

ISCSI

Introduction

iSCSI is a block-storage networking standard that works on TCP/IP layer and allows SCSI commands (CDBs) to be sent over local area networks (LAN) instead of over a local motherboard wire. In other words: it is used to access the storage on your network over TCP/IP using block-level data transfer; the LUNs (disks) presented by the iSCSI target can be entire disks, partitions, LVs or even just plain old files on the filesystem. In order to access the storage over iscsi, we need to have the following components:

iSCSI initiator – This is the iscsi client that can send iscsi commands over TCP/IP network.

iSCSI Target – This is your (remote) storage node

NOTA: There is often a comparison between NFS vs iSCSI. The key difference is NFS is file-level implementation whereas iSCSI is a block-level implementation. In block-level implementations such as iSCSI and Fiber Channel, the storage array offers up a collection of blocks to the client, which then formats that raw storage with whatever file system it decides to use.

Explicació més extensa:

El protocol SCSI original s'utilitza per comunicar dispositius compatibles (com ara discos durs o lectors de DVD de tipus "SCSI") amb la màquina a la qual estan "enxufats" físicament; amb iSCSI, no obstant, les comandes SCSI de control i de dades es transmeten a través de la xarxa. És a dir, iSCSI ens permet tenir un disc dur connectat a través de la xarxa (en comptes de tenir-ho connectat físicament al nostre equip) i poder fer-lo servir exactament igual que com si fos local.

Algú podrà pensar que iSCSI és el mateix que Samba o NFS. No: aquests sistemes importen un sistema d'arxius a través de la xarxa, però iSCSI importa tot el dispositiu hardware a través de la xarxa, de forma que en el client és detectat com un dispositiu SCSI més. Això implica que, a diferència de Samba o NFS, en teoria iSCSI no soporti múltiples connexions a la vegada (és a dir, que dos equips no puguin utilitzar el mateix disc iSCSI per escriure-hi, ja que això seria igual a tenir un disc dur connectat a dues màquines a la vegada: el més probable és que apareguessin inconcistències a les dades). Tot i així, existeixen alternatives per a què iSCSI pugui suportar múltiples usuaris, com ara el sistema de fitxers GFS de RedHat, Ceph, Lustre, etc

El servidor iSCSI (anomenat normalment "target") és l'ordinador que ofereix un o més recursos iSCSI a la xarxa, gràcies a un software específic que ha de tenir funcionant. No cal que el dispositiu a exportar sigui un dispositiu SCSI necessàriament: a més de discos durs sencers (o lectors/gravadors de DVD) que poden ser SATA, també es poden exportar particions concretes o fins i tot arxius-image individuals. Aquest ordinador ha de tenir instal·lar un software específic que s'encarregui d'exportar els recursos especificats.

El servidor ha de tenir un nom determinat (el "wwn serial") que l'identifica davant dels clients. Aquest nom ha de seguir un determinat patró de la forma: iqn.yyyy-mm.nom_de_domini_invertit:identificador on "iqn" és un terme fixe i ha de figurar al principi sempre, "yyyy-mm" és (en teoria) la data d'alta del domini d'Internet de l'organització per a la què estem configurant el target (aquesta data es pot consultar si es vol amb la comanda "whois"), a continuació ha de figurar (en teoria) el nom del domini invertit i finalment, després dels ":" ha d'haver un identificador del nostre gust (que pot donar informació sobre el target). Un exemple vàlid seria: iqn.2005-02.com.empresa:elmeuservidor

El client iSCSI (anomenat normalment "iniciador") és l'ordinador que accedeix a un o més recursos iSCSI a través de la xarxa, gràcies a un software específic que ha de tenir funcionant. Cada client ha de tenir també un wwn reconegut pel target. Aquest nom (i el del propi target) es generaran amb els seus respectius software específics iSCSI: no cal cap infraestructura DNS ni res semblant.

No cal dir que iSCSI és independent del sistema operatiu: podríem tenir un target muntat en Linux i utilitzar els discos d'aquest target en sistemes Windows, o viceversa.

