HOWTO create a file-backed mirror of proxmox rpool as backup - PROOF OF CONCEPT

2024.Jun kneutron

NOTE these instructions are for proxmox-specific ZFS boot/root. I recommend if you are booting from SSD that you make the backup to spinning media, in order to avoid wear on the backup target.

\*\*\* WARNING - not for "noobs" - this is Advanced Sysadmin level stuff, and assumes you have a pretty good knowledge of ZFS and the possible risks \*\*\* You should already know what you are trying to accomplish (Disaster Recovery) by making this kind of backup! IF YOU ARE NOT SURE OF WHAT YOU ARE DOING, ASK FOR HELP.

To begin - we are only mirroring the ZFS partition to a file for simplicity, and will **dd** the efi partition separately to a gzipped file. Otherwise it would probably involve **kpartx**, or doing something similar to an iscsi disk if you wanted to copy all of the partitions on the boot/root disk. (You can try, but that is an exercise best left to others and is beyond the scope of this document.)

On your Proxmox server -

Mount a destination/target over Samba or **sshfs** - in this case it is /Volumes/sgtera2 on my Mac (non-zfs FS) - you could also set the target directory to a local XFS or ext4 filesystem.

**mkdir -pv /mnt/macpro-sgtera2**

**sshfs dave@macpro-static:/Volumes/sgtera2 /mnt/macpro-sgtera2**

The proxmox rpool (boot/root) thumbdrive I am making a backup of is "64GB"

**cd /mnt/macpro-sgtera2**

**fdisk -l /dev/sdb >fdisk-l-pve-portable-zfs-bootroot-64gig-thumbdrive.txt**

Disk /dev/sdb: 57.77 GiB, 62026416128 bytes, 121145344 sectors

Disk model: USB DISK 3.0

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: gpt

Device Start End Sectors Size Type

/dev/sdb1 34 2047 2014 1007K BIOS boot

/dev/sdb2 2048 1050623 1048576 512M EFI System \*\* dd this EFI to compressed file

/dev/sdb3 1050624 88080384 87029761 41.5G Solaris /usr & Apple ZFS

/dev/sdb4 88080386 121145309 33064924 15.8G Linux LVM

NOTE the LVM partition is skipped in this HOWTO.

Create a sparse file of sufficient size to mirror the sdX3 ZFS partition:

**truncate -s 58G rpool-mirror-64gig-thumbdrive-zfs-efi.disk** # on sshfs, samba mount, or local ext4/XFS - try to avoid zfs-on-zfs

Backup the EFI partition to a lightly gzipped file (for speed):

**time dd if=/dev/sdb2 bs=1M status=progress |gzip -1 >dd-efi-part2-rpool-mirror-64gig-thumbdrive-zfs.dd.gz**

zpool status -v

pool: rpool

state: ONLINE

config:

NAME STATE READ WRITE CKSUM

rpool ONLINE 0 0 0

usb-\_USB\_DISK\_3.0\_070D37CDAEC26025-0:0-part3 ONLINE 0 0 0

errors: No known data errors

It is probably best to do this on local xfs/ext4-backed file, but we can also do it on a network mounting. Attach the file-based disk to the rpool and let it resilver to make an easy copy:

/mnt/macpro-sgtera2 # **time zpool attach rpool usb-\_USB\_DISK\_3.0\_070D37CDAEC26025-0:0-part3 \**

**$PWD/rpool-mirror-64gig-thumbdrive-zfs-efi.disk**

# zfs-watchresilver

https://github.com/kneutron/ansitest/blob/master/ZFS/zfs-watchresilver-boojum.sh

NOTE - DO NOT USE ZPOOL ADD, this is a common mistake. Attach creates a mirror, add leaves you with a striped RAID0 with no redundancy!

