

Este proyecto se ha realizado para el último módulo del Ciclo de Administración de Sistemas Informáticos en Red y va dirigido al estudio e implantación de un sistema de almacenamiento iSCSI.

PROYECTO INTEGRADO FIN DE CICLO 2017

Almacenamiento iSCSI

FRANCISCO MANUEL HIDALGO FERNÁNDEZ

INDICE

1. Idea del proyecto	Pág.- 03
2. Introduccion SCSI	Pág.- 04
3. Introduccion iSCSI	Pág.- 04
4. Diferencias con Samba o NFS	Pág.- 05
5. Razones para usar iSCSI	Pág.- 05
6. Velocidad de las transferencias	Pág.- 06
7. Acceso a los datos	Pág.- 06
8. Multipath	Pág.- 07
9. Sistema de ficheros OCFS2	Pág.- 07
10. Arquitectura iSCSI	Pág.- 08
10.1. Targets	Pág.- 09
10.2. Initiator	Pág.- 10
11. Software para implementar iSCSI	Pág.- 11
11.1. Servidor iSCSI	Pág.- 11
11.2. Clientes Linux	Pág.- 11
11.3. Clientes Windows	Pág.- 11
12. Otros Software usados en el proyecto	Pág.- 12
12.1. Servidor iSCSI	Pág.- 12
12.2. Clientes Linux	Pág.- 12
12.3. Clientes Windows	Pág.- 12

13.	Comandos usados en iSCSI	Pág.- 13
13.1.	Servidor iSCSI	Pág.- 13
13.2.	Cientes Linux	Pág.- 13
13.3.	Cientes Windows	Pág.- 13
14.	Ficheros editados para configuración iSCSI	Pág.- 14
14.1.	Servidor iSCSI	Pág.- 14
14.2.	Cientes Linux	Pág.- 14
14.3.	Cientes Windows	Pág.- 14
15.	Caso práctico del Proyecto	Pág.- 15
15.1.	Configuración Router	Pág.- 17
15.2.	Configuración Servidor iSCSI	Pág.- 19
15.2.1.	Configuración Red y Raid	Pág.- 19
15.2.2.	Configuración iSCSI	Pág.- 22
15.3.	Configuración Clientes Linux	Pág.- 27
15.3.1	Implantación de Sistema de Ficheros OCFS2	Pág.- 30
15.4.	Pruebas Raid	Pág.- 37
15.5.	Configuración Clientes Windows	Pág.- 40
16.	Infraestructuras y tecnologías usadas	Pág.- 50
16.1.	Infraestructura	Pág.- 50
16.2.	Virtualización	Pág.- 50
16.3.	Tecnologías usadas	Pág.- 50
17.	Fuentes de información	Pág.- 52

1. Idea del Proyecto

La idea para el Proyecto y como complemento de formación al ciclo de Administración de Sistemas Informáticos en Red, es estudiar e implementar un servicio de almacenamiento no visto durante dicho ciclo y que pienso que es bastante interesante.

Este sistema de Almacenamiento sería; iSCSI.

En el proyecto se montará un escenario donde habrá un servidor (Linux) con un par de discos conectados y que tendrán las mismas características.

Habrán también clientes (Linux y Windows) a los que se les creará una unidad de disco iSCSI para aumentarles la capacidad de almacenamiento y poder centralizar los datos en el servidor iSCSI.

El servidor replicará la estructura y el contenido el disco que está sirviendo a los clientes en el otro disco que tiene, creando un RAID1 por si hay fallos un uno de los discos no perder la información.

Todo se montará bajo dos redes privada que harán las funciones; una de sacar las máquinas Internet y la otra crear una red interna para distribuir los IQN iSCSI.

Al final el objetivo será implementar un sistema de almacenamiento emulando una pequeña empresa en la que los clientes tienen sus datos centralizados en un servidor.

Pienso que puede ser interesante el estudio de esta tecnología.

2. Introducción SCSI

Vamos a comenzar viendo que es SCSI para comprender mejor iSCSI.

SCSI es un estándar que se usa para definir a las conexiones de la placa base con los componentes del PC.

Si pensamos en SCSI, pensamos en dispositivos de almacenamiento como discos duros, pero no solo se usa en esos casos la conexión SCSI sino que también se usa en escáneres, lectoras de CDs, grabadoras, Cintas de almacenamientos...

SCSI se usa (y cada vez menos) en servidores ya que es una tecnología cara.

Sirve para poder conectar muchos discos simultáneamente en una máquina. Solo se plantea el uso de este tipo de conexión si se piensan conectar más de 4 o 5 dispositivos a la vez. Su uso conlleva a crear RAID y mantener los datos en alta disponibilidad. Es lógico puesto que al tener más discos tenemos mayor probabilidad de fallos y de perder la información almacenada.

Ahora que tenemos una idea de SCSI vamos a ver lo que nos interesa, iSCSI.

3. Introducción iSCSI

iSCSI es una extensión de SCSI y abreviatura de Internet SCSI.

Como vimos anteriormente SCSI es un estándar de conexión de dispositivos, así que con iSCSI tendremos conexión de dispositivos pero por red.

Usa el protocolo de la capa de transporte de la pila TCP/IP, definido en las especificaciones SCSI-3.

Al ser la comunicación por red podremos tener varias “unidades de disco” en una o varias máquinas pero en vez de tenerlas conectadas físicamente, las podremos tener remotamente conectadas por red.

4. Diferencias con Samba o NFS

Los sistemas Samba y los sistemas NFS trabajan importando al cliente un sistema de archivos mediante la red. Por ello solo vemos como una unidad donde almacenar datos.

iSCSI sin embargo importa por la red todo el “hardware” de manera que en la máquina cliente es detectado como un dispositivo “físico” y puede ser tratado como tal. Todo de forma totalmente transparente para el usuario.

5. Razones para usar iSCSI

Tenemos varias razones por la que usar iSCSI.

Una de esas razones es la de poder centralizar el almacenamiento de una serie de maquina en un servidor.

Otra de las razones es la de tener independencia de datos, cada máquina tiene su almacenamiento y no influyen en nada con el almacenamiento de otra máquina con la que el servidor comparte su disco.

Podemos tener también maquina sin discos físicos conectados, de forma que si la maquina falla o hubiese que hacerle algún tipo de mantenimiento podemos coger el “disco” que le estamos pasando por red y conectarlo a otra máquina para así tenerla funcionando un poco tiempo.

6. Velocidad en las transferencias

La velocidad de transferencia es lo más importante a la hora de la elección de un disco.

Los discos SCSI suelen dar muy buenos rendimientos en cuanto a velocidad aunque en la actualidad están teniendo mucho éxito los discos con conexión SATA ya que el rendimiento también es muy bueno y sus costes son más bajos.

Como el funcionamiento de iSCSI es pasa todo el “disco” por la red podemos pensar que su rendimiento no va a ser del todo bueno. Todo depende del tipo de conexión del que dispongamos.

Al elegir montar iSCSI se recomienda que la red de la que se disponga sea una red de alta velocidad (Gigabit Ethernet) si lo que queremos es tener un almacenamiento con un buen rendimiento, pero puede funcionar también en redes más lentas (Fast Ethernet) aunque si lo montamos en una red lenta no podremos pedir mucho rendimiento a iSCSI.

La velocidad de transferencia del iSCSI es de 1000 MB/Seg, aunque debido al protocolo, la velocidad baja hasta 800 MB/Seg. En caso de que utilicemos tarjetas DUAL CHANNEL, como las que tienen los QNAP, podremos llegar a 1600 MB/Seg, teniendo en cuenta las pérdidas por protocolo.

7. El acceso a los datos

El acceso a los datos en iSCSI es uno de los pocos inconvenientes que tiene este tipo de tecnologías ya que un mismo IQN no debería de conectarse a dos maquina a la vez ya que pueden haber inconsistencias en los datos.

Para poder tener un directorio donde poder escribir varias maquina a la vez están las opciones de NFS o Samba.

En los sistemas operativos RedHat existe la posibilidad conexiones múltiples con el global filesystem (GFS).

Pueden haber varias conexiones desde el iniciador al target para mantener una alta disponibilidad en la conexión mediante Multi-Path.

8. Multi-Path en iSCSI

A la hora de montar un sistema de almacenamiento lo que siempre vamos a intentar buscar es el máximo rendimiento, optimizar recursos, minimizar los tiempos de respuesta y evitar sobrecargas o cuellos de botella.

Para ello en los sistemas iSCSI podemos implementar el Multi-Path que es una conexión de rutas múltiples que ofrecen más fiabilidad en cuanto al acceso y conexión, además de mejorar el rendimiento en la velocidad.

Multi-Path cuenta con gran soporte y es compatible con distintas tecnologías, incluidos controladores de disco y protocolo iSCSI. Además, cuenta con mayor respaldo por parte de varias empresas de software, como Linux, VMware y Microsoft.

La técnica que usa Multi-Path es la de tener dos conexiones al dispositivo de almacenamiento en lugar de una. Así si proporciona una ruta alternativa en caso de que una de las conexiones caiga o se sature.

Esto hace que tengamos una mayor tolerancia a fallos en la conexión.

9. Sistema de fichero OCFS2

OCFS2 es un sistema de ficheros para nodos.

Esta herramienta nos permite que 2 máquinas o más (no mejor pasar de 3 nodos) accedan simultáneamente a un sistema de ficheros.

Es especialmente recomendable para ambientes donde hay que compartir datos.

Con esta herramienta podemos crear también tener información en alta disponibilidad.

10. La arquitectura iSCSI

La arquitectura de iSCSI se divide básicamente en dos partes.

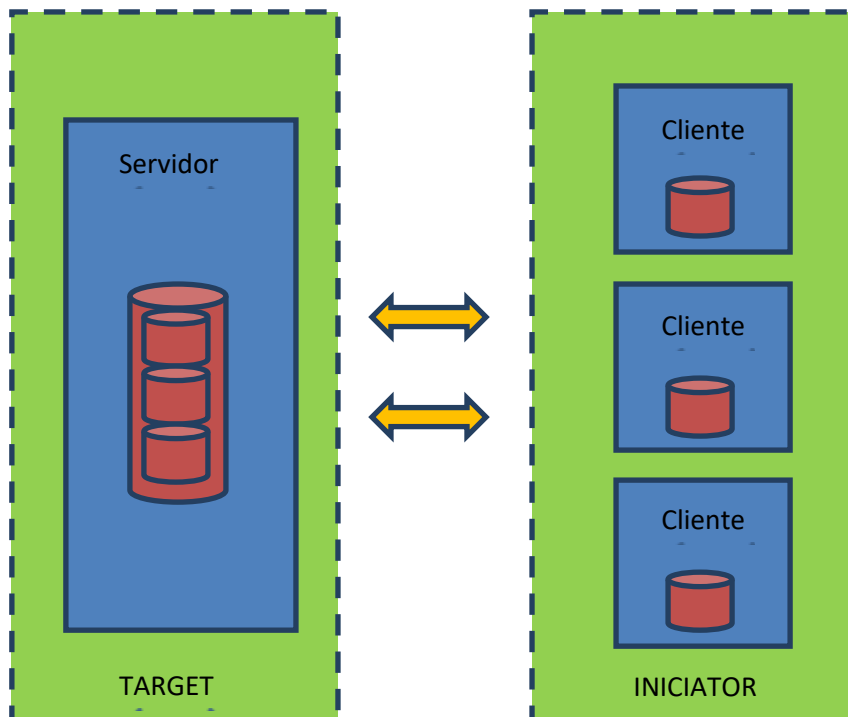
El target (parte del servidor).

El iniciador (parte del cliente).

Vemos en la imagen una representación gráfica de la arquitectura.

La zona “Azul” representaría las máquinas físicas (el servidor y los clientes)

La zona “Verde” representa la parte a la que correspondería dentro de la arquitectura iSCSI (el servidor = target y los clientes = iniciador)



10.1. El target iSCSI

El target iSCSI es en pocas palabras el servidor.

