

TECHNICAL WHITE PAPER

# Intra-cloud Disaster Recovery for Microsoft SQL Server

Disaster recovery for Microsoft SQL Server databases between regions and availability zones in Microsoft Azure.

# Contents

- Introduction .....3**
- How to Use This Paper .....3**
- Pure Cloud Block Store.....3**
- Solution Architecture.....5**
  - Asynchronous Replication ..... 5
  - ActiveDR ..... 6
  - Combining ActiveDR and Asynchronous Replication for Multiple Recovery Point Objectives ..... 7
  - Public Cloud Infrastructure and Software Environment ..... 7
- Implementation Guides .....9**
  - Preparing Storage ..... 9
  - Asynchronous Replication for a Five-minute Recovery Point Objective ..... 11
  - ActiveDR for Near-zero Recovery Point Objectives ..... 20
  - Seeding Replicas Across Regions in an Always On Availability group ..... 27
- Ransomware Remediation with SafeMode SnapShots ..... 34**
- Conclusion ..... 35**



## Introduction

[Microsoft SQL Server](#) (SQL Server) is a relational database management system developed, distributed, and supported by [Microsoft Corporation](#). SQL Server is available in several editions, each of which provides different functional capabilities such as scale, availability, and tooling. Deployment is highly versatile with support for multiple operating systems, deployment platforms (bare metal and virtualized), and many service offerings.

According to [DB-Engines](#), as of August 2022 Microsoft SQL Server is the [third most popular database](#) in the world. As many business operations are underpinned by these databases it is critical that there are suitable disaster recovery capabilities in place in the event of data, system, or site failure.

SQL Server is provided with several built-in mechanisms for recovery in the event of a data loss failure; however, even with the most advanced of capabilities sometimes core issues such as recovery time objectives (RTO) and recovery point objectives (RPO) are still difficult to achieve. With the rich data services from Pure Cloud Block Store™ these core issues can be mitigated or eliminated entirely depending on what functionality is utilized.

This technical white paper is intended to be used as a how-to and best practice guide when implementing disaster recovery capabilities with Pure Cloud Block Store in Microsoft Azure. The target audience for this document includes, but is not limited to, systems architects, database administrators, storage administrators and IT professionals.

---

## How to Use This Paper

This white paper focuses on implementing disaster recovery capabilities with Microsoft SQL Server deployed on Microsoft Azure Infrastructure-as-a-Service systems when using Pure Cloud Block Store as persistent database storage. Data services, such as asynchronous replication and ActiveDR, will be covered to address different recovery point objectives. How to implement and attain the best outcome for each tool will be provided in the form of outcome guidance.

The concepts, architectures, and methodologies set out in this guide can also be used for any Pure Storage® FlashArray™ model.

## Pure Cloud Block Store

[Pure Cloud Block Store](#) provides seamless data mobility across on-premises and cloud environments with a consistent experience regardless of where your 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 Cloud Block Store is available in the [Amazon Web Services Marketplace](#) and the [Microsoft Azure Marketplace](#).



# Pure Cloud Block Store

Software Defined Block Storage, Delivered Natively in the Cloud



Figure 1. Pure Cloud Block Store

Pure Cloud Block Store provides several benefits for storage in the public cloud:

- Simplicity:** By using Azure managed applications, customers easily deploy a Pure Cloud Block Store managed application into their desired virtual network in a few minutes. There are no RAID pools or manual data tiering required. Once iSCSI connections between Azure VM hosts and the application managed by Pure Cloud Block Store are established, customers can easily create and mount Pure Cloud Block Store volumes to the desired Azure VMs in seconds.
- Higher availability and efficiency:** Due to its unique architecture, Pure Cloud Block Store provides higher availability for critical application access while reducing the amount of storage capacity consumed with always-on compression and deduplication.
- Hybrid mobility and rich protection capabilities:** Using the rich data services inside Pure Cloud Block Store, a business can move data efficiently between on-premises FlashArray systems and public cloud infrastructure. Data services can also assist in providing mechanisms for higher availability, disaster recovery, and data protection workflows.
- Consistent APIs to drive automation workflows:** All management operations can be codified using the REST API included in FlashArray and Pure Cloud Block Store. When moving from FlashArray on-premises to public cloud, developers do not need to redesign their applications to use different APIs. For language specific automation, Software development kits for [PowerShell](#), [Python](#), and [Ansible](#) are available.

Pure Cloud Block Store is available in two versions: //V10 and //V20. Each version provides different capacity and performance capabilities.

