

Ubuntu LVM Guide - Part 2 (Snapshots)

In this LVM Guide, we'll show you how to use lvm snapshots for backups and how to revert a logical volume to a previous state e.g. fix your root filesystem if some "experiment" has broken your system.

If you're not familiar with logical volume manager (lvm) software in Ubuntu/Linux, you may want to take a look at [LVM Guide -Part 1](#). All the steps in the sections below have been tested on Ubuntu 12.10 (quantal quetzal) - the same principles should apply to any linux distribution which uses lvm2.

LVM Snapshots

The logical volume manager software allows you to take a **snapshot** of any logical volume.

This can be used to bring your logical volume back to a known good state or to merge changes from a modified snapshot back to the original logical volume. The latter is possible because lvm snapshots are also writeable (as of lvm2, this is not the case for the older lvm1).

You can use snapshots as a form of backup which allows you to recover lost data or recover from a dodgy system following some experiment.

For example before you try some experiment such as tweaking important system configuration files or installing new software, you could take a snapshot of your good system before trying something out. **If things don't work as you planned, you can revert to a good working system by using the snapshot.**

Note: its important to mention that if a snapshot runs out of space, the whole snapshot is dropped and cannot be used. However if you notice the

snapshot is running out of space (check this using the `lvdisplay` or `lvs` commands), you can increase its size just like any other logical volume.

Taking A Snapshot

When you take a snapshot, use the `lvcreate` command along with the `"-s"` or `--snapshot` option as shown below and remember not to use any `"-"` (i.e. dash/hyphen) in the name as software like `grub2` will have trouble using it:

```
sudo lvcreate --size 1G -s -n snapshot_lv /dev/vgroup/origin
```

So the above would create a logical volume called `snapshot_lv` which is a snapshot of `original_lv` (in the volume group called `vgroup`). Note that the size you need to use will depend on what you're doing. Bear in mind that snapshots use "copy on write", so data is only copied to the snapshot when it deviates from the original data. So when you access data in the snapshot volume, you effectively access the original logical volume's data unless: (a) the original volume's data has changed, in which case you access a copy of the original which is stored in the snapshot volume, (b) you have written to the snapshot, in which case the modified data resides in the snapshot volume. Thus, you can create a snapshot of say, 100GiB using only a few GiB, because most of the "snapshot" data still resides on the original volume.

Mounting A Snapshot

You can mount a snapshot which allows a limited form of experimentation, e.g. you can run software on files there without damaging the original files. If you're happy with the outcome, then you can run the software using your original files. Or you could just merge that snapshot if you'd like to change your originals to the state of those in the snapshot (instructions to merge are shown in next section).

To mount a snapshot, create a directory for the snapshot in /mnt for example:

```
sudo mkdir /mnt/snap
```

Then for a snapshot called "mysnap", you could mount it using:

```
sudo mount /dev/ubuntu/mysnap /mnt/snap
```

Then cd to the directory to access the snapshot's files:

```
cd /mnt/snap
```

Reverting/Merging To The Snapshot's State

Once you have a snapshot available, you can change the original logical volume to the state of that snapshot should the need arise.

Before changing the state of the logical volume to that of it's snapshot, you need to have unmounted that logical volume if you want the change to happen immediately. Alternatively, if you can't unmount, you can carry out the command but the changes will not take effect until you do a reboot. In order to revert to the snapshot, you would simply use the **lvconvert** command along with the **--merge** option, as shown below:

```
lvconvert --merge /dev/vgroup/snapshot-lv
```

Now the original logical volume has been reverted/merged to the state of the snapshot (remember this will probably require a reboot if one of the volumes is mounted).

Note that once you've used a snapshot to revert/merge, the snapshot disappears from your system.

Removing A Snapshot

If you no longer have use for a snapshot, you can remove it using the **lvremove** command:

```
sudo lvremove /dev/vgroup/snapshot_lv_name
```

Where **vgroup** is the volume group in which the snapshot resides, and **snapshot_lv_name** is the name of your snapshot logical volume.