Un target puede ofrecer uno o más recursos iSCSI por la red. En las soluciones Linux para iSCSI, no hace falta que el dispositivo a exportar sea necesariamente un disco SCSI; medios de almacenamiento de distinta naturaleza se pueden usar, como por ejemplo:

- Particiones RAID
- Particiones LVM
- Discos enteros
- Particiones comunes
- Archivos
- Dispositivos de CD, cintas, etc.

En el target se crean IQN.

El IQN (iSCSI Qualified Name) son los tickets que se crean y que posteriormente se pasan a los iniciadores (Clientes).

IQN se pone en el inicio del nombre del ticket y puede tener hasta 255 caracteres de longitud, con el siguiente formato:

`iqn.year-mo.reverse_domain_name:unique_name`

`iqn` es una parte fija. Siempre hay que ponerlo.

`year-mo` representa el año y el mes en que se registró el dominio.

`reversed_domain_name` es el nombre del dominio oficial, en orden inverso.

`unique_name` es cualquier nombre que queramos utilizar. Suele utilizarse el hostname del servidor.

Un ejemplo de IQN sería el siguiente (este ejemplo será el que veamos durante la parte práctica del proyecto):

`iqn.2017-09.org.fran.tg1`

10.2. El iniciador iSCSI

El iniciador es el cliente de iSCSI.

En Linux se usa el iniciador “Open iSCSI” y en los sistemas Windows viene por defecto un iniciador instalado, estos hacen que el sistema operativo pueda reconocer discos de tipo iSCSI y un programa que gestiona las conexiones a dichos discos.

Existen iniciadores para una amplia variedad de sistemas operativos.

Demás está decir que iSCSI es independiente del sistema operativo. Podemos tener un target montado en Linux y utilizar los discos en dicho target en un sistema Windows, MacOSX o incluso Solaris.

11. Software usado para implementar iSCSI

11.1. SERVIDOR iSCSI

Vamos a empezar viendo los software que nos harán falta para montar el Servidor iSCSI.

Nos hará falta el software mdadm con el que podremos crear RAID.

```
apt-get install mdadm
```

Nos hará falta el software lvm2 con el que podremos crear volúmenes lógicos.

```
apt-get install lvm2
```

Nos hará falta el software tgt para la creación de tickets.

```
apt-get install tgt
```

Otro paquete es el propio de iSCSI. Se instalara tanto en el servidor como en el cliente.

```
apt-get install open-iscsi
```

11.2. CLIENTES Linux

Instalaremos el paquete para iSCSI

```
apt-get install open-iscsi
```

Instalaremos también Apache2 para poder servir desde los clientes una página web.

11.3. CLIENTES Windows

En los clientes Windows no es necesario la instalación de ningún software ya que desde Windows 7 en adelante viene instalado de forma predeterminada.

Si tenemos que usar una versión anterior de Windows entonces si es necesario descargar e instalar el software iniciador.

12. Otros Software usados para el proyecto

12.1. SERVIDOR iSCIS

No se ha necesitado ningún software más aparte de lo descrito en el apartado 11.

12.2. CLIENTES Linux

Instalaremos el paquete para apache2, con este software lograremos montar un servidor web para servir una página web.

```
apt-get install apache2
```

Instalaremos también ocfs2-tools para crear un sistema de ficheros distribuido y poder crear así un cluster con los dos Clientes Linux .

```
apt-get install ocfs2-tools
```

12.3. CLIENTES Windows

No ha hecho falta la instalación de ningún software en la parte de Windows.

13. Comandos usados para iSCSI.

Vamos a ver de forma detallada los comandos que vamos a utilizar en la configuración de iSCSI.

13.1. SERVIDOR iSCIS

Comando usados por el servido.

Con este comando podemos ver los IQN configurados en el Servidor.

```
iscsiadm --mode discovery --type st --portal 10.0.10.2
```

13.2. CLIENTES Linux

Comandos usados por el cliente Linux.

Vemos ahora los comando que nos harán falta durante la configuración de la máquina Cliente Linux iSCSI.

Con este comando al igual que en el Servidor, podemos ver los IQN configurados en el Servidor y que se están sirviendo al cliente.

```
iscsiadm --mode discovery --type sendtargets --portal 10.0.10.2
```

Con el siguiente comando agregamos al cliente el IQN que elijamos del servidor.

```
iscsiadm --mode node --target name iqn.2017-09.org.fran.tg1 --portal 10.0.10.2 --login
```

Podemos hacer que la conexión sea persistente y que al reiniciar la maquina se conecte el IQN de manera automática solo reiniciando el servicio con el siguiente comando.

```
iscsiadm -m node --targetname iqn.2017-09.org.fran.tg1 -p 10.0.10.2 -o update -n node.conn[0].startup -v automatic
```

13.3. CLIENTE Windows

Para la configuración del iniciador en el cliente Windows no vamos a usar comando ni en la cmd ni en la powershell, lo vamos a hacer de forma gráfica a través de ventanas de Windows.

14. Ficheros de configuración iSCSI

Vamos a ver los ficheros que debemos de editar para la configuración de iSCSI.

14.1. SERVIDOR iSCSI

En este fichero lo que vamos a configurar va a ser los IQN que se van a ver en los clientes.

```
/etc/tgt/targets.conf
```

En este fichero configuraremos la posibilidad de autenticar IQN en los clientes mediante clave.

```
/etc/iscsi/iscsid.conf
```

14.2. CLIENTES Linux

En la configuración de iSCSI en las máquinas Linux no vamos a necesitar configurar ningún fichero.

14.3. CLIENTE Windows

En la configuración de iSCSI en las máquinas Windows no vamos a necesitar configurar ningún fichero.

15. Caso Práctico del Proyecto

Para hacer un ejemplo de implantación de iSCSI, vamos a crear para este proyecto un escenario con máquinas virtuales en VMWare.

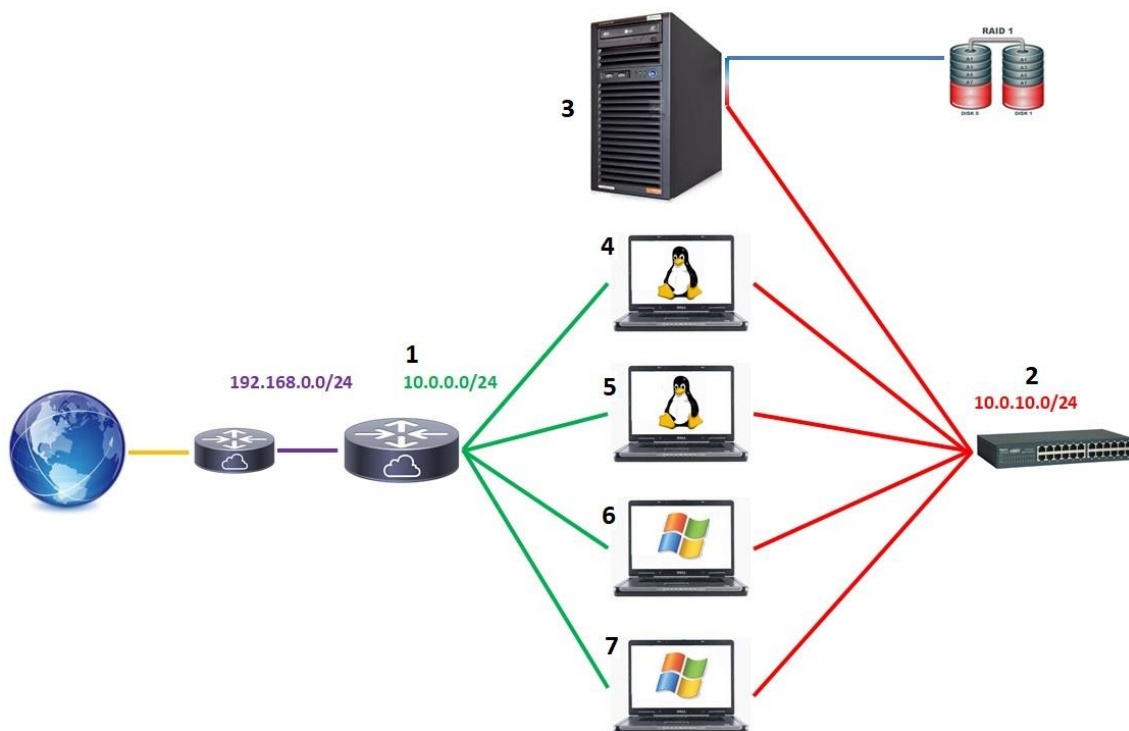
La idea es crear un entorno donde un servidor iSCSI (3) va a servir IQN a los distintos clientes (4, 5, 6 y 7).

El servidor tendrá dos discos de 10 GB cada montado en RAID1.

Los clientes Linux (4 y 5) van a recibir un IQN de 2 GB y se creará un Cluster con las dos maquina dándoles un sistema de ficheros OCFS2.

Las maquinas Windows solo recibirán los IQN como ampliación de disco para la máquina.

A continuación vemos una imagen para ver la composición del escenario.



Vamos a detallar las máquinas de este escenario:

1. Un Router (Debian jessie) con una Red que sale a internet (red morada en la imagen) y otra red privada para las máquinas clientes (10.0.0.0/24 red verde en la imagen). Este Router da IP por DHCP a las máquinas clientes que se conectan a la red privada.

2. Switch para conectar la red privada (10.0.10.0/24 red roja en la imagen).

3. Un Servidor de disco iSCSI (Debian jessie) que estará conectado solo a la red privada (10.0.10.0/24 red roja en la imagen). Esta máquina es la que sirve los target iSCSI a través de la red. Tendrá también dos discos (de 10 GB cada uno) que estarán configurados con un RAID1 para tener alta disponibilidad de datos.

Puntualmente se conectara a la red 10.0.0.0/24 para poder hacer la instalación de los paquetes que sean necesarios, una vez estén instalados se le quitara esa conexión.

4, 5, 6 y 7. Cuatro Clientes iSCSI (dos cliente Debian jessie y dos clientes Windows 7) que estarán conectados a las dos redes privadas (red verde y red roja de la imagen). Reciben las IP a través de DHCP por la 10.0.0.0 y de forma estática la red 10.0.10.0. Estas máquinas pueden salir a internet (a través red privada verde y red morada de la imagen). Reciben los IQN iSCSI a través de la red privada (10.0.10.0/24 red roja en la imagen)

15.1. CONFIGURACION ROUTER DEBIAN (1)

Comenzamos con la configuración de la máquina que va a actuar como router.

Este router es por el que saldrán a internet el resto de máquinas.

Vamos a ver la configuración de esta máquina.

Esta configuración la haremos con el usuario “root”

1. Configuración de interfaces de red (vemos el fichero completo)

```
nano /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 interface
auto eth0
iface eth0 inet dhcp

up iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth0 -j SNAT --to
[IP_ETH0]

#ip interna
auto eth1
iface eth1 inet static
address 10.0.0.1
netmask 255.255.255.0
network 10.0.0.0
broadcast 10.0.0.255
```

2. Activamos Forward

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
nano /etc/sysctl.conf
```

3. Descomentamos la siguiente línea

```
net.ipv4.ip_forward=1
```

4. Reiniciamos el Servicio networking

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

5. Configuración de un servicio DHCP

5.1. Instalamos el siguiente paquete

```
apt-get install isc-dhcp-server
```

5.2. Editamos el fichero isc-dhcp-server para poner la interfaz por donde servirá el dhcp

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

Ponemos la interfaz por donde servirá el servicio DHCP las IP

```
INTERFACES="eth1"
```

5.3. Editamos el fichero dhcpd.conf agregando al final del fichero la siguiente configuración.

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

```
#Configuracion DHCP
subnet 10.0.0.0 netmask 255.255.255.0
{
    #rango
    range 10.0.0.2 10.0.0.10;
    #puerta de enlace
    option routers 10.0.0.1;
    #dns
    option domain-name-servers 8.8.8.8;
    #broadcast
    option broadcast-address 10.0.0.255;
}
```

5.4. Reiniciamos el servicio dhcp después de los cambios

```
/etc/init.d/isc-dhcp-server restart
```

15.2. SERVIDOR iSCSI (3)

15.2.1. Configuración Red y RAID

En esta máquina vamos a tener en principio dos interfaces de red. Las dos van a recibir IP a través del servicio dhcp.

Una vez se hayan instalado los paquetes necesarios y se hayan hecho las configuraciones oportunas, se desactivara la interfaz de red que sale a internet (10.0.0.0).

Vamos a empezar configurando la red.

1. Configuración de interfaces de red (vemos el fichero completo).

