

Simulating Disk Partitioning and Persistent Mounting Using a Loopback Device on Ubuntu

Objective:

Since there is no second physical disk available (like `/dev/sdb`), you will simulate a disk using a loopback device backed by a file. You'll:

- Create a 1 GiB virtual disk file.
- Partition it using `fdisk` and `parted`.
- Format partitions using `ext4` and `XFS`.
- Label the file systems.
- Mount them to specific directories.
- Configure persistent mounting using UUIDs in `/etc/fstab`.

Verify block devices

Command:

lsblk -o NAME,SIZE,TYPE,MOUNTPOINT

Concept:

- Lists block devices and their partitions.

Command Part	Description
lsblk	Lists information about block devices (e.g. disks, partitions) in a tree-like format.
-o	Specifies the output columns to display.
NAME	Column showing the device name (e.g., sda, sda1).
SIZE	Column showing the size of each device or partition (e.g., 500G, 100M).
TYPE	Column indicating the type of device (e.g., disk, part, rom).
MOUNTPOINT	Column showing where the device is mounted in the filesystem (e.g., /, /home).

tiago@Linux:~\$ **lsblk -o NAME,SIZE,TYPE,MOUNTPOINT**

```
=====
NAME      SIZE TYPE MOUNTPOINT
loop0     4K loop /snap/bare/5
loop2     73.9M loop /snap/core22/1908
loop3     269.8M loop /snap/firefox/4793
loop4     11.1M loop /snap/firmware-updater/147
...
=====
```

Loop devices: These (loop0 to loop15) are read-only virtual block devices used to mount Snap packages.

Main disk: nvme0n1 is the primary NVMe SSD with 476.9 GB.

It has multiple partitions:

nvme0n1p1: EFI System Partition, mounted at /boot/efi

nvme0n1p3: A large partition (417.8 GB) mounted under /media/tiago-paquete/Windows — a Windows partition.

nvme0n1p5: The root filesystem / (53.7 GB)

No /dev/sdb listed: This means a secondary disk (like an external USB drive or internal second disk) is currently **not connected**, **not recognized**, or **not powered on**.

Concept: /dev/sdb Without Partitions

If you plug in a new (unpartitioned or unformatted) disk, it should appear as /dev/sdb (or similar) **without child entries** like sdb1, sdb2, etc.

That state confirms it has **no existing partitions**.

Create a 1 GiB virtual disk image

Command:

```
dd if=/dev/zero of=~/loopdisk.img bs=1M count=1024
```

Concept:

- This creates a blank 1 GiB file that we'll use as a virtual disk.
- /dev/zero provides zero-filled data.

```
tiago@Linux:~$ dd if=/dev/zero of=~/loopdisk.img bs=1M count=1024
```

```
=====
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 0.483757 s, 2.2 GB/s
=====
```

This command creates a 1 GB empty disk image (`loopdisk.img`) filled with zeroes, which can later be formatted or mounted using loopback.

Command Part	Description
dd	A Unix utility to convert and copy files, often used for low-level copying or disk imaging.
if=/dev/zero	Input file: /dev/zero is a special file that produces null bytes (\0) endlessly.
of=~/loopdisk.img	Output file: Specifies the file to create (loopdisk.img) in the user's home directory.
bs=1M	Block size: Sets the size of each read/write block to 1 megabyte .
count=1024	Number of blocks: Specifies that 1024 blocks should be copied, resulting in 1 GB total.

Associate the file with a loop device

Command:

```
sudo losetup -fP ~/loopdisk.img
```

Then check the assigned loop device:

```
sudo losetup -a
```

Concept:

- losetup binds a file to a loop device.
- The `-P`` flag automatically creates loop partitions (e.g., `/dev/loop16p1`).

```
tiago@Linux:~$ sudo losetup -fP ~/loopdisk.img
```

```
tiago@Linux:~$ sudo losetup -a | head -n 1
```

```
=====
/dev/loop1: [66309]:1607245 (/home/tiago-paquete/loopdisk.img)
=====
```

Command Part	Description
sudo	Executes the command with root privileges , required for loop device setup.
losetup	Tool to set up and control loop devices , which map files as block devices.
-f	Finds and uses the first available loop device (e.g., <code>/dev/loop0</code>).
-P	Automatically scan and create partition mappings (e.g., <code>/dev/loop0p1</code>).
~/loopdisk.img	The image file to associate with a loop device.

