-

UD 10. LINUX: NETWORKING

Computer Systems
CFGS DAW

Borja Salom
b.salomsantamaria@edu.gva.es
2022/2023

Version:230217.1050

Licence

Nomenclature

Throughout this unit different symbols will be used to distinguish important elements within the content. These symbols are:

 Importante

 Atención

 Interesante

ÍNDICE DE CONTENIDO

UD10. LINUX: NETWORKING

1. Samba

Samba is a communication protocol to connect windows with linux file system and thus see linux disks or folders as network drives in windows.

Installing and configuring samba is quite simple, you have to install it from the command line:

```
apt-get install samba
```

Now you have to configure it to access linux folders from windows.

To configure it you have to edit the file

```
;[profiles]
; comment = Users profiles
; path = /home/samba/profiles
; guest ok = no
; browseable = no
; create mask = 0600
; directory mask = 0700
```

We copy and modify it as we want our directory to be displayed in windows:

```
[test_samba]
  comment = samba test directory
  path = /home/user/samba_test
  guest ok = no
  browseable = yes
  create mask = 0600
  directory mask = 0700
```

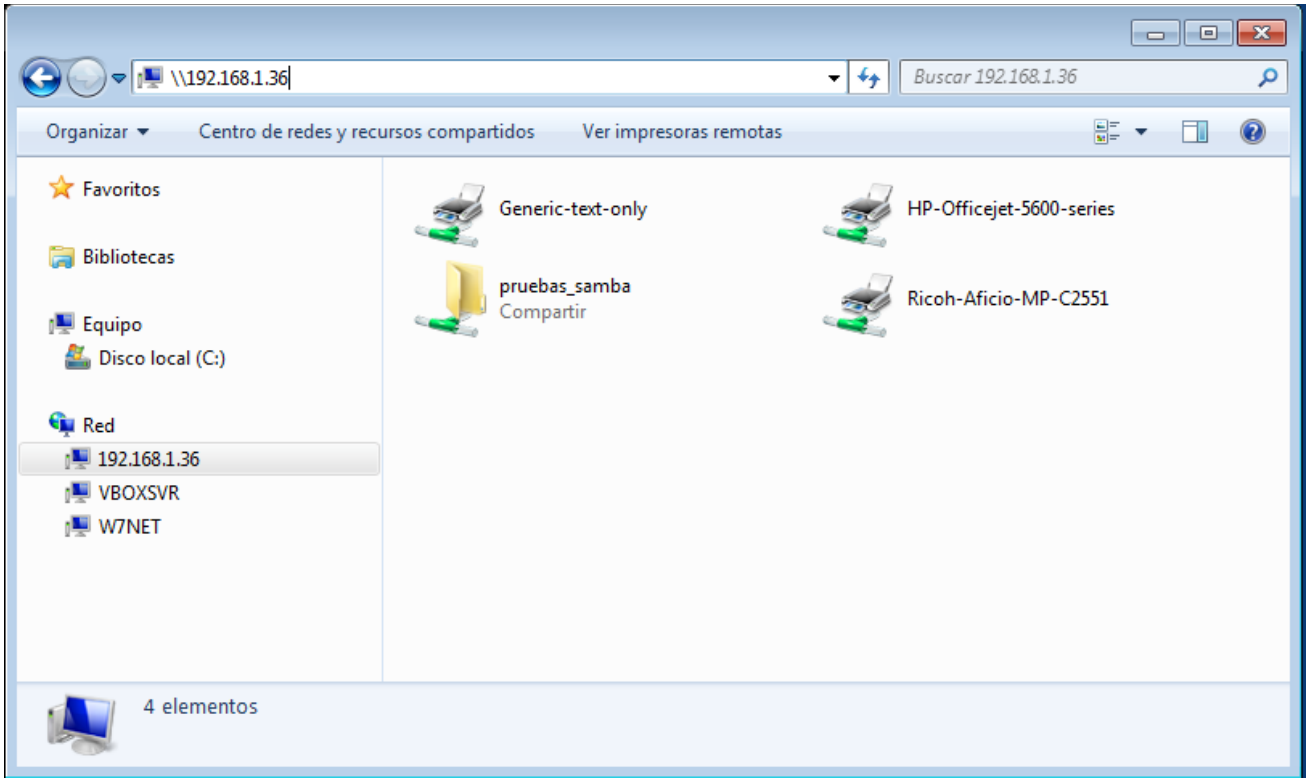
Then we put a samba password to the user that we want to have access to the directory

```
sudo smbpasswd -L -a username
# we put a password
sudo smbpasswd -L -e username
```

We enable the user for samba and restart the service

```
sudo /etc/init.d/smbd restart
```

Now we go to windows and write our IP in the windows explorer.



