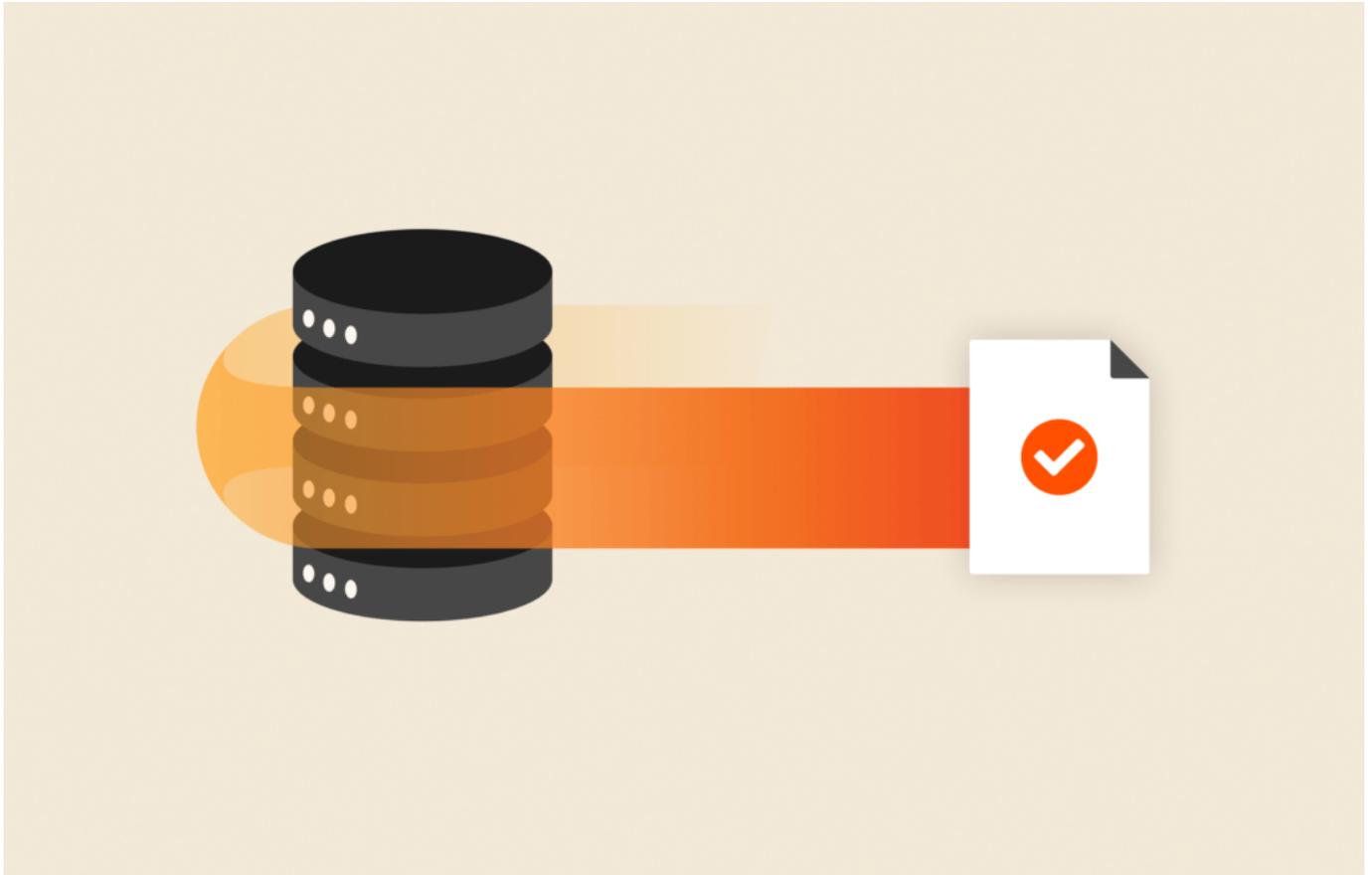


Rapidly Recovering from SQL Server Data Mishaps with FlashArray Snapshots



This article originally appeared on [Andy Yun's blog](#). It has been republished with the author's credit and consent.

Almost all of us have experienced a time when someone incorrectly modifies or deletes a subset of data in a database. And usually, the only recourse is to run a database restore to get the data that we need back.

But...what if we cannot simply run a RESTORE operation over the entire database? Maybe the accident occurred a few hours prior and you cannot roll everything back. You need to do a side-by-side RESTORE.

But...what if the database in question is absolutely massive and will take hours to restore? And what if your server doesn't have enough available storage to fit a

second copy of that database at all?

FlashArray Snapshots to the Rescue

If you're fortunate to have your SQL Servers backed by Pure Storage, you can solve this quickly with minimal headache. The key is [FlashArray™ snapshots](#).

This entire process does *not* require application-consistent snapshots either. You can utilize crash-consistent snapshots (that do *not* require VSS and do *not* stun your SQL Server when taken), and hopefully, you're already taking these today on your FlashArray system. Because of how our snapshots work behind the scenes, you will not consume additional capacity on your storage array either. Best of all, it does not matter if your database was 50GB or 50TB—all of these actions will be nearly instantaneous.

I Need to Execute an Object-level Restore *Right Now*

Here is a step-by-step video walkthrough. Watch and pause it along the way to emulate the steps I'm doing.

1. Add a new volume(s) to your SQL Server of the exact same size as the existing volume(s) containing your database's data and log files.
2. In FlashArray, in the Protection Group's UI, select a snapshot, and find the volume(s) you wish to clone (aka "Copy Snapshot"). Overwrite the volume(s) created in Step 1 with the member volume of the Protection Group Snapshot.
3. In the Windows OS, in Disk Management, find the newly added volume(s)/disk(s) and Set Online.
4. Attach the newly cloned database files in SQL Server (with a different name, of course).
5. Use T-SQL to insert/update/copy from your restored database back to your main database.

6. Clean up: Drop the restored database, set the volume(s)/disk(s) offline in Windows, and delete the disks from your VM.

Assumptions and Prerequisites:

- Your SQL Server is a VMware VM—this will work with bare metal as well but with a slight adjustment to the steps.
- Your VMware VM is using vVols—this will work with RDMs as well but with a slight adjustment to the steps. VMFS will also work, but you'll need a slightly different set of steps, which I'll cover in a future blog.
- The volumes containing your SQL Server data and log files are being snapshotted by FlashArray on a pre-scheduled basis, all as part of a Protection Group.

[Read: Hyper-V: Cluster Shared Volumes, SQL Server, and FlashArray Snapshots](#)

Conclusion

Hopefully this blog has given you a sense of how it easy it is to leverage FlashArray snapshots on Hyper-V. I'd encourage you to think creatively on how you can start leveraging snapshots in your workplace. Think about how snapshots can be used creatively, in lieu of having to take a database backup and restore it "somewhere else" - that scenario alone is a huge time saver.