Command Part	Description
sudo	Executes the command with root privileges .
losetup	Calls the loop device utility.
-a	Lists all active loop devices and the files associated with them.

Partition the loop device with fdisk

Command:

sudo fdisk /dev/loop1

Command (m for help): n ← create new partition

Partition type: p ← primary

Partition number: 1 ← first partition

First sector: [press Enter] ← accept default (usually 2048)

Last sector: +500M ← size of the first partition

Command (m for help): n ← create another new partition

Partition type: p ← primary

Partition number: 2 ← second partition

First sector: [press Enter] ← accept default (right after previous partition)

Last sector: [press Enter] ← use rest of disk

Command (m for help): w ← write changes and exit

tiago@Linux:~\$ **sudo fdisk /dev/loop1**

=====

Welcome to fdisk (util-linux 2.39.3).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Device does not contain a recognized partition table.

Created a new DOS (MBR) disklabel with disk identifier 0xfb43e628.

Command (m for help): **n**

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p): **p**

Partition number (1-4, default 1): **1**

First sector (2048-2097151, default 2048): **[Return/Enter]**

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, default 2097151): +500M

Created a new partition **1** of type 'Linux' and of size 500 MiB.

Command (m for help): **n**

Partition type

p primary (1 primary, 0 extended, 3 free)

e extended (container for logical partitions)

Select (default p): **p**

Partition number (2-4, default 2): **2**

First sector (1026048-2097151, default 1026048): [\[Return/Enter\]](#)

Last sector, +/-sectors or +/-size{K,M,G,T,P} (1026048-2097151, default 2097151):
[\[Return/Enter\]](#)

Created a new partition 2 of type 'Linux' and of size 523 MiB.

Command (m for help): [w](#)

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

=====

Inform the kernel about the new partitions

Command:

```
sudo partprobe /dev/loop1
lsblk /dev/loop1
```

- Concept:
- partprobe updates the kernel's view of the partition table.
 - lsblk confirms the existence of /dev/loop1p1 and /dev/loop1p2.

```
tiago@Linux:~$ sudo partprobe /dev/loop1
=====
[sudo] password for tiago-paquete:
=====

tiago@Linux:~$ lsblk /dev/loop1
=====
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop1      7:1  0   1G  0 loop
└─loop1p1 259:8   0 500M  0 part
└─loop1p2 259:9   0 523M  0 part
=====
```

Component	Description
sudo	Runs the command with root privileges , required for managing devices.
partprobe	Utility that informs the kernel of partition table changes on a device.
/dev/loop1	The loop device to refresh, so the system sees new partitions (e.g. p1, p2).

Component	Description
lsblk	Lists block devices in a tree-like format (disks, partitions, etc.).
/dev/loop1	Limits the output to the specified loop device and its partitions.

Format the partitions

Command:

```
sudo mkfs.xfs -f /dev/loop1p1
sudo mkfs.ext4 -F /dev/loop1p2
```

Concept:

- You format partition 1 with XFS and partition 2 with ext4.
- These file systems are optimized for different purposes.

Component	Description
sudo	Executes the command with root privileges (required for formatting).
mkfs.xfs	Creates a new XFS filesystem , a high-performance journaling file system.
-f	Force the creation, even if the partition is not empty.
/dev/loop1p1	The target partition to format with XFS.

