

**100 NEW
HACKS**

LINUX SERVER HACKS™

Volume Two

*Tips & Tools for
Connecting, Monitoring,
and Troubleshooting*



O'REILLY®

*William von Hagen
& Brian K. Jones*

HACK
#48

Create a Copy-on-Write Snapshot of an LVM Volume

Logical volumes don't just provide a great way to supply flexible storage—they can also provide a great way to preserve files that have changed recently, simplifying restores and reducing restore requests.

A snapshot is a copy of a logical volume that reflects the contents of that logical volume when the snapshot was created. With a *copy-on-write snapshot*, each time a file changes in the original volume, the contents of the original file (as of the time that the snapshot was made) are preserved in the snapshot volume. In other words, the complete contents of the original file are copied to the snapshot volume when you write changes to the file in the original volume. Implementing a copy-on-write volume to track changed files is like having a built-in backup mechanism, because it provides you with a point-in-time copy of the filesystem that is contained on your logical volume. This copy of your filesystem can then be used for retrieving files that have accidentally been deleted or modified. For system administrators, copy-on-write snapshots can be particularly useful in preserving the original copies of system configuration files (just in case you ever make a mistake). However, their real beauty is in preserving copies of volumes containing users' home directories. I've found that taking a nightly snapshot of the logical volume that contains the users' home directories and automatically mounting it enables most users to satisfy their own restore requests by simply retrieving the original copies of lost or incorrectly modified files from the snapshot. This makes them happier and also lightens my workload. Not a bad combination!

This hack explains how to create a snapshot of an existing volume and mount it, and provides some examples of how the snapshot preserves your original files when they are modified in the parent volume.

Kernel Support for Snapshots

Snapshots of logical volumes are created and maintained with the help of the *dm_snapshot* filesystem driver. This is built as a loadable kernel module on most modern Linux distributions. If you cannot find this module or snapshots simply do not work on your system, `cd` to your kernel source directory (typically */usr/src/linux*) and check your kernel configuration file to make sure this module is either built in or available as a kernel module, as in the following example:

```
$ cd /usr/src/linux
$ grep -i DM-SNAPSHOT .config
CONFIG_SM_SNAPSHOT=m
```

Create a Copy-on-Write Snapshot of an LVM Volume

In this case, the *dm-snapshot* driver is available as a loadable kernel module. If the value of the `CONFIG_DM_SNAPSHOT` configuration variable is *n*, this option is not available in your kernel. You will have to rebuild your kernel with this driver built in (a value of *y*) or as a loadable kernel module (a value of *m*) in order to take advantage of logical volume snapshots as discussed in this hack.



Even if the *dm-snapshot* module is available on your system, you may need to manually load it using the standard `modprobe` command, as in the following example:

```
# modprobe dm_snapshot
```

Creating a Snapshot

This section explains how to create a snapshot of an existing filesystem. The filesystem that you are taking a snapshot of must reside on a logical volume, as shown by the presence of the device mapper directory in the following example:

```
# df -Ph /test
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/testvg-testvol 485M   18M 442M   4% /test
```

Next we'll use the `dd` command to create a few sample files in the test volume for use in testing later in this hack:

```
# dd if=/dev/zero of=/test/5M bs=1048576 count=5
5+0 records in
5+0 records out
# dd if=/dev/zero of=/test/10M bs=1048576 count=10
10+0 records in
10+0 records out
```

To create a snapshot of the *testvol* volume, execute a command like the following:

```
# lvcreate -s -L 100M -n testsnap /dev/testvg/testvol
Logical volume "testsnap" created
```

In this example, I allocated 100 MB for the snapshot. This means that we can make 100 MB in changes to the original volume before the snapshot is full. Snapshots eventually fill up because they are preserving old data, and there is no way to purge the files that it has preserved because it is a snapshot of another volume, not an original logical volume itself. Once a snapshot is 100% used, it becomes useless—you must remove it and create a new snapshot.

