```
=======================
ZFS - Zetabyte File System
=======================
Note: ZFS filesystems need a ZFS pool.  When you create a ZFS pool, a
ZFS filesystem is automatically created.

ZFS Pools
-----------
List Pools: zpool list
Create a single pool: zpool create <name> <path>
Destroy a pool: zpool destroy <name>
Check for pool errors: zpool scrub <name>
Show data allocation: zpool iostat -v <name>
Detach device from a pool: zpool detach <name> <path>
Attach device to a pool: zpool attach <name> <path>

ZFS Mirrored Pools
-----------------------
Create a mirrored pool: zpool create <name> mirror <path1> <path2>
Add to mirrored pool: zpool add <name> mirror <path1> <path2>

ZFS Filesystems
-------------------
Show ZFS information: zfs list <name>
Show ZFS dataset information: zfs list -r <dataset>
Show sharefs and mount information: zfs list -o <dataset>
Specify dataset types to display: zfs list -t <dataset type> <dataset>
  - Dataset Types: filesystem, volume, snapshot (comma delimited)
Other zfs parameters:
-H #: Omit the header and replace whitespace with <tab>s
Create ZFS filesystems within a pool:  zfs create <pool>/
<new_zfs_name>
Destroy a ZFS filesystem: zfs destroy <pool>/<zfs_name>
Show Space usage: zfs list -o space <pool>/<zfs_name>
Show Space usage with snapshots: zfs list -ro space <pool>/<zfs_name>

ZFS Mount Points
--------------------
Change default mount point: zfs set mountpoint=<path> <zfs_name>
Mount ZFS point: zfs mount <pool>/<zfs_name>
Unmount ZFS point: zfs unmount <pool>/<zfs_name>

ZFS Filesystem properties
------------------------------
Get current properties: zfs get <properties> <all> <pool>/<zfs_name>
Parameters:
* -r               recursively get all child filesystem properties
* -p              reports exact values  i.e. 9M, 1G
* -H              omits header fields for script parsing
* -o <fields> specify output fields
```

Output:

```
* -             read only file system information
* default     the property is the ZFS default value
* local        property is set directly on this filesystem
* inherited  the property is inherited from a parent filesystem
```

Set properties: zfs set <parameters> <pool>/<zfs_name>

ZFS Quotas and Reservations: Note: zfs get can use properties alone to filter results

```
- Quota=<size_in_M, size_in_G>           Limit file system size
- Reservation=<size_in_M, size_in_G>   Ensure <size> is always available
```

ZFS Snapshots

List snapshots: zpool get listsnapshots <pool> -OR- zfs list -t snaphost -o <fields> <pool>/<zfs_name>

Create a snapshot: zfs snapshot <pool>/<zfs_name>@<snapshot_name>

-r == Recursively snapshot or rename all descendent filesystems

Destroy a snapshot: zfs destroy <pool>/<zfs_name>@<snapshot_name>

Rename a snapshot: zfs rename <pool>/<zfs_name>@<snaphost_name> <pool>/<zfs_name>@<new_name>

NOTE: Snapshots are read-only. Parameters cannot be set. Snapshots are stored in the .zfs directory within the ZFS' root directory

ZFS Clone

Create a clone: zfs clone <pool>/<zfs_name>@<snapshot_name> <pool>/<clone_name>

Destroy a clone: zfs destroy <pool>/<clone_name>

Replace ZFS filesystem with a ZFS clone: zfs prompte <pool>/<clone_name>

NOTE: You can only clone from a snapshot. You can't clone to a different pool.

NOTE-2: A promoted clone causes the original filesystem to become the clone, allowing you to destroy or rename the original file system.