



Administración de Sistemas Operativos y Redes de Computadores 2021-22

Práctica 2

Nikita Polyanskiy

Y4441167L

Índice

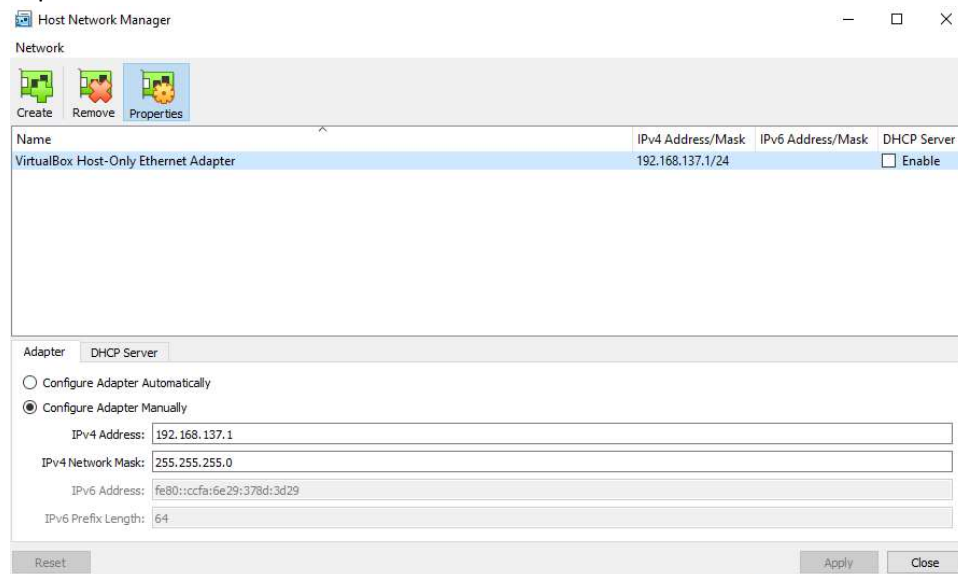
1.Unix (FreeBSD)	3
1.1 Antes de comenzar	3
1.2 SSH + SFTP + SCP	4
1.2.1 SSH	4
1.2.2 SFTP	4
1.2.3 SCP	4
1.3 VNC:	5
1.4 RDP:	7
1.5 DNS	9
1.6 DHCP	11
1.7 NFS	12
1.8 SAMBA	14
2. Rocky Linux	16
2.1 SSH	16
2.2 SFTP	16
2.3 SCP	16
2.4 VNC	17
2.5 RDP	18
2.6 DNS	19
2.7 DHCP	22
2.8 NFS	24
2.9 SAMBA	25
3. Windows Server 2022	27
3.1 SSH + SFTP + SCP	27
3.2 VNC	28
3.3 RDP	29
3.4 DNS	32
3.5 DHCP	34
3.6 NFS	37
3.7 SAMBA	40

1.Unix (FreeBSD)

1.1 Antes de comenzar

Debemos instalar un entorno de escritorio, no se recomienda GNOME ya que no funcionan algunos protocolos de comunicación. Por lo que instalaremos XFCE, las instrucciones se pueden encontrar en la [pagina oficial de FreeBSD](#) .

Antes que nada, deberemos desactivar el servicio DHCP en los ajustes de red de anfitrión, y configuraremos el adaptador manualmente con la IP que aparece en la captura:



Y en FreeBSD configuraremos la IP estática 192.168.137.221 configurando /etc/rc.conf para que se vea así:

```
hostname="np31"
keymap="es.kbd"
ifconfig_em0="DHCP"
ifconfig_em0_ipv6="inet6 accept_rtadv"
ifconfig_em1="192.168.137.221"
sshd_enable="YES"
moused_enable="YES"
ntpd_enable="YES"
powerd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
zfs_enable="YES"
```

1.2 SSH + SFTP + SCP

1.2.1 SSH

Primero modificamos el archivo `/etc/ssh/sshd_config`, descomentando la línea

`PermitRootLogin no`

`PermitEmptyPasswords no`

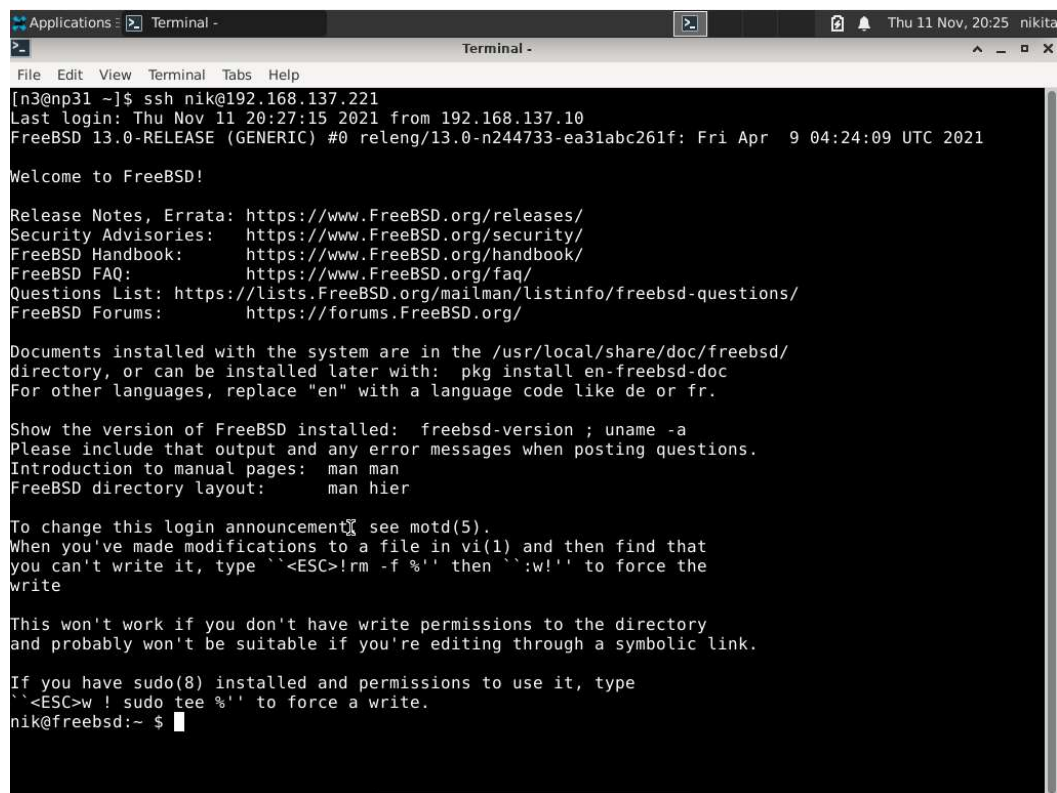
Y agregamos la línea `AllowUsers n3`

Desde nuestro cliente, en mi caso desde otra máquina virtual con FreeBSD, generamos una clave con el siguiente comando, y luego copiamos la clave publica a nuestro usuario de FreeBSD:

`ssh-keygen -t rsa`

`ssh-copy-id -i .ssh/id_rsa.pub nik@192.168.137.221`

Una vez hecho esto podremos utilizar ssh para conectarnos a nuestro servidor en FreeBSD sin la necesidad de utilizar la contraseña:



```
Applications - Terminal -
Terminal -
File Edit View Terminal Tabs Help
[n3@np31 ~]$ ssh nik@192.168.137.221
Last login: Thu Nov 11 20:27:15 2021 from 192.168.137.10
FreeBSD 13.0-RELEASE (GENERIC) #0 releng/13.0-n244733-ea31abc261f: Fri Apr  9 04:24:09 UTC 2021

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories:  https://www.FreeBSD.org/security/
FreeBSD Handbook:     https://www.FreeBSD.org/handbook/
FreeBSD FAQ:          https://www.FreeBSD.org/faq/
Questions List:       https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/
FreeBSD Forums:       https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with:  pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.

Show the version of FreeBSD installed:  freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages:  man man
FreeBSD directory layout:      man hier

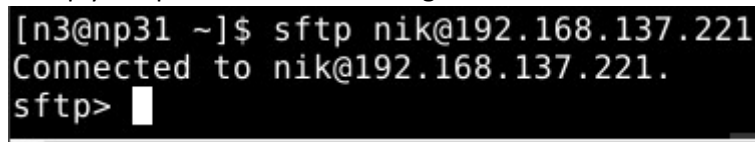
To change this login announcement see motd(5).
When you've made modifications to a file in vi(1) and then find that
you can't write it, type ``<ESC>!rm -f %' then ``:w!' to force the
write

This won't work if you don't have write permissions to the directory
and probably won't be suitable if you're editing through a symbolic link.

If you have sudo(8) installed and permissions to use it, type
``<ESC>w ! sudo tee %' to force a write.
nik@freebsd:~$ █
```

1.2.2 SFTP

El sftp ya se puede utilizar con el siguiente comando una vez hecho los ajustes de ssh:



```
[n3@np31 ~]$ sftp nik@192.168.137.221
Connected to nik@192.168.137.221.
sftp> █
```

1.2.3 SCP

SCP ya se puede utilizar con el siguiente comando una vez hecho los ajustes de ssh:



```
[n3@np31 ~]$ scp prueba.txt nik@192.168.137.221:/home/nik/
prueba.txt                                100%   0    0.0KB/s   00:00
[n3@np31 ~]$ █
```

1.3 VNC:

Primero instalamos el paquete tightvnc:

```
pkg install tightvnc
```

luego iniciamos el servicio vncserver:

```
$ vncserver
xauth: (argv):1: bad display name "freebsd:1" in "add" command

New 'X' desktop is freebsd:1

Starting applications specified in /home/nik/.vnc/xstartup
Log file is /home/nik/.vnc/freebsd:1.log

$ █
```

Editamos el archivo `/.vnc/startup` que se encuentra en la carpeta del usuario, para que quede así:

```
GNU nano 5.8 xstartup
#!/bin/sh

xrdb $HOME/.Xresources
#xsetroot -solid grey
#xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
#twm &

unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS

startxfce4 &
```

Reiniciamos el servidor vnc:

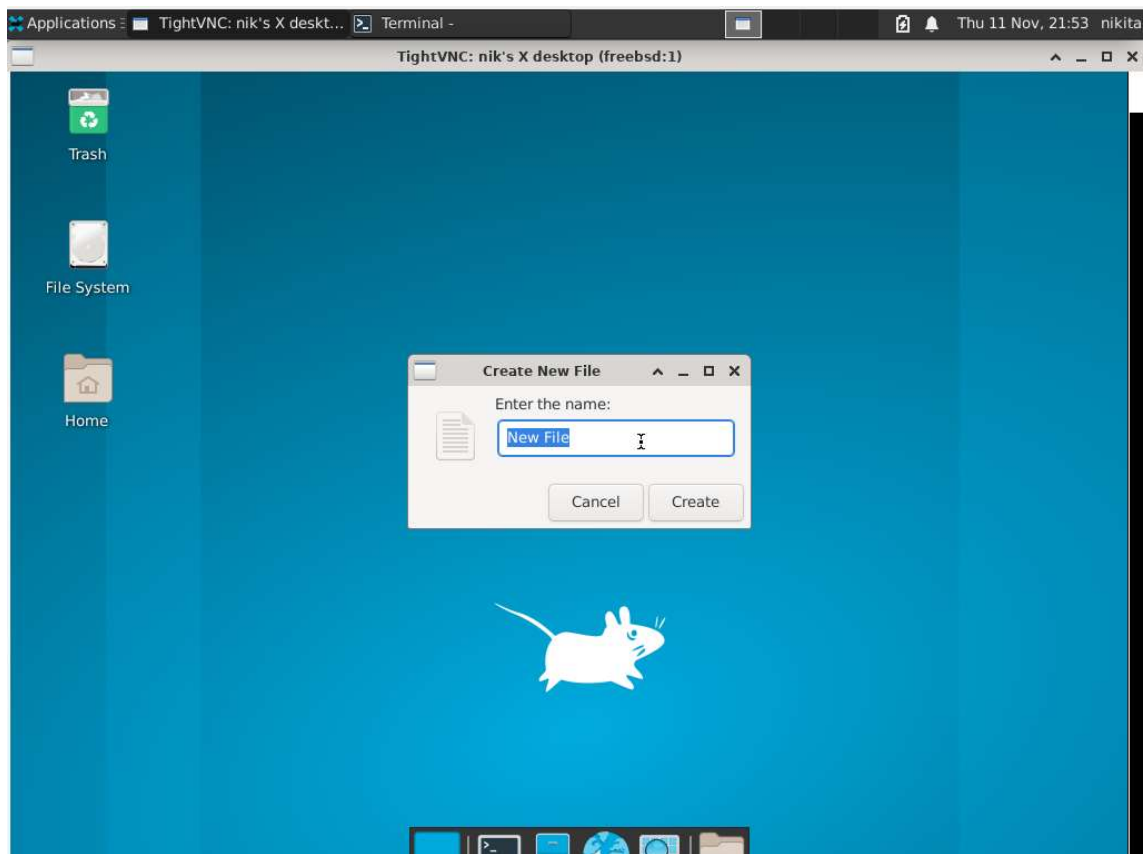
```
vncserver -kill :1
```

```
vncserver
```

