Table des matières

ZFS snapshots 3
Infos disques + perfs 3
Monter un snap ZFS 3
gpart 3
zpool dégradé 3
Install avec zpool mirroré (RAID 1) 3
FreeBSD ports 5
FreeBSD packages 5
Empêcher routes dynamiques 5
Monter image ISO 5
Démonter image ISO 5
Augmenter la taille d'un zpool mirroré 5
Exclure package - upgrade 6

ZFS snapshots

- Voir espace utilisé :

```
[root@nas /ZP_dataM1/ovh]# zfs list -o space
```

- Cleaner des snapshots

```
[root@nas /ZP_dataM1/ovh]# zfs list -t snap  |grep ovh |tail -4
ZP_dataM2/ovh@zfs-auto-snap_weekly-2015-11-28-10h30               16.2M      -   129G  -
ZP_dataM2/ovh@zfs-auto-snap_weekly-2015-11-29-10h30                644M      -   128G  -
ZP_dataM2/ovh@zfs-auto-snap_daily-2015-11-30-08h30                    0      -   128G  -
ZP_dataM2/ovh@zfs-auto-snap_weekly-2015-11-30-10h30                   0      -   128G  -
```

```
[root@nas /ZP_dataM1/ovh]# zfs destroy ZP_dataM2/ovh@zfs-auto-snap_weekly-2015-11-30-10h30
```

Infos disques + perfs

```
[root@freebsdVM ~]# diskinfo -ctv da2
da2
        512             # sectorsize
        8589934592      # mediasize in bytes (8.0G)
        16777216        # mediasize in sectors
        0               # stripesize
        0               # stripeoffset
        1044            # Cylinders according to firmware.
        255             # Heads according to firmware.
        63              # Sectors according to firmware.
                        # Disk ident.

I/O command overhead:
        time to read 10MB block      0.089807 sec       =    0.004 msec/sector
        time to read 20480 sectors   6.371639 sec       =    0.311 msec/sector
        calculated command overhead                     =    0.307 msec/sector

Seek times:
        Full stroke:      250 iter in   2.603755 sec =   10.415 msec
        Half stroke:      250 iter in   5.275366 sec =   21.101 msec
        Quarter stroke:   500 iter in   7.446248 sec =   14.892 msec
        Short forward:    400 iter in   3.744817 sec =    9.362 msec
        Short backward:   400 iter in   3.695824 sec =    9.240 msec
        Seq outer:       2048 iter in   0.680164 sec =    0.332 msec
        Seq inner:       2048 iter in   0.904928 sec =    0.442 msec
Transfer rates:
        outside:       102400 kbytes in   0.844724 sec =   121223 kbytes/sec
        middle:        102400 kbytes in   0.892766 sec =   114700 kbytes/sec
        inside:        102400 kbytes in   1.150101 sec =    89036 kbytes/sec
```

Monter un snap ZFS

```
zfs list -t snap -r ZP_dataM1/mp3
mount -t zfs ZP_dataM1/mp3@zfs-auto-snap_daily-2015-11-21-08h30 /mnt
```

gpart

```
gpart show -l da0
gpart show da0
```

zpool dégradé

```
camcontrol rescan all
zpool online system /dev/gpt/system1
```

Install avec zpool mirroré (RAID 1)

⇒ choisir “Shell” lors du step de partitionnement

- Lister les disques

```
camcontrol devlist
```

- Création de la table de partition, sur chaque disque :

```
# gpart create -s gpt da0
# gpart add -b 34 -s 512k -t freebsd-boot -l boot0 da0
# gpart add -s 2G -t freebsd-swap -l swap0 da0
# gpart add -s 10G -t freebsd-zfs -l system0 da0
```

```
# gpart create -s gpt da1
# gpart add -b 34 -s 512k -t freebsd-boot -l boot1 ada1
# gpart add -s 2G -t freebsd-swap -l swap1 ada1
# gpart add -s 10G -t freebsd-zfs -l system1 ada1
```

- Install du bootcode

```
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada0
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1
```

- Création du zpool

```
# zpool create -m none -o altroot=/mnt -o cachefile=/var/tmp/zpool.cache \
  system mirror /dev/gpt/system0 /dev/gpt/system1
# zfs set mountpoint=/ system
```

- Création des FS

```
# zfs create -o compression=on -o setuid=off system/tmp
# chmod 1777 /mnt/tmp

# zfs create system/usr
# zfs create system/usr/home
# cd /mnt
# ln -s usr/home home
# zfs create system/usr/local
# zfs create -o compression=on -o setuid=off system/usr/ports
# zfs create -o exec=off -o setuid=off system/usr/ports/distfiles
# zfs create -o exec=off -o setuid=off system/usr/ports/packages
# zfs create system/usr/obj
# zfs create -o compression=on -o exec=off -o setuid=off system/usr/src

# zfs create system/var
# zfs create -o exec=off -o setuid=off system/var/backups
# zfs create -o compression=on -o exec=off -o setuid=off system/var/crash
# zfs create -o exec=off -o setuid=off system/var/db
# zfs create -o exec=on -o compression=on -o setuid=off system/var/db/pkg
# zfs create -o exec=off -o setuid=off system/var/empty
# zfs create -o compression=on -o exec=off -o setuid=off system/var/log
# zfs create -o compression=on -o exec=off -o setuid=off system/var/mail
# zfs create -o exec=off -o setuid=off system/var/run
# zfs create -o compression=on -o setuid=off system/var/tmp
# chmod 1777 /mnt/var/tmp

# zpool set bootfs=system system
# mkdir -p /mnt/boot/zfs
# cp /var/tmp/zpool.cache /mnt/boot/zfs/zpool.cache
```

