

Database cloning with ZFS - Part I



Most of Pure FlashArray customers use Oracle on ASM or cooked filesystems like EXT4 or XFS. Rarely I do see ZFS and generally, those are Solaris users. Still, I do periodically get questions about ZFS with Pure FlashArray, specifically on database cloning where the database is hosted on FlashArray volumes. Hence, this post is to explain the database cloning process with ZFS. I will not even go into the pros and cons of ZFS but certainly wanted to showcase what is possible with ZFS on Pure FlashArray.

Purely from a performance perspective for Pure FlashArray customers, I would recommend ASM over ZFS for Oracle database as the data services offered by ZFS like dedupe, compression, encryption, snapshots and cloning are best handled by the storage array allowing the database host CPUs to focus on, database. As a matter of fact, we don't even recommend using NORMAL/HIGH redundancy for ASM or RAID-Z for ZFS as Purity (the operating system of Pure FlashArray) provides RAID-3D behind the scene in addition to all data services. Pure FlashArray offers the best snapshot functionality that allows the DBAs to clone database instances instantly consuming only metadata space at the time of creation. See this [post](#) for more details on Pure FlashRecover snapshots.

The source database that is hosted on ZFS on top of Pure FlashArray volume(s) is generally cloned to another host but I do constantly hear from customers wanting to attach the cloned volume to the same host as most of them have multiple development databases running on the same host and they use different snapshots from the same source database to create the clones.

This is the first part of two-part series of database cloning with ZFS. In this post, I will cover cloning the database on to another host. In the second part, I will cover cloning the database on to the same host which is generally not recommended by Oracle be it on ASM or ZFS due to the fact that there is a high chance for messing up the database.

Assumption

1. The following cloning process is based on single pool model where the database is hosted within a single pool. If your environment uses multiple pools (separate pools for DB, redo, FRA) then remember to repeat any zpools commands given below for the other pools as well.

Note: As I had access only to Linux systems within my lab, the following procedures were performed and tested with ZFS on Linux.

Cloning a database on to a different host

One time process

1. Take a snapshot of the FlashArray volume(s) that makes up the source database. (If the database is made up of multiple volumes, make sure to either include all volumes in the same

- command or preferably use protection group that comprises of all relevant volumes, when taking the snapshot for consistency).
2. Copy the snapshot to the new volume(s) and attach to the target host. This can be performed on the Pure GUI or through CLI command, *purevol copy*.
 3. On the target host run
[crayon-6424b1a87b166121462716/]
 4. On the target host, recreate the database with the new database name. (Make sure to update the *create control file* script with right directory structure and also create the adump and admin directories for the new cloned instance on the target server).

Periodic refresh process

1. On the target host, shut down the target instance.
2. On the target host, run
[crayon-6424b1a87b16f433598740/]
3. Take a snapshot of the source database FlashArray volume.
4. Copy the snapshot onto the same target volume(s) by overwriting it. This can only be done through Purity CLI or REST-based API and not through GUI for safety measures. (If the database is made up of multiple volumes, repeat the following command for every volume).
[crayon-6424b1a87b170520851836/]
5. On the target host, run
[crayon-6424b1a87b174504755595/]
6. On the target host, recreate the database with the new database name. (Make sure to update the create control file script with right directory structure and also create the adump and admin directories for the new cloned instance on the target server).