

ZFS /home Partition/Drive

This tutorial/HOWTO is about installing/setup/using ZFS (file system, fs) such as for your /home partition/drive, for advanced/large storage or shared with a Unix operating system (OS) distribution (distro). Feel free to expand.

ZFS Introduction

ZFS is a journalling fs created on Solaris Unix, then made Free/Libre/Open-source Software (FLS, OSS, FOSS, FLOSS) as OpenSolaris' [OpenZFS](#). ZFS was ported to classic/most *BSD Unix (NetBSD, FreeBSD, OpenBSD, ...), Slackware & other GNU/Linux distros, as [OpenZFS](#), and has more features than EXT*. ZFS is more a fs & RAID manager than just fs. If you have large amount of storage, you may want ZFS for that (and helpful features, such as compression, snapshots) or may want to use it for a /home partition (also possible to share with a real Unix, but some things, such as more complex X window managers (WM) & desktop environments (DE), and some/many their programs, almost certainly won't work well between them... arecordednd there's a real chance some shells & command-line programs might not either, if different configurations/versions/variants).

If you just want multi-/dual-boot OS (including Unix) with same /home, ZFS is the only realistic solution... another maybe being NTFS, but has certain problems, such as maybe no to use Unix-like, rather than Windows-like, file-names so also utilities, and ZFS is safer. ZFS may be the only journalling fs Unix and GNU/Linux can both write to safely.

Of course, Xen (a hypervisor virtual machine) is another excellent option for multiple native-on-hardware OS, including Unix, but you might just have to copy between fs and not all OS write well... so you could even try Xen with ZFS storage...).

Always Backup /home Regularly. Make the ZFS package

Here's how you can prepare to create a ZFS /home.

If you have a current /home with files you'll use, *make a backup* (preferably more than one)!

Build/install `openzfs` from [SlackBuilds.org](#) (SBo, such as with `sbopkg`, available on [SBopkg.org](#)). If you update Linux (kernel) *you must build openzfs again!* (before you can access ZFS).

If you created any new /home/user initial/'skeleton' directory on the root filesystem partition, you can move it to /root temporarily, because it's just going to get mounted over (not seen) but still take space (inaccessible until unmounting). One might have a new /home if using a Unix and Slackware, so later you'll want any skeleton directory (default user files like `.*shrc`, `.profile`, etc. for that OS distro.)

Research/read At Least A Little

Before first actual ZFS commands, you may want to view `/usr/doc/openzfs-*`, `cat` or more (or less,

most) /var/log/packages/openzfs-*, 'man zpool,' 'man zfs.' A zpool holds and manages one or more zfs and has many options, which for advanced/large storage, you will want some/all (maybe depending on compatibility with other systems accessing)... that case would be for new sections or advanced storage HOWTO... here it's assumed one may not use most/any extra options because might not be a data-centre (but one feature will be mentioned) rather than wanting to share /home with a Unix, which many may not have all options (as OpenSolaris/IllumOS' ZFS and OpenZFS are slightly different-at different stages of overall ZFS design, so some versions have things older ones don't, and won't even be readable with... you should check/view those project sites (linked here sooner or later), even though OpenZFS people didn't bother to address the most long-time/influential GNU/Linux-Slackware-some/many command/option examples for others-even if not Unix-like-will be similar/same).

Create & Start ZFS

Once you installed ZFS and all various /home files are backed up, start ZFS with rc script (run it or reboot)... login root (or su -) or use sudo for those commands below. Remember, this will be with no extra options (-d option is 'no options' unless specified) other than use a 'modern' sector size (unless you have a drive so old it should probably be replaced if even still working, this size should be reasonable)... '-o ashift=12' option creates 2¹², i.e., 4KB sectors (you shouldn't actually use larger/8KB unless using *older* solid state drive, SSD designed 8KB). You can read zpool & zfs manpages to find/add more options. If you do and share a /home, install ZFS on both OSes, dump zpool option list sets on each to see what's usable (only the [set intersection](#)). Most likely both will have compression and snapshots, so if you want those, read examples/manpages.

Below is to create a ZFS drive (replace L with drive block device letter you want).

```
zpool create -d -o ashift=12 home /dev/sdL
```

Alternatively, if not using an entire drive (but normally a Solaris GPT partition) below is to create a ZFS partition (replace L,N with drive partition block device letter,number you want).

```
zpool create -d -o ashift=12 home /dev/sdLN
```

Start/mount zpool/ZFS.

```
/etc/rc.d/rc.zfs start
zpool import home
```

Your ZFS should now be ready. However, you can create multiple ZFS (even levels/hierarchies, 'datasets') in each zpool, such as for swap, each user, other/server purposes (SBo's openzfs maintainer could explain datasets if you ask, and documented by Oracle, OpenZFS for various distros, usually similar/same commands). Move back any Slackware & Unix/etc. initial/skeleton user directories and/or (maybe after installing/testing ZFS on other/optional OS sharing it, first, just to make sure it'll work, before a potentially hours-long copy/restore/rysnc/etc. back) your /home from the backups. You will probably want something such as the following.

```
#!/bin/sh
#/etc/rc.d/rc.local
#... (your other lines)
```

```
#Activate zpool & its zfs.  
#If only using on one OS, you shouldn't need the '-f.'  
zpool import -f home  
# ON LINES BELOW, YOU CAN START ALL PROCESSES ACCESSING THAT FS.  
  
#... (your other lines)
```

Speed Up ZFS

If you don't need access time (atime) logged for files, disable atime to speed up ZFS.

```
zfs set atime=off home
```

How to Stop ZFS

```
#!/bin/sh  
#/etc/rc.d/rc.local_shutdown  
#... (your other lines)  
  
# Unmount/stop ZFS. may not actually need this if rc.zfs iss stopped  
automatically and does it, but be safe or find out.  
# ON LINES ABOVE, STOP ALL PROCESSES ACCESSING THIS FS.  
# Can be either unmount or umount.  
zfs unmount home  
  
#... (your other lines)
```

If after Slackware, you initially managed to read & write to a ZFS on a Unix, then copied your /home back on Slackware, you may soon see messages and think it's not going to mount on one or the other, but as rc.local above, you will probably actually have to 'zpool -f home' on both OS. Some Unixes do '-f' automatically if you enable to merely use ZFS software as an option in a configuration file.

Repairs/restarts

If you mess up making a ZFS or want to delete one, it can be hard. If you want to delete one, you can't just delete and overwrite, but must wipefs, or just dd 4MB at beginning & end to zero and remove its zpool cache... those can take a long time, but on a SSD, you can also just use the tool like the Unix trim, that is blkdiscard.

Sources

* Originally written by [David Chmelik](#)

[howtos](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

https://docs.slackware.com/howtos:zfs_home

Last update: **2024/02/12 14:12 (UTC)**

