Table des matières

1 ZFS

1.1 Introduction

Système de fichier 128 bits

Checksum 64 bits

Plusieurs systèmes de fichiers dans 1 pool ZFS.

Copy On Write (COW)

ZFS détermine lui-même par ses checksum s'il y a un pb de cohérence sur un miroir et le corrige automatiquement.

Pool : ensemble de disques
VDEV : 1 groupe miroir ZFS
Clone : copie accessible en RW
Réargenture : resynchro
Jeux de données (dataset) :

Taille mini d'un disque : 128MB

Format EFI : table de partitionnement va de 0 à 8, la 8 est réservée (8M), le 2 n'est plus réservée.

Le label par défaut était SMI.

Pour passer d'un format de disque à un autre :

Format -e
--> label
0 SMI
1 EFI

1.2 ZPOOL

1.2.1 Création de pool (les différentes possibilités)

 # zpool create pool1 c1t0d0 c1t1d0
 # zpool create pool1 mirror c1t0d0 c1t1d0 mirror c3t0d0 c3t1d0
 # zpool create pool1 raidz c1t0d0 c1t1d0 c3t0d0 c3t1d0 c4t0d0
 # zpool create pool1 raidz2 c1t0d0 c2t1d0 c3t1d0

!! le répertoire /pool1 ne doit pas exister ou doit être vide.

Option -n pour simuler la création

Stauts du pool : # zpool status -v pool1

Point de montage : # zpool create -m /export/home users c2t1d0

Suppression d'un pool : #zpool destroy pool1

zpool add : on ajoute un élément (une concaténation ou un miroir)
zpool attach : on ajoute un élément en concaténation à un élément donné.

Pour agrandir un miroir on fait un attach sur chacun des VDEV.

zpool detach p1 c1t1d0 : supprime le device de la configuration
offline p1 c1t1d0 : on garde le device dans la config mais il n'y a plus d'I/O dessus
online p1 c1t1d0 : on redonne l'accès en RW
clear p1 : nettoyage des messages d'erreur

zpool iostat [pool] [frequence]

1.2.2 Disque de spare

Association d'un disque de hot spare à un miroir :
zpool create pool1 mirror c1t0d0 c1t1d0 spare c3t1d0

zpool status -x
=> on voit dans la config qu'il y a un disque de spare associé au miroir.

Remettre un disque de spare dans une fonction spare (après son utilisation suite à erreur disque):
zpool detach pool1 c3t1d0

1.2.3 Modification d'un paramètre

zpool get all pool1 : donne des infos sur le pool : taille, état, guid, autoreplace (quand on change le disque), failmode (wait, panic, continue)

wait : plus d'I/O sur le pool
continue : on accède aux données en RO
panic : systeme part en panic!!

zpool [get|set] autoreplace[=on/off] pool1

Historique des commandes :
zpool history [-l] [-i] [pool] : donne l'historique des commandes passées

1.2.4 Migration d'un pool

Machine1: # **zpool export [-f] pool1**

Machine2: # **zpool import** => liste tous les pools importables
zpool import pool1 [pool12] => importe effectivement le pool, et le renomme si on indique un autre nom.

zpool destroy pool1
zpool import -D pool1 => liste les pools qui peuvent récupérés

zpool import -Df pool1 => pour forcer la récupération
zpool online pool1 cxtxdx

1.2.5 Manipulations

*** Création d'un pool concaténé**

global# **zpool create -f p1 c1d0s4**
global# **zpool add p1 c1d0s5**
global# **zpool list**

```
NAME    SIZE    USED  AVAIL    CAP  HEALTH  ALTROOT
p1       30G  79,5K  30,0G     0%  ONLINE  -
```

global# zpool status

```
 pool : p1
état : ONLINE
purger : aucun requis
configuration :

        NAME         STATE     READ WRITE CKSUM
        p1           ONLINE       0     0     0
          c1d0s4     ONLINE       0     0     0
          c1d0s5     ONLINE       0     0     0
```

global# df -h
Système de fichiers taille utilisé dispo capacité Monté sur
p1   30G   21K   30G   1%   /p1

