btrfs Copy-on-write filesystem

START HERE

`btrfs filesystem show`
List filesystems and devices

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

`btrfs filesystem usage <path>`
Real space use

`btrfs balance start -dusage=50 <path>`
Compact chunks

`btrfs subvolume list <path>`
List subvolumes

`btrfs subvolume create <path>`
New subvolume

`btrfs subvolume snapshot -r <src> <dst>`
Read-only snapshot

`btrfs scrub start <path>`
Verify checksums

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

df is misleading on btrfs. Use btrfs filesystem usage, which shows allocated and unallocated space per profile.

Most commands need root and take a path inside the mounted filesystem, not the device.

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 filesystem usage -T <path>`
Show usage as a table

INSPECT (cont.)

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

`btrfs filesystem du -s <path>`
Show exclusive and shared size

`btrfs filesystem label <path>`
Show the filesystem label

`btrfs device stats <path>`
Show per-device IO error counters

SUBVOLUMES

`btrfs subvolume create <path>`
Create a subvolume

`btrfs subvolume list <path>`
List subvolumes

`btrfs subvolume delete <path>`
Delete a subvolume
The directory vanishes at once; data is freed in the background. Add -c to wait for a commit.

`btrfs subvolume list -t <path>`
List subvolumes as a table

`btrfs subvolume list -s <path>`
List snapshots only

`btrfs subvolume list -o <path>`
List subvolumes directly below a path

SUBVOLUMES (cont.)

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

`btrfs subvolume create -p <path>`
Create it with missing parent dirs

`btrfs subvolume delete -c <path>`
Delete and wait for the commit

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

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

`btrfs subvolume snapshot <src> <src>/.snap-<name>`
Snapshot into the source itself

A snapshot does not include nested subvolumes: they show up as empty directories.

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

`btrfs send <snap> | ssh <host> btrfs receive <dir>`
Send over SSH

`btrfs send -c <clone> <snap>`
Use another snapshot as clone source

Only read-only subvolumes can be sent. An incremental send needs the same parent snapshot present on both sides.

SCRUB AND BALANCE

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

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

SCRUB AND BALANCE (cont.)

`btrfs scrub start -B <path>`
Scrub in the foreground

`btrfs scrub cancel <path>`
Stop a scrub, keeping progress

`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

`btrfs balance pause <path>`
Pause a running balance

`btrfs balance resume <path>`
Continue a paused balance

`btrfs balance cancel <path>`
Cancel a balance

`btrfs balance start -dconvert=raid1 -mconvert=raid1 <path>`
Convert to another RAID profile

A balance needs unallocated space on the devices. Check btrfs filesystem usage first, and start with a low usage filter.

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

`btrfs replace start <old> <new> <path>`
Replace a device while mounted

`btrfs replace status <path>`
Show replace progress

`btrfs filesystem resize max <path>`
Grow to fill the device

`btrfs filesystem resize +<size> <path>`
Grow by an amount

`btrfs filesystem resize -<size> <path>`
Shrink by an amount

REPAIR

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

Try scrub, a read-only mount (-o ro,rescue=all) and btrfs restore before any check --repair.