A Simple Example

As an example, let's create a snapshot of the logical volume (**music_lv**) which we created for our music in [LVM Guide - Part 1](http://www.tutronics.com/2012/12/lvm-guide-part-1) (you could do this before trying out some new media player software, just in case your music

files get corrupted or deleted somehow by the new software due to a bug). In this case, let's say we have 10GiB of music:

```
sudo lvcreate --size 2G -s -n music_lv_snapshot /dev/vgroup/
```

The 2GiB of data means you're covered if a few of your files get deleted. If instead you used 10GiB as the snapshot size, you can rest assured that even if every file was accidentally deleted, the snapshot can hold enough data to recover it all. If the new software did actually mess up your music files, you can revert your files to their original state when you took the snapshot by using the command:

```
lvconvert --merge /dev/ubuntu/music_lv_snapshot
```

So using snapshots can give you more flexibility for your backups and snapshot size is based on the risk assessment of how your data may change.

Creating A "Complete Backup" From A Snapshot

Snapshots only grow if data on the original logical volume changes or if you write to the snapshot yourself (more about the write aspect later). So if we forget about us manually writing to snapshots for the moment, the only data in the snapshot are chunks of original volume data which have changed on the original volume since the snapshot was taken. Have you noticed that a snapshot still relies on the unchanged original data being OK?

In other words, you can only use a snapshot to revert an original logical volume if the original still exists. So snapshots are a fantastic feature as you'll see below, but they are not backups in the conventional sense of having a complete copy of all the original data.

If you want a completely independent separate copy of your data at the time of a snapshot, you can take a copy in the same way you would for any other device, e.g. using the `cp` or `dd` commands.

Note that this way of backing up data ensures a **consistent view of the data** is quickly taken at a specific point in time. This is important when backing up databases for example. Obviously creation of a consistent view requires tables to be locked, but the beauty of lvm is that they need only be locked for a very small amount of time (which makes it viable when

compared to locking tables and waiting hours for data to be physically copied).

Preparing For Recovery Before Experimenting/Tweaking/Trying Something New

Have you ever wrecked your system by tweaking things, or adding software that didn't quite do what you expected, or by removing something important by mistake? Wouldn't it be nice to be able to undo the mess and return your system to its former working state?

Well using a snapshot you can return to that good state - that is, only if you took a snapshot before you started your experimentation!

There are two ways to do this.

Method 1: The first method is to change your current system and revert to the snapshot if something goes wrong.

Method 2: Alternatively you can boot using the snapshot as your root logical volume, make the changes to the snapshot, then if something goes wrong just reboot and remove the snapshot, or if you would like to keep the changes just merge the snapshot.

Using either approach, **you'll need take a snapshot of the system in a known good state before you start any tweaks** or try out new stuff (and if you're changing the kernel, the very first thing you should do is back up /boot if your install has /boot on a normal /dev/sda1 partition which is not part of lvm, more about this below).

Recovery Method 1: Trying Out New Stuff On Your Root Logical Volume (Your Normal System)

Using this approach, you take a snapshot of the root file system's logical volume then carry out your tweaks on the root logical volume. If there is a mistake or problem you revert to the snapshot which has the root volume's (thus the root filesystem's) original state.

Now is the time to back up /boot if you are going to change the kernel (see if this applies to you by reading the "Backing Up /boot" section, below).

Ok, having backed up /boot (only if you needed to), now we can get on with taking the snapshot.

In Ubuntu 12.10, the root file system resides on the logical volume /dev/ubuntu/root (also /dev/mapper/ubuntu-root).

So, based on this you could create a snapshot of this logical volume, using:

```
sudo lvcreate --size 2G -s -n rootsnap /dev/ubuntu/root
```

You should base the size on whatever job you are doing, unfortunately there is no rule-of-thumb for calculating this size. Think about how much data will change, e.g. if it's only for a few config file changes or for trying out a new graphics driver, 2GiB is overkill. But if you plan on changing a lot of software, take the size of the software into account. If you have the space, it's recommended to be generous with the size of your snapshot - in any case, once you've finished the snapshot will be removed anyway thus relinquishing the space to your system.

Once you have taken your snapshot, crack on with trying out some experimentation or new software config etc ...

If you are happy with the changes, you just need to remove the snapshot (see "Removing A Snapshot" section above) and you're done.

