

13.4. Taking a Backup Using Snapshots

Following on from the previous example we now want to use the extra space in the "ops" volume group to make a database backup every evening. To ensure that the data that goes onto the tape is consistent we use an LVM snapshot logical volume.

A snapshot volume is a special type of volume that presents all the data that was in the volume at the time the snapshot was created. For a more detailed description, see [Section 3.8](#), Snapshots. This means we can back up that volume without having to worry about data being changed while the backup is going on, and we don't have to take the database volume offline while the backup is taking place.



In LVM1, this type of volume was read-only, but LVM2 creates read/write snapshots by default.

13.4.1. Create the snapshot volume

There is a little over 500 Megabytes of free space in the "ops" volume group, so we will use all of it to allocate space for the snapshot logical volume. A snapshot volume can be as large or as small as you like but it must be large enough to hold all the changes that are likely to happen to the original volume during the lifetime of the snapshot. So here, allowing 500 megabytes of changes to the database volume which should be plenty.

```
# lvcreate -L592M -s -n dbbackup /dev/ops/databases
lvcreate -- WARNING: the snapshot must be disabled if it gets full
lvcreate -- INFO: using default snapshot chunk size of 64 KB for "/dev/ops/dbbackup"
lvcreate -- doing automatic backup of "ops"
lvcreate -- logical volume "/dev/ops/dbbackup" successfully created
```



Full snapshot are automatically disabled

If the snapshot logical volume becomes full it will be dropped (become unusable) so it is vitally important to allocate enough space. The amount of space necessary is dependent on the usage of the snapshot, so there is no set recipe to follow for this. If the snapshot size equals the origin size, it will never overflow.

13.4.2. Mount the snapshot volume

We can now create a mount-point and mount the volume

```
# mkdir /mnt/ops/dbbackup
# mount /dev/ops/dbbackup /mnt/ops/dbbackup
mount: block device /dev/ops/dbbackup is write-protected, mounting read-only
```

If you are using XFS as the filesystem you will need to add the `nouuid` option to the mount command:

```
# mount /dev/ops/dbbackup /mnt/ops/dbbackup -onouuid,ro
```

13.4.3. Do the backup

I assume you will have a more sophisticated backup strategy than this!

```
# tar -cf /dev/rmt0 /mnt/ops/dbbackup
tar: Removing leading `/' from member names
```

13.4.4. Remove the snapshot

When the backup has finished you can now unmount the volume and remove it from the system. You should remove snapshot volume when you have finished with them because they take a copy of all data written to the original volume and this can hurt performance.

```
# umount /mnt/ops/dbbackup
# lvremove /dev/ops/dbbackup
lvremove -- do you really want to remove "/dev/ops/dbbackup"? [y/n]: y
lvremove -- doing automatic backup of volume group "ops"
           lvremove -- logical volume "/dev/ops/dbbackup" successfully removed
```

[Prev](#)

Add a new disk to a multi-disk SCSI system

[Home](#)
[Up](#)

[Next](#)

Removing an Old Disk