

Using iSCSI On Debian Lenny (Initiator And Target)

By Falko Timme

Published: 2009-03-10 20:05

Using iSCSI On Debian Lenny (Initiator And Target)

Version 1.0

Author: Falko Timme <ft [at] falkotimme [dot] com>

Last edited 02/24/2009

This guide explains how you can set up an iSCSI target and an iSCSI initiator (client), both running Debian Lenny. The iSCSI protocol is a storage area network (SAN) protocol which allows iSCSI initiators to use storage devices on the (remote) iSCSI target using normal ethernet cabling. To the iSCSI initiator, the remote storage looks like a normal, locally-attached hard drive.

I do not issue any guarantee that this will work for you!

1 Preliminary Note

I'm using two Debian Lenny servers here:

- *server1.example.com* (Initiator): IP address *192.168.0.100*

- *server2.example.com* (Target): IP address *192.168.0.101*

2 Setting Up The Target (server2)

[server2:](#)

First we set up the target (*server2*):

```
aptitude install iscsitarget iscsitarget-modules-`uname -r`
```

Open `/etc/default/iscsitarget...`

```
vi /etc/default/iscsitarget
```

... and set `ISCSITARGET_ENABLE` to `true`:

```
ISCSITARGET_ENABLE=true
```

We can use unused logical volumes, image files, hard drives (e.g. `/dev/sdb`), hard drive partitions (e.g. `/dev/sdb1`) or RAID devices (e.g. `/dev/md0`) for the storage. In this example I will create a logical volume of 20GB named `storage_lun1` in the volume group `vg0`:

```
lvcreate -L20G -n storage_lun1 vg0
```

(If you want to use an image file, you can create it as follows:

```
mkdir /storage
```

```
dd if=/dev/zero of=/storage/lun1.img bs=1024k count=20000
```

This creates the image file `/storage/lun1.img` with a size of 20GB.

)

Next we edit `/etc/ietd.conf...`

```
vi /etc/ietd.conf
```

... and comment out everything in that file. At the end we add the following stanza:

```
[...]  
Target iqn.2001-04.com.example:storage.lun1  
    IncomingUser someuser secret  
    OutgoingUser  
    Lun 0 Path=/dev/vg0/storage_lun1,Type=fileio  
    Alias LUN1  
    #MaxConnections 6
```

The target name must be a globally unique name, the iSCSI standard defines the "iSCSI Qualified Name" as follows: *iqn.yyyy-mm.<reversed domain name>[:identifier]*; *yyyy-mm* is the date at which the domain is valid; the identifier is freely selectable. The *IncomingUser* line contains a username and a password so that only the initiators (clients) that provide this username and password can log in and use the storage device; if you don't need authentication, don't specify a username and password in the *IncomingUser* line. In the *Lun* line, we must specify the full path to the storage device (e.g. */dev/vg0/storage_lun1*, */storage/lun1.img*, */dev/sdb*, etc.).

Now we tell the target that we want to allow connections to the device *iqn.2001-04.com.example:storage.lun1* from the IP address *192.168.0.100* (*server1.example.com*)...

```
vi /etc/initiators.allow
```

```
[...]  
iqn.2001-04.com.example:storage.lun1 192.168.0.100
```

... and start the target:

```
/etc/init.d/iscsitarget start
```

3 Setting Up The Initiator (server1)

server1:

On *server1*, we install the initiator:

```
aptitude install open-iscsi
```

Next we open `/etc/iscsi/iscsid.conf...`

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

... and set `node.startup` to `automatic`:

```
[...]  
node.startup = automatic  
[...]
```

Then we restart the initiator:

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

Now we connect to the target (*server2*) and check what storage devices it has to offer:

```
iscsiadm -m discovery -t st -p 192.168.0.101
```

```
server1:~# iscsiadm -m discovery -t st -p 192.168.0.101
```

```
192.168.0.101:3260,1 iqn.2001-04.com.example:storage.lun1
server1:~#
```

```
iscsiadm -m node
```

```
server1:~# iscsiadm -m node
192.168.0.101:3260,1 iqn.2001-04.com.example:storage.lun1
server1:~#
```