Pool: rpool - scrub started: Fri Jun 7 05:32:19 PM MDT 2024

pool: rpool

state: ONLINE

scan: scrub repaired 0B in 00:01:17 with 0 errors on Fri Jun 7 17:33:39 2024

config:

NAME STATE READ WRITE CKSUM

rpool ONLINE 0 0 0

mirror-0 ONLINE 0 0 0

usb-\_USB\_DISK\_3.0\_070D37CDAEC26025-0:0-part3 ONLINE 0 0 0

/mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk ONLINE 0 0 0

errors: No known data errors

o Scrub rpool start: Fri Jun 7 05:32:19 PM MDT 2024 // Completed: Fri Jun 7 05:33:46 PM MDT 2024

Since this is on a network mount, you may have to issue a ' **zpool clear rpool** ' for any CKSUM errors.

Advanced users will probably be thinking about now, "Ok, how do we keep this copy updated?" You can look into zfs send/recv if you want; for simplicity's sake I would recommend just doing a full copy/resilver every time and maybe update it every couple of months. You can use my recommended bkpcrit script on a nightly basis to make a current backup of critical files to separate media / NAS.

https://github.com/kneutron/ansitest/tree/master/proxmox

After the resilver is finished, take the mirror file-based copy offline:

**zpool offline rpool /mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk**

**zpool status -v**

pool: rpool

state: DEGRADED

status: One or more devices has been taken offline by the administrator.

Sufficient replicas exist for the pool to continue functioning in a

degraded state.

action: Online the device using 'zpool online' or replace the device with

'zpool replace'.

scan: scrub repaired 0B in 00:01:17 with 0 errors on Fri Jun 7 17:33:39 2024

config:

NAME STATE READ WRITE CKSUM

rpool DEGRADED 0 0 0

mirror-0 DEGRADED 0 0 0

usb-\_USB\_DISK\_3.0\_070D37CDAEC26025-0:0-part3 ONLINE 0 0 0

/mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk OFFLINE 0 0 0

errors: No known data errors

At this point you can **reboot** and (as long as the target containing the rpool\*efi.disk isn't mounted) you can detach the efi.disk mirror copy from the pool without issues, the data on the backup file will stay intact.

**zpool detach rpool /mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk**

NOTE The target mountpoint should NOT be mounted at this time. If zfs can still "talk to" that file, I cannot guarantee it will stay usable after an "active" detach.

================

PART 2 - RESTORE in a Disaster Recovery situation (get back up and running quick)

Reconstructing your proxmox rpool from the image file (single-disk pool died, or both drives in the mirror are toast) and getting it to boot:

Boot into a non-zfs root Linux system (systemrescuecd with zfs should do, or you can try the Proxmox installer ISO but I have not tested this) - my recovery process was actually done from SuSE with ReFind installed.

https://github.com/nchevsky/systemrescue-zfs/releases

[https://www.rodsbooks.com/refind/installing.html#installsh](https://www.rodsbooks.com/refind/installing.html" \l "installsh)

From your recovery environment, remount where you put the file-backed copy of the rpool partition.

**mkdir -pv /mnt/macpro-sgtera2**

**sshfs dave@macpro-static:/Volumes/sgtera2 /mnt/macpro-sgtera2**

# or mount it locally, or via CIFS/Samba/NFS, whatever suffices.

We need to mount the rpool but WITHOUT mounting any datasets, or it might over-mount our current rootfs:

**cd /mnt/macpro-sgtera2; zpool import -f -N -d $PWD rpool** # without mounting any datasets

Breakdown of options:

**-f** # Force - likely necessary since we offlined this disk and it wasn't "exported properly"

**-N** # Do not mount zfs datasets, just import this pool

**-d $PWD** # Look in the current directory for the pool disk(s)

zpool status -v

pool: rpool

state: DEGRADED

status: One or more devices could not be used because the label is missing or

invalid. Sufficient replicas exist for the pool to continue

functioning in a degraded state.

action: Replace the device using 'zpool replace'.

see: https://openzfs.github.io/openzfs-docs/msg/ZFS-8000-4J

scan: scrub repaired 0B in 00:01:17 with 0 errors on Fri Jun 7 17:33:39 2024

config:

NAME STATE READ WRITE CKSUM

rpool DEGRADED 0 0 0

mirror-0 DEGRADED 0 0 0

2798824641078101509 UNAVAIL 0 0 0

was /dev/disk/by-id/usb-\_USB\_DISK\_3.0\_070D37CDAEC26025-0:0-part3

 /mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk ONLINE 0 0 0

errors: No known data errors

Now we need to detach the original bootable zfs disk from this file-backed pool (since it's probably dead)

**zpool detach rpool 2798824641078101509**

The new target disk is sdc below - I have cheated a bit here for the sake of convenience, by **dd**'ing the first 513MB from the original rpool disk to sdc after creating a new GPT partition table on it. Then went in with **gdisk** and corrected the partition table layout. You can recreate the partition table with **parted** or similar - example here, look for the **sgdisk** stanza:

https://github.com/kneutron/ansitest/blob/master/proxmox/proxmox-replace-zfs-mirror-boot-disks-with-bigger.sh

sgdisk -g \

-n 1:0:+1M \

-n 2:0:+1G \

-n 3:0:0 \

-t 1:8300 \

-t 2:EF00 \

-t 3:BF01 \

-p /dev/$disk

The -n options are sizes, and the -t sets the filesystem type.

If you don't want to cheat, you should restore the gzipped EFI partition at this point after recreating the partition table.

**gzip -cd** **dd-efi-part2-rpool-mirror-64gig-thumbdrive-zfs.dd.gz |dd of=/dev/sdc2 bs=1M status=progress**

MAKE SURE you specify the correct destination disk and partition! I am not responsible for data loss!

NOTE I have not bothered to re-size the ZFS partition on the replacement disk to take advantage of the increased disk space, since this is just testing the process.

Tips for resizing here:

https://sirlagz.net/2023/07/03/updated-live-resize-lvm-on-linux/

**fdisk -l /dev/sdc**

Disk /dev/sdc: 465.76 GiB, 500107862016 bytes, 976773168 sectors

Disk model: Portable SSD T5

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 33553920 bytes

Disklabel type: gpt

Disk identifier: 589F8C89-336B-44A9-A2CB-3173BD9887A0

Device Start End Sectors Size Type

/dev/sdc1 34 2047 2014 1007K BIOS boot

/dev/sdc2 2048 1050623 1048576 512M EFI System

/dev/sdc3 1050624 88080384 87029761 41.5G Solaris /usr & Apple ZFS

This will temporarily mirror the file-backed disk to the new boot disk at partition 3:

**zpool attach rpool /mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk \**

**ata-Samsung\_Portable\_SSD\_T5\_S49WNV0MC04217F-part3**

watchresilver

Pool: - NOW: Fri Jun 7 22:16:28 MDT 2024 -- Watchresilver started: Fri Jun 7 22:12:34 MDT 2024

pool: rpool

state: ONLINE

scan: resilvered 3.48G in 00:04:03 with 0 errors on Fri Jun 7 22:16:20 2024

config:

NAME STATE READ WRITE CKSUM

rpool ONLINE 0 0 0

mirror-0 ONLINE 0 0 0

/mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk ONLINE 0 0 0

ata-Samsung\_Portable\_SSD\_T5\_S49WNV0MC04217F-part3 ONLINE 0 0 0

errors: No known data errors

Now we take the file-backed copy offline, leaving it intact for later use if needed - NOTE you do NOT want to issue a **detach** or a **zpool split** at this point because this is an rpool, and a detach will not guarantee usable data on the detached disk. We just offline it, and reboot to get out from under it (so to speak.)

**zpool offline rpool /mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk**

# **reboot**

After rebooting, we get dumped into the initramfs because ZFS has temporarily lost its mind and can't import from the cache file properly.

Easy fix:

(initramfs) **zpool import -f**

Then hit **Control+D**, boot process resumes and should survive further reboots.

Finally, without the target/backup being mounted, we detach the file-backed copy from the rpool to get it out of DEGRADED state:

**zpool detach rpool /mnt/macpro-sgtera2/rpool-mirror-64gig-thumbdrive-zfs-efi.disk**

And we're back to a bootable single-disk Proxmox rpool.