

# **iStorage Server: iSCSI SAN for Linux**

Friday, May 28, 2010

KernSafe Technologies, Inc.

[www.kernsafe.com](http://www.kernsafe.com)

Copyright © KernSafe Technologies 2006-2010. All right reserved.

## Table of Contents

|   |    |
|---|----|
| Overview.....                             | 3  |
| Install Linux.....                        | 3  |
| Configuring on iStorage Server .....      | 4  |
| Choose the Authentication Mechanism ..... | 4  |
| Create Target .....                       | 6  |
| Configure Linux.....                      | 12 |
| Install open-iscsi .....                  | 12 |
| Discovery iSCSI Target .....              | 14 |
| Log on to iSCSI Target.....               | 15 |
| Format Disk.....                          | 19 |
| Mount Disk .....                          | 20 |
| Contact.....                              | 22 |

## Overview

iStorage Server is a network based storage virtualization software powered by KernSafe Technologies, Inc. Being a powerful, full-featured and software-only iSCSI Target SAN solution, that can quickly convert existing Windows computer into IP SAN. Storage media of iSCSI Target can include existing storage devices such as the entire hard disks or partitions, CD-RWs, tapes and USB storage devices, as well as disk image file or CD image files including ISO9660(.iso), .bin, .mdf, .cdi, .b5i, .nrg, .ccd, .sub, .img, .raw and other image file formats. Furthermore, iStorage Server also supports a lot of features such as: VHD (Virtual Hard Disk), snapshots, STPI, RAID-1 and failover, these features are very important and popular in storage industry world and make iStorage Server is suitable for any size of business.

Linux is an operating system -- very much like UNIX -- that has become very popular over the last several years.

This article demonstrates how iStorage Server works with Linux. Such powerful combination will expand the application scope of your Linux server and workstation, thereby enabling WINDOWS server to expand the storage of your Linux computer. It also allows you to directly use the storage devices of the existing Windows server for Linux Server. With IP SAN solution provided by iStorage Server, you may install application and server software, as well as store data required wish you like. Your Linux computer's storage can be expanded in the following 3-most-commonly-used ways:

- Use Virtual Image File Disk Device to create a file-based virtual storage device for Linux, this allows quick data migration and backup.
- Directly use the physical disk or partition of Windows server. This enables you to make good use of resource. No additional configurations, just add the storage media to iSCSI Targets.
- Use CD/DVD/RW bridge device or Virtual CD/DVD to map physical CD/DVD drives or CD/DVD image files (iso, .bin, .mdf, .cdi, .b5i, .nrg, .ccd, .sub, .img, .raw) on your Windows server to CD/DVD devices on Linux.

After iStorage Server 2.0, it supports server side mirroring, synchronous replication and failover which allows user to create a high-availability iSCSI SAN for Linux.

## Install Linux

Linux must first be installed on to a suitable machine. For how to obtain or install Linux, please contact the Linux supplier.

## Configuring on iStorage Server

### Choose the Authentication Mechanism

Decide which authentication mechanisms you would want to use: **Anonymous**, **CHAP**, **IP address** or **Mixed** authentication.

#### 1), **Anonymous:**

All initiators will get full access permission without any authorization required.

#### 2) **CHAP (Challenge-handshake authentication protocol)**

All initiators need to specify a CHAP user and secret to connect to the target. iStorage Server has a built-in user called "Guest", which is used for initiators without CHAP secret specified.

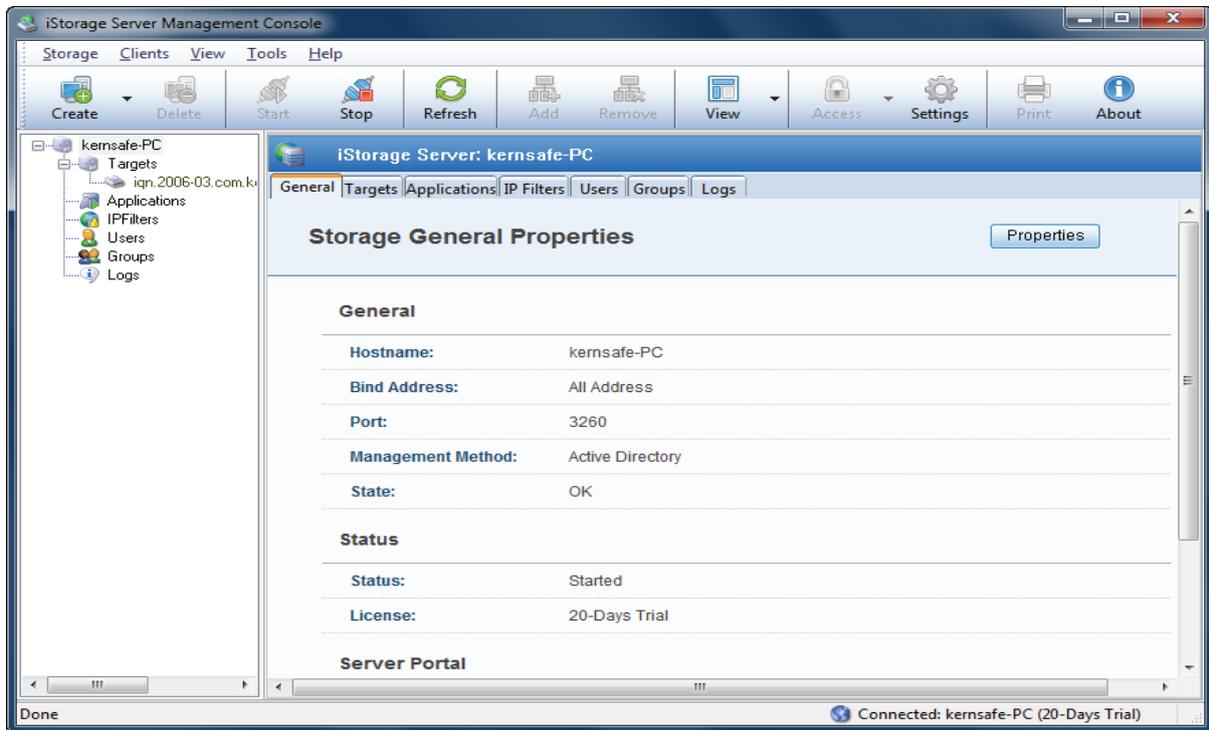
#### 3) **IP Filters**

All initiators will be authorized by the incoming IP address defined by IP Filter roles.

#### 4) **Mixed**

Security policy is determined by both CHAP and IP Filters.

Open **iStorage Server Management Console**.