Desde un cliente iniciamos el servicio vncviewer para poder conectarnos a la maquina servidor:

```
vncviewer 192.168.137.221:1
```

password: 123456



1.4 RDP:

Primero instalamos el paquete xrdp en nuestro servidor FreeBSD:

```
pkg install xrdp
```

Luego agregamos las siguientes líneas al archivo `/etc/rc.conf`:

```
xrdp_enable="YES"
```

```
xrdp_sesman_enable="YES"
```

Luego, como estamos utilizando el entorno de escritorio XFCE, agregamos la siguiente línea al archivo `/usr/local/etc/xrdp/startwm.sh`:

```
exec startxfce4
```

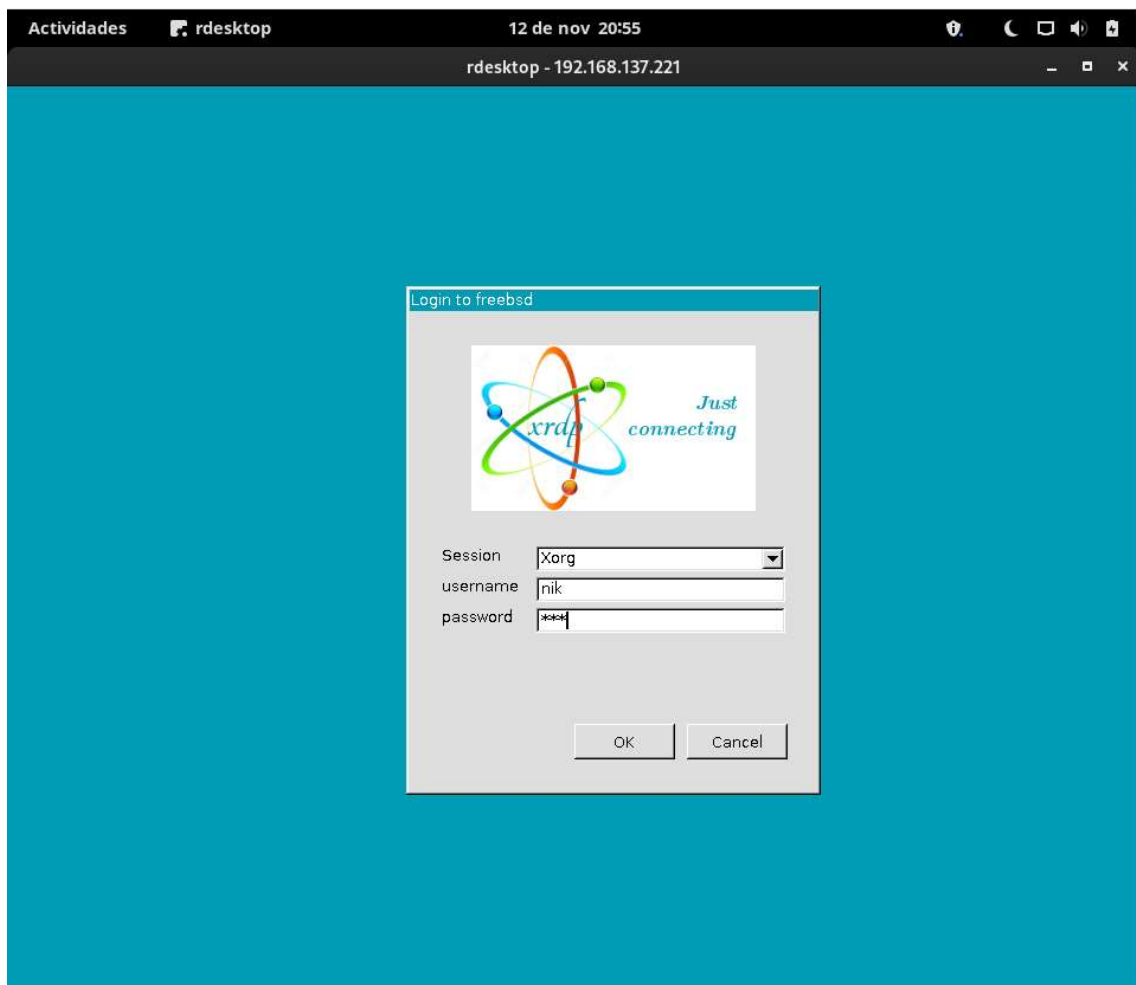
por último reiniciamos el servicio

```
service xrdp restart
```

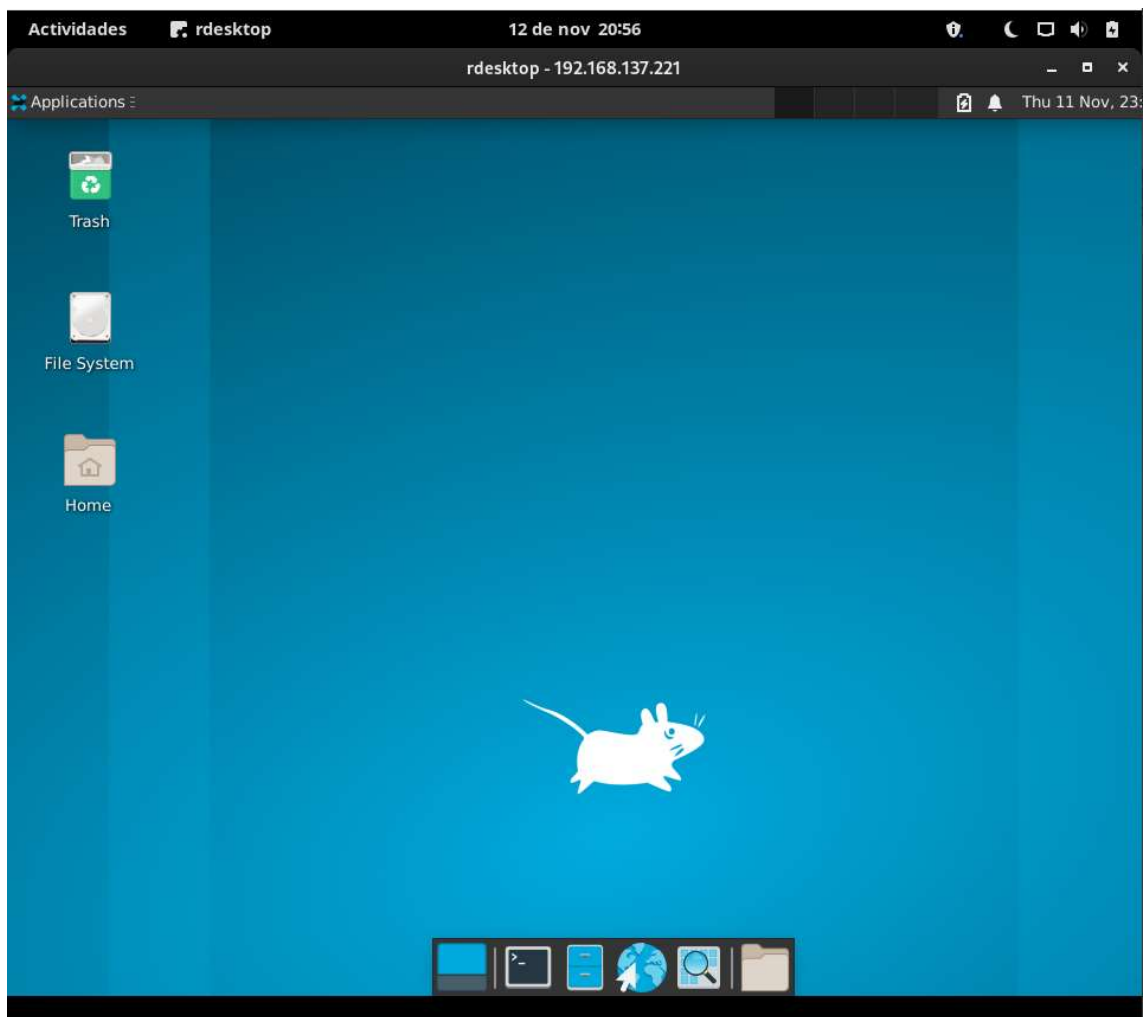
Desde el cliente (en este caso desde Manjaro Linux), primero instalamos un cliente rdp y lo ejecutamos:

```
pacman -S rdesktop
```

```
rdesktop 192.168.137.221
```



Introducimos el usuario y la contraseña para acceder a FreeBSD.



1.5 DNS

Primero instalamos el paquete bind y generamos una clave:

```
pkg install bind911 && rndc-confgen -a
```

Luego modificamos el archivo /etc/rc.conf y añadimos la siguiente línea:

```
named_enable="YES"
```

Luego creamos un fichero zona (en mi caso llamado nik.com.db) en /usr/local/etc/namedb/master/, donde pondremos lo siguiente:

```
GNU nano 5.8 /usr/local/etc/namedb/master/nik.com.db
$TTL 3h
@ SOA nik.com. nobody.nik.com. 42 1d 12h 1w 3h
    ;Serial, Refresh, Retry, Expire, Neg. cache TTL
    NS      ns.nik.com.
    A       192.168.137.221
ns         A       192.168.137.221
www        CNAME   nik.com.
```

Y otro para la búsqueda inversa (nik.com.rev):

```
GNU nano 5.8 nik.com.rev
$TTL 3h
@ SOA nik.com. nobody.nik.com. 42 1d 12h 1w 3h
    ;Serial, Refresh, Retry, Expire, Neg. cache TTL
    NS      ns.nik.com.
    A       192.168.137.221
ns         A       192.168.137.221
@          IN PTR  nik.com.
www        CNAME   nik.com.
221        IN PTR  nik.com.
```

Luego editamos el archivo de configuración /usr/local/etc/namedb/named.conf donde añadiremos nuestra IP en el campo listen-on, y las IP de Google en forwarders:

```
listen-on {127.0.0.1; 192.168.137.221;}
```

```
forwarders {8.8.8.8; 8.8.4.4;}
```

Y luego agregamos nuestra zona:

```
zone "nik.com" {
    type master;
    file "/usr/local/etc/namedb/master/nik.com.db";
};

zone "137.168.192.in-addr_arpa" {
    type master;
    file "/usr/local/etc/namedb/master/nik.com.rev";
};
```

Crearemos un nuevo archivo `/etc/resolv.conf.head` donde indicaremos el ip de nuestro servidor:

`nameserver 192.168.137.221`

Reiniciamos el servicio:

`resolvconf -u && service named restart`

y Podemos comprobar que el servicio funciona:

```
root@freebsd:/usr/local/etc/namedb/master # nslookup nik.com
Server:      192.168.137.221
Address:     192.168.137.221#53

Name:   nik.com
Address: 192.168.137.221

root@freebsd:/usr/local/etc/namedb/master # nslookup 192.168.137.221
221.137.168.192.in-addr.arpa    name = nik.com.

root@freebsd:/usr/local/etc/namedb/master #
```

1.6 DHCP

Primero instalamos el paquete del servidor DHCP:

```
pkg install isc-dhcp44-server
```

Modificamos el archivo /etc/rc.conf añadiendo las siguientes líneas:

```
dhcp_enable="YES"
```

```
dhcpcd_enable="YES"
```

```
dhcp_ifaces="em1"
```

Luego modificamos el archivo /usr/local/etc/dhcpd.conf y añadimos lo siguiente:

```
host nik_dhcp {
    hardware ethernet 08:00:27:a1:00:4f;
    fixed-address 192.168.137.221;
}

subnet 192.168.137.0 netmask 255.255.255.0 {
    range 192.168.137.51 192.168.137.100;
    option domain-name-servers 8.8.8.8;
    option routers 192.168.137.10;
    option broadcast-address 192.168.137.255;
    default-lease-time 600;
    max-lease-time 7200;
}
```

Donde la IP y la MAC es la del servidor.