*** Création d'un pool miroir**

global# **zpool create -m /data p1 mirror c1d0s4 c1d0s5**
global# zpool status p1

```
  pool : p1
état : ONLINE
purger : aucun requis
configuration :

        NAME         STATE     READ WRITE CKSUM
        p1           ONLINE       0     0     0
          mirror     ONLINE       0     0     0
            c1d0s4   ONLINE       0     0     0
            c1d0s5   ONLINE       0     0     0
```

erreurs : aucune erreur de données connue
global# df -h
Système de fichiers taille utilisé dispo capacité Monté sur
p1   15G   21K   15G   1%   /data

*** Créations d'un pool avec 2 miroirs concaténés**

global# **zpool create p1 mirror c1d0s4 c1d0s5**
global# **zpool add p1 mirror c1d0s3 c1d0s6**

global# **zpool list**

```
NAME    SIZE   USED  AVAIL    CAP  HEALTH  ALTROOT
p1       30G  79,5K  30,0G     0%  ONLINE  -
```

global# **zpool status p1**

```
 pool : p1
 état : ONLINE
 purger : aucun requis
configuration :

        NAME        STATE     READ WRITE CKSUM
        p1          ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            c1d0s4  ONLINE       0     0     0
            c1d0s5  ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            c1d0s3  ONLINE       0     0     0
            c1d0s6  ONLINE       0     0     0
```

*** Mettre un device offline**

global# zpool offline p1 c1d0s6
global# zpool status

```
  pool : p1
 état : DEGRADED
statut : Un ou plusieurs périphériques ont été placés hors ligne par
l'administrateur. Un nombre suffisant de répliques existe pour que le pool
continue à fonctionner dans un état endommagé.
action : mettez en ligne le périphérique en utilisant 'zpool online' ou
remplacez-le avec 'zpool replace'.
 purger : aucun requis
configuration :

        NAME        STATE     READ WRITE CKSUM
        p1          DEGRADED     0     0     0
          mirror    ONLINE       0     0     0
            c1d0s4  ONLINE       0     0     0
            c1d0s5  ONLINE       0     0     0
          mirror    DEGRADED     0     0     0
            c1d0s3  ONLINE       0     0     0
            c1d0s6  OFFLINE      0     0     0
```

*** Remettre le device online**

global# **zpool online p1 c1d0s6**
global# **zpool status p1**

```
 pool : p1
 état : ONLINE
 purger : resilver completed après 0h0m avec 0 erreurs sur Wed May 19 15:08:47
2010
configuration :

        NAME        STATE     READ WRITE CKSUM
        p1          ONLINE       0     0     0
          mirror    ONLINE       0     0     0
```

```
            c1d0s4  ONLINE       0     0     0
            c1d0s5  ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            c1d0s3  ONLINE       0     0     0
            c1d0s6  ONLINE       0     0     0  10,5K resilvered
```

*** Création d'un miroir manuel**

global# **zpool create p1 c1d0s3**
global# **zpool attach -f p1 c1d0s3 c1d0s4**

global# **zpool status**

```
  pool : p1
 état : ONLINE
 purger : resilver completed après 0h0m avec 0 erreurs sur Wed May 19 15:13:37
2010
configuration :

        NAME        STATE     READ WRITE CKSUM
        p1          ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            c1d0s3  ONLINE       0     0     0
            c1d0s4  ONLINE       0     0     0
```

global# **zpool history**

```
Historique de 'p1':
2010-05-19.15:12:48 zpool create -f p1 c1d0s3
2010-05-19.15:13:37 zpool attach -f p1 c1d0s3 c1d0s4
```

*** Récupération d'un pool détruit (à partir de l'ID)**

global# **zpool import -D**

```
  pool : p1
    id: 13683450957060884102
 état : DEGRADED (DETRUIT)
état : un ou plusieurs périphériques contiennent des données endommagées.
action : le pool peut être importé même si des périphériques sont endommagés
ou manquants.  Cependant, la tolérance de pannes du pool importé risque d'être
compromise. reportez-vous au site : http://www.sun.com/msg/ZFS-8000-4J
configuration :

        p1          DEGRADED
          mirror    DEGRADED
            c1d0s4  FAULTED  données endommagées
            c1d0s5  ONLINE
          mirror    DEGRADED
            c1d0s3  OFFLINE
            c1d0s6  ONLINE

  pool : p1
    id: 10535841933837252604
 état : ONLINE (DETRUIT)
action : vous pouvez importer le pool en utilisant son nom ou son
identificateur numérique.
```

```
configuration :

        p1          ONLINE
          mirror    ONLINE
            c1d0s3  ONLINE
            c1d0s4  ONLINE
```