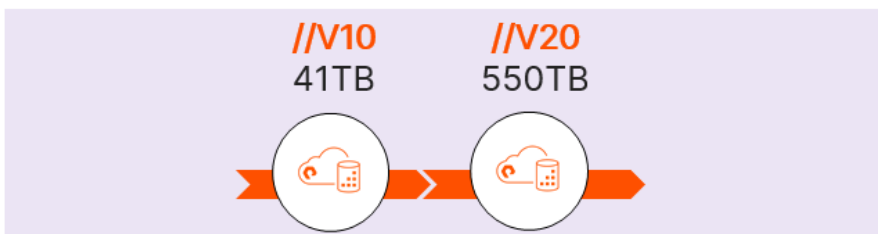


Figure 2. The V10 and V20 versions of Pure Cloud Block Store. The capacity listed for each model is effective capacity with a 4:1 data reduction rate.

Pure Cloud Block Store systems are powered by [Purity for FlashArray](#). Purity delivers rich, enterprise level data services that ensures data is stored in the most secure and efficient way while providing additional functionality to extend storage capabilities.

## Solution Architecture

The purpose of this technical white paper is to highlight how disaster recovery capabilities with Pure Cloud Block Store can be implemented for Microsoft SQL Server. Storage replication or software replication capabilities can be used to implement the use case. The storage replication data services in Pure Cloud Block Store which will be used are Asynchronous Replication and ActiveDR™ while the software replication in SQL Server to be used will be [Always On availability groups](#).

This solution architecture provides several benefits depending on the deployment type. For all scenarios utilizing storage replication with Pure Cloud Block Store you can gain the following benefits:

- A reduction in cost for cross availability zone/region traffic and capacity consumption
- More efficient data movement resulting in more consistent recovery point objectives

For software replication with always on availability groups across regions in Microsoft SQL Server 2022, the following benefits can be realized:

- Volume snapshots and asynchronous replication can be used to ensure that after failover high availability can be regained rapidly.
- Seeding and reseeding times can be drastically minimized.

## Asynchronous Replication

Asynchronous replication is the copying of volume snapshots from one FlashArray or Pure Cloud Block Store instance to another. Volume snapshots are immutable, point-in-time images of the contents of one or more volumes which can be created individually or as a part of a protection group. Protection Groups are management views for the snapshot operations of one or more volumes. They allow for functions such as scheduling, retention, and replication target management.

All systems powered by Purity incorporate the same fundamental engineering for its volume snapshot technology, making it suitable in any business scenario requiring recovery point objectives as low as five (5) minutes. Volume snapshots with Purity are unique for the following reasons:

- **Can be instantly created or recovered:** Volume snapshots are created using the redirect-on-write technique, meaning that only the blocks in a volume that have changed since the most recent snapshot are saved.
- **Are storage space efficient:** Volume snapshots are always thin-provisioned, deduplicated, and compressed, and require no snapshot capacity reservations. All volumes (and by extension snapshots) are thin-provisioned and a part of a global deduplication namespace, further reducing the storage space consumed.
- **Are portable and transport efficient:** Volume snapshots are portable, allowing them to be transferred to one or more FlashArray or Pure Cloud Block Store instances. All transfers are done using space efficient copies accompanied by the volume metadata allowing for any offloaded snapshot to be recovered to any FlashArray or Pure Cloud Block Store instance.



- **Can be used for multiple use cases:**– Volume snapshots can be used for many different use cases to make business operations more efficient and effective. More information on these use cases are found in the white paper “[Using Databases with FlashArray Volume Snapshots](#)”.

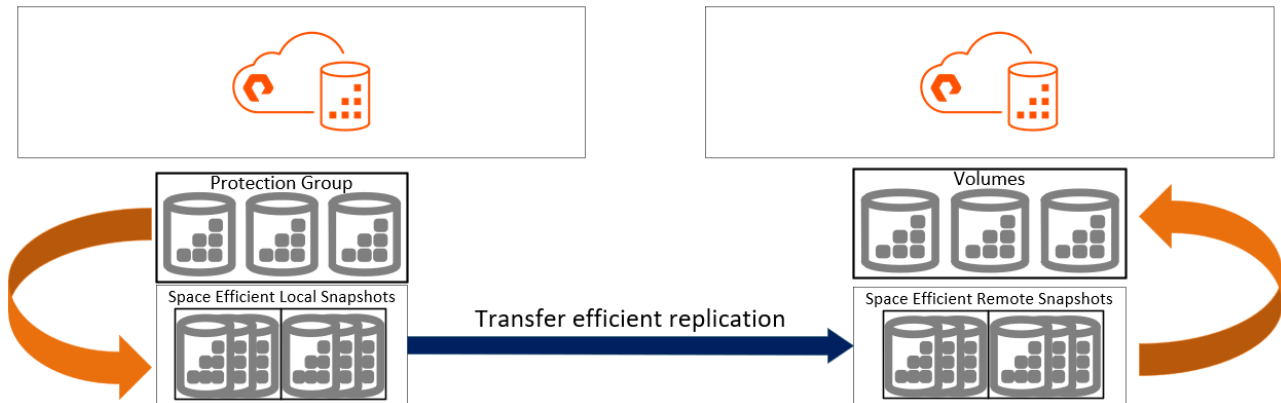


Figure 3. Asynchronous replication of volume snapshots from a source to a target

### ActiveDR

Pure Storage [ActiveDR](#) provides continuously active replication capabilities for block storage volumes from one FlashArray or Pure Cloud Block Store instance to another, with no impact on front-end application performance.