Luego reiniciamos el servicio con `service isc-dhcpd restart`

Y podemos comprobar desde una maquina cliente que el servicio funciona:

```
em1: flags=8863<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=481009b<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, VLAN_HWCSUM, VLAN_HWFILTER, NOMAP>
    ether 08:00:27:21:29:f5
    inet 192.168.137.3 netmask 0xfffff00 broadcast 192.168.137.255
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD, IFDISABLED, AUTO_LINKLOCAL>
```

1.7 NFS

Para crear un servidor NFS en 192.168.137.221, con un cliente 192.168.137.3, haremos lo siguiente:

Primero añadimos las siguientes líneas a /etc/rc.conf:

```
mountd_flags="-r"
rpcbind_enable="YES"
nfs_server_enable="YES"
nfs_server_flags="-u -t -n 4"
portmap_enable="YES"
```

Luego creamos un archivo /etc/exports, donde pondremos la siguiente línea:

```
/NFS -alldirs -maproot=root 192.168.137.3
```

Reiniciamos el servicio:

```
service nfsd restart
```

Desde el cliente agregamos las siguientes líneas en rc.conf:

```
nfs_client_enable="YES"
nfs_client_flags="-n 4"
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
```

Luego montaremos la carpeta compartida:

(En servidor y cliente ya debe existir una carpeta llamada NFS)

```
mount_nfs 192.168.137.221:/NFS /NFS
```

Y Podemos comprobar que se sincronizan los datos de dicha carpeta:

(Desde cliente):

```
root@np31:/ # cat /NFS-Server/prueba1.txt
Hola (Server)
root@np31:/ # echo "Hola (Cliente)" >> /NFS-Server/prueba1.txt
root@np31:/ # cat /NFS-Server/prueba1.txt
Hola (Server)
Hola (Cliente)
root@np31:/ #
```

(Desde servidor):

```
root@freebsd:/ # cat /NFS/prueba1.txt
Hola (Server)
root@freebsd:/ # cat /NFS/prueba1.txt
Hola (Server)
Hola (Cliente)
root@freebsd:/ #
```

1.8 SAMBA

En el servidor (192.168.137.221):

Primero instalaremos el paquete:

```
pkg install samba413
```

Luego deberemos crear un archivo de configuración en /usr/local/etc/smb4.conf:

```
GNU nano 5.8 /usr/local/etc/smb4.conf
[global]
workgroup = WORKGROUP
netbios name = user
hosts allow = 192.168.137. 127.
printcap name = cups
load printers = yes
cups options = raw

[private]
path = /usr/home/nik/Desktop/SAMBA
valid users = nik
writable = yes
public = no
```

En /etc/rc.conf agregamos la siguiente línea:

```
samba_server_enable="YES"
```

luego agregamos al usuario al grupo

```
pw groupadd smbprivate -M nik
```

creamos la contraseña con

```
smbpasswd -a nik
```

y reiniciamos el servicio

```
service samba_server restart
```

Desde el cliente deberemos instalar smbclient, y luego podremos comprobar que se sincronizan los archivos del servicio:

```
nikita-virtualbox nikita]# smbclient //192.168.137.221/private -U nik
Can't load /etc/samba/smb.conf - run testparm to debug it
Password for [WORKGROUP\nik]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0  Fri Nov 12 01:51:30 2021
..               D          0  Fri Nov 12 01:51:30 2021
16349820 blocks of size 1024. 16349408 blocks available
smb: \> touch sambatest.txt
touch: command not found
smb: \> put test.txt
test.txt does not exist
smb: \> ! echo "hola (cliente)" > text.txt
smb: \> put text.txt
putting file text.txt as \text.txt (0,3 kb/s) (average 0,3 kb/s)
smb: \> ls
.                D          0  Fri Nov 12 02:12:29 2021
..               D          0  Fri Nov 12 01:51:30 2021
text.txt         A         15  Fri Nov 12 02:12:29 2021
16349792 blocks of size 1024. 16349380 blocks available
smb: \> cat text.txt
cat: command not found
smb: \> nano text.txt
nano: command not found
smb: \> ls
.                D          0  Fri Nov 12 02:12:29 2021
..               D          0  Fri Nov 12 02:13:22 2021
text.txt         A         14  Fri Nov 12 02:13:50 2021
16349792 blocks of size 1024. 16349380 blocks available
smb: \> exit
```

```
$ cat SAMBA/text.txt
hola (cliente)
$ echo "hola (server)" > text.txt
$ clear

$ cat SAMBA/text.txt
hola (cliente)
$ echo "hola (server)" > SAMBA/text.txt
```

2. Rocky Linux

2.1 SSH

Primero en el servidor np31@192.168.137.222 añadimos las siguientes líneas en /etc/ssh/sshd_config:

PermitRootLogin no

PermitEmptyPasswords no

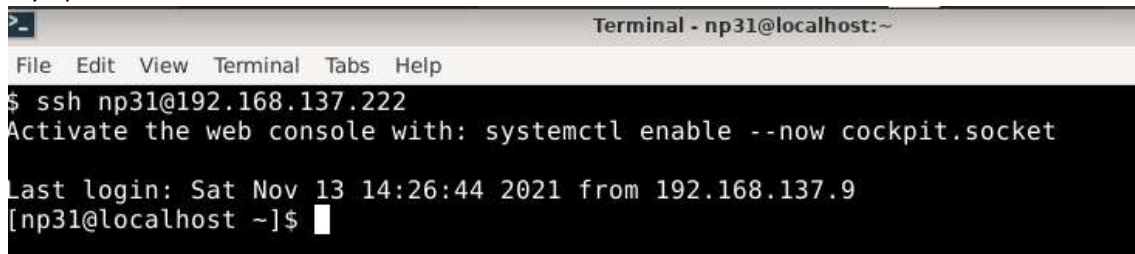
y agregamos AllowUsers np31

Desde el cliente generamos una clave y se la pasamos al servidor:

ssh-keygen -t rsa

ssh-copy-id -i .ssh/id_rsa.pub np31@192.168.137.222

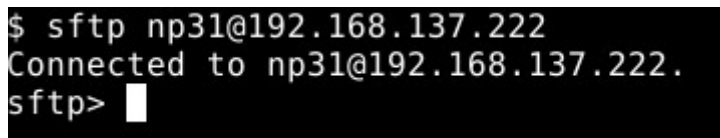
Y ya podremos conectarnos sin contraseña:



```
Terminal - np31@localhost:~
File Edit View Terminal Tabs Help
$ ssh np31@192.168.137.222
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Sat Nov 13 14:26:44 2021 from 192.168.137.9
[np31@localhost ~]$
```

2.2 SFTP

el servicio sftp funcionara una vez configurado ssh:



```
$ sftp np31@192.168.137.222
Connected to np31@192.168.137.222.
sftp>
```

2.3 SCP

el servicio scp funcionara una vez configurado ssh:



```
$ scp prueba.txt np31@192.168.137.222:~
prueba.txt 100% 0 0.0KB/s 00:00
$
```


2.4 VNC

primero instalaremos el paquete tigervnc-server

`yum install tigervnc-server`

Modificamos el archivo `/etc/gdm/custom.conf`

descomentamos la línea

`WaylandEnable=false`

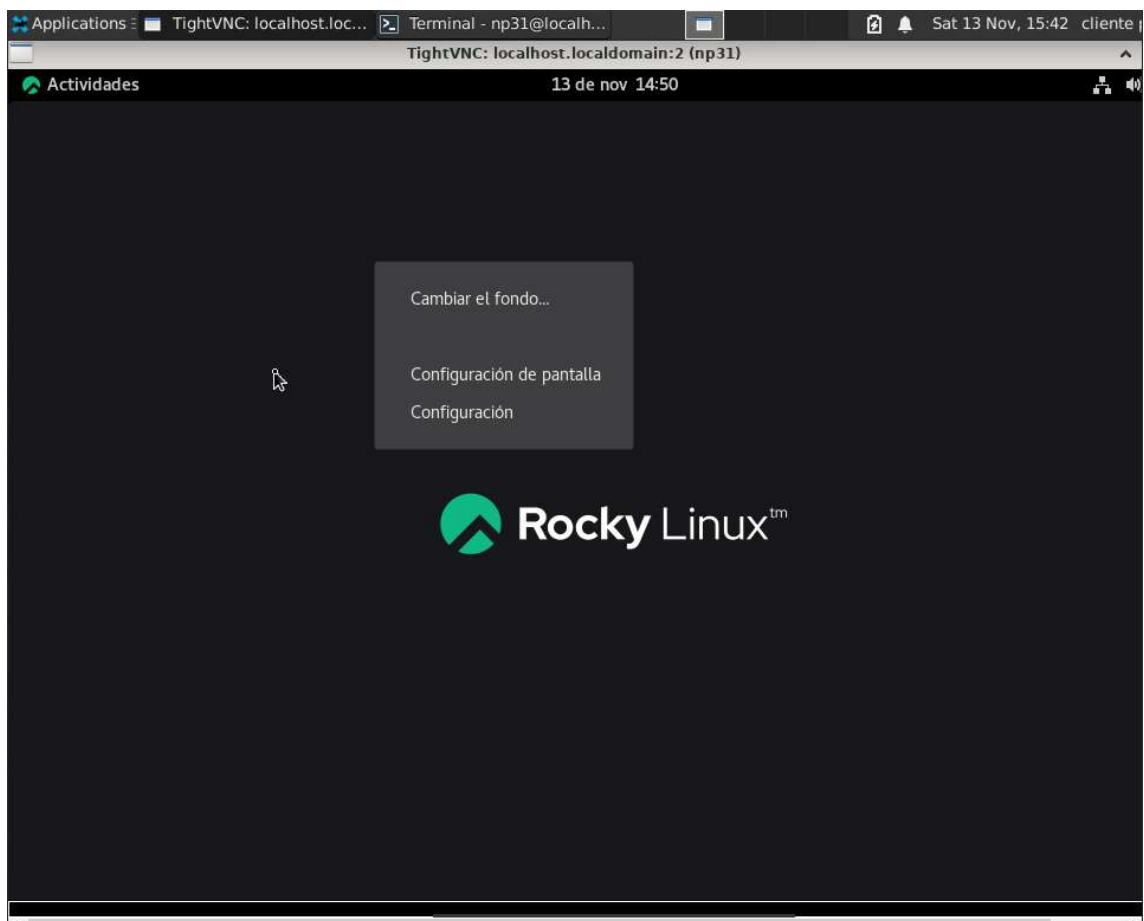
Reiniciamos la maquina

Iniciamos el servidor

`vncserver`

Y desde el cliente nos podemos conectar con un cliente vnc (tightvnc):

