Aaron Toponce

{ 2012 12 10 }

ZFS Administration, Part V- Exporting and Importing zpools

Table of Contents

Our previous post finished our discussion about VDEVs by going into great detail about the ZFS ARC. Here, we'll continue our discussion about ZFS storage pools, how to migrate them across systems by exporting and importing the pools, recovering from destroyed pools, and how to upgrade your storage pool to the latest version.

Motivation

As a GNU/Linux storage administrator, you may come across the need to move your storage from one server to another. This could be accomplished by physically moving the disks from one storage box to another, or by copying the data from the old live running system to the new. I will cover both cases in this series. The latter deals with sending and receiving ZFS snapshots, a topic that will take us some time getting to. This post will deal with the former; that is, physically moving the drives.

One slick feature of ZFS is the ability to export your storage pool, so you can disassemble the drives, unplug their cables, and move the drives to another system. Once on the new system, ZFS gives you the ability to import the storage pool, regardless of the order of the drives. A good demonstration of this is to grab some USB sticks, plug them in, and create a ZFS storage pool. Then export the pool, unplug the sticks, drop them into a hat, and mix them up. Then, plug them back in at any random order, and re-import the pool on a new box. In fact, ZFS is smart enough to detect endianness. In other words, you can export the storage pool from a big endian system, and import the pool on a little endian system, without hiccup.

Exporting Storage Pools

When the migration is ready to take place, before unplugging the power, you need to export the storage pool. This will cause the kernel to flush all pending data to disk, writes data to the disk acknowledging that the export was done, and removes all knowledge that the storage pool existed in the system. At this point, it's safe to shut down the computer, and remove the drives.

If you do not export the storage pool before removing the drives, you will not be able to import the drives on the new system, and you might not have gotten all unwritten data flushed to disk. Even though the data will remain consistent due to the nature of the filesystem, when importing, it will appear to the old system as a faulted pool. Further, the destination system will refuse to import a pool that has not been explicitly exported. This is to prevent race conditions with network attached storage that may be already using the pool.

To export a storage pool, use the following command:

```
# zpool export tank
```

This command will attempt to unmount all ZFS datasets as well as the pool. By default, when creating ZFS storage pools and filesystems, they are automatically mounted to the system. There is no need to explicitly unmount the filesystems as you with with ext3 or ext4. The export will handle that. Further, some pools may refuse to be exported, for whatever reason. You can pass the "-f" switch if needed to force the export

Importing Storage Pools

Once the drives have been physically installed into the new server, you can import the pool. Further, the new system may have multiple pools installed, to which you will want to determine which pool to import, or to import them all. If the storage pool "tank" does not already exist on the new server, and this is the pool you wish to import, then you can run the following command:

```
# zpool import tank
# zpool status tank
 state: ONLINE
 scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            sde     ONLINE       0     0     0
            sdf     ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            sdg     ONLINE       0     0     0
            sdh     ONLINE       0     0     0
          mirror-2  ONLINE       0     0     0
            sdi     ONLINE       0     0     0
            sdj     ONLINE       0     0     0

errors: No known data errors
```

Your storage pool state may not be "ONLINE", meaning that everything is healthy. If the system does not recognize a disk in your pool, you may get a "DEGRADED" state. If one or more of the drives appear as faulty to the system, then you may get a "FAULTED" state in your pool. You will need to troubleshoot what drives are causing the problem, and fix accordingly.

You can import multiple pools simultaneously by either specifying each pool as an argument, or by passing the "-a" switch for importing all discovered pools. For importing the two pools "tank1" and "tank2", type:

```
# zpool import tank1 tank2
```

For importing all known pools, type:

```
# zpool import -a
```

Recovering A Destroyed Pool

If a ZFS storage pool was previously destroyed, the pool can still be imported to the system. Destroying a pool doesn't wipe the data on the disks, so the metadata is still in tact, and the pool can still be discovered. Let's take a clean pool called "tank", destroy it, move the disks to a new system, then try to import the pool. You will need to pass the "-D" switch to tell ZFS to import a destroyed pool. Do not provide the pool name as an argument, as you would normally do:

```
(server A)# zpool destroy tank
(server B)# zpool import -D
   pool: tank
     id: 17105118590326096187
  state: ONLINE (DESTROYED)
 action: The pool can be imported using its name or numeric identifier.
 config:

        tank         ONLINE
          mirror-0  ONLINE
            sde     ONLINE
            sdf     ONLINE
          mirror-1  ONLINE
            sdg     ONLINE
            sdh     ONLINE
          mirror-2  ONLINE
            sdi     ONLINE
            sdj     ONLINE

   pool: tank
     id: 2911384395464928396
  state: UNAVAIL (DESTROYED)
 status: One or more devices are missing from the system.
 action: The pool cannot be imported. Attach the missing
        devices and try again.
   see: http://zfsonlinux.org/msg/ZFS-8000-6X
 config:

        tank          UNAVAIL  missing device
          sdk         ONLINE
          sdr         ONLINE

        Additional devices are known to be part of this pool, though their
        exact configuration cannot be determined.
```