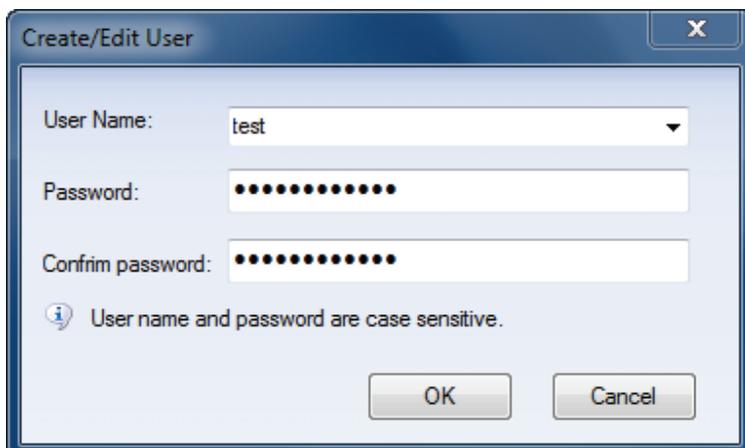


Create User

Right click **Users** Tree Node on the left tree view.

Press **Add** button in the tool bar.

**Create/Edit User** dialog is shown.



Type user name and password as you like, but we recommend that the password should be 12-16 characters. We take the user name **test** and password **111111111111** as an example.

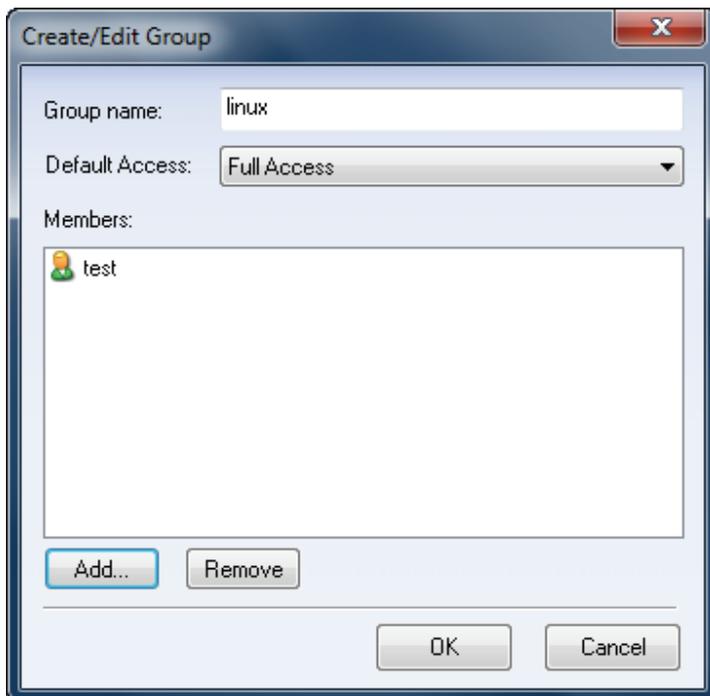
Press **OK** button to complete creating user.

Create group

After create user, we need a group to hold this user.

Right click **Groups** tree node in the left tree view.

Press **Add** button on the toolbar, the **Create/Edit Group** Dialog is shown.



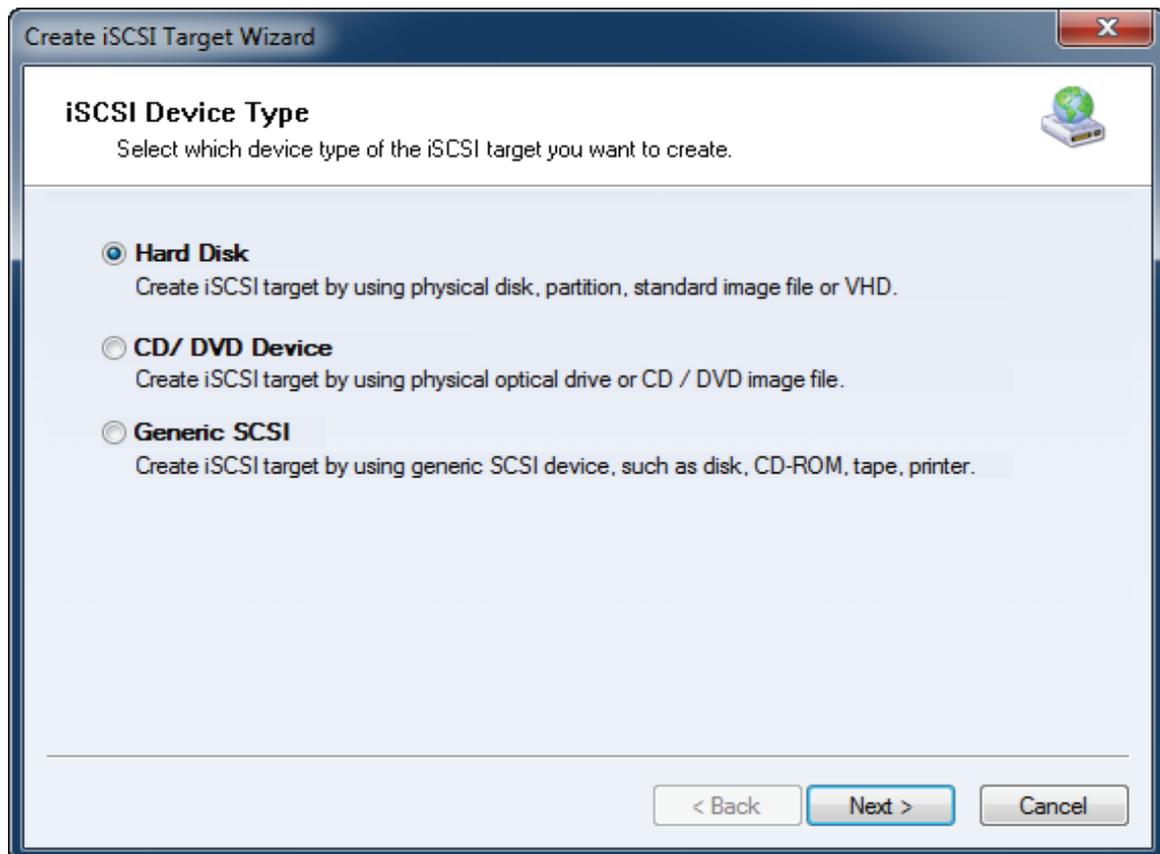
Take a group name as you like, we take **Linux** as an example.

Press the **Add** button and then select the user which we just created

## Create Target

Launch the **iStorage Server management console**, press the **Create** button on the toolbar of iStorage Server management console, the **Create Device Wizard** is shown.