`vncviewer 192.168.137.222:1`



2.5 RDP

En el servidor instalamos xrdp

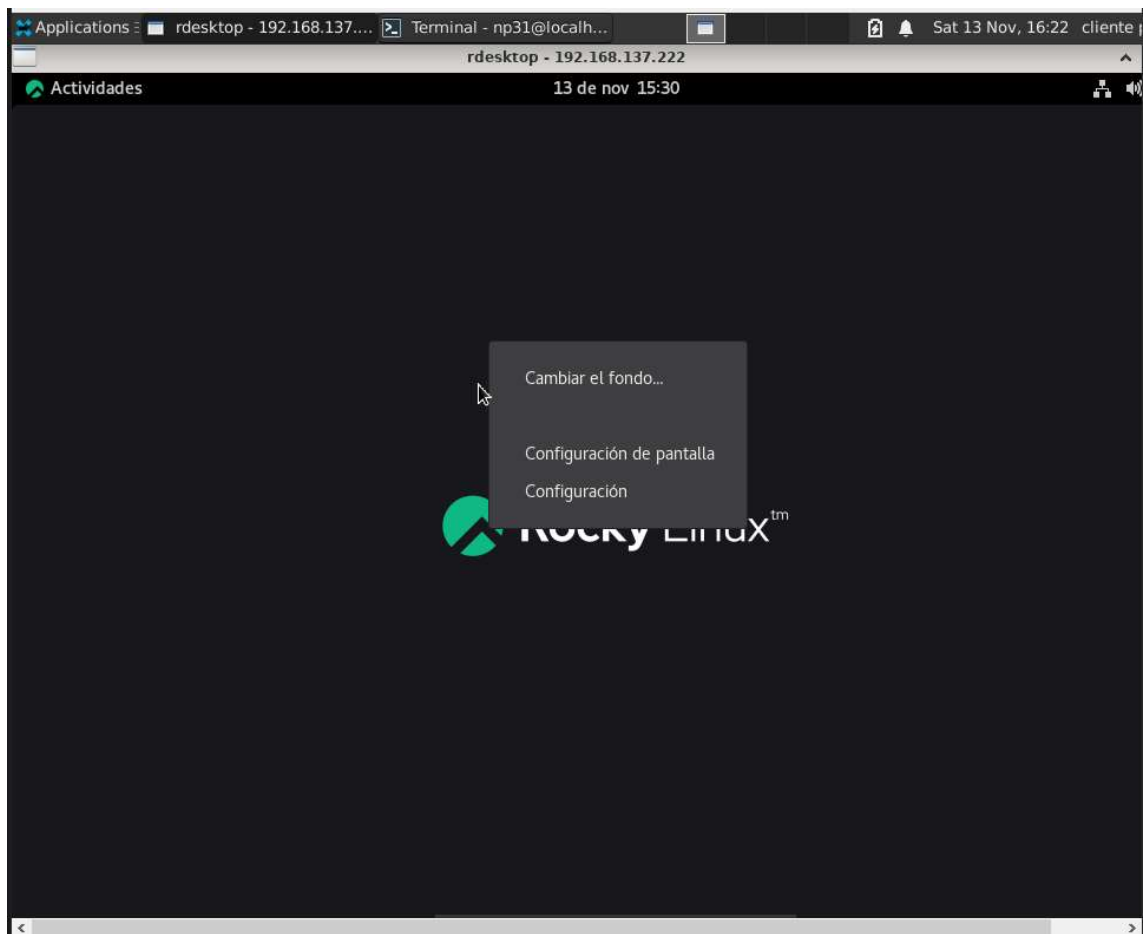
`yum install xrdp`

iniciamos el servicio

`systemctl start xrdp`

Desde el cliente instalamos una aplicación cliente rdp (rdesktop) y la utilizamos para conectarnos

`rdesktop 192.168.137.222`



2.6 DNS

Primero instalamos bind:

```
dnf -y install bind bind-utils
```

luego modificamos el archive /etc/named.conf, para que se vea así:

```
//  
// named.conf  
//  
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS  
// server as a caching only nameserver (as a localhost DNS resolver only).  
//  
// See /usr/share/doc/bind*/sample/ for example named configuration files.  
//  
acl internal-network{  
    192.168.137.0/24;  
};  
  
options {  
    listen-on port 53 { any; };  
    listen-on-v6 { any; };  
    directory "/var/named";  
    dump-file "/var/named/data/cache_dump.db";  
    statistics-file "/var/named/data/named_stats.txt";  
    memstatistics-file "/var/named/data/named_mem_stats.txt";  
    secroots-file "/var/named/data/named.secroots";  
    recursing-file "/var/named/data/named.recursing";  
    allow-query { localhost; internal-network; };  
};
```

Luego escribimos nuestras zonas en el mismo fichero:

```
GNU nano 2.9.8 /etc/named.conf  
  
include "/etc/named.rfc1912.zones";  
include "/etc/named.root.key";  
  
zone "nikrocky.com" IN {  
    type master;  
    file "nikrocky.com.lan";  
    allow-update {none;};  
    allow-query {any;};  
};  
  
zone "137.168.192.in-addr.arpa" IN {  
    type master;  
    file "137.168.192.db";  
    allow-update {none;};  
    allow-query {any;};  
};
```

Luego crearemos los archivos zona en /var/named/:

```
GNU nano 2.9.8 /var/named/nikrocky.com.lan

$TTL 86400
@      IN SOA  dlp.nikrocky.com. root.nikrocky.com. (
        2021111301      ;Serial
        3600      ;Refresh
        1800      ;Retry
        604800    ;Expire
        86400     ;Minimum TTL
)
      IN NS   dlp.nikrocky.com.
      IN A    192.168.137.222
      IN MX 10 dlp.nikrocky.com.
dlp    IN A    192.168.137.222
```

```
GNU nano 2.9.8 /var/named/137.168.192.db

$TTL 86400
@      IN SOA  dlp.nikrocky.com. root.nikrocky.com. (
        2021111301      ;Serial
        3600      ;Refresh
        1800      ;Retry
        604800    ;Expire
        86400     ;Minimum TTL
)
      IN NS   dlp.nikrocky.com.
222    IN PTR  dlp.nikrocky.com.
```

Una vez hecho todo esto, iniciamos el servicio

```
systemctl enable --now named
```

Luego modificamos nuestra conexion para que rediriga a nuestro DNS:

```
nmcli connection modify enp0s3 ipv4.dns 192.168.137.222
```

```
nmcli connection down enp0s3
```

```
nmcli connection up enp0s3
```

```
reboot
```

Para comprobar que funciona haremos `dig -x 192.168.137.222`, y `nslookup dlp.nikrocky.com.` :

```
np31@localhost:/
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

<<>> DiG 9.11.26-RedHat-9.11.26-4.el8_4 <<>> -x 192.168.137.222
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60256
; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 222b8dc703297719796bca40618fd3f766e2e89bb56e4386 (good)
; QUESTION SECTION:
222.137.168.192.in-addr.arpa.  IN      PTR

; ANSWER SECTION:
222.137.168.192.in-addr.arpa. 86400 IN    PTR    dlp.nikrocky.com.

; AUTHORITY SECTION:
137.168.192.in-addr.arpa. 86400 IN    NS      dlp.nikrocky.com.

; ADDITIONAL SECTION:
dlp.nikrocky.com.            86400  IN      A       192.168.137.222

; Query time: 0 msec
; SERVER: 192.168.137.222#53(192.168.137.222)
; WHEN: sáb nov 13 16:04:23 CET 2021
; MSG SIZE rcvd: 145

root@localhost /]#
```

```
[root@localhost /]# nslookup dlp.nikrocky.com
Server:          192.168.137.222
Address:         192.168.137.222#53

Name:   dlp.nikrocky.com
Address: 192.168.137.222

[root@localhost /]#
```

2.7 DHCP

Primero instalamos el paquete

`dnf -y install dhcp-server`

Luego modificamos el archivo `/etc/dhcp/dhcpd.conf` para que se vea así:

```
GNU nano 2.9.8 /etc/dhcp/dhcpd.conf

# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp-server/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#
option domain-name "nikrocky.com";

option domain-name-servers dlp.nikrocky.com;

default-lease-time 600;
max-lease-time 7200;

authoritative;

subnet 192.168.137.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.137.3 192.168.137.50;
    option broadcast-address 192.167.137.255;
    option routers 192.168.137.100;
}
```

Luego activamos el servicio

`systemctl enable --now dhcpd`

Si nos conectamos desde otra máquina a la red podremos ver desde `/var/lib/dhcpd/dhcpd.leases` que se le ha prestado la dirección IP a la máquina cliente:

```
[root@localhost /]# cat /var/lib/dhcpd/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.3.6

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

server-uid "\000\001\000\001)\\"223b\010\000'\234i!";

lease 192.168.137.3 {
    starts 6 2021/11/13 15:18:22;
    ends 6 2021/11/13 15:28:22;
    cltt 6 2021/11/13 15:18:22;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 08:00:27:65:76:ee;
    uid "\001\010\000'ev\356";
    client-hostname "FBSDaRocky";
}
[root@localhost /]#
```

```
root@FBSDaRocky:/ # ifconfig
em0: flags=8863<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=481009b<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, VLAN_HWCSUM, VLAN_HWFILTER, NOMAP>
    ether 08:00:27:65:76:ee
    inet 192.168.137.3 netmask 0xfffff00 broadcast 192.167.137.255
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
em1: flags=8863<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=481009b<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, VLAN_HWCSUM, VLAN_HWFILTER, NOMAP>
    ether 08:00:27:e1:e4:38
    inet 10.0.2.24 netmask 0xfffff00 broadcast 10.0.2.255
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
    options=680003<RXCSUM, TXCSUM, LINKSTATE, RXCSUM_IPV6, TXCSUM_IPV6>
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
    inet 127.0.0.1 netmask 0xff000000
    groups: lo
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@FBSDaRocky:/ #
```

2.8 NFS

Primero instalamos el paquete:

```
dnf -y install nfs-utils
```

modificamos el archivo /etc/idmapd.conf y ponemos la siguiente línea:

```
Domain = nikrocky.com
```

Añadimos la siguiente línea en /etc/exports

```
/home/np31/Escritorio/NFS 192.168.137.4(rw,no_root_squash,sync)
```

(La carpeta /Escritorio/NFS debe existir)

Reiniciamos los servicios:

```
systemctl enable rpcbind
```

```
systemctl enable nfs-server
```

```
systemctl restart rpcbind
```

```
systemctl restart nfs-server
```

Desde el cliente añadimos las siguientes líneas en /etc/rc.conf:

```
#NFS
nfs_client_enable="YES"
mountd_flags="-r"
rpcbind_enable="YES"
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
portmap_enable="YES"
```

Iniciamos el servicio:

```
service lockd start
```

Luego montamos la unidad:

```
mount -t nfs 192.168.137.222:/home/np31/Escritorio/NFS /home/cliente/NFS
```

Y comprobamos su funcionamiento:

```
$ cd NFS/
$ ls
$ touch texto.txt
$ ls
texto.txt
$ echo "Hola (Cliente)" >> texto.txt
$ cat texto.txt
Hola (Cliente)
$
```

```
[np31@localhost Escritorio]$ ls
NFS
[np31@localhost Escritorio]$ cd NFS/
[np31@localhost NFS]$ ls
texto.txt
[np31@localhost NFS]$ cat texto.txt
Hola (Cliente)
```


2.9 SAMBA

Primero instalamos el paquete:

```
dnf -y install samba
```

Creamos el directorio a compartir y le damos permisos

```
mkdir /home/np31/Escritorio/Samba
```

```
chmod 777 /home/np31/Escritorio/Samba
```

```
chown np31:np31 /home/np31/Escritorio/Samba
```

En el archivo `/etc/samba/smb.conf` añadimos lo siguiente

```
hosts allow = 192.168.137. 127.
```

```
[Share]
```

```
path=/home/np31/Escritorio/Samba
```

```
valid users = np31
```

```
writable= yes
```

```
read only=no
```

Luego establecemos una contraseña:

```
smbpasswd -a np31
```

Iniciamos los servicios

```
systemctl enable smb nmb && systemctl restart smb nmb
```

Desde el cliente nos conectamos:

```
smbclient //192.168.137.222/Share -U np31
```

Y comprobamos que funciona:

```
root@FBSDaRocky:/home/cliente # smbclient //192.168.137.222/Share -U np31
smbclient: Can't load /usr/local/etc/smb4.conf - run testparm to debug it
Enter WORKGROUP\np31's password:
Try "help" to get a list of possible commands.
smb: \> ! echo "hola (cliente)" >> prueba.txt
smb: \> put prueba.txt
putting file prueba.txt as \prueba.txt (4.9 kb/s) (average 4.9 kb/s)
```

```
np31@localhost:~/Escritorio/Samba
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[np31@localhost Samba]$ ls
[np31@localhost Samba]$ ls
prueba.txt
[np31@localhost Samba]$ cat prueba.txt
hola (cliente)
[np31@localhost Samba]$ echo "Hola (servidor)" >> prueba.txt
[np31@localhost Samba]$ cat prueba.txt
hola (cliente)
Hola (servidor)
[np31@localhost Samba]$
```

3. Windows Server 2022

3.1 SSH + SFTP + SCP

En la maquina servidor instalamos OpenSSH Server y lo iniciamos

```
start-service sshd
```

Luego iremos a la carpeta c:\ProgramData\ssh y modificamos el archivo sshd_config:

Port 22

SyslogFacility AUTH

LogLevel DEBUG

StrictModes no

PubkeyAuthentication yes

PasswordAuthentication yes

Y comentamos las siguientes líneas:

```
#Match Group administrators
    #AuthorizedKeysFile __PROGRAMDATA__/ssh/administrators_authorized_keys
```

Después de modificar el archivo reiniciamos el servicio

```
restart-service sshd
```