Synchronous replication technologies, such as [ActiveCluster™](#), provide higher availability at the cost of requiring network latency between the two targets to be as low as possible. ActiveDR does not require the target system to acknowledge that data has been received before allowing the source to continue. This eliminates the need for a low latency network and allows for continuous replication over larger distances.

The management of ActiveDR is vastly simplified making it suitable for implementation in any business scenario requiring a near zero (0) recovery point objective. Some management aspects of ActiveDR include:

- **POD replication:** Pods are management containers for volumes using ActiveCluster or ActiveDR. Pods provide a simple management construct to organize data volumes and associated settings into groups. Once pods are linked together on separate systems via a replica link, data in that pod automatically starts replicating.
- **Continuous change tracking:** Automatically manage changes without the need to provision or monitor journal devices.
- **Single-command failover:** ActiveDR makes it simple to implement, test and manage disaster recovery. This true disaster recovery testing ensures that any runbook or orchestration steps for the entire environment stay the same during a test or in an actual failover event to minimize risk.
- **Multi-direction replication:** Configure multiple pods in different directions between two FlashArray systems.



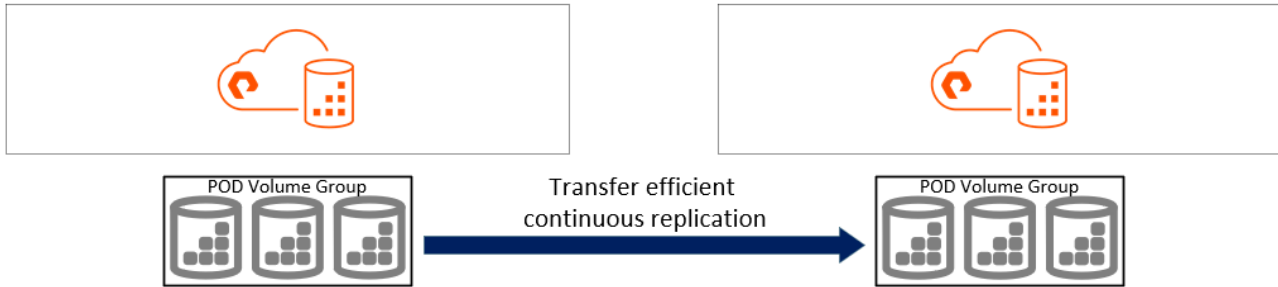


Figure 4. Continuous replication of volumes from a source to a target

### Combining ActiveDR and Asynchronous Replication for Multiple Recovery Point Objectives

ActiveDR and asynchronous replication can be combined to provide the best of both scenarios where near zero (0) and five (5) minute RPOs can happen simultaneously. This is achieved by adding a protection group with a snapshot-and-replicate schedule to an existing ActiveDR Pod.

### Public Cloud Infrastructure and Software Environment

The public cloud infrastructure used in this disaster recovery solution architecture is comprised of:

- **Two (2) Pure Cloud Block Store Instances.** Each is deployed into a different availability zone or region. This document will showcase the instances deployed into the same region but different availability zones.
- **Two (2) Azure Virtual Machines.** Each virtual machine is deployed into a different availability zone or region. This document will showcase the virtual machines deployed into the same region but different availability zones. The virtual machines should be deployed into the same region and availability zone as the corresponding Pure Cloud Block Store instances for the best performance and disaster tolerance. The Azure virtual machines run Windows Server 2022. Each virtual machine was connected to the corresponding Pure Cloud Block Store instance using iSCSI connectivity. Microsoft SQL Server was installed on both systems with the user database files residing on Pure Cloud Block Store volumes.
- **One or more virtual networks.** It is possible for all the infrastructure to exist on the same subnet on the same virtual network if within a single region. The minimum requirements are:
  - The Azure virtual machine can connect to the Pure Cloud Block Store instance in its availability zone/region.
  - Each Pure Cloud Block Store instance can communicate with the other in another availability zone or region over its replication ports.