⇒ Continuer l'install, ouvrir un shell à la fin de l'install

```
# echo 'zfs_load="YES"' >> /boot/loader.conf
# echo 'vfs.root.mountfrom="zfs:system"' >> /boot/loader.conf
# echo 'zfs_enable="YES"' >> /etc/rc.conf
# cd /media
# mkdir cdrom flash
```

- /etc/fstab

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/gpt/swap0          none            swap    sw              0       0
/dev/gpt/swap1          none            swap    sw              0       0
/dev/cd0                /media/cdrom    cd9660  ro,noauto       0       0
```

- Après premier boot

```
# zfs set readonly=on system/var/empty
# rm /etc/motd
```

FreeBSD ports

- Mise à jour

```
# portsnap fetch extract
# portsnap fetch update
```

- Install

```
cd /usr/ports/net/samba41
make install clean
```

FreeBSD packages

```
pkg update
pkg search xxxxxxx
pkg install xxxxxxx
pkg info -D -x subsonic-jetty-5.2.1
```

Empêcher routes dynamiques

- **/etc/sysctl.conf**

```
net.inet.ip.redirect=0
net.inet.icmp.drop_redirect=1
net.inet.icmp.log_redirect=0
```

Monter image ISO

```
mdconfig -a -t vnode -f /path/to/image.iso -u 1
mount -t cd9660 /dev/md1 /mnt/cdrom
```

Démonter image ISO

```
mount -u /mnt/cdrom
mdconfig -d -u 1
```

Augmenter la taille d'un zpool mirroré

On a un zpool dataZP de 4 Go et on veut le faire passer à 8 Go. On retire un premier disque de 4 Go qu'on remplace par un disque de 8 Go. Puis on remplace le disque restant de 4 Go par un nouveau disque de 8 Go.

- Soit la config ci-dessous :

```
[root@freebsdVM ~]# egrep 'da[0-9]' /var/run/dmesg.boot|grep MB|grep -v trans
da0: 8192MB (16777216 512 byte sectors: 255H 63S/T 1044C)
da1: 8192MB (16777216 512 byte sectors: 255H 63S/T 1044C)
da2: 4096MB (8388608 512 byte sectors: 255H 63S/T 522C)
da3: 4096MB (8388608 512 byte sectors: 255H 63S/T 522C)
```

```
[root@freebsdVM ~]# zpool status dataZP
  pool: dataZP
 state: ONLINE
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri Nov  6 11:33:06 2015
config:

        NAME        STATE     READ WRITE CKSUM
        dataZP      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            da3     ONLINE       0     0     0
            da2     ONLINE       0     0     0

errors: No known data errors
```

```
[root@freebsdVM ~]# zpool list
NAME     SIZE  ALLOC   FREE   FRAG  EXPANDSZ    CAP  DEDUP  HEALTH  ALTROOT
dataZP  3.98G   164K  3.98G     0%         -     0%  1.00x  ONLINE  -
zroot   5.97G  1.06G  4.91G    10%         -    17%  1.00x  ONLINE  -
```

- On retire un disque de 4 Go :

```
[root@freebsdVM ~]# zpool status dataZP
  pool: dataZP
 state: DEGRADED
status: One or more devices has been removed by the administrator.
        Sufficient replicas exist for the pool to continue functioning in a
        degraded state.
action: Online the device using 'zpool online' or replace the device with
        'zpool replace'.
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri Nov  6 11:33:06 2015
config:

        NAME                      STATE     READ WRITE CKSUM
        dataZP                    DEGRADED     0     0     0
          mirror-0                DEGRADED     0     0     0
            15131538193711764791  REMOVED      0     0     0  was /dev/da3
            da2                   ONLINE       0     0     0

errors: No known data errors
```

- On ajoute un disque de 8 Go :

```
[root@freebsdVM ~]# diskinfo -v da3 |grep bytes
        8589934592      # mediasize in bytes (8.0G)
```

- On l'ajoute au pool :

```
zpool online dataZP da3
```

- On retire le disque de 4 Go restant et on ajoute un disque de 8 Go :

```
[root@freebsdVM ~]# zpool online dataZP da2
```

- On étend le ZP :

⇒ cf. EXPANDSZ = 4G

```
[root@freebsdVM ~]# zpool list
NAME     SIZE  ALLOC   FREE   FRAG  EXPANDSZ    CAP  DEDUP  HEALTH  ALTROOT
dataZP  3.98G   172K  3.98G     0%        4G     0%  1.00x  ONLINE  -
zroot   5.97G  1.06G  4.91G    10%         -    17%  1.00x  ONLINE  -
```

```
zpool online -e dataZP da2
zpool online -e dataZP da3
```

```
NAME     SIZE  ALLOC   FREE   FRAG  EXPANDSZ    CAP  DEDUP  HEALTH  ALTROOT
dataZP  7.98G   310K  7.98G     0%         -     0%  1.00x  ONLINE    -
zroot   5.97G  1.06G  4.91G    10%         -    17%  1.00x  ONLINE  -
```

Exclure package - upgrade

```
[root@nas /var/log]# pkg lock nut-2.7.3_3
nut-2.7.3_3: lock this package? [y/N]: y
Locking nut-2.7.3_3
```

From:
https://unix-bck.ndlp.info/ - **Where there is a shell, there is a way**

Permanent link:
https://unix-bck.ndlp.info/doku.php/informatique:bsd?rev=1486031848

Last update: **2017/02/02 10:37**