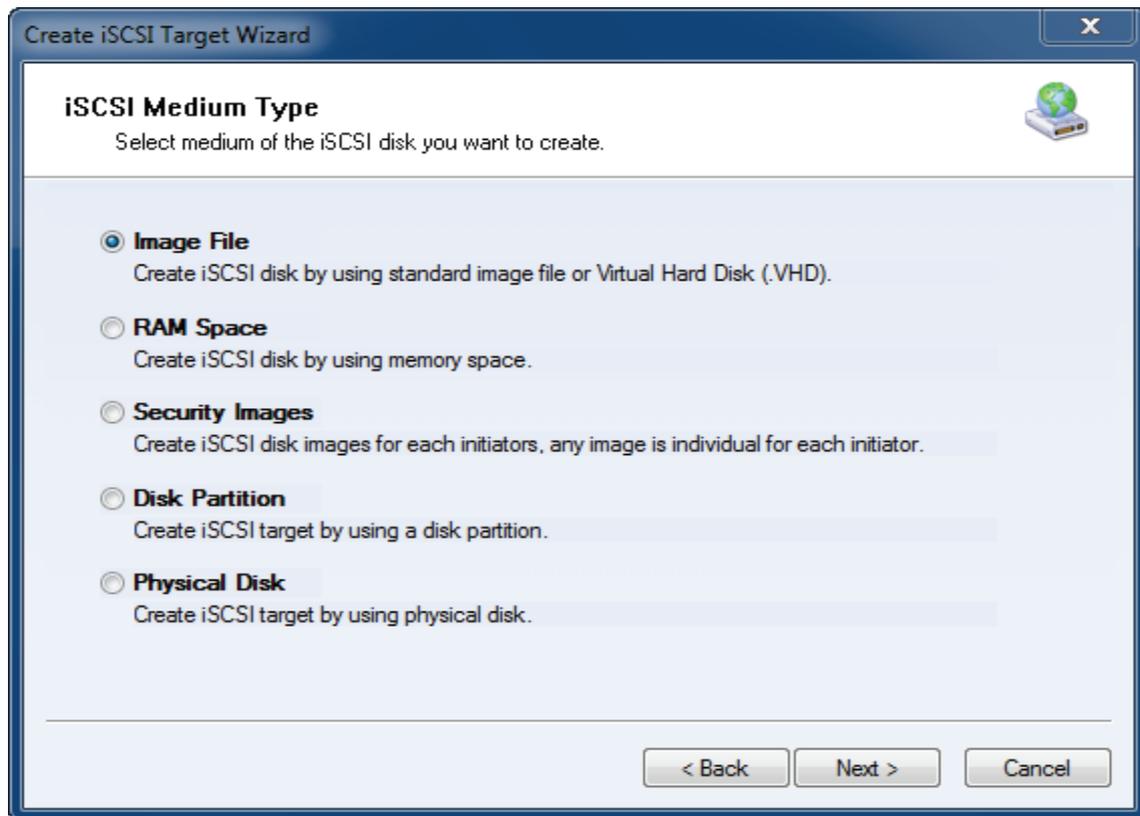
Select a device type



Choose **Hard Disk**.

Press the **Next** button to continue.

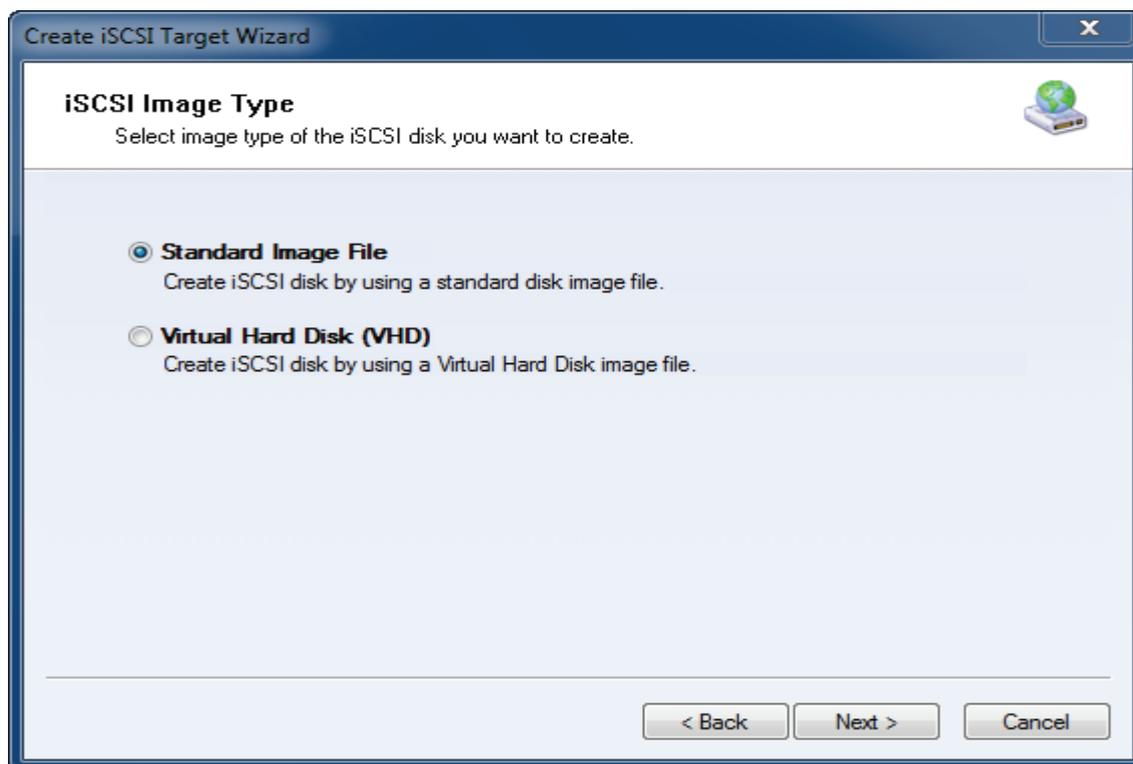
Select a medium type.



Choose **Image File** in **iSCSI Medium Type** window.

Then press **Next** button to continue.

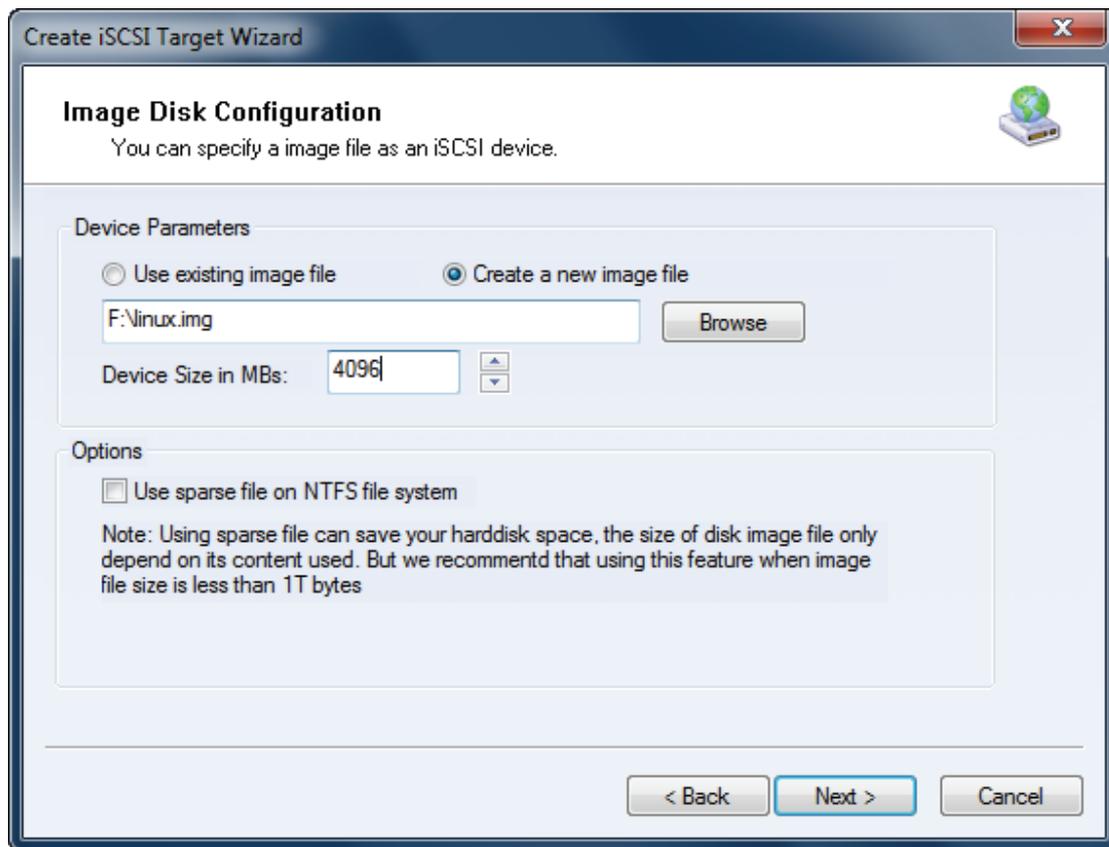
Select an Image type.



