PURESTORAGE®

WHITE PAPER

FlashArray ActiveCluster for Microsoft SQL Server

A solution for high availability, read scalability, and nondisruptive storage-level migration for Microsoft SQL Server databases.

Contents

Executive Summary

Ensuring high availability and seamless migration is critical for Microsoft SQL Server environments, where any downtime can lead to significant business disruptions. Organizations require storage solutions that ensure data consistency, minimize outages, and provide smooth failover during both planned and unplanned events.

Pure Storage® FlashArray™ ActiveCluster™ offers key features designed to meet these needs. This solution provides active-active replication across multiple sites, synchronous data replication, and automatic failover. These capabilities ensure continuous availability and nondisruptive database migrations for SQL Server workloads.

The primary benefits of ActiveCluster include zero recovery time objectives and zero recovery point objectives, ease of configuration, and the ability to maintain high availability without complex infrastructure. The simplicity and automation of ActiveCluster help reduce operational overhead for administrators while ensuring SQL Server databases remain available and performant during failures or migrations.

ActiveCluster delivers a powerful, scalable solution for SQL Server environments that demand high availability, reliability, and ease of management, providing peace of mind concerning critical business applications.

How to Use This Document

This document serves as a comprehensive resource designed to offer insights into the effective utilization of FlashArray and the ActiveCluster feature. It highlights the capabilities of ActiveCluster in providing redundancy and high availability at the storage layer for SQL Server databases, ultimately minimizing downtime and ensuring uninterrupted operations. Throughout this document, there is detailed guidance and best practices to help achieve a robust and resilient storage environment for SQL Server databases.

This document is intended for storage, SQL Server database, and system administrators who plan to implement FlashArray and ActiveCluster for a range of deployment scenarios including unclustered SQL Server databases and SQL Server Failover Cluster Instances.

Solution Overview

ActiveCluster provides a powerful solution for SQL Server workloads by enabling synchronous replication across FlashArray instances, ensuring identical volume identities across arrays. This capability supports use cases such as high availability for mission-critical databases and seamless nondisruptive storage migration. SQL Server hosts can be zoned to ActiveCluster on FlashArray or Pure Storage Cloud Dedicated block storage, ensuring uninterrupted access to databases while maximizing data resilience and operational efficiency.

This paper covers a common SQL Server environment consisting of multiple interconnected components, which include the following:

- SQL Server instances running on Windows Server or a supported Linux platform
- 2 x FlashArrays used for primary storage protected by ActiveCluster
- Pure Storage Cloud Dedicated for primary storage of SQL Server instances in Microsoft Azure or Amazon Web Services

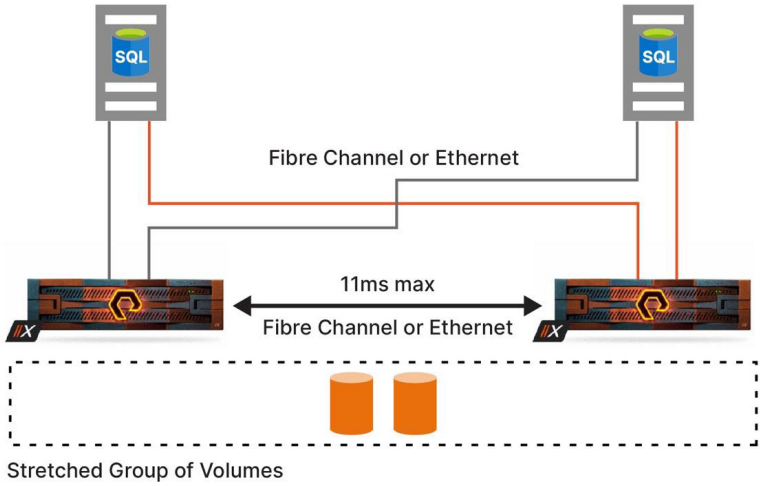


FIGURE 1 High-level overview of the ActiveCluster for SQL Server solution

Note: Pure Storage does not recommend setting up ActiveCluster between Pure Storage Cloud Dedicated and an on-premises FlashArray environment. Pure Storage cannot guarantee that synchronous data replication between on-premises storage and the cloud will work, or that such an arrangement can provide acceptable latency between sites.

Solution Benefits

ActiveCluster offers a range of critical benefits tailored to storage and SQL Server administrators. These features help ensure continuous database availability, simplify management, and reduce operational overhead, making ActiveCluster an ideal solution for both small and medium-sized businesses and enterprise IT environments.

- **Synchronous replication for data integrity:** SQL Server data is synchronously replicated with ActiveCluster across two FlashArray systems, ensuring data is written to non-volatile RAM on both arrays before acknowledging the write to the host. This helps ensure data consistency and reliability, which are key concerns for database administrators managing mission-critical workloads.
- **Active-active architecture for optimized access:** With a symmetric active-active configuration in ActiveCluster, SQL Server hosts can read and write to volumes on either array, improving data accessibility and enabling seamless operations across arrays. The ability to manage host-to-array site preferences also provides SQL Server administrators with greater control over data flow, helping ensure optimal performance.
- **Transparent failover for uninterrupted operations:** ActiveCluster supports automatic, transparent failover between arrays without disrupting SQL Server operations. Administrators can specify failover preferences, and the system will autonomously manage the process, ensuring SQL Server databases remain available even during array or site failures.
- **Simplified storage management:** SQL Server database administrators benefit from simplified management with the unified ActiveCluster interface. Provisioning storage, creating snapshots, and managing replication are all streamlined, reducing the complexity of maintaining a highly available database environment.
- **No additional licensing costs:** ActiveCluster requires no additional hardware or software licenses, allowing organizations to maximize their investments in FlashArray systems.
- **Integrated cloud-based failover mediation:** The integrated Pure1® Cloud Mediator solution automatically manages failovers between arrays, preventing split-brain scenarios and eliminating the need for additional on-premises components. This ensures SQL Server environments maintain high availability without additional infrastructure overhead.
- **Support for nondisruptive storage migration:** ActiveCluster also supports nondisruptive migration of SQL Server databases across arrays, ensuring that storage upgrades or site changes can occur without impacting database performance or availability. This is particularly beneficial for administrators managing dynamic environments with evolving storage needs.

By addressing the core challenges of SQL Server high availability, ActiveCluster helps storage and database administrators maintain business continuity, reduce downtime, and improve operational efficiency—all without the need for complex infrastructure.

Limitations of ActiveCluster

While ActiveCluster offers robust high availability and synchronous replication for SQL Server environments, there are specific limitations to consider for optimal deployment. These include proximity requirements for latency, architectural design considerations, and dependencies on additional components to enable automatic failover.

Distance Limitations for Optimal Performance

The ActiveCluster synchronous replication capability is highly effective for SQL Server environments requiring high availability across storage arrays. However, to ensure optimal performance, it is recommended to maintain latency below five milliseconds, which is typically achievable between data centers located within metro proximity of one another. Distances exceeding this threshold might introduce latency that could impact SQL Server application performance, making ActiveCluster best suited for geographically close data centers.

Design for Two-site Configurations

The ActiveCluster architecture supports active-active replication between two FlashArray systems, facilitating seamless data accessibility and redundancy across a dual-site setup. However, it is designed for a two-site configuration and does not support additional arrays within the same synchronous setup. For SQL Server environments requiring multisite disaster recovery, other solutions such as ActiveDR™ or asynchronous replication can be used.

Requirement for a Mediator or Witness for Automatic Failover

To enable automatic failover capabilities, ActiveCluster requires the use of a mediator or witness component, such as the Pure1 Cloud Mediator. This cloud-based solution, or an on-premises mediator, ensures SQL Server environments maintain high availability by managing failover autonomously. Without this component, manual intervention is needed in the event of an array failure, which can impact recovery times in critical scenarios.

Technology Overview

Pure Storage FlashArray ActiveCluster offers a unified storage solution that seamlessly integrates high availability, data replication, and failover capabilities. Built on the Purity operating environment, ActiveCluster provides SQL Server administrators with the ability to ensure data consistency and minimize downtime across geographically dispersed arrays.

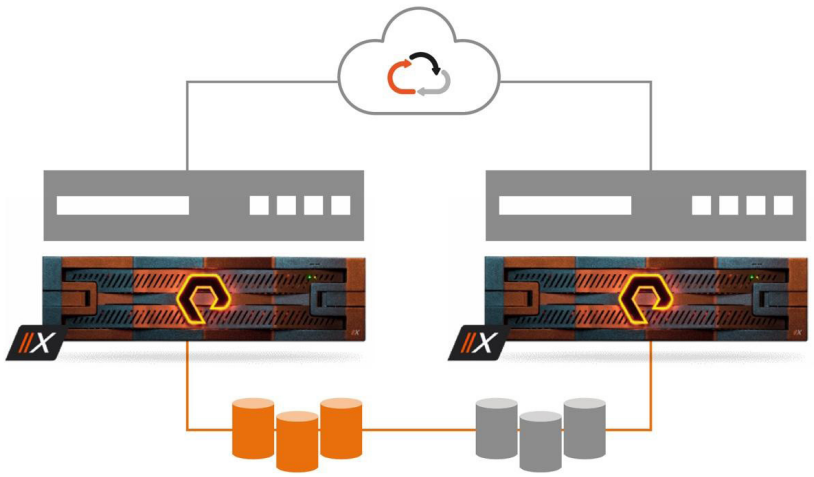


FIGURE 2 Technology overview of the ActiveCluster for SQL Server solution: Pure1 Cloud Mediator mediating between FlashArray systems to provide high availability for SQL Server data replicated between two sites

SQL Server

SQL Server is a widely used relational database management system that has gained popularity among organizations of all sizes due to its scalability and ease of management. SQL Server provides high availability features that database administrators can use, such as SQL Server Always On Failover Cluster Instances, and Always On Availability Groups.

SQL Server also provides a number of features that make it an ideal platform for multiple mixed workloads, from online transaction processing to complex online analytics processing. SQL Server handles structured data with atomicity, consistency, isolation, and durability compliance that protects data and ensures reliable transaction processing. Data protection features, such as transparent data encryption and role-based security, help organizations keep their data safe both at rest and in transit. Within this white paper, SQL Server is the core database platform that supports an organization's business applications and services.

SQL Server High Availability Solutions

SQL Server environments rely on high availability solutions to minimize downtime and ensure continuous access to critical data. Two primary SQL Server configurations for high availability are SQL Server Failover Cluster Instances and Always On Availability Groups, each providing unique benefits for maintaining SQL Server accessibility.

SQL Server Failover Cluster Instances and ActiveCluster

SQL Server Failover Cluster Instances use the Windows Server Failover Cluster service to provide instance-level redundancy and automatic failover within a cluster. In a SQL Server Failover Cluster Instance setup, SQL Server instances are configured with a shared storage model that connects multiple nodes, allowing operations to transfer automatically to a standby node if the primary node encounters an issue. This model reduces downtime and ensures SQL Server databases remain accessible during node failures.

ActiveCluster adds a powerful layer of storage-level high availability by addressing the singular fault domain inherent in the shared storage model used by SQL Server Failover Cluster Instances. By introducing a second FlashArray with synchronous replication, ActiveCluster enables identical copies of SQL Server database volumes to exist on separate arrays, thus mitigating the risks associated with a single storage fault domain. Storage-level redundancy and synchronous data replication with ActiveCluster ensure data consistency and seamless failover across FlashArray instances without data loss or disruption to SQL Server operations. This approach minimizes both planned and unplanned downtime, providing a robust solution for mission-critical SQL Server workloads.