Now we will test that we can write to the directory in order to work.

If we are sharing web directories in /var/www or something like that, you may have access problems but this can be solved by putting the following directives in the configuration file:

```
force user = root
force group = root
```

2. NFS Server

A Network File System (NFS) server is a system that allows clients to access remote file systems over the network.

The NFS server shares one or more file systems with clients using the NFS protocol, which is an Internet standard protocol for sharing files between Unix/Linux systems.

NFS clients can mount the remote file systems from the server as if they were local file systems, allowing them to access remote files and directories transparently as if they were on their own system.

The NFS server is widely used in Unix/Linux networked computer environments for sharing file storage resources and allowing data access over the network efficiently.

2.1 Installing NFS Server on Linux

What we are going to do first is update our servers:

```
$ sudo apt update && apt dist-upgrade
```

Once it's up to date and we reboot, we're going to install NFS.

```
$ sudo apt install -y nfs-kernel-server
```

The next step is to create the shared folder, in my case, it would look like this.

```
$ sudo mkdir -p /mnt/data
```

Let's specify some permissions so that "all" clients can write to that directory.

```
$ sudo chown nobody:nogroup /mnt/data
$ sudo chmod 777 /mnt/data
```

The next thing we're going to do is configure NFS to only accept connections from our clients. If this is not done, when we want to connect from a client, NFS will give us access denied.

To do this, we are going to edit the "export" file

```
$ sudo nano /etc/exports
```

The next thing is to add the folder that we want to share and we specify which will be the IP addresses that can write to that folder, for my example it would look like this:

```
/mnt/data       192.168.1.80(rw,sync,no_subtree_check)
/mnt/data       192.168.1.81(rw,sync,no_subtree_check)
/mnt/data       192.168.1.82(rw,sync,no_subtree_check)
```

And the complete file would look like this:

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#

/mnt/data       192.168.1.80(rw,sync,no_subtree_check)
/mnt/data       192.168.1.81(rw,sync,no_subtree_check)
/mnt/data       192.168.1.82(rw,sync,no_subtree_check)
```

Basically what I'm doing with this is letting my docker swarm cluster be able to access and write to that shared folder.

Once we have this ready, we are going to "publish" these exports:

```
$ sudo exportfs -a
```

The next thing is to restart the NFS server in this way so that it takes those changes:

```
$ sudo systemctl restart nfs-kernel-server
```

As an optional step, if there is a firewall activated on the servers, remember to make exceptions so that the traffic between clients and server flows.

2.2 Configure the clients

Something that I did not mention is that the IP address that I assigned to my NFS server is IP 192.168.1.83, that being said, we are going to install the NFS client on our nodes.

```
$ sudo apt update & apt install nfs-common
```

Once it is installed, we are going to create a local folder, to mount the remote folder. In this case, to avoid confusion, I'm going to call it "client1" but I like that they have the same name so I can standardize configurations.

```
$ sudo mkdir -p /mnt/cliente1
```

Once we have the folder, we are going to mount the remote directory in this one that we just created, it would look like this:

```
$ sudo mount 192.168.1.83:/mnt/data /mnt/cliente1
```

From then on, when they browse to "/mnt/client1" they will see the contents of "/mnt/data" from the NFS server.

Finally, the question surely comes, how do I make this change persistent and stay through reboots? and I answer you, what a good question.

So that the mount is done automatically when booting, we have to touch the /etc/fstab:

```
$ sudo nano /etc/fstab
```

In this file we will see the disk or disks that we have and we will add a line that is our NFS. The line should be something like this:

```
192.168.1.83:/mnt/data /mnt/data nfs defaults 0 1
```

And the file is, in my case, like this:

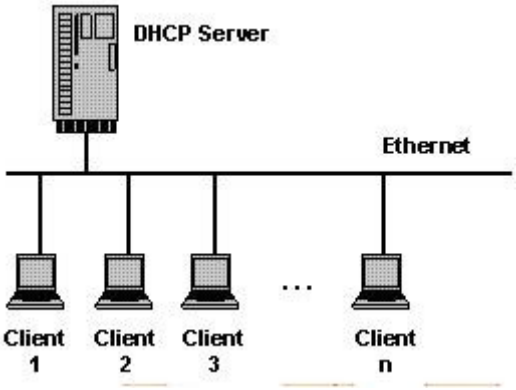
```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/726855df-27f8-4d31-9382-e71914e3bafb / ext4 defaults 0 0
/swap.img       none    swap    sw      0       0

