WHITE PAPER

Using ActiveDR with Microsoft SQL Server

Extend disaster recovery capabilities across data centers with FlashArray ActiveDR™.

Contents

Executive Summary

As data volumes continue to grow, database administrators who manage Microsoft SQL Server can find it difficult to protect their organizations against data loss. Traditional disaster recovery methods can be time-consuming and can negatively impact recovery time objectives and recovery point objectives for applications with sensitive service-level agreements. As a result, hardware failures, ransomware attacks, and disaster scenarios can lead to costly downtime.

The Pure Storage® FlashArray™ solution helps database and storage administrators create comprehensive disaster recovery strategies with the ActiveDR™ feature. This FlashArray feature provides continuous replication and a near-zero recovery point objective across geographically dispersed data centers. ActiveDR also simplifies disaster recovery workflows such as non-disruptive failover testing, live failovers, and failbacks.

How to Use This Document

This white paper provides an overview of ActiveDR and how to use it with SQL Server. It is intended for database administrators, storage administrators, and database reliability engineers who use SQL Server and FlashArray systems within the same data center, or across multiple geographically-dispersed data centers. This white paper discusses the benefits ActiveDR provides to database and storage administrators, while giving technical details on how to configure ActiveDR for use with SQL Server.

Solution Overview

Organizations rely on their SQL Server databases more than ever to keep their businesses running and their customers happy. As the volume of data continues to grow, database administrators need tools that give them the ability to quickly recover their databases without data loss. The combination of SQL Server and ActiveDR lets database administrators seamlessly maintain copies of their databases at remote sites that can be recovered quickly with near-zero recovery point objectives.

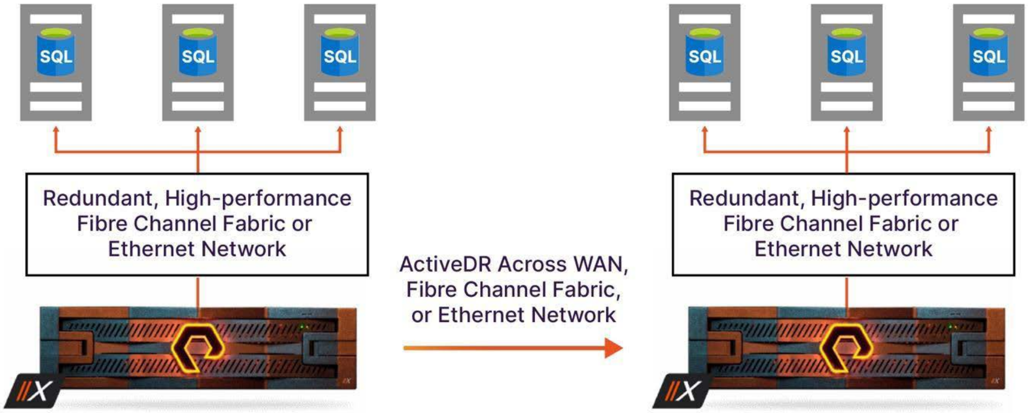

SQL
SQL
SQL
SQL
SQL
SQL
Redundant, High-performance Fibre Channel Fabric or Ethernet Network
Redundant, High-performance Fibre Channel Fabric or Ethernet Network
ActiveDR Across WAN, Fibre Channel Fabric, or Ethernet Network
//X
//X

FIGURE 1 High-level overview of SQL Server, FlashArray, and ActiveDR.

This solution includes the following:

- SQL Server instances running on Windows Server on either physical hardware or virtual machines
- FlashArray storage using block storage protocols such as Fibre Channel Protocol or iSCSI
- A wide area network between data centers if the FlashArray storage and SQL Server instances are geographically dispersed, or a Fibre Channel fabric or Ethernet network for FlashArray storage and SQL Server instances that are located within the same data center

Benefits of FlashArray and ActiveDR in a SQL Server Environment

This solution offers a number of benefits, including:

- **No additional licensing:** ActiveDR is included with FlashArray and Pure Cloud Block Store™ at no extra cost.
- **Reduced complexity:** ActiveDR replicates databases at the storage level. Database administrators don't have to worry about maintaining database-level replication to achieve their disaster recovery goals.
- **Simple, non-disruptive test failovers:** Storage and database administrators can test ActiveDR failovers with SQL Server without stopping replication, which does not impact recovery point objectives or recovery time objectives.
- **Fast recovery and failovers:** Storage administrators can perform failovers on protected volumes with a single command. Failovers include protected volumes and any protection group snapshots.
- **Near-zero recovery point objectives:** ActiveDR continuously streams writes between the source and target FlashArray systems, which provides near-zero recovery point objectives for protected databases without performance impact on the source FlashArray or SQL Server instance.
- **Near-synchronous data replication between sites:** ActiveDR provides near-synchronous data replication between primary and secondary sites. Data at a secondary site can be quickly attached and used by remote SQL Server instances in the event of a primary site failure, which can help simplify disaster recovery workflows.
- **No latency requirements:** Primary and secondary arrays can replicate at nearly any distance without affecting SQL Server performance.

Technology Overview

The following sections provide an overview of the technologies that are used in an ActiveDR environment for SQL Server.

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

ActiveDR Overview

ActiveDR, which is included in Purity//FA 6.0 and higher with no additional licensing, provides storage replication to remote data centers that can help protect SQL Server data against threats such as hardware failures, ransomware attacks, and user error. This replication enables near-zero recovery point objectives and simple disaster recovery. As an integrated function, ActiveDR helps simplify disaster recovery workflows in situations where data must be continually protected. Disaster recovery workflows, including non-disruptive test failovers, live failovers, resync, and failback, can be easily configured and executed with minimal complexity. Specifically, test failovers with ActiveDR are non-disruptive to production instances, allowing organizations to validate disaster recovery readiness without interrupting ongoing operations. These test failovers can be performed without halting replication, ensuring continuous protection of production workloads while validating recovery processes in a seamless, transparent manner.

Latency-sensitive applications are not impacted by ActiveDR replication due to the feature's focus on front-end application performance. ActiveDR relies on asynchronous streaming replication, which is ideal for replication across longer distances with higher-latency wide area networks. Synchronous replication requires a target array to acknowledge every write to the source array. This type of replication requires low network latency between the source and target arrays. Asynchronous replication does not require a remote array to acknowledge application writes to the source array, which means that replication can occur across networks with higher latency. The asynchronous nature of ActiveDR means that storage administrators can use existing wide area networks to replicate data without concern for distance or latency.

ActiveDR consists of three components:

- **Pods**: These are storage-management containers that organize storage objects and configuration settings into groups that are failed over and failed back as a unit. A pod can contain volumes, volume snapshots, and protection groups. Additionally, a pod can contain configuration settings such as protection group snapshot schedules, snapshot retention policies, and quality of service volume limits.
- **Replica links**: These provide replication between pods and provide directional and auto-reversing capabilities. Once a replica link is created, ActiveDR is automatically enabled.
- **FlashArray systems**: ActiveDR requires a minimum of two FlashArray systems connected over a network to replicate data between. With no latency requirements, these systems can be within the same data center or as far apart as on different continents.

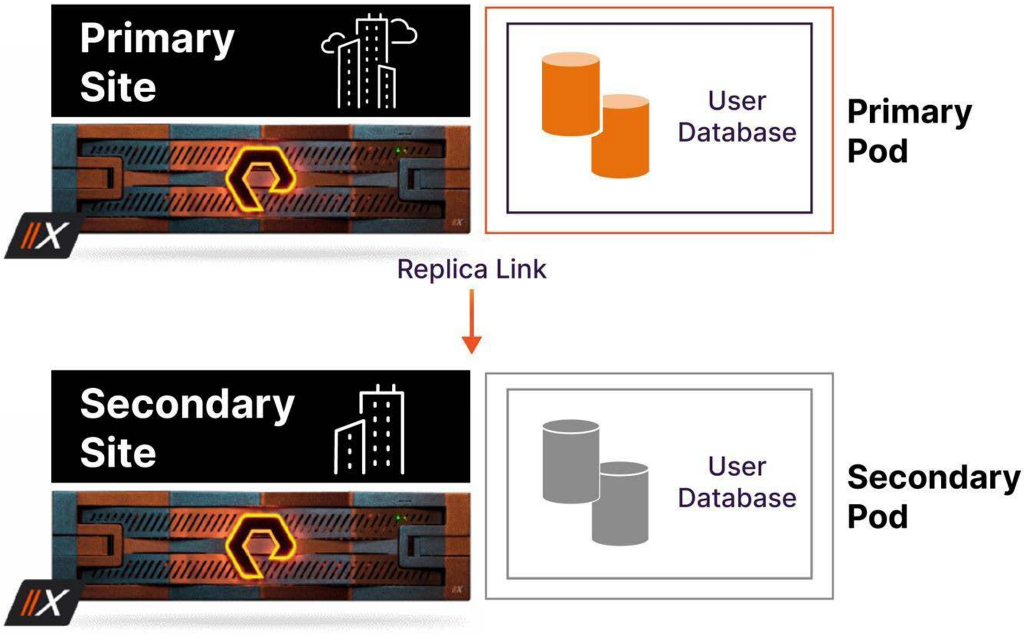


FIGURE 2 A typical ActiveDR deployment that includes pods and replica links between FlashArray systems.

Pod-based replication helps simplify storage management across sites by replicating all configuration changes made on a primary site array to the secondary site array, which helps simplify management and disaster recovery storage failovers. ActiveDR also supports multi-directional replication for different pods. For example, a database administrator might have a pod at their primary site that they want to replicate to their secondary site, while the secondary site might contain a pod that they want to replicate back to their primary site. ActiveDR lets them easily configure the pods to replicate in either direction between sites.

Microsoft 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. These features can be enhanced using ActiveDR to provide database administrators with more disaster recovery options.

SQL Server also provides a number of features that make it an ideal platform for multiple workloads, from online transaction processing to complex online analytics processing. SQL Server handles structured data with ACID 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. As a Microsoft product, SQL Server offers deep integration with other Microsoft enterprise-grade technologies, such as Active Directory and Microsoft Azure, and it can seamlessly support real-time data analysis, reporting, and complex workflows.

