PURESTORAGE®

TECHNICAL WHITE PAPER

SQL Server High Availability with VMware and Pure Storage

Combine Pure Storage®, VMware vSphere, and Microsoft SQL Server to increase resiliency.

Contents

Introduction

In today's modern on-premises data center, many organizations virtualize Microsoft SQL Server deployments. Arguably the most popular choice for virtualization is VMware vSphere. vSphere is the umbrella name for the entire virtualization platform of VMware. ESXi is the hypervisor component of vSphere. Deployments of Microsoft SQL Server inside a virtual machine (VM) not only have to perform well but also be up and running when the business needs them. This uptime is known as "availability."

Availability goes beyond local high availability in a single data center and can extend into disaster recovery across multiple data centers. Availability solutions that span different locations are known as disaster recovery while solutions in the same data center are high availability. Both strive to achieve the same goal: business continuity with minimal, and hopefully, transparent outages. While certain features or techniques may be shared between high availability and disaster recovery, the mindset for each is distinct and requires different planning. This is why it is important to craft an availability strategy that takes into account many aspects including the underlying architecture. Pure Storage, with its choice of operating system, is an integral part of that foundation when it provides the storage underneath ESXi and the VM containing Microsoft SQL Server.

This paper will cover how you can use Pure Storage, vSphere, and Microsoft SQL Server together to provide increased resiliency when virtualizing Microsoft SQL Server using vSphere. It includes how Pure-based features can enhance availability for Microsoft SQL Server instances and databases.

Availability Concepts for Virtualization

Making Microsoft SQL Server deployments highly available requires having solid requirements. The most important requirement is to have documented recovery time objectives (RTOs) and recovery point objectives (RPO). RTO is the amount of time that you have to get systems back up and running in the event of a problem. RPO is the point in time which the systems should be able to be recovered to—or later. Without defining either of those, it is effectively impossible to define an availability strategy, let alone implement a solution.

Availability can come from two layers: the platform (including hardware) and Microsoft SQL Server (usually in conjunction with some features of its underlying operating system). In the virtual world, the latter is considered an in-guest availability. An example of platform-layer availability includes ActiveCluster™ and ActiveDR™ from Pure as well as the Site Recovery Manager feature of vSphere. Features such as Always On Availability Groups (AGs) and Always On Failover Cluster Instances (FCIs) are examples of in-guest methods to make Microsoft SQL Server available.

These two approaches are not mutually exclusive and can be combined to achieve higher availability. Neither of these approaches works if you do not have a solid backup and recovery (not just restore) strategy. You should always plan for worst-case scenarios. As a general rule, it is recommended to employ a multi-layered approach when it comes to backups. For example, combining both storage-level as well as Microsoft SQL Server-based backups works well when planned correctly. This combination will be discussed further in the upcoming Snapshot Backups section.

From an implementation standpoint, the key to achieving availability is redundancy. Redundancy avoids single points of failure (SPOFs). You do not want a key component such as a single storage array to become a SPOF. Most modern hardware is deployed with redundant components (one way to increase resiliency), but catastrophic failures can and do happen. As part of a bigger discussion within your company, you need to define more than just RTOs and RPOs. What are your fault domains? A storage array is an example of a fault domain, but that could be defined for power, networking, and so on. There is a thing as too much redundancy because you will need to keep track of all your components. However, assuming any one component will never fail could lead to unexpected problems. You should avoid the illusion of availability when you deploy something (hardware, a Microsoft SQL Server feature) that is in reality it is built on SPOFs.

Disk Configuration Considerations

One way to ensure that Microsoft SQL Server is highly available and reliable is to ensure that everything at all layers is deployed correctly. Even if you are deploying an in-guest option such as an AG, it will be affected by everything else underneath it such as storage, networking, and noisy neighbors. This section will cover some highlights for using Microsoft SQL Server when you use Pure Storage arrays.

> Note: For virtualized deployments of Microsoft SQL Server with vSphere, always use the Paravirtual SCSI (PVSCSI) adapter. This has been the best practice for years, yet many still use LSI Logic.

Properly laying out your disks is important for availability as well as performance. Under ESXi, disks are virtual (also known as VMDKs). Virtual disks are pointers to some physical storage (such as a file on a file system), but look and feel like "real" disks on physical storage. Virtual Disks can also point to a raw block volume to be presented directly to the guest operating system known as Raw Device Mapping disks (RDMs or pRDMs in VMware terminology), or presented inside the guest using things such as iSCSI or SMB 3.0. One other method for virtual disks is to use Virtual Volumes (vVols). A vVol is an alternative to using traditional VMFS datastores but retains the concept of a datastore. More information on vVols can be found on the VMware site in the official documentation. vVols require support for the vSphere APIs for Storage Awareness. A Pure Storage FlashArray™ supports this capability.

First, treat a VM as if it was a physical server. This means you must adhere to standard best practices such as not putting any databases on system drives. Virtual hardware is not unlike physical and there are limitations. A VM supports up to four virtual storage adapters. Each virtual storage adapter can support multiple disks. A single virtual storage controller can affect performance.

That means you should not attach all of your virtual disks—especially ones that have databases and transaction logs—to the same virtual storage controller. A VM supports up to four virtual storage adapters. Each virtual storage adapter can support multiple disks. Organizations often have more than four disks so a 1:1 ratio from disk to controller is not always possible—or recommended. Spread the load out as evenly as possible.

When designing virtual deployments of Microsoft SQL Server, avoid "lift and shift" if at all possible. "Lift and shift" means taking the same way physical servers are deployed and replicating it on a VM right down to the disk structure. While this may be valid for some use cases, it is better to optimize for ESXi. Examples include, but are not limited to, concepts such as spreading virtual disks across multiple PVSCSI adapters, guaranteeing I/O resources at the VM-level if needed to ensure performance, and adjusting queue depths in-guest, at the host/VM layer, and at the physical layer to ensure proper throughput. VMware KB 1267 covers queue depth recommendations for popular host bus adapters. Topics like queue depth are covered in more depth in the resources linked at the end of the paper.

There are some cases, such as with FCIs, where disks must be configured in a certain way. For example, the OS disk for the nodes must be eager zeroed thick. This does not mean the storage underneath is not doing thin provisioning, but you should still follow VMware's supportability guidelines. Where this would come into play would be something like backup and recovery if it is done at the ESXi-level. Pure's recommendation is to have all disks be thin provisioned in that case. However, if another style of disk configuration is needed by ESXi, do that. Any storage put on a vVol is thin provisioned but does not take the same performance penalty as a standard VMDK on a datastore.