```
nano /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

# Interfaz eth0
auto eth0
iface eth0 inet dhcp

# Interfaz eth1
auto eth1
iface eth1 inet static
address 10.0.10.2
netmask 255.255.255.0
network 10.0.10.0
```

2. Reiniciamos el Servicio networking

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

En esta máquina vamos a contar también con dos discos adicionales de 10 GB cada uno.

Estos discos formaran un RAID1 para tener en alta disponibilidad nuestros datos.

Al formar un RAID1 la capacidad de almacenamiento que tendremos será de 10 GB ya que los otros 10 GB están actuando de espejo.

3. Instalamos el paquete mdadm

```
apt-get install mdadm
```

```
ALL
```

4. Vemos los discos con los que contamos

```
lsblk
```

```
root@debian:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   12G  0 disk
├─sda1       8:1    0 11,5G  0 part /
├─sda2       8:2    0    1K  0 part
└─sda5       8:5    0  549M  0 part [SWAP]
sdb          8:16   0   10G  0 disk
sdc          8:32   0   10G  0 disk
sr0         11:0    1 1024M  0 rom
```

Cogemos los discos que queremos reflejar. (ej. /dev/sdb y /dev/sdc)

5. Creamos el RAID1 (md0 es el nombre del "nuevo disco" que se crea de la fusión de /dev/sdb1 y /dev/sdc1)

```
mdadm -Cv /dev/md0 -l1 -n2 /dev/sdb1 /dev/sdc1
```

```
root@debian:~# mdadm -Cv /dev/md0 -l1 -n2 /dev/sdb1 /dev/sdc1
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
mdadm: size set to 10476544K
Continue creating array? yes
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

6. Para comprobar el proceso podemos ejecutar el siguiente comando

```
cat /proc/mdstat
```

```
root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      10476544 blocks super 1.2 [2/2] [UU]
      [=>.....] resync = 9.2% (966144/10476544) finish=3.5min speed=4
3915K/sec
```

```
unused devices: <none>
root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      10476544 blocks super 1.2 [2/2] [UU]
unused devices: <none>
```

6.1. Par más información del RAID

```
mdadm --query /dev/md0
mdadm --detail /dev/md0
```

```
root@debian:~# mdadm --query /dev/md0
/dev/md0: 9.99GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
root@debian:~# mdadm --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Tue Dec 19 12:57:12 2017
  Raid Level : raid1
  Array Size : 10477568 (9.99 GiB 10.73 GB)
  Used Dev Size : 10477568 (9.99 GiB 10.73 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Tue Dec 19 13:01:31 2017
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0


  Name : debian:0 (local to host debian)
  UUID : a337e97e:7a51551a:35e1fa31:35e31482
  Events : 17


   Number   Major   Minor   RaidDevice State
    0         8       16         0     active sync   /dev/sdb
    1         8       32         1     active sync   /dev/sdc
root@debian:~# █
```

15.2.2. Configuración iSCSI

Una vez tenemos la red y los discos del servidor preparados, vamos a proceder con la configuración iSCSI.

Para ellos vamos a crear cinco volúmenes en el RAID de 2 GB cada uno.

Estos volúmenes serán los que se asocien a los target y que posteriormente se distribuirán a los clientes.

Vamos a ver la configuración y lo iremos explicando paso a paso para que quede más claro.

1. Instalamos el paquete para crear volúmenes lógicos

```
apt-get install lvm2
```

2. Vemos los discos con los que contamos

```
lsblk
```

```
root@debian:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda          8:0    0   12G  0 disk
├─sda1       8:1    0  11,5G  0 part  /
├─sda2       8:2    0    1K  0 part
└─sda5       8:5    0   549M  0 part  [SWAP]
sdb          8:16   0    10G  0 disk
└─md0        9:0    0    10G  0 raid1
sdc          8:32   0    10G  0 disk
└─md0        9:0    0    10G  0 raid1
sr0         11:0    1  1024M  0 rom
```

3. Preparamos el discos que tenemos para crearles volúmenes (en este caso el disco del RAID md0)

```
pvccreate /dev/md0
```

```
root@debian:~# pvccreate /dev/md0
Physical volume "/dev/md0" successfully created
```

4. Creamos un grupo de volúmenes llamado; grupvol

```
vgcreate grupvol /dev/md0
```

```
root@debian:~# vgcreate grupvol1 /dev/md0
/proc/devices: No entry for device-mapper found
Volume group "grupvol1" successfully created
```

5. Creamos ahora 5 volúmenes lógicos de 2 GB cada uno en el disco donde tenemos el grupo de volúmenes

```
lvcreate -L 2000M -n lv1 grupvol
lvcreate -L 2000M -n lv2 grupvol
lvcreate -L 2000M -n lv3 grupvol
lvcreate -L 2000M -n lv4 grupvol
lvcreate -L 2000M -n lv5 grupvol
```

```
root@debian:~# lvcreate -L 2000M -n lv1 grupvol
Logical volume "lv1" created
root@debian:~# lvcreate -L 2000M -n lv2 grupvol
Logical volume "lv2" created
root@debian:~# lvcreate -L 2000M -n lv3 grupvol
Logical volume "lv3" created
root@debian:~# lvcreate -L 2000M -n lv4 grupvol
Logical volume "lv4" created
root@debian:~# lvcreate -L 2000M -n lv5 grupvol
Logical volume "lv5" created
```

6. Vemos los volúmenes lógicos que hemos creado

```
lvs
```

```
root@debian:~# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
lv1 grupvol -wi-a----- 1,95g
lv2 grupvol -wi-a----- 1,95g
lv3 grupvol -wi-a----- 1,95g
lv4 grupvol -wi-a----- 1,95g
lv5 grupvol -wi-a----- 1,95g
```

Hasta este punto tendríamos preparado en el Servidor iSCSI la configuración de red, el RAID1 para la alta disponibilidad de los datos y volúmenes en el RAID para crear los IQN.

Vamos a proceder ahora con la configuración de los IQN.

7. Vamos a instalar ahora la herramienta de configuración de iSCSI open-iscsi.

```
apt-get install open-iscsi
```


8. Vamos a editar el fichero iscsid.conf

En este fichero tenemos que descomentar algunas líneas para permitir que el servicio iSCSI pueda reconocer las credenciales de los IQN que llevan usuario y secreto (clave).

```
nano /etc/iscsi/iscsid.conf
```

```
node.session.auth.authmethod = CHAP  
  
node.session.auth.username = username  
node.session.auth.password = password
```

```
#node.session.auth.authmethod = CHAP  
#node.session.auth.username = username  
#node.session.auth.password = password
```

```
node.session.auth.authmethod = CHAP  
node.session.auth.username = username  
node.session.auth.password = password
```

9. Instalamos el software tgt.

```
apt-get install tgt
```

10. Editamos el fichero target.conf, en este fichero se van a agregar la configuración de los IQN que queremos crear.
Eliminamos todo el contenido que pudiese traer el fichero.

```
nano /etc/tgt/targets.conf
```

Vamos a ver los IQN que vamos a configurar en el fichero targets.conf

Los IQN como vimos anteriormente se crean con una nomenclatura descriptiva.

En este caso crearemos 4 IQN.

```
iqn.2017-09.org.fran.tg1
```

```
iqn.2017-09.org.fran.tg2
```

```
iqn.2017-09.org.fran.tg3
```

```
iqn.2017-09.org.fran.tg4
```

El primer IQN se crea usando uno de los volúmenes de 2 GB. Este estará configurado de forma que cuando llegue al iniciador (cliente) no haga falta contraseña para añadirlo a la máquina.

El segundo IQN también se creará usando un volumen de 2 GB. La diferencia con el primer IQN será que este para añadirlo al cliente si va a necesitar una contraseña.

El tercer IQN constará de dos volúmenes, de forma que el cliente cuando reciba el target verá como dos discos en su máquina. Este se podrá añadir sin necesidad de contraseña.

El cuarto IQN al igual que el primero se creará usando uno de los volúmenes de 2 GB. Este estará configurado de forma que cuando llegue al iniciador (cliente) tampoco haga falta contraseña para añadirlo a la máquina.

A continuación vemos la configuración que tenemos que poner en el fichero para conseguir lo que hemos descrito antes.

```
# Configura un volumen sin que los usuarios finales tengan que
autenticarse

    <target iqn.2017-09.org.fran.tg1>
        backing-store /dev/grupvol/lv1
    </target>

# Configura un volumen en el que los usuarios si se van a tener que
autenticar para usarlo

    <target iqn.2017-09.org.fran.tg2>
        backing-store /dev/grupvol/lv2
        incominguser usuario Franhidalgo86
    </target>

# Configura dos grupos de volúmenes para servir dos discos a los
clientes, para hacer un disco reflejado por ejemplo

    <target iqn.2017-09.org.fran.tg3>
        backing-store /dev/grupvol/lv3
        backing-store /dev/grupvol/lv4
    </target>

# Configura un volumen sin que los usuarios finales tengan que
autenticarse

    <target iqn.2017-09.org.fran.tg4>
        backing-store /dev/grupvol/lv5
    </target>
```

11. Una vez configurado los IQN (puede ser uno o los que queramos) vamos a reiniciar el servicio tgt

```
/etc/init.d/tgt reload
```

12. Vamos a ejecutar el siguiente comando para ver los targets (la ip es la del servidor iSCSI)

```
iscsiadm --mode discovery --type st --portal 10.0.10.2
```

```
root@debian:~# iscsiadm --mode discovery --type st --portal 10.0.10.2
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg1
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg2
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg3
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg4
root@debian:~#
```

13. Una vez se termine la configuración, vamos a desactivar la salida a internet, para dejar solo en el servidor la red (10.0.10.0) que es por la que se va a servir los target iSCSI.

Para ello lo primero es comentar las líneas referentes a la interfaz que queremos quitar, del fichero de configuración de red.

```
nano /etc/network/interfaces
```

```
# The primary network interface
#auto eth0
#iface eth0 inet dhcp
```

- 13.1. Seguidamente quitamos el “cable” o el adaptador virtual.
13.2. Una vez hecho esto reiniciamos el servicio networking.

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

15.3. CLIENTES iSCSI Linux (4 y 5)

Ahora vamos a configurar los clientes iSCSI Linux, para ello vamos a seguir los siguientes pasos.