Before implementing SQL Server Failover Cluster Instances with Windows Server Failover Cluster, review the following setup and installation documentation:

- Before Installing Failover Clustering
- Create a New Always On Failover Cluster Instance (Setup)
- SQL Server Failover Cluster Installation

Enhancing High Availability with ActiveCluster

With ActiveCluster, data is synchronously replicated across FlashArray instances, maintaining data consistency across sites and providing an additional layer of availability for SQL Server. This configuration allows for seamless failover and recovery in the event of a failure, ensuring business continuity without data loss. By integrating at the storage layer, ActiveCluster not only enhances redundancy but also simplifies storage management for SQL Server Failover Cluster Instances, making it ideal for reducing downtime and maintaining high availability in SQL Server environments.

Always On Availability Groups

Always On Availability Groups provide database-level high availability and disaster recovery by maintaining replicas of user databases across multiple servers. Unlike SQL Server Failover Cluster Instances, Always On Availability Groups allow SQL Server instances in an Availability Group to be active on different replicas simultaneously, enhancing read scalability and ensuring continuous availability. However, only one replica (the primary) can be configured as read/write; additional replicas remain read-only.

Note: Always On Availability Groups and ActiveCluster serve separate purposes and should be implemented independently. Choose the appropriate solution based on business requirements: Always On Availability Groups are managed at the database level, while ActiveCluster operates at the storage level.

SQL Server Always On Failover Cluster Instances

An Always On Failover Cluster Instance follows a shared storage model, where the SQL Server instance is installed on multiple hosts, but it runs on only one host at a time (in an active-passive configuration). The SQL Server instance is hosted on a primary active host, while the other hosts in the cluster remain passive. If the primary active host experiences a failure or goes offline, the Windows Server Failover Cluster service initiates a failover process, promoting one of the passive hosts to the primary active role. SQL Server then resumes operations on the newly promoted host, ensuring minimal downtime and high availability. To further enhance high availability, ActiveCluster introduces transparent failover, ensuring automatic, seamless recovery.

FlashArray

Built on all-flash storage, FlashArray provides storage and database administrators running SQL Server in their environments a fast, scalable, unified block- and file-storage platform that is ideal for high-performance databases.

By providing a unified interface and simple-to-use tools for storage administrators, FlashArray gives those administrators the ability to quickly and seamlessly replicate, move, and manage data. FlashArray also deduplicates and compresses all data before it is written, efficiently reducing the size of data without impacting performance. Storage and database administrators can further increase storage by using the FlashArray snapshot capabilities to create snapshots of production databases, and they can use those snapshots in development or testing environments.

The FlashArray family consists of the following:

- **FlashArray//C™:** low-latency storage for capacity-oriented workloads
- **FlashArray//X™:** high-performance, high-capacity storage that is ideal for performance-oriented workloads
- **FlashArray//XL™:** high-performance storage at scale that helps reduce the number of arrays needed to run large applications
- **FlashArray//E™:** economical-at-scale storage for workloads that aren't time-sensitive

ActiveCluster

ActiveCluster is an advanced solution for data replication, offering a fully symmetrical, bidirectional replication capability that ensures zero recovery point objectives and zero recovery time objectives. It achieves this by replicating data synchronously and enabling automatic, seamless failover. Its design enables clustered arrays and hosts to operate in a flexible, active-active configuration within a data center setup, maintaining high availability and reliability across connected sites.

FIGURE 3 Pure1 and ActiveCluster architecture: Pure1 mediates between clustered FlashArray systems to provide high availability for SQL Server

ActiveCluster stands on three pillars: Pure1 Cloud Mediator, paired active-active clustered arrays, and pods (stretched storage containers).

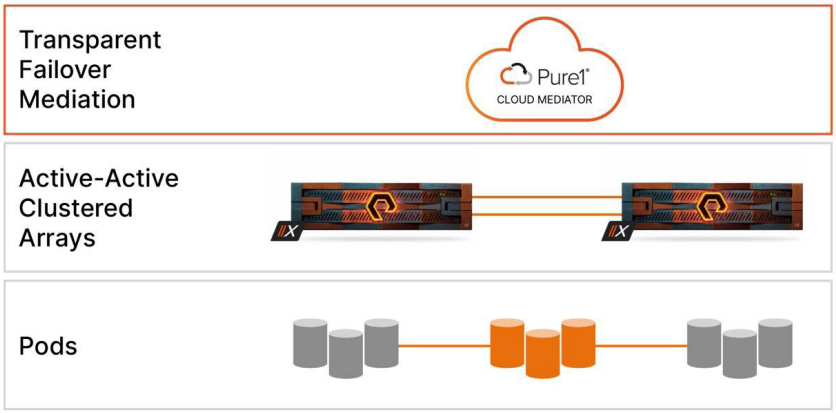


FIGURE 4 ActiveCluster components

Pure1 Cloud Mediator ensures seamless communication, while the paired arrays provide the infrastructure for bidirectional data access:

- **Pure1 Cloud Mediator:** An essential component within the solution, Pure1 Cloud Mediator plays a critical role in determining which storage array will assume data services in the event of an outage, ensuring the continuity of operations within the environment.
- **Active-active clustered FlashArray systems:** This configuration leverages synchronous replication to uphold identical data copies on each array and present them as a unified and consistent copy to hosts connected to either or both arrays. This approach ensures data accessibility and consistency across the clustered arrays.
- **Stretched pods:** ActiveCluster introduces a new management entity known as "pods." A pod delineates a group of objects that undergo synchronous replication together and specifies between which arrays this replication occurs. Pods not only facilitate the grouping of storage objects but also ensure a consistent input/output continuation behavior for the storage objects contained within them, enhancing data availability and reliability.

Pure1 Cloud Mediator

A mediator is a critical component for managing and coordinating data replication and failover between two FlashArray systems in an active-active configuration. ActiveCluster ensures synchronous replication across FlashArray systems. Pure1 Cloud Mediator ensures that if communication is lost between arrays, only one array remains active for each pod, preventing conflicts.

If the arrays lose connectivity, both temporarily halt input/output and contact the mediator. The array that reaches the mediator first continues serving its pods, while the other stops input/output to prevent a "split-brain" scenario. This process occurs within standard host input/output timeouts, ensuring applications experience only a brief pause before resuming operations, minimizing disruptions to SQL Server databases.

On-premises Failover Mediator

Failover mediation for ActiveCluster can also be facilitated by deploying an on-premises mediator, which is distributed as an Open Virtualization Format file and can be set up as a virtual machine. The failover procedures remain consistent with those previously described, regardless of whether Pure1 Cloud Mediator or an on-premises mediator is employed. More information can be found in the on-premises mediator documentation.

An on-premises mediator must adhere to certain fundamental requirements:

- **Virtual machine deployment**: The on-premises mediator can only be deployed as a virtual machine on virtualized hardware; it is not installable as a standalone application.
- **High availability**: To ensure high availability, the mediator's host should implement a reliable failover mechanism, such as VMware high availability or Microsoft Hyper-V high availability clustering.
- **Immutable configuration**: The array hosting the mediator must be configured to prevent any rollback of the mediator's configuration to prior versions. This includes scenarios like storage snapshot restores or situations where the mediator resides on mirrored storage.
- **Configuration setting**: If using an on-premises mediator, the storage arrays must be configured to utilize the on-premises mediator instead of Pure1 Cloud Mediator.
- **Third-site deployment**: The mediator should be deployed in a third site, within a separate failure domain that remains unaffected by failures in either of the sites where the arrays are installed.
- **Independent network connectivity**: Both array sites must have distinct network connections to the mediator. This ensures that the failure of one network connection does not impede access to the mediator for both arrays, enhancing redundancy and reliability.

Stretched Pods

An array can accommodate multiple pods, and those pods can exist on a single array or span across two arrays concurrently, maintaining synchronous replication. When pods are synchronously replicated between two arrays, they are referred to as being "stretched between arrays." This innovative feature enhances the flexibility and management capabilities of ActiveCluster.

- **Pod consistency groups:** Pods can contain various components, including volumes, protection groups, and configuration details specifying the association between volumes and hosts on FlashArray. Essentially, a pod functions as a consistency group, guaranteeing that multiple volumes housed within the same pod maintain strict consistency in terms of the order in which write operations are executed. This ensures data integrity and coherence for the interconnected components within the pod.
- **Pod namespaces:** Pods also introduce the concept of volume namespaces, which means that different volumes can share the same volume name as long as they are located in different pods. In the scenario depicted in Figure 5, the volumes within Pod 3 and Pod 4 are distinct from those in Pod 1 and Pod 2, which are stretched active-active pods. This architectural choice enables the migration of workloads between arrays or the consolidation of workloads from multiple arrays onto a single one, without encountering conflicts related to volume names. This setup enhances flexibility and simplifies data management in diverse operational scenarios.

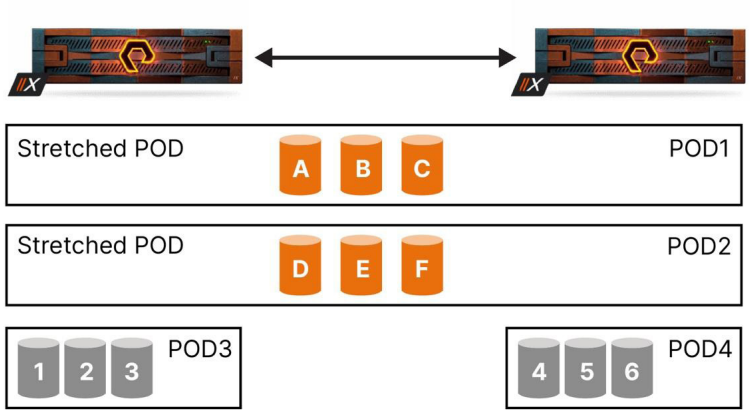


FIGURE 5 ActiveCluster pods

Transparent Failover

With ActiveCluster, failover is transparent, automatic, and requires no manual intervention from administrators. This ensures high availability and prevents data mismatches between arrays serving the same volume.

Note: ActiveCluster failover can also be manually managed for failover testing and nondisruptive storage array migration.

To maintain availability across two sites, a component called a witness (also known as a voter) is needed to manage failovers and prevent data mismatches. ActiveCluster uses Pure1 Cloud Mediator (or an on-premises failover mediator) for this purpose, making failover and site changes seamless and automatic in the event of issues, without the need for manual actions. In addition to transparent failover, ActiveCluster offers flexible deployment options for host access to optimize performance.

Flexible Deployment Options for Host Access

In ActiveCluster, hosts can be configured in two ways: uniform host access, where hosts can access both arrays, regardless of their physical locations, and non-uniform host access, where hosts interact solely with the local storage array.

Uniform Host Access

A standardized storage access model is applicable within environments featuring host-to-array connectivity via either Fibre Channel or Ethernet (for iSCSI), along with array-to-array Ethernet or Fibre Channel interconnectivity across two separate sites. In this deployment scenario, each host possesses access to identical volumes through both the local and remote arrays. This solution is adept at syncing the connection of arrays even when the round-trip time latency between them reaches up to 11 milliseconds, ensuring seamless and consistent data access and management.