Check Supportability of the Solution

Always ensure that your Pure configuration is compatible/supported with your target version of ESXi. This can be done by using the VMware Compatibility Guide. Figure 1 shows an example of a generic query.

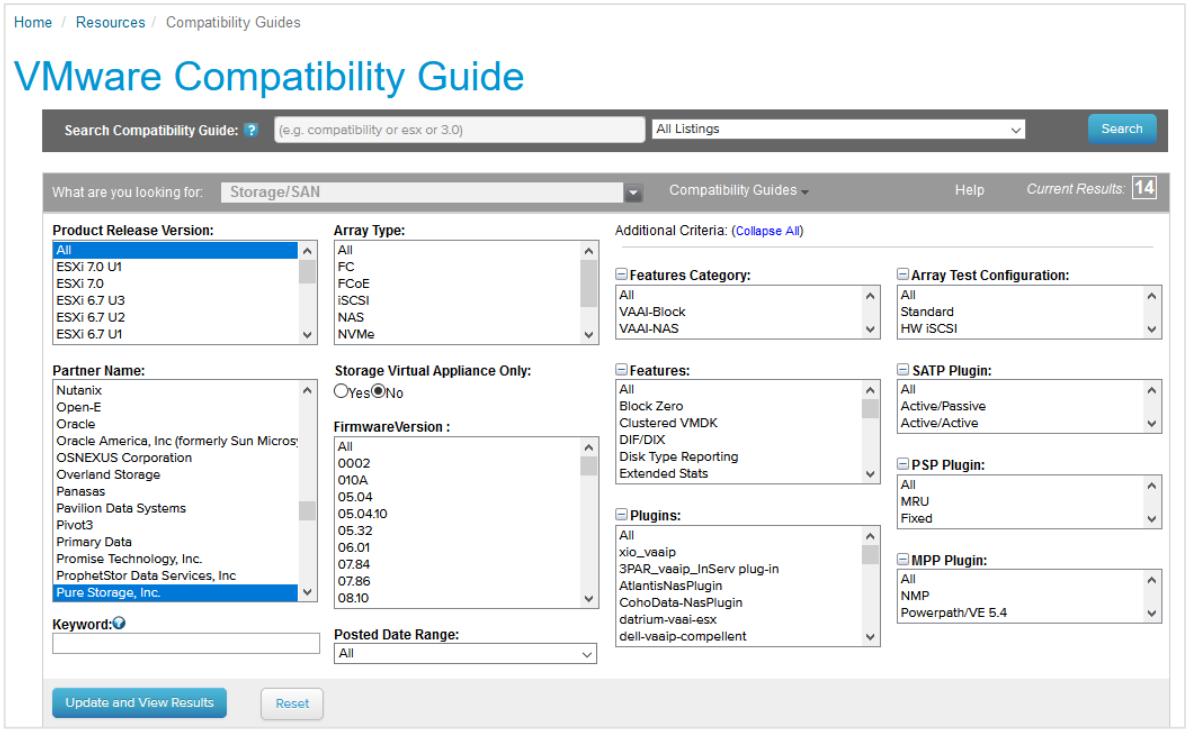
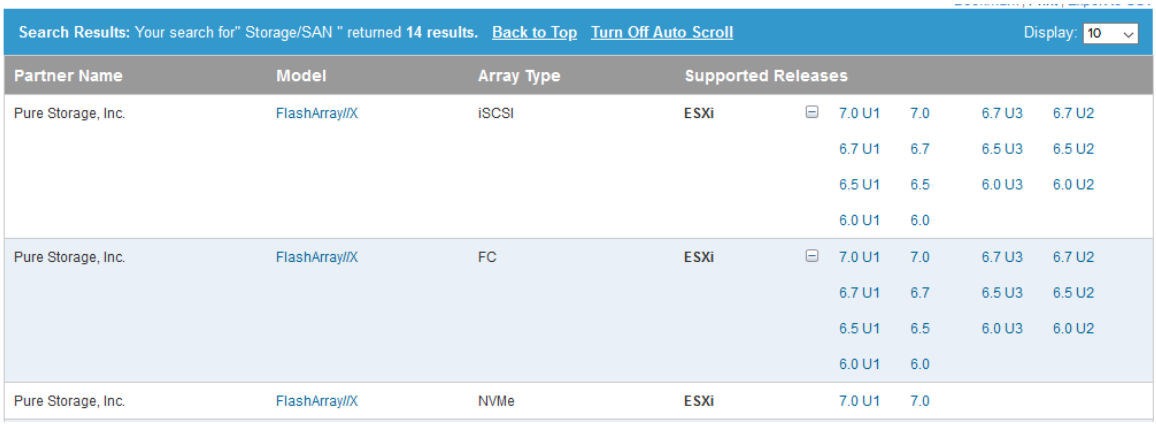


Figure 1. VMware Compatibility Guide example with Pure Storage.

Figure 2 shows some of the output of the query from Figure 1. For example, at the time of writing this paper, the FlashArray/X when configured with iSCSI and Fibre Channel are supported from ESXi 6.0 through 7.0 Update 1. The NVMe-oF variant is only supported by ESXi 7.0 and later.



Search Results: Your search for" Storage/SAN " returned **14 results.** Back to Top Turn Off Auto Scroll Display: 10

Partner Name	Model	Array Type	Supported Releases
Pure Storage, Inc.	FlashArray//X	iSCSI	ESXi 7.0 U1, 7.0, 6.7 U3, 6.7 U2, 6.7 U1, 6.7, 6.5 U3, 6.5 U2, 6.5 U1, 6.5, 6.0 U3, 6.0 U2, 6.0 U1, 6.0
Pure Storage, Inc.	FlashArray//X	FC	ESXi 7.0 U1, 7.0, 6.7 U3, 6.7 U2, 6.7 U1, 6.7, 6.5 U3, 6.5 U2, 6.5 U1, 6.5, 6.0 U3, 6.0 U2, 6.0 U1, 6.0
Pure Storage, Inc.	FlashArray//X	NVMe	ESXi 7.0 U1, 7.0

Figure 2. Results from the query shown in Figure 1.

If you click on the link, you will see further information about firmware version(s) supported and other important bits of information. Remember—a key to availability is a supported, well-implemented solution.