Within this white paper, SQL Server is the core database platform that supports an organization’s business applications and services.

Differences Between ActiveCluster, Asynchronous Replication, and ActiveDR

FlashArray has several storage replication functions that can be used to protect an organization's data. Among them are synchronous replication, which is used by the ActiveCluster™ solution; asynchronous replication storage array volume snapshots; and near synchronous replication with ActiveDR.

ActiveCluster

ActiveCluster uses synchronous replication to maintain copies of data between two FlashArrays. When data is written to a primary site FlashArray, it is simultaneously copied to a secondary site FlashArray. Once the data is written to both arrays, the write is acknowledged to the host system. This method of replication is recommended when the latency between arrays is 11ms or less, which means that ActiveCluster should be used between arrays in the same data center, or between data centers that have very-low-latency wide area network capabilities. For more information about ActiveCluster, see ActiveCluster Solution Overview.

Asynchronous Replication

Asynchronous replication is a snapshot-based solution that uses space-efficient snapshots to replicate data between FlashArray storage devices, while ActiveDR is a streaming-based solution that continuously replicates volume data between FlashArrays at different sites. When asynchronous replication is enabled on a volume at the primary site FlashArray, a snapshot of the volume is created on the primary site array and then replicated to the secondary site array. The first snapshot transfer is a baseline, which is a complete copy of the entire contents of the volume. All subsequent transfers are incremental transfers that result by comparing existing data on the storage array with the newly created snapshot to determine what data is sent to the secondary site array. For more information about asynchronous replication, see FlashArray Asynchronous Replication Configuration and Best Practices Guide.

ActiveDR

ActiveDR is different from the other replication solutions offered by FlashArray in that ActiveCluster is synchronous, with a zero recovery point objective and zero recovery time objective, whereas ActiveDR offers asynchronous replication, with recovery point objectives as low as five minutes. With ActiveDR, when data is written to a primary site array, the same data is immediately sent to the secondary site array. Worth mentioning that the data sent to the secondary pod is compressed, but it is not deduplicated before being sent. Deduplication occurs once the data lands on the secondary array. Because the data is sent asynchronously, the host that writes the data to the primary site array does not need an acknowledgement from the secondary site array. By using continuous replication, ActiveDR is an ideal solution for continuously replicating data between sites that are a greater distance from each other or that have higher-latency wide area networks between them. The continuous replication between sites can result in near-zero data loss, but when a failover is planned, SQL Server and the volumes protected by ActiveDR can be quiesced and replicated such that there is zero data loss.

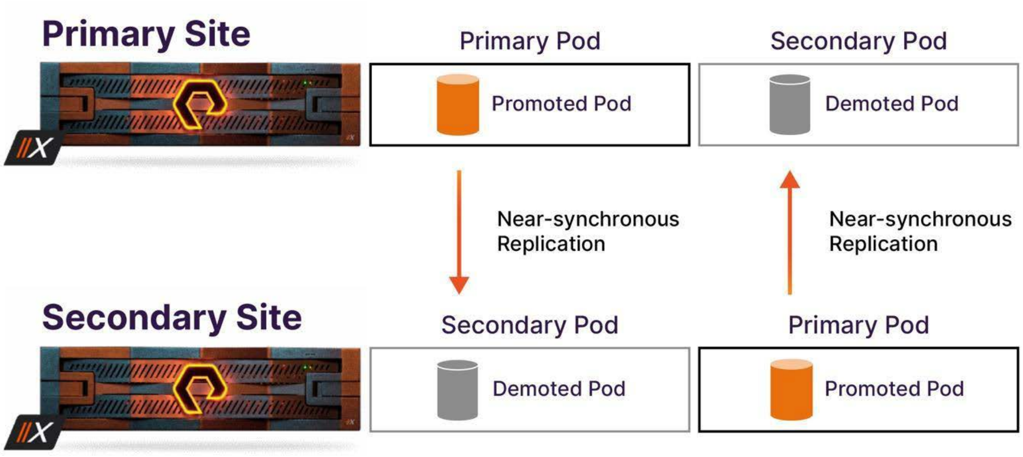


FIGURE 3 Pods in either the primary or secondary sites can be replicated to either site.

For more information about ActiveDR, visit the Pure Storage support site.

ActiveDR Disaster Recovery Strategies for SQL Server

Storage and database administrators can use ActiveDR to protect SQL Server data between a primary site array and a secondary site array.

Disaster Recovery Strategies for User Databases

A SQL Server instance contains several system databases and user databases. Each instance stores both the system database and user databases in files on a file system. These files include:

- **Database files (MDF and NDF)**: Every database has one MDF data file and, optionally, zero or more NDF files, with each file having an .mdf or .ndf extension. MDF files contain all data related to the database, including tables, views, stored procedures, and other database metadata.
- **Transaction log files (LDF):** Every database has one or more LDF transaction log files, which have an .ldf extension. The transaction log contains records of all changes made to a SQL Server database. These changes include updates, inserts, deletes, the start and end of each transaction, and other system transactions.

Note: Only the volumes that contain user database MDF, LDF, and NDF files should be replicated to a remote site with ActiveDR. TempDB files should not be replicated, though the SQL Server instance at the remote site does need to be configured with its own TempDB volume. System databases should be segregated into their own set of volumes apart from user databases. ActiveDR does not support the replication of system databases.

Once the primary site's SQL Server instance volumes are created and populated with user database files, the volumes are then placed in a FlashArray pod.

If a SQL Server instance contains multiple user databases, each set of user database files should be placed on different volumes. Using different volumes for different user database files helps storage and database administrators to:

- Manage recovery strategies on a per-database level. For example, a user database might require a shorter recovery time objective that benefits from replication to a remote site using ActiveDR, while another user database might require a less stringent recovery time objective that can use traditional backup and recovery methods or snapshots.
- Manage storage at a more granular level. Database administrators can monitor volume usage and expand volumes for different databases as needed. This avoids storage space conflicts where large databases impact the storage space needed for smaller databases.

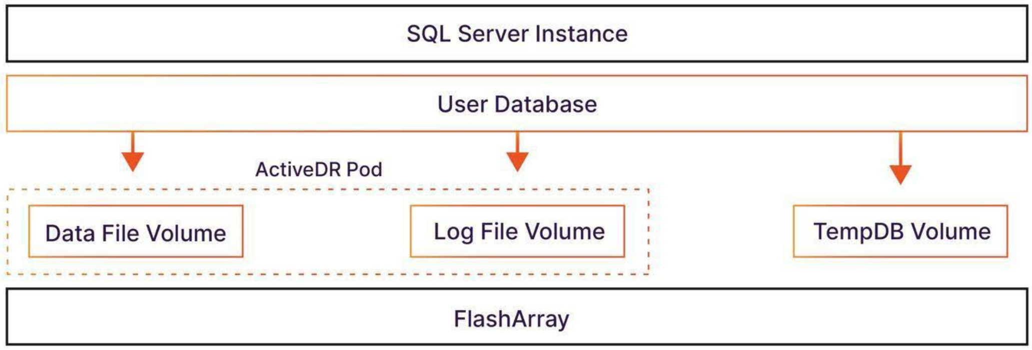


FIGURE 4 Volume and pod layout for databases on a single SQL Server instance.

Replicating a SQL Server VMware vSphere Virtual Machine

FlashArray provides integration with VMware vSphere that lets storage and database administrators replicate complete virtual machines to a remote location. In this scenario, FlashArray volumes are configured as VMware datastores (vVols are not supported) that are attached to VMware ESXi hosts. These volumes contain files for virtual machines and any attached virtual disks.

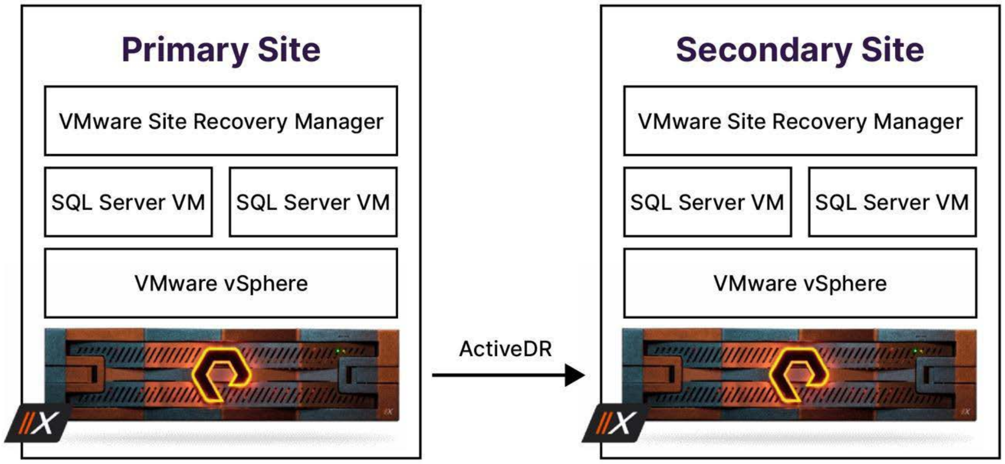


FIGURE 5 A virtualized SQL Server stack that uses Pure Storage FlashArray.

Database and storage administrators can configure SQL Server instances in virtual machines similar to physical SQL Server instances by creating separate virtual disks on separate FlashArray volumes that contain the virtual machine, SQL Server data files and log files. ActiveDR can then replicate the virtual machine, data file, and log file volumes to a remote site, where the volumes can be attached to a remote VMware ESXi host. This scenario is beyond the scope of this white paper, but more information is available in the ActiveDR with VMware User Guide.

ActiveDR and Always On Availability Groups

SQL Server Always On Availability Groups provide replication at the database layer.

The entirety of an Availability Group (with replicas participating in the Availability Group) can be added to an ActiveDR pod for disaster recovery failover/replication to another set of nodes in a remote location. Replicating a subset of an Availability Group is not supported with ActiveDR. Under certain circumstances in a failover scenario, only the primary replica will be available, and secondary replicas must be reseeded. This reseeding is dependent on the state of data movement occurring between the Availability Group replicas at the time of failover.

