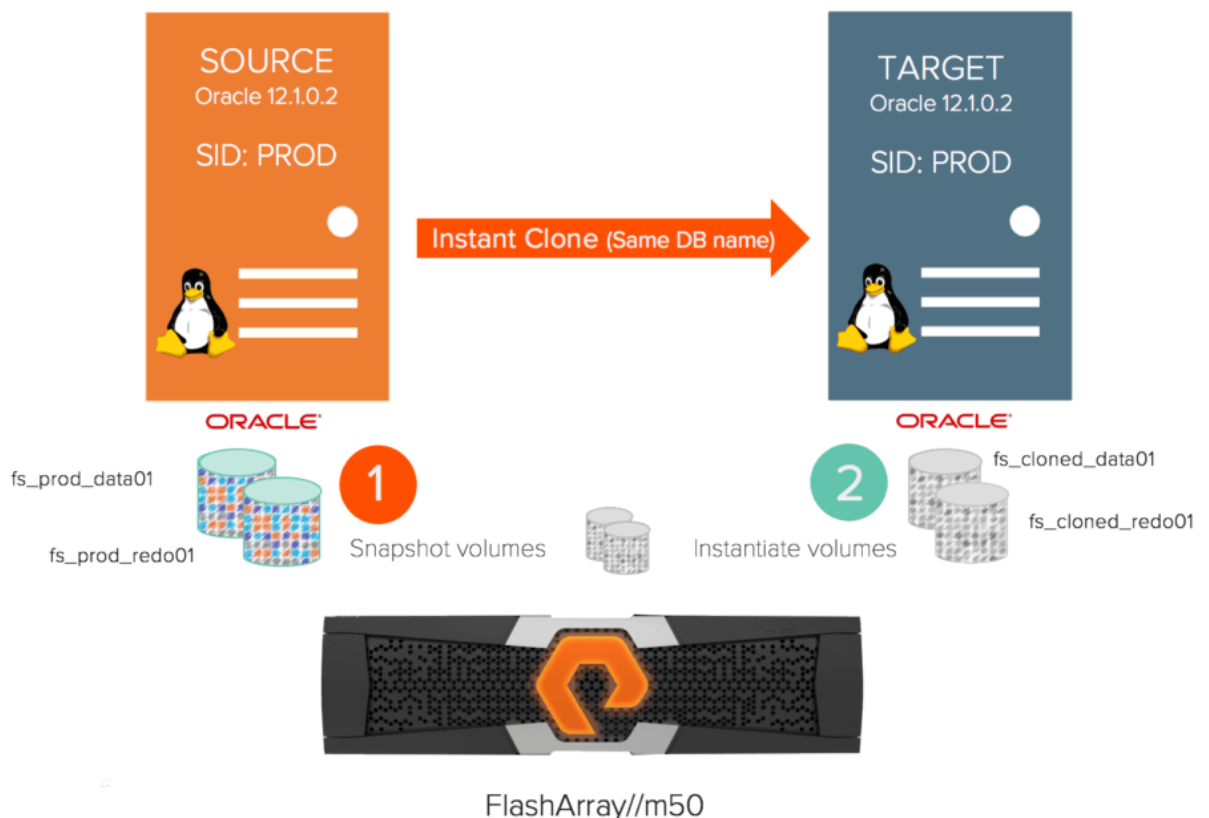


# Oracle Database cloning on Pure FlashArray - Part 1

## Instant Clone



FlashRecover Snapshots are crash-consistent snapshots and as such it meets the requirements laid out by Oracle (MOS: ID 604683.1).

Following are some of the Oracle database environment types we see in our customers;

- 1) Oracle databases on physical servers using filesystem
- 2) Oracle databases on physical servers using ASM
- 3) Oracle databases on virtual servers using filesystem (either VMFS or RDM)
- 4) Oracle databases on virtual servers using ASM (either VMFS or RDM)

However the database is laid out, if it is on Pure Storage, database cloning is not only possible but very simple and quick to accomplish.