Increasing Microsoft SQL Server Availability for Virtualized Deployments

Two Pure Storage products are FlashArray and FlashBlade. FlashArray is a 100% NVMe storage array that provides high-performance and low-latency as well as capacity-optimized block-level storage via Fibre Channel, iSCSI, and NVMe-oF protocols. FlashArray//X is designed to support mission-critical workloads such as Microsoft SQL Server and other critical business needs such as private cloud and consolidation. FlashArray//C, engineered for high capacity, provides the same data services as //X but is optimized for cost and economics for archive and back-up of large databases. FlashBlade is designed for fast file and object storage that can be used for native Microsoft SQL Server backups files and rapid restores. For more information on rapid restores, see Fast SQL Server Backup and Restore with Pure Storage.

Snapshot Backups

This section will discuss how snapshot backups can be used with Microsoft SQL Server.

A Microsoft SQL Server Backup and Recovery Primer

To understand where snapshot backups fit into the picture, you must understand how SQL Server works. There are three main types of database backups: full (which has variants such as copy only), differential, and transaction log. A full database backup is a backup of the database at a given time. A differential backup is the data from the last full backup. Neither a full nor a differential backup truncate and back up the transaction log which is crucial for SQL Server. The transaction log is where every transaction and change are written for a database. Transaction log backups flush the transaction log to disk and allow you to recover to a more granular point in time.

Each database has a recovery model. Depending on the recovery model, SQL Server may or may not have the ability to restore to a point in time. If your databases are set to Simple, you can only use a full backup which may mean data loss. If your recovery model is Full or Bulk-logged, you can use differential and/or transaction log backups to roll the database forward to a more recent time. This concept is important when discussing storage-based options.

Of course, system administrators may also want to back up the entire VM including the OS and all of its settings. SQL Server must be taken into account when full system backups are desired, especially with VMs which means snapshot backups. That topic will be addressed in the next section.

When Microsoft SQL Server is running inside a VM, there is often a question about how backups should be performed: Should you use the platform or generate native Microsoft SQL Server backups inside the VM and have another process copy them elsewhere? The answer is "it depends."

This paper is not going to settle that debate but will show how Pure Storage works in a virtual environment as it relates to backups and virtualized Microsoft SQL Server deployments.

How Snapshot Backups Work on a FlashArray

There are two fundamental concepts when considering using platform-based backups: crash consistent and application consistent. A crash-consistent (or sometimes, inconsistent) backup is purely a file-level backup that is not aware of what is going on in Microsoft SQL Server or what is happening in memory. File servers are an example of a system type that usually work well with a crash-consistent backup. Crash-consistent backups can provide data protection for databases with the understanding that they may not work in every scenario such as seeding an AG. You should always test to make sure this strategy works well in your environment and meets your backup needs.

An application-consistent backup knows that Microsoft SQL Server is running. It is better to perform an application-consistent backup if the option is available to you. From a SQL Server perspective, an application-consistent backup is a full database backup. Assuming your databases have a recovery model that allows it, you can use the methods described above to recover the database to a more recent point in time. Application-consistent snapshot backups cannot do that alone nor can they generate differential or transaction log backups. This is why you still need to consider appropriate SQL Server-based backups to achieve point-in-time recovery.

The provider integrates with the Volume Shadow Copy Service (VSS) to quiesce and account for things like transactions that are still executing. In the Microsoft SQL Server log, you will see a freeze and a thaw of I/O. Figure 3 shows an example taken from an FCI configuration using RDMs on a FlashArray using the Microsoft SQL Server Management Studio Extension that is discussed in the next section.

11/26/2020 2:56:07 AM	Backup	BACKUP DATABASE successfully processed 0 pages in 0.235 seconds (0.000 MB/sec).
11/26/2020 2:56:07 AM	Backup	Database backed up. Database: vFCISnapDB, creation date(time): 2020/11/26(02:52:07), pages dumped: 330, first LSN: 37:168:74, last LSN: 37:199:1, number of dump devices: 1, device information: (FILE=1, TYPE=VIRTUAL_DEVIC
11/26/2020 2:56:07 AM	Backup	BACKUP DATABASE successfully processed 0 pages in 0.235 seconds (0.000 MB/sec).
11/26/2020 2:56:07 AM	Backup	BACKUP DATABASE successfully processed 0 pages in 0.207 seconds (0.000 MB/sec).
11/26/2020 2:56:07 AM	Backup	BACKUP DATABASE successfully processed 0 pages in 0.242 seconds (0.000 MB/sec).
11/26/2020 2:56:06 AM	Backup	Database backed up. Database: msdb, creation date(time): 2019/09/24(14:21:42), pages dumped: 2004, first LSN: 154:943:128, last LSN: 154:997:1, number of dump devices: 1, device information: (FILE=1, TYPE=VIRTUAL_DEVICE:
11/26/2020 2:56:06 AM	Backup	Database backed up. Database: model, creation date(time): 2003/04/08(09:13:36), pages dumped: 330, first LSN: 36:261:37, last LSN: 36:278:1, number of dump devices: 1, device information: (FILE=1, TYPE=VIRTUAL_DEVICE: {'{9'
11/26/2020 2:56:06 AM	Backup	Database backed up. Database: master, creation date(time): 2020/11/26(02:44:15), pages dumped: 501, first LSN: 217:420:178, last LSN: 217:494:1, number of dump devices: 1, device information: (FILE=1, TYPE=VIRTUAL_DEVICE:
11/26/2020 2:56:06 AM	spid66	I/O was resumed on database vFCISnapDB. No user action is required.
11/26/2020 2:56:06 AM	spid57	I/O was resumed on database model. No user action is required.
11/26/2020 2:56:06 AM	spid67	I/O was resumed on database master. No user action is required.
11/26/2020 2:56:06 AM	spid65	I/O was resumed on database msdb. No user action is required.
11/26/2020 2:56:06 AM	spid67	I/O is frozen on database master. No user action is required. However, if I/O is not resumed promptly, you could cancel the backup.
11/26/2020 2:56:06 AM	spid66	I/O is frozen on database vFCISnapDB. No user action is required. However, if I/O is not resumed promptly, you could cancel the backup.
11/26/2020 2:56:06 AM	spid65	I/O is frozen on database msdb. No user action is required. However, if I/O is not resumed promptly, you could cancel the backup.
11/26/2020 2:56:06 AM	spid57	I/O is frozen on database model. No user action is required. However, if I/O is not resumed promptly, you could cancel the backup.

Figure 3. Freeze and thaw in the Microsoft SQL Server log.

Generally, this quiescing takes a very short time, but transaction-sensitive workloads can potentially be impacted by a snapshot backup. Always test these types of backups before using them in production because if they impact something like an AG, there may be negative downstream effects. Here is the workflow for a VSS backup:

1. The requester asks the Volume Shadow Copy Service to enumerate the writers, gather the writer metadata, and prepare for shadow copy creation.
2. Each writer creates an XML description of the components and data stores that need to be backed up and provides it to the Volume Shadow Copy Service. The writer also defines a restore method, which is used for all components. The Volume Shadow Copy Service provides the writer's description to the requester, which selects the components for back up.