global# **zpool import -D 13683450957060884102**

1.3 ZFS

1.3.1 Manipulation sur les filesystemes

Création d'un filesysteme
global# **zfs create p1/fs1**

global# zfs list
```
NAME      USED  AVAIL  REFER  MOUNTPOINT
p1        142K  29,5G    39K  /p1
p1/fs1     21K  29,5G    21K  /p1/fs1
```

global# zfs list p1
```
NAME   USED  AVAIL  REFER  MOUNTPOINT
p1     142K  29,5G    39K  /p1
```

global# zfs list -r p1
```
NAME      USED  AVAIL  REFER  MOUNTPOINT
p1        142K  29,5G    39K  /p1
p1/fs1     21K  29,5G    21K  /p1/fs1
```

Renommage d'un filesysteme
global# **zfs rename p1/fs1/fs2 p1/fs1/fs4**

global# **zfs get all p1**
```
NAME  PROPERTY              VALUE                  SOURCE
p1    type                  filesystem             -
p1    creation              mer. mai  19 15:05 2010  -
p1    used                  202K                   -
p1    available             29,5G                  -
p1    referenced            40K                    -
p1    compressratio         1.00x                  -
p1    mounted               yes                    -
p1    quota                 none                   default
p1    reservation           none                   default
p1    recordsize            128K                   default
p1    mountpoint            /p1                    default
p1    sharenfs              off                    default
p1    checksum              on                     default
p1    compression           off                    default
p1    atime                 on                     default
p1    devices               on                     default
```

```
p1    exec                  on                        default
p1    setuid                on                        default
p1    readonly              off                       default
p1    zoned                 off                       default
p1    snapdir               hidden                    default
p1    aclmode               groupmask                 default
p1    aclinherit            restricted                default
p1    canmount              on                        default
p1    shareiscsi            off                       default
p1    xattr                 on                        default
p1    copies                1                         default
p1    version               4                         -
p1    utf8only              off                       -
p1    normalization         none                      -
p1    casesensitivity       sensitive                 -
p1    vscan                 off                       default
p1    nbmand                off                       default
p1    sharesmb              off                       default
p1    refquota              none                      default
p1    refreservation        none                      default
p1    primarycache          all                       default
p1    secondarycache        all                       default
p1    usedbysnapshots       0                         -
p1    usedbydataset         40K                       -
p1    usedbychildren        162K                      -
p1    usedbyrefreservation  0                         -
```

Quand on a un – dans source cela signifie que le paramètre est non modifiable.

Modification du paramètre canmount

global# **zfs set canmount=off p1/fs1**
global# **zfs get canmount p1/fs1**

```
NAME    PROPERTY  VALUE     SOURCE
p1/fs1  canmount  off       local
```

p1/fs1 n'apparaît plus avec df -h

On peut monter/démonter un filesysteme avec zfs :
global# **zfs mount p1/fs1/fs2/fs3**

Changement du point de montage :
global# **zfs set mountpoint=/export/users p1/fs1**

```
p1/fs1/fs2              30G    23K    30G     1%    /export/users/fs2
```

global# **zfs set mountpoint=/export/users/titi p1/fs1/fs2**
global# df -h /export/users/titi

```
Système de fichiers  taille utilisé  dispo capacité  Monté sur
p1/fs1/fs2              30G    23K    30G     1%    /export/users/titi
```

Destruction récursive des filesystèmes :
global# **zfs destroy -r p1**

1.3.2 Partages NFS

global# **zfs create p1/fs1**
global# **zfs create p1/fs2**

global# **zfs set sharenfs=on p1**
global# **zfs get -r sharenfs p1**

```
NAME     PROPERTY  VALUE     SOURCE
p1       sharenfs  on        local
p1/fs1   sharenfs  on        inherited from p1
p1/fs2   sharenfs  on        inherited from p1
```

global# **dfshares**

```
RESOURCE                                  SERVER ACCESS    TRANSPORT
stagiaire6:/p1                          stagiaire6  -         -
stagiaire6:/p1/fs1                      stagiaire6  -         -
stagiaire6:/p1/fs2                      stagiaire6  -         -
```

bash-3.00# **share**

```
-         /p1   rw   ""
-         /p1/fs1   rw   ""
-         /p1/fs2   rw   ""
```

