

Initializing a New Hard Disk on Linux

(STEP-BY-STEP GUIDE FOR BEGINNERS AND TIER 1 TECHNICIANS)

TEBOGO MATSEDING

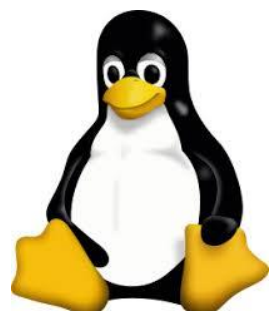


Table of Contents

Introduction

Create new hard disk

Identify hard disk

Create a New Partition (n)

Create a New Partition (p)

Create a New Partition (w)

Verify partition was created

Format the Partition

Mount the Drive Temporarily

Verify it is mounted

Configure Persistent Mounting

(Get the partition' s UUID)

Configure Persistent Mounting

(Edit /etc/fstab)

Configure Persistent Mounting

(add the UUID and path)

Configure Persistent Mounting

(Test with sudo mount -a)

Check Permission

Set Ownership && Set Permissions

Verify it blocks access

Verify user has permission

Create a shortcut for the user

Introduction

This guide explains how to set up a brand-new hard drive (HDD) in Linux so it can be used to store files, documents, and folders.

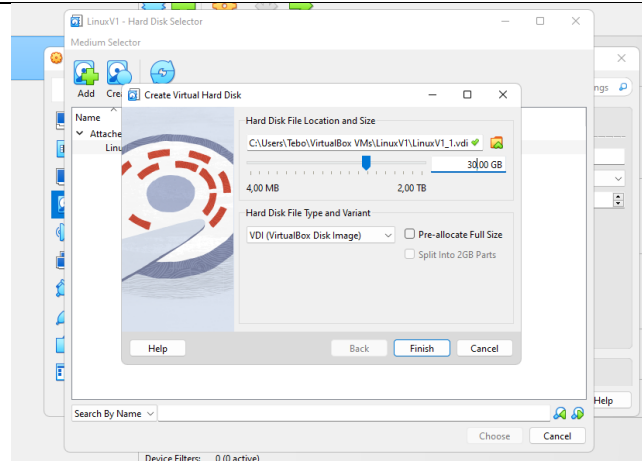
You'll learn how to identify the new drive, create a partition, format it, mount it, and make it easy for a user to access from the desktop.

This is ideal for demonstrations or new installations where a user needs quick, organized access to extra storage.

Create new hard disk

If you're using a virtual machine, add a new virtual hard disk in your VM settings.

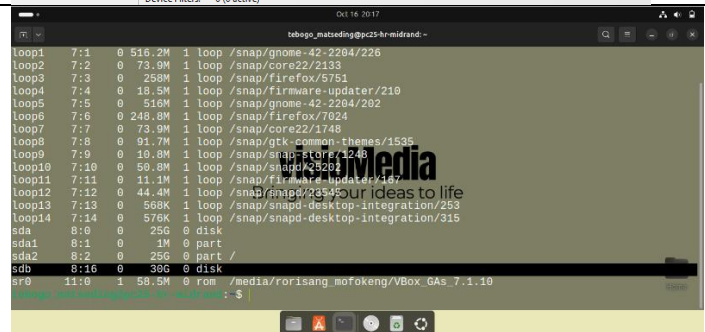
If it's a physical HDD, make sure it's properly connected and recognized by the system before booting into Linux.



Identify hard disk

Open the terminal and run:
`lsblk -l`

This command lists all disks on your system. The new hard drive usually appears as `/dev/sdb` or `/dev/sdc` and will not have any partitions listed under it.



Create a New Partition (n)

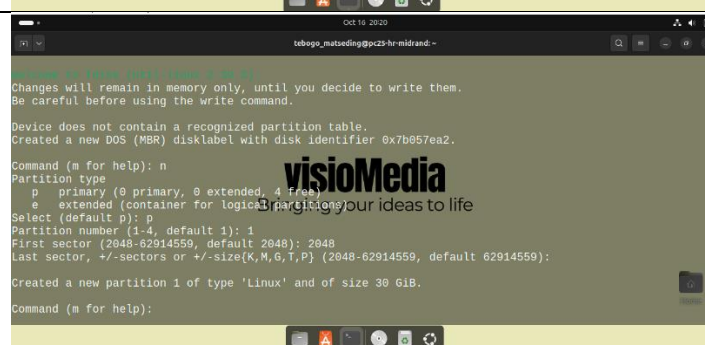
Start the partitioning tool:
`sudo fdisk /dev/sdb`

Then type `n` to create a new partition.