Choose **Standard Image File**.

Press the **Next** button to continue.

Specify image file path and size.



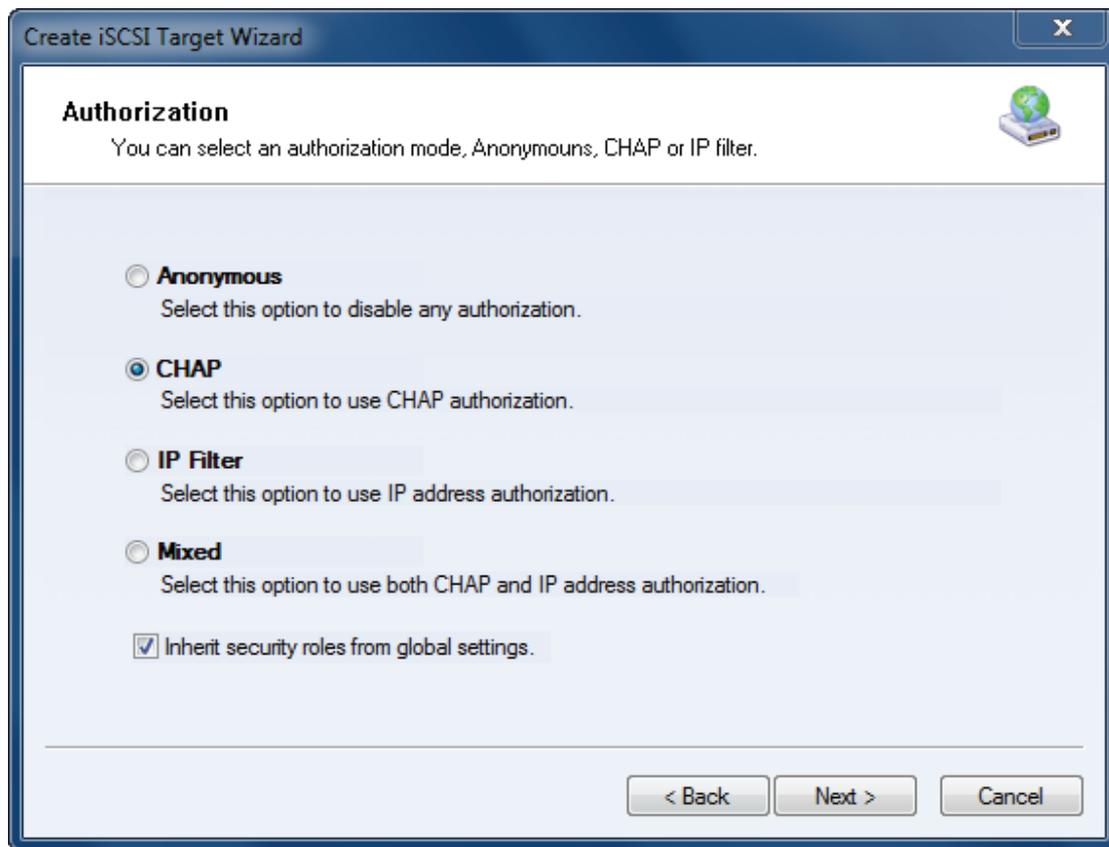
Specify the image file.

Specify the device size.

If you check **Use sparse file on NTFS file system**, the size of disk image file only depend on its content used, it can save your hard disk space.

Press the **Next** button to continue.

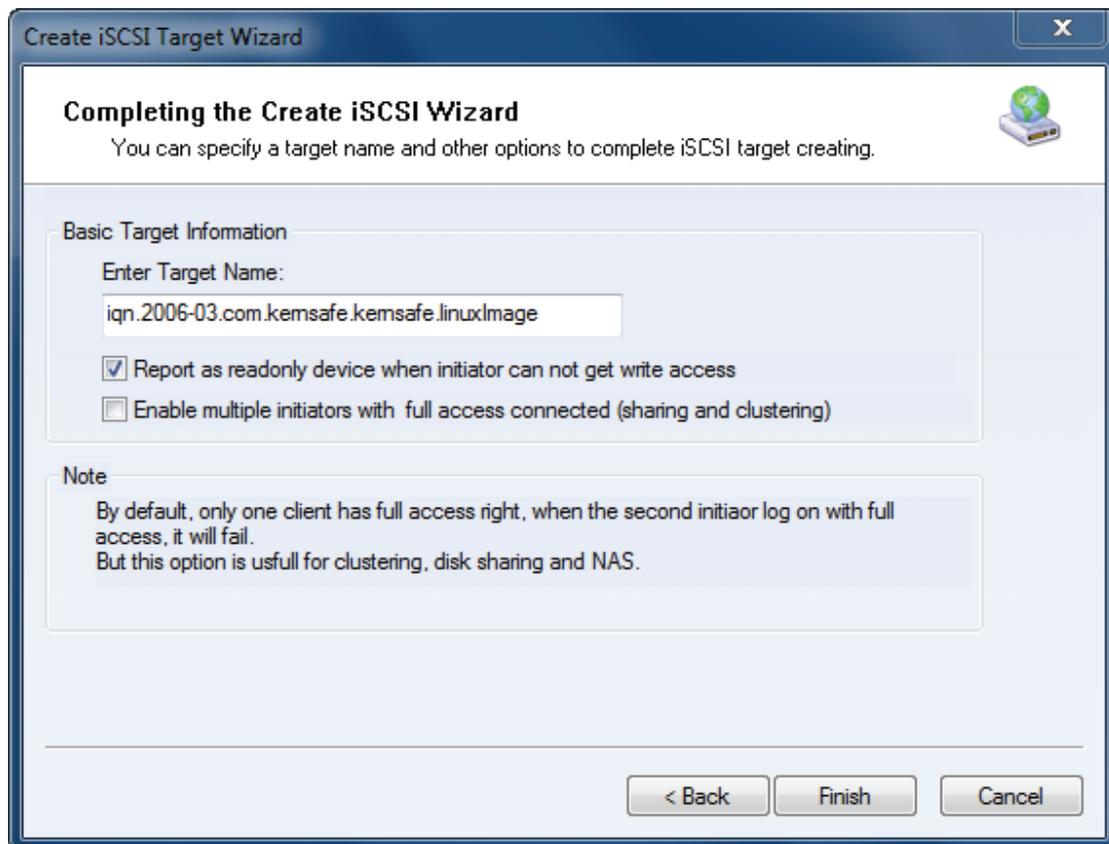
Set authorization mode.



