

---

# Demo Script

## Table of Contents

VM Setup Configuration .....	1
Dynamic Mirroring and Reconfiguration .....	1
RaidZ, Snapshots, Clones .....	3
Silent Corruption, Self healing .....	6
Replication and CDP .....	6

## VM Setup Configuration

Configure virtual machine with a six 2Gig disks in addition to the primary boot disk. These additional disks should show up as /dev/sdb, sdc, sdd, sde, sdf, sdg.

## Dynamic Mirroring and Reconfiguration

1. Check machine is running linux

```
[root@localhost spl]# uname -a
Linux localhost.localdomain 2.6.31.5 #9 SMP Wed May 5 15:59:16
IST 2010 x86_64 x86_64 x86_64 GNU/Linux
```

2. List all the disks available to us

### Note

fdisk warns that GPT (GUID Partition Table) is not supported. That is not a problem since the entire disk is managed by zfs

```
[root@localhost ~]# fdisk -l
```

```
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00051c07
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	3560	28588038+	83	Linux
/dev/sda2		3560	3916	2867200	82	Linux swap / Solaris

WARNING: GPT (GUID Partition Table) detected on '/dev/sdc'! The util fdisk doesn

```
Disk /dev/sdc: 2147 MB, 2147483648 bytes
256 heads, 63 sectors/track, 260 cylinders
Units = cylinders of 16128 * 512 = 8257536 bytes
Disk identifier: 0x00000000
```

```
Device Boot      Start          End      Blocks   Id  System
/dev/sdc1        1            261      2097151+  ee  GPT
Partition 1 has different physical/logical beginnings (non-Linux?):
    phys=(1023, 255, 63) logical=(0, 0, 2)
Partition 1 has different physical/logical endings:
    phys=(1023, 255, 63) logical=(260, 16, 16)
WARNING: GPT (GUID Partition Table) detected on '/dev/sde'! The util fdisk doesn't
```

```
Disk /dev/sde: 2147 MB, 2147483648 bytes
256 heads, 63 sectors/track, 260 cylinders
Units = cylinders of 16128 * 512 = 8257536 bytes
Disk identifier: 0x00000000
```

```
Device Boot      Start          End      Blocks   Id  System
/dev/sde1        1            261      2097151+  ee  GPT
Partition 1 has different physical/logical beginnings (non-Linux?):
    phys=(1023, 255, 63) logical=(0, 0, 2)
Partition 1 has different physical/logical endings:
    phys=(1023, 255, 63) logical=(260, 16, 16)
[root@localhost ~]#
```

### 3. Check zfs module is loaded correctly

```
[root@localhost ~]# lsmod |grep zfs
lzfs                32960  0
zfs                  702928  1 lzfs
zcommon              28864  1 zfs
zunicode             320256  1 zfs
znvpair              37088  2 zfs,zcommon
zavl                  6624  1 zfs
spl                  106200  6 lzfs,zfs,zcommon,zunicode,znvpair,zavl
zlib_deflate         19784  1 zfs
```

### 4. Clean up stale pools from previous demo's if any

```
[root@localhost ~]# zpool import -a
[root@localhost ~]# zpool list
NAME      SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
demopool  1.98G  113K  1.98G    0%  ONLINE  -
[root@localhost ~]# zpool destroy demopool
```

### 5. Create a demopool using sdb1

```
[root@localhost ~]# zpool create demopool sdb1
[root@localhost ~]# zpool list
NAME      SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
demopool  1.98G   74K  1.98G    0%  ONLINE  -
[root@localhost ~]# mount
/dev/sda1 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
```

```
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
demopool on /demopool type zfs (rw)
```

6. Copy some data into the pool

```
[root@localhost ~]# cp /etc/passwd /demopool/
```

7. The pool we have created has no redundancy. If sdb1 fails all data is lost. Lets add another disk as a mirror.