ActiveDR and SQL Server Always On Failover Cluster Instances

Microsoft provides SQL Server Always On Failover Cluster Instances that use Windows Server Failover Clustering to provide high availability at the SQL Server instance–level. Database and storage administrators can also use ActiveDR to replicate SQL Server data and log files at the storage level to a secondary site, typically on a different subnet. In this scenario, two or more SQL Server nodes utilize a shared Cluster Shared Volume, where the SQL Server database and log files are stored. The Cluster Shared Volumes are then placed into a FlashArray pod, which is then replicated to a remote FlashArray using ActiveDR. The replicated volumes at the remote site can then be attached to a secondary SQL Server Failover Cluster Instance should a failure at the primary site occur.

For more information about using SQL Server Always On Failover Cluster Instances between primary and secondary sites, see SQL Server Multi-Subnet Clustering (SQL Server).

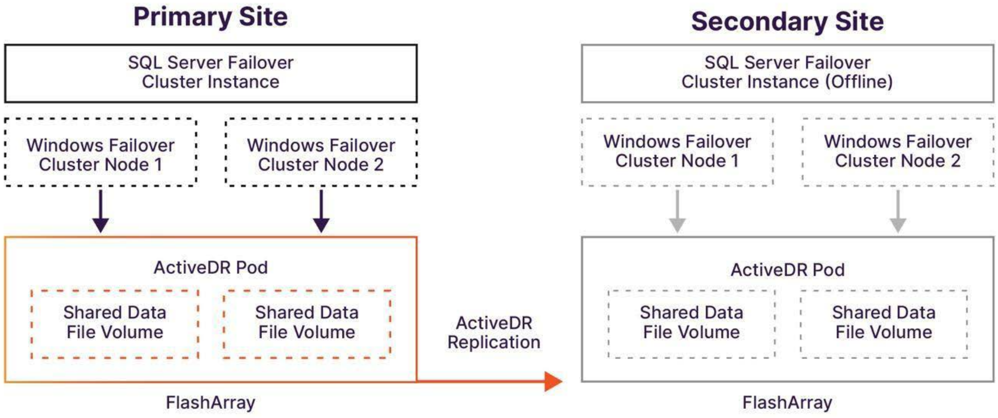

Primary Site
SQL Server Failover Cluster Instance
Windows Failover Cluster Node 1
Windows Failover Cluster Node 2
ActiveDR Pod
Shared Data File Volume
Shared Data File Volume
FlashArray
ActiveDR Replication
Secondary Site
SQL Server Failover Cluster Instance (Offline)
Windows Failover Cluster Node 1
Windows Failover Cluster Node 2
ActiveDR Pod
Shared Data File Volume
Shared Data File Volume
FlashArray

FIGURE 6 ActiveDR replication between two SQL Server Failover Cluster Instances.

Volumes that contain database files and log files should be configured in an ActiveDR pod.

Database administrators can configure SQL Server Failover Cluster Instances to automatically failover should a problem occur on one of the SQL Server Failover Cluster Instance nodes, or a failover can be manually initiated. In either an automatic or a manual failover, the node that takes over from the primary failed node continues to use the Cluster Shared Volume on the FlashArray at the primary site.

In a disaster recovery event where the primary site is offline, such as during a natural disaster, administrators can bring the replicated Cluster Shared Volumes online at the remote site. These volumes are already attached to the preconfigured remote SQL Server Failover Cluster Instance, allowing application requests to continue being processed seamlessly.

Once the primary site is brought back online, storage administrators can reverse the replication from the secondary site back to the primary site, and then failback to the primary site at the time of their choosing.

Using ActiveDR with SQL Server

The following sections describe how to configure and test ActiveDR with SQL Server. These sections assume that Windows Server servers at both the primary and secondary sites are already configured to work with FlashArray volumes. For more information, see Working with Volumes on a Windows Server Host.

While these sections describe configuring and testing ActiveDR using the Pure Storage user interface, administrators can also use PowerShell scripts to achieve the same goals. ActiveDR and other Pure Storage scripts are located on GitHub.

Configuring ActiveDR

The following steps must be completed using the Pure Storage user interface to initiate ActiveDR replication.

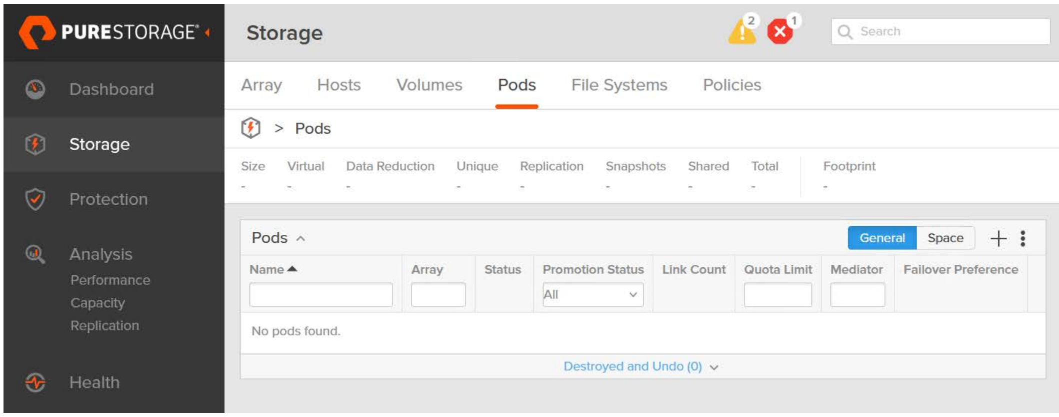
Configuring ActiveDR for SQL Server involves the following general steps:

1. Create volumes at the primary site to store SQL Server user database files.
2. Configure SQL Server to store user database files on FlashArray volumes at the primary site. Generally, database files, log files, and TempDB files are stored on separate volumes.
3. Move the primary site's SQL Server user database volumes into a pod. The TempDB volume should not be moved into a pod, as it does not need to be replicated.
4. Create an ActiveDR replica link from the source pod to the target replication array.

Configure the ActiveDR Primary Site Pod in the Pure Storage User Interface

To configure the primary site pod:

1. From the primary site's Pure Storage user interface, select Storage in the navigation pane, and then select the Pods tab



No pods found.

FIGURE 7 The Pods tab in the Pure Storage user interface.

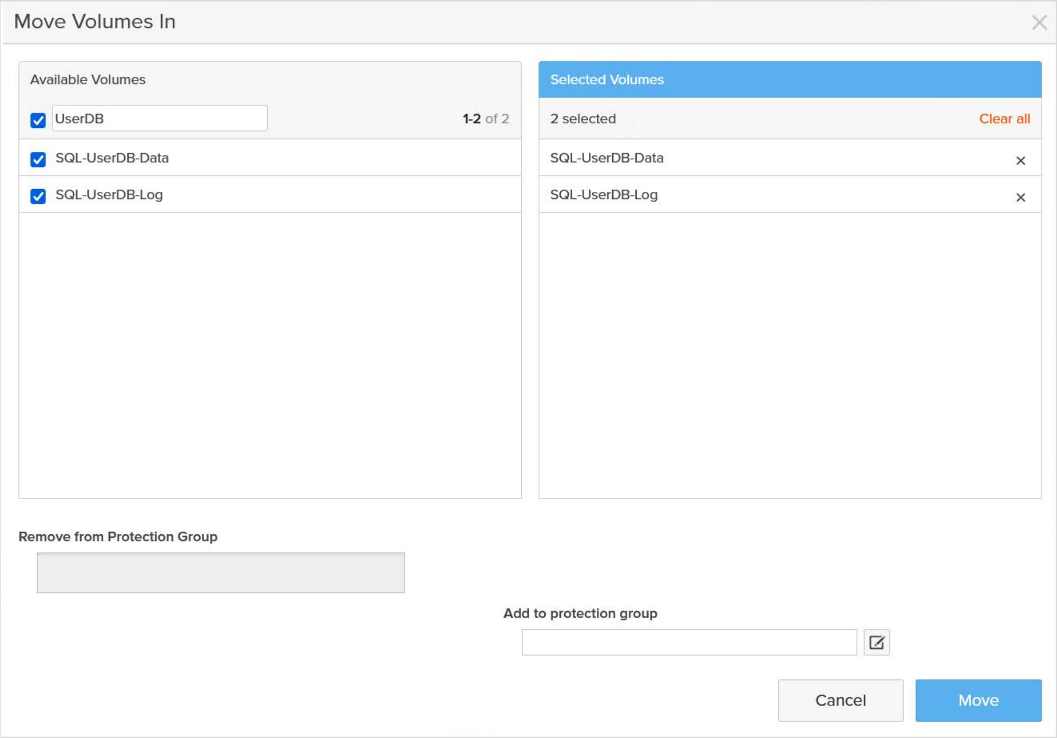
2. Click the **+** icon in the **Pods** group to create a new pod.

3. In the **Name** field in the **Create Pod** dialog box, enter a name for the pod, and then click **Create**.

   The pod appears in the Pods group.

4. In the **Pods** group, select the name of the pod that was created to display the **Pod management** view.

5. In the **Volumes** group, click the ellipsis, and then click **Move In** to display the **Move Volumes In** dialog box.

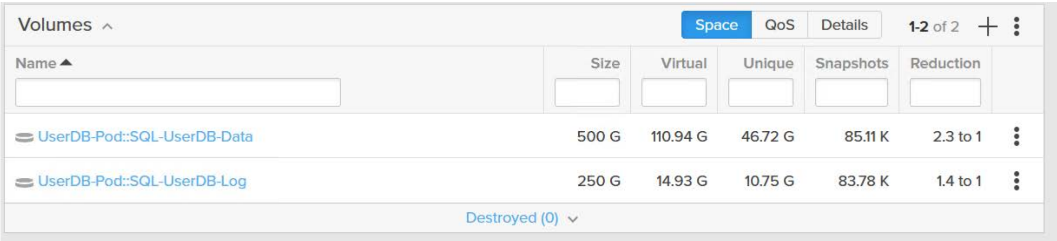


☑ UserDB 1-2 of 2
☑ SQL-UserDB-Data
☑ SQL-UserDB-Log

2 selected
SQL-UserDB-Data ×
SQL-UserDB-Log ×

FIGURE 8 The Pods tab in the Pure Storage user interface.

