Cleversafe dsNet User's Guide

Version 1.0.4b1

Cleversafe dsNet User's Guide: Version 1.0.4b1

Table of Contents

Chapter 1. Requirements

Software Requirements

This guide provides instructions for the 1.0.4b1 version of the Cleversafe dsNet. This and other versions of the software can be downloaded from Sourceforge at: http://sourceforge.net/projects/cleversafe[1]. These are the two packages that you will need to download:

- dsnet-accesser-1.0.4b1.zip
- dsnet-slicestor-1.0.4b1.zip

These are the other requirements needed to run a dsNet:

- Linux or an environment which supports a UNIX shell (/bin/sh)
- Sun Java JRE 1.5 or later

> **Important:** Ensure that files and folders have proper permissions. Some of the command line applications create files which are read by other entities. This is best ensured by running all commands as the same user when possible. By default, some distriubutions of the DSNet software run the processes as user "dsnet". You should run all commands in this document as user dsnet if this is the case. The default ZIP distribution only requires that all commands are run as the same user.

Notes

1. http://sourceforge.net/projects/cleversafe

Chapter 2. Usage

Setup

Setup the dsNet Slicestor and Accesser

1. Unzip the dsNet Accesser zip file to the directory of your choice

2. Enter the `dsnet-slicestor/setup` directory and run **setup.sh** (You must be inside `/dsnet-slicestor/setup` to run **setup.sh**)

3. Two new files should be generated: `/dsnet-slicestor/conf/wrapper-server.conf` and `/dsnet-slicestor/bin/slicestor`. The default configuration has the server run a single server rather than a multiple server setup. `/dsnet-slicestor/conf/wrapper-server.conf` contains a section that looks like this:

   ```
   # Parameters to run a single slicestor instance per server. If multiple
   # instances per server are desired, comment out line 4 and uncomment the
   # (commented by default) lines 4 and 6.
   wrapper.java.additional.1=-server
   wrapper.java.additional.2=-Dlog4j.configuration=@BASEPATH@/conf/log4j-local.xm
   wrapper.java.additional.3=-Dorg.cleversafe.xml.configuration=@BASEPATH@/conf/c
   # Comment out this line to run multiple instances per server.
   wrapper.java.additional.4=-Dorg.cleversafe.storage.ss.xml.configuration=@BASEP
   # Comment out this line to run multiple instances per server.
   #wrapper.java.additional.4=-Dorg.cleversafe.storage.ss.xml.configuration=@BASE
   wrapper.java.additional.5=-Dorg.cleversafe.xml.properties=@BASEPATH@/conf/prop
   # Comment out this line to run multiple instances per server.
   #wrapper.java.additional.6=-Dorg.cleversafe.storage.ss.daemons=8
   ```

   To run an 8-wide dsnet on the same machine for testing purposes, follow the instructions as provided in the comments section.

4. By default, the **/dsnet-slicestor/bin/slicestor** script has a section titled RUN_AS_USER. Change the username in this file to be any user which has read and write access to the `/dsnet-slicestor` folder. To run as the current user (as in the user which is logged in when the slicestor script is run), simply leave the line commented out.

   ```
   RUN_AS_USER="dsnet"
   ```

   Run **/dsnet-slicestor/bin/slicestor start**. Slicestor, unlike **setup.sh**, can be run from anywhere on the local system (paths are made un-relative as part of the setup process). The server should now be running. Also note that the previous steps only need be executed once per install.

5. Run **/dsnet-accesser/setup/setup.sh**. As with the slicestor, you must be inside `/dsnet-accesser/setup` to run **setup.sh**.

6. Start the accesser daemon: **dsnet-accesser/bin/accesser start**

7. Run **/dsnet-accesser/bin/dsnet-create-vault**: **dsnet-accesser/bin/dsnet-create-vault --name="vault-name" --descriptor="../conf/descriptor.xml"**

8. Now that the vault has been created, a new target needs to be created on the iSCSI portal with the **dsnet-create-target** command:

   ```
   dsnet-accesser/bin/dsnet-create-target --name="vault"
            --iqn="this-is-the-iqn"
   ```

At this point, the accesser should be started and ready for an initiator to connect.

Optional commands

A number of optional commands are available.

- **dsnet-list-vaults**: Connects to the accesser daemon and lists all created vaults.
- **dsnet-list-targets**: Connects to the accesser daemon and lists all created iSCSI targets.
- **dsnet-delete-target**: Connects to the accesser daemon and deletes an iSCSI target.
- **dsnet-delete-vault**: Connects to the accesser daemon and deletes a created vault (note that any targets using this vault must be deleted first).
- **dsnet-change-portal**: Change iSCSI portal.

For more information on these optional commands, please refer to the command line help interfaces provided when these commands are executed.

Stopping Accesser and Slicestor

The scripts in the bin directory of both the accesser and slicestor distributions (after running **setup.sh**) act like scripts that would normally be inserted into `/etc/init.d`. You can issue start, stop, or status commands to them.

Note that there is an issue where if the install path of the accesser or slicestor is long, the stop commands will not work. A workaround is pending.

Chapter 3. Uninstall

Since the Accesser and Slicestor are self-contained in one directory (including all configuration and data), simply delete the installation directory.

Chapter 4. Using the Accesser's iSCSI Target Functionality

Log on to the iSCSI Target

You can use the Open-iSCSI Project's[1] **iscsiadm** program to log in to the target by typing:

```
iscsiadm --mode node --targetname [target-name] --portal [portal] --login
```

`[target-name]` refers to the IQN name of the target. `[portal]` refers to the iSCSI Target host address and port like '127.0.0.1:3260'. If only a host address is specified the port defaults to 3260.

You can find the targets advertising on a particular IP address by issuing the command:

```
iscsiadm -m discovery -t sendtargets -p [portal]
```

Where `[portal]` is the portal name of the target which is composed of an IP address and a port in the form IP:port.

To log out of the target you need to type a command similar to the one for logging in except tat you need to replace the **--login** option with **--logout**.

Partitioning the Device

Use a partition manipulation program like fdisk or parted to partition the device created then the initiator logs into the target. This device could be something like `/dev/sdb` or `/dev/hdb` depending on your environment. With **fdisk**, you would type **fdisk /dev/sdb** and follow the interactive instructions to partition a device like `/dev/sdb`. Instructions to use **fdisk** can be found on its man page.

You can find the particular device created when you logged in the iSCSI initiator to the target by typing the following command:

```
iscsiadm -m session -i
```

The device name will appear under the 'Attached SCSI devices' section.

Creating an filesystem on the disk partition(s)

Use a filesystem creation tool like **mke2fs** to create an ext2 or ext3 filesystem on the partition created above, for example, by typing **mke2fs /dev/sdb1** where `/dev/sdb1` is the partition created after the previous step to create an ext2 partition.

Mounting the filesystem

Use the **mount** command to mount a mount point to the filesystem created above: **mount /dev/sdb1 [mount-point]**', where [mount-point] is a mount point of your choice. Unmount your partition with **umount**.

Using the Grid

You can use the mounted filesystem as you would any other, for example by copying files to it with **cp**, removing them with **rm**, etc.

Notes

1. http://www.open-iscsi.org/