3. The Volume Shadow Copy Service notifies all the writers to prepare their data for making a shadow copy.
4. Each writer prepares the data as appropriate, such as completing all open transactions, rolling transaction logs, and flushing caches. When the data is ready to be shadow copied, the writer notifies the Volume Shadow Copy Service.
5. The Volume Shadow Copy Service tells the writers to temporarily freeze application write I/O requests (read I/O requests are still possible) for the few seconds that are required to create the shadow copy of the volume or volumes. The application freeze is not allowed to take longer than 60 seconds. The Volume Shadow Copy Service flushes the file system buffers and then freezes the file system, which ensures that the file system metadata is recorded correctly and the data to be shadow copied is written in a consistent order.
6. The Volume Shadow Copy Service tells the provider to create the shadow copy. The shadow copy creation period lasts no more than 10 seconds, during which all write I/O requests to the file system remain frozen.
7. The Volume Shadow Copy Service releases file system write I/O requests.
8. VSS tells the writers to thaw application write I/O requests. At this point, applications are free to resume writing data to the disk that is being shadow copied.
9. The requester can retry the process (go back to step 1) or notify the administrator to retry at a later time.
10. If the shadow copy is successfully created, the Volume Shadow Copy Service returns the location information for the shadow copy to the requester.

The whole process is represented in Figure 4.

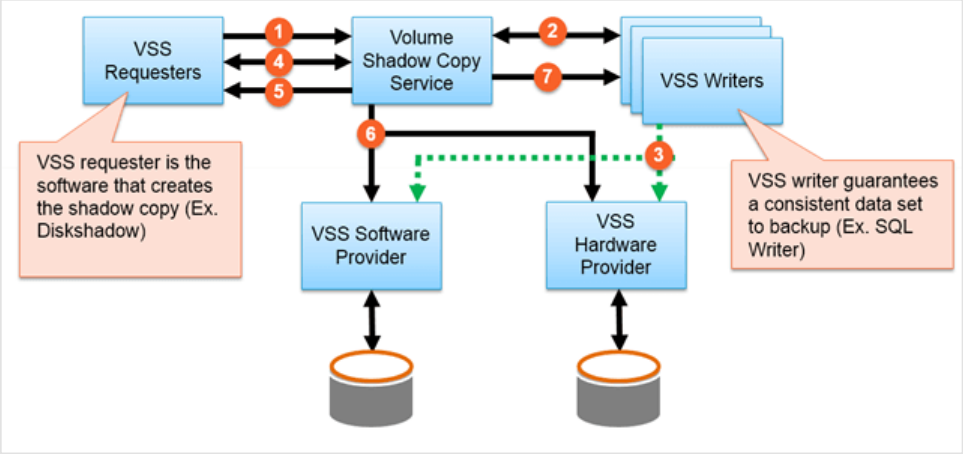


Figure 4. The VSS process.

There is one important thing to note when it comes to snapshots and can be seen in Figure 3: All the databases and drives will be quiesced at the same time. Figure 5 shows what the snapshots look like on a FlashArray.

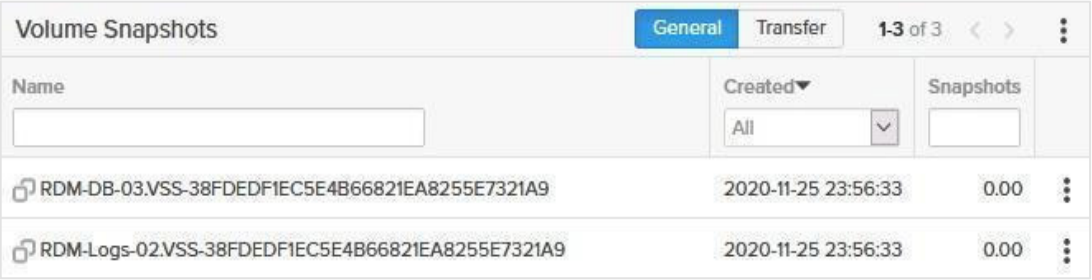


Figure 5. Snapshots in a FlashArray.

The time difference between SQL Server's log and the array is that the array's time is set to one time zone and the server is set to another. You can rename the snapshots to be more friendly as the default names are not as descriptive as most would want to have in real-world scenarios.

A VSS-based backup cannot generate a transaction log backup. To achieve point-in-time restores, transaction log backups must still be generated. Should you need to restore a snapshot, do it with NORECOVERY as you would for a native SQL Server backup and then use the transaction log backups to roll things forward.

Microsoft SQL Server Management Studio Extension

Storage-based snapshots are often hidden from DBAs. With Pure Storage, this does not have to be the case. A Microsoft SQL Server Management Studio (SSMS) extension allows DBAs to create an application, not crash, consistent snapshot backups of Microsoft SQL Server.

Virtualized FCIs that use iSCSI or RDMs for storage are fully supported with the SSMS extension. Ones that use vVols or datastores for shared disks are not. An example configuration for an ESXI-based FCI configuration is shown in Figure 6.

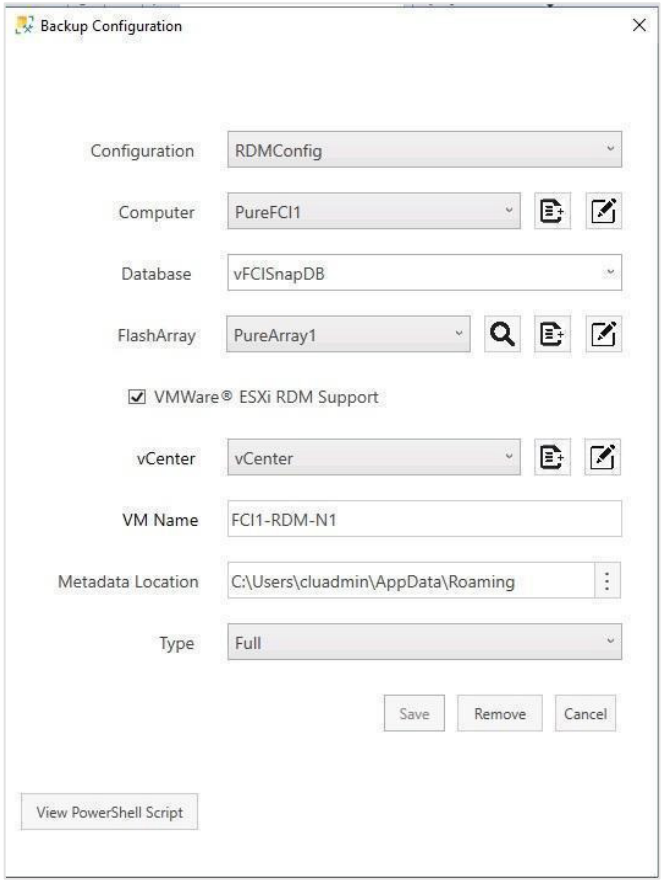


Figure 6. SSMS extension configured for ESXi

The plugin supports point-in-time recovery through NORECOVERY so transaction logs can be applied. The plugin can generate two different variants of a full backup—regular and copy only. As noted above, the application consistent backup generated can be part of a multi-layer backup and recovery strategy.

