ZFS Recover files from Volume Snapshot

If you want to recover an entire volume from a snapshot, the command is simple and fast:

```
zfs rollback dataset/path@snapshot
```

. This will take a snapshot and make it the active dataset (removing all intermediaries, I think).

However, in my case, I needed some configuration files off a snapshot (major failure in backup configuration). I did not want the old snapshot rolled back; there were changes we needed to keep.

With dataset snapshot, you can go into the (hidden) snapshot directory, usually *.zfs*, but with a volume, it is just that, a volume. A block of disk.

Ok, I knew the disk was an ext4 file system and it had one partition. I could look at the current copy this was a snapshot of.

Turns out, it was fairly simple, except for the whole ext4 thing (which BSD is crap at). Just clone the snapshot, do whatever you need to do to mount it, get your information off, umount it, then destroy the clone. Following are the steps I took, including the ext4 stuff (using ext2, and read only).

```
zfs clone path/to/snapshot path/to/new/name
ls /dev/zvol/path/to/new
# look at the names. partitions will have s1, s2 at the end for the slices
# so, volume 'new' with two slices (partitions) will have entries for
# /dev/zvol/path/to/new, /dev/zvol/path/to/news1, /dev/zvol/path/to/news2
# with the last two being slices of on the drive
mkdir /mnt/ext4_mount
# must use fusefs. FreeBSD has horrible ext2/4 support, but fuse works just
fine
pkg install fusefs-ext2
# load the library
kldload fusefs
# use fuse-ext2 to mount the partition
fuse-ext2 /dev/zvol/path/to/newslice /mnt/ext4_mount/
# do your stuff, then clean up
umount /mnt/ext4_mount
# make very sure you destroy the clone, not anything else.
zfs destroy path/to/new/name
```

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/unix/freebsd/zfs_volume_recover

Last update: **2025/08/24 08:03**