ZFS: Snapshots and clones on zfs filesystems

Article Number: 185 | Rating: Unrated | Last Updated: Sun, Jun 3, 2018 9:13 AM

ZFS: Snapshots and clones on zfs filesystems

```
# Tested on RHEL 6 & 7

# A snapshot is an only-read photograph of a filesystem. When taking a snapshot, it is
# stored in a way so further transactions on filesystem will only be carried out on origin
# filesystem and not on snapshot itself. This way it will be possible to get back to
# previous status by doing a "rollback"

# A clone is equivalent to a read-write copy of the snapshot

# Clones and snapshots are not data copies but state ones, so they don't use any space when
# created. It is when origin filesystem is modified when differences are being stored, thus
# consuming disk space. If a rollback is done, these differences are overwritten and space
# is freed-up again.

# Note: Clones may be created only from existing snapshots. First we take the "photo" of
# the origin filesystem and then we create the clone.

# Snapshots are very useful, for instance to carry out tests without being afraid of losing
# important data.
```

```
# Given following zfs

zfs list
NAME           USED  AVAIL  REFER  MOUNTPOINT
c_pool        2.15M  3.84G    19K  /c_pool
c_pool/zfs01  2.02M  3.84G  2.02M  /zfs01   <---




# Create a snapshot from a ZFS
# ------------------------------------------------------------------------
-----------------------

zfs snapshot c_pool/zfs01@snapshot01


zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                  2.50M  3.84G    19K  /c_pool
c_pool/zfs01            2.02M  3.84G  2.02M  /zfs01
c_pool/zfs01@snapshot01     0      -  2.02M  -         <---




# Rollback a ZFS to a previous state
# ------------------------------------------------------------------------
-----------------------

# First we do some modifications on filesystem

cd /zfs01
```

```
dd if=/dev/urandom of=temp.file.01 bs=1M count=2
2+0 records in
2+0 records out
2097152 bytes (2.1 MB) copied, 24.7394 s, 84.8 kB/s


dd if=/dev/urandom of=temp.file.02 bs=1M count=2
2+0 records in
2+0 records out
2097152 bytes (2.1 MB) copied, 25.3346 s, 82.8 kB/s


ls -lrt
total 4101
-rw-r--r-- 1 root root 2097152 Feb  3 17:10 temp.file.01
-rw-r--r-- 1 root root 2097152 Feb  3 17:11 temp.file.02


zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                  4.18M  3.84G    19K  /c_pool
c_pool/zfs01            4.03M  3.84G  4.02M  /zfs01
c_pool/zfs01@snapshot01     9K      -    19K  -            <----
note the differences


# and then, rollback

zfs rollback c_pool/zfs01@snapshot01


zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                   176K  3.84G    19K  /c_pool
c_pool/zfs01              20K  3.84G    19K  /zfs01
c_pool/zfs01@snapshot01    1K      -    19K  -            <---
```

```
# files have disappeared:

ls -lrt
total 0



# Remove a snapshot
# --------------------------------------------------------------------
----------------------

zfs destroy c_pool/zfs01@snapshot01



zfs list -t all
NAME           USED  AVAIL  REFER  MOUNTPOINT
c_pool         174K  3.84G    19K  /c_pool
c_pool/zfs01    19K  3.84G    19K  /zfs01




# To have different points of restoration, several snapshots may be
taken at different times
# --------------------------------------------------------------------
----------------------

dd if=/dev/urandom of=temp.file.00 bs=1M count=2
2+0 records in
2+0 records out
2097152 bytes (2.1 MB) copied, 25.2053 s, 83.2 kB/s


ls -lrt
total 2051
-rw-r--r-- 1 root root 2097152 Feb  3 16:57 temp.file.00
```

```
zfs snapshot c_pool/zfs01@snapshot01


dd if=/dev/urandom of=temp.file.01 bs=1M count=2
    2+0 records in
2+0 records out
2097152 bytes (2.1 MB) copied, 25.9611 s, 80.8 kB/s


ls -lrt
total 4101
-rw-r--r-- 1 root root 2097152 Feb  3 16:57 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 16:58 temp.file.01


zfs snapshot c_pool/zfs01@snapshot02


dd if=/dev/urandom of=temp.file.02 bs=1M count=2
    2+0 records in
2+0 records out
2097152 bytes (2.1 MB) copied, 25.5691 s, 82.0 kB/s


ls -lrt
total 6152
-rw-r--r-- 1 root root 2097152 Feb  3 16:57 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 16:58 temp.file.01