1. Configuración de interfaces de red (vemos el fichero completo Cliente Linux1)

```
nano /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 interface
auto eth0
iface eth0 inet dhcp

#ip interna
auto eth1
iface eth1 inet static
address 10.0.10.3
netmask 255.255.255.0
network 10.0.10.0
```

2. Le vamos a cambiar el nombre a la maquina (Cliente Linux1)

```
nano /etc/hostname
```

```
debian1
```

3. Configuración de interfaces de red (vemos el fichero completo Cliente Linux2)

```
nano /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 interface
auto eth0
iface eth0 inet dhcp

#ip interna
auto eth1
iface eth1 inet static
address 10.0.10.4
netmask 255.255.255.0
network 10.0.10.0
```

4. Le vamos a cambiar el nombre a la maquina (Cliente Linux2)

```
nano /etc/hostname
```

```
debian2
```

5. Instalamos el software que detectara los targets del servidor (instalación tanto en el Cliente Linux 1 como en el 2).

```
apt-get install open-iscsi
```

6. Instalaremos también en los dos clientes Linux el software apache2 (instalación tanto en el Cliente Linux 1 como en el 2)

```
apt-get install apache2
```

7. Instalaremos también el software para crear sistemas de fichero distribuido ocfs2 (instalación tanto en el Cliente Linux 1 como en el 2)

```
apt-get install ocfs2-tools
```

8. Una vez instalados los software, ejecutamos el siguiente comando, que buscara en la red que le indiquemos los IQN servidos por el servidor iSCSI. (Descubrimos los IQN tanto en el Cliente Linux 1 como en el 2)

```
iscsiadm --mode discovery --type sendtargets --portal 10.0.10.2
```

```
root@debian1:~# iscsiadm --mode discovery --type sendtargets --portal 10.0.10.2
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg1
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg2
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg3
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg4
```

```
root@debian2:~# iscsiadm --mode discovery --type sendtargets --portal 10.0.10.2
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg1
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg2
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg3
10.0.10.2:3260,1 iqn.2017-09.org.fran.tg4
```

15.3.1. Implantación de una página web estática en uno de los clientes y replicación en el otro cliente formando un sistema de cluster con el software OCFS2

1. Vemos en primer lugar los discos con los que cuentan las maquinas clientes. (Se hace en los dos clientes Linux)

```
lsblk
```

```
root@debian1:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0   12G  0 disk
├─sda1        8:1    0 11,5G  0 part /
├─sda2        8:2    0    1K  0 part
└─sda5        8:5    0   549M  0 part [SWAP]
sr0          11:0    1 1024M  0 rom
```

```
root@debian2:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0   12G  0 disk
├─sda1        8:1    0 11,5G  0 part /
├─sda2        8:2    0    1K  0 part
└─sda5        8:5    0   549M  0 part [SWAP]
sr0          11:0    1 1024M  0 rom
```

2. Ahora que hemos visto los IQN que sirve el servidor, vamos a proceder a conectarnos a uno de ellos. En este caso vamos a elegir el tg1 por ejemplo. Para ello lo vamos a hacer de la siguiente forma. Esto hará que nos prepare una conexión automática a este IQN nada más se reinicie el servicio. (Se hace en los dos clientes Linux)

```
iscsiadm -m node --targetname iqn.2017-09.org.fran.tg1 -p 10.0.10.2 -o
update -n node.conn[0].startup -v automatic
```

3. Reiniciamos el servicio para que se conecte el IQN. (Se hace en los dos clientes Linux)

```
/etc/init.d/open-iscsi restart
```

Ahora tendríamos el volumen lv1 del servidor iSCSI como si fuese un disco físico en los dos clientes iSCSI Linux.

4. Para comprobar que verdaderamente contamos con otro disco en las máquinas cliente iSCSI Linux, vamos a ejecutar el siguiente comando. (Se hace en los dos clientes Linux)

```
lsblk
```

```
root@debian1:~# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0  12G  0 disk
├─sda1 8:1    0 11,5G  0 part /
├─sda2 8:2    0    1K  0 part
└─sda5 8:5    0  549M  0 part [SWAP]
sdb   8:16   0    2G  0 disk
sr0   11:0   1 1024M  0 rom
```

```
root@debian2:~# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0    0  12G  0 disk
├─sda1 8:1    0 11,5G  0 part /
├─sda2 8:2    0    1K  0 part
└─sda5 8:5    0  549M  0 part [SWAP]
sdb   8:16   0    2G  0 disk
sr0   11:0   1 1024M  0 rom
```

Una vez hemos conectado el INQ a las máquinas y hemos comprobado que contamos con un nuevo disco, podemos hacer particiones en ese nuevo disco, dar formato... o hacer cualquier cosa que se puede hacer con cualquier disco conectado de forma física en la máquina.

5. Vamos a dar formato al nuevo disco. Para ello con el comando “fdisk” vemos los discos y las particiones que tienen. Como el nuevo disco está recién conectado, obviamente no va a tener ninguna partición. (Se hace en los dos clientes Linux)

```
fdisk -l
```

- 5.1. Ahora vamos a crearle una nueva partición. (solo Cliente Linux 1)

```
fdisk /dev/sdb
```

```
n
...INTRO
...INTRO
...INTRO
w
```


- 5.2. Ahora vamos a hacer lo mismo en el Cliente Linux 2, a este ya le saldrá que tiene una partición hecha, ya que estamos usando el mismo disco, pero de todas formas le repetimos el proceso de antes. (solo Cliente Linux 2)

```
n
...INTRO
...INTRO
w
```

6. Vamos a crear y editar ahora un fichero donde especificaremos las máquinas que van a formar parte del cluster que vamos a montar para poner nuestra web. (Se hace en los dos clientes Linux)

```
nano /etc/ocfs2/cluster.conf
```

Abajo vemos la configuración que le vamos a dar al fichero.

La configuración que aparece en este fichero indica:

En el apartado “cluster” el número de máquinas que lo van a formar y el nombre de dicho cluster

Seguidamente pondremos un apartado “node” por cada máquina que queramos añadir al cluster.

En este caso solo tendremos dos (Cliente Linux1 y Cliente Linux2) y que como nombre tienen los que le dimos al principio de la configuración (debian1 y debian2)

```
cluster:
    node_count = 2
    name = CentralCluster

node:
    number = 1
    cluster = CentralCluster
    ip_port = 7777
    ip_address = 10.0.10.3
    name = debian1

node:
    number = 2
    cluster = CentralCluster
    ip_port = 7777
    ip_address = 10.0.10.4
    name = debian2
```

7. Una vez se ha creado el fichero que configura el cluster de máquinas, hay que hacer una reconfiguración a mano del software ocfs2-tools.

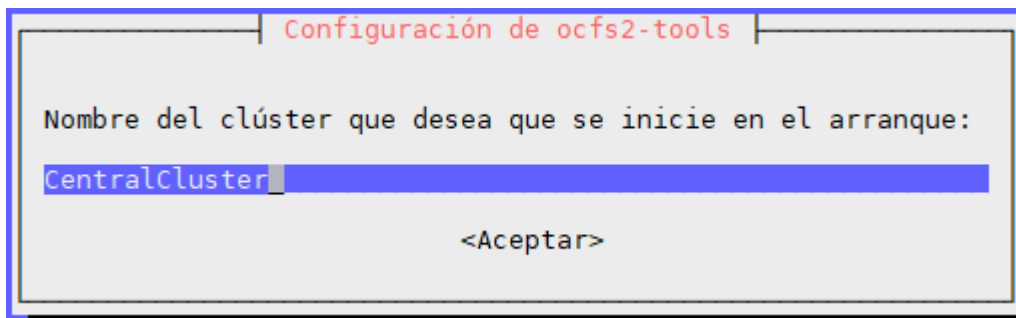
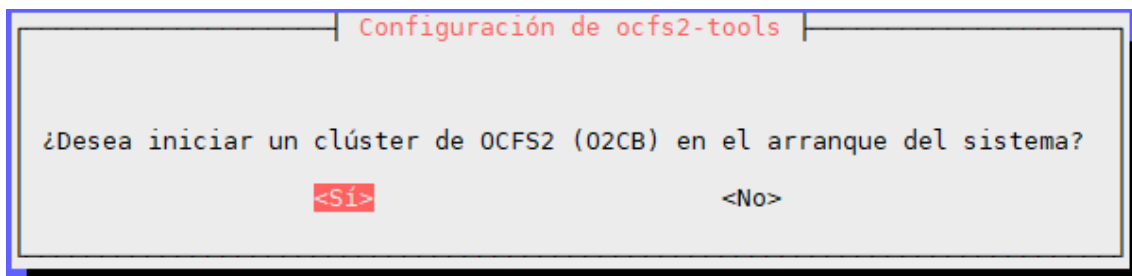
Durante la reconfiguración nos va a pedir una serie de parámetros.

En el primero que nos preguntara le diremos que “si” queremos que nos inicie el cluster.

Después le cambiaremos el nombre por defecto del cluster por el nombre que le dimos en el fichero de configuración (en este caso “CentralCluster”)

Los siguientes parámetros los dejaremos tal como vienen. (Se hace en los dos clientes Linux)

```
dpkg-reconfigure ocfs2-tools
```



8. Ahora vamos a hacerle un restart al servicio ocfs2 (Se hace en los dos clientes Linux)

```
systemctl restart ocfs2.service
```

9. Una vez hemos configurado todo esto, vamos a darle un sistema de ficheros. El sistema de ficheros que le vamos a dar será el OCFS2. (solo Cliente Linux 1)

```
mkfs.ocfs2 -N 8 -L "CentralCluster" /dev/sdb1
```

```
root@debian1:~# mkfs.ocfs2 -N 8 -L "CentralCluster" /dev/sdb1
mkfs.ocfs2 1.6.4
Cluster stack: classic o2cb
Label: CentralCluster
Features: sparse backup-super unwritten inline-data strict-journal-super xattr
Block size: 4096 (12 bits)
Cluster size: 4096 (12 bits)
Volume size: 2096103424 (511744 clusters) (511744 blocks)
Cluster groups: 16 (tail covers 27904 clusters, rest cover 32256 clusters)
Extent allocator size: 4194304 (1 groups)
Journal size: 67108864
Node slots: 8
Creating bitmaps: done
Initializing superblock: done
Writing system files: done
Writing superblock: done
Writing backup superblock: 1 block(s)
Formatting Journals: done
Growing extent allocator: done
Formatting slot map: done
Formatting quota files: done
Writing lost+found: done
mkfs.ocfs2 successful
root@debian1:~# █
```

10. Vamos ahora a reiniciar los dos nodos del cluster. (Se hace en los dos clientes Linux)

```
reboot
```

11. Una vez se inician de nuevo las dos máquinas tenemos que proceder con el reinicio de los servicios "open-iscsi" y "ocfs2-tools" (Se hace en los dos clientes Linux)

```
/etc/init.d/open-iscsi restart
systemctl restart ocfs2.service
```

12. Vamos a montar la partición en ese directorio. (Se hace en los dos clientes Linux)

```
mount /dev/sdb1 /var/www/html
```

```

root@debian1:~# mount /dev/sdb1 /var/www/html
root@debian1:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   12G  0 disk
├─sda1       8:1    0  11,5G  0 part /
├─sda2       8:2    0    1K  0 part
└─sda5       8:5    0   549M  0 part [SWAP]
sdb          8:16   0    2G  0 disk
└─sdb1       8:17   0    2G  0 part /var/www/html
sr0         11:0    1 1024M  0 rom
root@debian1:~# █

```

```

root@debian2:~# mount /dev/sdb1 /var/www/html
root@debian2:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   12G  0 disk
├─sda1       8:1    0  11,5G  0 part /
├─sda2       8:2    0    1K  0 part
└─sda5       8:5    0   549M  0 part [SWAP]
sdb          8:16   0    2G  0 disk
└─sdb1       8:17   0    2G  0 part /var/www/html
sr0         11:0    1 1024M  0 rom
root@debian2:~# █

```

13. Pasamos los ficheros que forman nuestra página web al directorio /var/www/html (solo Cliente Linux 1). Si hacemos un “ls” tanto en el Cliente Linux1 como en el Cliente Linux2 vamos a ver que los fichero nos los a duplicado de una máquina a la otra.

```

root@debian1:~# ls -l /var/www/html/
total 44
drwxr-xr-x 8 usuario usuario 3896 dic 18 20:00 contacto
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 css
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 documentos
drwxr-xr-x 6 usuario usuario 3896 dic 18 20:00 font-awesome
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 fonts
drwxr-xr-x 4 usuario usuario 3896 dic 18 20:00 img
-rw-r--r-- 1 usuario usuario 42892 dic 18 20:00 index.html
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 js
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 less
drwxr-xr-x 2 root root 3896 dic 18 19:53 lost+found
root@debian1:~# █

