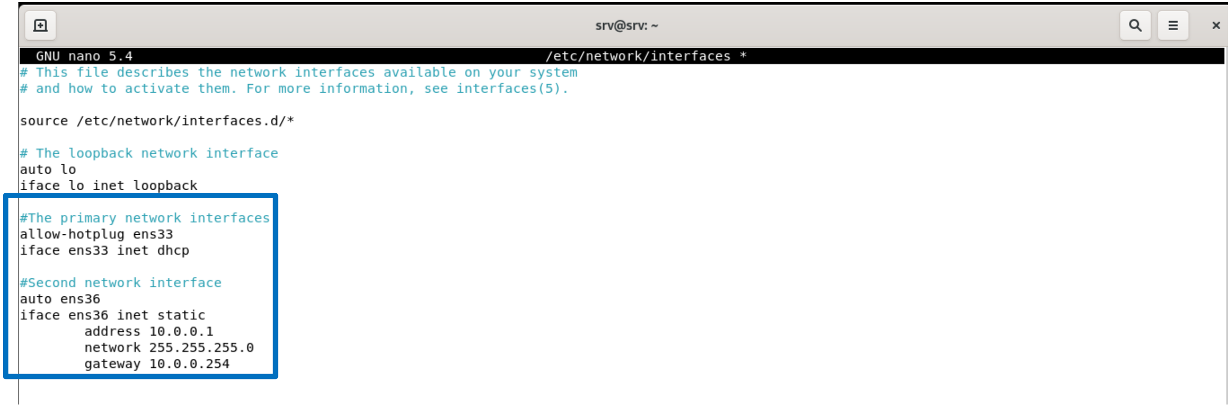


```
srv@srv:~$ su
Mot de passe :
root@srv:/home/srv# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:9f:76:70 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.221.131/24 brd 192.168.221.255 scope global dynamic noprefixroute ens33
       valid_lft 1520sec preferred_lft 1520sec
    inet6 fe80::20c:29ff:fe9f:7670/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:9f:76:7a brd ff:ff:ff:ff:ff:ff
    altname enp2s4
root@srv:/home/srv#
```

Nous allons éditer le fichier de configuration des interfaces pour configurer nos addresses pour chaque carte.

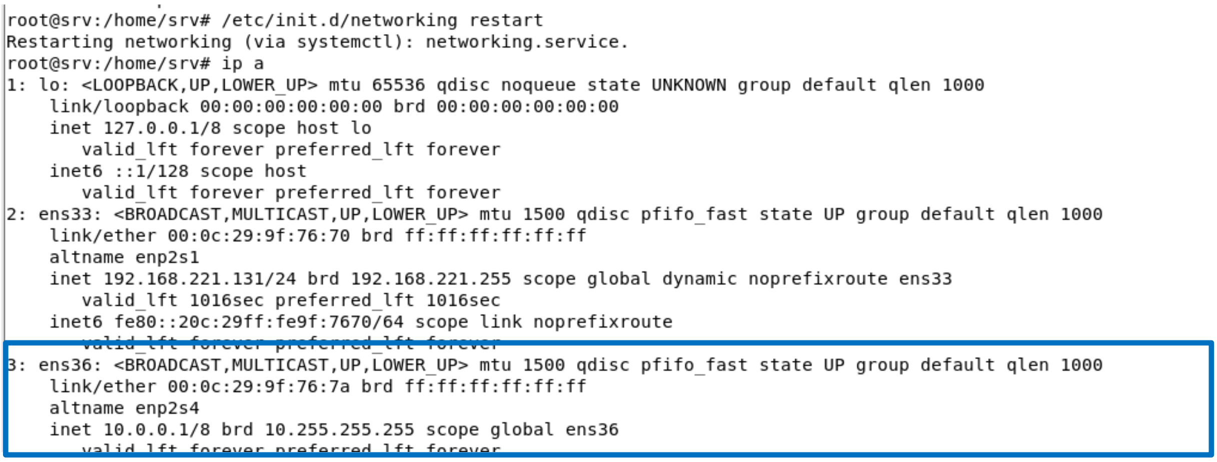


```
root@srv:/home/srv# nano /etc/network/interfaces
```



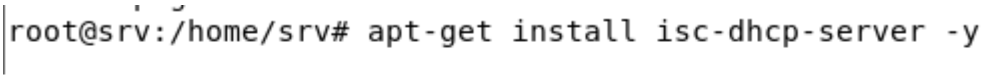
```
  GNU nano 5.4                          /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

#The primary network interfaces
allow-hotplug ens33
iface ens33 inet dhcp

#Second network interface
auto ens36
iface ens36 inet static
        address 10.0.0.1
        network 255.255.255.0
        gateway 10.0.0.254
```

Ensuite nous allons redemarer le service de mise en réseau du serveur.



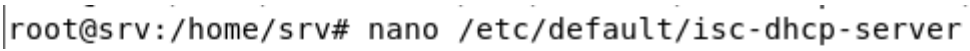
```
root@srv:/home/srv# /etc/init.d/networking restart
Restarting networking (via systemctl): networking.service.
root@srv:/home/srv# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:9f:76:70 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.221.131/24 brd 192.168.221.255 scope global dynamic noprefixroute ens33
       valid_lft 1016sec preferred_lft 1016sec
    inet6 fe80::20c:29ff:fe9f:7670/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:9f:76:7a brd ff:ff:ff:ff:ff:ff
    altname enp2s4
    inet 10.0.0.1/8 brd 10.255.255.255 scope global ens36
       valid_lft forever preferred_lft forever
```

Nous allons installer un paquet.