-rw-r--r-- 1 root root 2097152 Feb  3 16:59 temp.file.02



zfs list -t all
NAME                      USED  AVAIL  REFER  MOUNTPOINT
c_pool                   6.17M  3.84G    19K  /c_pool
c_pool/zfs01             6.04M  3.84G  6.03M  /zfs01
c_pool/zfs01@snapshot01     9K      -  2.02M  -
c_pool/zfs01@snapshot02     9K      -  4.02M  -
```

```
# If we try to rollback to oldest snapshot:

zfs rollback c_pool/zfs01@snapshot01
cannot rollback to 'c_pool/zfs01@snapshot01': more recent snapshots
or bookmarks exist
use '-r' to force deletion of the following snapshots and bookmarks:
c_pool/zfs01@snapshot02


# If we need to rollback to first snapshot, first we have to rollback
to the newer one,
# destroy it and, then, rollback to the oldest snapshot

zfs rollback c_pool/zfs01@snapshot02


ll
total 4101
-rw-r--r-- 1 root root 2097152 Feb  3 16:57 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 16:58 temp.file.01


zfs rollback c_pool/zfs01@snapshot01
cannot rollback to 'c_pool/zfs01@snapshot01': more recent snapshots
or bookmarks exist
use '-r' to force deletion of the following snapshots and bookmarks:
c_pool/zfs01@snapshot02


zfs destroy c_pool/zfs01@snapshot02


zfs rollback c_pool/zfs01@snapshot01
```

```
ls -lrt
total 2051
-rw-r--r-- 1 root root 2097152 Feb  3 16:57 temp.file.00



 # Otherwise, we could have used '-r' option to recursively rollback
to desired snapshot.
# This will destroy all intermediate snapshots.

zfs rollback -r c_pool/zfs01@snapshot01



zfs list -t all
NAME                      USED  AVAIL  REFER  MOUNTPOINT
c_pool                   2.15M  3.84G    19K  /c_pool
c_pool/zfs01             2.02M  3.84G  2.02M  /zfs01
c_pool/zfs01@snapshot01     1K      -  2.02M  -




# Displaying snapshots
# ------------------------------------------------------------------
-----------------------

zfs list -t snapshot
NAME                      USED  AVAIL  REFER  MOUNTPOINT
c_pool/zfs01@snapshot01     1K      -  2.02M  -




# Accessing snapshot contents
# ------------------------------------------------------------------
-----------------------
```

```
# As long as zfs "snapdir" property is set to "visible", snapshot's
contents should be
# accessible by entering ".zfs" directory under zfs's mount point.

# There should be one directory per snapshot containing
directory/file structures existing
# at the moment snapshot was taken

zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                  8.15M  9.62G    19K  /c_pool
c_pool/zfs01            8.05M  9.62G  8.03M  /zfs01
c_pool/zfs01@snapshot01   10K      -  4.02M  -
c_pool/zfs01@snapshot02   11K      -  6.03M  -


zfs get all  c_pool/zfs01 | grep snapdir
c_pool/zfs01  snapdir               visible                local


 cd /zfs01/.zfs/snapshot

ls -l
total 1
drwxr-xr-x. 2 root root 3 Feb  3 21:55 snapshot01
drwxr-xr-x. 2 root root 4 Feb  3 21:55 snapshot02

 # Each of the directories contains directory/file structures
existing at the moment when
# snapshot was taken:

ls -lR
.:
total 1
drwxr-xr-x. 2 root root 3 Feb  3 21:55 snapshot01
drwxr-xr-x. 2 root root 4 Feb  3 21:55 snapshot02

./snapshot01:
```

```
total 2051
-rw-r--r--. 1 root root 2097152 Feb  3 21:51 temp.file.01

./snapshot02:
total 4101
-rw-r--r--. 1 root root 2097152 Feb  3 21:51 temp.file.01
-rw-r--r--. 1 root root 2097152 Feb  3 21:55 temp.file.02


# I experienced some troubles while trying to access snapshot contents on virtual
systems
# (both VMWare and Oracle VM VirtualBox virtual systems). For the moment, I'll let
it drop



# because virtual servers are not the main target of this procedure.









# Cloning a snapshot
#
----------
------------------------------------------------------------------------
-----------

# Let's create one zpool, with one zfs and one snapshot

zpool create c_pool sdb sdc
zfs create -o mountpoint=/zfs01 c_pool/zfs01
cd /zfs01
dd if=/dev/urandom of=temp.file.00 bs=1M count=2
zfs snapshot c_pool/zfs01@snapshot01
dd if=/dev/urandom of=temp.file.01 bs=1M count=2



zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
```

```
c_pool                   4.21M  3.84G    19K  /c_pool
c_pool/zfs01             4.03M  3.84G  4.02M  /zfs01
c_pool/zfs01@snapshot01     9K      -  2.02M  -
```

