

How to install zfs on your Ubuntu

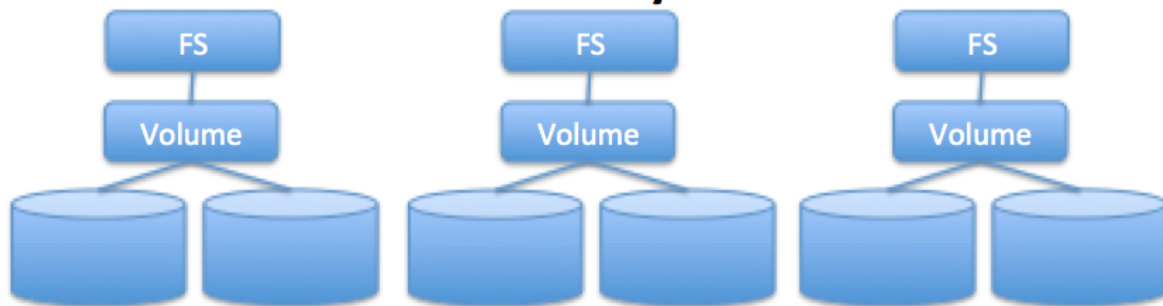
Jing, mqjing@gmail.com

google doc ([view](#))

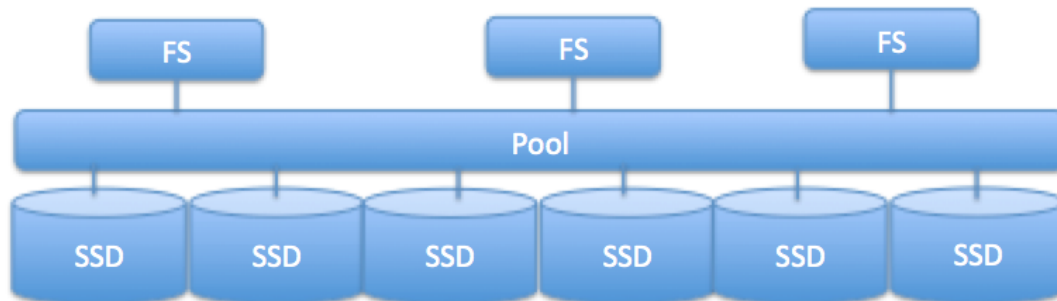
lab-zfs ([download](#))

- preinstall zfs
- allocate 4 disks with 500 GB

Classic File Systems



ZFS



- zfs 會保持每個 vdevs 剩餘空間比例全部都一樣

- 增加一個新的 vdev 到已經快滿的 zpool 中，會因為 zfs 為了保持每個 vdevs 相同剩餘空間比例，而把大部分新的資料都倒進讓新的 vdev，使得 I/O 全擠在一個 vdev 上，造成效能的損失。

- 一開始配置時，應該盡量讓每個 vdev 容量一樣。這樣才不會因為 zfs 為了保持每個 vdevs 相同剩餘空間比例，而把大部分資料都倒進容量比較大的那個 vdev 中，使得大多 I/O 全擠在一個 vdev 上，造成效能的損失。

- 不要加入一個只有一個硬碟的 vdev 到運行中的 zpool 中，這會讓大部分的資料寫在那個新加入的 vdev 中，悲慘的是那些資料只放在一個碟上，只要那顆硬碟出事，你的整個儲存堆疊就毀了。所以要加入一個 vdev 請加兩顆硬碟的組合並設定為彼此的 mirror。

- 你可以指定 每一個 zfs 屬性，例如：file system my source 指定壓縮，myimage file system 指定容量限制

- 你可以 snapshot 一個 filesystem，當你不小心誤刪了某個重要的檔案，可以重新 mount 舊的檔案系統回來，再取回重要的檔案。

- 你可以 clone 指定的 file system, 這個新的 filesystem 不佔任何空間, 除非你加了新東西。你可以利用 clone 的方式, 來處理 VM 的存放。基本款 VM 所在 filesystem + 一堆 clone VM

- 你可以使用 zfs send 指令, 將整個巨大 filesystem 的 snapshot 丟到到遠方的機器上做備份, 透過 -i 參數還可以做差異備份

Prerequisites and installation

Step 1: Install the package

```
sudo apt-get install python-software-properties
sudo apt-add-repository ppa:zfs-native/stable
sudo apt-get update
sudo apt-get install ubuntu-zfs
```