**Cloning Definition:** In this blog, whenever we say cloning, we mean that the source database is snapshotted and the target database is cloned out of that snapshot at that specific time on snapshot and not beyond the snapshot time. It is also called as *restartable databases* because when Oracle restarts the database located on clones of a FlashRecover snapshot, it performs “crash recovery” or “instance

recovery” automatically where it rolls all committed changes forward and rolls back any uncommitted changes, making the database consistent before opening it. In essence, this is similar to what would happen when the DB server crashed or an Oracle “shutdown abort” command was issued, where Oracle when started checks if the previous shutdown was normal and if not, automatically performs the “crash recovery” to keep the database consistent.

**Why is it important?** If you wanted to clone a database beyond the time of the snapshot, aka point-in-time recovery, you need the database to be in archivelog mode and have access to all the archivelogs and latest control file on top of the datafiles snapshot. If you really don’t care about that, which is predominantly the most use cases, all you need is the snapshot of datafiles, control file and online redo logs.

Simply put, database cloning requires recovery and as you might be aware, recovery options varies based on availability of key entities like control files, archived logs, online redo logs etc.,

**Use cases:** This type of cloning is well suited for DevOps scenarios like application development, testing of database layout changes, testing new database management system versions, analytics and reporting.

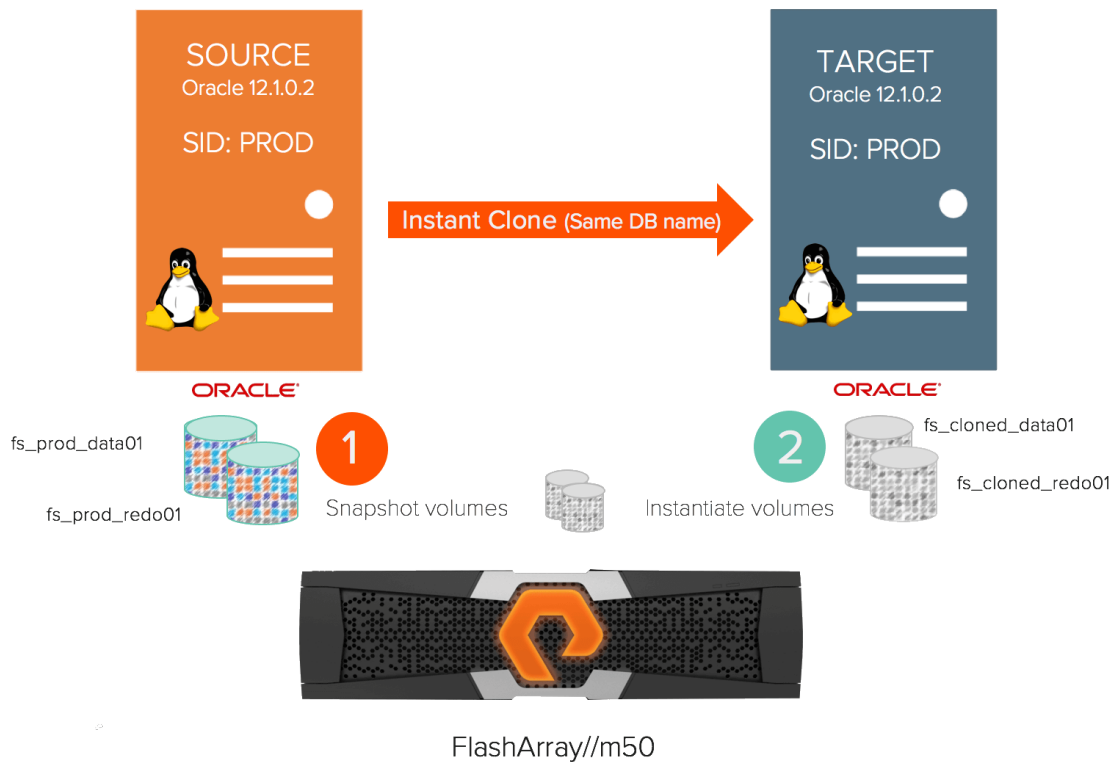
## **Part 1 - Oracle Database cloning on Pure FlashArray - Using physical servers & cooked filesystem**