Component	Description
sudo	Executes the command with root privileges .
mkfs.ext4	Creates a new ext4 filesystem , widely used and supported in Linux.
-F	Force formatting, even if the device is in use or not a special block file.
/dev/loop1p2	The target partition to format with ext4.

```
tiago@Linux:~$ sudo mkfs.xfs -f /dev/loop1p1
```

```
=====
[sudo] password for tiago-paquete:
```

```
meta-data=/dev/loop1p1      isize=512  agcount=4, agsize=32000 blks
    =                       sectsz=512  attr=2, projid32bit=1
    =                       crc=1      finobt=1, sparse=1, rmapbt=1
    =                       reflink=1   bigtime=1 inobtcount=1 nnext64=0
data      =                  bsize=4096  blocks=128000, imaxpct=25
    =                       sunit=0     swidth=0 blks
naming    =version 2         bsize=4096  ascii-ci=0, ftype=1
log       =internal log     bsize=4096  blocks=16384, version=2
    =                       sectsz=512  sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096  blocks=0, rtextents=0
Discarding blocks...Done.
```

```
=====
```

Key Output	Description
meta-data=/dev/loop1p1	Confirms the target device being formatted.
isize=512	Size of each inode (in bytes).
agcount=4	Number of allocation groups (XFS uses these for parallelism).
bsize=4096	Block size: 4 KB (default for most modern systems).
blocks=128000	Total number of data blocks.
log=internal log	XFS journaling is internal (stored within the same partition).
reflink=1	Reflink support enabled (used for copy-on-write).
bigtime=1	Extended timestamp range (beyond Year 2038).
Discarding blocks...Done.	Wipes blocks before formatting for clean setup.

tiago@Linux:~\$ **sudo mkfs.ext4 -F /dev/loop1p2**

=====

mke2fs 1.47.0 (5-Feb-2023)

Discarding device blocks: done

Creating filesystem with 133888 4k blocks and 33520 inodes

Filesystem UUID: 916a6186-2ccb-4adf-9062-9809132a00d6

Superblock backups stored on blocks:

32768, 98304

Allocating group tables: done

Writing inode tables: done

Creating journal (4096 blocks): done

Writing superblocks and filesystem accounting information: done

=====

Key Output	Description
Creating filesystem with 133888 4k blocks	Shows the total number of 4 KB blocks.
and 33520 inodes	Indicates how many inodes (file entries) were created.
Filesystem UUID: ...	Unique identifier for mounting or tracking the filesystem.
Superblock backups stored on blocks:	Lists where backup superblocks are stored in case the main one is corrupted.
Creating journal (4096 blocks)	Journaling is enabled for filesystem integrity.
Writing superblocks....: done	Confirms successful formatting.

/dev/loop1p1 → **XFS**, supports large files, reflinks, and modern features.

/dev/loop1p2 → **ext4**, stable and widely supported, with journaling and superblock backups.

Label the partitions

Command:

```
sudo xfs_admin -L SysBackup /dev/loop1p1  
sudo e2label /dev/loop1p2 DataBackup
```

Concept:

- Labels help identify file systems independently of device names.

```
tiago@Linux:~$ sudo xfs_admin -L SysBackup /dev/loop1p1
```

```
=====
```

```
writing all SBs
```

```
new label = "SysBackup"
```

```
=====
```

```
tiago@Linux:~$ sudo e2label /dev/loop1p2 DataBackup
```

Component	Description
sudo	Runs the command with root privileges , required for modifying device labels.
xfs_admin	Tool to manage XFS filesystem metadata.
-L SysBackup	Sets the label of the XFS filesystem to "SysBackup".
/dev/loop1p1	The target XFS partition to label.

Component	Description
sudo	Executes with root privileges.
e2label	Tool to set or view labels for ext2/ext3/ext4 filesystems.
/dev/loop1p2	The target ext4 partition to label.
DataBackup	The new filesystem label being assigned.

Create mount points

Command:

```
sudo mkdir -p /mnt/loop/system
sudo mkdir -p /mnt/loop/data
```

Concept:

- These directories will be the mount targets for the partitions.

```
tiago@Linux:~$ sudo mkdir -p /mnt/loop/system
[sudo] password for tiago-paquete:
tiago@Linux:~$ sudo mkdir -p /mnt/loop/data
```

Component	Description
sudo	Runs the command with root privileges , required to create directories in <code>/mnt</code> .
mkdir	Command to create a new directory.
-p	Creates parent directories as needed (e.g., <code>/mnt</code> , <code>/mnt/loop</code> ,
<code>/mnt/loop/system</code>	The full path of the directory to create — for mounting the system partition.

```
tiago-paquete@tiago-paquete-Linux:/mnt/loop$ ls -l
=====
total 8
drwxr-xr-x 2 root root 4096 May  2 17:38 data
drwxr-xr-x 2 root root 4096 May  2 17:37 system
=====
```