```
[root@localhost ~]# zpool attach demopool sdb1 sdc1
[root@localhost ~]# zpool status
  pool: demopool
 state: ONLINE
  scrub: resilver completed after 0h0m with 0 errors on Mon Nov  1 09:37:02 2010
config:
```

NAME	STATE	READ	WRITE	CKSUM	
demopool	ONLINE	0	0	0	
mirror	ONLINE	0	0	0	
sdb1	ONLINE	0	0	0	
sdc1	ONLINE	0	0	0	75K resilvered

```
errors: No known data errors
```

8. Lets verify the resilvering was really done. Remove the original disk from the pool then unmount/mount the filesystem and verify the data is really there.

```
[root@localhost ~]# zpool detach demopool sdb1
[root@localhost ~]# zfs unmount -a
[root@localhost ~]# zfs mount demopool
[root@localhost ~]# mount
demopool on /demopool type zfs (rw)
[root@localhost ~]# head /demopool/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

## RaidZ, Snapshots, Clones

1. Clean up the pools from the previous demo

```
[root@localhost ~]# zfs unmount -a
[root@localhost ~]# zpool destroy demopool
```

2. Create a raidz pool using disks sdb1 to sde1

```
[root@localhost ~]# zpool create demopool raidz sdb1 sdc1 sdd1 sde1
```

```
[root@localhost ~]# zpool status
pool: demopool
state: ONLINE
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
demopool	ONLINE	0	0	0
raidz1	ONLINE	0	0	0
sdb1	ONLINE	0	0	0
sdcl	ONLINE	0	0	0
sddl	ONLINE	0	0	0
sdel	ONLINE	0	0	0

errors: No known data errors

3. Lets copy some non-trivial amount of data. /usr/include is about 138M.

```
[root@localhost ~]# cp -r /usr/include/ /demopool/
[1] 3726
[root@localhost ~]# zpool iostat -v 10
```

pool	capacity		operations		bandwidth	
	used	avail	read	write	read	write
demopool	85.7M	7.85G	10	478	11.7K	2.61M
raidz1	85.7M	7.85G	10	478	11.7K	2.61M
sdb1	-	-	2	80	166K	944K
sdcl	-	-	1	73	109K	932K
sddl	-	-	2	80	166K	944K
sdel	-	-	1	75	109K	930K

pool	capacity		operations		bandwidth	
	used	avail	read	write	read	write
demopool	136M	7.81G	0	501	50	2.58M
raidz1	136M	7.81G	0	501	50	2.58M
sdb1	-	-	0	35	0	928K
sdcl	-	-	0	32	0	924K
sddl	-	-	0	34	0	926K
sdel	-	-	0	32	0	922K

```
[root@localhost ~]# ls /usr/include/
acl                gtk-2.0            pi-header.h
aio.h              gtk-unix-print-2.0 pi-hinote.h
aliases.h           gupnp-1.0          pi-inet.h
alloca.h            gypsy              pi-location.h
alsa                hal                 pi-macros.h
ansidecl.h          hesiod.h            pi-mail.h
```

4. Create a snapshot snap1 on the demopool

```
[root@localhost ~]# zfs snapshot demopool@snap1
[root@localhost ~]# zfs list -t all
NAME                USED  AVAIL  REFER  MOUNTPOINT
demopool            113M  5.73G  113M   /demopool
demopool@snap1      0      -    113M   -
```

5. Showsnapshot data via automount

```
[root@localhost ~]# cd /demopool/.zfs
[root@localhost .zfs]# ls
snapshot
[root@localhost .zfs]# cd snapshot/
[root@localhost snapshot]# ls
snap1
[root@localhost snapshot]# cd snap1/
[root@localhost snap1]# cd include/
[root@localhost include]# ls
acl                gtk-2.0            pi-header.h
aio.h              gtk-unix-print-2.0 pi-hinote.h
aliases.h          gupnp-1.0          pi-inet.h
alloca.h           gypsy             pi-location.h
alsa              hal               pi-macros.h
ansidecl.h         hesiod.h          pi-mail.h
a.out.h            hunspell          pi-md5.h
argp.h             ical.h           pi-memo.h
[root@localhost include]# cat /proc/mounts
rootfs / rootfs rw 0 0
...
demopool /demopool zfs rw,relatime,atime,noxattr,suid,devices,setuid,exec 0 0
demopool@snap1 /demopool/.zfs/snapshot/snap1 zfs ro,relatime,atime,noxattr,suid,
```

6. Lets remove all data from the primary and check if we can see the data in the snapshot