Create a New Partition (p)

When prompted, type `p` to make it a primary partition. Press `Enter` through the defaults unless you need a specific size.



Create a New Partition (w)
Once done, type w to write the changes to the disk and exit fdisk. You'll see a confirmation message saying the partition table has been updated.

```
Oct 16 2021
tebogo_matseding@pc25-hr-midrand:~$ fdisk /dev/sdb
Created a new partition 1 of type 'Linux' and of size 30 GiB.

Command (m for help): p
Disk /dev/sdb: 30 GiB, 32212254720 bytes, 62914560 sectors
Disk model: VBox HardDisk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x7b057ea2

Device Boot Start End Sectors Size Id Type
/dev/sdb1 2048 62914559 62912512 30G 83 Linux

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

tebogo_matseding@pc25-hr-midrand:~$
```

Verify partition was created
Run:
Lsblk -l
You should now see something like /dev/sdb1 that's your new partition.

```
Oct 16 2022
tebogo_matseding@pc25-hr-midrand:~$ lsblk -l
loop2 7:2 0 73.9M 1 loop /snap/core22/2133
loop3 7:3 0 258M 1 loop /snap/firefox/5751
loop4 7:4 0 18.5M 1 loop /snap/firmware-updater/210
loop5 7:5 0 516M 1 loop /snap/gnome-42-2204/202
loop6 7:6 0 248.8M 1 loop /snap/firefox/7024
loop7 7:7 0 73.9M 1 loop /snap/core22/1748
loop8 7:8 0 91.7M 1 loop /snap/gtk-common-themes/1535
loop9 7:9 0 10.8M 1 loop /snap/snap-store/1248
loop10 7:10 0 50.8M 1 loop /snap/snapd/2322
loop11 7:11 0 11.1M 1 loop /snap/firmware-updater/147
loop12 7:12 0 44.4M 1 loop /snap/snapd/2264
loop13 7:13 0 568K 1 loop /snap/snap-desktop-integration/453
loop14 7:14 0 576K 1 loop /snap/snap-desktop-integration/315
sda 8:0 0 25G 0 disk
sda1 8:1 0 1M 0 part
sda2 8:2 0 25G 0 part
sdb 8:16 0 30G 0 disk
sdb1 8:17 0 30G 0 part
sr0 11:0 1 50.5M 0 rom /media/rorisang_mofokeng/VBox_GAs_7.1.10

tebogo_matseding@pc25-hr-midrand:~$
```

Format the Partition
Format it with the EXT4 filesystem:
sudo mkfs.ext4 /dev/sdb1
This prepares the drive for storing data.

```
Oct 16 2024
tebogo_matseding@pc25-hr-midrand:~$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 7864064 4k blocks and 1966080 inodes
Filesystem UUID: c6e40886-f422-441e-8eaf-e226c34907ed
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

tebogo_matseding@pc25-hr-midrand:~$
```

Mount the Drive Temporarily
Make a folder where the drive will be mounted:
sudo mkdir /mnt/newhdd
Then mount it:
sudo mount /dev/sdb1 /mnt/newhdd

```
Oct 16 2025
tebogo_matseding@pc25-hr-midrand:~$ sudo mkdir /mnt/rorisang_new_disk
mkdir: cannot create directory '/mnt/rorisang_new_disk': File exists
tebogo_matseding@pc25-hr-midrand:~$ sudo mount /dev/sdb1 /mnt/rorisang_new_disk

tebogo_matseding@pc25-hr-midrand:~$
```

Verify it is mounted
Check if the drive is mounted:
df -h
If you see /mnt/newhdd listed, it's mounted and ready for use.

```
Oct 16 2029
tebogo_matseding@pc25-hr-midrand:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           392M  1.5M  391M   1% /run
/dev/sda2       25G   6.3G   17G  28% /
tmpfs           2.0G   0  2.0G   0% /dev/shm
tmpfs           5.0M   8.0K   5.0M   1% /run/lock
CLOUDVBOX      486G  186G  281G  40% /media/sf_CLOUDVBOX
tmpfs          392M  124K  392M   1% /run/user/1000
/dev/sdb1       30G   24K   28G   1% /mnt/rorisang_new_disk

tebogo_matseding@pc25-hr-midrand:~$
```