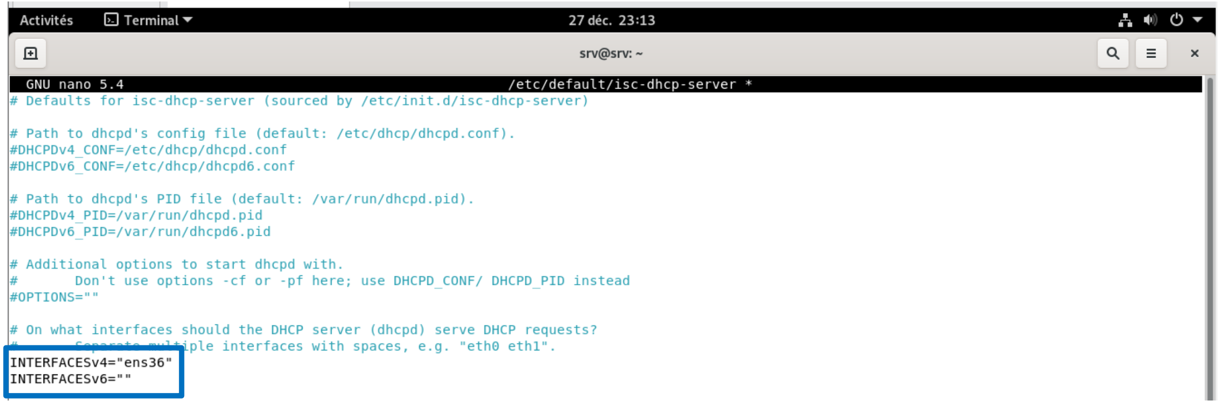


```
root@srv:/home/srv# apt-get install isc-dhcp-server -y
```

Ensuite, nous allons spécifier dans un fichier de configuration sur quelle interface réseau le serveur DHCP doit être écouter.

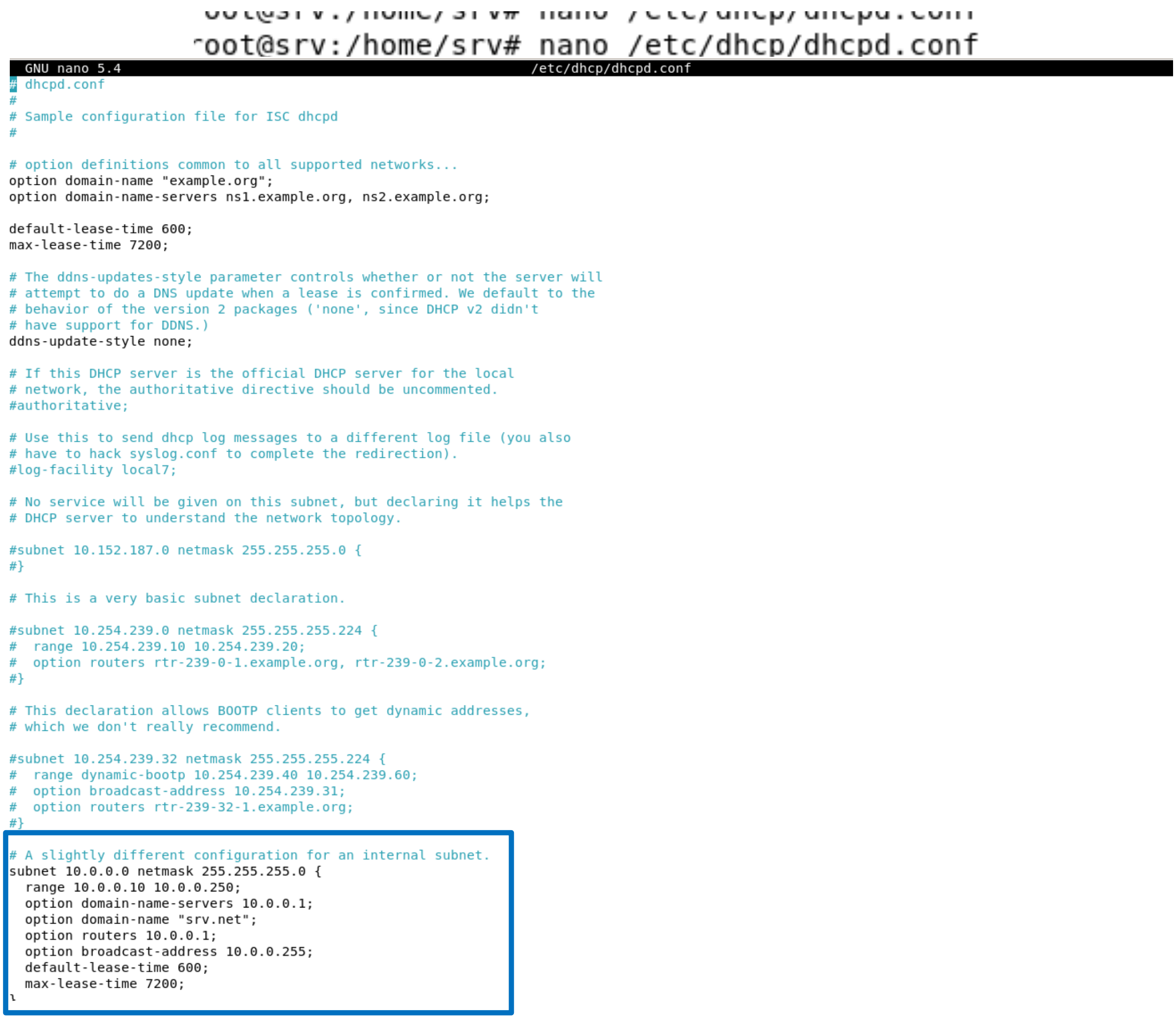


```
root@srv:/home/srv# nano /etc/default/isc-dhcp-server
```



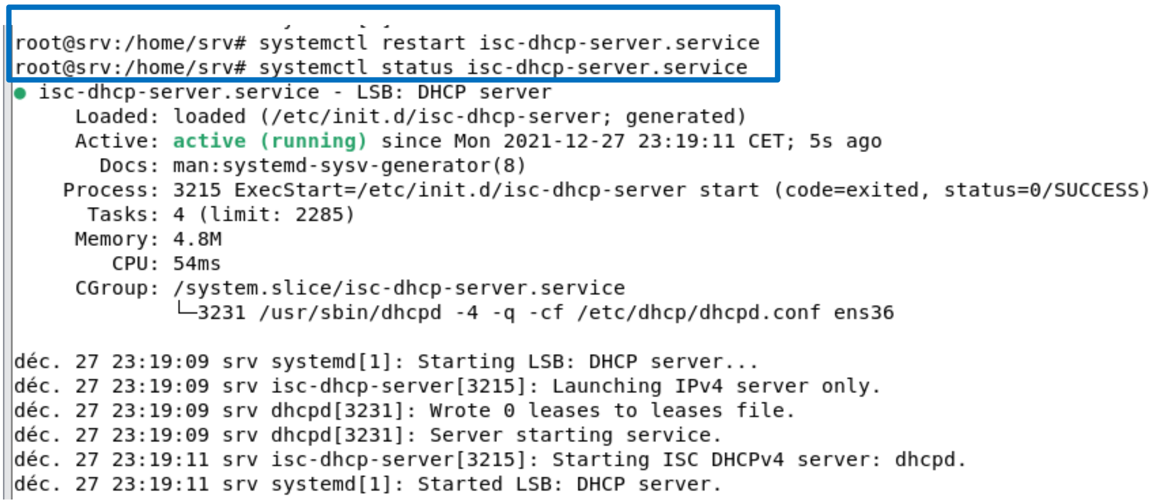
```
  GNU nano 5.4                                      /etc/default/isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# 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="ens36"
INTERFACESv6=""
```

