# btrfs Copy-on-write filesystem

## CONCEPTS

A subvolume looks like a directory but snapshots, quotas and send work per subvolume. Snapshots are cheap copy-on-write clones.

**Filesystem** one or more devices
- **Subvolume** top-level (ID 5) and children
  - **Snapshot** a subvolume sharing data

## INSPECT

`btrfs filesystem show`
List filesystems and their devices

`btrfs filesystem usage <path>`
Show allocated, free and unallocated space

`btrfs filesystem df <path>`
Show space per chunk type

`btrfs device usage <path>`
Show allocation per device

## SUBVOLUMES

`btrfs subvolume create <path>`
Create a subvolume

`btrfs subvolume list <path>`
List subvolumes

`btrfs subvolume delete <path>`
Delete a subvolume

`btrfs subvolume show <path>`
Show UUID, generation and flags

## SNAPSHOTS

`btrfs subvolume snapshot <src> <dst>`
Snapshot a subvolume (read-write)

`btrfs subvolume snapshot -r <src> <dst>`
Snapshot a subvolume read-only

`btrfs property get -ts <path> ro`
Check the read-only flag

## SNAPSHOTS (cont.)

`btrfs property set -ts <path> ro false`
Make a snapshot writable

## SEND AND RECEIVE

`btrfs send <snap> | btrfs receive <dir>`
Copy a read-only snapshot

`btrfs send -p <parent> <snap> | btrfs receive <dir>`
Send only the changes since a parent

`btrfs send -f <file> <snap>`
Write a send stream to a file

`btrfs receive -f <file> <dir>`
Read a send stream from a file

## SCRUB AND BALANCE

`btrfs scrub start <path>`
Verify checksums in the background

`btrfs scrub status <path>`
Show scrub progress and errors

`btrfs balance start -dusage=50 <path>`
Compact data chunks under 50% full

`btrfs balance start -dusage=0 <path>`
Free completely empty data chunks

`btrfs balance status <path>`
Show balance progress

## DEVICES AND MAINTENANCE

`btrfs device add <dev> <path>`
Add a device to a mounted filesystem

`btrfs device remove <dev> <path>`
Remove a device, moving its data off

## REPAIR

`btrfs check <dev>`
Check an unmounted filesystem (read-only)