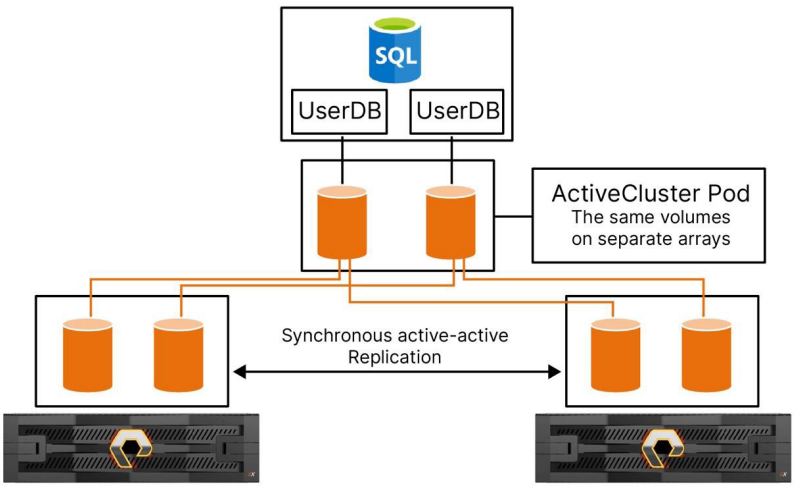


FIGURE 6 ActiveCluster uniform host access

Figure 6 provides a representation of the logical pathways connecting hosts and arrays, including the replication link connecting the two arrays, within the context of a uniform access model. In a uniform storage access model, where all hosts, regardless of their site locations, can access both arrays, it's important to note that different latency characteristics will be observed. Specifically, paths from hosts to the local array will exhibit lower latency, while paths from each local host to the remote array will demonstrate higher latency due to the geographical separation between sites. This diverse latency profile should be considered when optimizing data access and ensuring efficient data management across the environment.

For instance, if a database called "SQL Server B" writes to stretched volume A through Array A, the process might take longer compared to writing through Array B. This is because the data-write from SQL Server B has to travel the longer path between the host and Array A and then cross the replication path between Arrays A and B. It's important to be aware of these latency differences when deciding how to route data for better performance.

SCSI Asymmetric Logical Unit Access

ActiveCluster uses a feature called SCSI Asymmetric Logical Unit Access to encourage hosts to use local paths when accessing FlashArray volumes. Array administrators can select a preferred array for a host-volume connection, making it the best choice for local hosts. Remote paths are available but might not be as fast. While both local and remote paths can be used for reading and writing, hosts prefer the fast local paths and only use the slower remote paths when there are no fast ones available. This helps data access stay quick and efficient.

For example, consider Array A as local to Host A. Setting the preferred array to Array A is important when there is a big difference in latency between a host and its local array compared to the remote array. In cases where data center or small campus setups have similar latencies for host-array communication, utilizing all paths can enhance overall performance.

Non-uniform Host Access

ActiveCluster non-uniform host access is a deployment model where hosts connect to only one of the two FlashArray systems in an ActiveCluster setup, rather than both. Data remains synchronously replicated between arrays, but each host accesses just one array at a time. If one FlashArray becomes unavailable, hosts connected to it will need to reconnect or switch to the secondary array through either manual intervention or automated processes. This model reduces complexity and cabling, providing high availability and zero recovery point objectives without requiring direct host connections to both arrays, though it might require additional steps for failover.

Note: Non-uniform ActiveCluster configurations, when combined with SQL Server Always On Failover Cluster Instances, are unsupported and do not provide a reliable high availability posture. ActiveCluster and Windows Server clustering use different host access technologies that cannot coordinate with each other, which results in uncoordinated failover mechanisms between storage and application layers.

Replication Performance and Data Flow with ActiveCluster

Understanding the performance requirements and flow of data within ActiveCluster is crucial for optimizing SQL Server workloads. Synchronous replication with ActiveCluster ensures zero recovery point objectives by mirroring writes across FlashArray instances, providing robust data consistency and immediate failover capabilities. However, this mirrored write process can have an impact on SQL Server workload performance, especially in latency-sensitive environments.

Mirrored Writes and SQL Server Workload Impact

In ActiveCluster, each write from the SQL Server host is mirrored to both FlashArray systems, ensuring identical data is maintained across arrays. This process, known as "mirrored writes," involves writing data to the primary FlashArray and immediately replicating it to the secondary array before acknowledging the write back to the host. While this approach maintains high data consistency, it also means that replication latency can impact SQL Server response times. Monitoring and managing network latency between arrays is therefore essential, as elevated latency can slow SQL Server transactions that rely on synchronous acknowledgment from both arrays.

Requirements for Optimal Replication Performance

To maximize replication performance, keep the latency between the two FlashArray systems as low as possible, ideally below five milliseconds, and not to exceed 11 milliseconds. This requires a robust network infrastructure and consistent monitoring of latency and bandwidth to ensure reliable and high-speed data transfer. Ensuring that sufficient bandwidth is available for peak workloads can prevent potential bottlenecks and performance degradation in ActiveCluster.

Monitoring Replication Health

ActiveCluster offers built-in monitoring tools that help track replication status and performance. Regularly checking metrics such as replication lag, write latency, and array health is key to identifying potential issues before they affect SQL Server workloads. The Pure1 management platform provides a centralized view of these metrics, allowing administrators to proactively manage the replication environment and maintain optimal performance.

Data Flow of a Write Operation

In an ActiveCluster environment, the flow of a write operation proceeds as follows:

1. The SQL Server application initiates a write to the database, which is processed by the primary FlashArray.
2. The primary array writes the data to its local storage and simultaneously sends the data to the secondary array.
3. Once both arrays confirm the data has been successfully written to non-volatile RAM, the primary array acknowledges the write operation back to the SQL Server host.
4. The host application receives confirmation of the completed write, maintaining data availability and consistency across arrays.

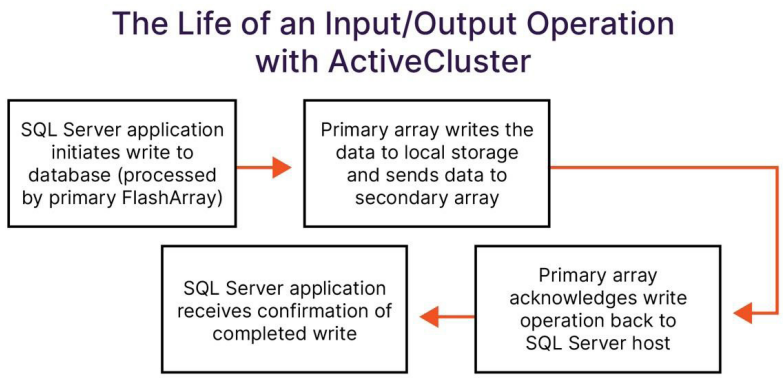


FIGURE 7 Overview of the life of an input/output operation with ActiveCluster

This data flow ensures that every write is mirrored in real time across both arrays, supporting the ActiveCluster promise of zero recovery point objectives and enabling fast, seamless failover when necessary.

Pure Storage Cloud Dedicated

Pure Storage Cloud Dedicated provides seamless data mobility across on-premises and cloud environments with a consistent experience, regardless of where data lives. It provides enterprise-grade storage features in the cloud, and its industry-leading data efficiency means you buy less capacity in the cloud without sacrificing agility and flexibility. Pure Storage Cloud Dedicated is available in the Amazon Web Services Marketplace and the Microsoft Azure Marketplace.

Pure Storage Cloud Dedicated is available in two versions, //V10 and //V20. Each version provides different capacity and performance capabilities.

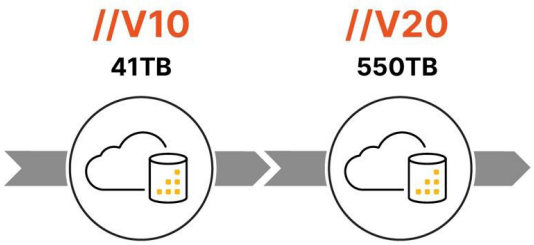


FIGURE 8 The //V10 and //V20 Pure Storage Cloud Dedicated versions provide the effective capacity listed with a 4:1 data reduction rate

VMware vSphere Integration with ActiveCluster

VMware vSphere integration with ActiveCluster provides a powerful solution for virtualized SQL Server environments, allowing seamless storage management, high availability, and scalability across multiple sites. This section covers aspects of that integration and focuses on the support for VMware vSphere VMFS and VMware vSphere Virtual Volumes, along with best practices and host/storage considerations.

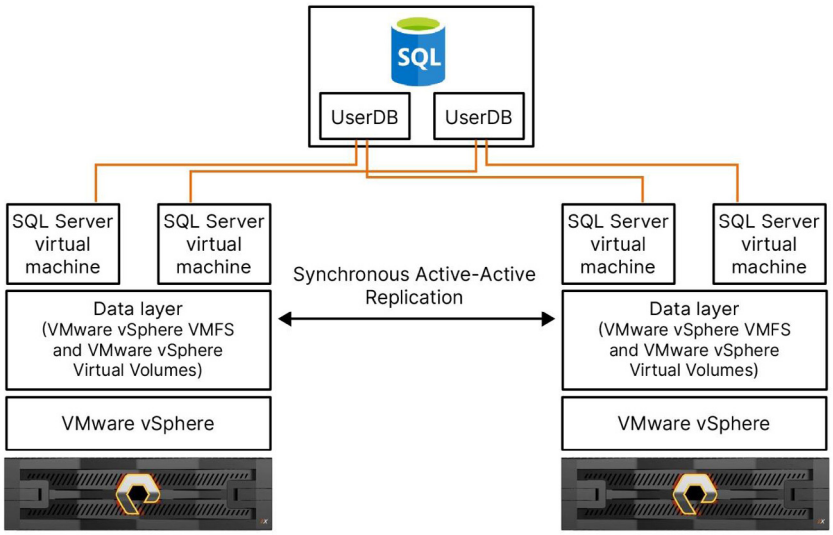


FIGURE 9 Example VMware environment deployed to make use of ActiveCluster

vSphere VMFS and vSphere Virtual Volumes Support

VMware environments support two key storage technologies, both of which integrate seamlessly with ActiveCluster and offer distinct benefits for managing SQL Server virtual machines:

- **vSphere VMFS:** This high-performance cluster file system is used by VMware ESXi hosts to store virtual machine files. It enables multiple ESXi hosts to read/write to the same storage concurrently, making it ideal for shared storage environments like ActiveCluster.
- **vSphere Virtual Volumes:** This storage technology offers granular storage control by abstracting physical storage resources and presenting them as logical storage objects directly aligned with individual virtual machines and their data. Unlike vSphere VMFS, which operates at a higher level, vSphere Virtual Volumes integrates more deeply with storage systems, enabling virtual machine–centric operations and direct storage management for each virtual machine.

Integrating VMware environments with ActiveCluster for SQL Server can provide the following benefits:

- **High availability:** Synchronous replication between arrays with ActiveCluster ensures that SQL Server databases hosted in virtual machines remain highly available across multiple sites or arrays. Failover is seamless, with minimal disruption to SQL Server services.
- **Simplified management:** ActiveCluster enables easy management of vSphere VMFS and vSphere Virtual Volumes environments, allowing VMware administrators to replicate entire datastores or individual virtual machines across sites.
- **Improved performance:** ActiveCluster enhances input/output performance by leveraging VMware solutions' multipathing and load balancing capabilities, ensuring that SQL Server workloads run efficiently even during peak demand.
- **Business continuity:** With ActiveCluster, VMware solution–based SQL Server environments can achieve near-zero recovery point objectives, ensuring data integrity and continuous availability during failures or site outages.

For detailed steps for configuring ActiveCluster for VMware solutions, see the ActiveCluster with VMware user guide.

Configuring and Managing ActiveCluster

A major benefit of using an ActiveCluster stretched storage solution is simplicity. This section walks you through aspects of configuring and managing ActiveCluster to harness that simplicity for your SQL Server workloads.