Choose **CHAP** Authorization.

Press the **Next** button to continue.

Finish creating iSCSI Target



Type a target name in the Target Name field, or use the default.

Press the **Finish** button to continue.

## Configure Linux

### Install open-iscsi

Download open-iscsi from <http://www.open-iscsi.org>.

Make sure current directory is the open-iscsi's source directory.

Type **make install** in the console.



```
root@localhost:~/open-iscsi-2.0-870.3
File Edit View Terminal Tabs Help
[root@localhost ~]# cd '/root/open-iscsi-2.0-870.3'
[root@localhost open-iscsi-2.0-870.3]# make
make -C utils/fwparam_ibft
make[1]: Entering directory `/root/open-iscsi-2.0-870.3/utils/fwparam_ibft'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/root/open-iscsi-2.0-870.3/utils/fwparam_ibft'
make -C usr
make[1]: Entering directory `/root/open-iscsi-2.0-870.3/usr'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/root/open-iscsi-2.0-870.3/usr'
make -C kernel
make[1]: Entering directory `/root/open-iscsi-2.0-870.3/kernel'
make -C /lib/modules/2.6.25-14.fc9.i686/build M=`pwd` KBUILD_OUTPUT= V=0 module
s
make[2]: Entering directory `/usr/src/kernels/2.6.25-14.fc9.i686'
Building modules, stage 2.
MODPOST 3 modules
make[2]: Leaving directory `/usr/src/kernels/2.6.25-14.fc9.i686'
make[1]: Leaving directory `/root/open-iscsi-2.0-870.3/kernel'
make -C utils
make[1]: Entering directory `/root/open-iscsi-2.0-870.3/utils'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/root/open-iscsi-2.0-870.3/utils'

Compilation complete                Output file
-----
Built iSCSI Open Interface module:  kernel/scsi_transport_iscsi.ko
Built iSCSI library module:         kernel/libiscsi.ko
Built iSCSI over TCP kernel module: kernel/iscsi_tcp.ko
Built iSCSI daemon:                 usr/iscsid
Built management application:       usr/iscsiadm

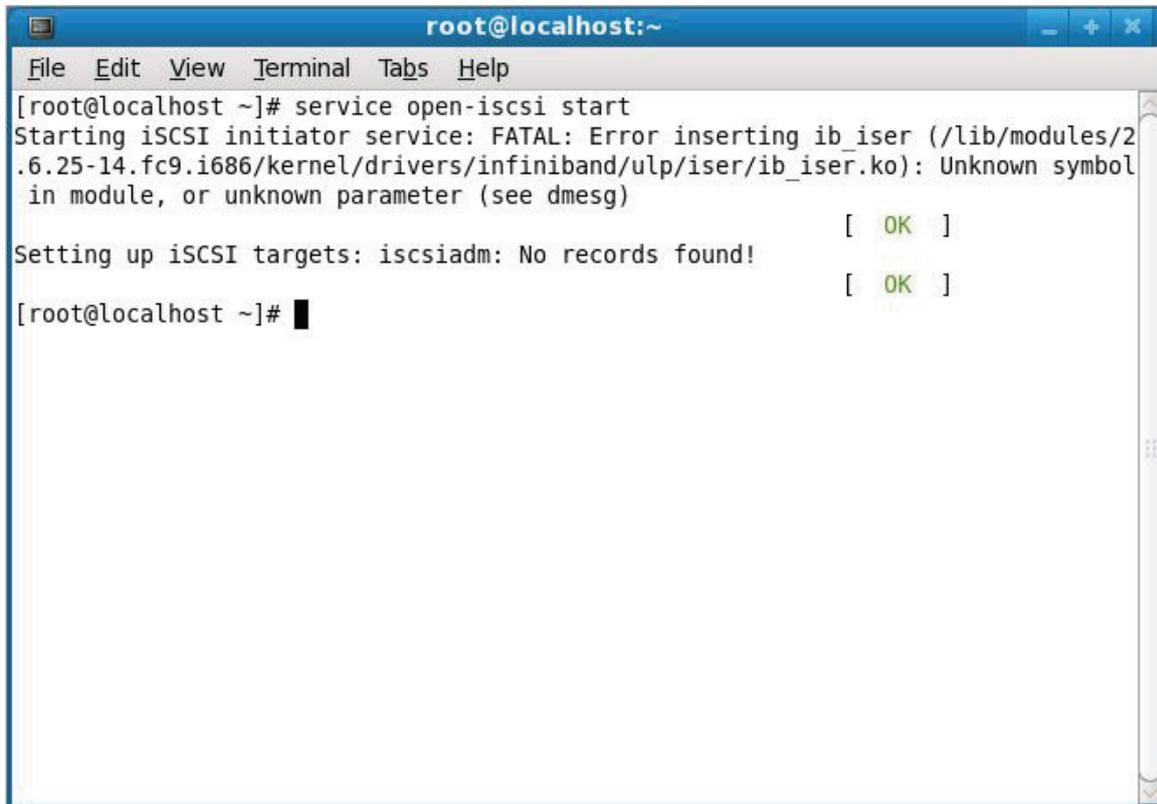
Read README file for detailed information.
[root@localhost open-iscsi-2.0-870.3]#
```

Hit the **Enter** key to continue.

Start iSCSI service.

By default, the `iscsi` service is stopped, you need to start it manually.

Type **service open-iscsi start** in the console.