```

```

root@debian2:~# ls -l /var/www/html/
total 44
drwxr-xr-x 8 usuario usuario 3896 dic 18 20:00 contacto
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 css
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 documentos
drwxr-xr-x 6 usuario usuario 3896 dic 18 20:00 font-awesome
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 fonts
drwxr-xr-x 4 usuario usuario 3896 dic 18 20:00 img
-rw-r--r-- 1 usuario usuario 42892 dic 18 20:00 index.html
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 js
drwxr-xr-x 2 usuario usuario 3896 dic 18 20:00 less
drwxr-xr-x 2 root root 3896 dic 18 19:53 lost+found
root@debian2:~# █

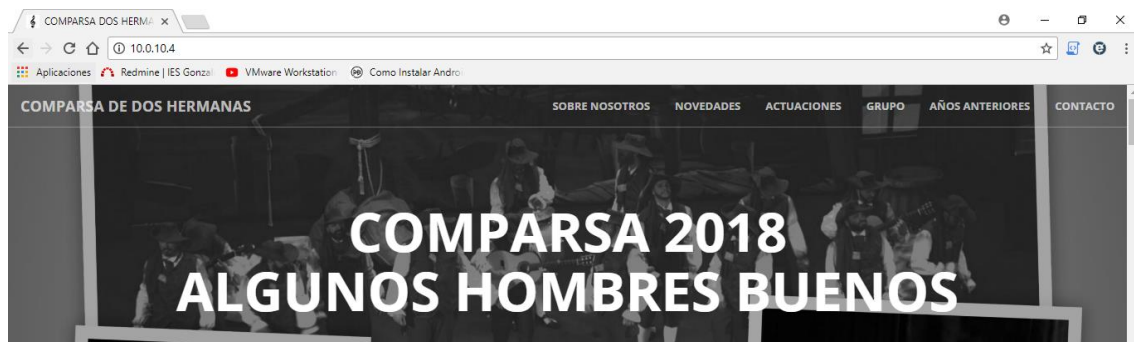
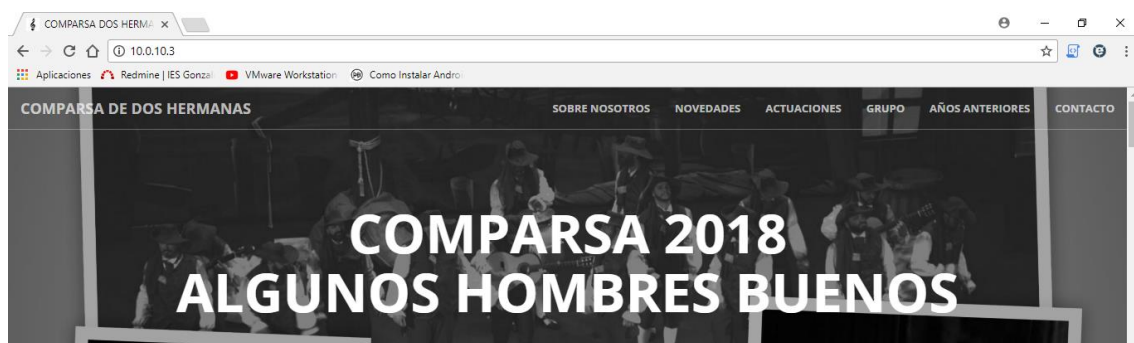
```

14. Cambiamos los permisos de ese directorio. (solo Cliente Linux 1).

Al estar sincronizados los directorios /var/www/html también se cambiarán los permisos en el nodo Cliente Linux2.

```
chown -R www-data:www-data /var/www/html
```

15. Accedemos desde un navegador a la página a través de la IP de Cliente Linux1 y después a través de la IP de Cliente Linux2, comprobamos que aunque la página solo la hemos colocado en una de las máquinas podemos acceder gracias al sistema de ficheros distribuido OCFS2 que nos replica la información en la otra máquina.



15.4. PRUEBAS RAID

Ya que tenemos configurado todo el escenario vamos a proceder a probar la alta disponibilidad de los datos que tenemos en los clientes iSCSI.

1. Vamos a producir un error mediante software de uno de los dispositivos que formen el RAID, para ello introduciremos el siguiente comando:

- 1.1. Forzado de fallo en un disco del raid

```
mdadm --fail /dev/md0 /dev/sdb
```

- 1.2. A continuación realizaremos el cat del archivo y podremos ver que efectivamente, el disco tiene un fallo:

```
cat /proc/mdstat
```

```
root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc[1] sdb[0](F)
      10477568 blocks super 1.2 [2/1] [_U]

unused devices: <none>
root@debian:~#
```

2. Ahora vamos a reconstruir el RAID ya que uno de los disco tiene un fallo, vamos a conectar un disco nuevo. Podemos hacerlo en caliente (con la maquina arrancada).

```
root@debian:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                 8:0      0   12G  0 disk
├─sda1                             8:1      0  11,5G  0 part  /
├─sda2                             8:2      0    1K  0 part
└─sda5                             8:5      0  549M  0 part  [SWAP]
sdb                                 8:16     0   10G  0 disk
├─md0                              9:0      0   10G  0 raid1
│   ├─grupvol-lv1                 253:0     0    2G  0 lvm
│   ├─grupvol-lv2                 253:1     0    2G  0 lvm
│   ├─grupvol-lv3                 253:2     0    2G  0 lvm
│   ├─grupvol-lv4                 253:3     0    2G  0 lvm
│   └─grupvol-lv5                 253:4     0    2G  0 lvm
sdc                                 8:32     0   10G  0 disk
├─md0                              9:0      0   10G  0 raid1
│   ├─grupvol-lv1                 253:0     0    2G  0 lvm
│   ├─grupvol-lv2                 253:1     0    2G  0 lvm
│   ├─grupvol-lv3                 253:2     0    2G  0 lvm
│   ├─grupvol-lv4                 253:3     0    2G  0 lvm
│   └─grupvol-lv5                 253:4     0    2G  0 lvm
sdd                                 8:48     0   10G  0 disk
sr0                                11:0     1 1024M  0 rom
```

3. Ahora desconectamos del disco con fallos del RAID.

```
mdadm --remove /dev/md0 /dev/sdb
```

4. Añadimos el nuevo disco al RAID 1

```
mdadm --add /dev/md0 /dev/sdd
```

5. Ejecutaremos de nuevo el cat para ver el estado del RAID, viendo que efectivamente el disco ha sido añadido:

```
cat /proc/mdstat
```

```
root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdd[2] sdc[1]
      10477568 blocks super 1.2 [2/2] [UU]

unused devices: <none>
root@debian:~#
```

6. En este punto tendremos reconstruido el RAID con los discos (sdc) y el nuevo (sdd):

Vemos información sobre la sincronización de los discos el RAID1

```
mdadm --query --detail /dev/md0
```

```
root@debian:~# mdadm --query --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Tue Dec 19 11:54:35 2017
  Raid Level : raid1
  Array Size : 10477568 (9.99 GiB 10.73 GB)
  Used Dev Size : 10477568 (9.99 GiB 10.73 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Tue Dec 19 12:33:24 2017
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Name : debian:0 (local to host debian)
  UUID : 80f7f407:d3431843:3f07f09f:1cb623eb
  Events : 71

  Number Major Minor RaidDevice State
    2      8     48        0  active sync  /dev/sdd
    1      8     32        1  active sync  /dev/sdc
```

7. Vemos el estado de los discos.

```
root@debian:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda                                 8:0      0   12G  0 disk
├─sda1                             8:1      0  11,5G  0 part  /
├─sda2                             8:2      0    1K  0 part
└─sda5                             8:5      0   549M  0 part  [SWAP]
sdb                                 8:16     0   10G  0 disk
sdc                                 8:32     0   10G  0 disk
├─md0                              9:0      0   10G  0 raid1
│   └─grupvol-lv1 253:0      0    2G  0 lvm
│       └─grupvol-lv2 253:1      0    2G  0 lvm
│           └─grupvol-lv3 253:2      0    2G  0 lvm
│               └─grupvol-lv4 253:3      0    2G  0 lvm
│                   └─grupvol-lv5 253:4      0    2G  0 lvm
└─sdd                              8:48     0   10G  0 disk
    ├─md0                              9:0      0   10G  0 raid1
    │   └─grupvol-lv1 253:0      0    2G  0 lvm
    │       └─grupvol-lv2 253:1      0    2G  0 lvm
    │           └─grupvol-lv3 253:2      0    2G  0 lvm
    │               └─grupvol-lv4 253:3      0    2G  0 lvm
    │                   └─grupvol-lv5 253:4      0    2G  0 lvm
    └─sr0                             11:0     1 1024M  0 rom
```

8. Ya solo queda "sacar" el disco de la máquina. Esto podremos hacerlo en caliente ya que VMWare nos lo permite.

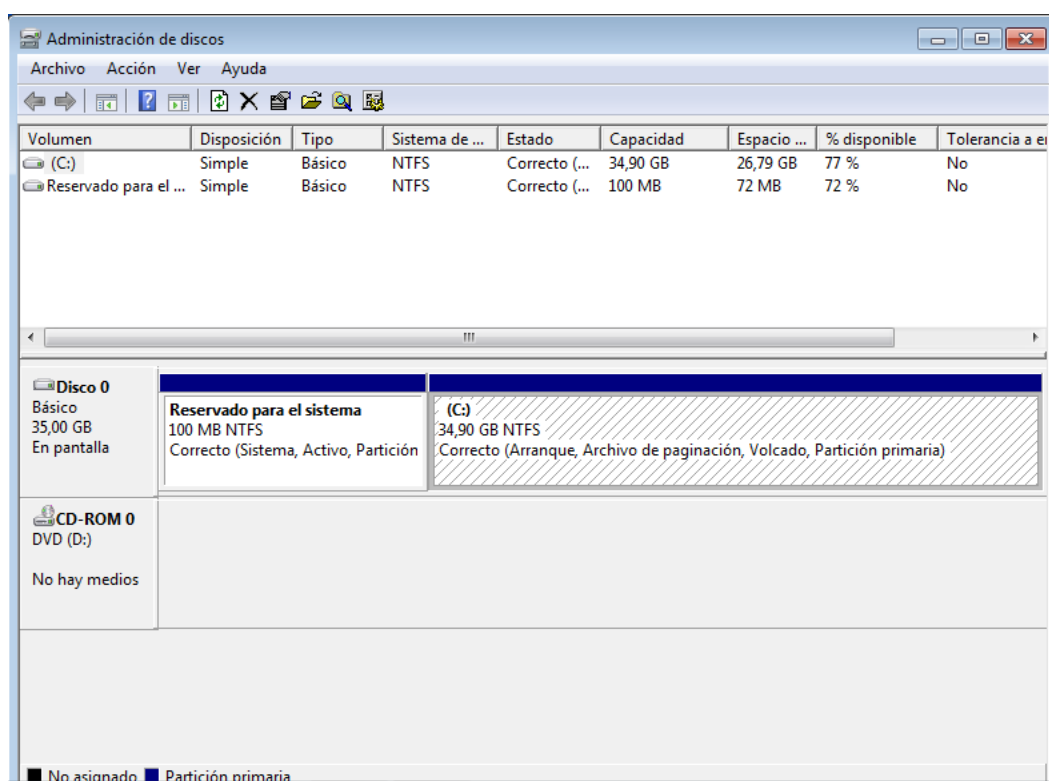
15.5. CLIENTES iSCSI Windows (6 y 7)

Vamos a configurar el servicio iSCSI en un cliente Windows.

Lo primero que haremos será ver los discos con los que contamos.