Before setting up ActiveCluster with SQL Server Failover Cluster Instances, it's important to review the FlashArray user guide, in addition to ActiveCluster, SQL Server, and Windows Server best practices documentation:

- Microsoft SQL Server
- Best Practices for Microsoft SQL Server on FlashArray
- Failover Clustering Feature
- ActiveCluster Solution Overview

Pre-workflow

Before beginning, define the storage to be used for ActiveCluster. Figure 10 showcases the volumes in the FlashArray graphical user interface.

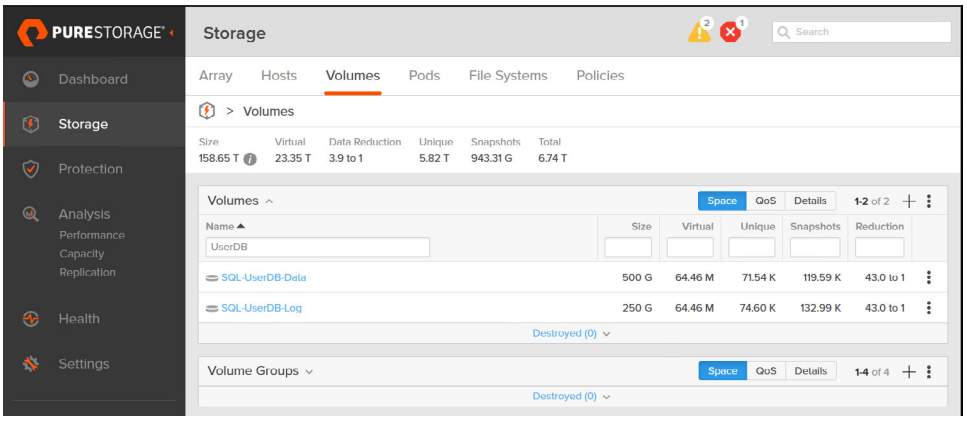


FIGURE 10 Pure1 storage console

For Windows systems, in **Disk Management**, check that your selected volumes are listed as available and healthy.

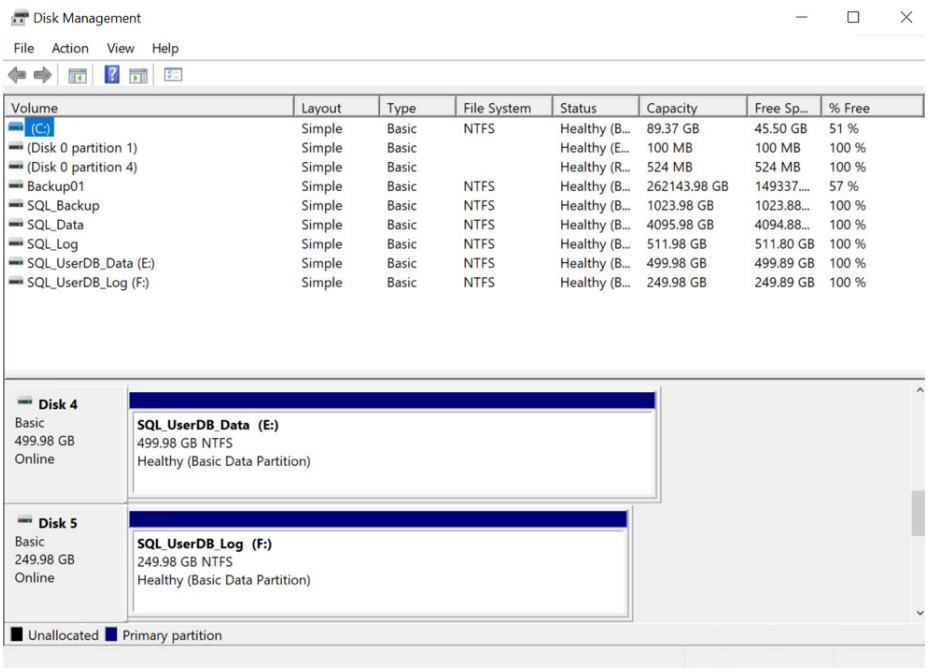


FIGURE 11 Windows disk management

For the SQL Server UserDB volume, check the database health and verify that the data file locations are as expected.

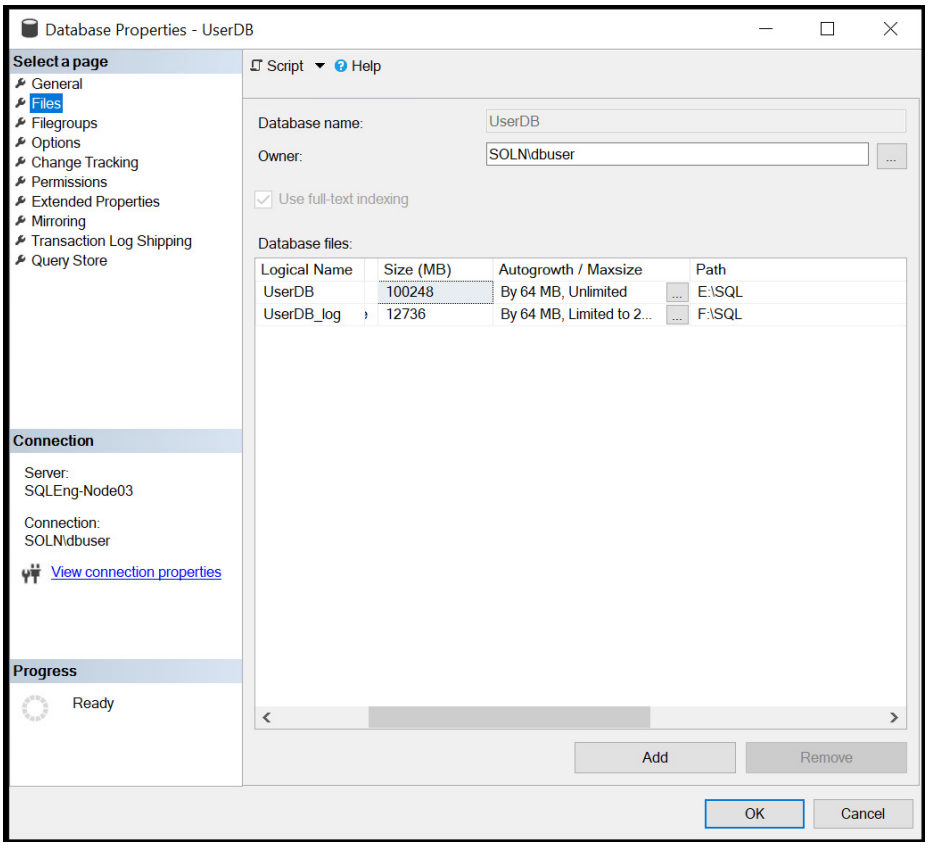


FIGURE 12 The user database properties shown using SQL Server Management Studio

Creating a Synchronous Connection

To set up ActiveCluster, start by creating a synchronous connection with another FlashArray. It's worth mentioning that you can use either of the FlashArray systems intended for ActiveCluster to connect to the other—both work equally well for the task.

To start this process, navigate to the **Array** tab in the **Protection** view of the graphical user interface, as shown in Figure 13.

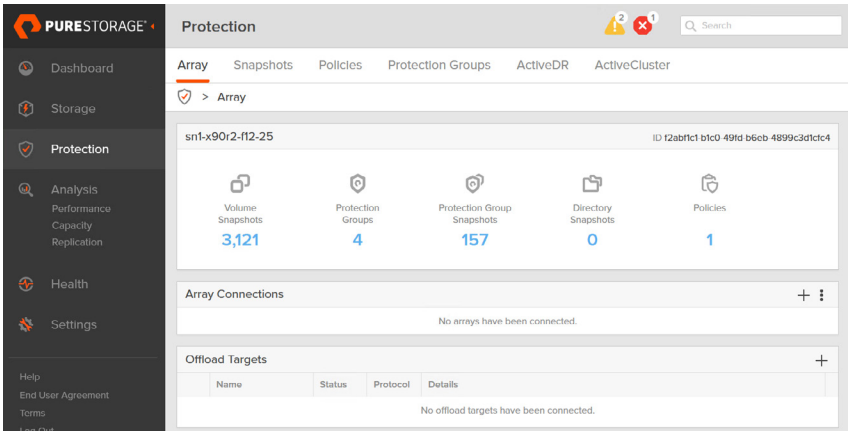


FIGURE 13 The Array tab in the Protection view of the FlashArray graphical user interface

Obtaining a Connection Key from a Target Array

For ActiveCluster to work, a Fibre Channel or TCP/IP connection must exist between the two FlashArray systems involved in replication. To establish this connection, obtain the connection key from the other FlashArray by using the graphical user interface.

To retrieve the connection key, click the vertical ellipsis in the right-hand corner of **Array Connections**, and then select **Get Connection Key**.

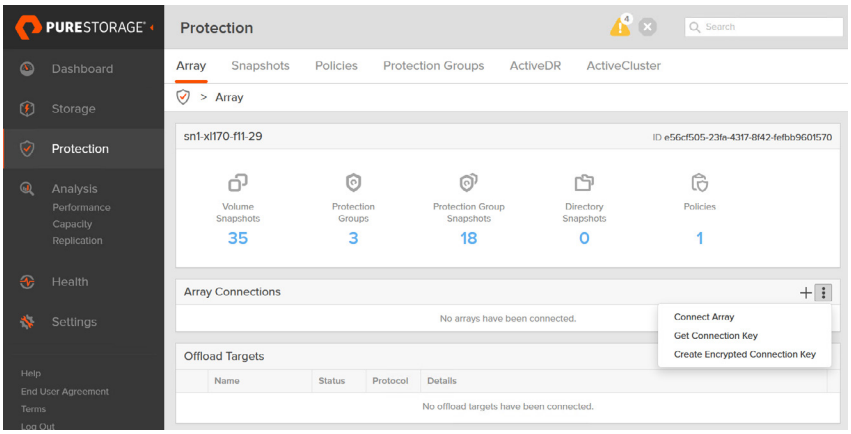


FIGURE 14 Array Connections context menu

The Connection Key dialog will display. Copy the key, and then return to the source array.

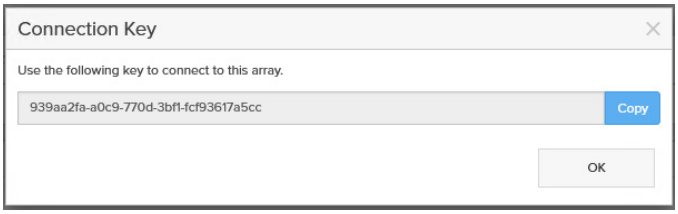


FIGURE 15 The Connection Key dialog

Connecting the Arrays for Synchronous Replication

On the corresponding FlashArray, utilize the connection key and the management IP address of the target array to establish a synchronization replication relationship. The provided example uses Ethernet as the replication transport, but it is equally feasible to employ Fibre Channel if preferred.

To connect the arrays, navigate to the **Arrays** tab within the **Protection** view, identify the **Array Connections** section, and then click the **+** to add a new array connection (as shown in Figure 16).