Monter en NFS /p1/fs1 :
bash-3.00# mount stagiaire6:/p1/fs1 /mnt

Supprimer le partage NFS :
bash-3.00# zfs set sharenfs=off p1

1.3.3 Quotas

Attribuer un quota à p1/fs1

bash-3.00# zfs set quota=200m p1/fs1

bash-3.00# zfs list -r p1

```
NAME           USED  AVAIL  REFER  MOUNTPOINT
p1            10,1M  29,5G    24K  /p1
p1/fs1        9,93M   190M  9,91M  /p1/fs1          => auparavant  29,5G
p1/fs1/sfs1     21K   190M    21K  /p1/fs1/sfs1
p1/fs2          21K  29,5G    21K  /p1/fs2
```

Annuler un quota
bash-3.00# zfs set quota=none p1/fs1

On peut créer des quotas sur des utilisateurs ou des groupes.

1.3.4 Réservation

On garantie un espace pour le filesysteme.

bash-3.00# zfs set reservation=100m p1/fs1
bash-3.00# zfs list -r p1

```
NAME           USED  AVAIL  REFER  MOUNTPOINT
p1             100M  29,4G    24K  /p1
p1/fs1        45,5K  29,5G  24,5K  /p1/fs1
p1/fs1/sfs1     21K  29,5G    21K  /p1/fs1/sfs1
p1/fs2          21K  29,4G    21K  /p1/fs2
```

=> auparavant 203K

1.3.5 Raw volume ZFS

bash-3.00# zfs create -V 1g p1/bdd
bash-3.00# zfs list -r

```
NAME           USED  AVAIL  REFER  MOUNTPOINT
p1            1,10G  28,4G    24K  /p1
p1/bdd           1G  29,4G    16K  -
p1/fs1        45,5K  28,5G  24,5K  /p1/fs1
p1/fs1/sfs1     21K  28,5G    21K  /p1/fs1/sfs1
p1/fs2          21K  28,4G    21K  /p1/fs2
```

bash-3.00# swap -l

```
swapfile             dev  swaplo blocs   libres
/dev/dsk/c1d0s1     102,1       8 8401984 8401984
```

bash-3.00# swap -a /dev/zvol/dsk/p1/bdd

bash-3.00# swap -l

```
swapfile             dev  swaplo blocs   libres
/dev/dsk/c1d0s1     102,1       8 8401984 8401984
/dev/zvol/dsk/p1/bdd 181,1       8 2097144 2097144
```

1.3.6 Snapshot

- Copie temporaire en lecture seule d'un système de fichier ou d'un volume à des fins de sauvegarde
- Création quasi immédiate
- Consomment de l'espace dans le pool de stockage auquel appartient le FS à partir duquel ils ont été créés
- Attention : le snapshot ne diminue pas en taille => possibilité saturation du pool de stockage

Le snapshot stocke les informations d'origine, avant modification.

- Création
  # zfs snapshot pool1/home/user1@friday
  # zfs snapshot -r pool1/home@now

- Suppression
  # zfs destroy pool1/home/user1@friday
  # zfs destroy -r pool1/home@now

- Renommage
  # zfs rename pool1/home/user1@080324 pool1/home/user1@friday
  # zfs rename pool1/home/user1@yesterday @2daysago

!! Les snapshots doivent se trouver dans le même dataset.

- Affichage
  # zfs list -r pool1 => n'affiche pas les snapshots
  # zfs list -t snapshot
  # zfs list -r snapshot -o name,creation pool1/home

- Accès
  # /pool1/home/user1/.zfs/snapshot

- Restauration

On revient à l'était du snapshot. Suppression des snapshots intermédiaires.

=> Si c'est le snapshot le plus récent :
  # zfs rollback pool1/home/user1@tuesday

=> Si ce n'est plus le snapshot le plus récent : message pour indiquer que de plus récents existent.
Il faut utiliser l'option -r.
  # zfs rollback -r pool1/home/user1@tuesday

!! Dans ce cas les snapshots intermédiaires seront supprimés.