```
[root@localhost include]# cd /demopool/
[root@localhost demopool]# ls
include
[root@localhost demopool]# rm -rf include/
[root@localhost demopool]# cd .zfs/snapshot/snap1/include/
[root@localhost include]# ls
acl                gtk-2.0            pi-header.h
aio.h              gtk-unix-print-2.0 pi-hinote.h
aliases.h          gupnp-1.0          pi-inet.h
alloca.h           gypsy             pi-location.h
alsa              hal               pi-macros.h
```

7. We can create a clone from the snapshot. The data in the clone can be modified.

```
[root@localhost include]# zfs clone demopool@snap1 demopool/democlone
[root@localhost include]# mount
/dev/sda1 on / type ext4 (rw)
.....
demopool on /demopool type zfs (rw)
```

```
demopool/democlone on /demopool/democlone type zfs (rw)
[root@localhost include]# cd /demopool/democlone/
[root@localhost democlone]# mv include/ include-rw
[root@localhost democlone]# ls
include-rw
[root@localhost democlone]# rm -rf include-rw/
[root@localhost democlone]# ls
[root@localhost democlone]#
```

## Silent Corruption, Self healing

1. Lets introduce silent corrupt in the raidz pool created earlier by dd'ing zeros onto one of the disks.

```
[root@localhost democlone]# dd if=/dev/zero of=/dev/sdc1 count=20000
20000+0 records in
20000+0 records out
10240000 bytes (10 MB) copied, 0.0532465 s, 192 MB/s
```

2. Verify all data on the pool by running scrub. Scrub will checksum the data to detect corruption. The data will be reconstructed from raid disk

```
[root@localhost democlone]# zpool scrub demopool
[root@localhost democlone]# zpool status -v
  pool: demopool
  state: ONLINE
status: One or more devices has experienced an unrecoverable error. An
        attempt was made to correct the error. Applications are unaffected.
action: Determine if the device needs to be replaced, and clear the errors
        using 'zpool clear' or replace the device with 'zpool replace'.
       see: http://www.sun.com/msg/ZFS-8000-9P
scrub: scrub in progress for 0h0m, 84.71% done, 0h0m to go
config:
```

NAME	STATE	READ	WRITE	CKSUM
demopool	ONLINE	0	0	0
raidz1	ONLINE	0	0	0
sdb1	ONLINE	0	0	0
sdc1	ONLINE	0	0	1.86K 5.39M repaired
sdd1	ONLINE	0	0	0
sde1	ONLINE	0	0	0

```
errors: No known data errors
[root@localhost democlone]#
```

## Replication and CDP

1. Send data from snapshot snap1 into a new filesytem demopool/recv. Note the data is going through the pipe, the recv could have been a remote machine.

```
[root@localhost ~]# zfs send demopool@snap1 |zfs recv demopool/recv
```

2. Verify the data has arrived correctly

```
[root@localhost ~]# mount
/dev/sda1 on / type ext4 (rw)
.....
demopool on /demopool type zfs (rw)
demopool/democlone on /demopool/democlone type zfs (rw)
demopool/recv on /demopool/recv type zfs (rw)
[root@localhost include]# cd /demopool/recv/include/
[root@localhost include]# ls
acl                gtk-2.0            pi-header.h
aio.h              gtk-unix-print-2.0 pi-hinote.h
aliases.h          gupnp-1.0          pi-inet.h
alloca.h           gypsy             pi-location.h
alsa               hal                pi-macros.h
```