# mount and lsblk Disks and mounts

**START HERE**

**`lsblk -f`**
Disks, filesystems, UUIDs

**`findmnt`**
What is mounted where

**`sudo mount <device> <dir>`**
Mount a filesystem

**`sudo mount -o ro <device> <dir>`**
Mount read-only

**`sudo umount <dir>`**
Unmount it

**`sudo blkid`**
UUIDs and labels

**`findmnt --verify`**
Check fstab for errors

**`sudo mount -a`**
Mount everything in fstab

**Block device** /dev/sdb1 `mkfs` → **Filesystem** UUID, label `mount` → **Mount point** /mnt/data `umount` → **Detached** safe to unplug

## CONCEPTS

Mount points are just directories: mounting hides whatever was in them until you unmount.

Use UUID= in fstab: /dev/sdX names can change between boots.

## DISCOVER

**`lsblk`**
Tree of block devices

**`lsblk -f`**
Add filesystem, label, UUID, mountpoint

**`lsblk -o NAME,SIZE,TYPE, FSTYPE,MOUNTPOINTS`**
Pick your own columns

**`lsblk -p`**
Show full /dev paths

**`lsblk -J`**
JSON output for scripts

**`sudo blkid`**
List UUID, label and type of devices

**`sudo blkid <device>`**
Probe a single device

**`findmnt`**
Tree of mounted filesystems

**`findmnt <dir>`**
Which filesystem backs this mount point

**`findmnt -T <path>`**
Mount holding any path

**`findmnt -t ext4,xfs`**
Only some filesystem types

**`findmnt --real`**
Hide pseudo filesystems

**`findmnt -D`**
Output like df

**`findmnt -s`**
Read fstab instead of live mounts

**`findmnt -o TARGET,SOURCE, FSTYPE,OPTIONS`**
Choose columns

**`mount`**
List mounts (noisy, includes pseudo)

**`mount -t <type>`**
List mounts of one type

## MOUNT AND UNMOUNT

**`sudo mount <device> <dir>`**
Mount, type auto-detected

**`sudo mount -t <type> <device> <dir>`**
Force the filesystem type

**`sudo mount -o <opts> <device> <dir>`**
Mount with comma-separated options

**`sudo mount -r <device> <dir>`**
Mount read-only

**`sudo mount UUID=<uuid> <dir>`**
Mount by UUID

**`sudo mount -L <label> <dir>`**
Mount by label

**`sudo mount -m <device> <dir>`**
Create the mount point first

**`sudo mount -a`**
Mount all fstab entries not yet mounted

**`sudo mount <dir>`**
Mount using the fstab entry for <dir>

**`sudo mount -o loop <file.iso> <dir>`**
Mount an image file

**`sudo mount -o ro,loop <file.iso> <dir>`**
Mount an ISO read-only

**`sudo mount -f -v <device> <dir>`**
Dry run, verbose

**`sudo umount <dir>`**
Unmount by mount point

**`sudo umount <device>`**
Unmount by device

**`sudo umount -l <dir>`**
Lazy: detach now, clean up later

*Use when busy and safe to detach; open files stay usable until closed.*

**`sudo umount -R <dir>`**
Unmount the whole tree below

**`sudo umount -f <dir>`**
Force (mostly for dead NFS)

> "target is busy": find the culprit with lsof +f -- <dir> or fuser -vm <dir>, or cd out of it.

## BIND AND REMOUNT

**`sudo mount --bind <dir> <other>`**
Make a directory visible at a second path

**`sudo mount --rbind <dir> <other>`**
Bind including submounts

**`sudo mount -o remount,ro <dir>`**
Switch a live mount to read-only

**`sudo mount -o remount,rw <dir>`**
Switch a live mount to read-write

**`sudo mount -o remount,noatime <dir>`**
Change options without unmounting

**`sudo mount --bind -o ro <dir> <other>`**
Read-only view of a directory

*Needs util-linux 2.27+; older: bind, then remount,bind,ro.*

**`sudo mount --move <old> <new>`**
Move a mount point atomically

**`sudo mount --make-private <dir>`**
Stop mount event propagation

**`sudo mount -t tmpfs -o size=<size> tmpfs <dir>`**
RAM-backed scratch space

**`sudo mount -t nfs <host>:<export> <dir>`**
Mount an NFS share

**`sudo mount -t cifs // <host>/<share> <dir> -o username=<user>`**
Mount an SMB share

*Needs the cifs-utils package.*

## FSTAB AND OPTIONS

### fstab fields (/etc/fstab)

| # | Field | Example |
|---|---|---|
| 1 | device or UUID= | UUID=1a2b-3c4d |
| 2 | mount point | /data |
| 3 | filesystem type | ext4 |
| 4 | options | defaults,noatime |
| 5 | dump (legacy) | 0 |
| 6 | fsck order | 2 |

### A typical fstab line

```
# device        dir    type
options           dump pass
UUID=<uuid>    /data  ext4
defaults,noatime  0    2
```

## FSTAB AND OPTIONS (cont.)

### Common options

| Option | Effect |
|---|---|
| `defaults` | rw,suid,dev,exec,auto,r |
| `ro / rw` | read-only / read-write |
| `noexec` | forbid running binaries |
| `nosuid` | ignore setuid bits |
| `nodev` | ignore device files |
| `noatime` | skip access-time writes |
| `nofail` | do not fail boot if absent |
| `noauto` | skip on mount -a and boot |
| `user` | non-root users may mount |
| `x-systemd.automount` | mount on first access |

**`findmnt --verify`**
Lint fstab before rebooting

**`sudo systemctl daemon-reload`**
Make systemd re-read fstab

**`findmnt --verify --verbose`**
Show every check

> Test an fstab edit with mount -a before rebooting; a bad entry can drop boot into emergency mode.

> Put nofail on removable or network disks so a missing one cannot block boot.

## RECIPES

### Mount a new USB disk

```
lsblk -f
sudo mkdir -p /mnt/usb
sudo mount /dev/sdb1 /mnt/usb
# when done
sudo umount /mnt/usb
```

### Persist a disk across reboots

```
lsblk -f                                #
note the UUID
sudo mkdir -p /data
echo 'UUID=<uuid> /data ext4
defaults,nofail 0 2' | sudo tee
-a /etc/fstab
findmnt --verify && sudo mount
-a
```

### Inspect an ISO

```
sudo mount -o loop,ro
image.iso /mnt
ls /mnt
sudo umount /mnt
```

### Fix a read-only root, then undo

```
sudo mount -o remount,rw /
# edit what you need
sudo mount -o remount,ro /
```