ZFS Pools and snapshots

Disks vdevs: mirror, raidz → `zpool create` → **Pool** shared storage → `zfs create` → **Dataset** filesystem or zvol → `zfs snapshot` → **Snapshot** read-only, cheap → `zfs clone / send` → **Clone / stream** writable copy or backup

CONCEPTS

Two commands: zpool manages disks and pools, zfs manages datasets (filesystems, volumes, snapshots) inside them.

Datasets are cheap and inherit properties from their parent: make one per thing you want to snapshot, quota or tune separately.

POOLS

`zpool status`
Health, errors and scrub state

`zpool status -x`
Show only unhealthy pools

`zpool list`
Size, used, free and health per pool

`zpool create <pool> mirror <disk1> <disk2>`
Create a mirrored pool

`zpool create <pool> raidz <disk1> <disk2> <disk3>`
Create a raidz pool

`zpool create -o ashift=12 <pool> <disk>`
Force 4K sector alignment

POOLS (cont.)

`zpool scrub <pool>`
Verify all checksums, repair from redundancy

`zpool import`
List pools available to import

`zpool import <pool>`
Import a pool

`zpool import -d /dev/disk/by-id <pool>`
Import searching a given directory

`zpool export <pool>`
Cleanly detach a pool

`zpool iostat -v <pool> 5`
Per-vdev I/O every 5 seconds

`zpool history <pool>`
Show commands run against the pool

`zpool clear <pool>`
Reset error counters after a fault

`zpool replace <pool> <old> <new>`
Swap a failed or old disk

DATASETS

`zfs list`
List filesystems and volumes

DATASETS (cont.)

`zfs list -r <pool>`
List a pool's datasets recursively

`zfs create <pool>/<name>`
Create a filesystem

`zfs create -p <pool>/<a>/<b>`
Create with missing parents

`zfs destroy <dataset>`
Delete a dataset

`zfs destroy -rn -v <dataset>`
Dry-run a recursive destroy

`zfs mount -a`
Mount all datasets

`zfs unmount <dataset>`
Unmount a dataset

`zfs list -o space`
Space broken down by use

PROPERTIES

`zfs get all <dataset>`
Show every property

`zfs get <prop> <dataset>`
Show one property

`zfs set compression=lz4 <dataset>`
Enable compression

PROPERTIES (cont.)

`zfs set mountpoint=<path> <dataset>`
Change where it mounts

`zfs set quota=<size> <dataset>`
Cap a dataset's size

SNAPSHOTS AND CLONES

`zfs snapshot <dataset>@<name>`
Create a snapshot

`zfs snapshot -r <pool>@<name>`
Snapshot a pool and all children

`zfs list -t snapshot`
List snapshots

`zfs rollback <dataset>@<snap>`
Revert to a snapshot

`zfs destroy <dataset>@<snap>`
Delete a snapshot

`zfs clone <dataset>@<snap> <pool>/<new>`
Writable copy of a snapshot

SNAPSHOTS AND CLONES (cont.)

Browse old versions without rolling back: read-only copies sit under <mountpoint>/.zfs/snapshot/<name>/.

SEND AND RECEIVE

`zfs send <dataset>@<snap> | zfs recv <pool>/<name>`
Copy a snapshot locally

`zfs send <dataset>@<snap> | ssh <host> zfs recv <pool>/<name>`
Replicate to another machine

`zfs send -i <old> <new> | zfs recv <ds>`
Send only changes between snapshots

ENCRYPTION

`zfs create -o encryption=on -o keyformat=passphrase <pool>/<name>`
Create an encrypted dataset

`zfs load-key <dataset>`
Unlock a dataset