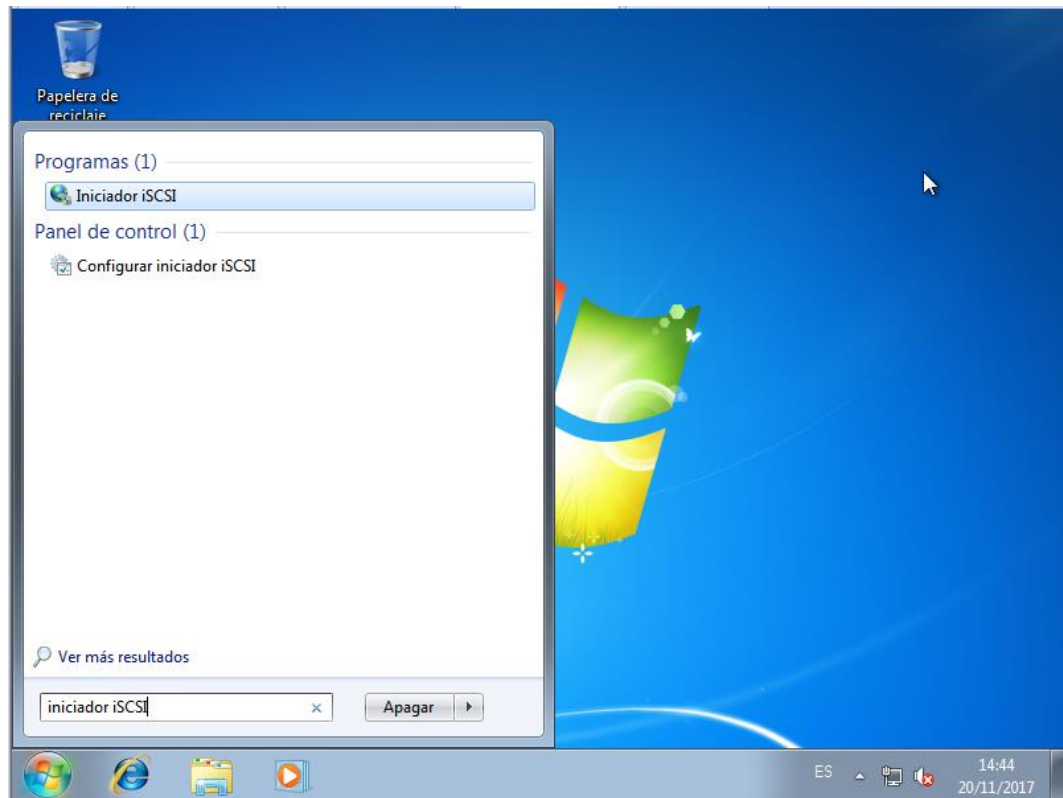
Para ello vamos al “Administrador de disco”. Vemos que solo tenemos un disco.

En principio solo tenemos uno, que es donde está instalado el sistema operativo.



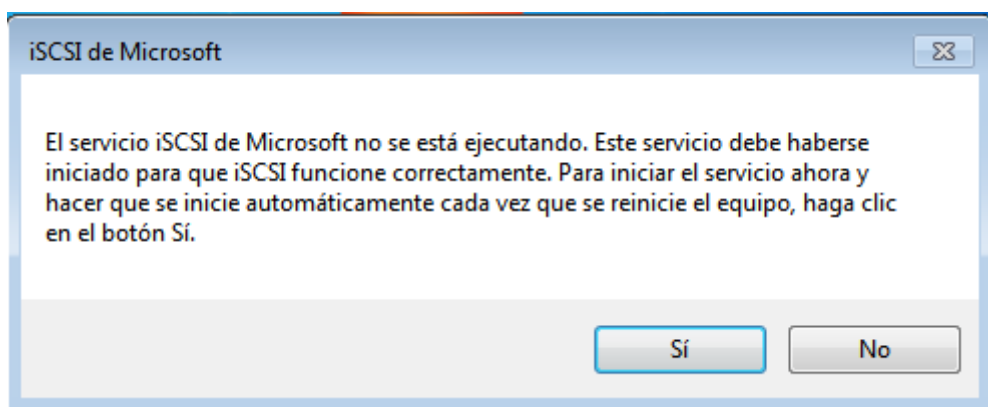
Vamos ahora a configurar el iniciador iSCSI.

Para ello vamos a inicio y buscamos “Iniciador iSCSI”. Una vez localizado le damos clic con el ratón.



Al darle clic nos mostrara un mensaje donde nos avisara que hay que iniciar el servicio iSCSI.

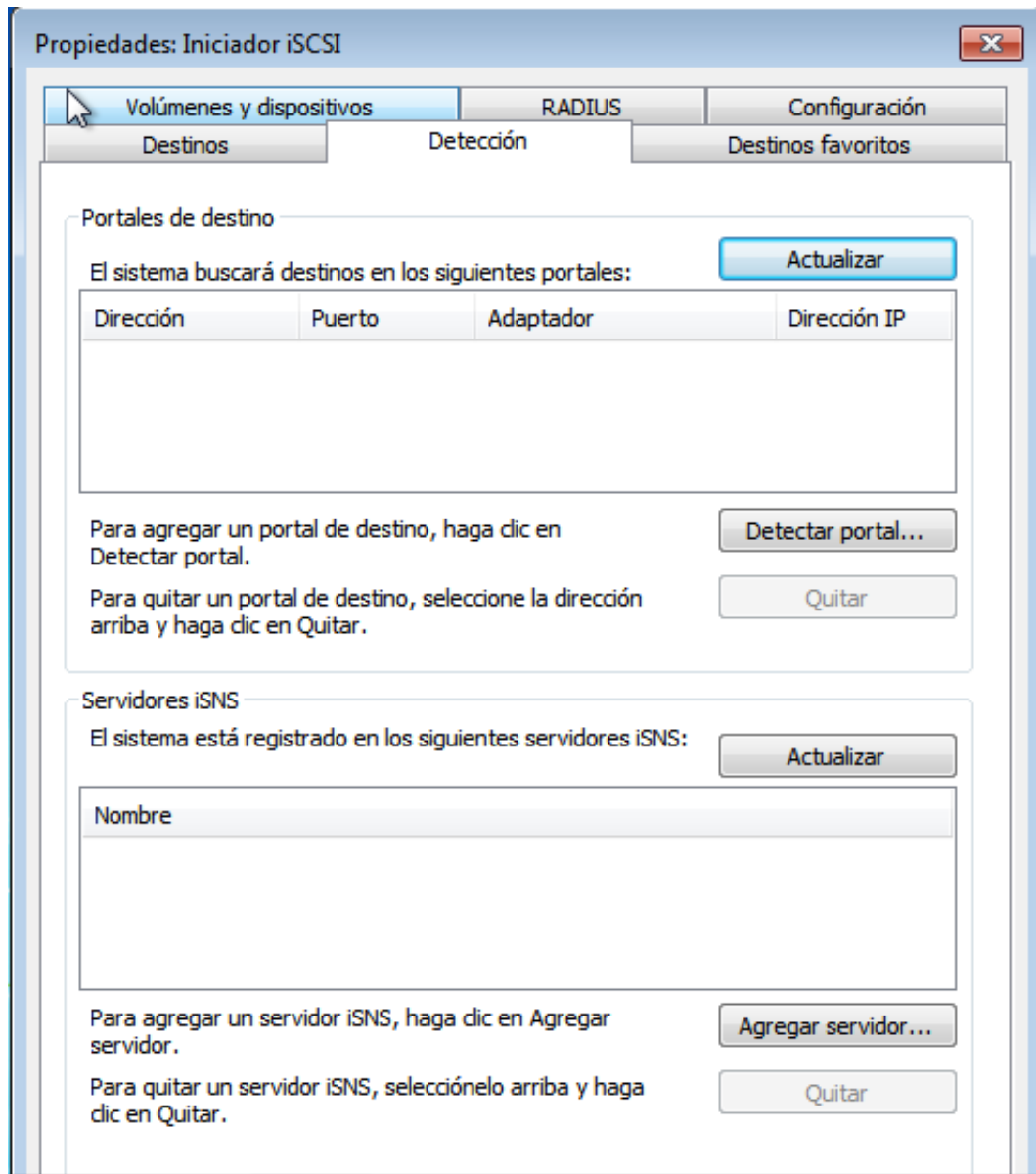
Le damos a “Si”



Seguidamente veremos una ventana de configuración.

Le damos a la pestaña “Detección”.

Una vez abramos esa pestaña, le damos a “Detectar portal...”

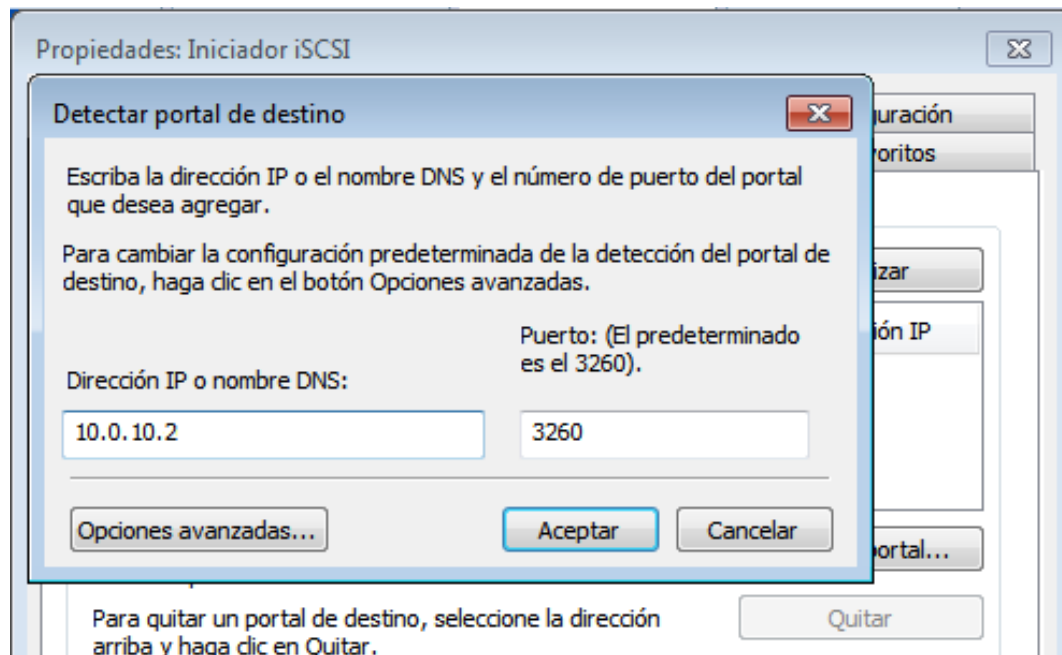


Nos abra otra ventana. Aquí tendremos que poner la IP del servidor iSCSI.

