

du - Disk Usage  
df - Disk Free

FileSystems -

Block Devices - store data in disk

Virtual Filesystems - store data in memory

Type	Storage Medium	Examples	Notes
Real Filesystems	Stored on physical disks	ext4, <del>xfs</del> , <del>vfat</del> , <del>squashfs</del>	Persist across reboots
Virtual Filesystems	Stored in memory (RAM)	proc, <del>sysfs</del> , <del>tmpfs</del> , cgroup2	Made to help the OS. Do <b>not</b> persist after reboot

Types of Filesystems

Filesystem Type	Where it's Used	Technical Role
ext4	Local disks	Journaling filesystem for reliability and speed. Most common in Linux.
<del>xfs</del>	Large filesystems	High-performance, supports parallel I/O. Good for large-scale systems.
<del>vfat</del>	USBs, EFI partitions	Booting Support
<del>squashfs</del>	Snap packages, ISOs	Compressed, read-only. Good for static content.
<del>tmpfs</del>	/tmp, /run, /dev/ <del>shm</del>	RAM-backed filesystem. Very fast, non-persistent.
proc	/proc	Interface to kernel and process information. Not real files.
<del>sysfs</del>	/sys	Shows device and kernel subsystem structure.
cgroup2	/sys/fs/cgroup	Resource control (CPU, memory, etc.) for processes.
nfs	Network shares	Mount remote filesystems over network.
autofs	/proc/sys/fs/binfmt_misc	Auto-mounts filesystems when accessed.

mount - is a command - to either list down the mounts or mount a filesystem to a mountpoint

MountPoint - Linux equivalent of Local Disk (E/F/G etc) - Address of the filesystem

### mount

- **What it does:**  
Connects a filesystem (e.g., disk, USB, NFS) to a directory path.
- **Example:**  
sudo mount /dev/sdb1 /mnt/data

### umount

- **What it does:**  
Safely disconnects a mounted filesystem.
- **Example:**  
sudo umount /mnt/data

Field	Description
DEVICE/SOURCE	The block device, virtual filesystem, or special resource being mounted (e.g., /dev/xyda1, tmpfs, proc)
MOUNT POINT	The directory path where the device or resource is mounted (e.g., /, /dev, /proc)
FILESYSTEM TYPE	Type of filesystem (e.g., ext4, tmpfs, proc, squashfs, nfs, etc.)
OPTIONS	Mount options like read/write mode, access control, caching, etc.

## fstab

/etc/fstab (short for FileSystem TABLE) is a configuration file that tells Linux:

- \* What filesystems to mount automatically at boot
- \* Where to mount them
- \* How to mount them (with what options)

This file helps automate and stabilize the mounting process.

Column	Description
DEVICE	What to mount — can be a device (/dev/sda1), a UUID (UUID=...), or a label (LABEL=...)
MOUNT_POINT	Where to mount it in the filesystem (e.g., /, /boot)
FSTYPE	Filesystem type (ext4, vfat, xfs, nfs, etc.)
OPTIONS	Mount options (e.g., defaults, ro, noexec, discard, etc.)
DUMP	Whether to backup with dump (0 = no; rarely used now)
PASS	Order for filesystem checks at boot by fsck (0 = skip, 1 = root, 2 = others)

## fsck - File System Consistency Check

## Adding a New Volume in an AWS server -

1. Attached volume from the console
2. lsblk to check all block devices ( last entry added in step 1)

```
root@ip-172-31-27-45:/# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0  27.6M  1 loop /snap/amazon-ssm-agent/11797
loop1       7:1      0  48.1M  1 loop /snap/snapd/25935
loop3       7:3      0   74M  1 loop /snap/core22/2163
loop4       7:4      0  27.8M  1 loop /snap/amazon-ssm-agent/12322
loop5       7:5      0  50.9M  1 loop /snap/snapd/25577
loop6       7:6      0   74M  1 loop /snap/core22/2193
xvda        202:0     0  100G  0 disk
├─xvda1     202:1     0   99G  0 part /
├─xvda14    202:14    0    4M  0 part
├─xvda15    202:15    0  106M  0 part /boot/efi
└─xvda16    259:0     0   913M  0 part /boot
xvde        202:64    0    5G  0 disk
```

3. mkfs - make filesystem (ext4 in this case)

**NOTE: THIS COMMAND FORMATS THE DISK's DATA. DO NOT RUN IT IF THERE'S ANY DATA ON THE DISK**

```
root@ip-172-31-27-45:/# mkfs.ext4 /dev/xvde
mke2fs 1.47.0 (5-Feb-2023)
/dev/xvde contains a ext4 file system
    last mounted on /demo on Mon Jan 12 17:24:36 2026
Proceed anyway? (y,N) y
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: e192620e-978e-4791-aec9-d26a05c16a8b
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

4. Create mount directory

mkdir /sample

5. Mount the volume:

```
root@ip-172-31-27-45:/# mount /dev/xvde /sample
mount: (hint) your fstab has been modified, but systemd still uses
    the old version; use 'systemctl daemon-reload' to reload.
root@ip-172-31-27-45:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        96G   2.7G   94G   3% /
tmpfs           7.9G     0   7.9G   0% /dev/shm
tmpfs           3.2G   952K   3.2G   1% /run
tmpfs           5.0M     0   5.0M   0% /run/lock
/dev/xvda16     881M  155M   665M  19% /boot
/dev/xvda15     105M   6.2M   99M   6% /boot/efi
tmpfs           1.6G   12K   1.6G   1% /run/user/1000
tarun           7.9G     0   7.9G   0% /data
/dev/xvde        4.9G   24K   4.6G   1% /sample
root@ip-172-31-27-45:/#
```

## 6. Check UUID from blkid command

```
root@ip-172-31-27-45:/# blkid /dev/xvde
/dev/xvde: UUID="e192620e-978e-4791-aec9-d26a05c16a8b" BLOCK_SIZE="4096" TYPE="ext4"
root@ip-172-31-27-45:/#
```

## 7. Put it in the fstab file

```
root@ip-172-31-27-45:/# cat /etc/fstab
LABEL=cloudimg-rootfs / ext4 discard,commit=30,errors=remount-ro 0 1
LABEL=BOOT /boot ext4 defaults 0 2
LABEL=UEFI /boot/efi vfat umask=0077 0 1
UUID="e192620e-978e-4791-aec9-d26a05c16a8b" /dev/xvde ext4 defaults 0 2
root@ip-172-31-27-45:/#
```

and you are done.....

## PROBLEM STATEMENT

Maya is responsible for a Linux server where:

- \* Disks fill up without warning
- \* Inodes run out because of log files
- \* Mounts disappear after reboots
- \* Applications fail when directories are missing
- \* She needs real-time alerts when files are added

Maya needs one small shell script that can:

- \* Check disk space
- \* Check inode usage
- \* Find large files
- \* Verify mounts
- \* Validate important directories
- \* Watch upload folders in real time

Take /var folder for testing