How to implement a iSCSI target

Do the following steps (as root):

```
apt install targetcli-fb          #Its official website is https://github.com/open-iscsi/targetcli-fb  
systemctl start rtslib-fb-targetcli  
targetcli
```

Inside *targetcli* shell you can use the *ls* command to know which sections are available from current one and the *cd /path/to/section* command to move to a specific section (path can be absolute or relative). Depending on current section, available commands will be different. More commands are listed with *help*

The first step to share a block device is to create a "block backstore" (that is, an iSCSI object "tied to" a local real block device on the server -such as a hard drive like /dev/sdc- ; this object is what will be shared to the network in next steps with the name of "LUN") or a "fileio backstore" (that is, an iSCSI object tied to a file on the server -with a predefined size- which, in next steps , will be shared to the network as a "LUN" as if it were a block device). To do this you must do this... :

```
cd /backstores/block  
create pepe /dev/sdc          #En comptes d'un disc podria indicar-se una partició (/dev/sdc1) o un LV!!
```

...or this:

```
cd /backstores/fileio  
create pepe /tmp/fileio 500M #El fitxer es crea en aquest moment amb un tamany dinàmic (és a dir, el fitxer no ocuparà d'entrada els 500M especificats sinò que anirà creixent a mesura que s'ompli de contingut).
```

Next step is create the LUN (that is, share the backstore created in former step). To do this, you must create a "target", which has all the information about which backstore you want to share (the "LUN"), to which initiators, from which interface (if server has more than one), if you want to authenticate users, etc. To do this:

```
cd /iscsi  
create iqn.2018-01.com.example:targetname #El nom ha de ser així: iqn.YYYY-MM.oi.nimod:nombre  
cd /iscsi/iqn.2018..../tpgt1/luns  
create /backstores/fileio/pepe o create /backstores/block/pepe #Comparteixo el block device a la xarxa  
cd /iscsi/iqn.2018..../tpgt1/acls  
create XXX #XXX representa l'identificador de cada màquina client que es vol que pugui accedir al LUN.  
#Aquest identificador es troba dins del fitxer /etc/iscsi/initiatorname.iscsi de cada màquina client.  
cd /iscsi/iqn.2018..../tpgt1/portals #Opcional. Ens ubiquem al lloc adient per aplicar la següent línia  
create 192.168.18.100:3260 #Opcional. Restringeix les escoltes dels clients a la IP i port indicat.  
cd /iscsi/iqn.2018.... #Opcional. Ens ubiquem al lloc adient per aplicar les següents línies  
set auth userid=usuari #Opcional. Crea un usuari reconegut pel target  
set auth password=contrasenya #Opcional. Crea una contrasenya reconeguda pel target  
cd /iscsi/iqn.2018..../tpgt1 #Opcional. Ens ubiquem al lloc adient per aplicar la següent línia  
set attribute authentication=1 #Opcional. Activa l'autenticació d'usuari/contraseña en l'accés al LUN  
saveconfig #Guarda tot el realitzat a /etc/target/saveconfig.json  
exit
```

How to implement a iSCSI initiator

Do once the following steps (as root):

apt install open-iscsi (a Fedora el paquet es diu "iscsi-initiator-utils")
systemctl start iscsid (també hi ha un altre dimoni anomenat "(open-)iscsi" però no el farem servir)

Then, do the following to connect to target server and attach the shared LUNs:

iscsiadm -m discovery -t sendtargets -p 192.168.18.100[:n°port] #Per veure els LUNs remots disponibles
iscsiadm -m node -T iqn.2018-01.com.example:target -l #Per "endollar" els LUNs remots a la màquina local

Un cop enxufats els LUNs, es poden veure com si fossin discos durs locals...:

dmesg #Per veure el seu endollament
lsblk --scsi

...i per tant, es podran particionar, formatejar i muntar com qualsevol altre disc:

fdisk -l
fdisk /dev/sdb
mkfs.xfs /dev/sdb1
mount /dev/sdb1 /mnt
df -h

Per deixar de treballar amb el LUN, caldrà desmuntar-lo primer i després "desendollar-lo"

umount /mnt
iscsiadm -m node -u

More useful commands are:

iscsiadm -m session -o show -P 0 #Per veure l'estat dels LUNs enxufats (-P 1,2,3...per més nivell verbositat)
iscsiadm -m node -o show #Una altra comanda informativa

NOTA: En general, la comanda *iscsiadm* té tres "modes" de funcionament: el mode "discovery" serveix per descobrir targets i agregar-los a la base de dades, el mode "node" serveix per administrar els targets descoberts, conèixer dades sobre ells i connectar-s'hi, i el mode "session" serveix per administrar els targets on s'ha connectat (on s'ha fet "login").

Si el target té activada l'autenticació, al client cal establir-la. Això es fa establint a l'arxiu */etc/iscsi/iscsid.conf* del client les següents línies:

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

D'altra banda, per a què l'endollament (i el muntatge) es faci de forma automàtica, en aquest mateix arxiu *"iscsid.conf"* cal establir la següent línia...:

node.startup = automatic

i editar convenientment l'arxiu */etc/fstab* indicant com opcions de muntatge *"noauto"* i *"x-systemd.automount"* per tal de realitzar el muntatge automàticament només quan la connectivitat de xarxa a la màquina client estigui establerta i no pas abans.