Step 2: Setup the Advanced Replacement Cache limitation

```
# /etc/modprobe.d/zfs.conf
#
# yes you really DO have to specify zfs_arc_max IN BYTES ONLY!
# 16GB=17179869184, 8GB=8589934592, 4GB=4294967296, 2GB=2147483648,
# 1GB=1073741824, 500MB=536870912, 250MB=268435456
#
options zfs zfs_arc_max=4294967296
```

Create a zfs Pool

Create a pool on files

(a) Create a pool on file

```
// step 1: create a image file
dd if=/dev/zero of=mystorage.img count=1024 bs=1M

// step 2: create a zfs pool on the file
sudo zpool create mypool /mystorage/mystorage.img
```

check

```
sudo zpool list
```

```
vagrant@node1:~$ sudo zpool list
```

NAME	SIZE	ALLOC	FREE	CAP	DEDUP	HEALTH	ALTROOT
mypool	1016M	123K	1016M	0%	1.00x	ONLINE	-

(b) Create a pool on three files

```
// step 1: create three image file
dd if=/dev/zero of=mystorage1.img count=1024 bs=1M
dd if=/dev/zero of=mystorage2.img count=1024 bs=1M
dd if=/dev/zero of=mystorage3.img count=1024 bs=1M

// step 2: create a zfs pool on the file
sudo zpool create mypool /mystorage/mystorage1.img /mystorage/mystorage2.img
/mystorage/mystorage3.img
```

check

sudo zpool list

```
vagrant@node1:~$ sudo zpool list
```

NAME	SIZE	ALLOC	FREE	CAP	DEDUP	HEALTH	ALTROOT
mypool	2.98G	129K	2.98G	0%	1.00x	ONLINE	-

Create a pool on disks

(a) dynamic strip

properties

- **no redundancy:** any disk failure will result in data loss.
- once a disk is added to the zfs pool, the device may not be removed the pool again

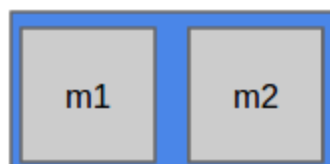
command

```
sudo zpool create -f mypool /dev/sdb /dev/sdc
```

```
vagrant@node1:~$ sudo zpool create -f mypool /dev/sdb /dev/sdc
vagrant@node1:~$ sudo zpool list
```

NAME	SIZE	ALLOC	FREE	CAP	DEDUP	HEALTH	ALTROOT
mypool	992G	126K	992G	0%	1.00x	ONLINE	-

(b) 2 way mirror (RAID-1)



[\(edit\)](#)

properties

- **redundancy 2**: two disks failure will result in data loss.
- You can detach a disk from the pool. The data will be available at the another disk

command

```
sudo zpool create mypool mirror /dev/sdb /dev/sdc
```

```
vagrant@node1:~$ sudo zpool create mypool mirror /dev/sdb /dev/sdc
vagrant@node1:~$ sudo zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
mypool    496G   123K   496G    0%   1.00x  ONLINE  -
vagrant@node1:~$ sudo zpool status
pool: mypool
state: ONLINE
scan: none requested
config:

    NAME            STATE        READ  WRITE  CKSUM
    mypool           ONLINE       0     0     0
    mirror-0         ONLINE       0     0     0
        sdb          ONLINE       0     0     0
        sdc          ONLINE       0     0     0

errors: No known data errors
```

(c) 3 way mirror



[\(edit\)](#)

properties

- **redundancy 3**: three disks failure will result in data loss.
- You can detach a disk from the pool. The data will be available at the another disk

command

```
sudo zpool create mypool mirror /dev/sdb /dev/sdc /dev/sdd
```

```
vagrant@node1:~$ sudo zpool create -f mypool mirror /dev/sdb /dev/sdc /dev/sdd
vagrant@node1:~$ sudo zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
mypool    496G   124K   496G    0%  1.00x  ONLINE  -
vagrant@node1:~$ sudo zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME            STATE        READ  WRITE  CKSUM
        mypool           ONLINE         0     0     0
          mirror-0       ONLINE         0     0     0
            sdb           ONLINE         0     0     0
            sdc           ONLINE         0     0     0
            sdd           ONLINE         0     0     0

errors: No known data errors
```