Protection Groups

Snapshots can be combined with protection groups. A protection group allows you to define a group of related objects (such as volumes at the FlashArray level) and then take point-in-time, consistent snapshots across them. That in turn enables the ability to recover from a disaster if these snapshots are replicated elsewhere and you may even be able to use them to seed a development or test environment on another array.

In a virtual environment, a protection group cannot be defined down to the virtual disk level where the databases would be stored for Microsoft SQL Server, so how you set up your various virtual disks will become important. One thing to note is that protection groups are crash, not application consistent.

Storage-Based Replication

FlashArray has three features that provide replication:

- Asynchronous replication: periodic, active-passive, and asynchronous
- ActiveCluster: continuous, active-active, and synchronous
- ActiveDR: continuous, active-passive, and asynchronous

The key to success with all three is a robust network with minimal latency to ensure that workloads will not be impacted.

FlashArray Asynchronous Replication

A FlashArray can asynchronously replicate to another FlashArray or Pure Cloud Block Store™ which is discussed in an upcoming section. For many, this would be a very good disaster recovery option especially if you have nothing today at any layer to protect your VMs. Replicating in this way is just standard block-level storage replication and is not dependent on VMware. It does directly integrate with vSphere Site Recovery Manager (SRM) but does not with any Microsoft SQL Server-based availability feature. Figure 7 shows a sample topology.

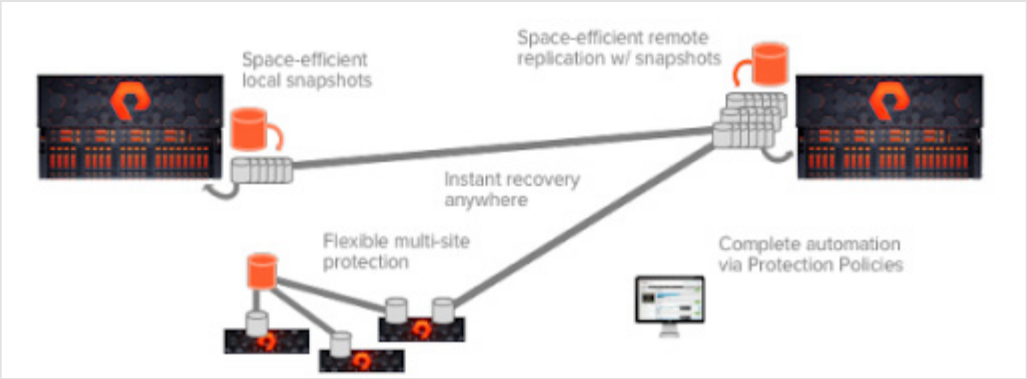


Figure 7. Asynchronous storage-based replication.

Note: To see more information and best practices for this feature, consult the configuration and best practices guide.

ActiveCluster

ActiveCluster is a feature built into a FlashArray that not only allows you to configure synchronous replication between arrays (for example, ones in the same data center assuming very low latency). Both arrays are live and available for use. Pure supports up to 11ms round-trip latency as documented in two different documents: the Pure ActiveCluster with SQL Server

white paper and this support article. For synchronous replication, failover is transparent. This is especially important for mission-critical workloads where if a full storage array failed, things would continue to work without disruption.

One way to implement ActiveCluster with VMware is a vSphere Metro Storage Cluster (vMSC). This is discussed in VMware KB 51656. You can find full ActiveCluster documentation on Pure's support site. As of the writing of this paper, ActiveCluster does not support vVols. Microsoft SQL Server is compatible with ActiveCluster. Read the previously linked white paper for more information.

ActiveDR

ActiveDR is a new feature introduced in Purity 6.0 which provides continuous asynchronous replication with a near-zero RPO between two FlashArrays. The replication for ActiveDR is not based on the use of snapshots. This makes it easier to have two Pure Storage arrays located farther apart to account for latency. One caveat is that the replicated storage is read-only on the target. The virtual disks should be offline for the VM that is replicated but if necessary, can be brought online in a read-only state. The replication can be reversed should the disaster recovery array become the primary array. You can replicate a whole VM or just the storage that contains the VMDKs with Microsoft SQL Server's data and transaction log files.

There are currently some known caveats with ActiveDR, Microsoft SQL Server, and VMware. For full information and any updates, consult the Microsoft SQL Server-specific support article, the VMware-specific for implementing ActiveDR as well as best practices and guidelines for ActiveDR with VMware environments.

Note: As of the writing of this paper, vVols are not yet supported. AGs are not a natural fit for use with ActiveDR

Why would you choose ActiveDR over the asynchronous replication feature in a FlashArray? The main difference between the two is ActiveDR has a lower RPO since it is continuously replicating. ActiveDR also replicates snapshots, policies, volume configuration, tags, and more so it also protects the configuration of the storage, not just the data stored on the array. A properly implemented ActiveDR solution can help with both RTO and RPO.

ActiveDR can also be used to easily help test disaster recovery since it can promote and demote storage for testing without taking production offline. The replicated storage can be "promoted" at a given time to create a test environment of all the replicated data, snapshots, and configurations (QoS, tags, etc.). This promotion does not interrupt replication and can be refreshed from the backend replication for repeated testing. All of this can be done natively at a storage layer or used underneath products like SRM. SRM provides workflows to a disaster recovery plan for VMs and requires some form of storage-based replication to work such as ActiveDR.

From a Microsoft SQL Server perspective, combining ActiveDR with snapshots is not dissimilar to a solution like log shipping. The snapshot, like the delayed application of the transaction log backup on a warm standby, accounts for scenarios such as the "oops" one where someone issues a DELETE without a WHERE clause.

Public Cloud-Based Solutions

If you are using Amazon Web Services (AWS), Pure has solutions that will help you extend your current on-premises investments that could be leveraged for Microsoft SQL Server.