The settings for the storage device `iqn.2001-04.com.example:storage.lun1` on `192.168.0.101:3260,1` are stored in the file `/etc/iscsi/nodes/iqn.2001-04.com.example:storage.lun1/192.168.0.101,3260,1/default`. We need to set the username and password for the target in that file; instead of editing that file manually, we can use the `iscsiadm` command to do this for us:

```
iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.101:3260" --op=update --name node.session.auth.authmethod --value=CHAP

iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.101:3260" --op=update --name node.session.auth.username --value=someuser

iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.101:3260" --op=update --name node.session.auth.password --value=secret
```

Now we can log in, either by running...

```
iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.101:3260" --login
```

```
server1:~# iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.101:3260" --login
Logging in to [iface: default, target: iqn.2001-04.com.example:storage.lun1, portal: 192.168.0.101,3260]
Login to [iface: default, target: iqn.2001-04.com.example:storage.lun1, portal: 192.168.0.101,3260]: successful
server1:~#
```

... or by restarting the initiator:

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

(If you want to log out, you can run

```
iscsiadm -m node --targetname "iqn.2001-04.com.example:storage.lun1" --portal "192.168.0.101:3260" --logout
```

)

In the output of

```
fdisk -l
```

you should now find a new hard drive (`/dev/sdb` in this example); that's our iSCSI storage device:

```
server1:~# fdisk -l
```

```
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00031334
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	3749	30113811	83	Linux
/dev/sda2		3750	3916	1341427+	5	Extended
/dev/sda5		3750	3916	1341396	82	Linux swap / Solaris

```
Disk /dev/sdb: 21.4 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders
```

```
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk identifier: 0x00000000
```

```
Disk /dev/sdb doesn't contain a valid partition table
server1:~#
```

To use that device, we must format it:

```
fdisk /dev/sdb
```

```
server1:~# fdisk /dev/sdb
```

```
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x882944df.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
```

```
The number of cylinders for this disk is set to 20480.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)
```

```
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
```

```
Command (m for help): <-- m
```

```
Command action
```

- a toggle a bootable flag
- b edit bsd disklabel
- c toggle the dos compatibility flag

```

d  delete a partition
l  list known partition types
m  print this menu
n  add a new partition
o  create a new empty DOS partition table
p  print the partition table
q  quit without saving changes
s  create a new empty Sun disklabel
t  change a partition's system id
u  change display/entry units
v  verify the partition table
w  write table to disk and exit
x  extra functionality (experts only)

```

Command (m for help): <-- n

Command action

```

e  extended
p  primary partition (1-4)

```

<-- p

Partition number (1-4): <-- 1

First cylinder (1-20480, default 1): <-- ENTER

Using default value 1

Last cylinder or +size or +sizeM or +sizeK (1-20480, default 20480): <-- ENTER

Using default value 20480

Command (m for help): <-- t

Selected partition 1

Hex code (type L to list codes): <-- L

```

0  Empty          1e  Hidden W95 FAT1 80  Old Minix        be  Solaris boot
1  FAT12          24  NEC DOS          81  Minix / old Lin bf  Solaris
2  XENIX root     39  Plan 9          82  Linux swap / So c1  DRDOS/sec (FAT-
3  XENIX usr      3c  PartitionMagic  83  Linux           c4  DRDOS/sec (FAT-

```