Nous allons ouvrir le fichier de configuration DHCP. Ce fichier permet de stocker des informations réseaux.



```
oot@srv:/home/srv# nano /etc/dhcp/dhcpd.conf
  GNU nano 5.4                                /etc/dhcp/dhcpd.conf
# 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;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
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;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

#subnet 10.254.239.0 netmask 255.255.255.224 {
#  range 10.254.239.10 10.254.239.20;
#  option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 10.0.0.0 netmask 255.255.255.0 {
  range 10.0.0.10 10.0.0.250;
  option domain-name-servers 10.0.0.1;
  option domain-name "srv.net";
  option routers 10.0.0.1;
  option broadcast-address 10.0.0.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

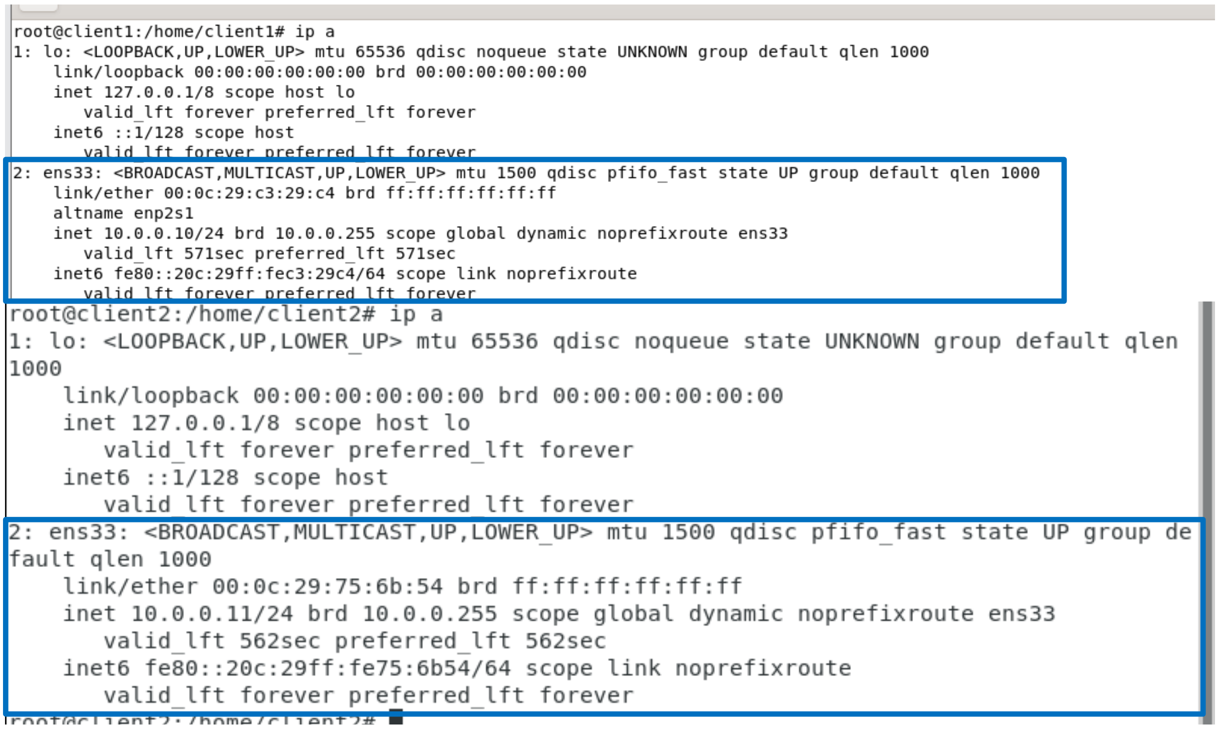
Nous allons maintenant demander à notre serveur de prendre en compte toutes les modifications.



```
root@srv:/home/srv# systemctl restart isc-dhcp-server.service
root@srv:/home/srv# 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 Mon 2021-12-27 23:19:11 CET; 5s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 3215 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
      Tasks: 4 (limit: 2285)
     Memory: 4.8M
        CPU: 54ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─3231 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf ens36

déc. 27 23:19:09 srv systemd[1]: Starting LSB: DHCP server...
déc. 27 23:19:09 srv isc-dhcp-server[3215]: Launching IPv4 server only.
déc. 27 23:19:09 srv dhcpd[3231]: Wrote 0 leases to leases file.
déc. 27 23:19:09 srv dhcpd[3231]: Server starting service.
déc. 27 23:19:11 srv isc-dhcp-server[3215]: Starting ISC DHCPv4 server: dhcpd.
déc. 27 23:19:11 srv systemd[1]: Started LSB: DHCP server.
```

3. Client

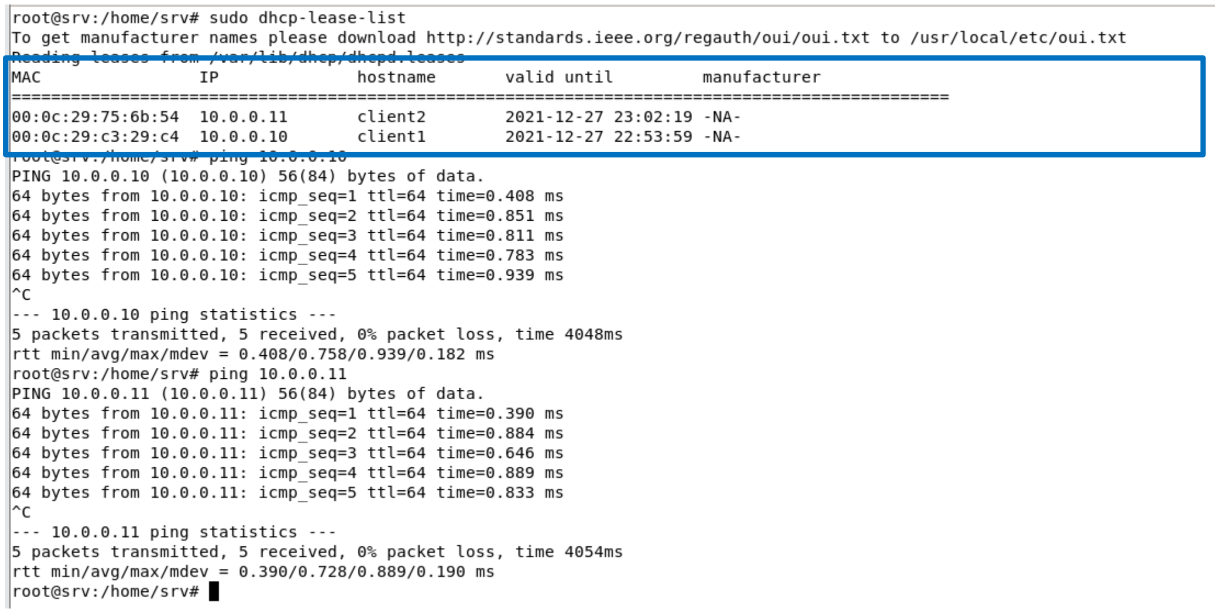
Nos 2 machines clientes arrivent bien avoir une adresse IP distribuer automatiquement