(d) RAIDZ

raidz1: 2 disks (single parity, RAID-5)

command

```
sudo zpool create mypool raidz1 /dev/sdb /dev/sdc
```

```
vagrant@node1:~$ sudo zpool create -f mypool raidz1 /dev/sdb /dev/sdc
vagrant@node1:~$ sudo zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
mypool    992G   182K   992G    0%  1.00x  ONLINE  -
vagrant@node1:~$ sudo zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME            STATE        READ  WRITE  CKSUM
        mypool           ONLINE         0     0     0
          raidz1-0       ONLINE         0     0     0
            sdb           ONLINE         0     0     0
            sdc           ONLINE         0     0     0

errors: No known data errors
```

raidz2: 3 disks (double parity, RAID-6)

command

```
sudo zpool create mypool raidz2 /dev/sdb /dev/sdc /dev/sdd
```

```
vagrant@node1:~$ sudo zpool create -f mypool raidz2 /dev/sdb /dev/sdc /dev/sdd
vagrant@node1:~$ sudo zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
mypool    1.46T  278K   1.46T    0%  1.00x  ONLINE  -
vagrant@node1:~$ sudo zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME      STATE    READ WRITE CKSUM
        mypool     ONLINE      0     0     0
          raidz2-0  ONLINE      0     0     0
            sdb     ONLINE      0     0     0
            sdc     ONLINE      0     0     0
            sdd     ONLINE      0     0     0

errors: No known data errors
```

raidz3: 4 disks (triple parity)

command

```
sudo zpool create mypool raidz3 /dev/sdb /dev/sdc /dev/sdd /dev/sde
```

```
vagrant@node1:~$ sudo zpool create -f mypool raidz3 /dev/sdb /dev/sdc /dev/sdd /dev/sde
vagrant@node1:~$ sudo zpool list
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT
mypool    1.94T  364K   1.94T    0%  1.00x  ONLINE  -
vagrant@node1:~$ sudo zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME      STATE    READ WRITE CKSUM
        mypool     ONLINE      0     0     0
          raidz3-0  ONLINE      0     0     0
            sdb     ONLINE      0     0     0
            sdc     ONLINE      0     0     0
            sdd     ONLINE      0     0     0
            sde     ONLINE      0     0     0

errors: No known data errors
```

Turn on the auto-replace feature to mypool

```
sudo zpool autoreplace=on mypool
```

Create multiple ZFS filesystems on pool

command

```
sudo zfs create mypool/fs1  
sudo zfs create mypool/fs2
```

Setup quota & Reservation

```
sudo zfs set quota=1024m mypool/fs1  
sudo zfs set reservation=200m mypool/fs1
```

```
vagrant@node1:~$ sudo zfs create mypool/fs1  
vagrant@node1:~$ sudo zfs create mypool/fs2  
vagrant@node1:~$ sudo zpool list  
NAME      SIZE  ALLOC   FREE   CAP  DEDUP  HEALTH  ALTROOT  
mypool    1.94T  1.32M  1.94T   0%  1.00x  ONLINE  -  
vagrant@node1:~$ sudo zfs list  
NAME                USED  AVAIL  REFER  MOUNTPOINT  
mypool              188K   488G   32K    /mypool  
mypool/fs1          30K   488G   30K    /mypool/fs1  
mypool/fs2          30K   488G   30K    /mypool/fs2
```

Set the mount point

default

fs1 => /mypool/fs1

fs2 => /mypool/fs2

set mount point

```
sudo zfs set mountpoint=/test mypool/fs1
```

Set the compression

```
sudo zfs set compress=on mypool/fs1
```

check

```
sudo dd if=/dev/zero of=test.txt count=10 bs=10M
```

```
sudo dd if=/dev/zero of=test2.txt count=10 bs=10M
```

```
vagrant@node1:/mypool/fs1/tt$ ls -la
total 102420
drwxr-xr-x 2 root root      4 Feb 26 05:56 .
drwxr-xr-x 3 root root      4 Feb 26 05:52 ..
-rw-r--r-- 1 root root 104857600 Feb 26 05:52 test.txt
-rw-r--r-- 1 root root 104857600 Feb 26 05:57 test2.txt
vagrant@node1:/mypool/fs1/tt$ du -h
101M  .
```

Create a snapshot

snapshot a file-system

snapshot fs1 to [mysnapshot2](#).

```
sudo zfs snapshot mypool/fs1@mysnapshot2
```

```
vagrant@node1:/mypool/fs1$ sudo zfs snapshot mypool/fs1@mysnapshot2
vagrant@node1:/mypool/fs1$ sudo zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
mypool/fs1@mysnapshot1              0      -   1.07G  -
mypool/fs1@mysnapshot2              0      -   1.07G  -
```