zfs clone c_pool/zfs01@snapshot01 c_pool/zfs02

```
# Snapshot c_pool/zfs01@snapshot01 has been copied and will be
writeable on
# c_pool/zfs02 clone
```

zfs list -t all

```
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                  4.22M  3.84G    19K  /c_pool
c_pool/zfs01            4.03M  3.84G  4.02M  /zfs01
c_pool/zfs01@snapshot01    9K      -  2.02M  -
c_pool/zfs02               1K  3.84G  2.02M  /c_pool/zfs02
```

ls -lrt /zfs01

```
total 4101
-rw-r--r-- 1 root root 2097152 Feb  3 17:15 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 17:15 temp.file.01
```

ls -lrt /c_pool/zfs02

```
total 2051
-rw-r--r-- 1 root root 2097152 Feb  3 17:15 temp.file.00
```

dd if=/dev/urandom of=/c_pool/zfs02/temp.file.02 bs=1M count=2

```
2+0 records in
2+0 records out
2097152 bytes (2.1 MB) copied, 25.0426 s, 83.7 kB/s
```

ls -lrt /c_pool/zfs02

```
total 4101
-rw-r--r-- 1 root root 2097152 Feb  3 17:15 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 17:17 temp.file.02


zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                  6.23M  3.84G    19K  /c_pool
c_pool/zfs01            4.03M  3.84G  4.02M  /zfs01
c_pool/zfs01@snapshot01    9K      -  2.02M  -
c_pool/zfs02            2.01M  3.84G  4.02M  /c_pool/zfs02



# Removing a clone/snapshot
# ------------------------------------------------------------------------
-----------------------

zfs destroy c_pool/zfs02


zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                  4.22M  3.84G    19K  /c_pool
c_pool/zfs01            4.03M  3.84G  4.02M  /zfs01
c_pool/zfs01@snapshot01    9K      -  2.02M  -


# Note: If a snapshot has one or more clones we won't be able to
destroy it unless clones
#       are destroyed first:

zfs list -t all
NAME                     USED  AVAIL  REFER  MOUNTPOINT
c_pool                  4.23M  3.84G    19K  /c_pool
c_pool/zfs01            4.03M  3.84G  4.02M  /zfs01
c_pool/zfs01@snapshot01    9K      -  2.02M  -
c_pool/zfs02               1K  3.84G  2.02M  /c_pool/zfs02
```

```
zfs destroy c_pool/zfs01@snapshot01
cannot destroy 'c_pool/zfs01@snapshot01': snapshot has dependent
clones
use '-R' to destroy the following datasets:
c_pool/zfs02


zfs destroy c_pool/zfs02

zfs destroy c_pool/zfs01@snapshot01

zfs list -t all
NAME           USED  AVAIL  REFER  MOUNTPOINT
c_pool        4.15M  3.84G    19K  /c_pool
c_pool/zfs01  4.02M  3.84G  4.02M  /zfs01



# Promoting a clone
# ------------------------------------------------------------------
-----------------------

# Once a clone in place, we can use to replace original dataset. We
will make clone
# independent of the snapshot it was created from and, then, remove
snapshot(s) and
# origin filesystem so our clone will replace it


zfs list -t all
NAME            USED  AVAIL  REFER  MOUNTPOINT
c_pool         4.16M  3.84G    19K  /c_pool
c_pool/product  4.02M  3.84G  4.02M  /product    <----


ll /product
```

```
total 4101
-rw-r--r-- 1 root root 2097152 Feb  3 17:15 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 17:20 temp.file.01
```