ATELIER

```
bash-3.00# zfs create p1/fs1
bash-3.00# zfs create p1/fs2
bash-3.00# zfs create p1/fs1/fs3
bash-3.00# zfs list
NAME         USED  AVAIL  REFER  MOUNTPOINT
p1           159K  14,8G    24K  /p1
p1/fs1        42K  14,8G    21K  /p1/fs1
p1/fs1/fs3    21K  14,8G    21K  /p1/fs1/fs3
p1/fs2        21K  14,8G    21K  /p1/fs2

bash-3.00# mkfile 10m /p1/fs1/fic1

bash-3.00# zfs list -r p1/fs1
NAME         USED  AVAIL  REFER  MOUNTPOINT
p1/fs1      10,0M  14,8G  10,0M  /p1/fs1
p1/fs1/fs3    21K  14,8G    21K  /p1/fs1/fs3

bash-3.00# zfs snapshot p1/fs1@lundi

bash-3.00# zfs list -r p1/fs1
NAME           USED  AVAIL  REFER  MOUNTPOINT
p1/fs1        10,0M  14,8G  10,0M  /p1/fs1
p1/fs1@lundi      0      -  10,0M  -
p1/fs1/fs3      21K  14,8G    21K  /p1/fs1/fs3


bash-3.00# mkfile 10m /p1/fs1/fic2
bash-3.00# zfs list -r p1/fs1
NAME           USED  AVAIL  REFER  MOUNTPOINT
p1/fs1        20,1M  14,7G  20,0M  /p1/fs1
p1/fs1@lundi    19K      -  10,0M  -
```

```
p1/fs1/fs3      21K  14,7G    21K  /p1/fs1/fs3

bash-3.00# zfs snapshot -r p1/fs1@mardi      => -r pour récursif sur sous-rep.
bash-3.00# zfs list -r p1/fs1
NAME               USED  AVAIL  REFER  MOUNTPOINT
p1/fs1            20,1M  14,7G  20,0M  /p1/fs1
p1/fs1@lundi        19K      -  10,0M  -
p1/fs1@mardi          0      -  20,0M  -
p1/fs1/fs3          21K  14,7G    21K  /p1/fs1/fs3
p1/fs1/fs3@mardi      0      -    21K  -

=> va créer des snapshots pour chaque filesystemes (à cause -r).

bash-3.00# zfs list -r -o space p1/fs1
NAME              AVAIL   USED  USEDSNAP  USEDDS  USEDREFRESERV  USEDCHILD
p1/fs1            14,7G  20,1M       19K   20,0M              0        21K
p1/fs1@lundi          -    19K         -       -              -          -
p1/fs1@mardi          -      0         -       -              -          -
p1/fs1/fs3        14,7G    21K         0     21K              0          0
p1/fs1/fs3@mardi      -      0         -       -              -          -

bash-3.00# zfs rollback -r p1/fs1@lundi

bash-3.00# zfs destroy -r p1/fs1@lundi

**
bash-3.00# mkfile 10m /p1/fs1/fic2
bash-3.00# zfs snapshot p1/fs1@lundi
bash-3.00# cd /p1/fs1/.zfs/snapshot/lundi
bash-3.00# ls -al
-rw------T   1 root     root     10485760 mai  20 11:16 fic1

=> on trouve sous .zfs les données sauvegardées dans le snapshot.

bash-3.00# mkfile 10m /p1/fs1/fic2
bash-3.00# zfs snapshot p1/fs1@mardi
bash-3.00# cd /p1/fs1/.zfs/snapshot/mardi
bash-3.00# ls -al
-rw------T   1 root     root     10485760 mai  20 11:16 fic1
-rw------T   1 root     root     10485760 mai  20 11:35 fic2
```

1.3.7 Clones

- Accessible en lecture/écriture
- Création quasi instantanée et ne consomme rien initialement
- Les clones se créent uniquement à partir d'un snapshot
- un clone n'hérite pas des propriétés du dataset à partir duquel il a été créé
- aucun espace disque supplémentaire
- un clone partage initialement son espace disque avec le snapshot d'origne

- Création
  # zfs snapshot projets/newprojet@today
  # zfs clone pprojets/newprojet@today projets/teamA/tempuser

zfs set sharenfs=on projets/teamA/tempuser
zfs set quota=5G projets/teamA/tempuser

- Suppression
  # zfs destroy projets/teamA/tempuser
  # zfs destroy -R projets/newprojet