To confirm that the snapshot was created correctly, use the `lvs` command to display logical volume status information:

```
# lvs
LV      VG      Attr  LSize  Origin Snap%  Move Copy%
testsnap testvg swi-a- 100.00M testvol 0.02
```

```
testvol testvg owi-ao 500.00M
```

Mounting a Snapshot

Having a snapshot of a logical volume is fairly useless unless you enable people to access it. To mount the sample *testsnap* snapshot, use a standard mount command such as the following:

```
# mount /dev/testvg/testsnap /testsnap
# df -Ph /test*
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/testvg-testvol 485M   18M  442M   4% /test
/dev/mapper/testvg-testsnap 485M   18M  442M   4% /testsnap
```



Note that a snapshot volume always lives in the same volume group as the logical volume of which it is a copy.

Just to be sure, you can use the `ls` command to verify that both the snapshot and the original volume are available:

```
# ls -l /test
total 15436
-rw-r--r-- 1 root root 10485760 Apr 21 23:48 10M
-rw-r--r-- 1 root root 5242880 Apr 21 23:48 5M
drwx----- 2 root root 12288 Apr 21 23:15 lost+found

# ls -l /testsnap/
total 15436
-rw-r--r-- 1 root root 10485760 Apr 21 23:48 10M
-rw-r--r-- 1 root root 5242880 Apr 21 23:48 5M
drwx----- 2 root root 12288 Apr 21 23:15 lost+found
```

Now, create a 50-MB file in the */test* filesystem and examine what happens to the */testsnap* filesystem and the snapshot usage (using our favorite `lvs` command):

```
# dd if=/dev/zero of=/test/50M bs=1048576 count=50
50+0 records in
50+0 records out
# df -Ph /test*
Filesystem                Size  Used Avail Use% Mounted on
/dev/mapper/testvg-testvol 485M   68M  392M  15% /test
/dev/mapper/testvg-testsnap 485M   18M  442M   4% /testsnap
# ls -l /test
total 66838
-rw-r--r-- 1 root root 10485760 Apr 21 23:48 10M
-rw-r--r-- 1 root root 52428800 Apr 22 00:09 50M
-rw-r--r-- 1 root root 5242880 Apr 21 23:48 5M
drwx----- 2 root root 12288 Apr 21 23:15 lost+found
# ls -l /testsnap/
total 15436
```

Create a Copy-on-Write Snapshot of an LVM Volume

```
-rw-r--r-- 1 root root 10485760 Apr 21 23:48 10M
-rw-r--r-- 1 root root 5242880 Apr 21 23:48 5M
drwx----- 2 root root 12288 Apr 21 23:15 lost+found
# lvs
LV      VG      Attr  LSize  Origin Snap%  Move Copy%
testsnap testvg swi-ao 100.00M testvol 50.43
testvol testvg owi-ao 500.00M
```

Notice that the 50-MB file does not immediately show up in */testsnap*, but some of the snapshot space has been used up (50.43%).

Next, simulate a user accidentally removing a file by removing */test/10M* and examine the results:

```
# rm /test/10M
rm: remove regular file `/test/10M'? y
# df -Ph /test*
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/testvg-testvol 485M   58M 402M  13% /test
/dev/mapper/testvg-testsnap 485M   18M 442M   4% /testsnap
```

Note that disk space utilization in your snapshot increased slightly:

```
# lvs
LV      VG      Attr  LSize  Origin Snap%  Move Copy%
testsnap testvg swi-ao 100.00M testvol 50.44
testvol testvg owi-ao 500.00M
```



When using the `lvs` command after significant file operations, you may need to wait a few minutes for the data that `lvs` uses to be updated.

If you now need to recover the file *10M*, you can get it back by simply copying it out of the snapshot (to somewhere safe). Say goodbye to most of your restore headaches!

Remember, once the snapshot is 100% full, its contents can no longer be relied upon, because no new files can be written to it and it is therefore no longer useful for tracking recent updates to its parent volume. You should monitor the size of your snapshots and recreate them as needed. I find that recreating them once a week and remounting them keeps them up to date and also usually prevents “snapshot overflow.”

See Also

- Snapshot section of the LVM HOWTO: http://www.tldp.org/HOWTO/LVM-HOWTO/snapshots_backup.html
- “Create Flexible Storage with LVM” [Hack #46]

- “Combine LVM and Software RAID” [Hack #47]

—*Lance Tost*