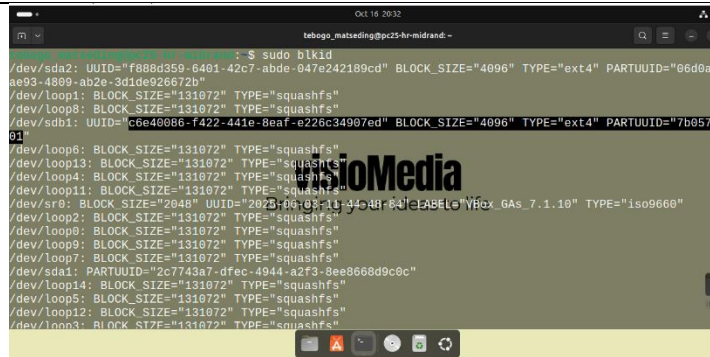
Configure Persistent Mounting

(Get the partition's UUID)

Find the UUID (unique ID) of the new hard drive:

`sudo blkid`

Copy the UUID for `/dev/sdb1`.



```
tebogo_matseding@pc25-hr-midrand:~$ sudo blkid
/dev/sda2: UUID="f888d359-6401-42c7-abde-047e242189cd" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="06d8a
ae93-4809-ab2e-3d1de926672b"
/dev/loop1: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop8: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/sdb1: UUID="c6e40866-f422-441e-8eaf-e226c34907ed" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="7b657
01"
/dev/loop3: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop13: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop4: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop11: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/sr0: BLOCK_SIZE="2048" UUID="2023-06-03-11:44:48-9413-ASUS_VBOX_GAs_7.1.10" TYPE="iso9660"
/dev/loop2: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop9: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop5: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop7: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/sda1: PARTUUID="2c7743a7-dfec-4944-a2f3-8ee8668d9c8c"
/dev/loop14: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop6: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop12: BLOCK_SIZE="131072" TYPE="squashfs"
/dev/loop3: BLOCK_SIZE="131072" TYPE="squashfs"
```

Configure Persistent Mounting

(Edit `/etc/fstab`)

Open the file:

`sudo nano /etc/fstab`



Configure Persistent Mounting

(add the UUID and path)

Add this line at the bottom (replace the UUID with yours):

`UUID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx /mnt/newhdd ext4 defaults 0 2`

Then save the file



Configure Persistent Mounting

(Test with `sudo mount -a`)

Test the configuration before rebooting:

`sudo mount -a`

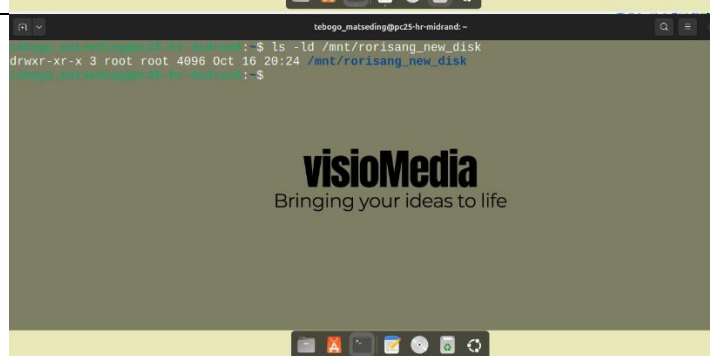
If no errors appear, the drive will mount automatically after every reboot.



Check Permission

Check who owns the drive:

`ls -ld /mnt/newhdd`



Set Ownership && Set

Permissions

Give ownership to the user

rorisang_mofokeng:

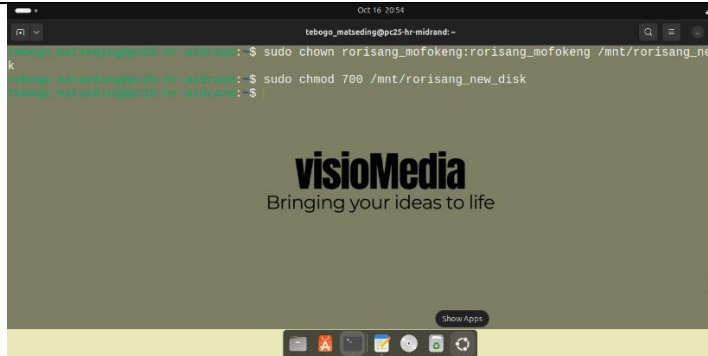
sudo chown

rorisang_mofokeng:rorisang_mofokeng /mnt/newhdd

sudo chmod 700 /mnt/newhdd

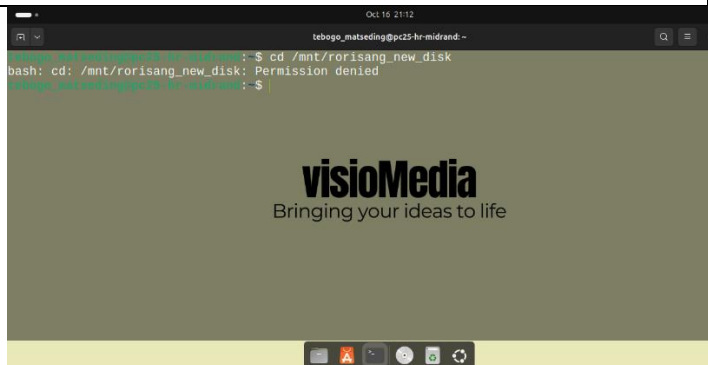
This means only that user can access the hard drive.

```
tebogo_matseding@pc25-hr-midrand:~$ sudo chown rorisang_mofokeng /mnt/rorisang_new_disk
tebogo_matseding@pc25-hr-midrand:~$ sudo chmod 700 /mnt/rorisang_new_disk
tebogo_matseding@pc25-hr-midrand:~$
```



Verify it blocks access

```
tebogo_matseding@pc25-hr-midrand:~$ cd /mnt/rorisang_new_disk
bash: cd: /mnt/rorisang_new_disk: Permission denied
tebogo_matseding@pc25-hr-midrand:~$
```



Verify user has permission

```
rorisang_mofokeng@pc25-hr-midrand:~$ cd /mnt/rorisang_new_disk
rorisang_mofokeng@pc25-hr-midrand:~/rorisang_new_disk$ ls
touch_code_of_conduct.pdf
rorisang_mofokeng@pc25-hr-midrand:~/rorisang_new_disk$ touch code_of_conduct.pdf
rorisang_mofokeng@pc25-hr-midrand:~/rorisang_new_disk$ ls
code_of_conduct.pdf touch_code_of_conduct.pdf
rorisang_mofokeng@pc25-hr-midrand:~/rorisang_new_disk$
```



Create a shortcut for the user

Make the drive appear on the desktop:

In -s /mnt/newhdd

~/Desktop/NewHDD

The user will now see a "NewHDD" folder on their desktop that opens the hard drive directly.

```
rorisang_mofokeng@pc25-hr-midrand:~$ ln -s /mnt/rorisang_new_disk ~/Desktop/NewHDD
rorisang_mofokeng@pc25-hr-midrand:~$
```



Project Overview

This project focused on setting up a new hard drive on a Linux system and preparing it for use by a specific user. It covered every stage of the process from identifying the drive to creating partitions, formatting, and making the drive accessible on the desktop.

The purpose of this project was to demonstrate how a support technician or student can prepare a storage device for immediate, secure use. While this version was designed as a demonstration, the same process applies in production environments, where drives must be prepared for users or servers.

By completing this project, you practiced essential Linux administration tasks such as disk management, mounting, permissions, and user access all foundational skills for technical support, systems administration, or cloud roles.

Possible Improvements

If this project were expanded, the following improvements could be added:

Add Encryption (LUKS): Secure the hard drive with disk encryption so that even if removed, the data cannot be accessed.

Automate Setup with a Script: Create a simple shell script that performs the entire setup automatically, useful for multiple systems.

Set Up Quotas: Limit how much storage each user can use on the drive.

Add Monitoring: Use tools like `df -h` or `smartctl` in a cron job to track drive health and available space.

Use a Dedicated Mount Directory: Mount the drive in `/media` or `/data` for a cleaner filesystem layout.

Add Backup or Snapshot Feature: Set up automated backups or `rsync` synchronization to another drive.

Create a GUI Shortcut: Instead of a terminal link, create a `.desktop` launcher with a custom icon for a more user-friendly look.