- remplacement du FS par un clone
  # zfs create pool1/test/productA => FS d'origine
  # zfs snapshot pool1/test/productA@today => snapshot
  # zfs clone pool1/test/productA@today pool1/test/productAbeta => clone
  # zfs list -r pool1/test => le USED du clone est à 0
  # zfs promote pool1/test/productAbeta => le clone devient un FS (non cloné)
  # zfs list -r pool1/test => on voit que le FS Abeta est noté avec un USED
  # zfs rename pool1/test/productA pool1/test/productAorig
  # zfs rename pool1/test/productAbeta pool1/test/productA => le clone est maintenant nommé et monté comme le FS d'origine.
- Sauvegarde/restauration : send/receive

  # zfs send pool1/dana@snap1 > /sauve/snap.dmp
  # zfs receive pool1/mark@snp < /sauve/snap.dmp => va créer le FS pool1/mark + le snapshot
  # zfs send pool1/dana@snap1 | zfs receive spool/ds01 (ou ssh host2 zfs receive newpool1/dana)

Cas des incrémentales (le FS destination doit exister et on doit utiliser l'option -F) :

zfs send -i pool1/dana@snap2 pool1/dana@snap3 | ssh host2 zfs receive -F newpool1/dana

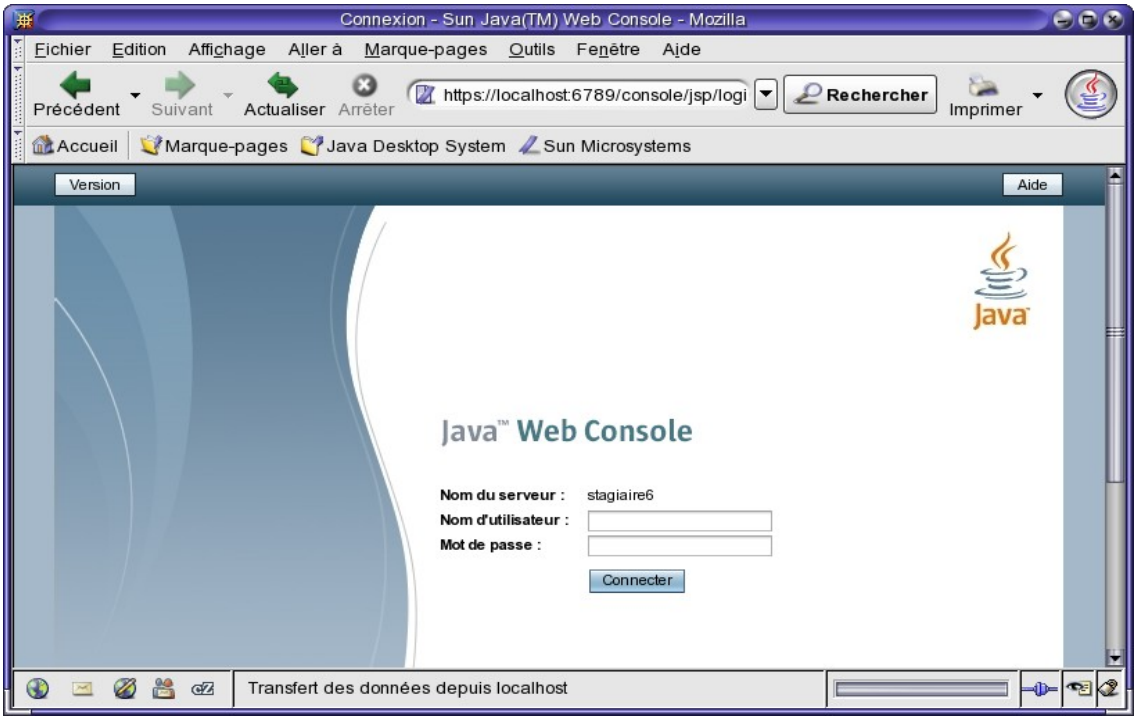
=> on envoie (send) le delta entre snap2 et snap3. Il faut avoir au préalable snap2 et snap3.