snapshot all file-systems in the pool

```
sudo zfs snapshot mypool@mysnapshot_all
```

```
vagrant@node1:/mypool/fs1$ sudo zfs snapshot -r mypool@mysnapshot_all
vagrant@node1:/mypool/fs1$ sudo zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
mypool@mysnapshot_all              0      -    32K  -
mypool/fs1@mysnapshot1              0      -   1.07G  -
mypool/fs1@mysnapshot2              0      -   1.07G  -
mypool/fs1@mysnapshot_all           0      -   1.07G  -
mypool/fs2@mysnapshot_all           0      -    30K  -
```

destroy a snapshot

```
sudo zfs destroy mypool/fs1@mysnapshot1
```

rename a snapshot

```
sudo zfs rename mypool/fs1@mysnapshot1 mypool/fs1@mysnapshot_new1
```

disk space accounting for zfs snapshot

```
sudo zfs list -t snapshot -o space
```

Rolling back a zfs snapshot

Properties

- To rollback to an earlier snapshot, all intermediate snapshots must be destroyed.

Command

```
sudo zfs list -t snapshot // check all snapshot  
sudo zfs rollback mypool/fs1@mysnapshot1 // rollback an earlier snapshot
```

```
vagrant@node1:/mypool/fs1$ sudo zfs list -t snapshot  
NAME                                USED  AVAIL  REFER  MOUNTPOINT  
mypool@mysnapshot_all              0      -    32K    -  
mypool/fs1@mysnapshot1             0      -   1.07G  -  
mypool/fs1@mysnapshot2             0      -   1.07G  -  
mypool/fs1@mysnapshot_all          0      -   1.07G  -  
mypool/fs2@mysnapshot_all          0      -    30K    -  
vagrant@node1:/mypool/fs1$ sudo zfs rollback mypool/fs1@mysnapshot1  
cannot rollback to 'mypool/fs1@mysnapshot1': more recent snapshots exist  
use '-r' to force deletion of the following snapshots:  
mypool/fs1@mysnapshot2  
mypool/fs1@mysnapshot_all  
vagrant@node1:/mypool/fs1$ sudo zfs rollback -r mypool/fs1@mysnapshot1  
vagrant@node1:/mypool/fs1$ ls  
test.txt  tt
```

Create zfs clone

- Like snapshot, the clone function initially consumes no additional disk space.
- To clone a file system is instantaneous

Command

```
sudo zfs snapshot mypool/fs1@new_modify # snapshot current file system  
sudo zfs clone mypool/fs1@new_modify mypool/fs1-clone # clone it to fs1-clone
```

```
vagrant@node1:/mypool/fs1$ sudo zfs clone mypool/fs1@new_modify mypool/fs1-clone2  
vagrant@node1:/mypool/fs1$ sudo zfs list  
NAME                                USED  AVAIL  REFER  MOUNTPOINT  
mypool                             1.07G  487G    33K    /mypool  
mypool/fs1                         1.07G  487G   1.07G    /mypool/fs1  
mypool/fs1-clone                    1K    487G   1.07G    /mypool/fs1-clone  
mypool/fs1-clone2                   1K    487G   1.07G    /mypool/fs1-clone2  
mypool/fs2                          30K    487G    30K    /mypool/fs2
```

Replacing current filesystem with a zfs clone

- Without clone promotion, you cannot destroy an original file system of active clones.

- You can use the zfs promote command to replace an active ZFS file system with a clone of that file system. This feature enables you to clone and replace file systems so that the original file system becomes the clone of the specified file system. ([ref](#))

```
sudo zfs snapshot mypool/fs1@1-ok      # snapshot current file system to mypool/1-ok
sudo zfs list                          # list all file system

# switch the active filesystem to the mypool/fs1@1-ok system
sudo zfs promote mypool/fs1@1-ok
```

Share NFS/SMB ([ref: NFS setup](#))

Command

```
sudo zfs create -o mountpoint=/ds -o sharenfs=on mypool/ds
```

Next

- <http://www.slideshare.net/relling/zfs-tutorial-lisa-2011>
-

References

- <http://arstechnica.com/information-technology/2014/02/ars-walkthrough-using-the-zfs-next-gen-file-system-on-linux/>
- zfs tutorial, <http://thegeekdiary.com/zfs-tutorials-creating-zfs-pools-and-file-systems/>
- zfs administration guide, http://docs.oracle.com/cd/E23823_01/html/819-5461/gaypw.html
- zfs snapshot, <http://docs.oracle.com/cd/E19253-01/819-5461/6n7ht6r4f/index.html>
- http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide#ZFS_Administration_Considerations
- zfs tutorial, <http://www.slideshare.net/relling/zfs-tutorial-lisa-2011>
-