```

4 FAT16 <32M      40 Venix 80286      84 OS/2 hidden C: c6 DRDOS/sec (FAT-
5 Extended       41 PPC PReP Boot   85 Linux extended c7 Syrinx
6 FAT16          42 SFS            86 NTFS volume set da Non-FS data
7 HPFS/NTFS     4d QNX4.x         87 NTFS volume set db CP/M / CTOS / .
8 AIX           4e QNX4.x 2nd part 88 Linux plaintext de Dell Utility
9 AIX bootable  4f QNX4.x 3rd part 8e Linux LVM       df BootIt
a OS/2 Boot Manag 50 OnTrack DM     93 Amoeba         e1 DOS access
b W95 FAT32      51 OnTrack DM6 Aux 94 Amoeba BBT     e3 DOS R/O
c W95 FAT32 (LBA) 52 CP/M          9f BSD/OS        e4 SpeedStor
e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a0 IBM Thinkpad hi eb BeOS fs
f W95 Ext'd (LBA) 54 OnTrackDM6    a5 FreeBSD      ee EFI GPT
10 OPUS          55 EZ-Drive      a6 OpenBSD      ef EFI (FAT-12/16/
11 Hidden FAT12   56 Golden Bow    a7 NeXTSTEP     f0 Linux/PA-RISC b
12 Compaq diagnost 5c Priam Edisk   a8 Darwin UFS   f1 SpeedStor
14 Hidden FAT16 <3 61 SpeedStor     a9 NetBSD       f4 SpeedStor
16 Hidden FAT16   63 GNU HURD or Sys ab Darwin boot   f2 DOS secondary
17 Hidden HPFS/NTF 64 Novell Netware b7 BSDI fs       fd Linux raid auto
18 AST SmartSleep 65 Novell Netware b8 BSDI swap     fe LANstep
1b Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid ff BBT
1c Hidden W95 FAT3 75 PC/IX

```

Hex code (type L to list codes): <-- 83

Command (m for help): <-- W

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

server1:~#

Afterwards, the output of

```
fdisk -l
```

should look as follows:

```
server1:~# fdisk -l
```

```
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00031334
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	3749	30113811	83	Linux
/dev/sda2		3750	3916	1341427+	5	Extended
/dev/sda5		3750	3916	1341396	82	Linux swap / Solaris

```
Disk /dev/sdb: 21.4 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Disk identifier: 0x882944df
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	20480	20971504	83	Linux

```
server1:~#
```

Now we create a filesystem on /dev/sdb1...

```
mkfs.ext3 /dev/sdb1
```

... and mount it for test purposes:

```
mount /dev/sdb1 /mnt
```

You should now see the new device in the outputs of...

```
mount
```

```
server1:~# mount
/dev/sda1 on / type ext3 (rw,errors=remount-ro)
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
udev on /dev type tmpfs (rw,mode=0755)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
/dev/sdb1 on /mnt type ext3 (rw)
server1:~#
```

... and

```
df -h
```

```
server1:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       29G   685M   27G   3% /
tmpfs           253M     0  253M   0% /lib/init/rw
udev            10M    88K   10M   1% /dev
tmpfs           253M     0  253M   0% /dev/shm
/dev/sdb1       20G   173M   19G   1% /mnt
server1:~#
```

You can unmount it like this:

```
umount /mnt
```

To have the device mounted automatically at boot time, e.g. in the directory `/storage`, we create that directory...

```
mkdir /storage
```

... and add the following line to `/etc/fstab`:

```
vi /etc/fstab
```

```
[...]  
/dev/sdb1 /storage ext3 defaults,auto,_netdev 0 0
```

For test purposes, you can now reboot the system:

```
reboot
```

After the reboot, the device should be mounted:

```
mount
```

```
server1:~# mount  
/dev/sda1 on / type ext3 (rw,errors=remount-ro)  
tmpfs on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)  
proc on /proc type proc (rw,noexec,nosuid,nodev)  
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)  
udev on /dev type tmpfs (rw,mode=0755)  
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)  
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=620)
```

```
/dev/sdb1 on /storage type ext3 (rw,_netdev)
server1:~#
```

```
df -h
```

```
server1:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       29G   685M   27G   3% /
tmpfs           253M    0   253M   0% /lib/init/rw
udev            10M    88K   10M   1% /dev
tmpfs           253M    0   253M   0% /dev/shm
/dev/sdb1       20G   173M   19G   1% /storage
server1:~#
```

4 Links

- Open-iSCSI: <http://www.open-iscsi.org/>
- iSCSI Enterprise Target: <http://iscsitarget.sourceforge.net/>
- Debian: <http://www.debian.org>