For AWS, there is Pure Cloud Block Store. Pure Cloud Block Store is a software-defined storage solution that leverages the resources of AWS Elastic Compute Cloud (EC2). Combined with FlashArray native replication to Pure Cloud Block Store, the

solution could be one way of achieving disaster recovery to a native EC2 instance—what AWS calls a VM—to use in the event of having to switch if your primary solution is unavailable.

Pure Cloud Block Store for Microsoft Azure is currently in beta, and planned for availability in H2 2021.

In-Guest Availability Using Microsoft SQL Server Features

This section will cover how using Pure Storage can enhance Microsoft SQL Server availability when Always On AGs and Always On FCIs are used.

Always On Availability Groups

An AG provides database-level protection. Each instance of Microsoft SQL Server participating as a replica in an AG has a copy of the data. In a normal scenario, while this increases the storage cost, the copies of the data are completely separate if they are on separate storage. As noted earlier, If an AG is created for local availability with a single FlashArray behind it, this may not be the case. Your VMs should be hosted on not only different ESXi hosts, but the virtual disks for the primary and secondary replicas should be on different FlashArrays. Also keep in mind that because a FlashArray does deduplication, a single FlashArray is suboptimal for an AG used for availability. This avoids the illusion of availability discussed earlier.

Having said that, there is still value in configuring an AG with a single FlashArray. For example, you can use one for configuring readable replicas if that fits your workload. Keep in mind that virtual platforms should have a lot of redundancy underneath the covers and be properly designed. This includes the ESXi hosts with redundant components such as dual storage controllers as well as having redundant storage arrays in their own availability configuration.

Snapshot backups cannot be used directly to seed a secondary replica. However, it is possible to do so via storage-based options. You can see how to do this in this video.

Always On Failover Cluster Instance

An FCI is the traditional form of Microsoft SQL Server that uses an underlying cluster. That cluster is either a Windows Server Failover Cluster (WSFC) for Windows Server-based deployments or Linux, Pacemaker. FCIs require some form of shared storage. Presenting shared storage is different on Linux and Windows Server.

Shared Storage

This section will cover how Pure Storage solutions can present shared storage to virtualized FCI configurations with vSphere.

Linux: For Linux FCIs, Microsoft supports iSCSI, Server Message Block (SMB), and Network File System (NFS). This means that there is no native way from a VMware perspective to present shared storage; it would be configured only via in-guest methods. Pure can present storage to a Linux-based FCI only if it uses iSCSI. An iSCSI target is built into a FlashArray so you can leverage that natively. Figure 8 shows what iSCSI looks like on a FlashArray.

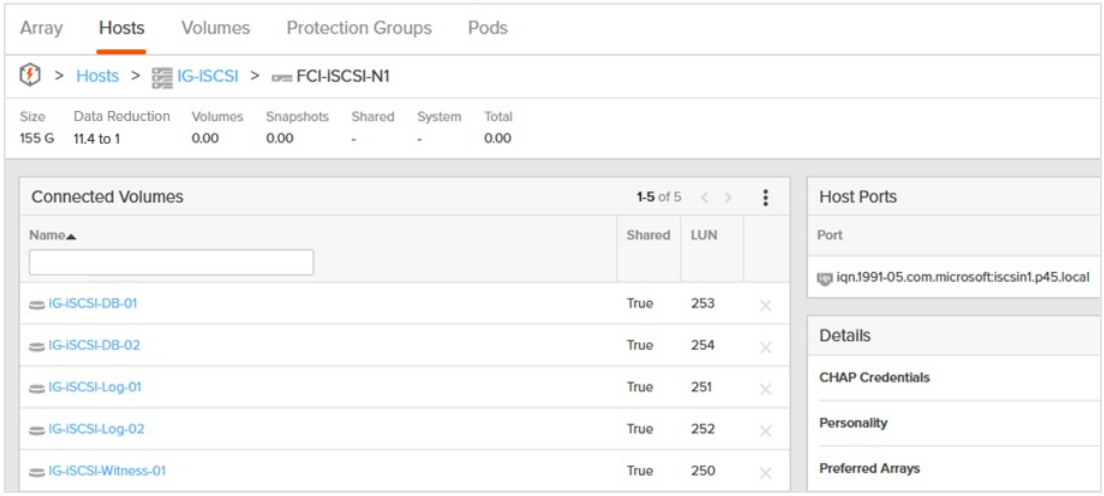


Figure 8. Example iSCSI configuration as seen in a FlashArray.

While Microsoft SQL Server supports SMB 3.0 or later, FlashArray and FlashBlade are not supported for data and transaction log files using SMB natively on the array. Similarly, Microsoft requires version 4.2 or later for use with Linux-based FCIs. At the time of writing this paper, Pure only supports NFS versions 3.x and 4, so using a FlashArray for NFS supporting shared storage for FCIs is not possible.

Windows Server: There are two ways to approach shared storage for virtualized FCIs with Windows Server: in-guest options and using ESXi to present shared storage natively to the VM. It is more common to see people use native options in ESXi versus in-guest.

Before going any further, if deploying an FCI under ESXi, always refer to either VMware KB 2147661 "Microsoft Windows Server Failover Clustering (WSFC) with shared disks on VMware vSphere 6.x: Guidelines for supported configurations" or VMware KB 79616 "Microsoft Windows Server Failover Clustering (WSFC) with shared disks on VMware vSphere 7.x: Guidelines for supported configurations" depending on your version of ESXi. These are the definitive support statements from VMware around shared storage for WSFCs under ESXi. While VMware generally does not certify specific versions of Windows Server (and Microsoft SQL Server), they do when it comes to shared storage. For example, if you want to deploy an FCI with Windows Server 2019 as the underlying operating system, if using ESXi 6.x, the hosts must be at either 6.5 Update 3 or 6.7 Update 3.

For disks to be "seen" properly by a WSFC for shared storage, they must be compliant with SCSI-3 persistent reservations. Pre-ESXi 6.7, this was not possible with VMDKs. The only way to support FCIs natively was to use RDMs in physical mode. A FlashArray can present RDMs to a VM. Figure 9 shows the LUNs with the same numbers at the storage layer.