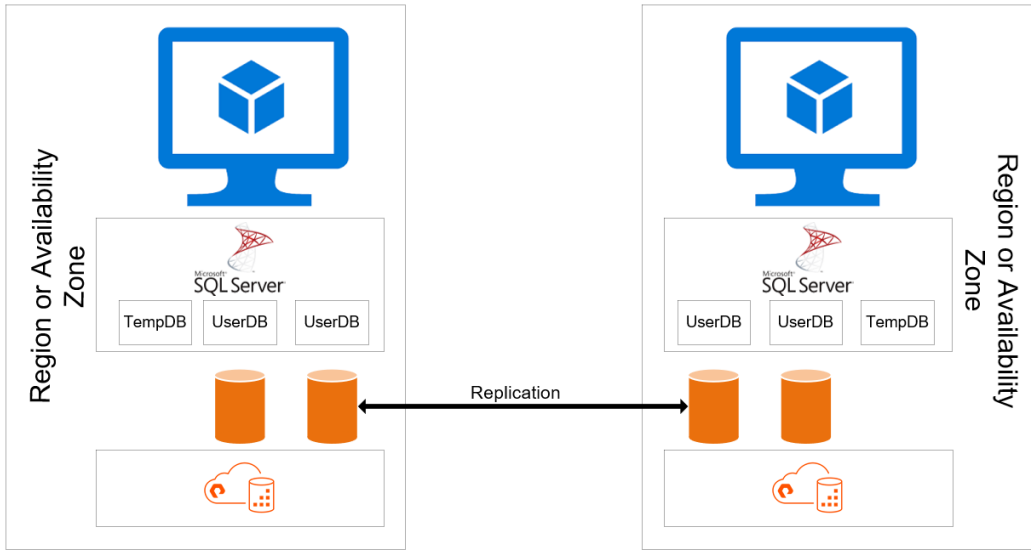


Figure 5. High-level diagram of public cloud infrastructure being used for storage replication

The software environment is made up of Microsoft SQL Server instances where user databases are created on dedicated block storage volumes from Pure Cloud Block Store. For details on volume placement and configuration, please see [this article](#) on the use of Pure Cloud Block Store with SQL Server.

For scenarios utilizing always on availability groups, the rapid seeding of replica databases is only possible in SQL Server 2022 for systems which remain online. It is possible to seed a replica database in prior versions of SQL Server. However, the SQL Server instance must be stopped prior to the volume being snapshotted and cloned.

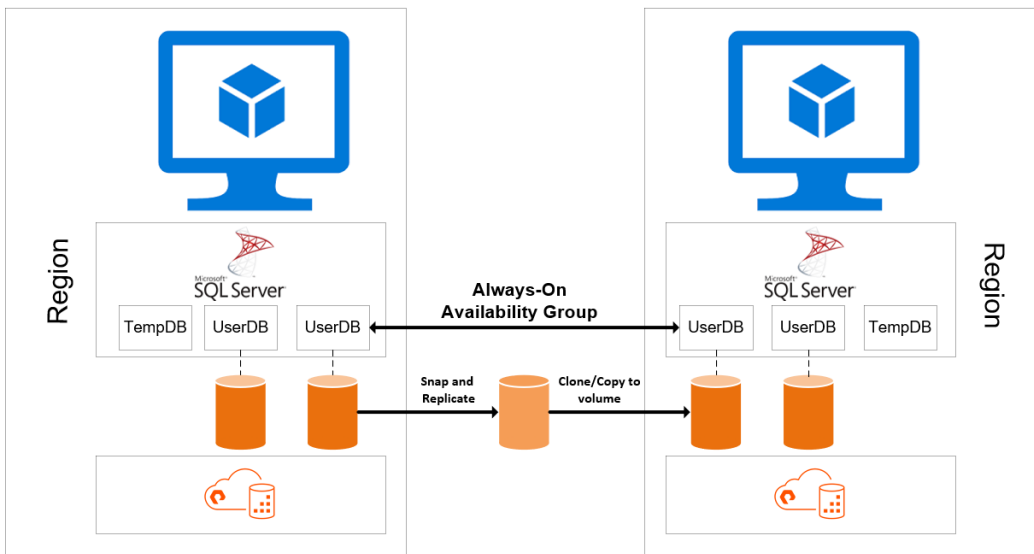


Figure 6. High-level diagram of public cloud infrastructure being used for storage and software replication





## Implementation Guides

The following implementations showcase how disaster recovery can be achieved with the following use cases:

- As low as five (5) minute recovery point objectives for SQL Server Databases with storage asynchronous replication.
- Near zero (0) recovery point objectives for SQL Server Databases with ActiveDR continuous replication.
- Combining storage asynchronous replication with SQL Server Always On availability groups across regions to achieve rapid replica seeding. The implementation steps provided for this use case is only possible with SQL Server 2022 and later.

Additional information and guidance on topics covered here on replication with Pure Cloud Block Store can be found in the [Replication User Guide for Pure Cloud Block Store on Azure](#). General information on implementing Pure Cloud Block Store can be found on the [Pure Storage support site](#).

The implementation of these use cases is not limited to SQL Server on Pure Cloud Block Store. SQL Server on FlashArray//C™, FlashArray//X™, and FlashArray//XL™ can benefit from them, as well.

For the purposes of these demonstrations, a single database (**UserDB**) is created in a primary SQL Server instance (**SQL-AZ1**) and populated with 100GB of data. The data and log files for the database are located on a single volume inside a Pure Cloud Block Store instance in the same region and availability zone (CBS-AZ1) as the primary SQL Server instance. The systems provisioned in another availability zone or region for disaster recovery capabilities are a SQL Server instance (SQL-AZ2) and a Pure Cloud Block Store instance (CBS-AZ2).