Desde el cliente generamos la llave

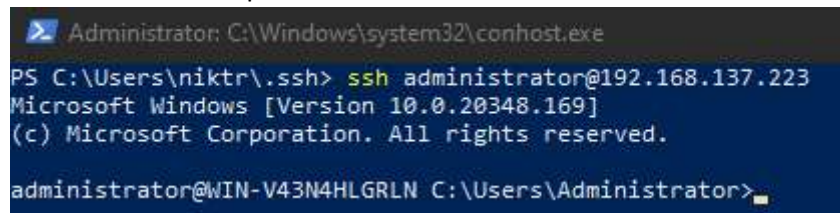
```
ssh-keygen
```

```
ssh-add c:\users\niktr\.ssh\id_rsa
```

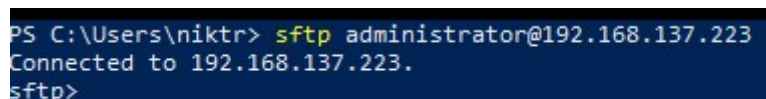
Y le pasamos la clave publica al servidor

```
scp c:\users\niktr\.ssh\id_rsa.pub
administrator@192.168.137.222:c:\users\administrator\.ssh\authorized_keys
```

una vez hecho esto podremos acceder al servidor sin utilizar la contraseña:



```
Administrator: C:\Windows\system32\conhost.exe
PS C:\Users\niktr\.ssh> ssh administrator@192.168.137.223
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.
administrator@WIN-V43N4HLGRLN C:\Users\Administrator>
```



```
PS C:\Users\niktr> sftp administrator@192.168.137.223
Connected to 192.168.137.223.
sftp>
```

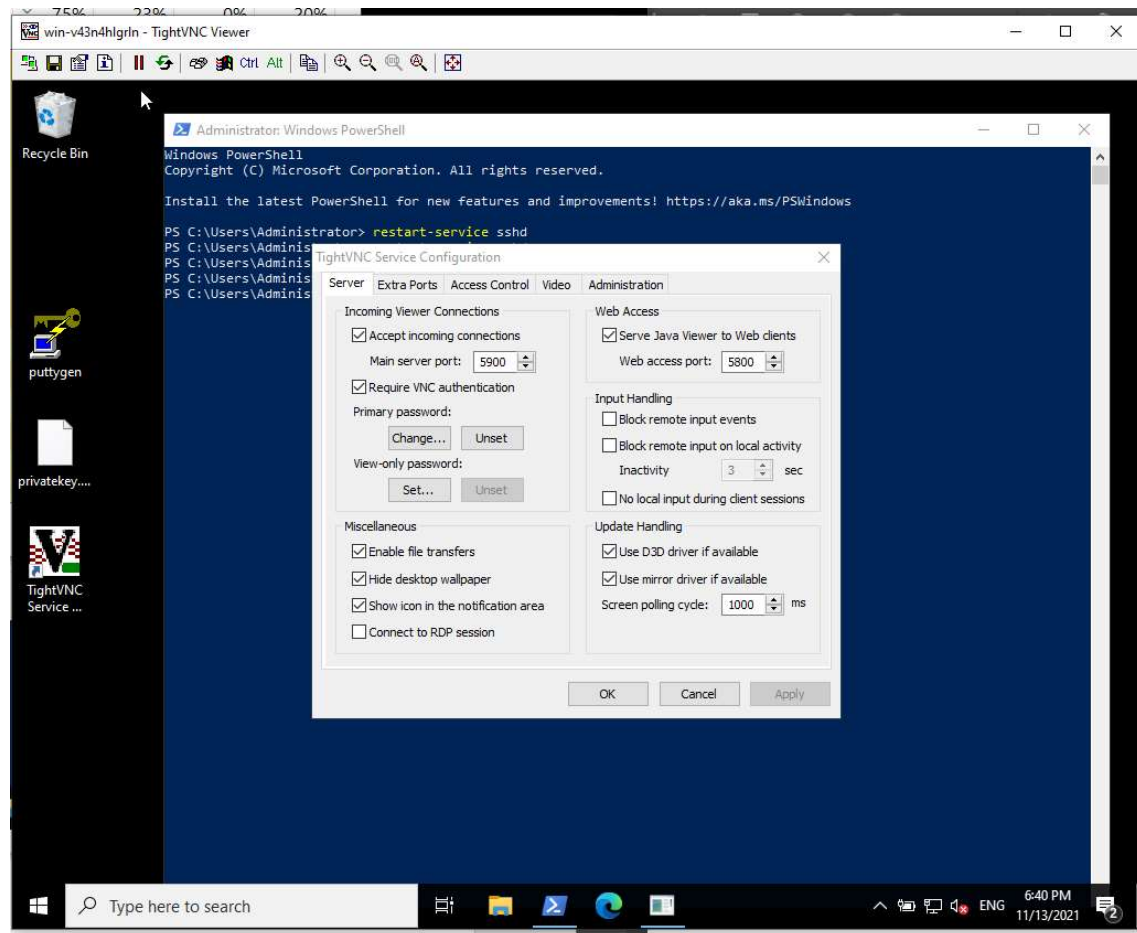
3.2 VNC

Instalamos el programa TightVNC de internet.

Una vez instalado, accedemos a TightVNC Service Configuration, y colocamos una contraseña.

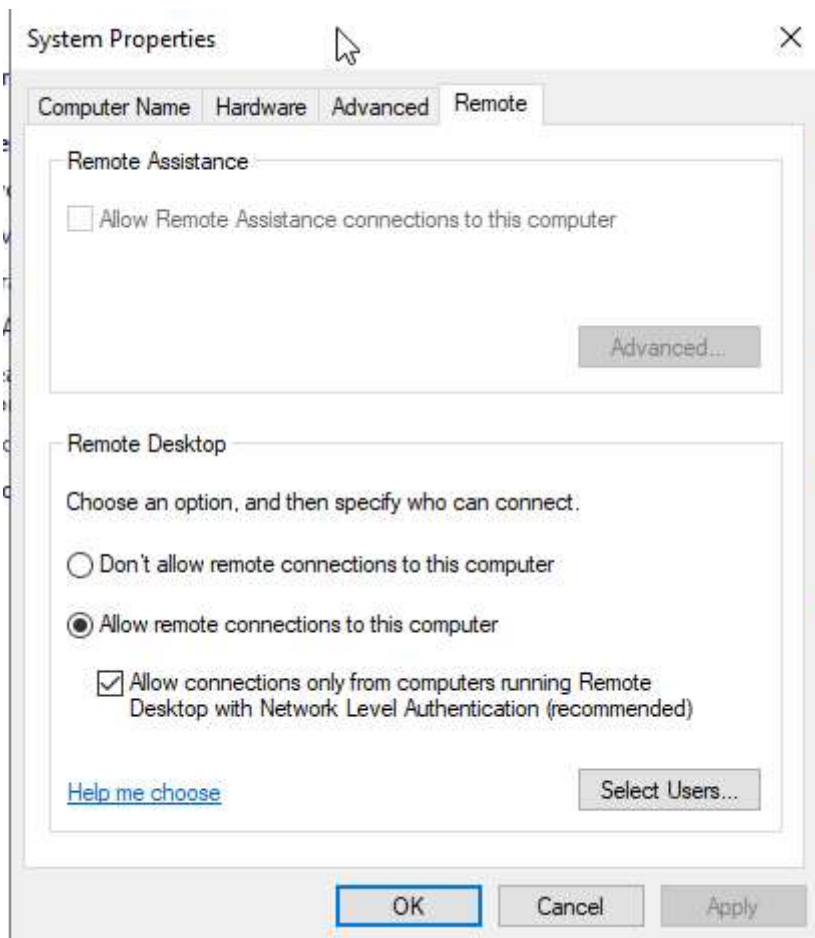
Desde la maquina cliente abrimos TightVNC Viewer, e introducimos la IP y puerto para acceder

192.168.137.223::5900

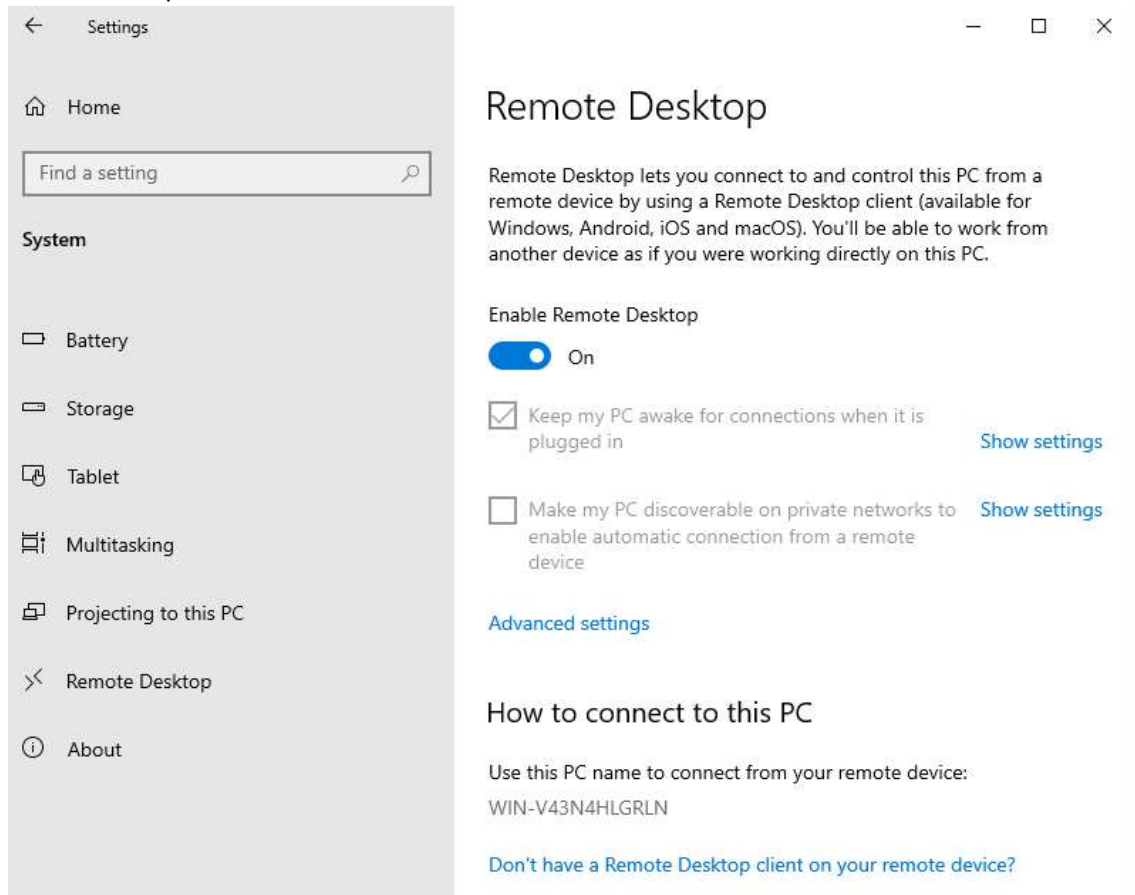


3.3 RDP

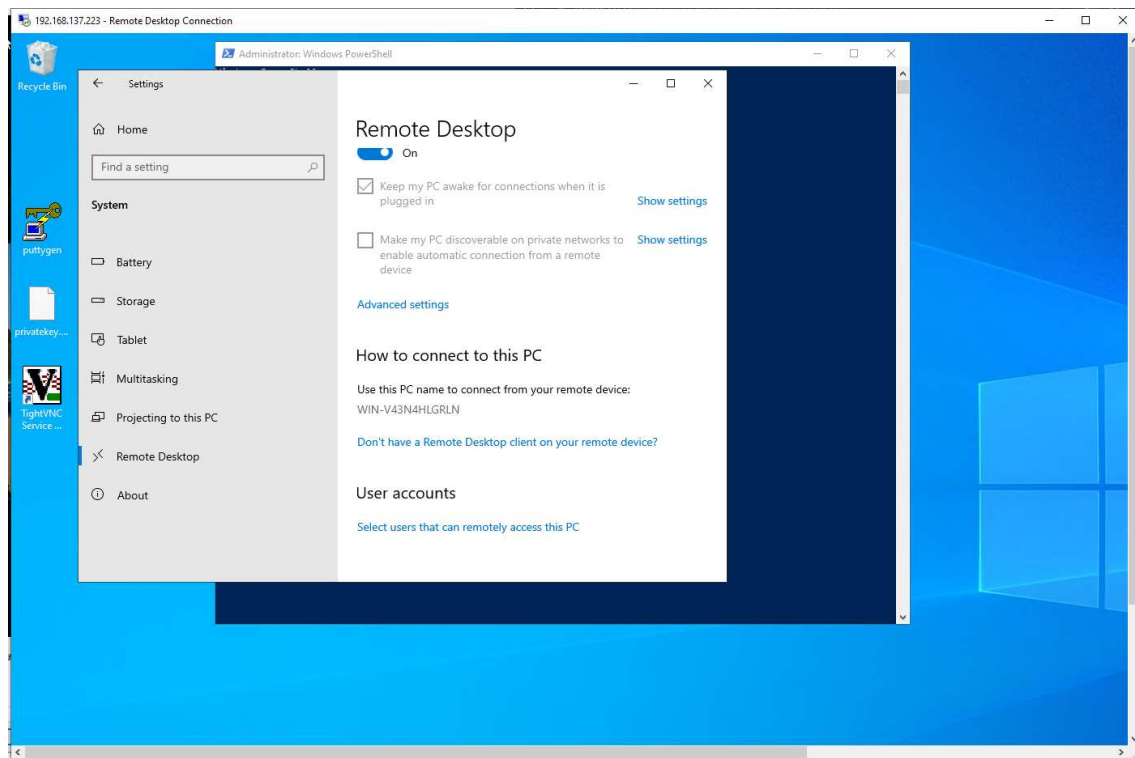
Accedemos a panel de control – sistema y seguridad – sistema – configuración avanzada del sistema – acceso remoto – permitir conexiones remotas:



Verificamos que está habilitado:

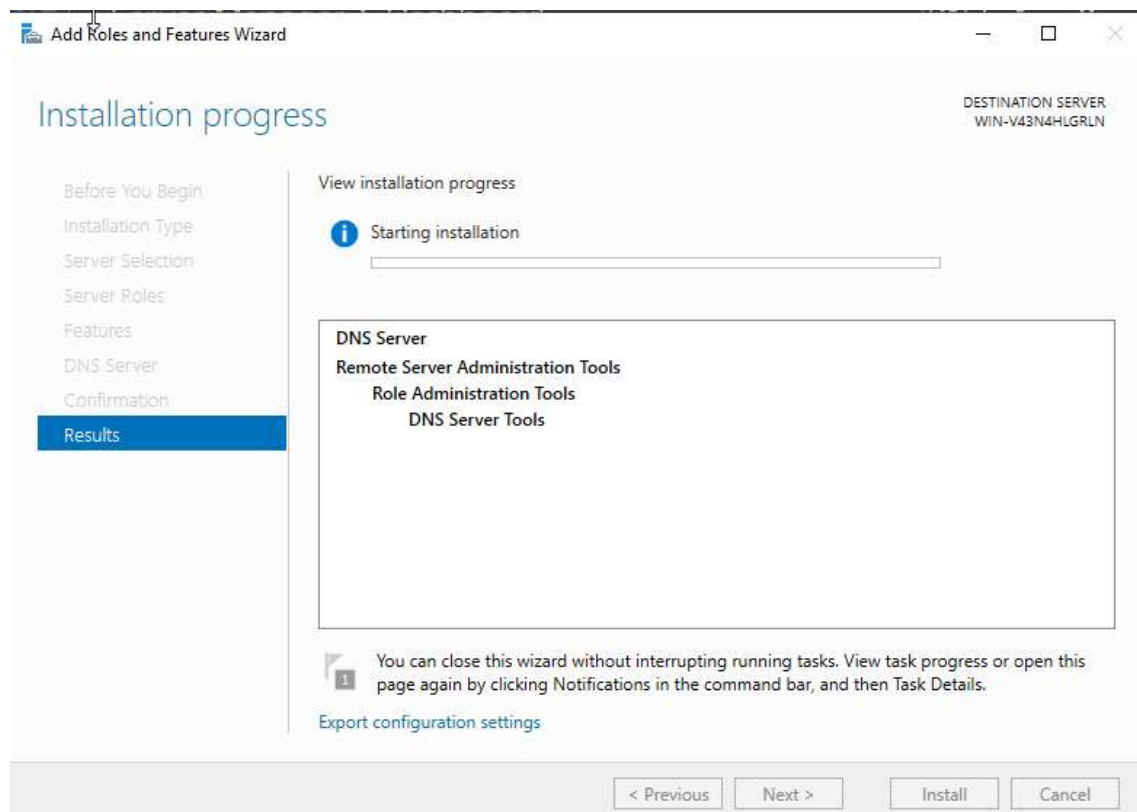
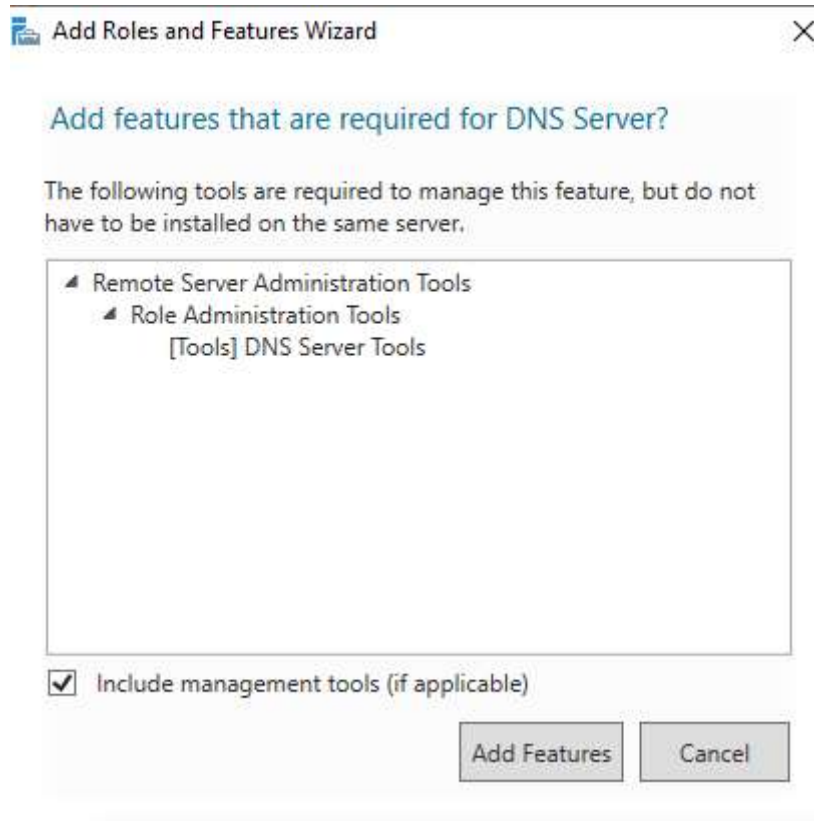


Desde el cliente nos conectamos con la ip 192.168.137.223, y el usuario y contraseña que utilizemos:



3.4 DNS

Accedemos al administrador del servidor – agregar roles y características – instalación basada en características y roles – seleccionamos un servidor del grupo de servidores – seleccionamos DNS, y seguimos con la instalación:

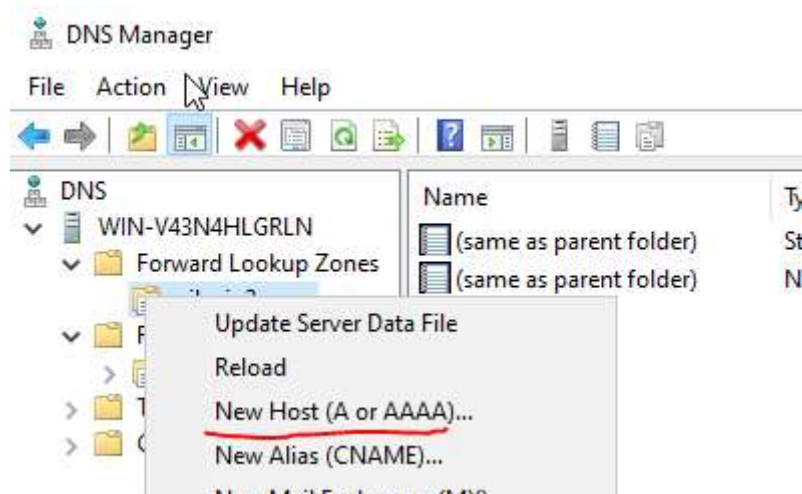


Una vez instalado iremos a la pestaña de DNS, y en nuestro servidor DNS seleccionamos configurar DNS:

Crear una zona de búsqueda directa - este servidor mantiene la zona – windnik como nombre de la zona – windnik.dns como nombre de archivo – no admitir actualizaciones dinámicas – agregamos los forwarders 193.145.233.5 193.145.233.6 8.8.8.8 8.8.4.4

Creamos una zona de búsqueda inversa. Le damos todo a siguiente, cuando nos pida la IP, ponemos 192.168.137

Desde la zona de búsqueda directa crearemos un nuevo host, con nombre nik, y con nuestra dirección IP 192.168.137.223, y creamos un registro del puntero PTR