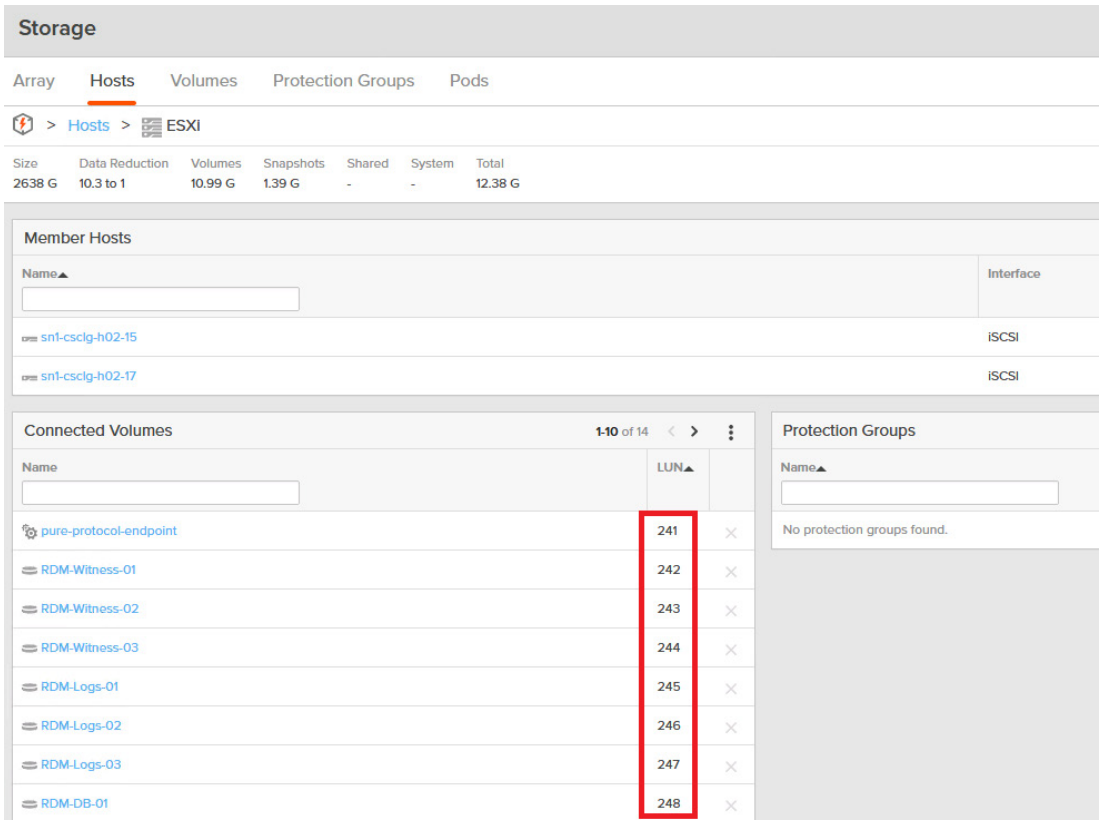


Figure 9. LUNs created on a FlashArray for use as RDMs

When adding an RDM to a VM, note that the LUN number in vCenter's Select Target LUN dialog matches the ones highlighted in Figure 10 shows what things look like in vCenter. Note that while this example shows matching LUN numbers to demonstrate the concept of tying RDMs back to the storage, a better way in a production environment would be to use serial numbers since they are persistent and globally unique; a LUN number may change or be the same if different arrays are involved with the same host, or different across hosts for the same volume.

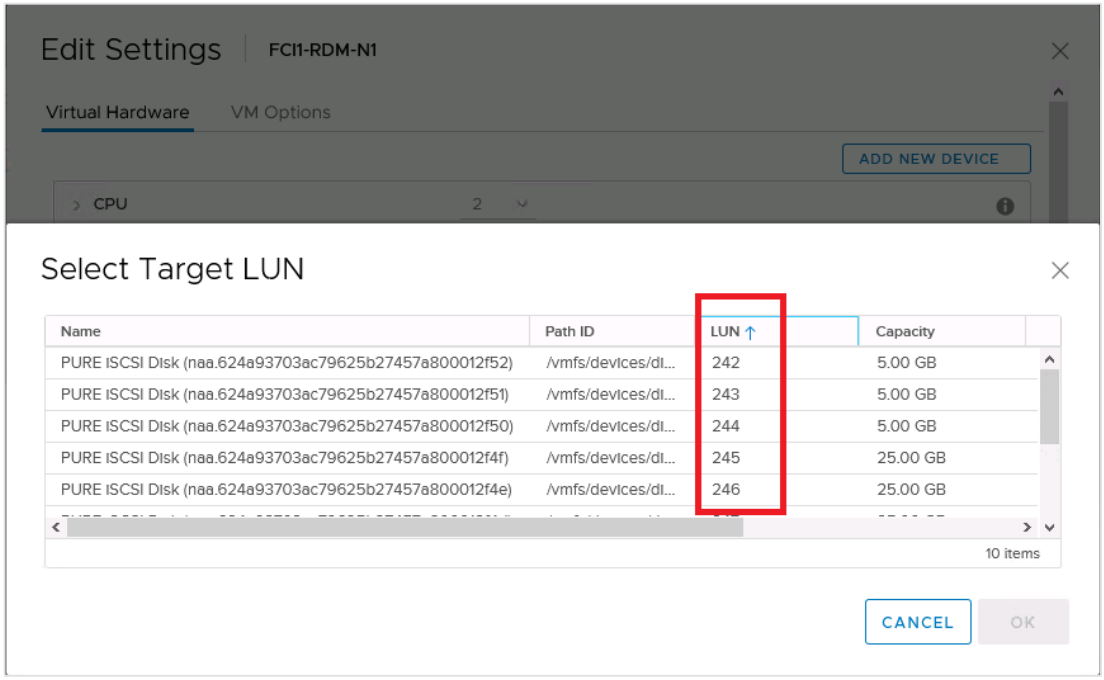


Figure 10. Adding an RDM to a VM.

Starting with ESXi 6.7, using VMDKs for shared storage is possible with vVols. Because a FlashArray supports VASA as noted earlier, you can take advantage of this newer feature in vSphere.natively with a FlashArray. In vSphere 7.0, this capability was expanded to be able to use traditional VMFS datastores for shared disks using standard virtual disks if the storage use either Fibre Channel for connectivity to the host. Depending on your configuration and/or version of vSphere, you can create FCIs by using a standard virtual disk for shared storage.

While less common, in-guest options for presenting shared storage to a Windows Server-based VM are iSCSI or SMB 3.0+ just like Linux. In addition, Windows Server-based FCIs can also place TempDB on a local drive; it technically does not need to be on shared storage which means it can be placed within a standard virtual disk. A word of caution: do not use this TempDB configuration without testing to ensure it meets your needs. If you are using iSCSI or RDMs, both those storage options are fully supported by a FlashArray natively.