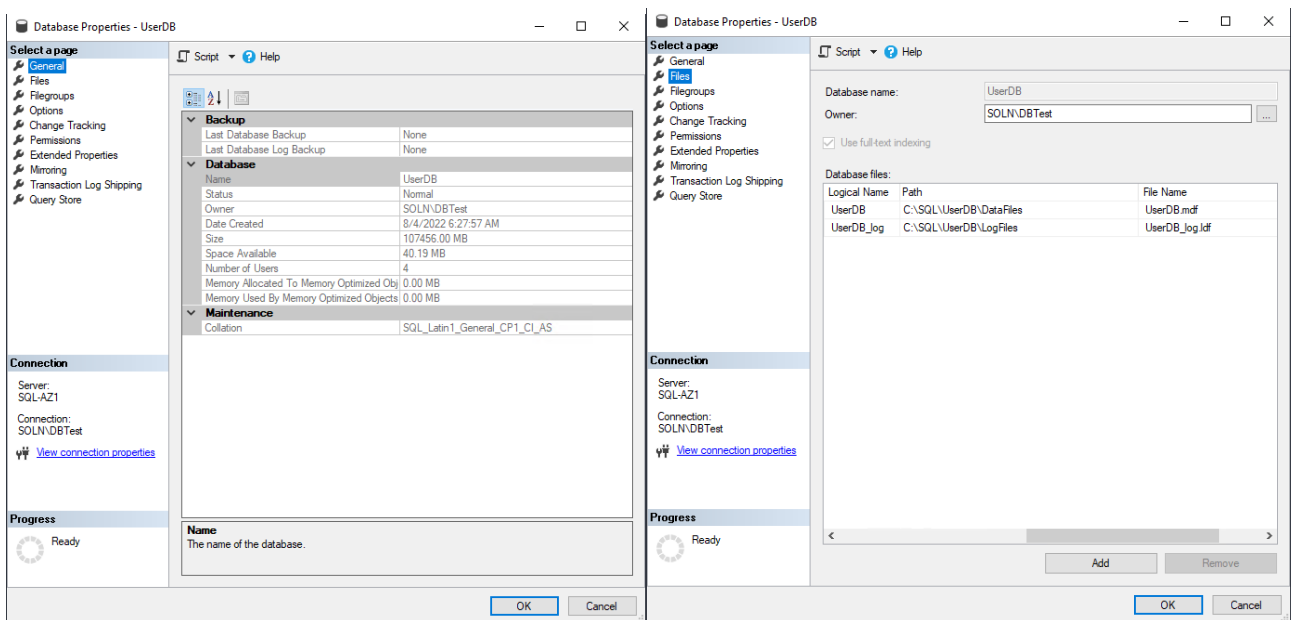


Figure 7. Properties dialog for the UserDB database used to illustrate each use case

## Preparing Storage

Prior to any steps being taken with asynchronous replication or ActiveDR, it is important that the two Pure Cloud Block Store instances are connected for asynchronous replication. To do this, the graphical user interface is used.



Prior to connecting the two Pure Cloud Block Store instances, the connection key from the intended target instance must be acquired. This can be found in the graphical user interface, navigating to the **Storage** area, and then finding the three ellipses next to the section for Array Connections. Selecting the three ellipses brings up a menu. Select the option **Get Connection Key**.