Podemos comprobar con nslookup que funciona:

```
C:\Windows\system32\cmd.exe - C:\Windows\system32\nslookup.exe - fe80::5469:a64:16ba:b4d1
Default Server:  UnKnown
Address:  fe80::5469:a64:16ba:b4d1

> nik.windnik
Server:  UnKnown
Address:  fe80::5469:a64:16ba:b4d1

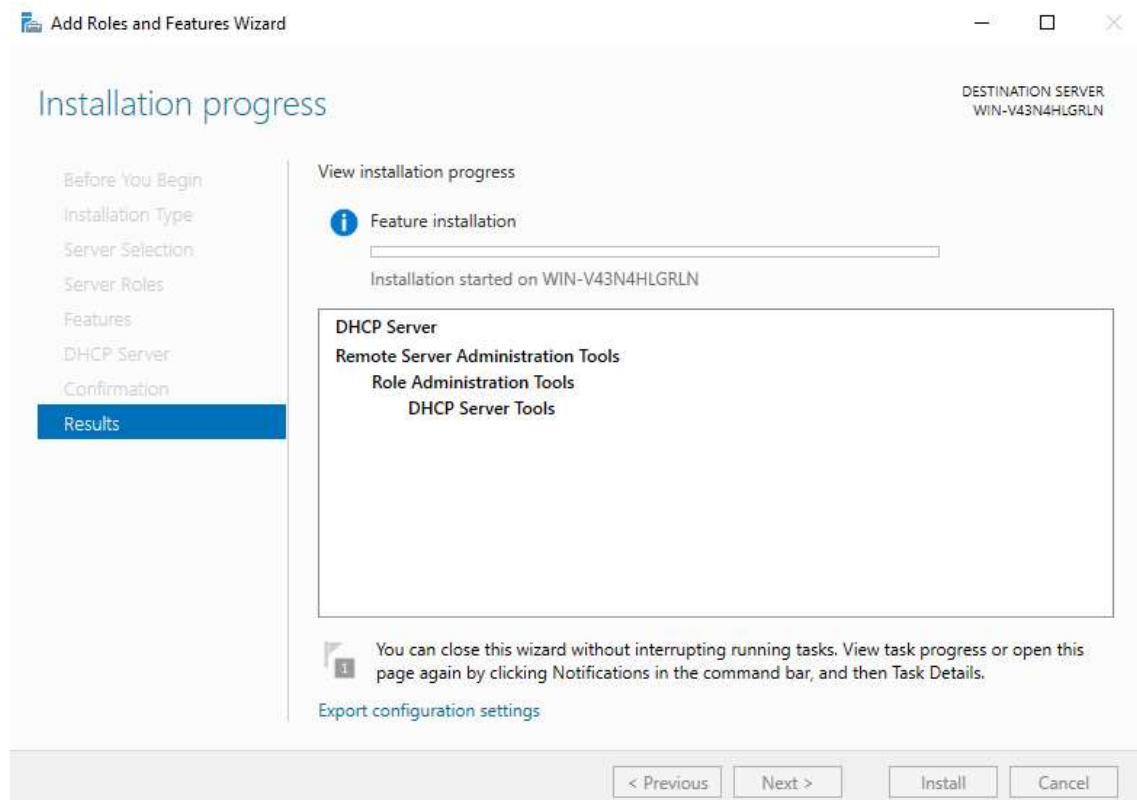
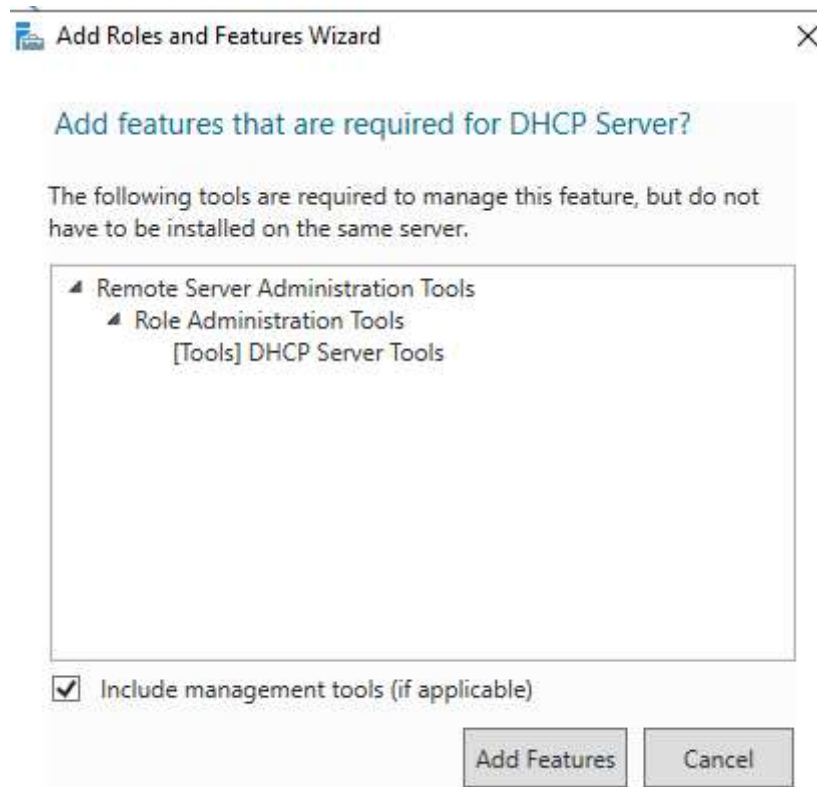
Name:     nik.windnik
Address:  192.168.137.223

> 192.168.137.223
Server:  UnKnown
Address:  fe80::5469:a64:16ba:b4d1

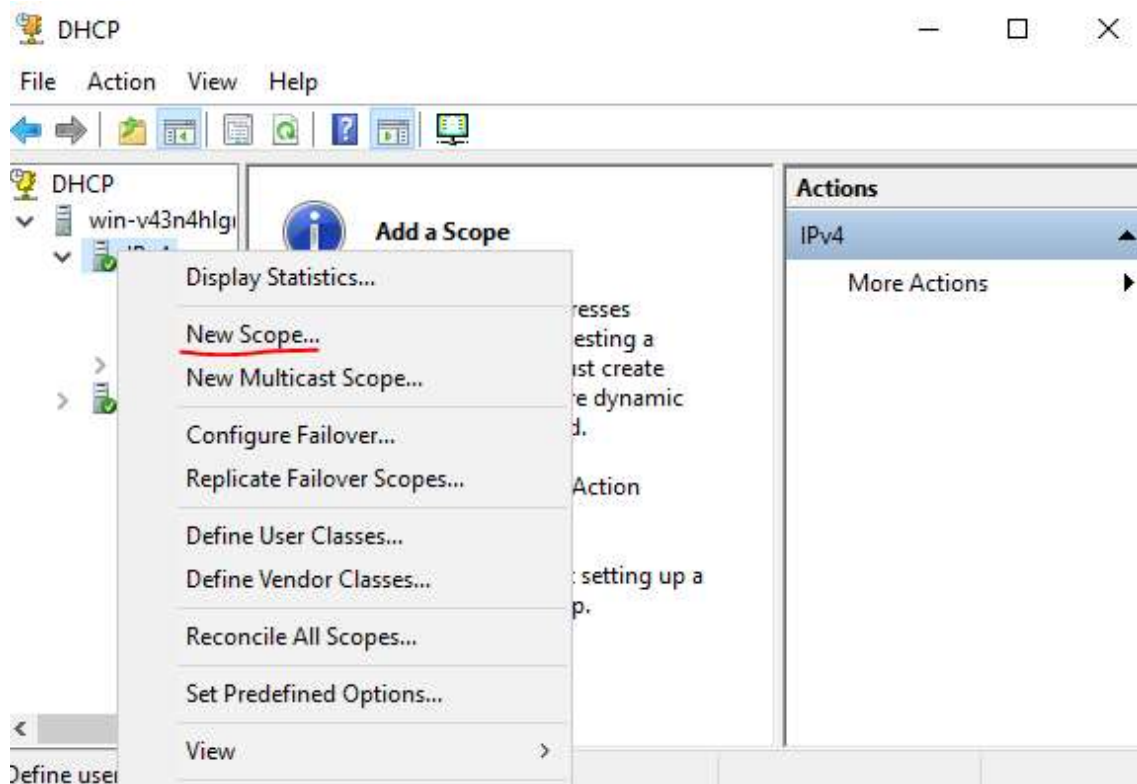
Name:     nik.windnik
Address:  192.168.137.223
```

3.5 DHCP

Se instala igual que DNS



Luego en la ventana de DHCP creamos un nuevo ámbito:



New Scope Wizard

IP Address Range

You define the scope address range by identifying a set of consecutive IP addresses.

Configuration settings for DHCP Server

Enter the range of addresses that the scope distributes.

Start IP address:

End IP address:

Configuration settings that propagate to DHCP Client

Length:

Subnet mask:

< Back Next > Cancel

New Scope Wizard

Router (Default Gateway)

You can specify the routers, or default gateways, to be distributed by this scope.



To add an IP address for a router used by clients, enter the address below.

IP address:

Add

192.168.137.100

Remove

Up

Down

< Back

Next >

Cancel

New Scope Wizard

Domain Name and DNS Servers

The Domain Name System (DNS) maps and translates domain names used by clients on your network.



You can specify the parent domain you want the client computers on your network to use for DNS name resolution.

Parent domain: 192.168.137.223

To configure scope clients to use DNS servers on your network, enter the IP addresses for those servers.

Server name:

IP address:

Add

Resolve

192.168.0.1
192.168.137.223
8.8.8.8

Remove

Up

Down

< Back

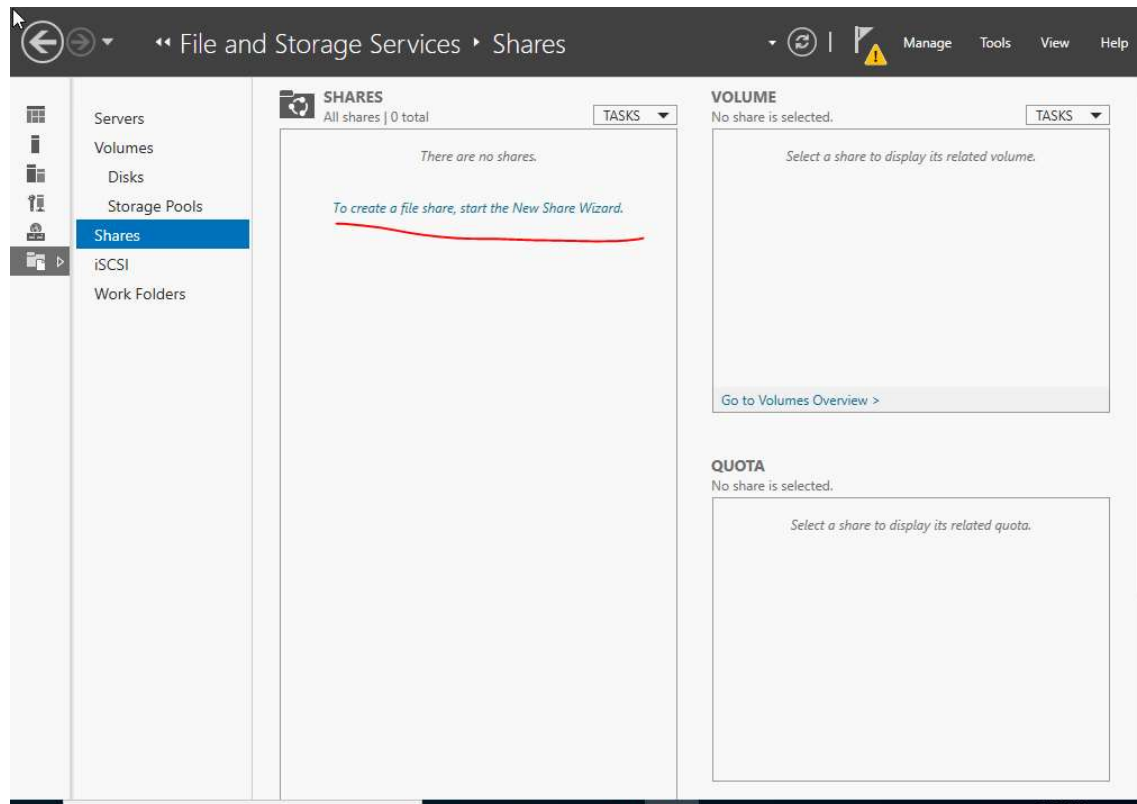
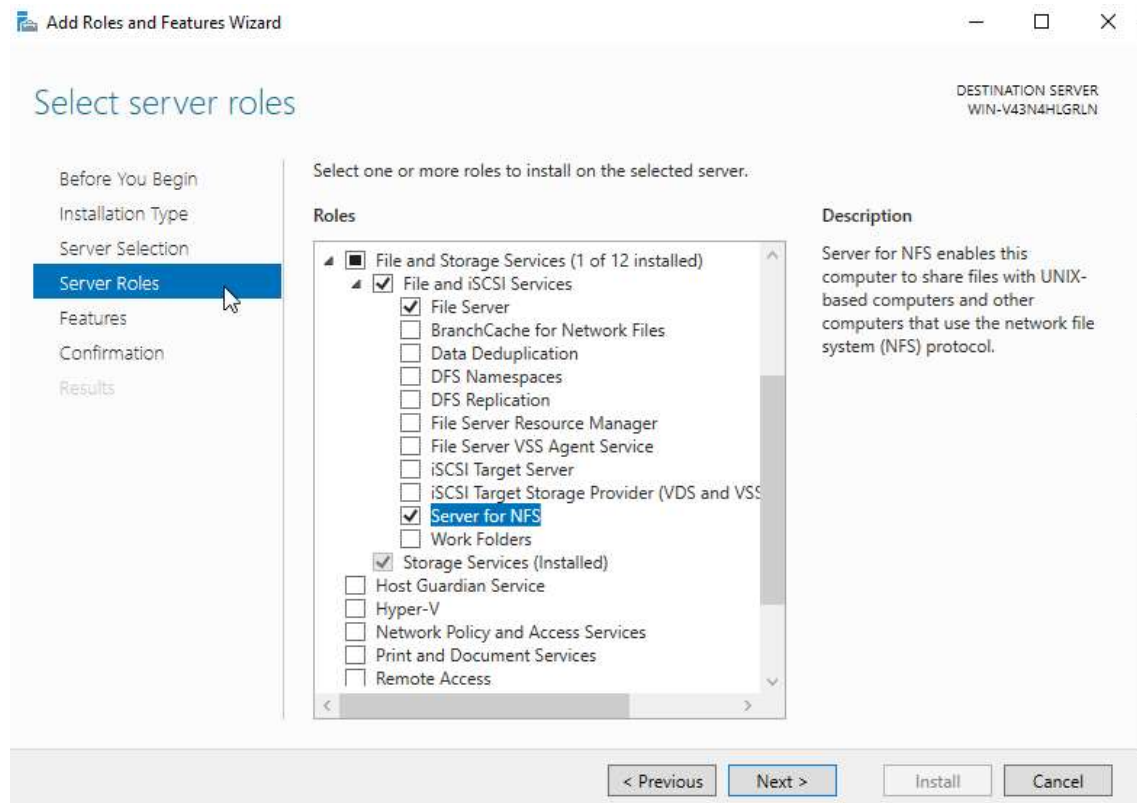
Next >

Cancel

Desde otra maquina comprobamos que funciona:

```
Starting dhclient.  
DHCPREQUEST on em0 to 255.255.255.255 port 67  
DHCPACK from 192.168.137.223  
bound to 192.168.137.10 -- renewal in 345600 seconds.  
root@FBSDaWin:/ #
```

3.6 NFS



New Share Wizard

Specify authentication methods

Select Profile
Share Location
Share Name
Authentication
Share Permissions
Permissions
Confirmation
Results

Specify the authentication methods that you want to use for this NFS share.