6. Select the SQL Server database file volume and log file volume that will be moved into the pod, and then click **Move**. The volumes appear in the **Volumes** group using the naming convention, *<pod name>::<volume name>*.



Name	Size	Virtual	Unique	Snapshots	Reduction
UserDB-Pod::SQL-UserDB-Data	500 G	110.94 G	46.72 G	85.11 K	2.3 to 1
UserDB-Pod::SQL-UserDB-Log	250 G	14.93 G	10.75 G	83.78 K	1.4 to 1

FIGURE 9 Displaying the volumes in a pod.

The primary site pod is now configured with the database file and log file volumes. The next step is to create a replica link to the remote site's FlashArray, which enables ActiveDR replication.

Create a Replica Link in the Pure Storage User Interface

1. From the remote site's Pure Storage user interface, select **Protection** in the navigation pane.
2. In the **Array Connections** group, click the ellipsis, and then select **Get Connection Key**.

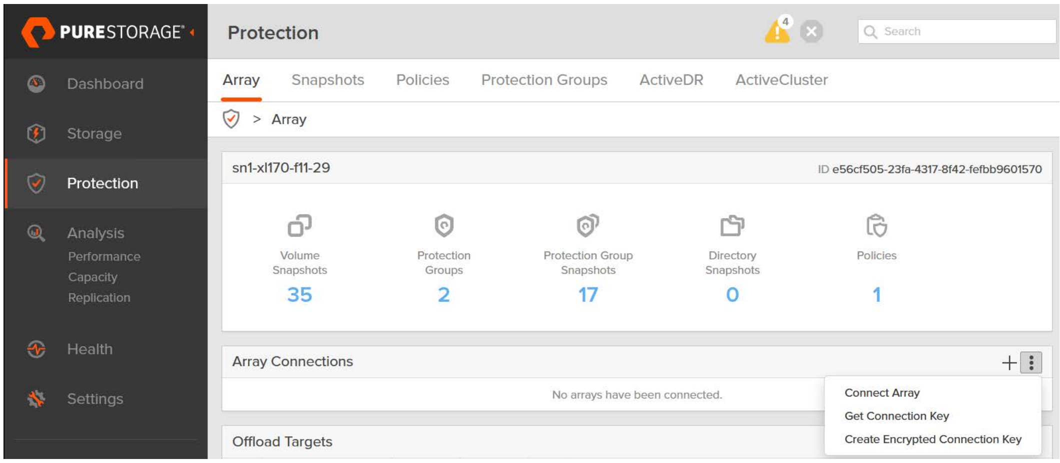
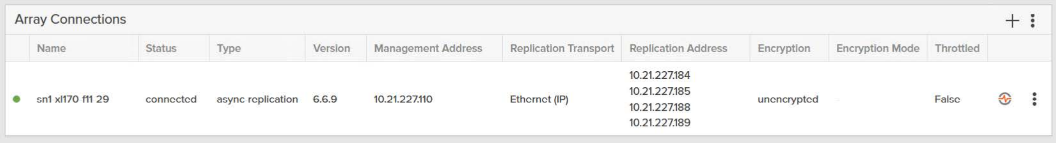


FIGURE 11 Getting a connection key on the Array tab.

3. In the **Connection Key** dialog, click **Copy** to copy the connection key to the clipboard.
4. From the primary site's Pure Storage user interface, click **Protection**.
5. In the **Array Connections** group, click the ellipsis, and then select **Connect Array**.
6. In the **Connect Array** dialog box, enter or select the following information:

   - **Management Address:** The IP address of the remote site's FlashArray.
   - **Type:** Select **Async Replication** from the drop-down list.
   - **Connection Key:** Paste the connection key that you copied from the remote site's Pure Storage user interface.
   - **Replication Transport:** Select the transport option from the drop-down list.
   - **Encrypted:** Select the button to toggle replication encryption on or off.

7. Click **Connect.** The secondary site array appears in the **Array Connections** group.

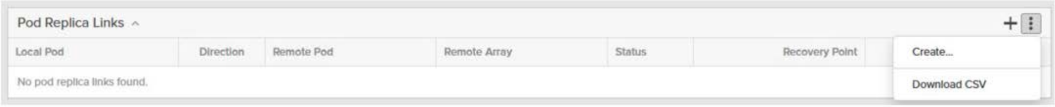


Array Connections

	Name	Status	Type	Version	Management Address	Replication Transport	Replication Address	Encryption	Encryption Mode	Throttled
●	sn1 xl170 f11 29	connected	async replication	6.6.9	10.21.227.110	Ethernet (IP)	10.21.227.184 10.21.227.185 10.21.227.188 10.21.227.189	unencrypted	-	False

FIGURE 12 Displaying the secondary site in the Array Connections group.

8. In the **Pod Replica Links** group, click the ellipsis, and then select **Create**.



Pod Replica Links

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point
No pod replica links found.					

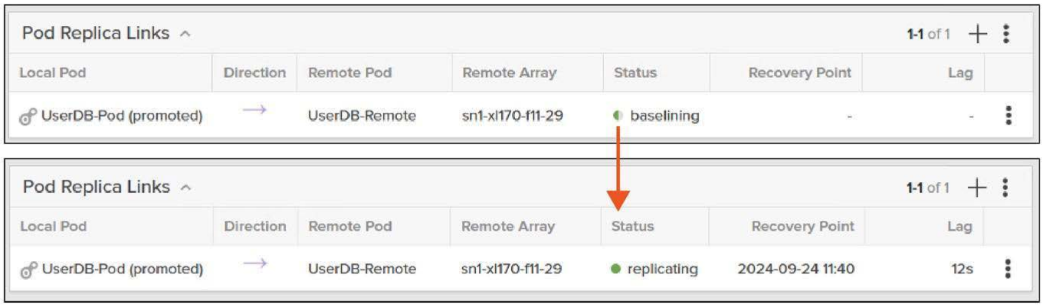
FIGURE 13 Creating a replica link.

9. In the **Create Replica Link** dialog box, select the remote site FlashArray from the **Remote Array** drop-down list, and then select a remote pod.

Note: If a pod hasn't been created on the remote site's FlashArray, click **Create Remote Pod**, enter a pod name in the **Name** field, and then click **OK**.

10. Select **Create**.

Once a replica link has been created from the primary site's pod to the secondary site's pod, the primary site's pod begins a baseline replication of the volumes to the secondary site's pod. During a baseline replication operation, the primary site FlashArray replicates a full copy of the volumes to the secondary site, which depending on the size of the data, can take time. The status of the baselining progress can be monitored in the **Status** field in the **Pod Replica Links** group. When the baseline replication completes, the **Status** field changes from "baselining" to "replicating."



Pod Replica Links 1-1 of 1

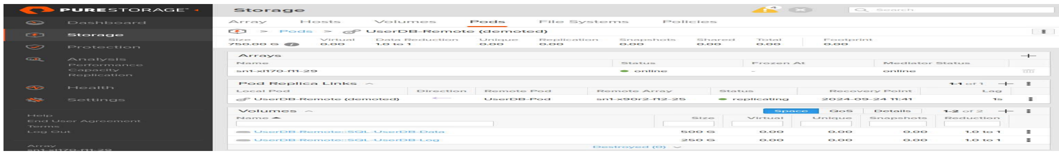
Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Pod (promoted)	→	UserDB-Remote	sn1-xl170-f11-29	baselining	-	-

Pod Replica Links 1-1 of 1

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Pod (promoted)	→	UserDB-Remote	sn1-xl170-f11-29	● replicating	2024-09-24 11:40	12s

FIGURE 14 Monitoring pod replica link statuses on the primary site's FlashArray.

In addition to monitoring the pod replica link status at the primary site, the replica link status can also be monitored in the secondary site's Pure Storage user interface in the **Pod Replica Links** group.



Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footprint
750.00 G	0.00	1.0 to 1	0.00	0.00	0.00	0.00	0.00	0.00

Arrays

Name	Status	Frozen At	Mediator Status
sn1-xl170-f11-29	● online	-	online

Pod Replica Links 1-1 of 1

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Remote (demoted)	←	UserDB-Pod	sn1-x90r2-f12-25	● replicating	2024-09-24 11:41	1s

Volumes Space QoS Details 1-2 of 2

Name	Size	Virtual	Unique	Snapshots	Reduction
UserDB-Remote::SQL-UserDB-Data	500 G	0.00	0.00	0.00	1.0 to 1
UserDB-Remote::SQL-UserDB-Log	250 G	0.00	0.00	0.00	1.0 to 1

Destroyed (0)

FIGURE 15 Monitoring the pod replica link status on the secondary site's FlashArray.

Non-disruptive Failover Testing

Non-disruptive failover with ActiveDR allows any disaster recovery/target site to bring a pod online without interrupting the production/primary pod. During a non-disruptive test, the databases in the primary pod remain online and continue to handle transactions seamlessly. This ensures uninterrupted production operations while enabling failover testing and allowing the target database to be used for testing and development purposes.

When using this process, the primary site will still queue changes, which will be applied to the target site once the process is undone automatically, with no manual intervention required. .

To perform a non-disruptive failover, the state of the environment must be as follows:

- There is a functioning database residing on one or more volumes in a promoted ActiveDR pod.
- There is a demoted remote pod on a separate array in the same active pod replica link.
- The volumes in the demoted pod are connected to, and have been discovered by, a host with a running SQL Server instance.
- The demoted volumes are offline.