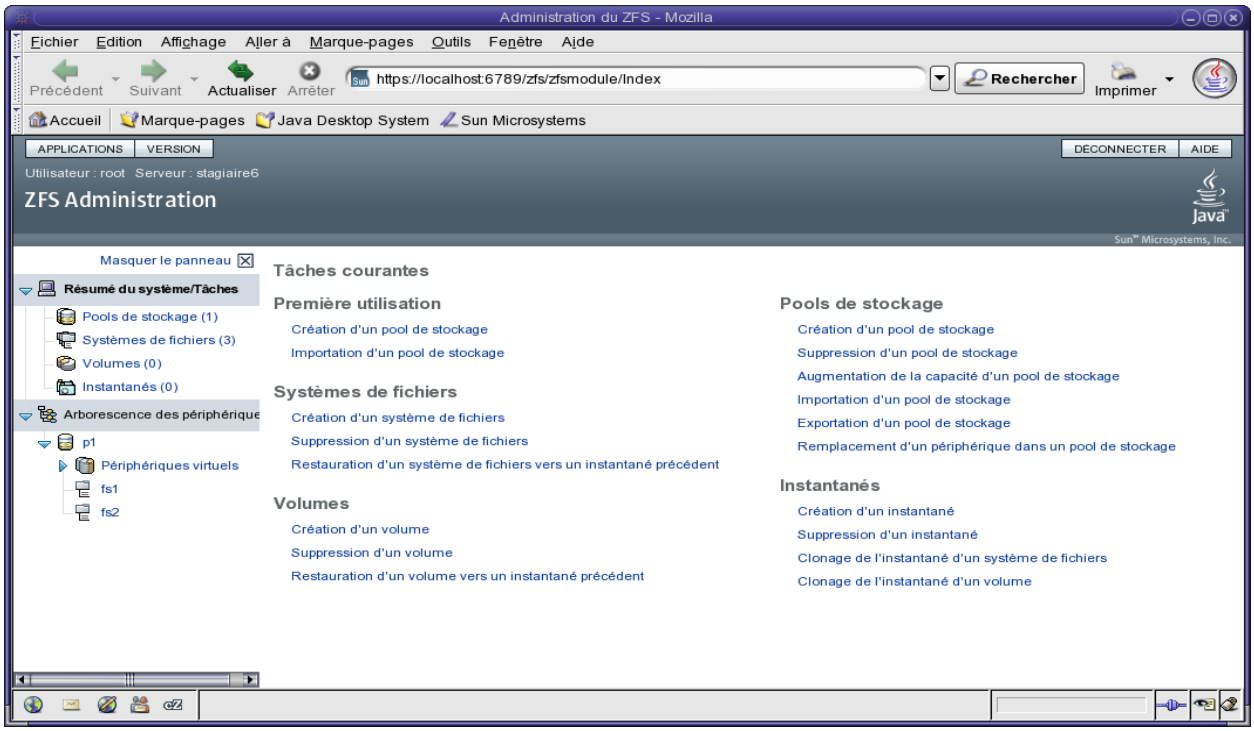
Remarque : option -R du send pour répliquer toute l'arborescence des snapshots (le FS racine + tous les sous points de montage).

users
users/user1
users/user2
users/user3

1.3.8 Interface graphique

Accessible à l'URL https://localhost:6789





1.4 ZFS et zones

zonecfg -z zion

zonecfg:zion> add fs

zonecfg:zion:fs> set type=zfs

zonecfg:zion:fs>set special=pool1/zone/zion

zonecfg:zion:fs> set dir=/export/shared
zonecfg:zion:fs> end

Délégation de dataset à une zone non globale => pour pouvoir faire des créations zfs depuis la zone non-globale

zonecfg:zion> add dataset
zonecfg:zion:device> set name=pool1/zone/zion
zonecfg:zion:device> end

zonecfg:zion> add device
zonecfg:zion:device> set match=/dev/zvol/dsk/pool1/vol
zonecfg:zion:device> end

Mot clé zoned : gestion des risques de sécurité, présence de binaires setuid, de liens, ... qui pourraient compromettre la sécurité de la zone globale.
ZFS utilise la propriété zoned pour indiquer qu'un dataset a été délégué à une zone non globale à in moment donné.

=> pour protéger la zone globale.

1.5 Les ACL

setfacl -m m:7,u:marc:5 /mnt/fic
getfacl

ls -V fic
ls -v fic

ID de l'index owner@ group@ everyone@ user group privileges heritage allow/deny

chmod A+acl-specification filename

A+ ajout
A- retrait
A= remplacement