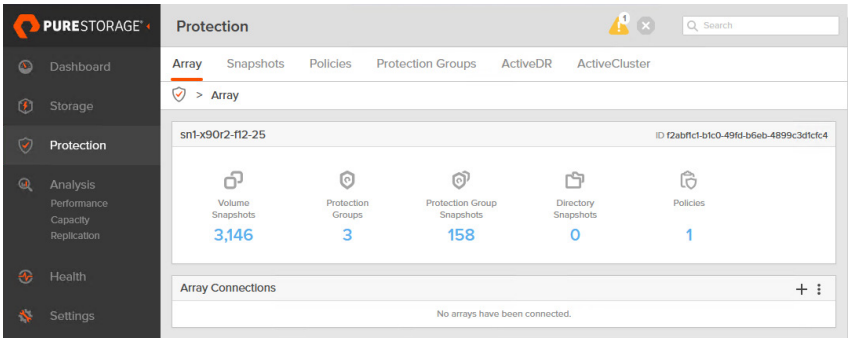


FIGURE 16 The Array Connections section in the Array tab on the target array

In the **Connect Array** dialog, enter the **Management Address** and **Connection Key** of the target array, and then select **Sync Replication** as the **Type**, adjusting any other parameters as required. Click **Connect** when complete.

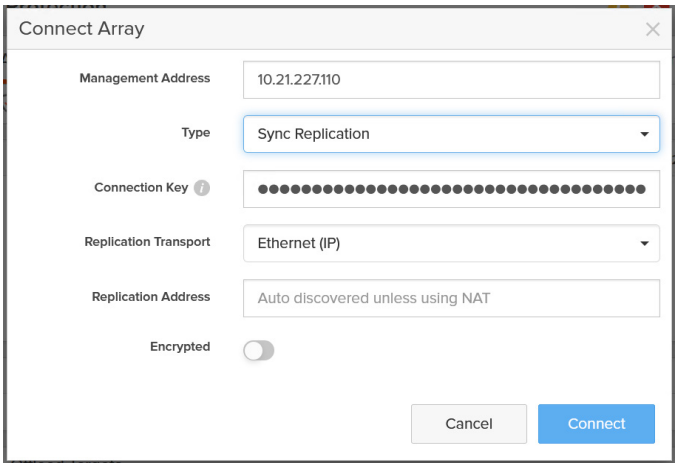


FIGURE 17 The Connect Array dialog

Verify the Connection Between Arrays

To confirm that the arrays are successfully connected, review each array's status in the **Array Connections** pane.

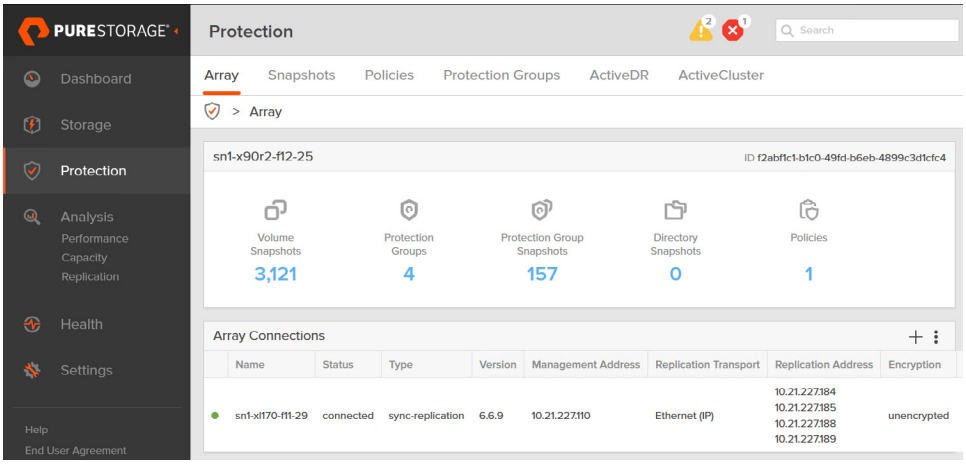


FIGURE 18 Check the arrays' connection statuses in the Array Connections pane

Managing a Pod

ActiveCluster replicates storage objects like volumes, snapshots, and schedules within consistency groups, referred to as pods. Each array can accommodate multiple pods, with each pod acting as an independent namespace for the objects it houses. An array administrator has the capability to stretch (replicate) a pod across two arrays, enhancing data redundancy and availability. An entire pod is synchronously replicated, ensuring everything in it stays consistent. If replication is interrupted, the first array to contact the mediator takes control of the entire pod.

Note: There are limits for ActiveCluster that differ based on the version of Purity being used. See the "ActiveCluster Synchronous Replication Limits" document for FlashArray and Pure Storage Cloud Dedicated for these limits.

All volumes that pertain to a single user database need to be in the same pod; it is also recommended to put related volumes in the same pod. This is useful for volumes with similar needs or when you want them to be consistent. It also makes administration easier. For volumes with different requirements, use separate pods to keep things organized.

If pre-existing database volumes are present, create the pod on the array to which it is localized.

The following steps showcase how to create a pod.

1. From the primary site's Pure Storage user interface, select **Storage** in the navigation pane, and then select the **Pods** tab. Click the **+** icon in the **Pods** group to create a new pod.

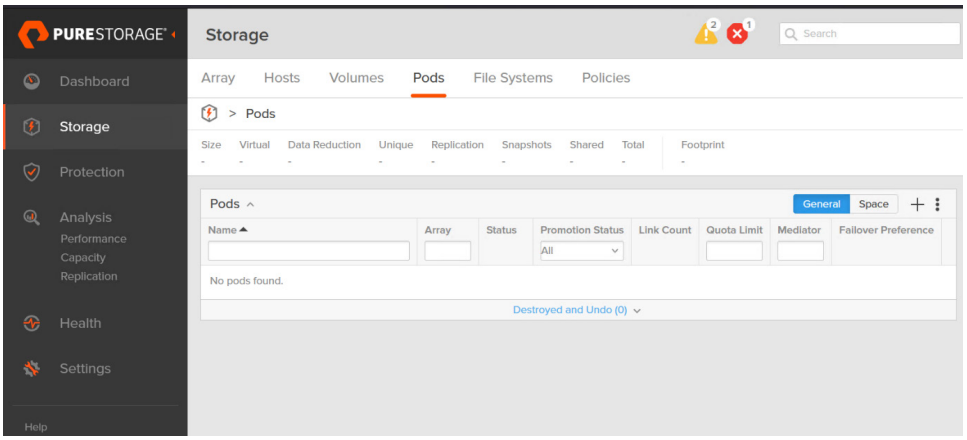


FIGURE 19 The Pods tab in the Storage view

2. In the **Name** field in the **Create Pod** dialog box, enter a name for the pod, and then click **Create**. The new pod will appear in the list.

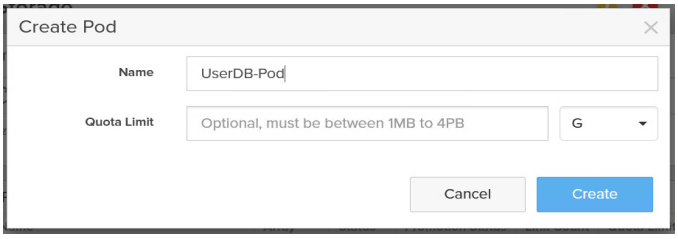


FIGURE 20 The Create Pod dialog

Adding Volumes to a Pod

Before stretching a pod (enabling synchronous replication to another array), it's possible to move existing volumes into it. The following steps showcase how to add volumes to an existing pod.

1. Select the pod created in Figure 20 from the list of pods.

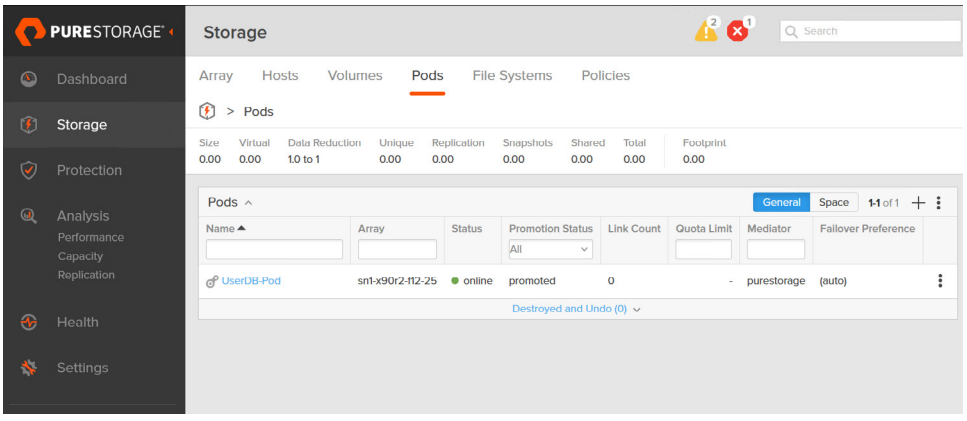


FIGURE 21 The Pods list

2. In the pod details view, click the ellipsis in the **Volumes** pane, and then select **Move In** to move existing volumes into the pod.

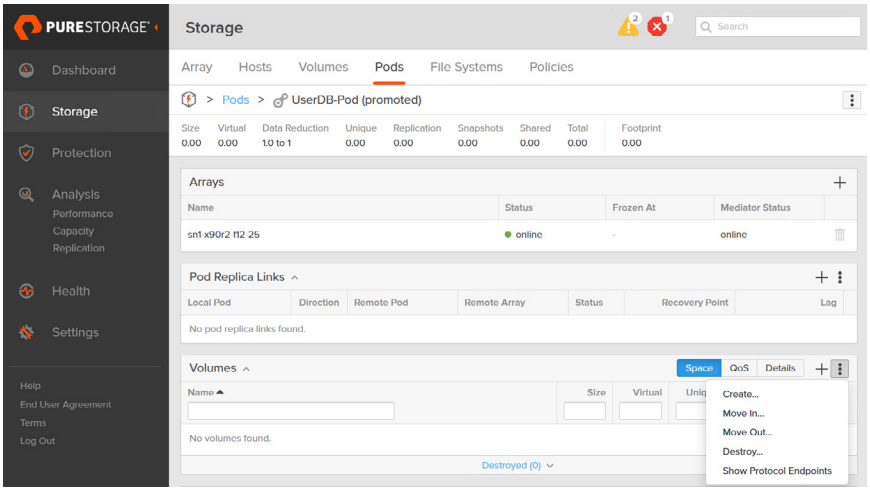


FIGURE 22 The Volumes pane in the Pods console

Moving Existing Volumes into a Pod

1. Existing volumes can be moved into the pod if the pod is not in a stretched state. When a volume is moved into a pod, it is renamed by adding the pod name followed by two colons as a prefix to its original name. This naming convention helps maintain clarity and organization within the pod.

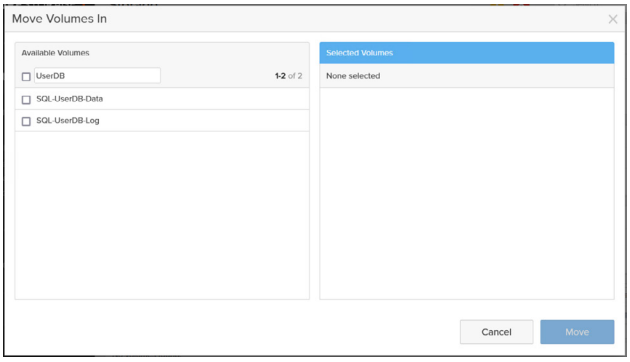


FIGURE 23 The existing pod with no volumes in the Move Volumes In dialog

2. Select the SQL Server database volumes to add to the existing pod, and then click **Move**.

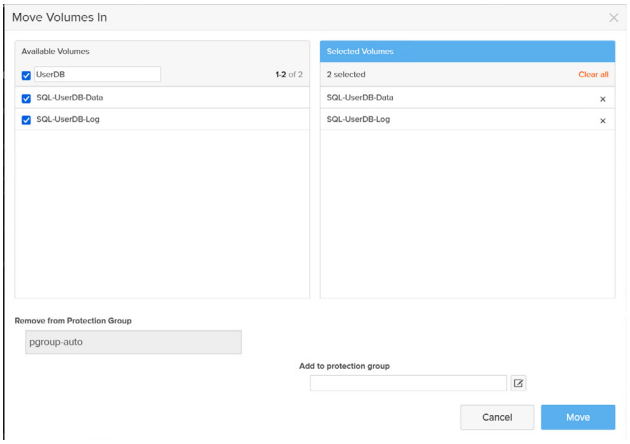


FIGURE 24 The existing pod with volumes selected in it in the Move Volumes In dialog

Creating a New Volume in a Pod

1. To create a new volume in a pod (whether stretched or unstretched), open the **Pods** screen, and then click the **+** symbol in the **Volumes** pane to create a new volume.