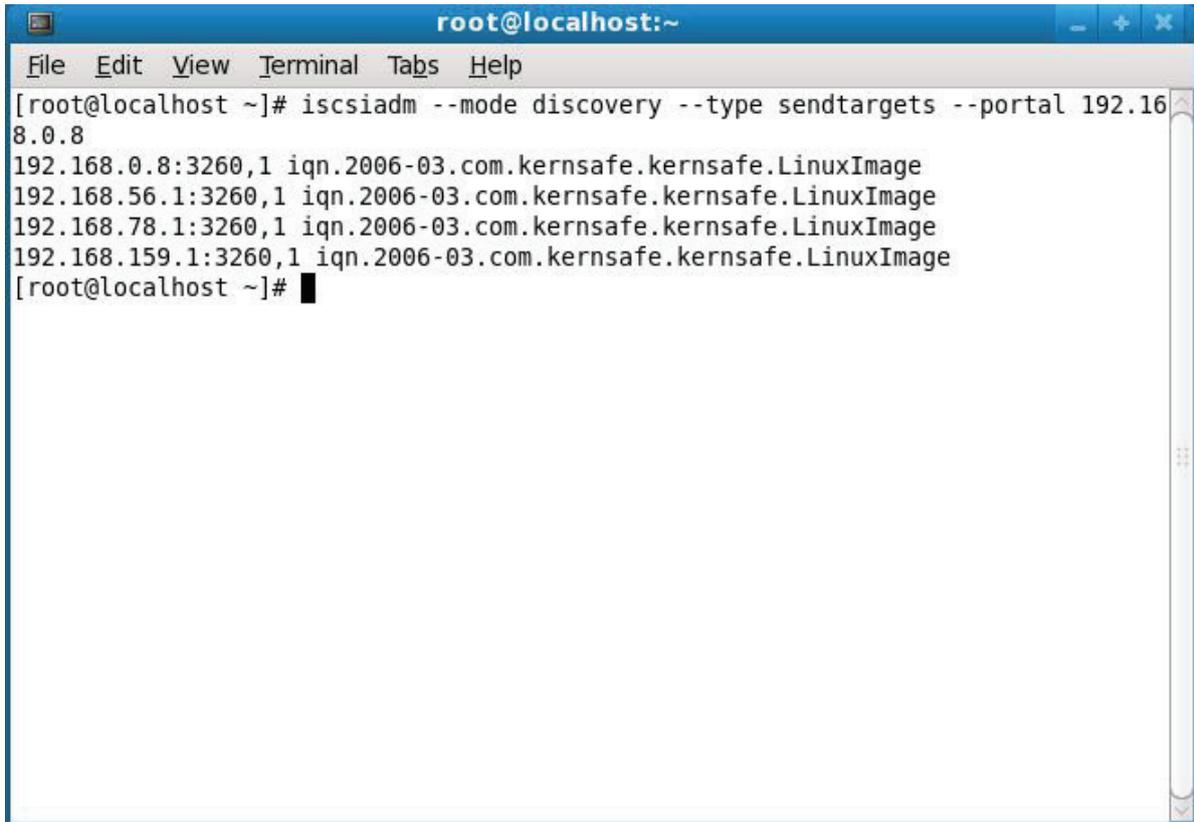
A terminal window titled "root@localhost:~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal output shows the command "service open-iscsi start" being executed. The output is: "Starting iSCSI initiator service: FATAL: Error inserting ib\_iser (/lib/modules/2.6.25-14.fc9.i686/kernel/drivers/infiniband/ulp/iser/ib\_iser.ko): Unknown symbol in module, or unknown parameter (see dmesg)", followed by a green "[ OK ]" status. Then, "Setting up iSCSI targets: iscsiadm: No records found!" is shown, followed by another green "[ OK ]" status. The prompt "[root@localhost ~]#" is visible at the bottom with a cursor.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# service open-iscsi start  
Starting iSCSI initiator service: FATAL: Error inserting ib_iser (/lib/modules/2  
.6.25-14.fc9.i686/kernel/drivers/infiniband/ulp/iser/ib_iser.ko): Unknown symbol  
in module, or unknown parameter (see dmesg) [ OK ]  
Setting up iSCSI targets: iscsiadm: No records found! [ OK ]  
[root@localhost ~]#
```

Hit the **Enter** key to continue.

## Discovery iSCSI Target

Type `iscsiadm --mode discovery --type sendtargets --portal 192.168.0.8` in the console, the server address **192.168.0.8** can be changed to the IP address of your own iSCSI Target server.

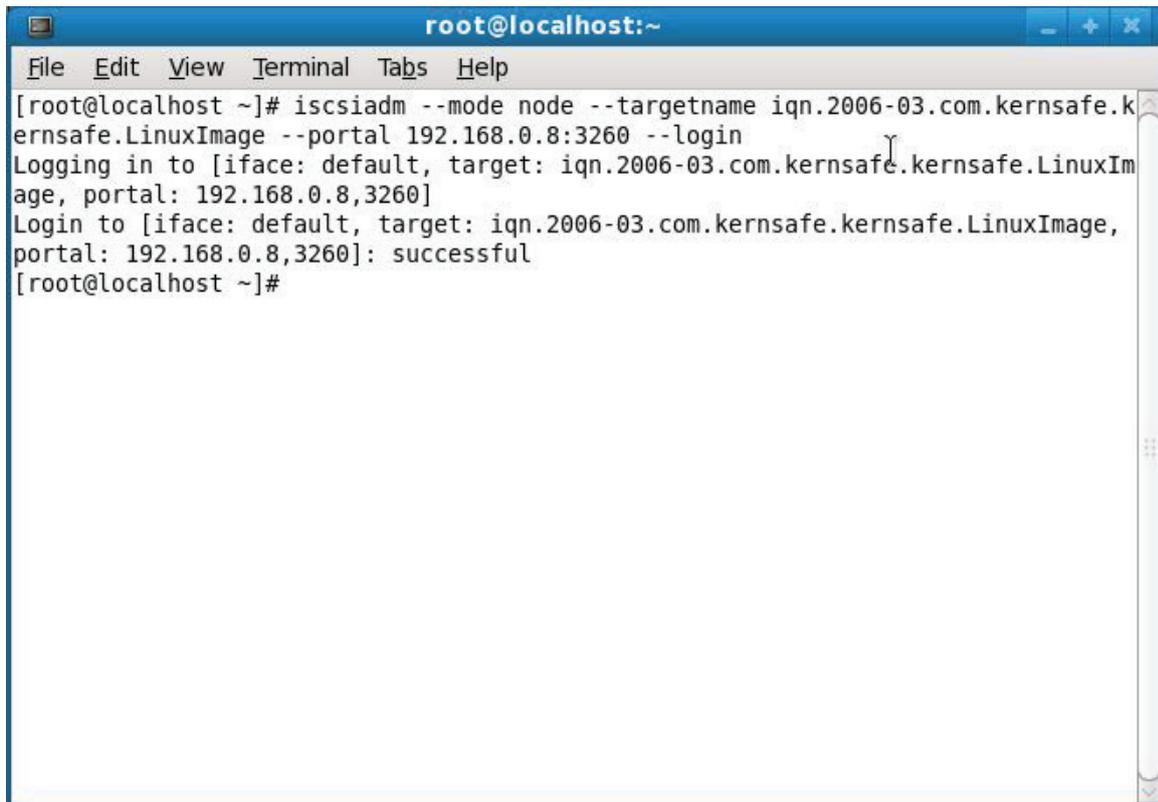