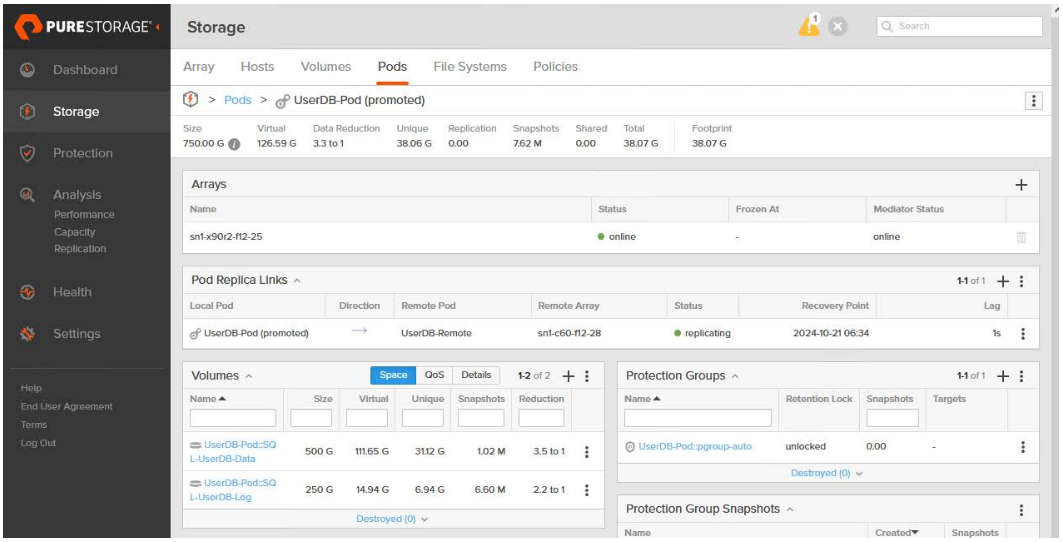
A non-disruptive failover consists of the following steps:

1. Promote the ActiveDR pod on the target array.
2. Set the disks to online and, if required, provide a drive letter or mount to a folder.
3. Attach the database to the SQL Server instance.

A failback after a non-disruptive failover only requires that the pod be demoted on the target system. Replication will automatically resume from the source pod to the target.

To complete the non-disruptive failover and failback between two arrays and different SQL Server host systems:

1. In the **Storage** section on the production/primary array, click the **Pods** tab, and then observe the state of the **Pod Replica Links** in a specific pod's details, which will show as **promoted**.



> Pods > UserDB-Pod (promoted)

Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footprint
750.00 G	126.59 G	3.3 to 1	38.06 G	0.00	7.62 M	0.00	38.07 G	38.07 G

Arrays

Name	Status	Frozen At	Mediator Status
sn1-x90r2-f12-25	● online	-	online

Pod Replica Links 1-1 of 1

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Pod (promoted)	→	UserDB-Remote	sn1-c60-f12-28	● replicating	2024-10-21 06:34	1s

Volumes 1-2 of 2

Name ▲	Size	Virtual	Unique	Snapshots	Reduction
UserDB-Pod::SQL-UserDB-Data	500 G	111.65 G	31.12 G	1.02 M	3.5 to 1
UserDB-Pod::SQL-UserDB-Log	250 G	14.94 G	6.94 G	6.60 M	2.2 to 1

Destroyed (0)

Protection Groups 1-1 of 1

Name ▲	Retention Lock	Snapshots	Targets
UserDB-Pod::pgroup-auto	unlocked	0.00	-

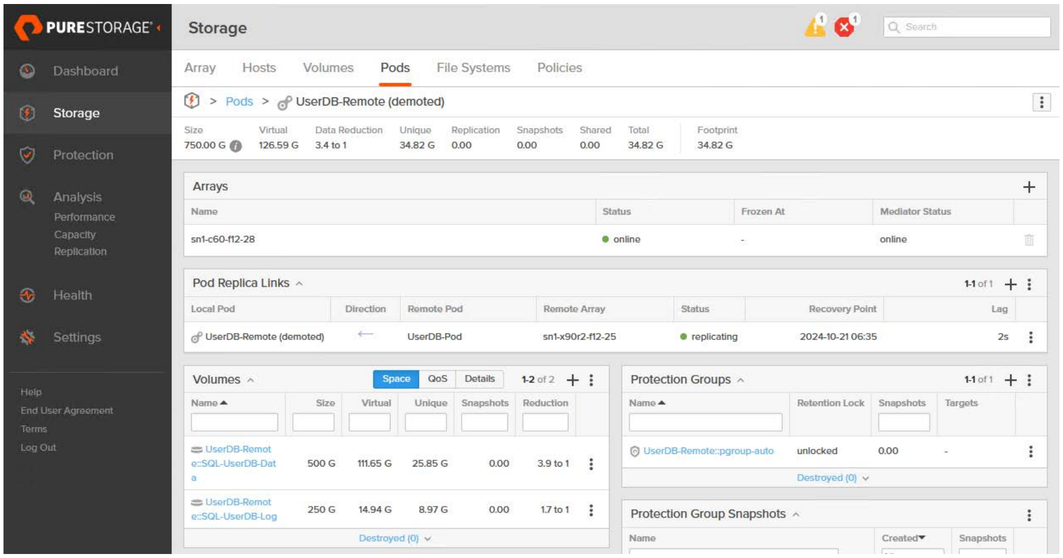
Destroyed (0)

Protection Group Snapshots

Name	Created▼	Snapshots

FIGURE 16 Displaying the state of the production/primary array pod.

2. In the **Storage** section on the target/disaster recovery array, click the **Pods** tab, and then observe the state of the **Pod Replica Links**, which will show as **demoted**.



> Pods > UserDB-Remote (demoted)

Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footprint
750.00 G	126.59 G	3.4 to 1	34.82 G	0.00	0.00	0.00	34.82 G	34.82 G

Arrays

Name	Status	Frozen At	Mediator Status
sn1-c60-f12-28	online	-	online

Pod Replica Links 1-1 of 1

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Remote (demoted)	←	UserDB-Pod	sn1-x90r2-f12-25	replicating	2024-10-21 06:35	2s

Volumes 1-2 of 2

Name	Size	Virtual	Unique	Snapshots	Reduction
UserDB-Remote::SQL-UserDB-Data	500 G	111.65 G	25.85 G	0.00	3.9 to 1
UserDB-Remote::SQL-UserDB-Log	250 G	14.94 G	8.97 G	0.00	1.7 to 1

Protection Groups 1-1 of 1

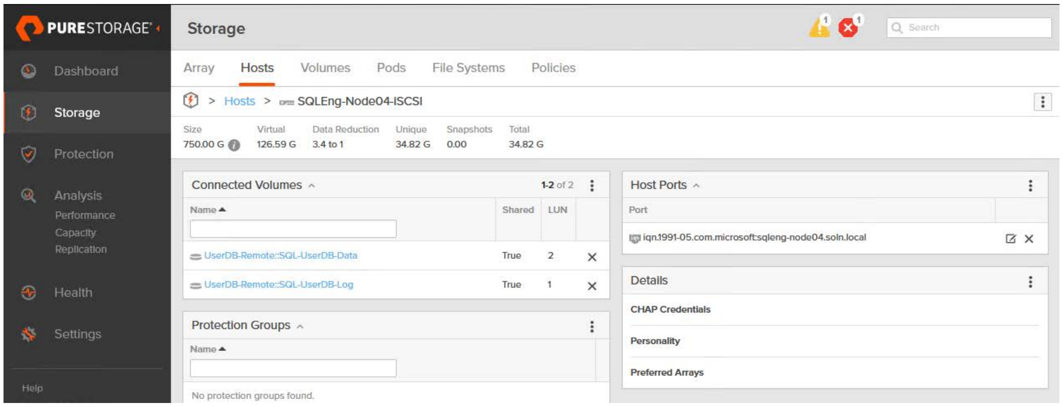
Name	Retention Lock	Snapshots	Targets
UserDB-Remote::pgroup-auto	unlocked	0.00	-

Protection Group Snapshots

Name	Created	Snapshots

FIGURE 17 Displaying the state of the target/disaster recovery array pod.

3. On the target array, attach the pod's volumes to a specific host. This can be done on the **Hosts** tab for a specific host.



> Hosts > SQLEng-Node04-iSCSI

Size	Virtual	Data Reduction	Unique	Snapshots	Total
750.00 G	126.59 G	3.4 to 1	34.82 G	0.00	34.82 G

Connected Volumes 1-2 of 2

Name	Shared	LUN
UserDB-Remote::SQL-UserDB-Data	True	2
UserDB-Remote::SQL-UserDB-Log	True	1

Host Ports

Port
iqn.1991-05.com.microsoft:sqleng-node04.soln.local

Details

CHAP Credentials

Personality

Preferred Arrays

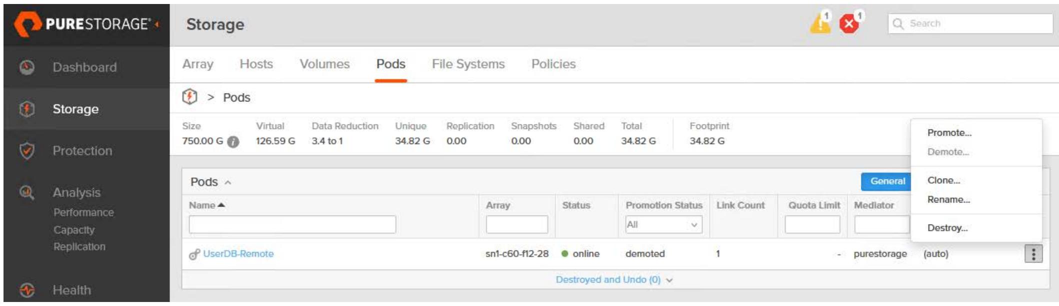
Protection Groups

Name

No protection groups found.

FIGURE 18 Attaching the pod's volumes to a host.

4. When ready to perform the failover, navigate to the specific pod in **Pods** under **Storage** on the target/disaster recovery array, click the ellipsis, and then select **Promote**.



> Pods

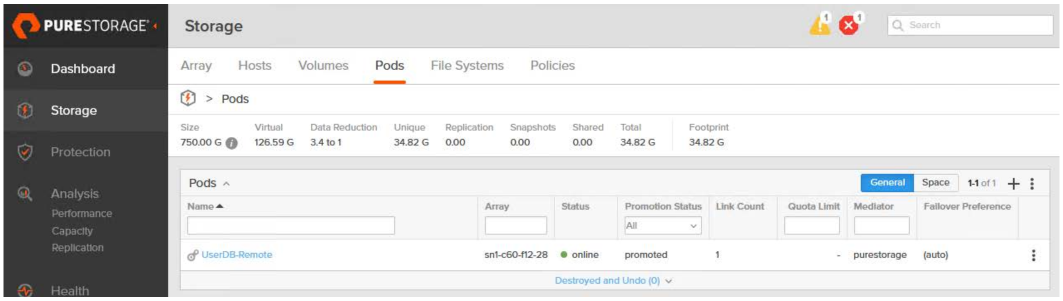
Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footprint
750.00 G	126.59 G	3.4 to 1	34.82 G	0.00	0.00	0.00	34.82 G	34.82 G

Pods

Name	Array	Status	Promotion Status	Link Count	Quota Limit	Mediator	
			All				
UserDB-Remote	sn1-c60-f12-28	online	demoted	1	-	purestorage	(auto)

FIGURE 19 Promoting the target/disaster recovery array pod.