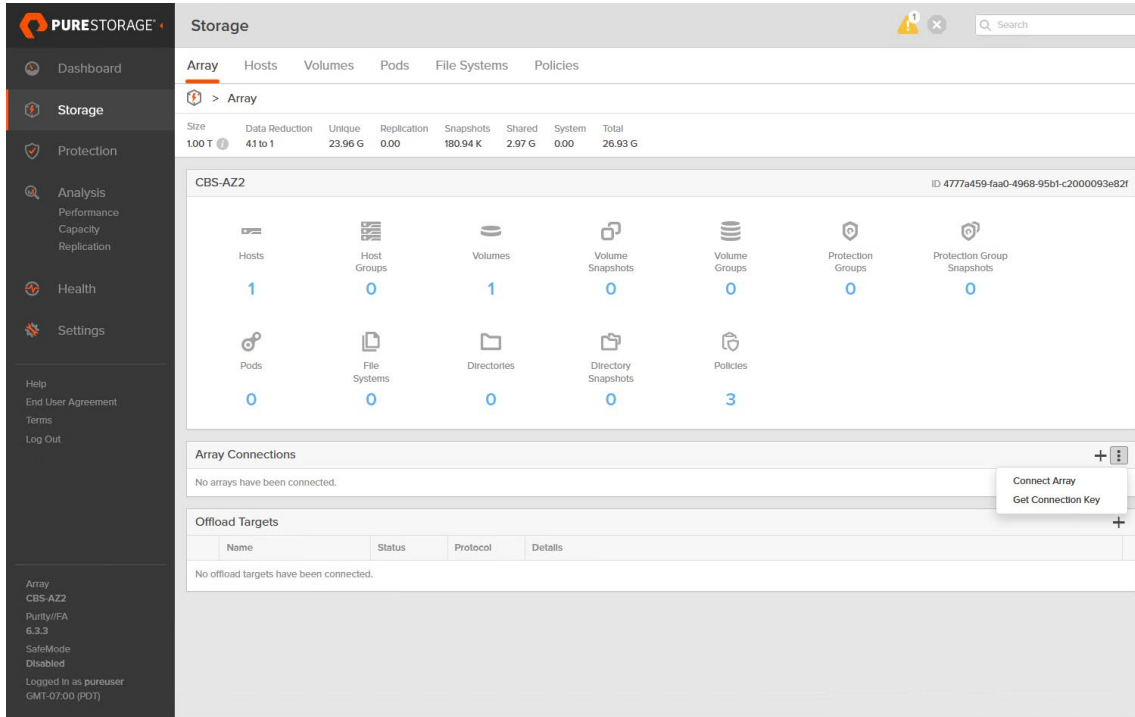


Figure 8. The storage section on the target instance with the options for array connections expanded

In the prompt that appears, copy the Connection Key.



Figure 9. The prompt with the connection key for the target instance

On the source instance, navigate to the options in Array Connections and select **Connect Array**. When the prompt appears enter in the Management Address for the target, ensure the Type is Async-Replication, enter the Connection Key from the target system, ensure the Replication Transport is Ethernet (IP), and optionally enter any Replication Addresses for the target system.

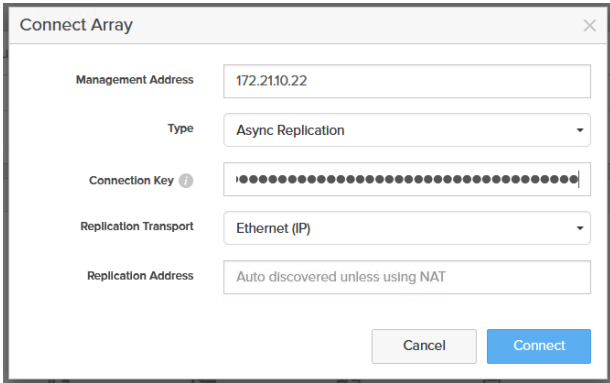


Figure 10. The Connect Array dialog on the source instance

Upon successful connection, the Array Connections section will be populated with the target instances information.

Name	Status	Type	Version	Management Address	Replication Transport	Replication Address	Throttled
CBS-AZ2	connected	async-replication	6.3.3	172.2110.22	Ethernet (IP)	172.2110.20 172.2110.26	False

Figure 11. The array connections section on the source instance after a connection has been created

## Asynchronous Replication for a Five-minute Recovery Point Objective

To achieve a five-minute RPO, asynchronous replication of volume snapshots between Pure Cloud Block Store instances is used. Volumes are added to a Protection Group with the target instance as a replication target.

### Initial Setup

The process in this initial setup of a protection group and asynchronous replication will use a single volume, SQL-UserDB.