This is in follow up to the [FlashRecover Snapshots post](#) and part 1 of 4 part blog series that illustrates the simplicity of instantly cloning an Oracle database specifically on physical servers and using filesystem.

Environment

Following diagram shows the setup in my environment where the source database (PROD) is made up of two volumes (fs\_prod\_data01 and fs\_prod\_redo01) and the target server will have two corresponding cloned volumes (fs\_cloned\_data01 and fs\_cloned\_redo01). This works on single instance Oracle on cooked filesystem.

# Instant Clone



## Sequence of steps

1. Unmount the target filesystems.
2. Take snapshot of the source volumes on Pure and copy them on to the target volumes using “force” option (equivalent to an overwrite).  
*Note: Overwriting is enabled only through CLI or REST based APIs and not through GUI to avoid any accidental overwrites.*
3. Mount the target filesystems.
4. Startup the Oracle database.

**Note: Irrespective of the Oracle database version, it need not have to be placed in hot backup mode for the FlashRecover snapshot to work on restartable databases.**

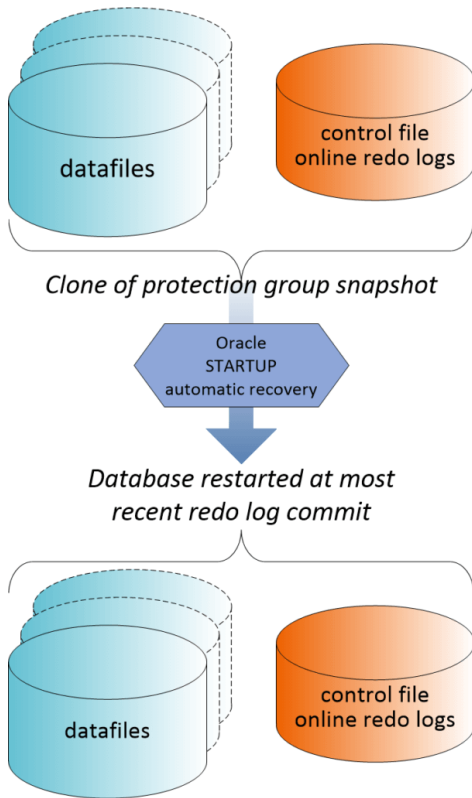
## Instant Database Cloning Prerequisites

1. The target volumes/luns should have been discovered and mounted to target server.
2. Target server to have the same Oracle binaries installed as that of source.
3. Copy the init.ora file from source and make the necessary changes to reflect the target server.

## Snapshots Requirements

For instant database cloning to work, the database should be in crash-consistent state, i.e., all datafiles, all online redo logs and control files are all in their respective states of the instant of the crash (“snapshot”). As this might be across multiple volumes/luns, protection group should be used to group all these volumes

and snapped. Following diagram shows the requirement.



## Instant Database cloning Script

The crux of the cloning script is based on Pure FlashArray's CLI commands based on bash. Pure Storage FlashArray enables access of the storage across multiple programming interfaces like REST based API, PowerShell SDK, PowerShell Toolkit and Python Toolkit on top of CLI.

[crayon-6424b8fa43a60558015797/]

The first command issues a snapshot of the protection group with a predefined SUFFIX. The next two commands forcefully copies (overwrites) the volumes individually onto its corresponding target volumes. Rest of the script includes unmounting/mounting filesystems and shutdown/startup of the database.

The complete script can be found on my [github repository](#).