5. When the pod has been promoted, it will have a promotion status of **promoted**.



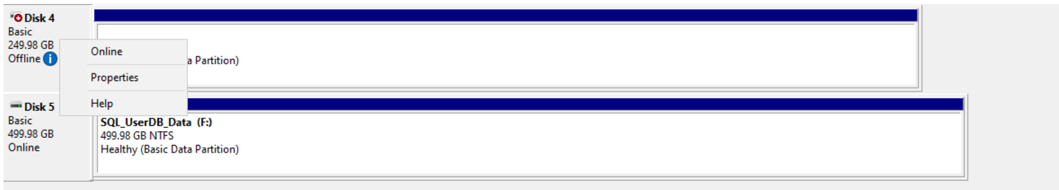
Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footprint
750.00 G	126.59 G	3.4 to 1	34.82 G	0.00	0.00	0.00	34.82 G	34.82 G

Pods

Name	Array	Status	Promotion Status	Link Count	Quota Limit	Mediator	Failover Preference
UserDB-Remote	sn1-c60-f12-28	● online	promoted	1	-	purestorage	(auto)

FIGURE 20 The promoted pod on the target/disaster recovery array.

6. The disks can be set online in Windows Disk Manager.



Disk 4
Basic
249.98 GB
Offline

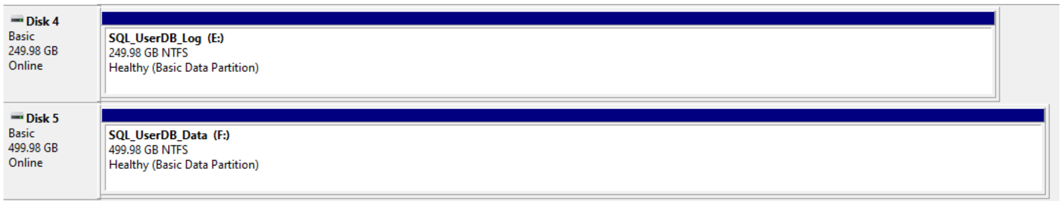
a Partition)

Disk 5
Basic
499.98 GB
Online

SQL_UserDB_Data (F:)
499.98 GB NTFS
Healthy (Basic Data Partition)

FIGURE 21 Bringing the disks online in Windows Disk Manager.

7. Once all of the disks are online, they might require drive letters or mount points to be assigned. In the example shown here, the drive letters have persisted between hosts after failover. In the event that the disks come online in a read-only state, set them to read/write prior to the next step.



Disk 4
Basic
249.98 GB
Online

SQL_UserDB_Log (E:)
249.98 GB NTFS
Healthy (Basic Data Partition)

Disk 5
Basic
499.98 GB
Online

SQL_UserDB_Data (F:)
499.98 GB NTFS
Healthy (Basic Data Partition)

FIGURE 22 Volumes attached as drives on the target/disaster recovery SQL Server host.

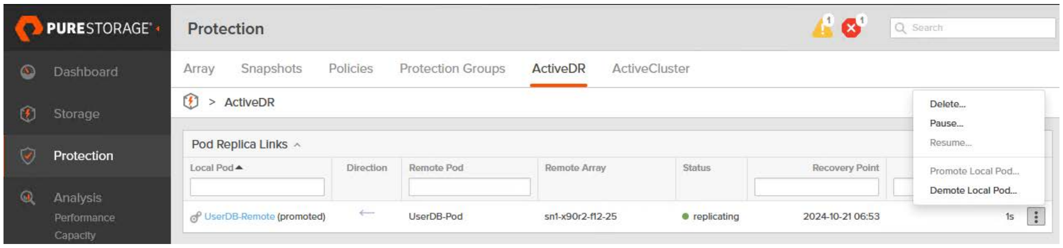
8. The database can be attached to the instance by executing the **CREATE DATABASE T-SQL** command with the **FOR ATTACH** property and the appropriate paths to each data file.

```
CREATE DATABASE [UserDB] ON
( FILENAME = 'E:\SQL\UserDB.mdf' ),
( FILENAME = 'F:\SQL\UserDB_log.ldf' )
 FOR ATTACH
```

The failed-over user database is now available for use. Any data changes made in the database will not be replicated back to the primary pod.

Failback from a Planned Non-disruptive Failover

1. To rollback from a non-disruptive failover, navigate to the ActiveDR pod on the target/disaster recovery array, and then select **Demote Local Pod.** This will apply all changes from the production/primary instance and then set the pod as read-only.

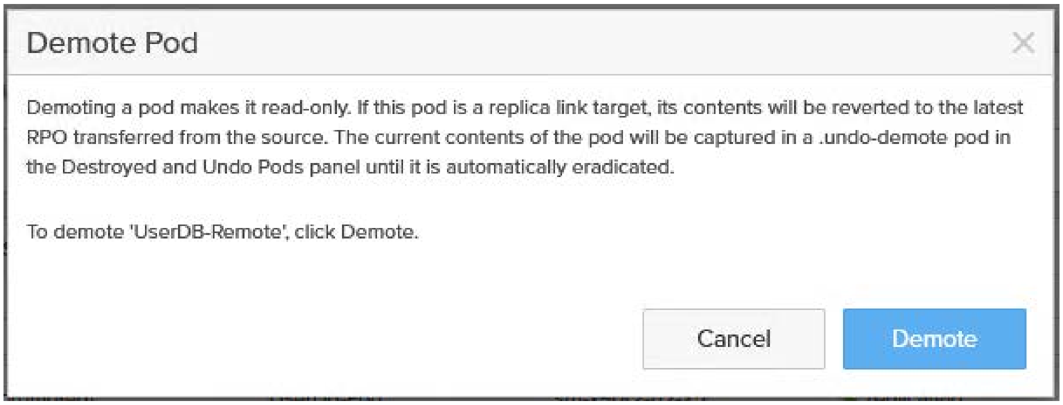


Pod Replica Links

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	
UserDB-Remote (promoted)	←	UserDB-Pod	sn1-x90r2-f12-25	● replicating	2024-10-21 06:53	1s

FIGURE 23 Demoting the local pod.

2. When selecting **Demote Local Pod**, a prompt to confirm demotion appears. Select **Demote** when ready.



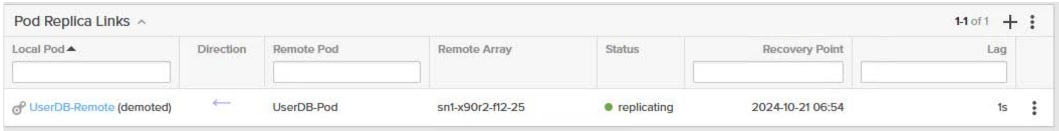
Demote Pod

Demoting a pod makes it read-only. If this pod is a replica link target, its contents will be reverted to the latest RPO transferred from the source. The current contents of the pod will be captured in a .undo-demote pod in the Destroyed and Undo Pods panel until it is automatically eradicated.

To demote 'UserDB-Remote', click Demote.

FIGURE 24 Confirming the pod demotion.

3. Once the pod has been demoted, it will show as **demoted** in the Pod Replica Links pane.



Pod Replica Links

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Remote (demoted)	←	UserDB-Pod	sn1-x90r2-f12-25	● replicating	2024-10-21 06:54	1s

FIGURE 25 Displaying the pod status.

Attach the Database to the Remote Site's SQL Server Instance Using SQL Server Management Studio

1. In SQL Server Management Studio, connect to the remote site's SQL Server instance.
2. Right-click the **Databases** object, and then select **Attach**.
3. Select **Add**, and then navigate to the drive and folder that contain the database MDF file.
4. Select all database data and log files, and then select **OK**.
5. Select **Add**, and then navigate to the drive and folder that contain the log files.
6. Select the database LDF file, and then select **OK**.
7. Select **OK** to attach and bring the database online.

The remote site's SQL Server instance is now able to service application requests. Verify connectivity to the database at the remote site and that applications that use the database perform as expected. Once connectivity to the database has been verified and applications perform as expected, a failback can be performed.

Performing a Full Failover for Disaster Recovery Testing Only

This section is intended for situations where a full manual failover needs to be executed. Once ActiveDR is configured, replicated data can be used at the remote site by performing a manual failover. Note that a manual failover requires SQL Server downtime, and without careful planning and coordination, it can result in business disruption. For testing a failover, see the Non-disruptive Failover Testing section of this document.

A planned failover consists of the following general steps:

1. Take the primary site's SQL Server database offline.
2. Demote the primary site's pod.
3. Allow for data replication to complete.
4. Promote the secondary site's pod.
5. Bring the secondary site's database online in SQL Server at the secondary site.

The Pure Storage user interface and various SQL Server and Windows Server tools can be used to accomplish these steps.

Take the SQL Server Database Offline in SQL Server Management Studio

1. In SQL Server Management Studio, connect to the SQL Server instance whose data is being replicated to the secondary site.
2. In **Object Explorer**, navigate to the database that will be taken offline.
3. Right-click the database, select **Tasks**, and then select **Take Offline**.