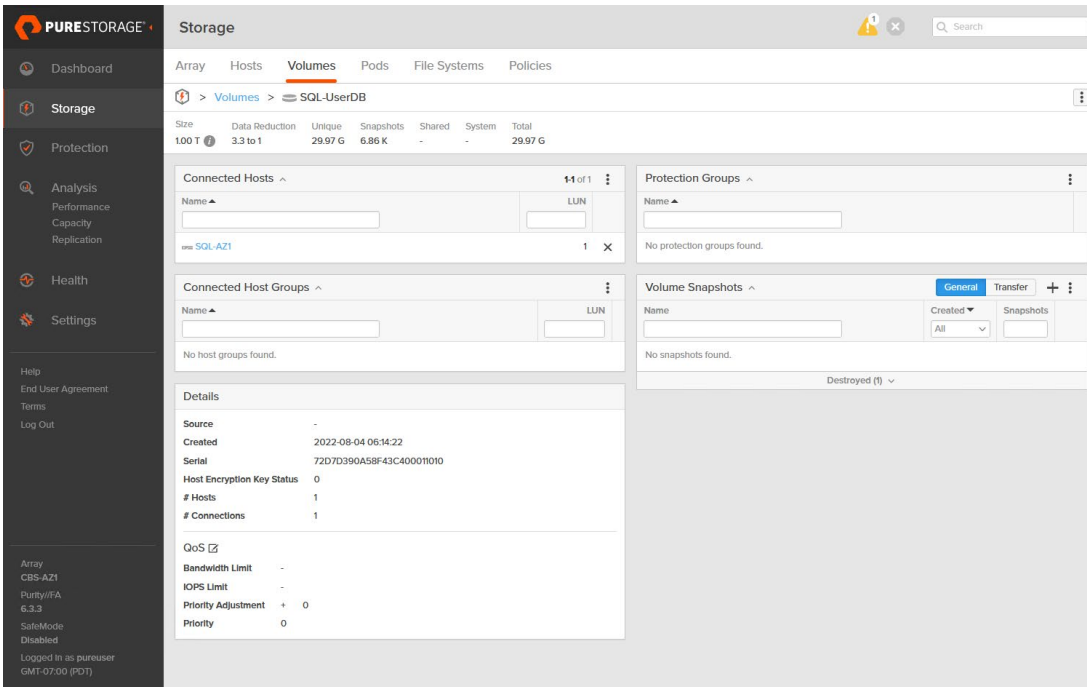


Figure 12. A single volume will be added to the protection group. This volume contains the UserDB database.



In the graphical user interface of the source instance, navigate to the Protection section and select **Protection Groups**.

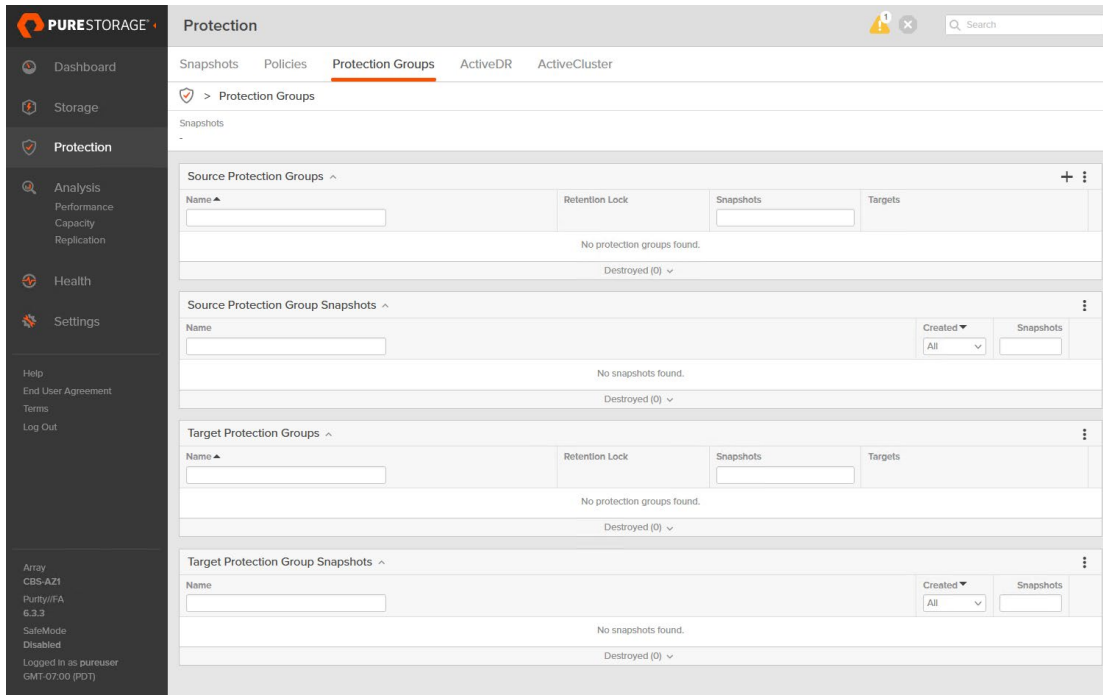


Figure 13. The Protection Groups management view on the source instance

Create a new protection group by identifying Source Protection Groups and selecting the +. In the dialog which appears, provide a name for the protection group and select **Create**.

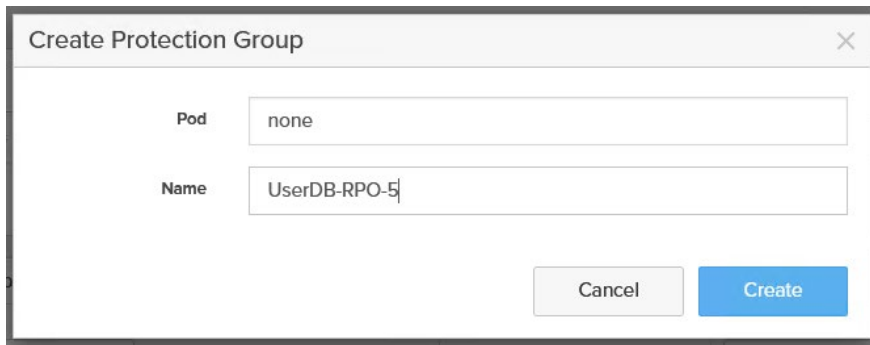


Figure 14. The Create Protection Group dialog

Once the protection group has been created, it will be displayed as a source protection group. To manage a specific protection group, select it from this list.

