fdisk Disk partitioning

START HERE

lsblk
Find the right disk

n
Add a partition (in fdisk)

sudo fdisk -l *<disk>*
Show its partitions

w
Write and exit (in fdisk)

sudo fdisk *<disk>*
Open interactive editor

sudo mkfs.ext4 *<part>*
Format the partition

g
New GPT table (in fdisk)

sudo mount *<part> <dir>*
Use it

Partition table fdisk / sfdisk → `partprobe` → **Partition** /dev/sdb1 → `mkfs` → **Filesystem** ext4, xfs, swap → `mount` → **Mount point** /mnt/data

CONCEPTS

Triple-check the device name with lsblk before you write. Writing a partition table to the wrong disk destroys its data.

fdisk keeps every change in memory until you press w. q leaves without touching the disk.

MBR (dos) tops out at 2 TiB and 4 primary partitions. GPT has neither limit and is the default choice for new disks.

Whole-disk device is /dev/sdb, the partition is /dev/sdb1. NVMe and loop devices add a p: /dev/nvme0n1p1.

INSPECT

lsblk
Tree of disks and partitions

lsblk -f
Add filesystem type, label, UUID

sudo fdisk -l
List partition tables of all disks

sudo fdisk -l *<disk>*
List one disk's table

lsblk -o NAME,SIZE,UUID
Pick your own columns

sudo blkid
UUIDs and filesystem types

sudo sfdisk -F *<disk>*
List free unpartitioned areas

INTERACTIVE FDISK

sudo fdisk *<disk>*
Edit a disk's partition table

m
Show the command menu

p
Print the table as edited so far

g
New empty GPT table

INTERACTIVE FDISK (cont.)

o
New empty MBR (DOS) table

n
Add a partition

d
Delete a partition

t
Change partition type

w
Write changes and exit

q
Quit without saving

v
Verify the table

i
Info about one partition

F
List free unpartitioned space

l
List known partition types

a
Toggle bootable flag (MBR only)

I / O
Load / dump layout as sfdisk script

CREATING A PARTITION

n Enter Enter +*<size>*
Next number, first free sector, given size

Press Enter to accept defaults. Sizes take K, M, G, T suffixes, e.g. +20G.

n Enter Enter Enter
Use all remaining space

t *<n>* **linux**
Linux filesystem type (alias)

Aliases exist in fdisk 2.36 and later; otherwise pick a number from L.

t *<n>* **swap**
Linux swap type

t *<n>* **lvm**
Linux LVM type

CREATING A PARTITION (cont.)

t *<n>* **uefi**
EFI System partition type

APPLY AND FORMAT

sudo partprobe *<disk>*
Tell the kernel about new partitions

Part of the parted package; not installed everywhere.

sudo mkfs.ext4 *<part>*
Format as ext4

sudo mkfs.xfs *<part>*
Format as XFS

sudo mkfs.vfat -F 32 *<part>*
Format as FAT32 (EFI, USB)

sudo mkswap *<part>*
Prepare swap space

sudo swapon *<part>*
Enable swap now

sudo mkfs.ext4 -L *<label> <part>*
Format with a label

sudo mount *<part> <dir>*
Mount it

sudo blkid *<part>*
UUID for /etc/fstab

SFDISK (SCRIPTING)

sudo sfdisk -d *<disk>* **>** *<file>*
Dump the table to a file

sudo sfdisk *<disk>* **<** *<file>*
Restore or apply a layout

Overwrites the table without asking when stdin is a file.

sudo sfdisk --delete *<disk> <n>*
Delete partition n

sudo sfdisk -b *<disk>*
Back up table sectors first

SAFETY AND CLEANUP

sudo wipefs *<disk>*
List signatures (no changes)

sudo wipefs -a *<disk>*
Erase all signatures

Destructive: the disk stops looking like a disk with data.

sudo umount *<part>*
Unmount before editing

fdisk will not stop you from deleting a mounted or used partition. Unmount first, and keep a table dump from sfdisk.

Resizing a partition does not resize its filesystem. Grow the partition first, then resize2fs / xfs_growfs; shrink in the reverse order.

To undo a mistake after w, you need the dumped table. Restore with sfdisk, then check with fsck before mounting.

RECIPES

New GPT disk with one ext4 partition

```
lsblk
sudo fdisk /dev/sdb      # g,
n, Enter x3, w
sudo mkfs.ext4 /dev/sdb1
sudo mount /dev/sdb1 /mnt
```

Clone a layout to another disk

```
sudo sfdisk -d /dev/sda >
sda.dump
# review sda.dump, then:
sudo sfdisk -n /dev/sdb <
sda.dump
sudo sfdisk /dev/sdb < sda.dump
```

Scripted GPT layout, no prompts

```
sudo sfdisk /dev/sdb <<EOF
label: gpt
,512M,U
,,L
EOF
```

Partition for LVM

```
sudo fdisk /dev/sdb      # g,
n, t, lvm, w
sudo pvcreate /dev/sdb1
sudo vgcreate vg0 /dev/sdb1
```