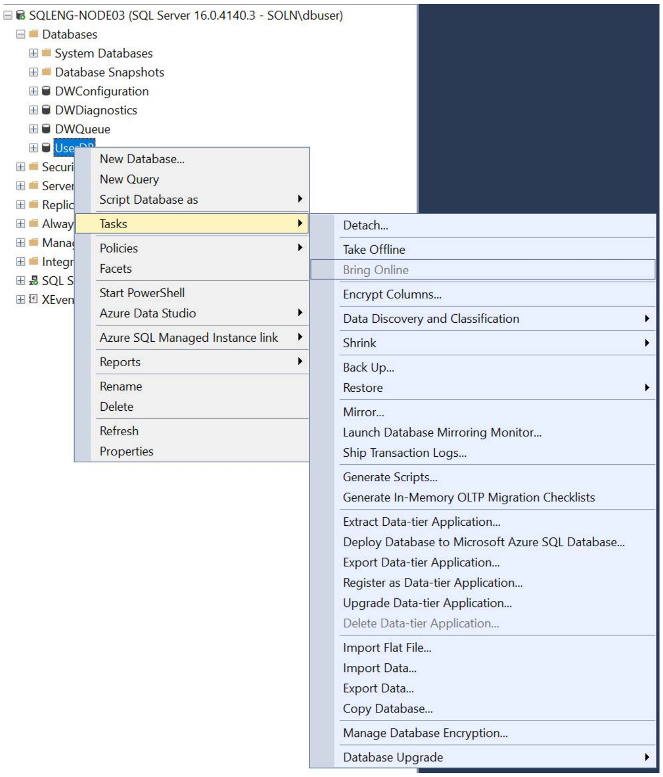


FIGURE 26 Taking the database offline.

4. In the **Take Database Offline** dialog, verify that the database that is to be taken offline is highlighted, and then click **OK**. The database is taken offline.

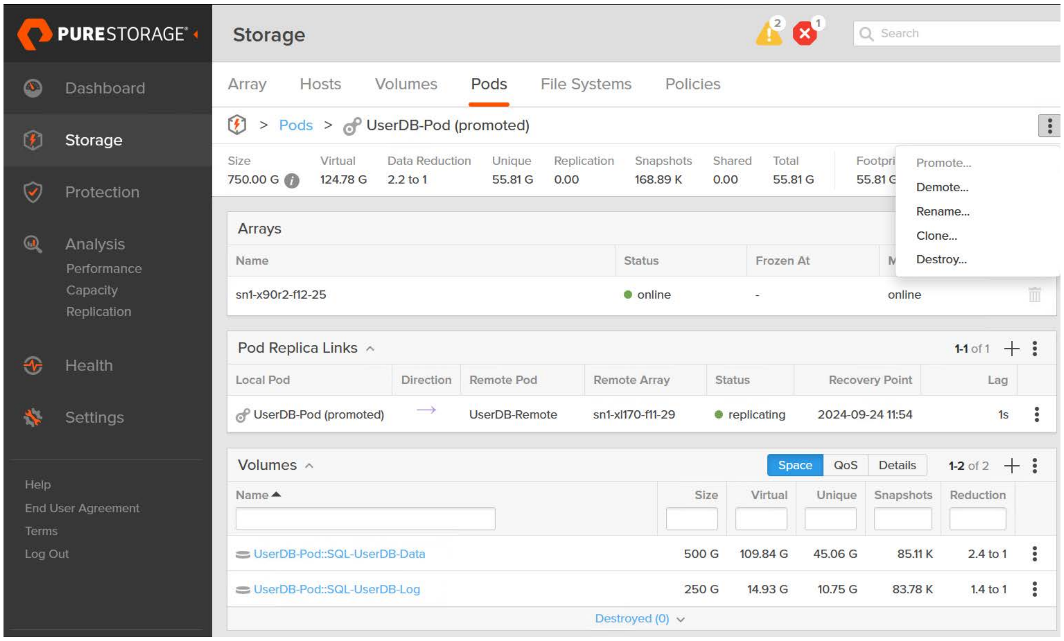
Note: You can also use the following SQL statement in SQL Server Management Studio to take the database offline:

```
ALTER DATABASE [Database Name] OFFLINE;
```

Once the SQL Server database is taken offline, the primary site's pod can be demoted.

Demote the Pod at the Primary Site in the Pure Storage User Interface

1. Verify that the SQL Server database is offline in SQL Server Management Studio.
2. From the primary site's Pure Storage user interface, select **Storage** in the navigation pane, and then select the **Pods** tab.
3. Navigate to the primary site's pod, click the ellipsis, and then select **Demote**.



Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footpri
750.00 G	124.78 G	2.2 to 1	55.81 G	0.00	168.89 K	0.00	55.81 G	55.81 G

Arrays

Name	Status	Frozen At	M
sn1-x90r2-f12-25	● online	-	online

Pod Replica Links 1-1 of 1

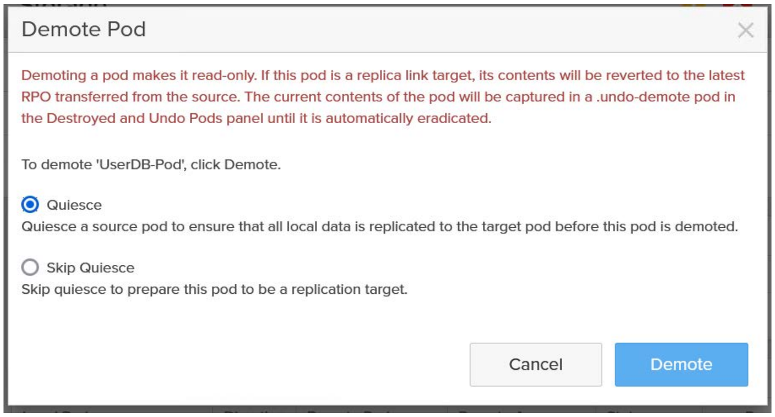
Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Pod (promoted)	→	UserDB-Remote	sn1-xl170-f11-29	● replicating	2024-09-24 11:54	1s

Volumes 1-2 of 2

Name ▲	Size	Virtual	Unique	Snapshots	Reduction
UserDB-Pod::SQL-UserDB-Data	500 G	109.84 G	45.06 G	85.11 K	2.4 to 1
UserDB-Pod::SQL-UserDB-Log	250 G	14.93 G	10.75 G	83.78 K	1.4 to 1

FIGURE 27 Demoting the primary site's pod.

4. From the **Demote** dialog, select **Quiesce**, and then click **Demote**. The primary site's pod is demoted and put in a read-only state.



Demote Pod

Demoting a pod makes it read-only. If this pod is a replica link target, its contents will be reverted to the latest RPO transferred from the source. The current contents of the pod will be captured in a .undo-demote pod in the Destroyed and Undo Pods panel until it is automatically eradicated.

To demote 'UserDB-Pod', click Demote.

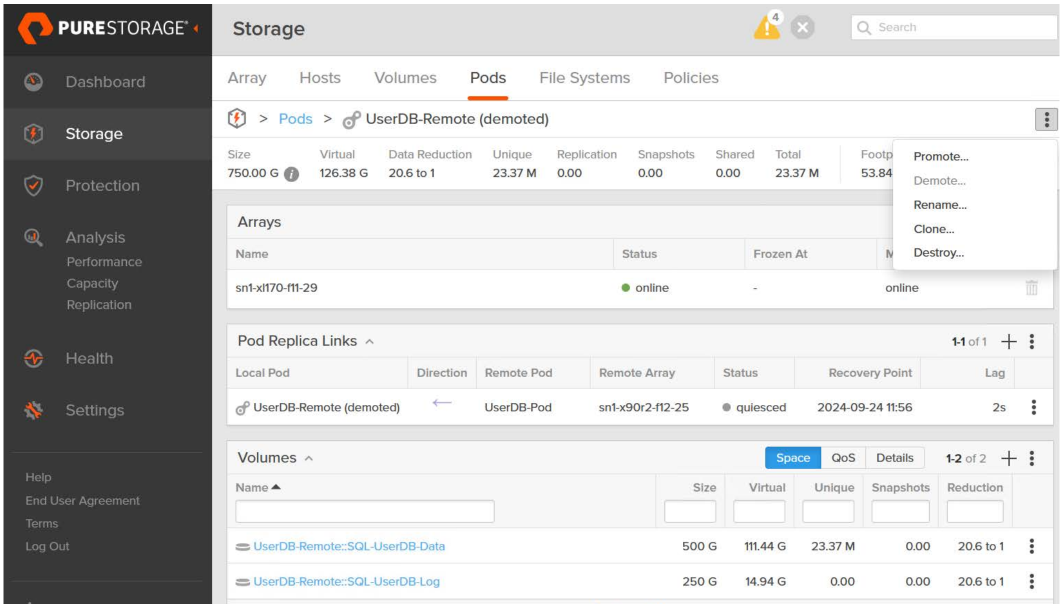
● Quiesce
Quiesce a source pod to ensure that all local data is replicated to the target pod before this pod is demoted.

○ Skip Quiesce
Skip quiesce to prepare this pod to be a replication target.

FIGURE 28 Confirming the pod's demotion.

Promote the Pod at the Secondary Site in the Pure Storage User Interface

1. Navigate to the secondary site's Pure Storage user interface, select **Storage** in the navigation pane, and then select the **Pods** tab.
2. Navigate to the secondary site's pod, click the ellipsis, and then select **Promote**.



