

Application Note

Infortrend[®]

Install and Manage iSCSI Volumes Using Linux Initiators

For CentOS and RedHat Linux

Abstract

This document describes how to enable the access to Infortrend's iSCSI storage using Linux's iSCSI initiators.

Infortrend[®] Technology, Inc.

Networked Storage Solution Provider

Revision 1.0

October, 2008

Infortrend Technology makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Infortrend reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation to notify of such changes.

Using Linux Initiator with EonStor iSCSI Arrays

Open-iSCSI Project

[Open-iSCSI](#) project is a high-performance, transport independent, multi-platform implementation of iSCSI. Open-iSCSI is partitioned into user and kernel parts.

Instructions are tested on:

[a] RHEL 5

[b] CentOS 5

[c] Fedora 7

[d] Debian / Ubuntu Linux

Install Required Package

iscsi-initiator-utils RPM package - The iSCSI package provides the server daemon for the iSCSI protocol, as well as the utility programs used to manage it. iSCSI is a protocol for distributed disk access using SCSI commands sent over Internet Protocol networks. This package is available under Redhat Enterprise Linux / CentOS / Fedora Linux and can be installed using the yum command:

```
# yum install iscsi-initiator-utils
```

A note about Debian / Ubuntu Linux

If you are using **Debian / Ubuntu** Linux install open-iscsi package, enter:

```
$ sudo apt-get install open-iscsi
```

iSCSI Configuration

There are three steps for setting up a configuration using iSCSI storage:

1. iSCSI startup using the init script or manual startup. You need to edit and configure iSCSI settings via the “/etc/iscsi/iscsid.conf” file
2. Discover targets.
3. Automate target logins for future system reboots.
4. You also need to obtain iSCSI username, password and storage host port IP addresses.

Step # 1: Configure iSCSI

Open /etc/iscsi/iscsid.conf with vi text editor:

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

Setup username and password:

```
node.session.auth.username = My_ISCSI_USR_NAME  
node.session.auth.password = MyPassword  
discovery.sendtargets.auth.username = My_ISCSI_USR_NAME  
discovery.sendtargets.auth.password = MyPassword
```

Where,

- **node.session.*** is used to set a CHAP username and password for initiator authentication by the target(s).
- **discovery.sendtargets.*** is used to set a discovery session CHAP username and password for the initiator authentication by the target(s)

You may also need to tweak and set other options.

```
# /etc/init.d/iscsi start
```

Step # 2: Discover targets

Now use `iscsiadm` command, which is a command-line tool allowing discovery and login to iSCSI targets, as well as access and management of the open-iscsi database. If your storage port IP address (target port) is 192.168.1.5, enter:

```
# iscsiadm -m discovery -t sendtargets -p 192.168.1.5
# /etc/init.d/iscsi restart
```

Now there should be a block device under `/dev` directory. To obtain new device name, type:

```
# fdisk -l
```

or

```
# tail -f /var/log/messages
```

Output:

```
Oct 31 16:08:35 localhost kernel: scsi6 : iSCSI Initiator over TCP/IP
Oct 31 16:08:35 localhost kernel: scsi7 : iSCSI Initiator over TCP/IP
Oct 31 16:08:36 localhost kernel: Vendor: IFT Model: S12E-R1132-4 Rev: 364J
Oct 31 16:08:36 localhost kernel: Type: Direct-Access ANSI SCSI revision: 04
Oct 31 16:08:36 localhost kernel: Vendor: IFT Model: S12E-R1132-4 Rev: 364J
Oct 31 16:08:36 localhost kernel: Type: Direct-Access ANSI SCSI revision: 04
Oct 31 16:08:36 localhost kernel: SCSI device sdb: 142589952 512-byte hdwr sectors (73006 MB)
Oct 31 16:08:36 localhost kernel: sdb: Write Protect is off
Oct 31 16:08:36 localhost kernel: SCSI device sdb: drive cache: write back
Oct 31 16:08:36 localhost kernel: SCSI device sdc: 142589952 512-byte hdwr sectors (73006 MB)
Oct 31 16:08:36 localhost kernel: SCSI device sdb: 142589952 512-byte hdwr sectors (73006 MB)
Oct 31 16:08:36 localhost kernel: sdc: Write Protect is off
Oct 31 16:08:36 localhost kernel: sdb: Write Protect is off
Oct 31 16:08:36 localhost kernel: SCSI device sdc: drive cache: write back
Oct 31 16:08:36 localhost kernel: SCSI device sdb: drive cache: write back
Oct 31 16:08:36 localhost kernel: sdb: sdb1
Oct 31 16:08:36 localhost kernel: sdb: p1 exceeds device capacity
Oct 31 16:08:36 localhost kernel: sd 6:0:0:0: Attached scsi disk sdb
Oct 31 16:08:36 localhost kernel: sd 6:0:0:0: Attached scsi generic sg1 type 0
Oct 31 16:08:36 localhost kernel: SCSI device sdc: 142589952 512-byte hdwr sectors (73006 MB)
Oct 31 16:08:36 localhost kernel: sdc: Write Protect is off
Oct 31 16:08:36 localhost kernel: SCSI device sdc: drive cache: write back
Oct 31 16:08:36 localhost kernel: sdc: sdc1
Oct 31 16:08:36 localhost kernel: sdc: p1 exceeds device capacity
Oct 31 16:08:36 localhost kernel: sd 7:0:0:0: Attached scsi disk sdc
Oct 31 16:08:36 localhost kernel: sd 7:0:0:0: Attached scsi generic sg2 type 0
Oct 31 16:08:36 localhost iscsid: transport class version 2.0-724. iscsid version 2.0-868
Oct 31 16:08:36 localhost iscsid: iSCSI daemon with pid=19084 started!
Oct 31 16:08:36 localhost iscsid: received iferror -38
Oct 31 16:08:36 localhost last message repeated 2 times
Oct 31 16:08:36 localhost iscsid: connection1:0 is operational now
```

The `/dev/sdx` is a RAID volume that is recognized as a new block device.

Step # 3: Format and Mount iSCSI Volume

You can now partition and create a filesystem on the target using usual fdisk and mkfs.ext3 commands:

```
# fdisk /dev/sdd  
# mke2fs -j -m 0 -O dir_index /dev/sdd1
```

- OR -

```
# mkfs.ext3 /dev/sdd1
```

Tip: If your volume is large, such as 1TB, run mkfs.ext3 in the background using nohup:

```
# nohup mkfs.ext3 /dev/sdd1 &
```

Mount new partition:

```
# mkdir /mnt/iscsi  
# mount /dev/sdd1 /mnt/iscsi
```

Step #4: Mount iSCSI drive automatically at boot time

First make sure iscsi service is turned on at boot time:

```
# chkconfig iscsi on
```

Open /etc/fstab file and append config directive:

```
/dev/sdd1 /mnt/iscsi ext3 _netdev 0 0
```

Save and close the file.