































?? 1: ZFS ??

ZFS은 FreeBSD에서 개발된 파일 시스템으로, 데이터 무결성과 성능을 극대화하기 위해 설계되었습니다. ZFS은 RAID-Z와 유사한 RAID 구성을 지원하며, 데이터 중복성을 높여줍니다. 또한, ZFS은 snapshots과 clones 기능을 제공하여 데이터 백업과 복구 작업을 용이하게 합니다.

ZFS 的 目 的 是 在 存 儲 系 統 中 實 現 高 效 的 存 儲 和 備 份 功 能 。 它 是 基 於 操 作 系 統 的 存 儲 系 統 ， 可 以 在 多 個 操 作 系 統 上 運 行 ， 如 Linux 、 FreeBSD 10.1 等 。 ZFS 的 特 點 是 具 有 高 效 的 存 儲 和 備 份 功 能 ， 可 以 在 多 個 操 作 系 統 上 運 行 ， 如 Linux 、 FreeBSD 10.1 等 。 ZFS 的 特 點 是 具 有 高 效 的 存 儲 和 備 份 功 能 ， 可 以 在 多 個 操 作 系 統 上 運 行 ， 如 Linux 、 FreeBSD 10.1 等 。

ZFS ??? ?? (ZFS Datasets)

[illegible]

```
zfslist  2  ZFS  5  2  3  5 .
```

```
$ zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
zroot	429M	13.0G	96K	none
zroot/ROOT	428M	13.0G	96K	none
zroot/ROOT/default	428M	13.0G	428M	/
zroot/tmp	104K	13.0G	104K	
/tmp zroot/usr		428K	13.0G	96K /usr

```
***
```

```
mount(8) df(1)  , UFS  extfs  
```

```

[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] . ZFS [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] ZFS storage pool, [ ]
zpool[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] . [ ] [ ] [ ] [ ] [ ] [ ] zroot[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] root dataset[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] .

```


我们 **/var/log** 我们 在 我们 我们 我们 我们 我们 我们 我们 我们 我们 ,
我们 ZFS 我们 我们 我们 我们 我们 .

我们 我们 我们 在 我们 我们 ZFS 我们 在 我们 . 我们 我们 , **quota** 我们 我们 我们
我们 我们 在 我们 我们 在 ZFS 我们 我们 我们 我们 我们 我们 . ZFS 我们
我们 **zfs(8)** 我们 .

```
$ zfs set quota=2G zroot/var/log
```

zfs get 我们 我们 我们 .

```
$ zfs get quota zroot/var/log
```

NAME	PROPERTY	VALUE	SOURCE
zroot/var/log	quota	2G	local

zfs get all 我们 ZFS 我们 我们 我们 我们 我们 在 我们 .
4我们 ZFS 我们 我们 我们 , 6我们 我们 我们 我们 我们 我们 .

ZFS ??(ZFS Limits)

我们我们 我们 我们 我们 我们 我们 . 我们 我们 我们 我们 FAT 我们 我们 我们
32MB 我们 我们 我们 我们 我们 , 我们 2GB, 4GB 我们 . 我们 FAT32
2TB 我们 我们 我们 我们 . UFS 我们 ext2/3/4fs 我们 我们 我们 我们
. 我们 我们 我们 我们 我们 我们 我们 , 我们 在 我们 我们 我们
我们 我们 我们 我们 我们 . 我们 我们 我们 我们 我们 我们 我们
我们 我们 我们 我们 我们 我们 我们 我们 我们 .

ZFS 我们 ZFS 我们 我们 我们 我们 我们 我们 , 我们 我们 我们 . ZFS
在 我们 我们 我们 128我们 我们 我们 我们 我们 我们 我们
我们 我们 在 我们 我们 我们 . 我们 我们 我们 16我们 2^{48}
我们 我们 在 我们 . 我们 我们 256我们 , 我们 2^{78} 我们 我们 . 我们 我们
我们 2^{64} 我们 我们 我们 在 我们 , 我们 我们 我们 2^{64} 我们 我们 我们 在 我们 .

我们 我们 我们 我们 我们 我们 我们 我们 我们 我们 . 我们 我们 我们 我们
在 我们 我们 我们 我们 我们 我们 我们 . 我们 ZFS 我们 我们
我们 在 我们 我们 我们 我们 我们 我们 . 我们 我们 我们
FAT/UFS/extfs 我们 在 我们 我们 .

???? ?

ZFS 我们 我们 我们 我们 我们 . 我们 我们 我们 我们 我们 我们
我们 , 我们 我们 在 我们 我们 在 我们 我们 我们 我们 我们 我们 .
zpool(8) 我们 我们 我们 我们 我们 我们 在 我们 . 我们 我们 FreeBSD 我们
我们 .

\$ zpool status

pool: zroot

state: ONLINE

scan: none requested

config:

NAME	STATE	READ	WRITE	CKSUM
zroot	ONLINE	0	0	0
gpt/zfs0	ONLINE	0	0	0

errors: No known data errors

在 ZFS 中，数据块被存储在池中的设备。ZFS 池由一个或多个设备组成，每个设备都有自己的元数据。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

在 ZFS 中，数据块被存储在池中的设备。ZFS 池由一个或多个设备组成，每个设备都有自己的元数据。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

?? ?? (Virtual Devices)

在 ZFS 中，数据块被存储在池中的设备。ZFS 池由一个或多个设备组成，每个设备都有自己的元数据。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

在 ZFS 中，数据块被存储在池中的设备。ZFS 池由一个或多个设备组成，每个设备都有自己的元数据。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

在 ZFS 中，数据块被存储在池中的设备。ZFS 池由一个或多个设备组成，每个设备都有自己的元数据。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。ZFS 池的元数据存储在池的根目录下。