> Pods > UserDB-Remote (demoted)

Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footp
750.00 G	126.38 G	20.6 to 1	23.37 M	0.00	0.00	0.00	23.37 M	53.84

Arrays

Name	Status	Frozen At	M
sn1-xl170-f11-29	online	-	online

Pod Replica Links 1-1 of 1

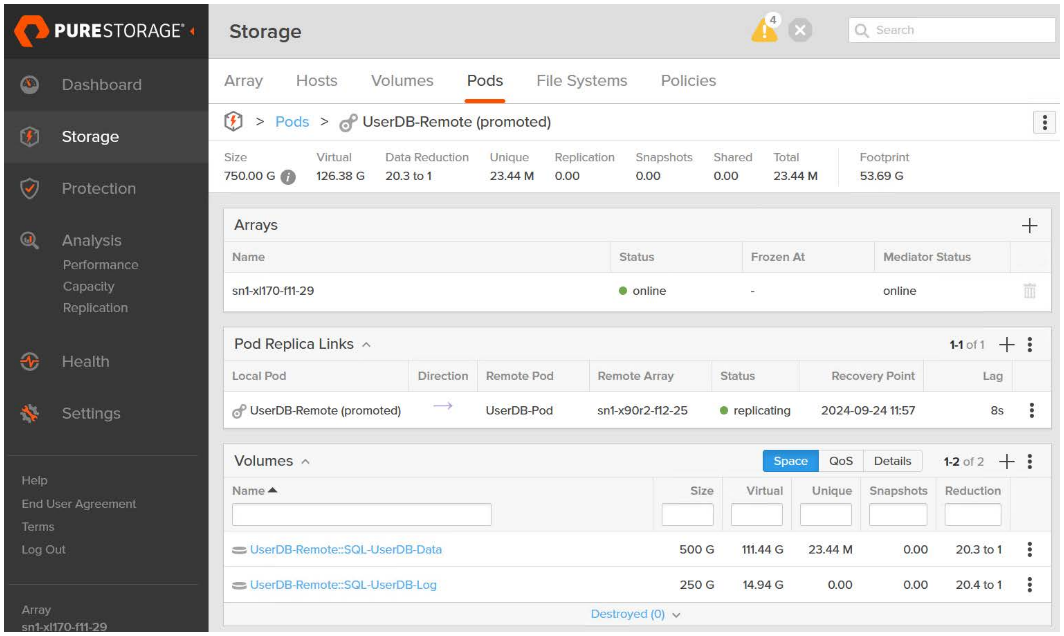
Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Remote (demoted)	←	UserDB-Pod	sn1-x90r2-f12-25	quiesced	2024-09-24 11:56	2s

Volumes 1-2 of 2

Name ▲	Size	Virtual	Unique	Snapshots	Reduction
UserDB-Remote::SQL-UserDB-Data	500 G	111.44 G	23.37 M	0.00	20.6 to 1
UserDB-Remote::SQL-UserDB-Log	250 G	14.94 G	0.00	0.00	20.6 to 1

FIGURE 29 Promoting the secondary site's pod.

3. From the **Promote** dialog, select **Promote**. The secondary site's pod is promoted and put into a read/write state.



> Pods > UserDB-Remote (promoted)

Size	Virtual	Data Reduction	Unique	Replication	Snapshots	Shared	Total	Footprint
750.00 G	126.38 G	20.3 to 1	23.44 M	0.00	0.00	0.00	23.44 M	53.69 G

Arrays

Name	Status	Frozen At	Mediator Status
sn1-xl170-f11-29	online	-	online

Pod Replica Links 1-1 of 1

Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag
UserDB-Remote (promoted)	→	UserDB-Pod	sn1-x90r2-f12-25	replicating	2024-09-24 11:57	8s

Volumes 1-2 of 2

Name ▲	Size	Virtual	Unique	Snapshots	Reduction
UserDB-Remote::SQL-UserDB-Data	500 G	111.44 G	23.44 M	0.00	20.3 to 1
UserDB-Remote::SQL-UserDB-Log	250 G	14.94 G	0.00	0.00	20.4 to 1

FIGURE 30 Confirming the pod's promotion.

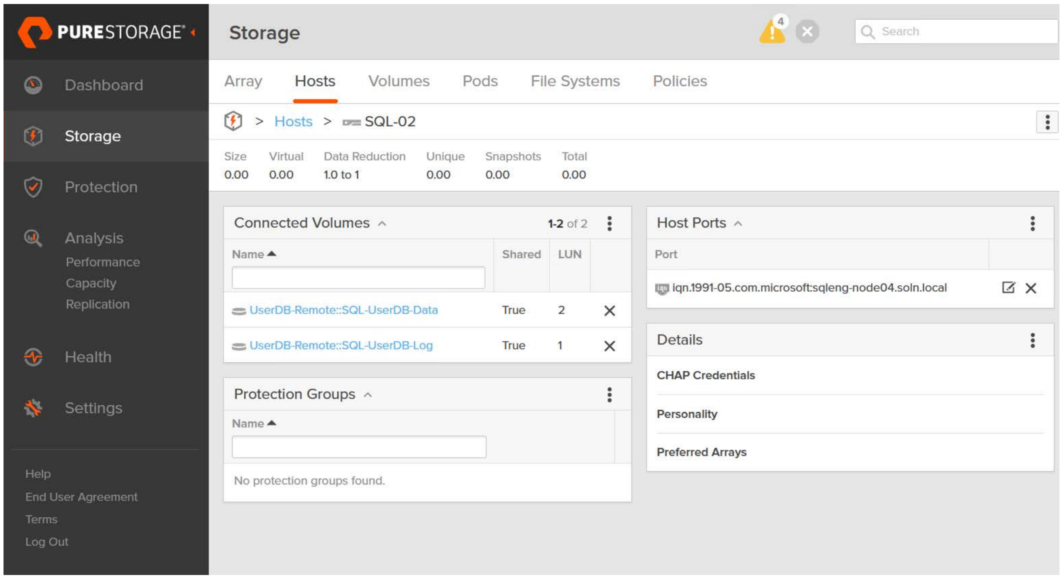
With the secondary site's pod promoted and in a read/write state, the volumes can be attached to the remote site's Windows Server instance.

Note: The remote site's Windows Server instance should already be attached to the target Windows instance and offline.

Bring the Disks Online Using the Windows Server Disk Management Application

1. From the remote site's Pure Storage user interface, select **Storage** in the navigation pane, and then select the **Hosts** tab.

2. Verify that the remote site's Windows Server environment where SQL Server is installed appears in the **Hosts** group.



Size	Virtual	Data Reduction	Unique	Snapshots	Total
0.00	0.00	1.0 to 1	0.00	0.00	0.00

Connected Volumes 1-2 of 2

Name	Shared	LUN
UserDB-Remote::SQL-UserDB-Data	True	2
UserDB-Remote::SQL-UserDB-Log	True	1

Protection Groups

Name

No protection groups found.

Host Ports

Port

iqn.1991-05.com.microsoft:sqleng-node04.soln.local

Details

CHAP Credentials

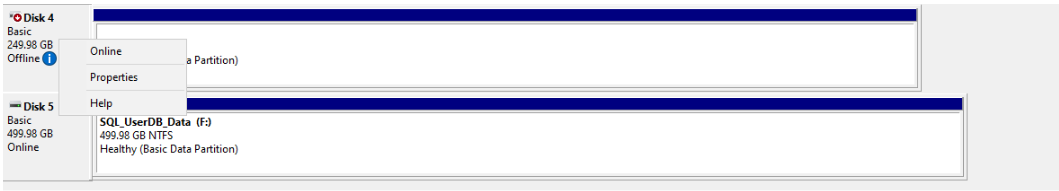
Personality

Preferred Arrays

FIGURE 31 Verifying the SQL Server host.

3. On the remote site's Windows Server instance where SQL Server is installed, open the **Disk Management** application. The database file and log file Pure Storage volumes appear as offline disks.

4. Right-click each offline disk, and then select **Online.**



Disk 4
Basic
249.98 GB
Offline

a Partition)

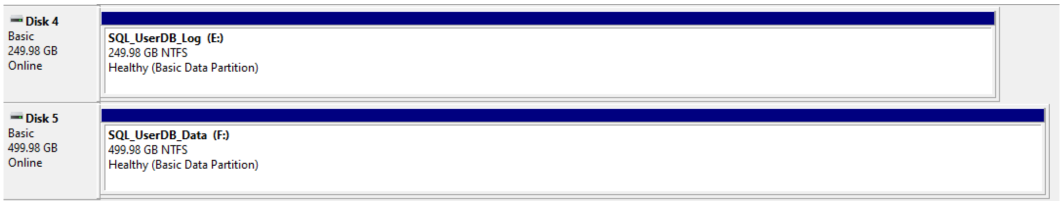
Disk 5
Basic
499.98 GB
Online

SQL_UserDB_Data (F:)
499.98 GB NTFS
Healthy (Basic Data Partition)

FIGURE 32 Bringing the disks online in Windows Disk Manager.

5. If the disk does not have a drive letter assigned, right-click the disk, and then select **Change Drive Letter and Paths**.

6. Assign the same drive letter as was used before the failover.



Disk 4
Basic
249.98 GB
Online

SQL_UserDB_Log (E:)
249.98 GB NTFS
Healthy (Basic Data Partition)

Disk 5
Basic
499.98 GB
Online

SQL_UserDB_Data (F:)
499.98 GB NTFS
Healthy (Basic Data Partition)

FIGURE 33 Volumes attached as drives on the SQL Server host.

7. If the disks come online in a read-only state, right-click each disk, and then set them to read/write.

Attach the Database to the Remote Site's SQL Server Instance Using SQL Server Management Studio

1. In SQL Server Management Studio, connect to the remote site's SQL Server instance.
2. Right-click the **Databases** object, and then select **Attach**.
3. Select **Add**, and then navigate to the drive and folder that contain the database MDF file.
4. Select all database data and log files, and then select **OK**.
5. Select **Add**, and then navigate to the drive and folder that contain the log files.
6. Select the database LDF file, and then select **OK**.
7. Select **OK** to attach and bring the database online.

The remote site's SQL Server instance is now able to service application requests. Verify connectivity to the database at the remote site and that applications that use the database perform as expected. Once connectivity to the database has been verified and applications perform as expected, a failback can be performed.

Conclusion

ActiveDR gives database and storage administrators a robust disaster recovery tool for SQL Server environments.
By enabling continuous replication with near-zero recovery point objectives, ActiveDR ensures protection against data loss during hardware failures, ransomware attacks, or natural disasters, while allowing for live failovers and failbacks with minimal complexity to enhance business continuity.

ActiveDR also supports non-disruptive failover testing, an essential function for verifying disaster recovery plans without impacting production workloads. This capability allows organizations to validate recovery strategies and ensure they meet recovery time objectives and recovery point objectives as outlined in their service-level agreements. Regular failover testing enables administrators to identify potential issues, fine-tune disaster recovery procedures, and ensure rapid recovery during critical events.

This solution simplifies disaster recovery workflows, reduces downtime risks, and helps organizations maintain the performance and availability of SQL Server databases during unforeseen disruptions.

For more information, visit purestorage.com/microsoft or try the functionality in a test drive at www.purestorage.com/products/unified-block-file-storage/flasharray-x/test-drive.html.