```
root@client1:/home/client1# 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:c3:29:c4 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 10.0.0.10/24 brd 10.0.0.255 scope global dynamic noprefixroute ens33
       valid_lft 571sec preferred_lft 571sec
    inet6 fe80::20c:29ff:fec3:29c4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
root@client2:/home/client2# 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group de
fault qlen 1000
    link/ether 00:0c:29:75:6b:54 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.11/24 brd 10.0.0.255 scope global dynamic noprefixroute ens33
       valid_lft 562sec preferred_lft 562sec
    inet6 fe80::20c:29ff:fe75:6b54/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
root@client2:/home/client2#
```

A l'aide de la commande « sudo dhcp-lease-list » nous pouvons afficher la liste des baux attribués aux clients DHCP.



```
root@srv:/home/srv# sudo dhcp-lease-list
To get manufacturer names please download http://standards.ieee.org/regauth/oui/oui.txt to /usr/local/etc/oui.txt
Reading leases from /var/lib/dhcp/dhcpd.leases
MAC                IP              hostname       valid until         manufacturer
===============================================================================================
00:0c:29:75:6b:54  10.0.0.11       client2        2021-12-27 23:02:19 -NA-
00:0c:29:c3:29:c4  10.0.0.10       client1        2021-12-27 22:53:59 -NA-
root@srv:/home/srv# ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=0.408 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.851 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.811 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.783 ms
64 bytes from 10.0.0.10: icmp_seq=5 ttl=64 time=0.939 ms
^C
--- 10.0.0.10 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4048ms
rtt min/avg/max/mdev = 0.408/0.758/0.939/0.182 ms
root@srv:/home/srv# ping 10.0.0.11
PING 10.0.0.11 (10.0.0.11) 56(84) bytes of data.
64 bytes from 10.0.0.11: icmp_seq=1 ttl=64 time=0.390 ms
64 bytes from 10.0.0.11: icmp_seq=2 ttl=64 time=0.884 ms
64 bytes from 10.0.0.11: icmp_seq=3 ttl=64 time=0.646 ms
64 bytes from 10.0.0.11: icmp_seq=4 ttl=64 time=0.889 ms
64 bytes from 10.0.0.11: icmp_seq=5 ttl=64 time=0.833 ms
^C
--- 10.0.0.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4054ms
rtt min/avg/max/mdev = 0.390/0.728/0.889/0.190 ms
root@srv:/home/srv# 
```

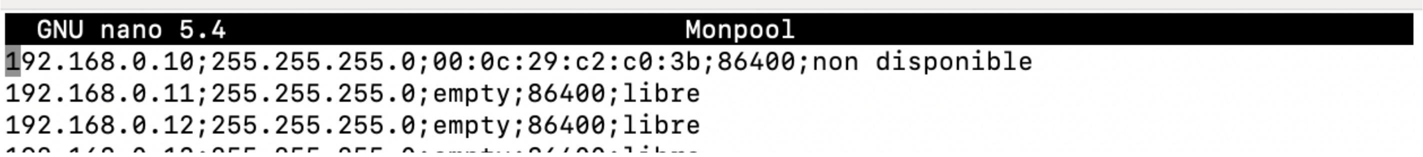
Mini serveur DHCP

1. Préréquis

1 Machine Linux Serveur
2 Machines Linux Clients

2. Format du fichier Monpool

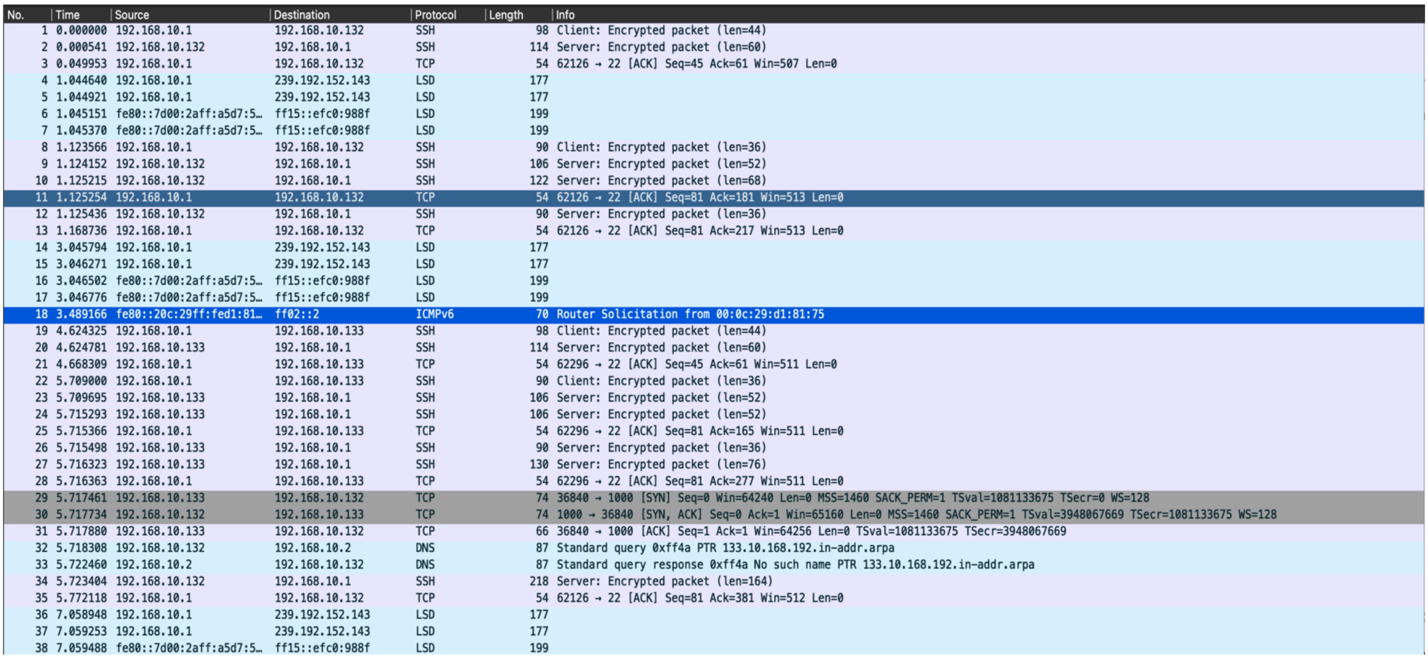
Le fichier est composé :
Adresse IP ; Masque ; MAC Adresse ; bail ; libre ou non disponible