Get the UUIDs of the partitions

Command:

```
sudo blkid /dev/loop1p1 /dev/loop1p2
```

Output Example:

```
/dev/loop1p1: UUID="abc123..." TYPE="xfs" LABEL="SysBackup"  
/dev/loop1p2: UUID="def456..." TYPE="ext4" LABEL="DataBackup"
```

Concept:

- UUIDs are reliable identifiers that persist across reboots and device renaming.

Component	Description
sudo	Runs the command with root privileges , needed to read block device metadata.
blkid	Lists information about block devices , such as UUIDs, filesystem types, and labels.
/dev/loop1p1	First partition (XFS) to inspect — labeled SysBackup.
/dev/loop1p2	Second partition (ext4) to inspect — labeled DataBackup.

```
tiago@Linux:~$ sudo blkid /dev/loop1p1 /dev/loop1p2
```

```
=====
/dev/loop1p1: LABEL="SysBackup" UUID="784c0940-0d74-44df-885a-c41a1b80eb18"
BLOCK_SIZE="512" TYPE="xfs" PARTUUID="fb43e628-01"
```

```
/dev/loop1p2: LABEL="DataBackup" UUID="916a6186-2ccb-4adf-9062-9809132a00d6"
BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="fb43e628-02"
=====
```

Field	Meaning
LABEL	The user-assigned label (via <code>xfs_admin</code> or <code>e2label</code>).
UUID	The unique identifier for the filesystem, useful for mounting via <code>/etc/fstab</code> .
TYPE	Filesystem type (e.g., <code>xfs</code> , <code>ext4</code>).

Add entries to /etc/fstab for persistent mounting

Command:

sudo nano /etc/fstab

Add the following lines at the bottom (replace UUIDs with your actual ones):

```
UUID=abc123... /mnt/loop/system xfs defaults,noatime 0 2
UUID=def456... /mnt/loop/data   ext4 defaults,noatime 0 2
```

Concept:

- /etc/fstab ensures the partitions are mounted at boot.
- noatime improves performance by disabling access time updates.

tiago@Linux:~\$ **sudo nano /etc/fstab**

```
=====
GNU nano 7.2 /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>    <dump> <pass>
# / was on /dev/nvme0n1p5 during curtin installation
/dev/disk/by-uuid/0386062d-xxxx-4808-xxxx-75b28c9a6xxx / ext4 defaults 0 1
# /boot/efi was on /dev/nvme0n1p1 during curtin installation
/dev/disk/by-uuid/42E0-3CCC /boot/efi vfat defaults 0 1
/swap.img      none    swap    sw      0       0

UUID=784c0940-xxxx-xxxx-885a-c41a1b80exxx /mnt/loop/system xfs
defaults,noatime 0 2
UUID=916a6186-xxxx-4adf-xxxx-9809132a0xxx /mnt/loop/data  ext4 defaults,noatime
0 2
=====
```

Field	Description
UUID=...	Refers to the unique ID of each partition.
/mnt/loop/system / /mnt/loop/ data	Mount points — directories that must exist before mounting.
xfs / ext4	Filesystem types, matching what you've formatted each partition with.
defaults,noatime	Mount options: standard settings + disables atime updates (improves performance).
0	Dump (rarely used, safe to leave at 0).
2	Filesystem check order (1 = root, 2 = others, 0 = skip check).

Mount all entries from fstab

Command:

```
sudo mount -a
```

Confirm with:

```
df -h | grep /mnt/loop
```

```
findmnt | grep loop
```

Concept:

- mount -a mounts everything listed in fstab.
- df and findmnt verify successful mounting.

Purpose	Description
df -h	Shows disk usage of mounted filesystems in human-readable format .
grep /mnt/loop	Filters to show only the entries mounted at /mnt/loop/...