zfs snapshot c_pool/product@snapshot01

zfs list -t all

```
NAME                        USED  AVAIL  REFER  MOUNTPOINT
c_pool                     4.16M  3.84G    19K  /c_pool
c_pool/product             4.02M  3.84G  4.02M  /product
c_pool/product@snapshot01      0      -  4.02M  -           <----
```

zfs clone -o mountpoint=/clone c_pool/product@snapshot01 c_pool/clone

zfs list -t all

```
NAME                        USED  AVAIL  REFER  MOUNTPOINT
c_pool                     4.19M  3.84G    19K  /c_pool
c_pool/clone                  1K  3.84G  4.02M  /clone
<----
c_pool/product             4.02M  3.84G  4.02M  /product
c_pool/product@snapshot01      0      -  4.02M  -
```

```
# Make some modifications to clone (this is clone's purpose indeed)
```

vi /clone/mynewfile

```
[...]
```

ll /product /clone

```
/product:
total 4101
-rw-r--r-- 1 root root 2097152 Feb  3 17:15 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 17:20 temp.file.01
```

```
/clone:
total 4102
-rw-r--r-- 1 root root      20 Feb  3 17:23 mynewfile
-rw-r--r-- 1 root root 2097152 Feb  3 17:15 temp.file.00
-rw-r--r-- 1 root root 2097152 Feb  3 17:20 temp.file.01



# Promote the clone

zfs promote c_pool/clone



# Among other things, the existing snapshot becomes dependent of the
clone that has been
# promoted.
# Take a look to new "USED" space value for the clone too (It's not a
copy of the clone
# anymore but an independent dataset)

zfs list -t all
NAME                        USED  AVAIL  REFER  MOUNTPOINT
c_pool                     4.55M  3.84G    19K  /c_pool
c_pool/clone               4.03M  3.84G  4.02M  /clone    <----
c_pool/clone@snapshot01       9K      -  4.02M  -         <----
c_pool/product                 0  3.84G  4.02M  /product



# If we try, for instance, to remove the promoted clone we won't be
able because now
# it has a dependent snapshot:

zfs destroy c_pool/clone
cannot destroy 'c_pool/clone': filesystem has children
use '-r' to destroy the following datasets:
c_pool/clone@snapshot01
```

```
# Should we need a current snapshot of promoted clone, we have to
create a new one because
# the existing one is a snapshot from the original contents


# Now we are ready to replace the original dataset with the new one
(promoted clone).
# Take into account that to be able to rename mountpoints (if needed)
we'll have to
# remount the datasets

zfs rename c_pool/product c_pool/product.orig

zfs get all c_pool/product.orig | grep mountpoint
c_pool/product.orig  mountpoint            /product
local

zfs set mountpoint=/product.orig c_pool/product.orig

# On RHEL 7 F.S. has been already mounted so following two lines are
not necessary:
mkdir /product.orig
zfs mount c_pool/product.orig

zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
c_pool               4.62M  3.84G    19K  /c_pool
c_pool/clone         4.03M  3.84G  4.02M  /clone
c_pool/product.orig     9K  3.84G  4.02M  /product.orig


zfs rename c_pool/clone c_pool/product

zfs set mountpoint=/product c_pool/product


# On RHEL 7 F.S. has been already mounted so following lines is not
necessary:
```

```
zfs mount c_pool/product

zfs list -t all
NAME                        USED  AVAIL  REFER  MOUNTPOINT
c_pool                     4.24M  3.84G    19K  /c_pool
c_pool/product             4.03M  3.84G  4.02M  /product
c_pool/product@snapshot01     9K      -  4.02M  -
c_pool/product.orig           9K  3.84G  4.02M  /product.orig
```