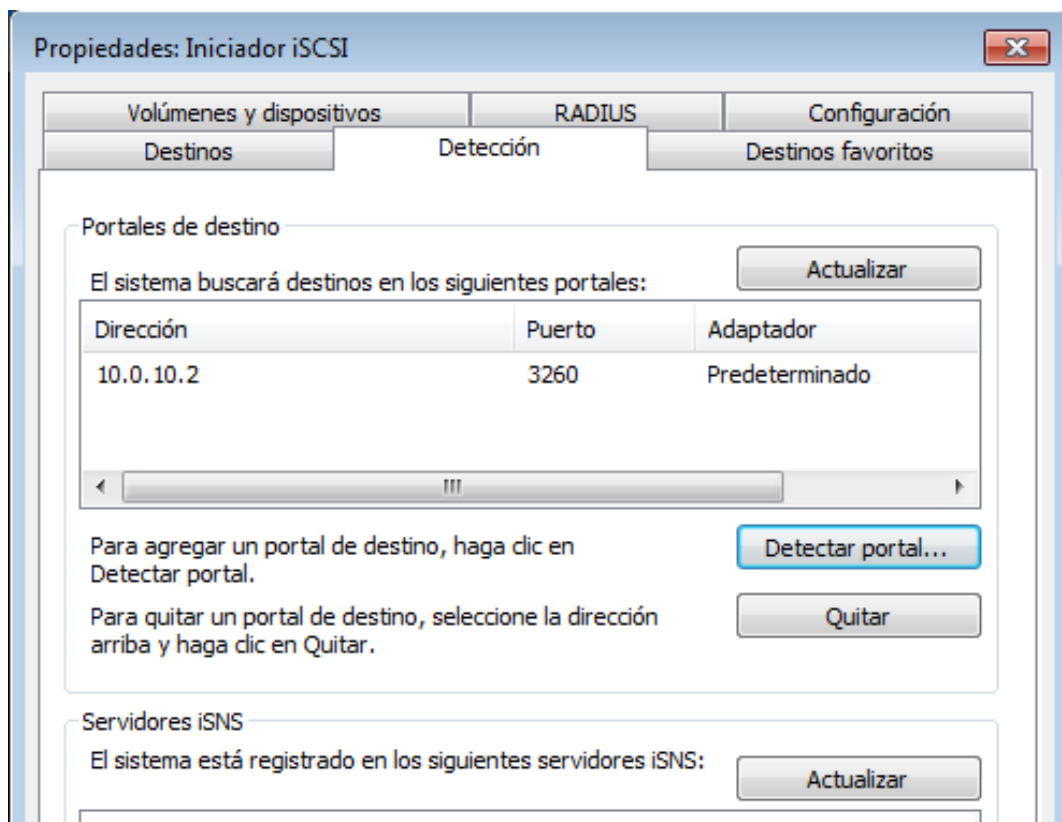
Como vemos en la imagen la IP es la 10.0.10.2

El puerto no lo vamos a tocar. Le vamos a dejar el que viene por defecto, 3260

Una vez configurado esto, le damos a “Aceptar”



Vemos como se ha agregado el servidor iSCSI.

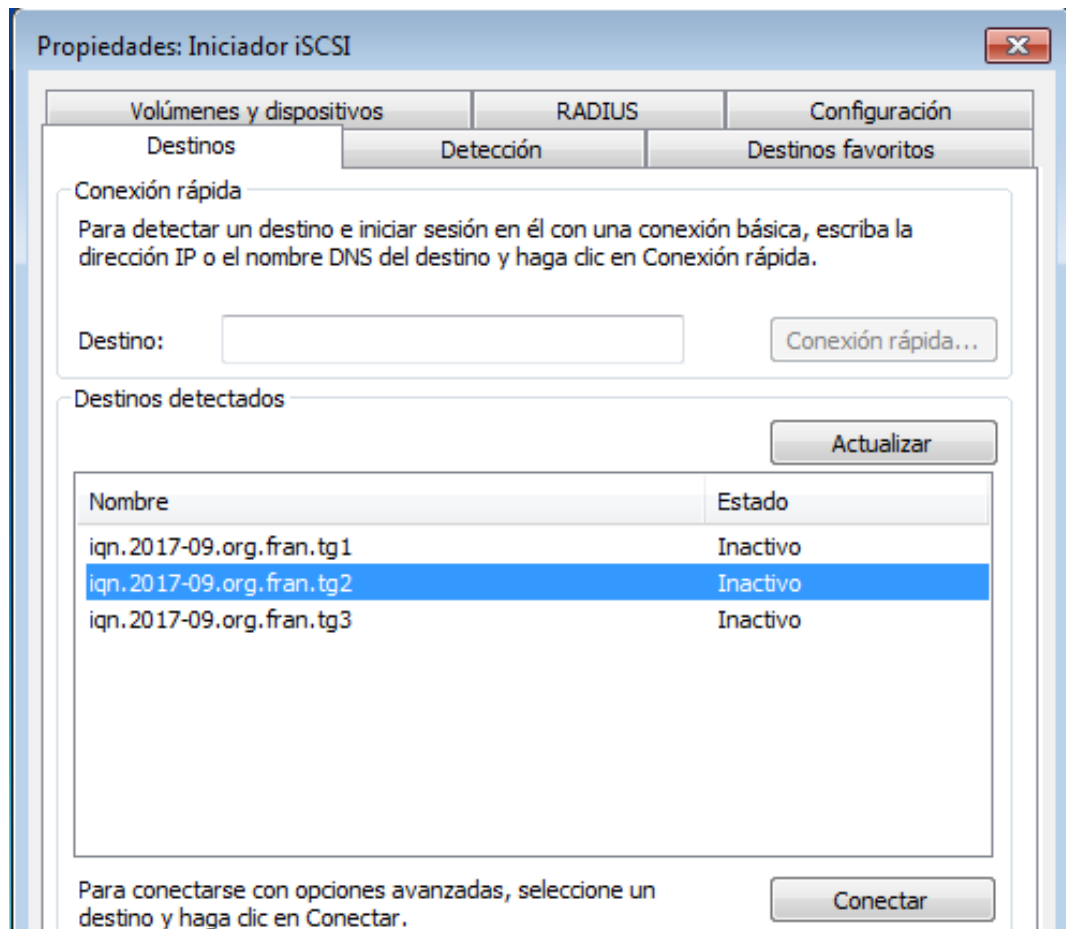


Cambiamos ahora de pestaña para ir a “Destinos” y vemos que aparecen los targets del servidor.

Los tres salen en estado Inactivo.

Vamos a conectar con el targets `iqn.2017-09.org.fran.tg2`

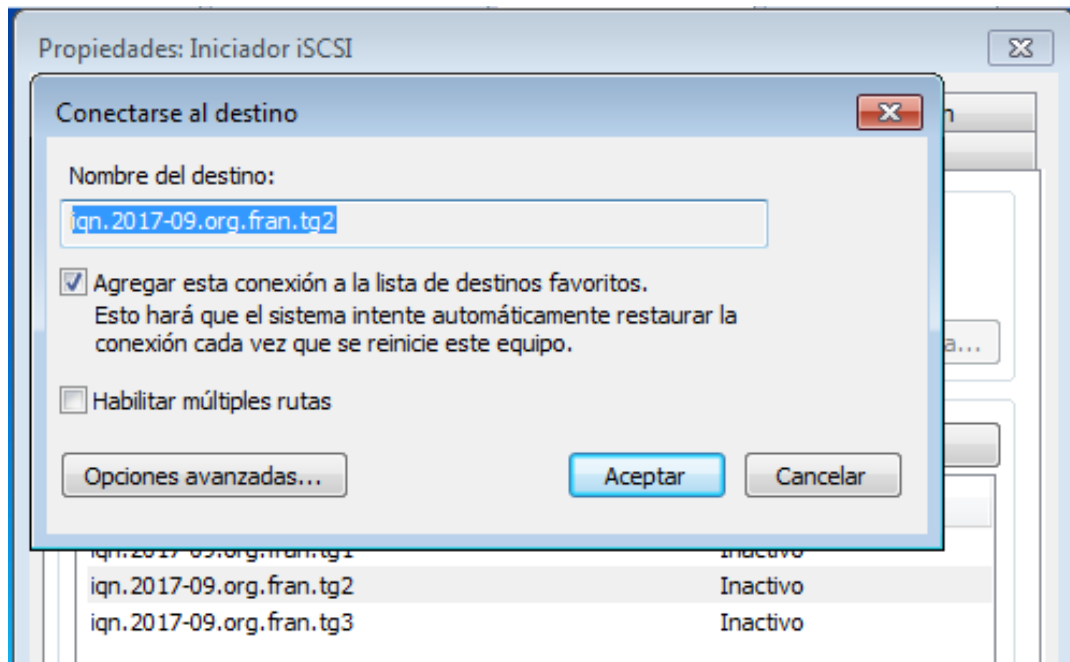
Lo seleccionamos y hacemos clic en “Conectar”



Volveremos a ver la ventana anterior.

Solo bastaría dar a “Aceptar” para conectar con el targets.

Si recordamos, este target tenía unas credenciales para conectarse. Así para conectar con el vamos a dar en “Opciones Avanzadas...”



Nos abrirá una nueva ventana. En esta ventana vamos a configurar las credenciales para conectarnos.

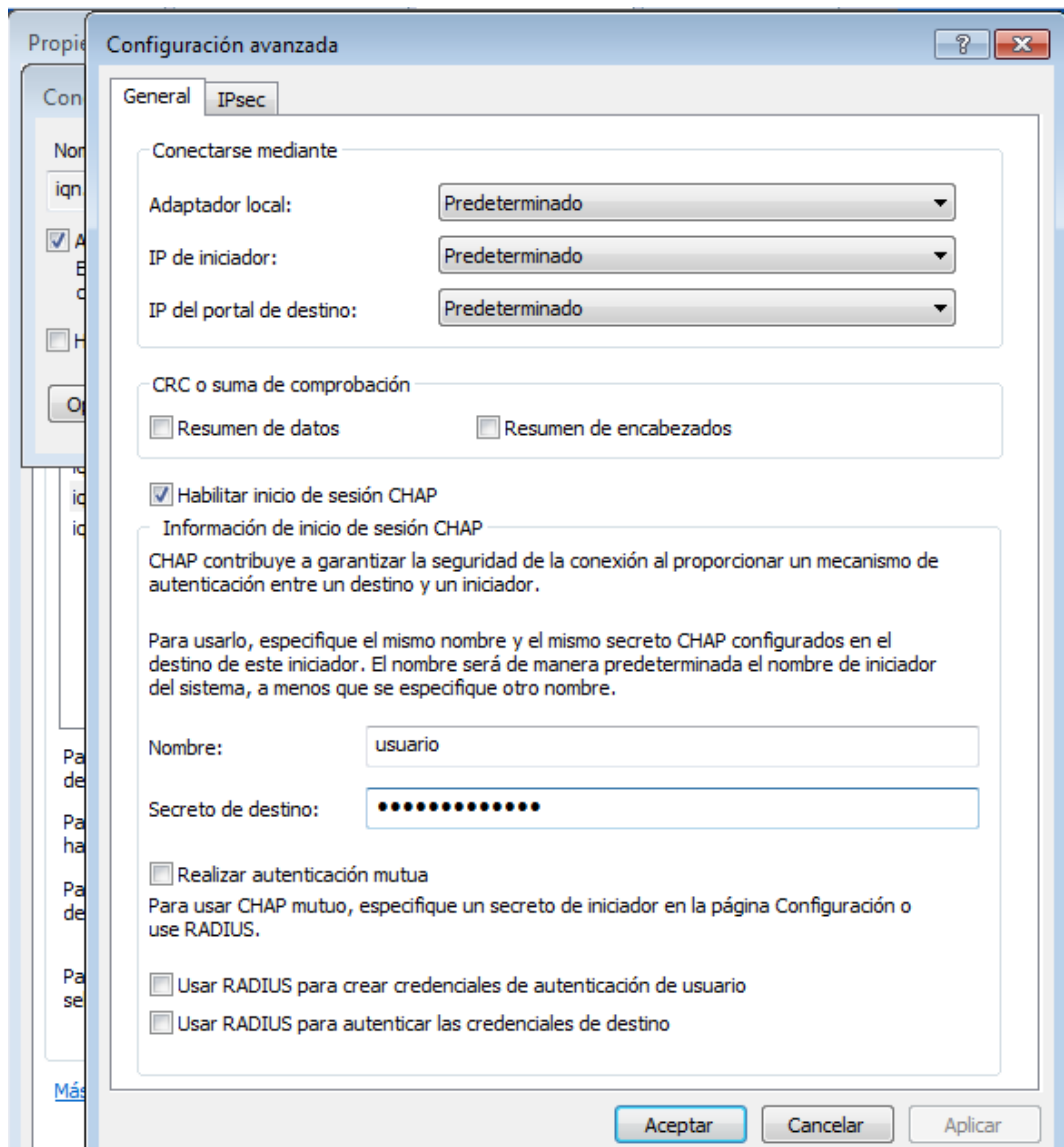
Habilitamos la casilla “Habilitar inicio de sesión CHAP”

En la línea “Nombre” vamos a poner el usuario que configuramos en el target y en la línea de “Secreto de destino” ponemos la contraseña también del target.