192.168.1.83:/mnt/data /mnt/data nfs defaults 0 1
```

We save, close and restart. When we log in, we go to /mnt/client1 and we should see the remote folder mounted.

3. DHCP Protocol

3.1 What is DHCP?

A Dynamic Host Configuration Protocol (DHCP) server dynamically assigns IP addresses and other settings for a given network to other client computers that are connected to the network. This simplifies network administration and makes connecting new equipment to the network much easier.



ll the IP addresses of all the computers are stored in a database that resides on a server.

A DHCP server can provide configuration settings using two methods.

Address Range

This method is based on the definition of a pool of IP addresses for DHCP clients (also called IP address pool) that supply their configuration properties dynamically as requested by client computers. When a DHCP client is no longer on the network for a specified period, the configuration expires and the polled IP address is released for use by other DHCP clients.

MAC address

This method is based on using the DHCP protocol to identify the unique hardware address of each network card connected to the network and then a constant configuration is assigned as well as the same IP address each time the client's DHCP configuration makes a request. to the DHCP server from the same network device.

3.2 Install a DHCP service on Ubuntu and Debian

To install the automatic IP address assignment server, we execute the command:

```
sudo apt-get install dhcp3-server
```

This simple step installs the server on our linux.

3.3 Configuring the DHCP server

In the event that you have two network interfaces (NICs) on your Linux server, you have to select which one you are going to use to listen to DHCP requests. To configure the service, we edit the `/etc/default/dhcp3-server` file , and change `INTERFACES=”eth0″` to the internal network card.

It is necessary to make a backup copy of the configuration file:

```
cp /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf.back
```

Configure using the address range method (IP pool)

We edit the configuration by typing:

```
sudo nano /etc/dhcp3/dhcpd.conf
```

And in this file we change the following sections

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option domain-name-servers 192.168.1.9, 192.168.1.10;
option domain-name “guatewireless.org”;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.200;
}
```

We save and exit the file. The above text configures the DHCP server with the following parameters:

- Assignment to clients IP addresses in the range of 192.168.1.10 to

192.168.1.200

- It will provide the IP address for a minimum of 600 seconds, and a maximum of 7200 seconds.
- Set the subnet mask to 255.255.255.0
- Broadcast address of 192.168.1.255
- As gateway/network gateway/router the address 192.168.1.1
- And the servers 192.168.1.9 and 10 as your DNS servers

IP reservations for certain machines

For this at the end of the file we add the following block

```
# SERVER
host SERVER {
hardware ethernet xx:xx:xx:xx:xx:xx;
fixed-address 192.168.1.210;
option broadcast-address 192.168.0.255;
}
```

Where SERVER is the name of the machine, xx:xx:xx:xx:xx:xx is the mac address of the machine, 192.168.1.210 is the IP we want to reserve and 192.168.0.255 is the broadcast address

We save the file we restart the dhcp server using.

```
etc/init.d/dhcp3-server restart
```

Configure using the MAC address method

With this method you can reserve some or all of the IP addresses of our network for certain machines. As you can see, the configuration is very similar to the previous one, with the exception that to reserve the assignment of an IP to a certain NIC (network card interface) we must use the host label

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
```

```
option domain-name-servers 192.168.1.9, 192.168.1.10;
option domain-name “guatewireless.org”;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.200;
}
host oracle{
hardware ethernet 00:03:47:31:e1:7f;
fixed-address 192.168.1.20;
}
host printer {
hardware ethernet 00:03:47:31:e1:b0;
fixed-address 192.168.1.21;
}
```

Now we restart the dhcp server by executing the following command:

```
sudo /etc/init.d/dhcp3-server restart
```

3.4 Configuring the DHCP client on Linux Ubuntu

If you want to configure a desktop or machine with linux as a DHCP client, follow these steps:

- We edit the network interfaces file

```
sudo vi /etc/network/interfaces
```

- We must have the following lines, taking into account that eth0 is an example

```
auto lo eth0
iface eth0 inet dhcp
iface lo inet loopback
```

- We save and exit the file
- We restart the network services of Linux Ubuntu

```
sudo /etc/init.d/networking restart
```

In order to know the addresses assigned to client machines

```
tail -n 15 /var/lib/dhcp3/dhclient.*.leases
```