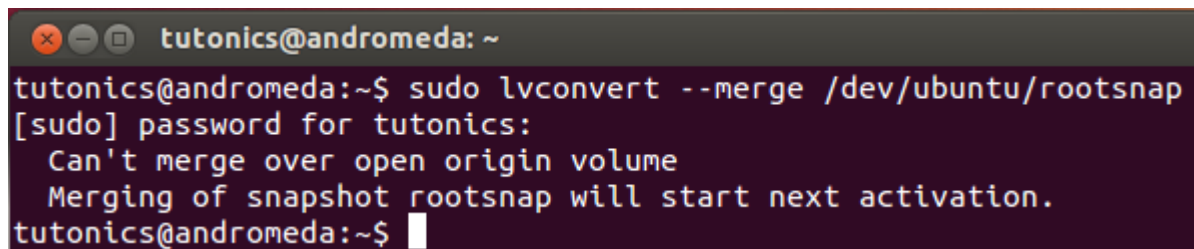
Otherwise, if this new stuff messes up your system, the steps required to fix things will depend on how badly things went...

If the system boots up OK, you can run the `lvconvert` command below while the root filesystem is mounted, a reboot will then be required before the logical volume is actually reverted to the snapshot.

```
sudo lvconvert --merge /dev/ubuntu/rootsnap
```

If you can't boot you should start up a live cd or live usb then run the above `lvconvert` command.

When you run the `lvconvert` command you should see something like this:



```
tutonics@andromeda: ~  
tutonics@andromeda:~$ sudo lvconvert --merge /dev/ubuntu/rootsnap  
[sudo] password for tutonics:  
Can't merge over open origin volume  
Merging of snapshot rootsnap will start next activation.  
tutonics@andromeda:~$
```

Then reboot.

If you still can't boot, (this may happen if `/boot` has been changed somehow!) for example if you've changed to a different kernel as part of what you tried since taking the snapshot (In which case you'll see something like "An error occurred while mounting `/boot`"), then you'll need to fix `/boot`.

To fix `/boot`: follow the "Restoring `/boot`" section below. When you've restored `/boot`, you just need to reboot normally - now things will be right back as they were before you started your experiments!

Quick Summary:

- (1) Backup `/boot` if required. (Follow steps in "Backing up `/boot`" section below) .
- (2) Take a snapshot of your current root logical volume
- (3) Carry out your tweaks
- (4) If you're not happy with the changes, revert to snapshot using `lvconvert`. Otherwise if you are happy with the changes, remove the snapshot as you will not need it.
- (5) If still can't boot because `/boot` got messed up, follow steps in "Restoring `/boot`" section below.

Recovery Method 2: Trying Out New Stuff On

The Snapshot Instead

By default lvm2 snapshots are read/write, meaning that if you mount the snapshot and change things, when a merge is done, the original will be changed to the snapshot's modified state.

For some people, using the snapshot's write functionality in this way may conceptually seem more suitable for trying out tweaks or they may feel more comfortable doing things this way (although both approaches work equally well).

OK, lets start by saying that **now is the time to back up /boot** if you are going to change the kernel (see if this applies to you by reading the "Backing Up /boot" section, below).

Next, create a snapshot of the root logical volume as before:

```
sudo lvcreate --size 2G -s -n rootsnap /dev/ubuntu/root
```

Now boot using the snapshot as your root filesystem. The easiest way to do this is to hold **Shift** while your system starts up. This will bring up the grub menu, when you can see the menu you should hit **e** to edit. This will bring up some grub script text which you can edit on the fly.

Move to the line starting with **linux** (Navigate using the cursor arrows) and change:

```
root=/dev/mapper/ubuntu-root
```

to use your snapshot name, e.g:

```
root=/dev/mapper/ubuntu-rootsnap
```

When you've typed that, simply hit **F10** to boot.

You should now boot into your snapshot.

(Note that this is only a temporary change that you'll need to carry out each time you want to boot into your snapshot. If you want something more permanent, you could add an entry to grub2 menu - but snapshots are usually temporary entities, so a menu entry would normally be overkill, let us know if you'd like us to include instructions about how to add a permanent entry!).