```
 GNU nano 5.4                                 Monpool
192.168.0.10;255.255.255.0;00:0c:29:c2:c0:3b;86400;non disponible
192.168.0.11;255.255.255.0;empty;86400;libre
192.168.0.12;255.255.255.0;empty;86400;libre
192.168.0.13;255.255.255.0;empty;86400;libre
```

3. Format des requêtes échangées entre client et serveur

Nous avons plusieurs colonnes. Nous allons nous intéresser à « Source », « Destination » et « Protocol » SSH, TCP.
192.168.10.132 → adresse IP Serveur
192.168.10.133 → adresse IP Client
192.168.10.1 → Passerelle (Carte réseaux VMware)

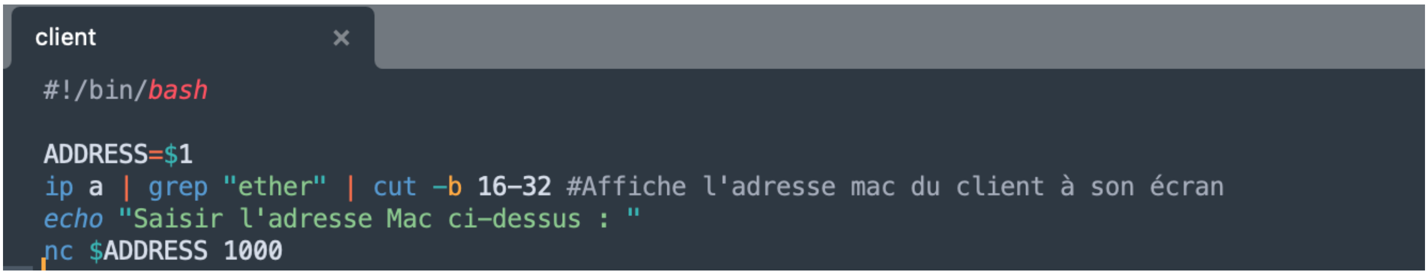


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.1	192.168.10.132	SSH	98	Client: Encrypted packet (len=44)
2	0.000541	192.168.10.132	192.168.10.1	SSH	114	Server: Encrypted packet (len=60)
3	0.049953	192.168.10.1	192.168.10.132	TCP	54	62126 → 22 [ACK] Seq=45 Ack=61 Win=507 Len=0
4	1.044640	192.168.10.1	239.192.152.143	LSD	177	
5	1.044921	192.168.10.1	239.192.152.143	LSD	177	
6	1.045151	fe80::7d00:2aff:a5d7:5…	ff15::efc0:988f	LSD	199	
7	1.045370	fe80::7d00:2aff:a5d7:5…	ff15::efc0:988f	LSD	199	
8	1.123566	192.168.10.1	192.168.10.132	SSH	90	Client: Encrypted packet (len=36)
9	1.124152	192.168.10.132	192.168.10.1	SSH	106	Server: Encrypted packet (len=52)
10	1.125215	192.168.10.132	192.168.10.1	SSH	122	Server: Encrypted packet (len=68)
11	1.125254	192.168.10.1	192.168.10.132	TCP	54	62126 → 22 [ACK] Seq=81 Ack=181 Win=513 Len=0
12	1.125436	192.168.10.132	192.168.10.1	SSH	90	Server: Encrypted packet (len=36)
13	1.168736	192.168.10.1	192.168.10.132	TCP	54	62126 → 22 [ACK] Seq=81 Ack=217 Win=513 Len=0
14	3.045794	192.168.10.1	239.192.152.143	LSD	177	
15	3.046271	192.168.10.1	239.192.152.143	LSD	177	
16	3.046502	fe80::7d00:2aff:a5d7:5…	ff15::efc0:988f	LSD	199	
17	3.046776	fe80::7d00:2aff:a5d7:5…	ff15::efc0:988f	LSD	199	
18	3.489166	fe80::20c:29ff:fed1:81…	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:d1:81:75
19	4.624325	192.168.10.1	192.168.10.133	SSH	98	Client: Encrypted packet (len=44)
20	4.624781	192.168.10.133	192.168.10.1	SSH	114	Server: Encrypted packet (len=60)
21	4.668309	192.168.10.1	192.168.10.133	TCP	54	62296 → 22 [ACK] Seq=45 Ack=61 Win=511 Len=0
22	5.709000	192.168.10.1	192.168.10.133	SSH	90	Client: Encrypted packet (len=36)
23	5.709695	192.168.10.133	192.168.10.1	SSH	106	Server: Encrypted packet (len=52)
24	5.715293	192.168.10.133	192.168.10.1	SSH	106	Server: Encrypted packet (len=52)
25	5.715366	192.168.10.1	192.168.10.133	TCP	54	62296 → 22 [ACK] Seq=81 Ack=165 Win=511 Len=0
26	5.715498	192.168.10.133	192.168.10.1	SSH	90	Server: Encrypted packet (len=36)
27	5.716323	192.168.10.133	192.168.10.1	SSH	130	Server: Encrypted packet (len=76)
28	5.716363	192.168.10.1	192.168.10.133	TCP	54	62296 → 22 [ACK] Seq=81 Ack=277 Win=511 Len=0
29	5.717461	192.168.10.133	192.168.10.132	TCP	74	36840 → 1000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1081133675 TSecr=0 WS=128
30	5.717734	192.168.10.132	192.168.10.133	TCP	74	1000 → 36840 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3948067669 TSecr=1081133675 WS=128
31	5.717880	192.168.10.133	192.168.10.132	TCP	66	36840 → 1000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1081133675 TSecr=3948067669
32	5.718308	192.168.10.132	192.168.10.2	DNS	87	Standard query 0xff4a PTR 133.10.168.192.in-addr.arpa
33	5.722460	192.168.10.2	192.168.10.132	DNS	87	Standard query response 0xff4a No such name PTR 133.10.168.192.in-addr.arpa
34	5.723404	192.168.10.132	192.168.10.1	SSH	218	Server: Encrypted packet (len=164)
35	5.772118	192.168.10.1	192.168.10.132	TCP	54	62126 → 22 [ACK] Seq=81 Ack=381 Win=512 Len=0
36	7.058948	192.168.10.1	239.192.152.143	LSD	177	
37	7.059253	192.168.10.1	239.192.152.143	LSD	177	
38	7.059488	fe80::7d00:2aff:a5d7:5…	ff15::efc0:988f	LSD	199	

4. Copie des écrans prouvant le bon fonctionnement

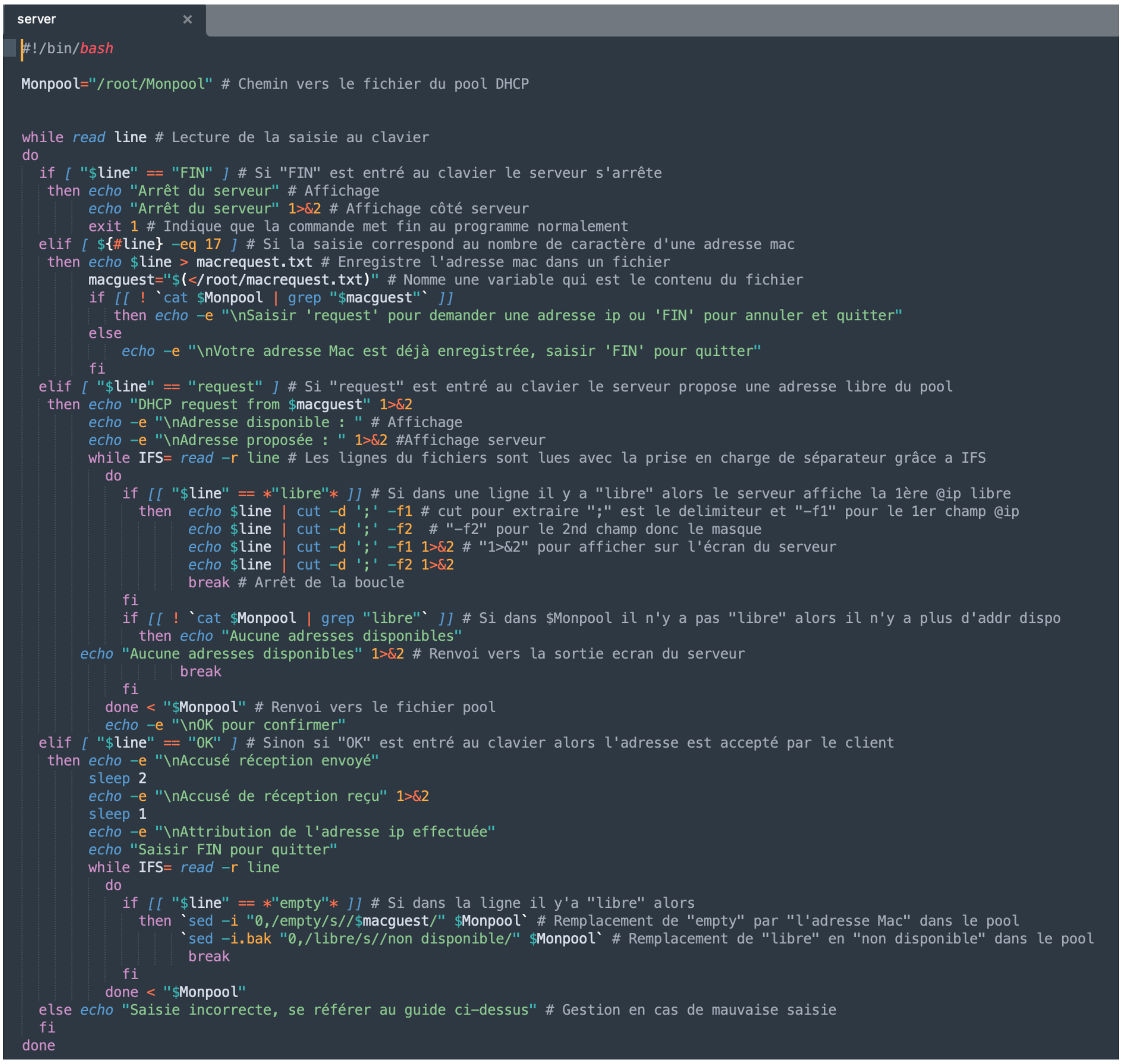
a) Serveur et client dans la même machine virtuelle

Script client :



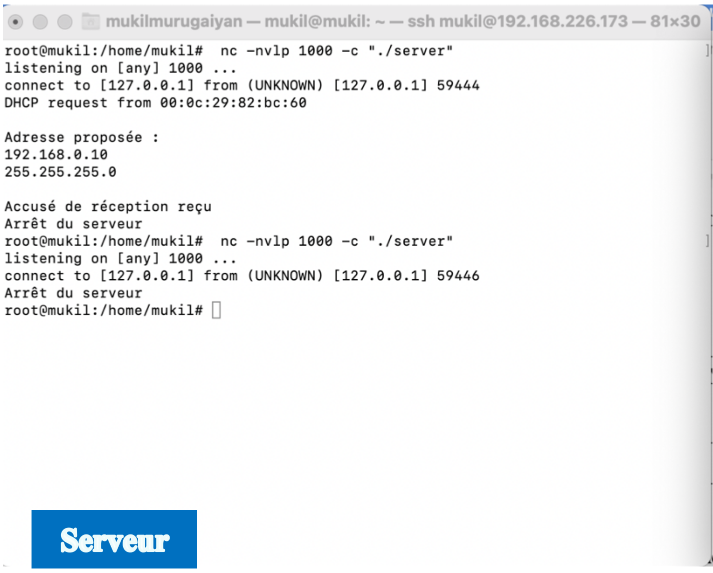
```
#!/bin/bash