```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# iscsiadm --mode discovery --type sendtargets --portal 192.168.0.8  
192.168.0.8:3260,1 iqn.2006-03.com.kernsafe.kernsafe.LinuxImage  
192.168.56.1:3260,1 iqn.2006-03.com.kernsafe.kernsafe.LinuxImage  
192.168.78.1:3260,1 iqn.2006-03.com.kernsafe.kernsafe.LinuxImage  
192.168.159.1:3260,1 iqn.2006-03.com.kernsafe.kernsafe.LinuxImage  
[root@localhost ~]#
```

Hit the **Enter** key to continue.

## Log on to iSCSI Target

Log on without authorization, type `iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal 192.168.0.8:3260 --login` in the console.

A terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal output shows the execution of the 'iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal 192.168.0.8:3260 --login' command. The output includes 'Logging in to [iface: default, target: iqn.2006-03.com.kernsafe.kernsafe.LinuxImage, portal: 192.168.0.8,3260]' and 'Login to [iface: default, target: iqn.2006-03.com.kernsafe.kernsafe.LinuxImage, portal: 192.168.0.8,3260]: successful'. The prompt returns to '[root@localhost ~]#'.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal 192.168.0.8:3260 --login  
Logging in to [iface: default, target: iqn.2006-03.com.kernsafe.kernsafe.LinuxImage, portal: 192.168.0.8,3260]  
Login to [iface: default, target: iqn.2006-03.com.kernsafe.kernsafe.LinuxImage, portal: 192.168.0.8,3260]: successful  
[root@localhost ~]#
```

Hit the Enter key to continue.

Log on with CHAP user, type the following command in the console:

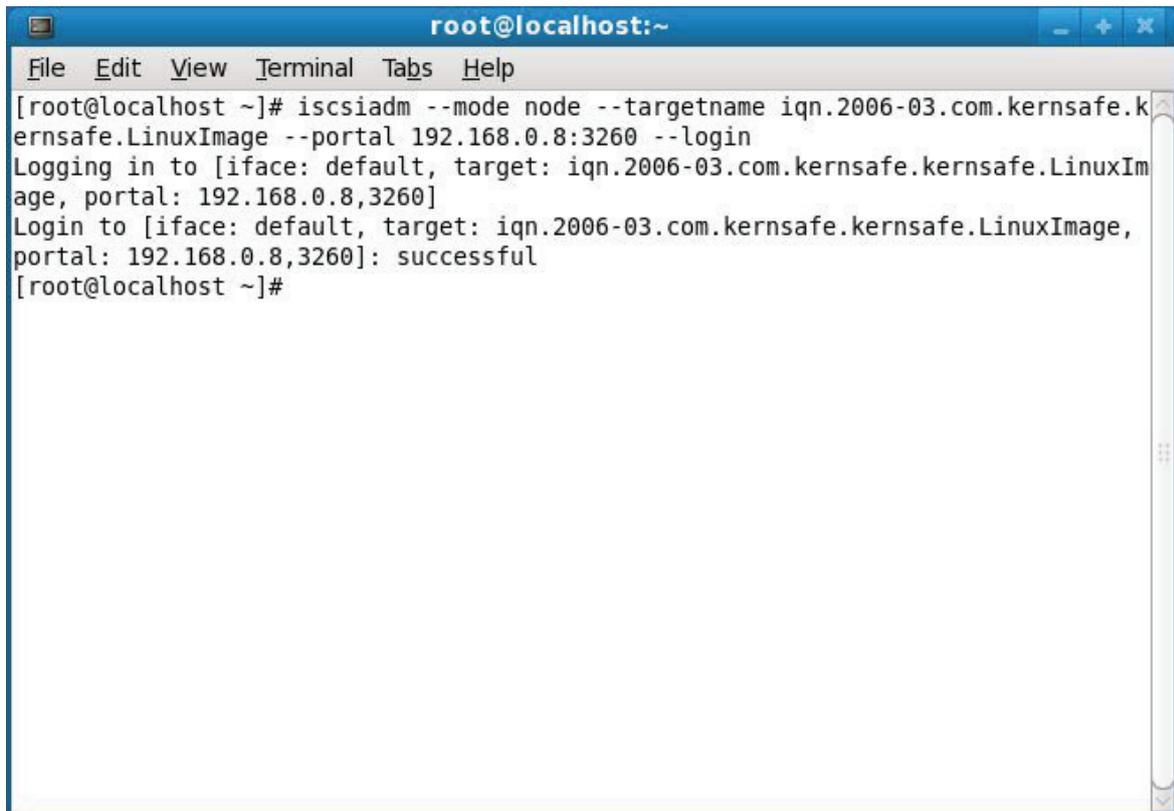
```
iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal  
192.168.0.8:3260 --name node.session.auth.authmethod --value=CHAP --op=update
```

```
iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal  
192.168.0.8:3260 --name node.session.auth.username --value=test --op=update
```

```
iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal  
192.168.0.8:3260 --name node.session.auth.password --value=111111111111 --op=update
```

Hit the **Enter** key for each command to modify the configuration file of open-iscsi, so that open-iscsi can make CHAP log on.

Type **iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal 192.168.0.8:3260 --login** in the console.

A terminal window titled "root@localhost:~" with a menu bar containing "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows the execution of the command "iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal 192.168.0.8:3260 --login". The output indicates a successful login to the specified target and portal.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# iscsiadm --mode node --targetname iqn.2006-03.com.kernsafe.kernsafe.LinuxImage --portal 192.168.0.8:3260 --login  
Logging in to [iface: default, target: iqn.2006-03.com.kernsafe.kernsafe.LinuxImage, portal: 192.168.0.8,3260]  
Login to [iface: default, target: iqn.2006-03.com.kernsafe.kernsafe.LinuxImage, portal: 192.168.0.8,3260]: successful  
[root@localhost ~]#
```

Hit the **Enter** key to complete iSCSI log on.

## Partition Disk

Use the **fdisk** utility to make disk partition, type **fdisk /dev/sdb** in the console.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# 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 0xa0467a14.  
Changes will remain in memory only, until you decide to write them.  
After that, of course, the previous content won't be recoverable.  
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)  
Command (m for help): p  
Disk /dev/sdb: 2147 MB, 2147483648 bytes  
67 heads, 62 sectors/track, 1009 cylinders  
Units = cylinders of 4154 * 512 = 2126848 bytes  
Disk identifier: 0xa0467a14  


| Device | Boot | Start | End | Blocks | Id | System |
|--------|------|-------|-----|--------|----|--------|
|--------|------|-------|-----|--------|----|--------|

  
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)
```

Hit the **Enter** key to continue.