Standard best practices for in-guest options apply such as using a separate network for iSCSI traffic which also means an additional vNIC. For iSCSI, using the Microsoft iSCSI initiator is fully supported under ESXi as per VMware KB 1010547. One small caveat: if you are using iSCSI and reboot a storage controller, there are rare occasions where FCIs can fail due to storage timeouts. Pure has documentation covering this topic.

> NOTE: To be supported by Microsoft, a WSFC must pass validation. For FCIs, this means with shared storage, any disks presented to the proposed nodes of the WSFC must be tested and pass. Should any problems be detected during validation, they must be addressed before creating the WSFC and subsequently the FCI. You do not want problems in the event you need to fail to another node. Read the output of validation carefully as it will guide you as to what is wrong. For example, if storage is only presented to one server instead of all possible nodes.

Geographically Dispersed Failover Cluster Instances

Another way that a Pure Storage array can give you increased availability is through its storage replication technologies, creating a geographically dispersed FCI configuration. A geographically-dispersed FCI generally does not use any in-guest

method to synchronize (unless you are using Storage Replica which is a built-in WSFC feature). All synchronization happens below Microsoft SQL Server and should be transparent to what is going on inside the VM. All nodes participating in the FCI are part of the same WSFC in a traditional geographically dispersed FCI.

This means that you would be using some form of storage-based replication to ensure the other array(s) have the disks needed in the event of a failover. This would not be ActiveCluster since a geographically dispersed FCI spans locations. The storage would need to be live on the other storage array since the nodes of the WSFC are live. Replication across longer distances is usually asynchronous, so ActiveDR and the replication feature of a FlashArray should work in this case.

Another way to stretch an FCI across multiple locations outside of any native features of Microsoft SQL Server like log shipping or an AG is to use native disk-based replication. At the destination, you would have VMs that are not live. Use storage-based replication such as ActiveDR to replicate the disks. When it comes time to bring the VMs online (assuming a failure or other condition to trigger such an event), present the storage to the VMs. This is not a traditional geographically dispersed FCI.

Now that you understand how Pure arrays natively can be used for availability with in-guest Microsoft SQL Server features there is one piece of the puzzle left: vSphere itself.

Virtual Machine Availability Using vSphere Features

vSphere has a few features to enhance the availability of VMs both in a local high availability capacity as well as for disaster recovery. The usual suspects of VMware HA, vMotion, and Fault Tolerance (FT) all can be implemented with no special considerations on the Pure side. There are some Microsoft SQL Server considerations such as FT not supporting AG or FCI configurations. Read VMware documentation for more on those features.

The intersection of Pure, VMware, and Microsoft SQL Server comes into play with features like vCenter Site Recovery Manager (SRM). Underneath SRM there is a requirement to mirror storage elsewhere. The following replication technologies on the Pure Storage FlashArray are supported with SRM:

- Periodic Asynchronous Replication (Protection Group-based)
- ActiveCluster
- ActiveDR
- ActiveCluster + Periodic Asynchronous Replication (failover to a third site)

Choosing an Availability Strategy

Many different options were presented above. Which option or options should you use to make Microsoft SQL Server deployments more available with the combination of Pure Storage, vSphere, and what SQL Server brings to the table? The answer is "it depends." Any solution must be based on requirements. All solutions should be:

- Manageable (including ease of implementation, day-to-day management, as well as failing over)
- Supported by the vendors
- Reliable
- Able to meet the RPO/RTO requirements of the business

The reality of any availability solution—whether it is for local high availability or disaster recovery—is that some systems and applications are more critical to a company and those will need the best protection. Others may be used infrequently and have different needs. Many implement more than one solution but standardize on a small number to keep the surface area down.

A best-of-breed solution often involves multiple layers. For example, you can use FCIs for local availability with Pure-based, application-consistent snapshots for full backups and then using either SRM or one of the AG variants for disaster recovery.

That may work for the big systems, but what about VMs that have no in-guest availability solution? That is where platform- and storage-based options may give you the availability that is needed. Even doing something as simple as implementing ActiveCluster or ActiveDR protects the virtual disks associated with the VM running Microsoft SQL Server inside.

There is no "one size fits all." Use what is best and makes the most sense in your environment. One way to think about the different layers is as follows:

- **Native Microsoft SQL Server availability features:** Designed to protect instances and databases and has the best integration. The features account for transactions which means possibly lower RPO in some scenarios but may not play as "nicely" with the platforms underneath. This gives the DBA the most control, but visibility to other administrators is more limited outside of resource consumption.
- **vSphere-based features:** Designed to protect the VM, but usually not aware of what is going on inside a VM. When you need transactional consistency, not always the best option for Microsoft SQL Server and RPO. Often compliments and integrates better or even very well with hardware-based options.
- **Pure-based features:** Compliment existing availability and backup strategies in-guest or in vSphere with storage-level features and redundancy that are largely transparent to Microsoft SQL Server or use array-based features such as ActiveCluster to provide business continuity across metro distances. Can provide good RTO and RPO with the right strategy. The very nature of the FlashArray provides high availability, but that can be enhanced with FlashArray replication technologies to protect over greater difference with minimal to no disruption.

Summary

Pure Storage's solutions are not only designed to work well with Microsoft SQL Server deployments from a performance perspective, but they can also be employed to add resiliency and availability beyond what VMware and Microsoft provide. As noted above, there are some considerations to take into account when devising availability solutions that include Microsoft SQL Server. The best strategies are multi-layered and have a robust backup and recovery plan. Whether you choose in-guest availability options, ones at the platform layer in vSphere, at the storage layer with FlashArray or FlashBlade, or a combination of them, you should be able to meet your company's RTOs and RPOs.

Resources

VMware White Papers

- Architecting Microsoft SQL Server on VMware vSphere
- Planning Highly Available, Mission Critical SQL Server Deployments with VMware vSphere

Pure Storage Support Content

- Best Practices: Microsoft SQL Server
- Microsoft Platform Guide
- VMware Platform Guide and Site Recovery Manager Integration

  Unless noted where there may be a difference, always follow published VMware guidelines.

About the Author

Allan Hirt, managing partner and founder of SQLHA, LLC, is a consultant, trainer, author, and business continuity expert. He has been working with SQL Server since 1992 when it was still a Sybase product, clustering in Windows Server since the late 1990s when it was called Wolfpack, and VMware since the early 2000s. Currently a dual Microsoft MVP (Data Platform; Cloud and Datacenter Management) as well as a VMware vExpert, Allan helps customers across the globe, delivers training, and is a frequent speaker.

Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041