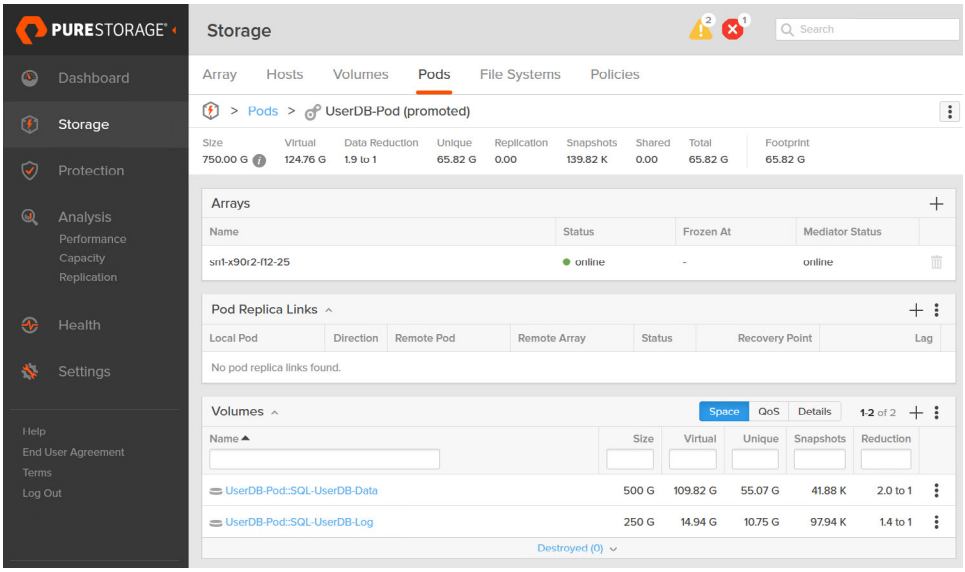


FIGURE 25 View of all pods

Stretching a Pod

Expanding a pod initiates the replication of volumes added to that pod and conducts the initial baseline copy of these volumes from the source to the target array. Once this baseline process is finished, the pod achieves high availability on both arrays, ensuring data redundancy and accessibility. To stretch a pod, follow these steps:

1. Navigate to the **ActiveCluster** tab in the **Protection** view in the FlashArray graphical user interface. From the **ActiveCluster Pods** section, click the **+** to stretch an existing pod.

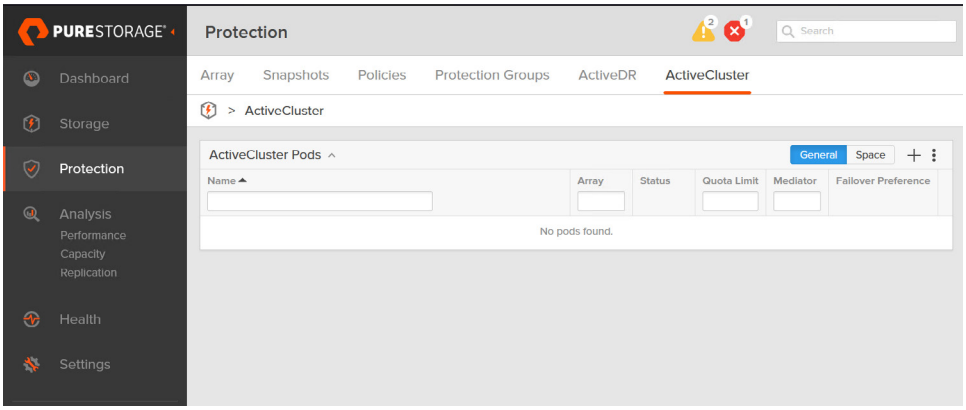


FIGURE 26 ActiveCluster pod

2. To stretch a pod, enter the **Local Pod** name and **Remote Array** name in the **Stretch Pod** dialog box, and then click **Stretch**.

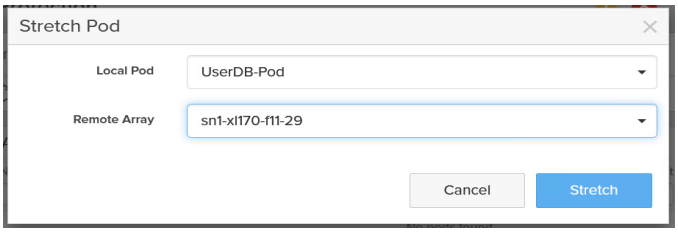


FIGURE 27 The Stretch Pod dialog with **Local Pod** name and **Remote Array** name entered

Note: While both stretched pods and protection groups are used to manage and replicate groups of storage objects, there are key differences. Protection groups are typically associated with asynchronous replication and are used to manage the recovery of groups of volumes, ensuring data consistency during disaster recovery scenarios. On the other hand, stretched pods in ActiveCluster are designed specifically for synchronous replication, ensuring that all objects in the pod remain perfectly in sync across arrays. This ensures that any failover event maintains data availability and consistency with no data loss, making stretched pods ideal for active-active, high availability configurations.

Verifying a Pod's Status

When stretching a pod, a baseline is performed where data is replicated to the target array.

Note: ActiveCluster leverages a baseline that is a full virtual copy of the data on the volumes within the pod. However, this baseline is data-reduced, ensuring efficiency by matching the target blocks already present with those to be transferred. Only the required block differences between the source and target are transmitted, which optimizes bandwidth usage and significantly reduces replication time.

Once this baseline is complete, the status of the target array will change from resyncing to online.

1. Using the pod status view, check the status details of the arrays for a pod.

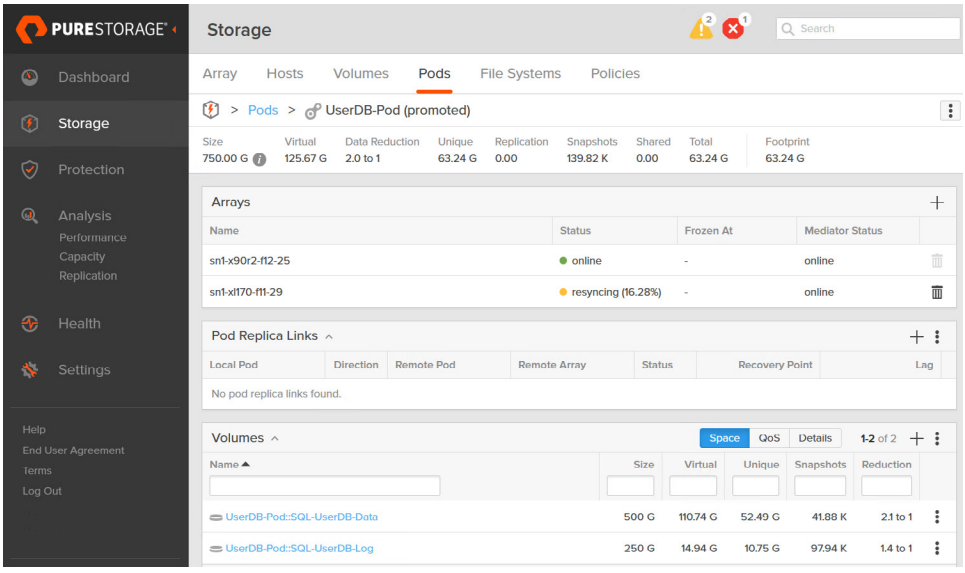


FIGURE 28 View of the status of a single pod with the target array resyncing

2. When the pod shows that both arrays have the online status, they are available and protected by both arrays.

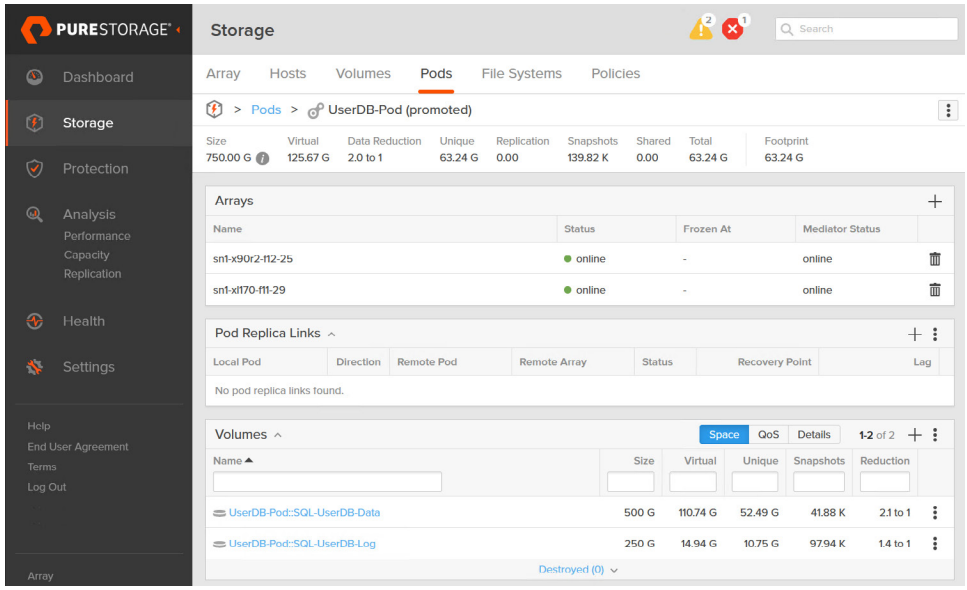


FIGURE 29 View of a single pod with ActiveCluster array status online

Configuring Windows Server Hosts

FlashArray provides support for both iSCSI and Fibre Channel host connections, and these options are fully compatible with ActiveCluster. To facilitate storage provisioning to a host, an array administrator is required to create a corresponding host object. Additionally, administrators have the flexibility to group hosts with shared storage provisioning needs, such as SQL Server hosts, into host groups. This management feature streamlines storage management and allocation.