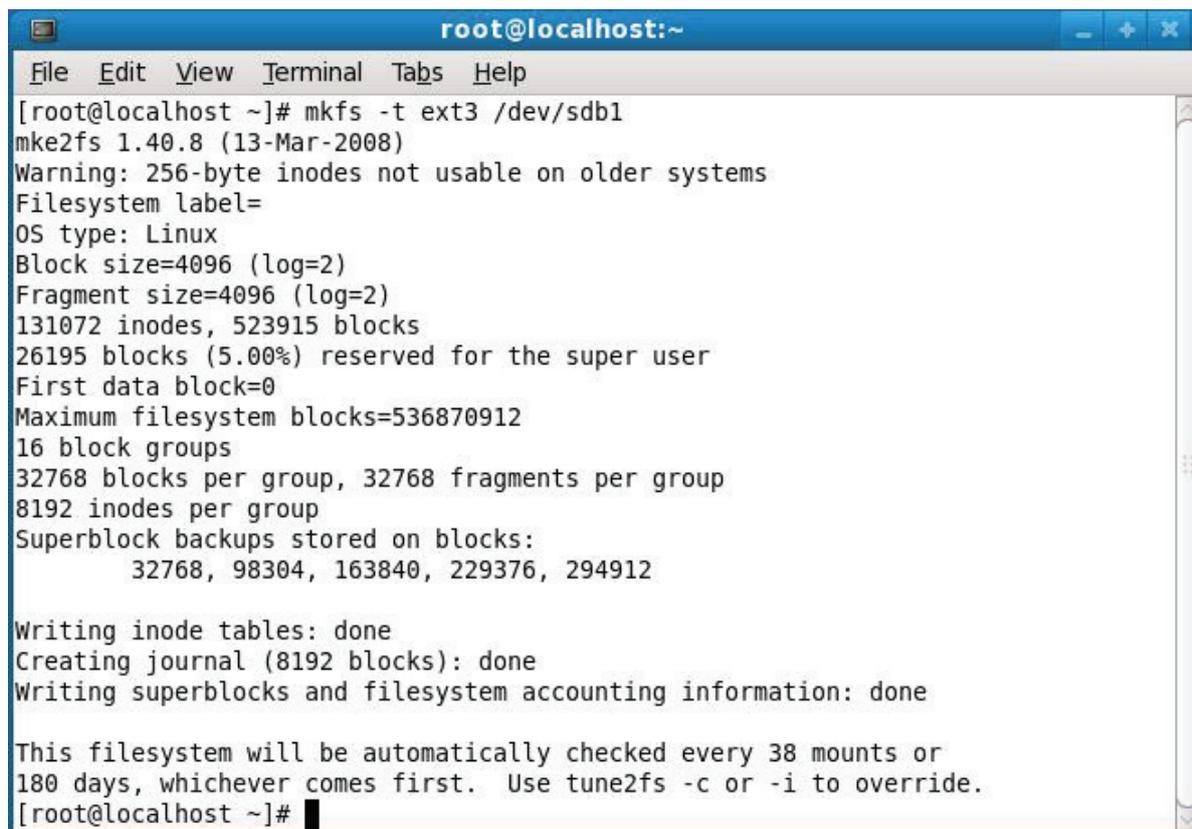
We found the blank disk in the screen, type the command followed by the below screen.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# 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 0xa7bf5b1c.  
Changes will remain in memory only, until you decide to write them.  
After that, of course, the previous content won't be recoverable.  
  
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)  
  
Command (m for help): n  
Command action  
  e   extended  
  p   primary partition (1-4)  
p  
Partition number (1-4): 1  
First cylinder (1-1009, default 1): 1  
Last cylinder or +size or +sizeM or +sizeK (1-1009, default 1009): 1009  
  
Command (m for help): p  
  
Disk /dev/sdb: 2147 MB, 2147483648 bytes  
67 heads, 62 sectors/track, 1009 cylinders  
Units = cylinders of 4154 * 512 = 2126848 bytes  
Disk identifier: 0xa7bf5b1c  
  
   Device Boot      Start         End      Blocks   Id  System  
/dev/sdb1            1         1009     2095662    83  Linux  
  
Command (m for help): w  
The partition table has been altered!  
  
Calling ioctl() to re-read partition table.  
Syncing disks.  
[root@localhost ~]#
```

Type **w** command to complete partition table creation.

## Format Disk

Type **mkfs -t ext3 /dev/sdb1** in the console, or you can change **ext3** to another file system.



```
root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# mkfs -t ext3 /dev/sdb1
mke2fs 1.40.8 (13-Mar-2008)
Warning: 256-byte inodes not usable on older systems
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
131072 inodes, 523915 blocks
26195 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=536870912
16 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 38 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@localhost ~]#
```

Hit the **Enter** key to continue.

## Mount Disk

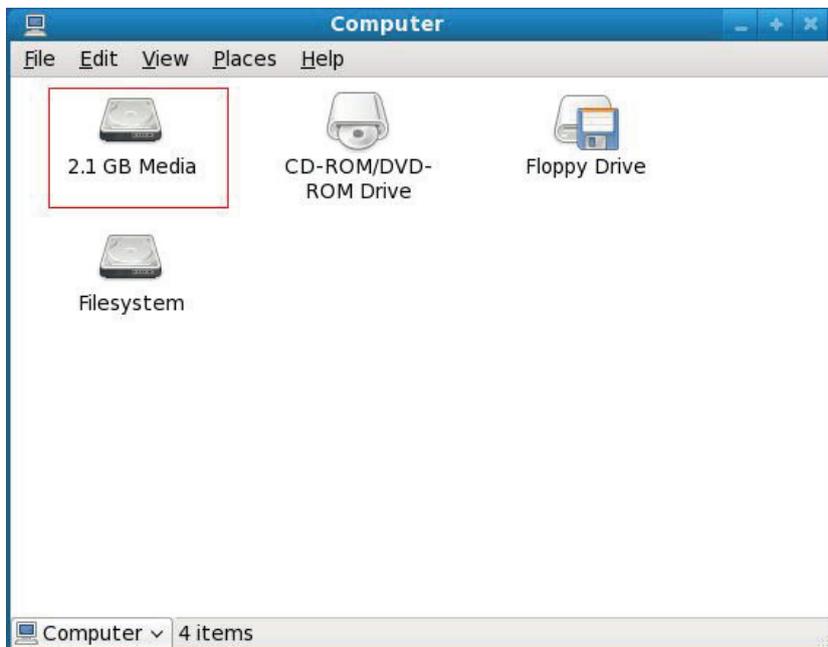
Type `mkdir /linuximage` and hit **Enter** key to create a folder to hold his iSCSI disk.

Type `mount /dev/sdb1 /linuximage` in the console.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# mkdir /LinuxImage  
[root@localhost ~]# mount /dev/sdb1 /LinuxImage  
[root@localhost ~]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup00-LogVol00  
6.7G  4.2G  2.2G  66% /  
/dev/sda1      190M   13M  168M   7% /boot  
tmpfs          643M   48K  643M   1% /dev/shm  
gvfs-fuse-daemon 6.7G  4.2G  2.2G  66% /root/.gvfs  
/dev/sdb1      2.0G   36M  1.9G   2% /LinuxImage  
[root@localhost ~]#
```

Hit the **Enter** key to mount.

We will find the disk in the machine.



Now, you can do anything with the disk just as local hard disk.

## Contact

Support: [support@kernsafe.com](mailto:support@kernsafe.com)  
Sales: [sales@kernsafe.com](mailto:sales@kernsafe.com)  
Home Page: <http://www.kernsafe.com/>  
Product Page: <http://www.kernsafe.com/product.aspx?id=5>  
Licenses <http://www.kernsafe.com/product.aspx?id=5&name=License+Types>  
Forum: <http://www.kernsafe.com/forum/>

KernSafe Technologies, Inc.

[www.kernsafe.com](http://www.kernsafe.com)

Copyright © KernSafe Technologies 2006-2010. All right reserved.