You can verify that you're using the snapshot as your root filesystem by running:

```
df .
```

If you used "rootsnap" for your snapshot name, you'll see something like this:

```
tutonics@andromeda:~$ df .  
Filesystem            1K-blocks      Used Available Use% Mounted on  
/dev/mapper/ubuntu-rootsnap 552935480 10231372 515436872   2% /
```

You should see your snapshot name in the result.
Now you can proceed with your tweaks and experimentation.
When you're finished your tweaks, if you decide you like what you've done, just run:

```
sudo lvconvert --merge /dev/ubuntu/rootsnap
```

You can run the above command from either the snapshot system or your normal system. If you run it from the snapshot, you'll see something like this:

```
tutonics@andromeda: ~  
tutonics@andromeda:~$ sudo lvconvert --merge /dev/ubuntu/rootsnap  
Can't merge when snapshot is open  
Merging of snapshot rootsnap will start next activation.  
tutonics@andromeda:~$
```

Now when you reboot, you'll have the changes on your root logical volume (i.e. your normal system will have the tweaks).

Otherwise, if your changes messed up the snapshot, you just need to boot normally (don't run lvconvert command) then remove the snapshot using:

```
sudo lvremove /dev/ubuntu/rootsnap
```

If things got messed up, and your changes involved changing the kernel, you'll probably need to follow the "Restoring /boot" section below.

Quick Summary:

- (1) Backup /boot if required. (Follow steps in "Backing up /boot" section below).
- (2) Take a snapshot of your current root logical volume
- (3) Boot using your snapshot as the root filesystem (see above).
- (4) Carry out your tweaks
- (5) If you are happy with the changes, merge to snapshot using `lvconvert`
- (6) If still can't boot because /boot got messed up, follow steps in "Restoring /boot" section below.

Renaming Logical Volumes (and thus Snapshots)

If you want to rename a snapshot, simply use the `lvrename` command. This may be required for example if you used a logical volume name with a "-" in it when creating your snapshot.

If you ever wanted to boot up via grub2 into a root file system snapshot who's named like that, such as "root-snap", you could change the name to something useable like "rootsnap" by using this command:


```
sudo lvrename /dev/ubuntu/root-snap ubuntu/rootsnap
```

Where ubuntu is the name of the volume group.

Backing Up /boot

If your /boot is on its own partition, e.g. /dev/sda1 (as is the case for a default Ubuntu 12.10 lvm install), you'll want to back that up also - **only if what you want to try out involves a kernel change, or any other change to /boot**, otherwise there is no need to back up /boot.

To take a backup of /boot, run the following command **before taking the snapshot** of your root filesystem (so you'll have a copy on snapshot also):

```
dd if=/dev/sda1 of=boot_backup.img
```

Do take note of whereabouts in your root filesystem you've put boot_backup.img because if you have to resort to using it later you'll need to know where to find it!

Restoring /boot

To fix /boot using boot_backup.img (created earlier): start up a live cd, or live usb, then mount your root filesystem, e.g.:

```
sudo mkdir /mnt/rtfs
```

```
sudo mount /dev/ubuntu/root /mnt/rtfs
```

Locate your boot_backup.img, then revert to the original /boot by running the below command (use your path to boot_backup.img, and take care to use the correct device!):

```
sudo dd if=/mnt/rtfs/boot_backup.img of=/dev/sda1
```

When this command has finished, your /boot partition has been restored.

Still Can't Boot?

If you still cannot recover your system (e.g. you changed the kernel, but forgot to back up /boot), you can try holding **Shift** while your system restarts. This will throw up your grub menu from which you should be able to boot to an older kernel. Once up and running, you can use the software updater (or similar) to update your kernel to its former self.

What's next?

In our next LVM guide, we'll be covering lvm mirrors.

Thanks to the programmers of LVM2 and also the authors of the [LVM-HOWTO](#)

Thank you for reading this article. Please share if you liked it.



**MOVE OUT OF YOUR DATA CENTER
WITHOUT PACKING A BOX.**

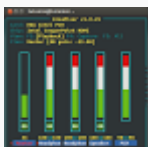
[SEE HOW >](#)

vmware

You might also like:



How to Change Update
Check Frequency &
Automatically Install
Fix No Sound / Turn Off
Security Updates
Auto Mute in Ubuntu



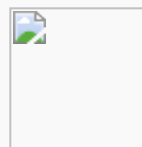
12.10



Ubuntu LVM Guide -
Part 1



Preparing VirtualBox
for Virtual OS



Installations
How to find out if your
Processor is 64-bit or
32-bit



Enable Ubuntu
Restricted Extras &

© 2017 Tutonics