ADDRESS=$1
ip a | grep "ether" | cut -b 16-32 #Affiche l'adresse mac du client à son écran
echo "Saisir l'adresse Mac ci-dessus : "
nc $ADDRESS 1000
```

Script serveur :

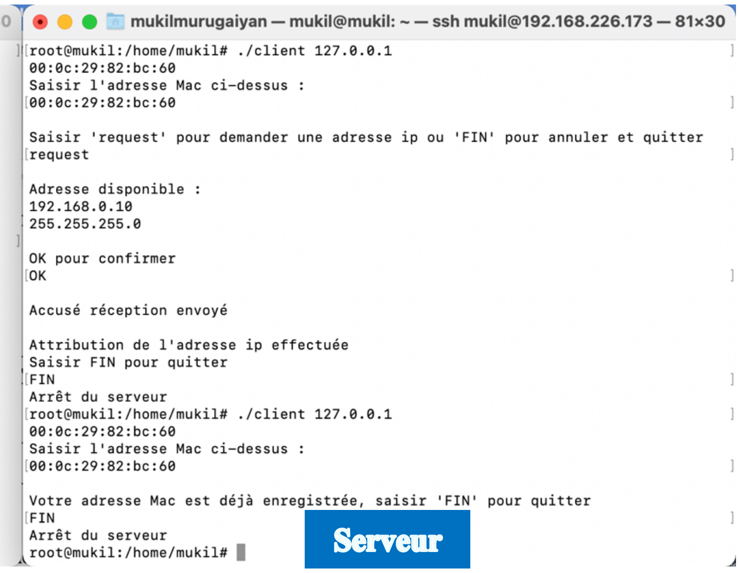