Notice that the state of the pool is "ONLINE (DESTROYED)". Even though the pool is "ONLINE", it is only partially online. Basically, it's only been discovered, but it's not available for use. If you run the "df" command, you will find that the storage pool is not mounted. This means the ZFS filesystem datasets are not available, and you currently cannot store data into the pool. However, ZFS has found the pool, and you can bring it fully

ONLINE for standard usage by running the import command one more time, this time specifying the pool name as an argument to import:

```
(server B)# zpool import -D tank
cannot import 'tank': more than one matching pool
import by numeric ID instead
(server B)# zpool import -D 17105118590326096187
(server B)# zpool status tank
  pool: tank
 state: ONLINE
 scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            sde     ONLINE       0     0     0
            sdf     ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            sdg     ONLINE       0     0     0
            sdh     ONLINE       0     0     0
          mirror-2  ONLINE       0     0     0
            sdi     ONLINE       0     0     0
            sdj     ONLINE       0     0     0

errors: No known data errors
```

Notice that ZFS was warning me that it found more than on storage pool matching the name "tank", and to import the pool, I must use its unique identifier. So, I pass that as an argument from my previous import. This is because in my previous output, we can see there are two known pools with the pool name "tank". However, after specifying its ID, I was able to successfully bring the storage pool to full "ONLINE" status. You can identify this by checking its status:

```
# zpool status tank
  pool: tank
 state: ONLINE
status: The pool is formatted using an older on-disk format.  The pool can
        still be used, but some features are unavailable.
action: Upgrade the pool using 'zpool upgrade'.  Once this is done, the
        pool will no longer be accessible on older software versions.
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        tank        ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            sde     ONLINE       0     0     0
            sdf     ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            sdg     ONLINE       0     0     0
            sdh     ONLINE       0     0     0
          mirror-2  ONLINE       0     0     0
```

```
            sdi     ONLINE       0     0     0
            sdj     ONLINE       0     0     0
```

Upgrading Storage Pools

One thing that may crop up when migrating disk, is that there may be different pool and filesystem versions of the software. For example, you may have exported the pool on a system running pool version 20, while importing into a system with pool version 28 support. As such, you can upgrade your pool version to use the latest software for that release. As is evident with the previous example, it seems that the new server has an update version of the software. I am going to upgrade.

WARNING: Once you upgrade your pool to a newer version of ZFS, older versions will not be able to use the storage pool. So, make sure that when you upgrade the pool, you know that there will be no need for going back to the old system. Further, there is no way to revert the upgrade and revert to the old version.

First, we can see a brief description of features that will be available to the pool:

```
# zpool upgrade -v
This system is currently running ZFS pool version 28.

The following versions are supported:

VER  DESCRIPTION
```

```
 ---  ---------------------------------------------------------
  1   Initial ZFS version
  2   Ditto blocks (replicated metadata)
  3   Hot spares and double parity RAID-Z
  4   zpool history
  5   Compression using the gzip algorithm
  6   bootfs pool property
  7   Separate intent log devices
  8   Delegated administration
  9   refquota and refreservation properties
  10  Cache devices
  11  Improved scrub performance
  12  Snapshot properties
  13  snapused property
  14  passthrough-x aclinherit
  15  user/group space accounting
  16  stmf property support
  17  Triple-parity RAID-Z
  18  Snapshot user holds
  19  Log device removal
  20  Compression using zle (zero-length encoding)
  21  Deduplication
  22  Received properties
  23  Slim ZIL
  24  System attributes
  25  Improved scrub stats
  26  Improved snapshot deletion performance
 27  Improved snapshot creation performance
 28  Multiple vdev replacements

For more information on a particular version, including supported releases,
see the ZFS Administration Guide.
```

So, let's perform the upgrade to get to version 28 of the pool:

```
# zpool upgrade -a
```

As a sidenote, when using ZFS on Linux, the RPM and Debian packages will contain an /etc/init.d/zfs init script for setting up the pools and datasets on boot. This is done by importing them on boot. However, at shutdown, the init script does not export the pools. Rather, it just unmounts them. So, if you migrate the disk to another box after only shutting down, you will be not be able to import the storage pool on the new box.

Conclusion

There are plenty of situations where you may need to move disk from one storage server to another. Thankfully, ZFS makes this easy with exporting and importing pools. Further, the "zpool" command has enough subcommands and switches to handle the most common scenarios when a pool will not export or import. Towards the very end of the series, I'll discuss the "zdb" command, and how it may be useful here. But at this point, steer clear of zdb, and just focus on keeping your pools in order, and properly exporting and importing them as needed.