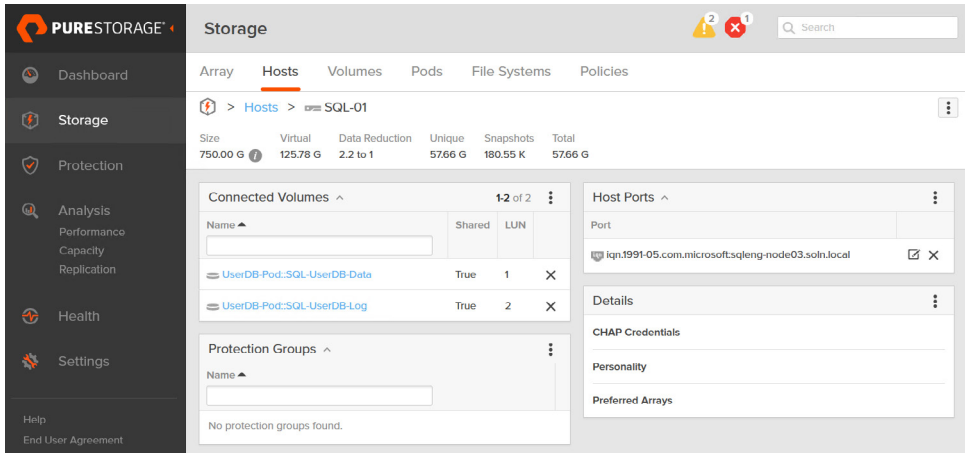


FIGURE 30 View of all SQL Server hosts

Uniform Configuration for Primary Site A

1. Create site A hosts on a FlashArray.
2. Present FlashArray storage volumes to each site A host.
3. Create a site A host group.
4. Associate all site A hosts to the newly created host group.
5. Connect site A FlashArray storage volumes.

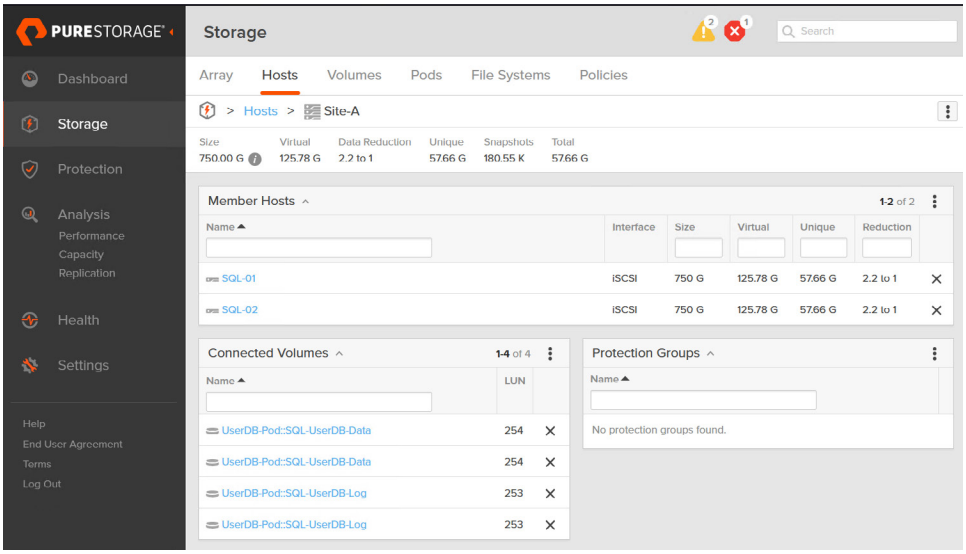


FIGURE 31 View of SQL Server database volumes on a single site (A)

6. In Windows Server, ensure storage array volumes are connected at each node located at site A.

7. Ensure online FlashArray storage volumes are only at the site A primary node where SQL Server will be active.

8. Ensure that the non-primary node volumes are not online (Figure 32).

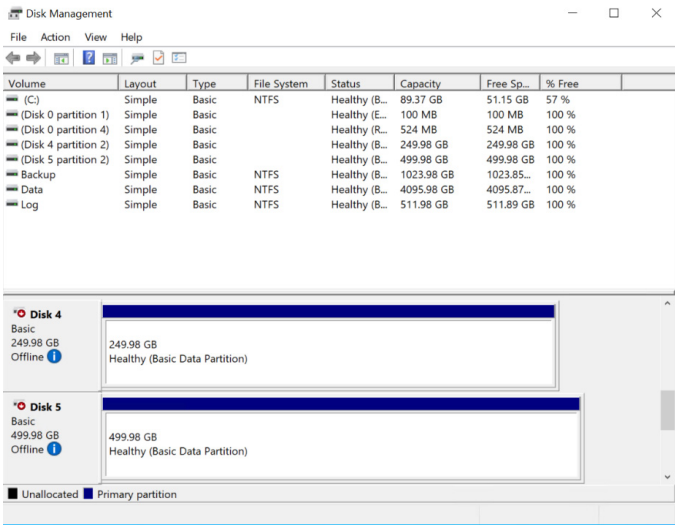


FIGURE 32 The Windows Server Disk Management console

Note: Online volumes only at the site A primary active node. On site A, only the primary active node initializes, formats, and assigns drive letters and labels to the storage volumes listed in the preceding steps.

9. Stretch the site A FlashArray storage to the site B FlashArray storage.

Note: Hosts cannot be connected to the volumes on the secondary array until the stretch is complete and the pod is online at both the primary and the secondary arrays.

10. Verify that pod status is listed as online and the baseline stretch is completed.

Uniform Configuration for Primary Site B

1. Create site B hosts on a FlashArray.
2. Present FlashArray storage volumes to each site B host.
3. Create the site B host group.
4. Associate all site B hosts to the newly created host group.
5. Connect the site B FlashArray storage volumes.
6. In Windows Server, ensure the storage array volumes are connected at each node located at site B.

Note: Online volumes only at the site A primary active node. On site A, only the primary active node initializes, formats, and assigns drive letters and labels to the storage volumes listed in the preceding steps.

Create and Configure Site B Host Group Access for Site A

1. Create a site B host group located on the site A FlashArray.
2. Associate site B hosts to the new host group created at the site A FlashArray.
3. Connect SQL Server cluster FlashArray volumes according to the host group created at the site A FlashArray.

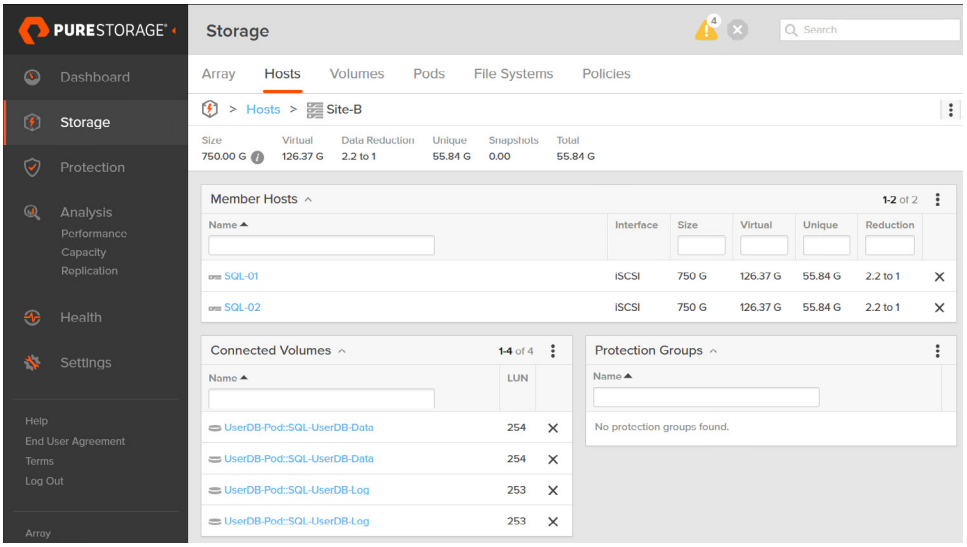


FIGURE 33 View of all SQL Server database volumes at the second site (B)

Preferred Paths

The default behavior is that all paths from a FlashArray to a host will be actively used by the SQL Server primary active node—even ones from the secondary FlashArray.

FlashArray offers an option to intelligently tell the SQL Server primary active node which FlashArray should optimally service input/output in the event the SQL Server primary active node can see paths to both FlashArray systems for a given device. This is a FlashArray host object setting called “Preferred Arrays.”

In a situation where the FlashArray systems are in geographically different data centers, it is important to set the preferred array for a host on both FlashArray systems. For each host, log in to the FlashArray web interface for the array that is local to that host. Once logged in, select the **Storage** console and **Hosts** tab, and then choose the host to be configured. In the **Details** submenu, click the vertical ellipsis, and then select the **Add Preferred Arrays** option.

Note: Caution is advised when replication occurs over extended distances; this is generally not ideal. In situations where the sites are far apart, two performance-impacting issues arise:

Half of the writes (assuming both FlashArray systems offer an equal number of paths for each device) sent from a host in site A will be sent to the FlashArray in site B. Using preferred arrays mitigates unnecessary latency by ensuring hosts prioritize writing to the local FlashArray. Without preferred arrays, writes from a host in site A may be sent directly to the remote FlashArray in site B. Since writes must be acknowledged at both sites, this can cause data to traverse the wide-area network twice—first from the host to the remote FlashArray and then back to the local FlashArray—introducing significant latency. Preferred arrays ensure writes are sent to the local FlashArray first, which then forwards them to the remote FlashArray, reducing traversal to a single pass and optimizing performance.

Half of the reads (assuming both FlashArrays offer an equal number of paths for each device) sent from a host in site A will be sent to the FlashArray in site B. However, under normal circumstances, all reads can be serviced locally by either array without traversing the wide-area network. This extra, unnecessary read traffic simply uses up network bandwidth.

Best practice: For every host that has access to both FlashArray systems hosting an ActiveCluster volume, set the preferred FlashArray for the host on both FlashArray systems. Ensure on FlashArray A that it is preferred for host A and ensure on FlashArray B that FlashArray A is preferred for host A. Doing this on both FlashArray systems allows a host to automatically know which paths are optimized and which are not.

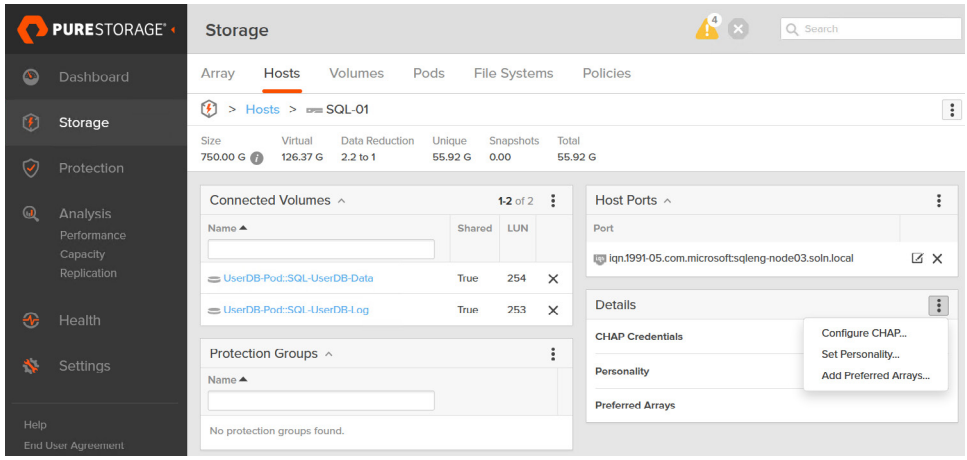


FIGURE 34 The preferred array menu on the Hosts console in Pure1

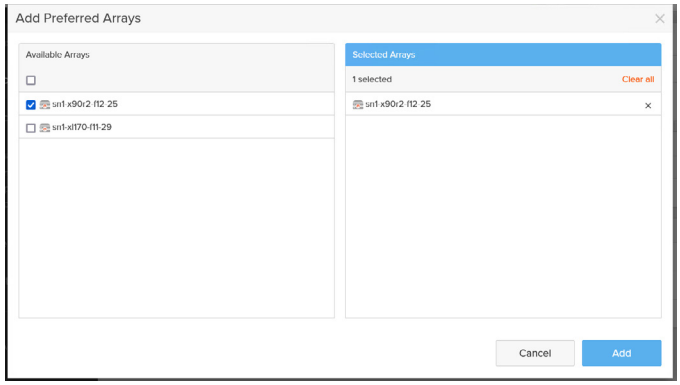


FIGURE 35 The Add Preferred Array dialog

Host Groups

To simplify the provisioning of storage for a cluster, host objects that correspond to the cluster's nodes can be grouped together on a host, known as a host group. This approach simplifies the management of storage resources for the entire cluster, enhancing efficiency and organization.