```
#!/bin/bash

Monpool="/root/Monpool" # Chemin vers le fichier du pool DHCP


while read line # Lecture de la saisie au clavier
do
  if [ "$line" == "FIN" ] # Si "FIN" est entré au clavier le serveur s'arrête
   then echo "Arrêt du serveur" # Affichage
        echo "Arrêt du serveur" 1>&2 # Affichage côté serveur
        exit 1 # Indique que la commande met fin au programme normalement
  elif [ ${#line} -eq 17 ] # Si la saisie correspond au nombre de caractère d'une adresse mac
   then echo $line > macrequest.txt # Enregistre l'adresse mac dans un fichier
        macguest="$(</root/macrequest.txt)" # Nomme une variable qui est le contenu du fichier
        if [[ ! `cat $Monpool | grep "$macguest"` ]]
           then echo -e "\nSaisir 'request' pour demander une adresse ip ou 'FIN' pour annuler et quitter"
        else
            echo -e "\nVotre adresse Mac est déjà enregistrée, saisir 'FIN' pour quitter"
        fi
  elif [ "$line" == "request" ] # Si "request" est entré au clavier le serveur propose une adresse libre du pool
   then echo "DHCP request from $macguest" 1>&2
        echo -e "\nAdresse disponible : " # Affichage
        echo -e "\nAdresse proposée : " 1>&2 #Affichage serveur
        while IFS= read -r line # Les lignes du fichiers sont lues avec la prise en charge de séparateur grâce a IFS
          do
            if [[ "$line" == *"libre"* ]] # Si dans une ligne il y a "libre" alors le serveur affiche la 1ère @ip libre
              then  echo $line | cut -d ';' -f1 # cut pour extraire ";" est le delimiteur et "-f1" pour le 1er champ @ip
                    echo $line | cut -d ';' -f2  # "-f2" pour le 2nd champ donc le masque
                    echo $line | cut -d ';' -f1 1>&2 # "1>&2" pour afficher sur l'écran du serveur
                    echo $line | cut -d ';' -f2 1>&2
                    break # Arrêt de la boucle
            fi
            if [[ ! `cat $Monpool | grep "libre"` ]] # Si dans $Monpool il n'y a pas "libre" alors il n'y a plus d'addr dispo
              then echo "Aucune adresses disponibles"
       echo "Aucune adresses disponibles" 1>&2 # Renvoi vers la sortie ecran du serveur
                   break
            fi
          done < "$Monpool" # Renvoi vers le fichier pool
          echo -e "\nOK pour confirmer"
  elif [ "$line" == "OK" ] # Sinon si "OK" est entré au clavier alors l'adresse est accepté par le client
   then echo -e "\nAccusé réception envoyé"
        sleep 2
        echo -e "\nAccusé de réception reçu" 1>&2
        sleep 1
        echo -e "\nAttribution de l'adresse ip effectuée"
        echo "Saisir FIN pour quitter"
        while IFS= read -r line
          do
            if [[ "$line" == *"empty"* ]] # Si dans la ligne il y'a "libre" alors
              then `sed -i "0,/empty/s//$macguest/" $Monpool` # Remplacement de "empty" par "l'adresse Mac" dans le pool
                   `sed -i.bak "0,/libre/s//non disponible/" $Monpool` # Remplacement de "libre" en "non disponible" dans le pool
                    break
            fi
          done < "$Monpool"
  else echo "Saisie incorrecte, se référer au guide ci-dessus" # Gestion en cas de mauvaise saisie
  fi
done
```

Exécution des clients et du serveur :

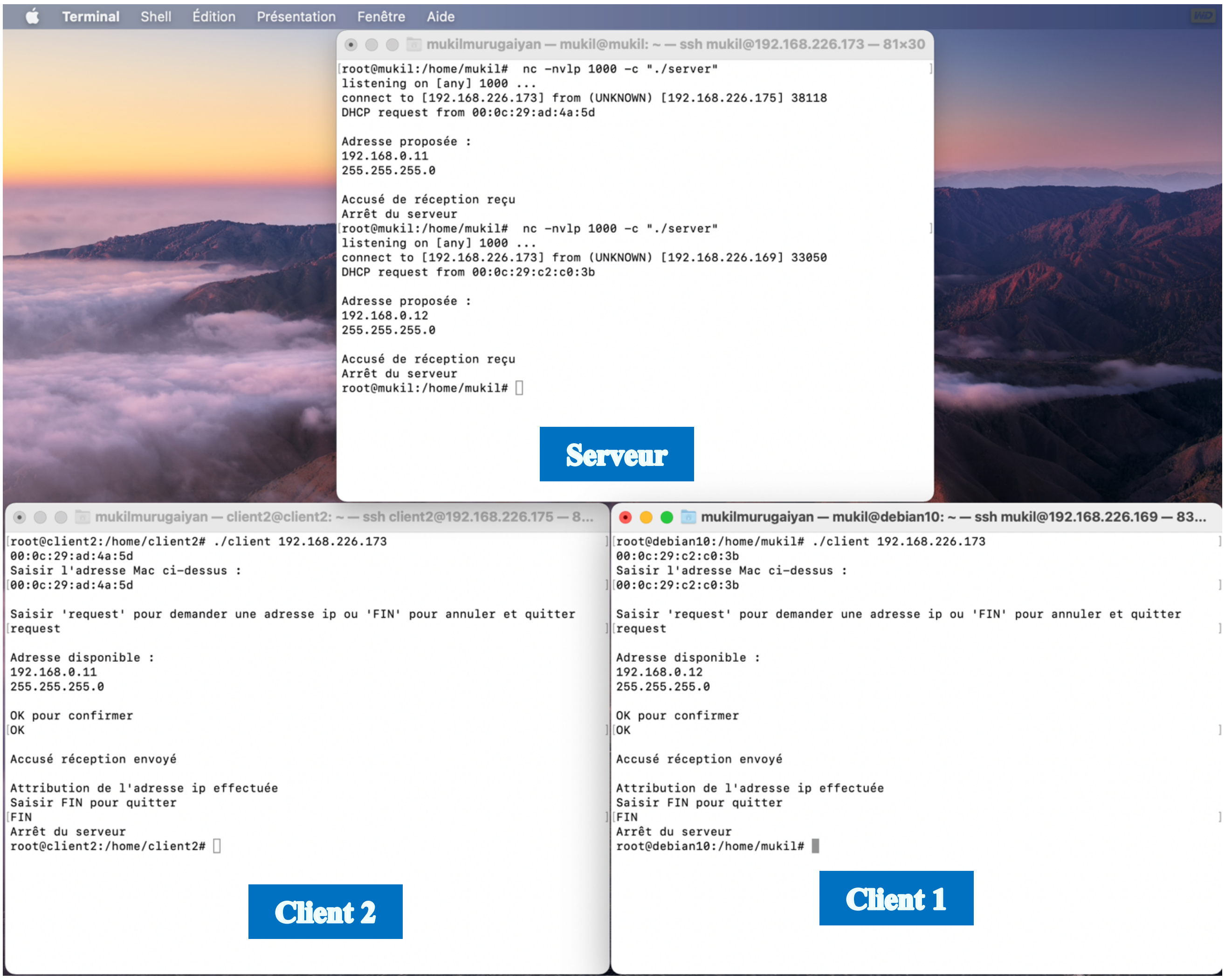

mukilmurugaiyan — mukil@mukil: ~ — ssh mukil@192.168.226.173 — 81×30
Serveur