Purpose	Description
findmnt	Lists all mounted filesystems in a structured tree.
grep loop	Filters output to show mounts related to loop devices .

```
tiago@Linux:~$ sudo mount -a
```

```
=====
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
=====
```

```
tiago@Linux:~$ sudo df -h | grep /mnt/loop
```

```
=====
/dev/loop1p1    436M   34M  403M    8% /mnt/loop/system
/dev/loop1p2    498M    24K  462M    1% /mnt/loop/dat
=====
```

```
tiago@Linux:~$ findmnt | grep loop |tail -n 2
```

```
=====
└─/mnt/loop/system                               /dev/loop1p1
xfs      rw,noatime,attr2,inode64,logbufs=8,logbsize=32k,noquota
└─/mnt/loop/data                                /dev/loop1p2
ext4      rw,noatime
=====
```

Generate sample data using /dev/urandom

Command:

```
head -c 1M /dev/urandom | sudo tee /mnt/loop/data/testfile.bin > /dev/null
ls -lh /mnt/loop/data/testfile.bin
```

Concept:

- head + /dev/urandom generates 1 MiB of random data to test write capability.

```
tiago@Linux:~$ head -c 1M /dev/urandom | sudo tee /mnt/loop/data/testfile.bin > /dev/null
tiago@Linux:~$ ls -lh /mnt/loop/data/testfile.bin
=====
-rw-r--r-- 1 root root 1.0M May  2 18:15 /mnt/loop/data/testfile.bin
=====
```

Component	Description
head	A command that outputs the first part of a file or stream.
-c 1M	Tells head to output the first 1 megabyte (1M = 1,048,576 bytes) of data.
/dev/urandom	A special file that generates random bytes . Used here as a source of test data.
`	` (pipe)
sudo	Runs the next command (tee) with root privileges (required to write in /mnt).
tee /mnt/loop/data/testfile.bin	tee writes the data it receives from the pipe into the specified file and echoes it to stdout .
> /dev/null	Redirects stdout to /dev/null, so it's discarded (prevents random bytes from printing).

Clean up (optional, for test environments)

Commands:

```
sudo umount /mnt/loop/system /mnt/loop/data
sudo losetup -d /dev/loop1
rm ~/loopdisk.img
```

Concept:

- Unmount partitions, detach the loop device, and remove the image file to clean up.

sudo umount /mnt/loop/system /mnt/loop/data

Component	Description
sudo	Runs with root privileges , required for unmounting filesystems.
umount	Unmounts the specified mounted filesystems .
/mnt/loop/system	Mount point for the XFS partition — this will be unmounted.
/mnt/loop/data	Mount point for the ext4 partition — also unmounted.

Both partitions must be unmounted before detaching the loop device.

sudo losetup -d /dev/loop1

Component	Description
sudo	Required for modifying loop devices.
losetup	Tool to manage loopback devices.
-d	Detach the loop device, releasing /dev/loop1 from the image file.
/dev/loop1	The loop device to detach (was linked to loopdisk.img).

This step **frees** the loop device for future use.

rm ~/loopdisk.img

Component	Description
rm	Deletes a file.
~/loopdisk.img	Refers to the disk image file in your home directory.

This deletes the 1 GB image file you created earlier.

```
tiago@Linux:~$ sudo umount /mnt/loop/system /mnt/loop/data
tiago@Linux:~$ sudo losetup -d /dev/loop1
tiago@Linux:~$ rm loopdisk.img
```

Concepts Covered:

- Creating virtual disks using `dd` and `/dev/zero`
- Using `losetup` to simulate block devices
- Partitioning with `fdisk`, refreshing with `partprobe`
- Formatting partitions with `mkfs.xfs` and `mkfs.ext4`
- Labeling file systems with `xfs_admin` and `e2label`
- Mounting partitions persistently using `/etc/fstab` and UUIDs
- Testing write access using `head` and `/dev/urandom`

Important Files and Directories:

- | | |
|------------------------------------|--|
| - ~/loopdisk.img | → simulated virtual disk |
| - /dev/loop1p1, /dev/loop1p2 | → simulated partitions |
| - /mnt/loop/system, /mnt/loop/data | → mount points |
| - /etc/fstab | → file for persistent mounting |
| - /dev/disk/by-uuid | → persistent UUID symlinks for all disks |