Kerberos v5 authentication

- ☐ Kerberos v5 authentication(Krb5)
- ☐ Kerberos v5 authentication and integrity(Krb5i)
- ☐ Kerberos v5 authentication and privacy(Krb5p)

No server authentication

- ☒ No server authentication (AUTH_SYS)
 - ☒ Enable unmapped user access
 - ☒ Allow unmapped user access by UID/GID
 - ☐ Allow anonymous access

< Previous Next > Create Cancel

Add Permissions

Grant permissions to access the NFS share to a host, client group, or netgroup.
Select the access and language encoding for the share.

☐ Host:

☐ Netgroup:

☐ Client group:

☒ All Machines

Language encoding: Share permissions:

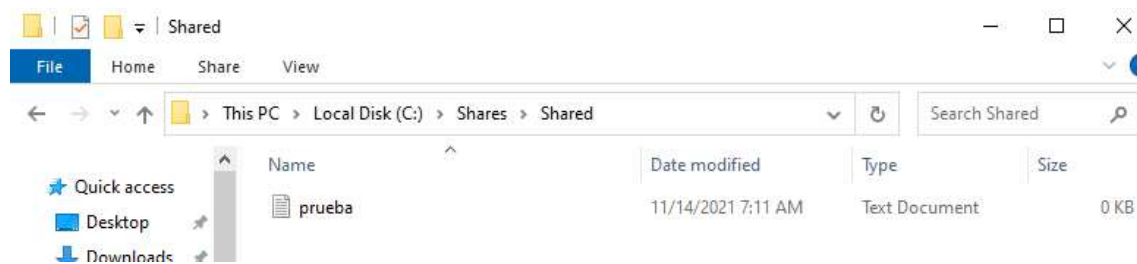
☐ Allow root access (not recommended)

Add Cancel

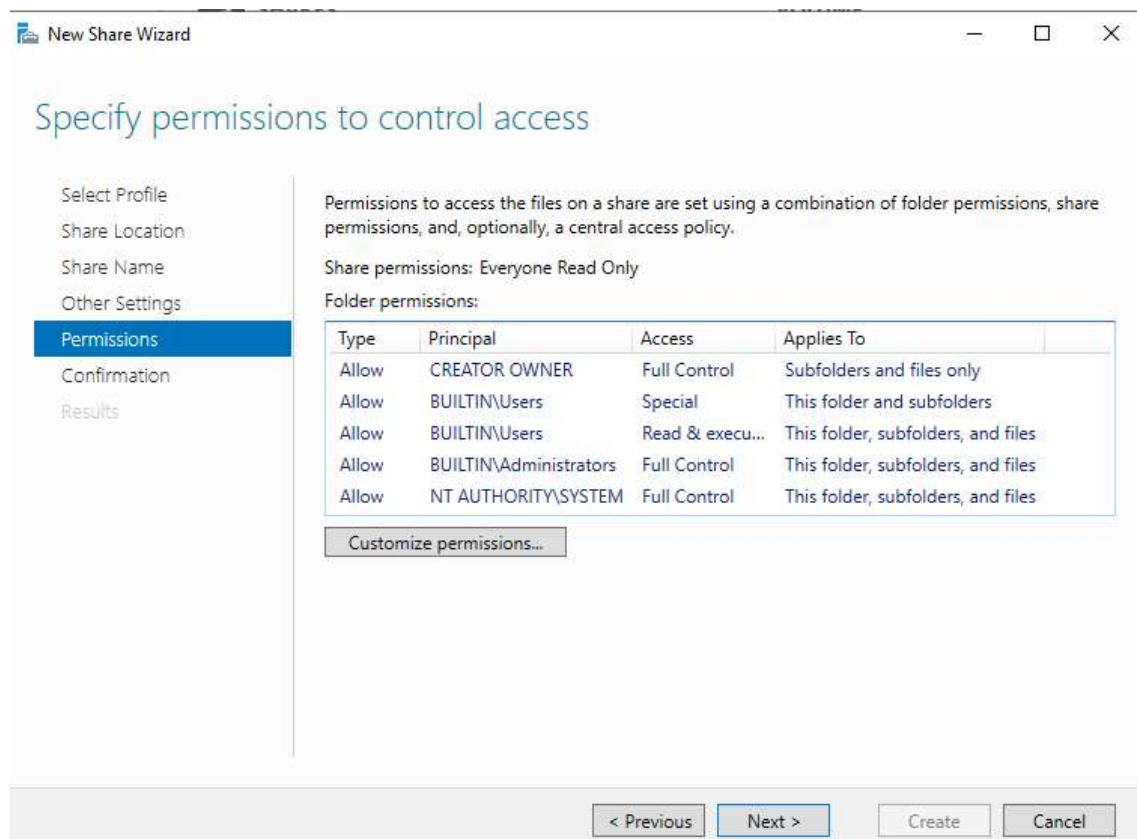
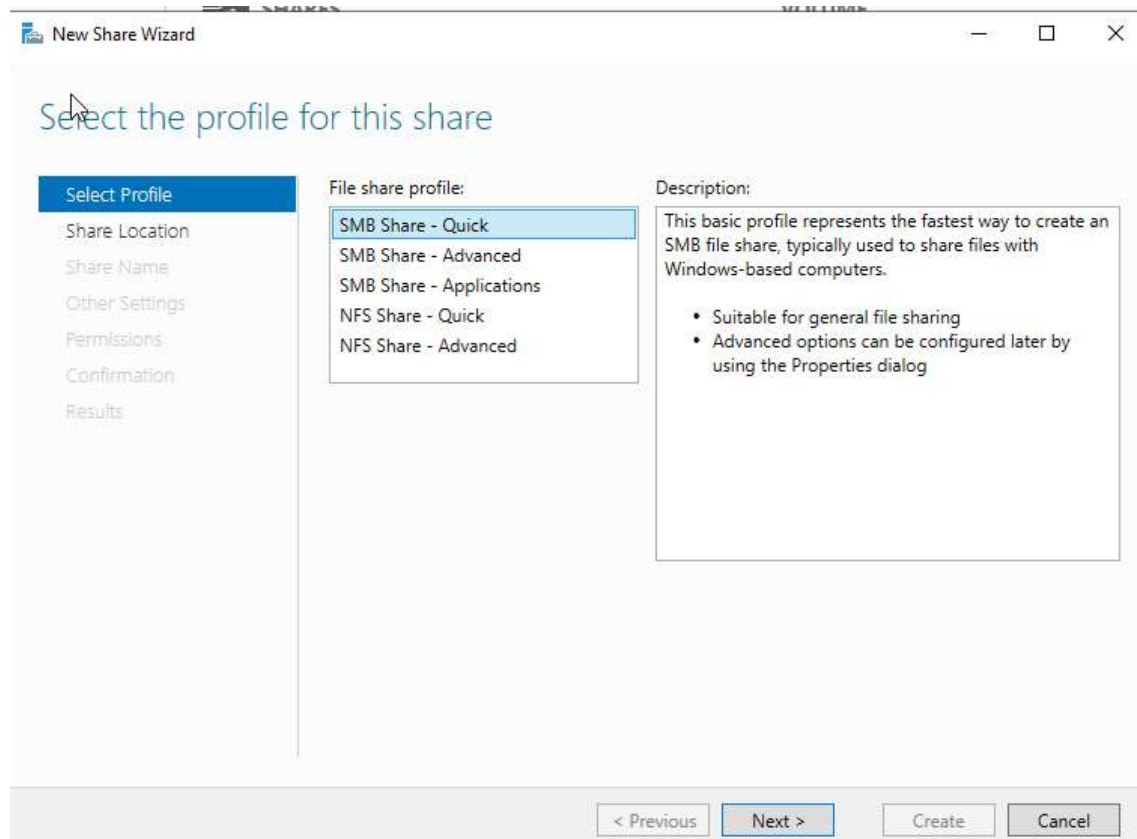
En nuestro cliente FreeBSD montamos la compartición de NFS:

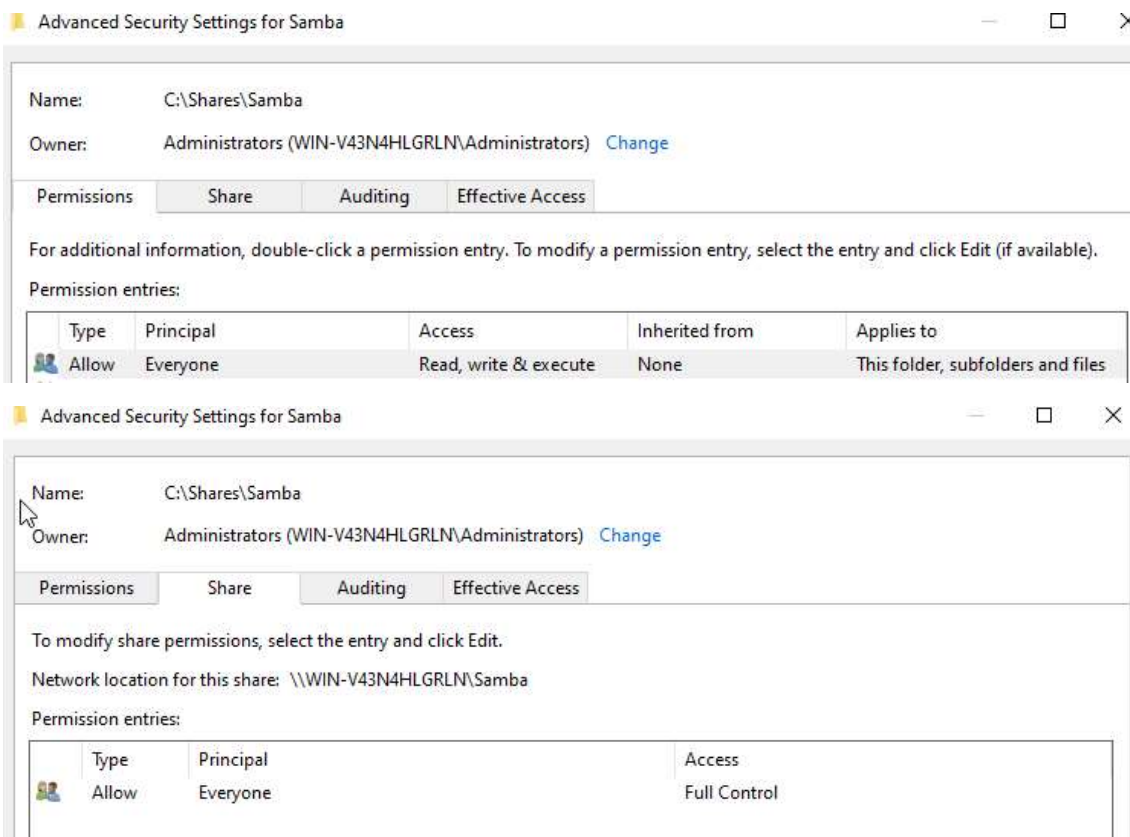
`mount -t nfs 192.168.137.223:\Shared /NFS-win`

```
root@FBSDaWin:/ # mount -t nfs 192.168.137.223:\Shared /NFS-win
root@FBSDaWin:/ # ls
.cshrc      bin         home        net         sys
.profile   boot       lib         proc        tmp
.xinitrc    dev        libexec     rescue      usr
COPYRIGHT  entropy    media       root        var
NFS-win     etc        mnt        sbin        zroot
root@FBSDaWin:/ # cd NFS-win/
root@FBSDaWin:/NFS-win # touch prueba.txt
root@FBSDaWin:/NFS-win # ls
prueba.txt
root@FBSDaWin:/NFS-win #
```



3.7 SAMBA





Desde el cliente comprobamos que funciona:

```
root@FBSDaWin:/NFS-win # smbclient //192.168.137.223/Samba -U administrator
smbclient: Can't load /usr/local/etc/smb4.conf - run testparm to debug it
Enter WORKGROUP\administrator's password:
Try "help" to get a list of possible commands.
smb: \> ! echo "Hola (cliente)" >> prueba2.txt
smb: \> put prueba2.txt
putting file prueba2.txt as \prueba2.txt (1.1 kb/s) (average 1.1 kb/s)
smb: \>
```