```
root@mukil:/home/mukil#  nc -nvlp 1000 -c "./server"
listening on [any] 1000 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 59444
DHCP request from 00:0c:29:82:bc:60

Adresse proposée :
192.168.0.10
255.255.255.0

Accusé de réception reçu
Arrêt du serveur
root@mukil:/home/mukil#  nc -nvlp 1000 -c "./server"
listening on [any] 1000 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 59446
Arrêt du serveur
root@mukil:/home/mukil# 
```


mukilmurugaiyan — mukil@mukil: ~ — ssh mukil@192.168.226.173 — 81×30
Serveur

```
root@mukil:/home/mukil# ./client 127.0.0.1
00:0c:29:82:bc:60
Saisir l'adresse Mac ci-dessus :
00:0c:29:82:bc:60

Saisir 'request' pour demander une adresse ip ou 'FIN' pour annuler et quitter
request

Adresse disponible :
192.168.0.10
255.255.255.0

OK pour confirmer
OK

Accusé réception envoyé

Attribution de l'adresse ip effectuée
Saisir FIN pour quitter
FIN
Arrêt du serveur
root@mukil:/home/mukil# ./client 127.0.0.1
00:0c:29:82:bc:60
Saisir l'adresse Mac ci-dessus :
00:0c:29:82:bc:60

Votre adresse Mac est déjà enregistrée, saisir 'FIN' pour quitter
FIN
Arrêt du serveur
root@mukil:/home/mukil# 
```

b) Serveur et clients dans des machines distinctes


Terminal
Shell
Édition
Présentation
Fenêtre
Aide
mukilmurugaiyan — mukil@mukil: ~ — ssh mukil@192.168.226.173 — 81×30
Serveur
mukilmurugaiyan — client2@client2: ~ — ssh client2@192.168.226.175 — 8...
Client 2
mukilmurugaiyan — mukil@debian10: ~ — ssh mukil@192.168.226.169 — 83...
Client 1

```
root@mukil:/home/mukil#  nc -nvlp 1000 -c "./server"
listening on [any] 1000 ...
connect to [192.168.226.173] from (UNKNOWN) [192.168.226.175] 38118
DHCP request from 00:0c:29:ad:4a:5d

Adresse proposée :
192.168.0.11
255.255.255.0

Accusé de réception reçu
Arrêt du serveur
root@mukil:/home/mukil#  nc -nvlp 1000 -c "./server"
listening on [any] 1000 ...
connect to [192.168.226.173] from (UNKNOWN) [192.168.226.169] 33050
DHCP request from 00:0c:29:c2:c0:3b

Adresse proposée :
192.168.0.12
255.255.255.0

Accusé de réception reçu
Arrêt du serveur
root@mukil:/home/mukil# 
```

```
root@client2:/home/client2# ./client 192.168.226.173
00:0c:29:ad:4a:5d
Saisir l'adresse Mac ci-dessus :
00:0c:29:ad:4a:5d

Saisir 'request' pour demander une adresse ip ou 'FIN' pour annuler et quitter
request

Adresse disponible :
192.168.0.11
255.255.255.0

OK pour confirmer
OK

Accusé réception envoyé

Attribution de l'adresse ip effectuée
Saisir FIN pour quitter
FIN
Arrêt du serveur
root@client2:/home/client2# 
```

```
root@debian10:/home/mukil# ./client 192.168.226.173
00:0c:29:c2:c0:3b
Saisir l'adresse Mac ci-dessus :
00:0c:29:c2:c0:3b

Saisir 'request' pour demander une adresse ip ou 'FIN' pour annuler et quitter
request

Adresse disponible :
192.168.0.12
255.255.255.0

OK pour confirmer
OK

Accusé réception envoyé

Attribution de l'adresse ip effectuée
Saisir FIN pour quitter
FIN
Arrêt du serveur
root@debian10:/home/mukil# 
```