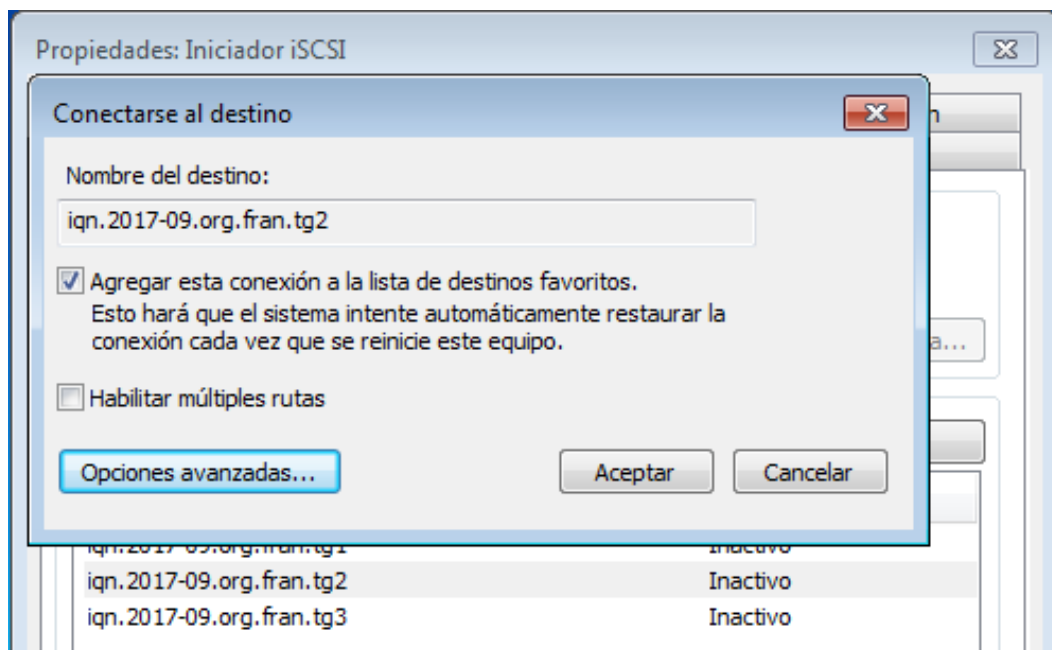
Vamos a recordar las credenciales que pusimos:

```
<target iqn.2017-09.org.fran.tg2>  
    backing-store /dev/grupvol/lv2  
    incominguser usuario Franhidalgo86  
</target>
```

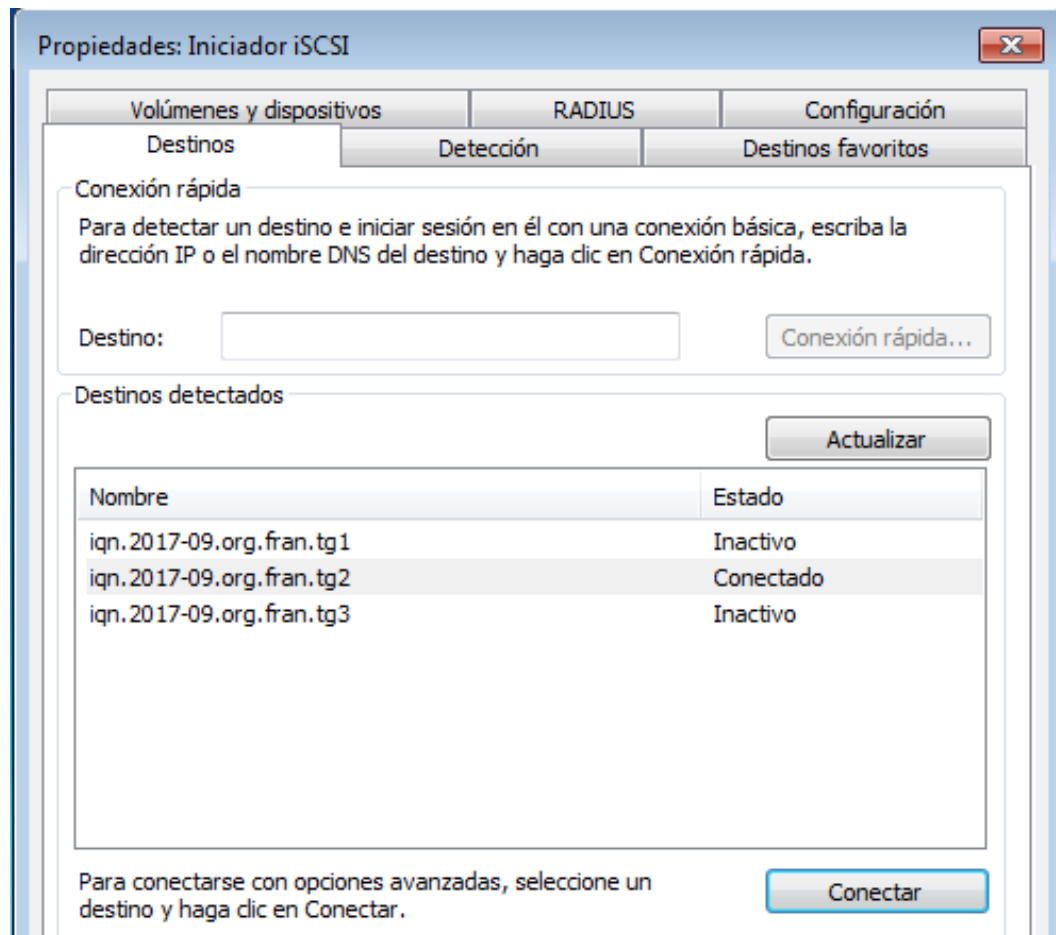
Una vez configurado esto, le damos a “Aceptar”



Saldremos de la ventana de configuracion y le volvemos a dar en “Aceptar”



Podemos ver ahora como el target está en estado “Conectado”.

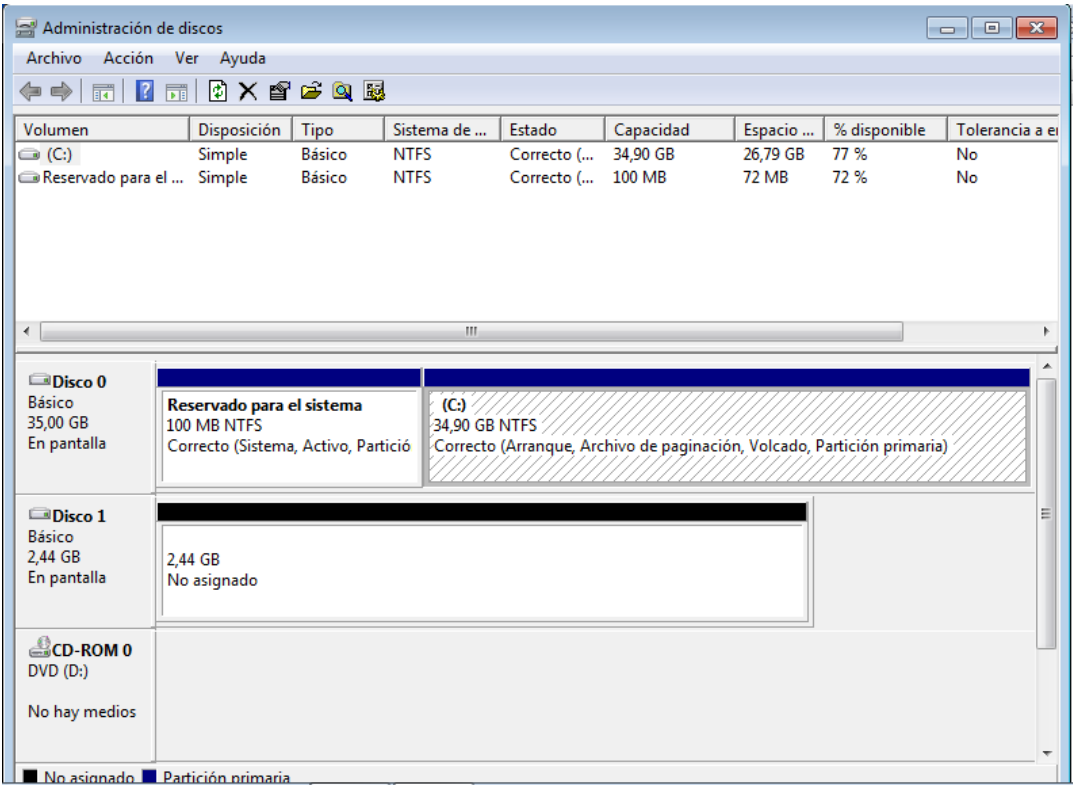


Para probar que está correcto, vamos al “Administrador de disco”

Hay podemos observar que ahora tenemos dos discos.

El disco físico y el que acabamos de pasar por red.

Con el disco que acabamos de pasar por red podemos hacer las mismas gestiones que con un disco físico, particionar, formatear...



16. Infraestructuras y tecnologías usadas

16.1. Infraestructura

Todo se monta sobre un PC portátil Acer que hace la tarea de servidor.

Las características de este PC con las siguientes.

Placa Base:

Soporta virtualización.

Sistema operativo:

Windows 10 Pro de 64bit

Memoria RAM:

8GB en dos módulos de 4GB

Procesador:

Intel Core i3 con 2 procesadores de 1.40 GHz y 4 hilos

Disco duro:

150GB

16.2. La virtualización se realiza con los siguientes software:

El proyecto lo realizamos con las siguientes herramientas d virtualización.

Virtualización:

VMWare Workstation PRO

Cliente de administración de máquinas:

MobaxTerm

16.3. Tecnologías Usadas

Vemos a continuación una lista de las tecnologías que usamos durante la realización de este proyecto.

Servicios:

Servicio DHCP

Servicio apache2

Hardware:

RAID1.

Gestión de discos.

Sistemas:

Instalación de paquetes.

Gestión de Almacenamiento “iSCSI”

Automatización con script “bash”

Sistema de fichero “ocfs2”

Redes:

Configuración router Linux.

Configuración de redes estáticas.

17. FUENTES DE INFORMACIÓN.

17.1. Configuración iSCSI

<https://federicosayd.wordpress.com/2007/09/13/montando-un-iniciador-iscsi-en-linux/>

http://wiki.woop.es/Configuracion_iniciador_iSCSI_CentOS

<https://www.howtoforge.com/using-iscsi-on-debian-squeeze-initiator-and-target>

<https://alejandronieto90.wordpress.com/2014/12/19/instalacion-y-configuracion-de-servidor-y-cliente-iscsi/>

https://www.server-world.info/en/note?os=Debian_9&p=iscsi&f=3

<https://conexiona.com/como-crear-un-destino-iscsi-en-debian/>

17.2. Raid 1

<http://www.linux-party.com/60-software/9247-como-crear-y-configurar-un-raid-1-por-software-con-mdadm-en-linux>

<http://www.linuxuserexpo.com/2015/03/como-administrar-un-raid-1-en-linux-paso-a-paso/>

17.3. OCSF2

<https://administratorlabs.wordpress.com/tag/ocfs2/>

<https://adrianjimenezb.wordpress.com/2015/01/20/ocfs2-iscsi-en-gnulinux/>

17.4. Documentación mobaxterm

<https://aprendiendoavirtualizar.com/mobaxterm-x-server-and-ssh-client/>