The Critical Importance of Proper Multipath I/O (MPIO) Configuration at the Host Level

In an ActiveCluster environment, Multipath I/O (MPIO) is essential for ensuring high availability and optimal performance of SQL Server instances. MPIO allows the host to maintain multiple redundant paths to both arrays, enabling continuous access to storage even if one or more paths fail.

Why MPIO Matters

MPIO matters for three principal reasons:

1. **Redundancy:** MPIO provides fault tolerance by ensuring that if one network path fails (due to a network issue, hardware failure, or site failure), the SQL Server instance can continue accessing the storage volumes using alternate paths. Without MPIO configured correctly, a single path failure could lead to a complete loss of storage connectivity.
2. **Load balancing:** Properly configured MPIO can balance input/output traffic across multiple paths, ensuring that SQL Server efficiently uses available bandwidth. This minimizes latency and optimizes throughput, particularly in high-demand environments.
3. **Seamless failover:** During failover scenarios, such as when switching to the secondary array in ActiveCluster, MPIO ensures that SQL Server can seamlessly transition its input/output operations to the available paths without interruption. Ensuring that MPIO is correctly configured across both arrays and hosts is critical to maintaining seamless operations.

Key Considerations for MPIO Configuration

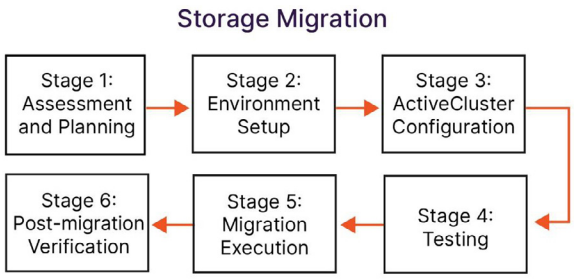
The key considerations for MPIO configuration include:

- **Driver installation:** Ensure that the appropriate MPIO drivers are installed on all SQL Server hosts and that they are compatible with your storage arrays.
- **Path validation:** Regularly test and validate all paths to the storage arrays to ensure that each path is functional and that there are no misconfigurations.
- **Load balancing policies:** Implement optimal load balancing policies (such as round robin or least queue depth) based on the environment to maximize performance.
- **Regular monitoring:** Continuously monitor the status of MPIO paths to ensure there are no issues, such as path failures or imbalanced traffic, that could degrade performance.

Note: Getting MPIO configuration right is not only about ensuring optimal performance; it's about ensuring that the SQL Server environment remains available and resilient even in the face of infrastructure challenges. Proper configuration and testing of MPIO at the host level are crucial for the success of an ActiveCluster deployment.

ActiveCluster for Nondisruptive Storage Migration

With the ever-increasing demand for availability and business continuity, organizations often require the flexibility to move their critical SQL Server databases. Reasons for moving databases can range from maintenance to disaster recovery to capacity to performance management between different arrays where another array could be faster. ActiveCluster can step in as a reliable, real-time storage replication and failover solution that simplifies the database migration process.



Six stages of non-disruptive storage migration using ActiveCluster.

ActiveCluster facilitates seamless and secure migration of SQL Server databases across sites or arrays located on the same site or across sites. It streamlines the migration process, reduces downtime, and ensures that critical SQL Server databases remain accessible and responsive during the migration process.

When migrating data using ActiveCluster, a carefully planned phased approach can help ensure a smooth and efficient transition.

Nondisruptive Storage Migration Between FlashArrays

Proceed through the following stages to execute a successful storage migration from one array to another.

Stage 1: Assessment and Planning

Start with a thorough assessment of the SQL Server database migration requirements, and then plan the migration process, including the source and target arrays, data replication, and failover procedures.

Stage 2: Environment Setup

Ensure that the SQL Server database environment, including the source and target sites, meets the hardware and software requirements for ActiveCluster.

Stage 3: ActiveCluster Configuration

Configure ActiveCluster by connecting the arrays and setting up synchronous replication to have the SQL Server database replicated to the target array. Refer to the Configuring and Managing ActiveCluster section of this document for detailed steps on how to configure ActiveCluster.

- Set up ActiveCluster on the source and target arrays.
- Configure the necessary synchronous replication relationship between the sites or arrays to ensure real-time data replication.
- Configure the SQL Server host to have uniform access to both arrays. Be sure to configure MPIO.

Stage 4: Testing

Perform comprehensive testing to validate the migration process and the failover mechanisms by disabling the paths to the source array and validating data availability from the target array.

Test data accessibility and the responsiveness of the array at the target site.

Stage 5: Migration Execution

When ready to migrate, initiate the failover from the source to the target site or array. The failover process is as simple as removing the paths of the source array from the SQL Server hosts. This ensures minimal downtime.

- Un-stretch the pod.
- Remove SQL Server hosts from Array A paths.
- Remove synchronous replication between the arrays.
- The SQL Server hosts can now access the database from Array B.

Stage 6: Post-migration Verification

After the migration is complete, verify the integrity of the SQL Server database on the target site to ensure that data remains consistent and accessible.

Ensuring Consistency and Availability During Migration

Maintaining data consistency and ensuring continuous availability are critical factors during SQL Server database migrations using ActiveCluster. The following best practices can help ensure that these goals are achieved; by following these steps, you can maintain data consistency and availability throughout the migration process, minimizing downtime and preventing data loss during the transition between storage arrays.

Synchronous Replication to Ensure Consistency

- ActiveCluster synchronous replication ensures that all data changes are immediately written to both the source and target arrays before being acknowledged. This helps ensure that the data is consistent across both arrays throughout the migration process.
- It's essential to monitor the replication status closely before, during, and after the migration to confirm that data is in sync. This can be done by regularly checking the health of the replication link and ensuring no replication lag exists.

Quorum and Host Monitoring

- During the migration, ActiveCluster uses the Pure1 mediator to maintain quorum and avoid split-brain scenarios. It is vital to ensure the witness is operational and reachable by both arrays.
- Monitor the connections of SQL Server hosts to the arrays to ensure they maintain a stable and consistent state, especially if failover is triggered.

Testing Failover and Data Path Validation

- Before executing the migration, thoroughly test the failover process. During testing, simulate failure conditions by disabling paths to the source array and validating that the SQL Server hosts can seamlessly access the data from the target array.
- Test data consistency by querying the database on both arrays during replication to ensure that no data loss or corruption occurs. Ensure that the SQL Server high availability features, such as SQL Server Always On Failover Cluster Instances, are configured correctly to manage failover without data loss.

MPIO for Redundant Access

- Ensure that multipathing is configured for SQL Server hosts. MPIO ensures that SQL Server hosts maintain continuous access to both arrays in case one path fails. This redundancy ensures that data access remains uninterrupted during storage migration.

Post-migration Data Validation

- After the migration and failover are complete, thoroughly validate the data on the target array. Run consistency checks on SQL Server (for example, using DBCC CHECKDB) to confirm that no data corruption occurred during migration. This step is crucial to ensure that data integrity is maintained.
- Ensure that all SQL Server workloads and applications can connect to the new array and that performance remains within acceptable parameters.

Additional Considerations

When performing a storage migration for SQL Server using ActiveCluster, there are several additional factors to consider in order to help ensure a smooth and efficient process. These considerations include the types of disks being used, their compatibility, and their performance implications.

Failure Scenarios

This section examines the availability of storage and network resources in different failure scenarios, and it also discusses troubleshooting common issues.

ActiveCluster Component Failures

Table 1 lists storage failure scenarios that can occur with ActiveCluster, and how data availability to hosts is affected.

Components				Storage Availability	SQL Server
Array 1	**Array 2**	**Replication Link**	**Mediator**		
Up	Up	Up	Up	Available on both arrays	Up and running on all the hosts, with access to both arrays
Up	Down	Up	Up	Available on the surviving array	Up and running on all the hosts, with access to Array 1 only
Up	Up	Down	Up	Available on one array	Up and running on all the hosts, with access to the chosen arrays
Up	Up	Up	Down	Available on both arrays	Up and running on all the hosts, with access to both arrays
Up	Down	Down	Up	Available on one array	Up and running on all the hosts, with access to Array 1 only
Up	Up	Down	Down*	Unavailable	Down, as both the replication link and the mediator are down
Up	Down	Down	Down*	Unavailable	Down, as both the replication link and the mediator are down, along with an array
Down	Down	Not applicable	Up	Unavailable	Down, as both of the arrays are down

TABLE 1 ActiveCluster component availability scenarios

* Entries in Table 1 labeled with asterisks (*) denote failures of components other than inaccessibility of the mediator. If the mediator becomes inaccessible after an array or the replication link has already experienced a failure, storage remains accessible on one array.

Host and Storage Network Failures

Table 2 lists the observable behavior of select storage components in the event of various failure scenarios.

Failure Scenario	Behavior
Single or multiple host failure	Applications can automatically fail over to other hosts on the same site or to other hosts at another site connected to the other array. This is driven by the Windows Server Failover Cluster feature assuming clusters are stretched between sites.
Stretched storage area network fabric outage (Fibre Channel or iSCSI) (Failure of storage area network interconnect between sites)	Host input/output continues automatically on local paths within the local site: • Encounters some storage path failures to the remote array but maintains input/output on paths to the local array. • In each site, retains access to local volumes with only a brief pause in input/output.
Storage area network fabric outage at one site	Applications can automatically switch to hosts at another site connected to a different array, driven by the host SQL Server cluster: • In the site without a storage area network issue, there might be some storage path problems for paths to the remote array, but operations traffic continues through paths to the local array. • The volumes on the remote array would be accessible, but latency would increase. The administrator could fail over the cluster to a node in the unaffected site.

TABLE 2 Storage component failure scenarios

Conclusion

ActiveCluster active-active replication on Pure Storage FlashArray, combined with Windows Server Failover Cluster and SQL Server Failover Cluster Instances, provides a robust solution for high availability and nondisruptive database migration. This configuration ensures that SQL Server workloads can operate with minimal disruption, even in the face of storage or site failures.

The simplicity and flexibility of FlashArray, along with the built-in capabilities of ActiveCluster, allow administrators to easily configure a highly available storage solution without the complexity of additional hardware or licensing. With automatic failover, synchronous replication, and seamless integration into SQL Server environments, organizations can maintain data consistency and continuous operations.

For businesses seeking cost-effective, scalable, and resilient storage solutions, ActiveCluster delivers an ideal blend of performance and ease of management, ensuring SQL Server environments remain online and efficient under any conditions.

For More Information

- Visit purestorage.com/microsoft.
- Try the functionality in a test drive.

PURESTORAGE®

©2025 Pure Storage, the Pure Storage P Logo, ActiveCluster, ActiveDR, FlashArray, FlashArray//C, FlashArray//E, FlashArray//X, FlashArray//XL, Pure1, and the marks in the Pure Storage Trademark List are trademarks or registered trademarks of Pure Storage Inc. in the U.S. and/or other countries. The Trademark List can be found at purestorage.com/trademarks. Other names may be trademarks of their respective owners.

PS2713-02-en 08/25