Clone Linux Drive

If a hard-drive is to be cloned to a new drive (for example larger HDD to SSD or replacing system) below are steps that could be useful.

Firstly you will need a live USB stick, build this or use an existing stick - this will be used throughout.

Install the new hard drive and build it as a new machine as normal.

Once complete, boot the PC to the live environment with only the old hard drive installed.

Both the new and the old hard drive will have the same volume group name, so we will need to rename the old hard volume group.

Once booted, open a terminal and use the vgscan command to return the volume group name - at time of writing the volume group name used is vgmint.

```
~# vgscan
  Reading volume groups from cache.
  Found volume group "vgmint" using metadata type lvm2
~#
```

Next use the vgrename command to rename the volume group to something different, for example old_hard_drive.

```
vgrename <old_vg_name> <new_vg_name>
```

Power off the PC and re-install the new hard-drive, both new and old should be installed, boot into the live environment.

Make 2 new directories that we will use to mnt the drives too:

```
mkdir /mnt/oldHardDrive
mkdir /mnt/newHardDrive
```

Use vgscan again to confirm the names of the 2 volume groups, we will then mount those to the folders we have created:

Activate the volume groups:

```
vgchange -ay
```

Then use lvdisplay to show the logical volume paths:

```
lvdisplay
```

```
mount <old_drive_vg_root_path> /mnt/oldHardDrive
mount <new_drive_vg_root_path> /mnt/newHardDrive
```

We'll now be in a position to rsync the content of the old drive to the new drive:

```
rsync -avHX /mnt/oldHardDrive/ /mnt/newHardDrive/
```

options: a- archive mode v- increase verbosity H- preserve hard links X- preserve extended attributes

It may be sensible to use the –exclude option to exclude any directories you do not wish to copy/overwrite.

Once the copy is complete, we need to re-install the grub bootloader on new drive.

First, bind mount some system directories needed by grub, then chroot:

```
umount /mnt/oldHardDrive
mount --bind /proc /mnt/newHardDrive/proc
mount --bind /sys /mnt/newHardDrive/sys
mount --bind /dev /mnt/newHardDrive/dev
mount --bind /run /mnt/newHardDrive/run
chroot /mnt/newHardDrive

#in case of separate boot partition, also mount it to /boot
#in case of EFI system, also identify and mount EFI partition into /boot/efi
fdisk -l /dev/sda | grep EFI
# example output:
# /dev/sda1          2048     309247    307200   150M EFI System
mount /dev/sda1 /mnt/boot/efi
```

Then install grub in the Master Boot Record of your hard drive, and/or update grub config file (with the new uuids…):

```
#update SWAP UUID in /etc/initramfs-tools/conf.d/resume
update-initramfs -u -k all

#if grub not already present
grub-install /dev/sda
#then
update-grub
```

the last thing to do is to confirm /etc/fstab is pointing to the correct label, as root, edit /etc/fstab ensure the root and swap partitions are pointing to the correct label or UUID, the following commands may be useful in identifying:

```
ls -l /dev/disk/by-uuid/
blkid /dev/sdaX
vgscan
```

Three 2-4 partition entries required for root, boot, efi and swap e.g.:

```
UUID=36a059df-ab43-4e88-a069-5cafbf5c177d / ext4 errors=remount-ro 0 1
UUID=61C5-B375  /boot/efi vfat umask=0077 0 1
```

```
/dev/mapper/vgmint-swap_1 none          swap     sw            0         0
```

Now shutdown the PC, remove the old drive and live USB stick, boot to the new drive.

It is worth noting that as the Ethernet device name has changed we need to:

1. Select the correct Ethernet adapter in the virtual-box configuration.
2. Confirm that the mount points have applied, this may require a restart.

Live remote cloning

Target system:

```
sudo apt-get install openssh-server
sudo sed -i 's/#PermitRootLogin/PermitRootLogin/' /etc/ssh/sshd_config
sudo service ssh restart
```

On source system:

```
sudo su -
apt-get install sshfs rsync
mkdir /mnt/newsystem
mount -t fuse.sshfs root@IP:/ /mnt/newsystem
cp /mnt/newsystem/etc/fstab /root/newfstab
rsync -ahPHAXx --delete --
exclude={/dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media/*,/lost+found} /
/mnt/newsystem

#or for overall bandwidth stats and brief output:
rsync -ahPHAXx --info=progress2 --info=name0 --delete --
exclude={/dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media/*,/lost+found} /
/mnt/newsystem
```

Adjust /mnt/newsystem/etc/fstab file accordingly

reset GRUB:

```
for i in /dev /dev/pts /proc /sys /run; do sudo mount -B $i
/mnt/newsystem$i; done
sudo chroot /mnt/newsystem
sudo grub-install --recheck /dev/sdX
sudo update-grub
```

From:
<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:
**http://wuff.dyndns.org/doku.php?id=linux:clone-linux-drive&rev=1756859
945**

Last update: **2025/09/03 01:39**