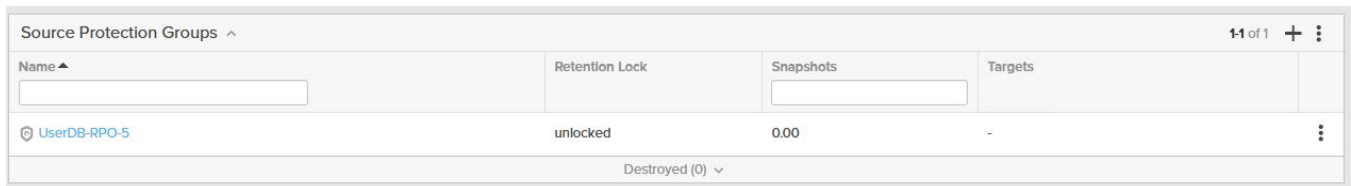


Figure 15. The new protection group is shown in the list of source protection groups

When selecting the specific protection group, its management view will be shown. This is where the various policies can be applied. Members, such as volumes or hosts, are added, or targets for replication are set for it.

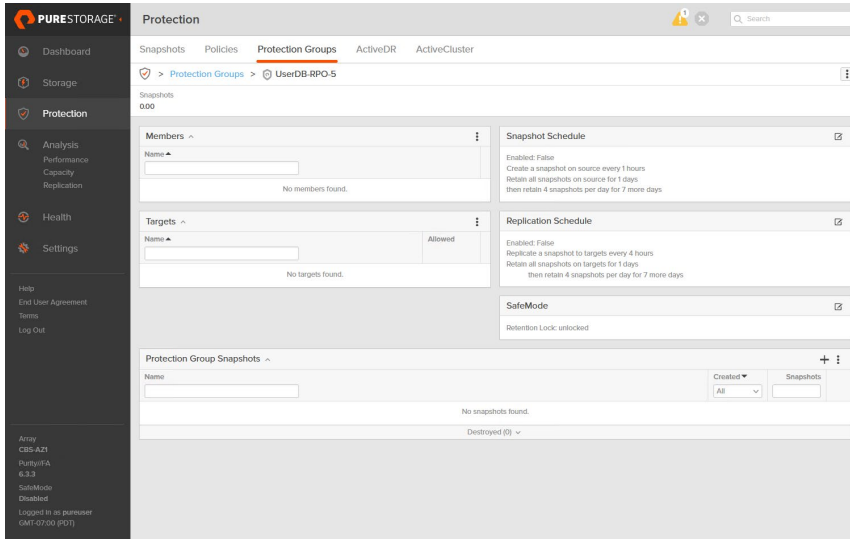


Figure 16. The UserDB-RPO-5 protection group management view

In the management view, identify the Members section and select the three ellipses next to it. In the dialog which appears, select **Add Volumes**.

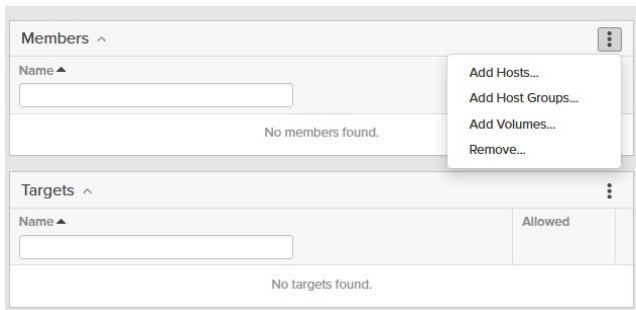


Figure 17. The Members section of the protection group

In the Add Members dialog, select the volumes to add and then select **Add**.

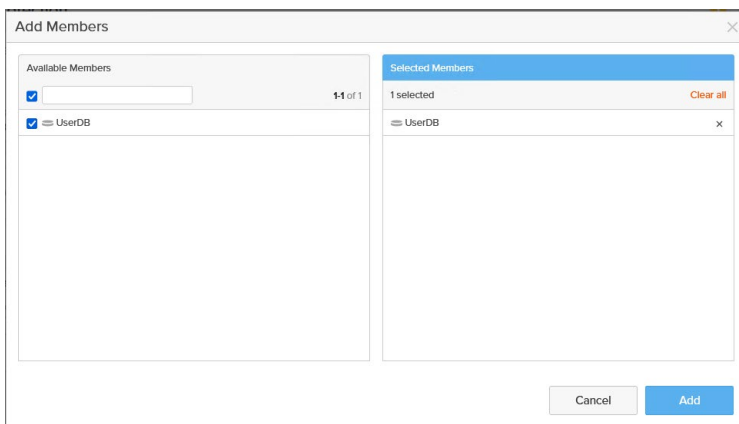


Figure 18. The Add Members dialog



Once the members have been added, identify the Targets section, select the three ellipses next to it and then select **Add**.

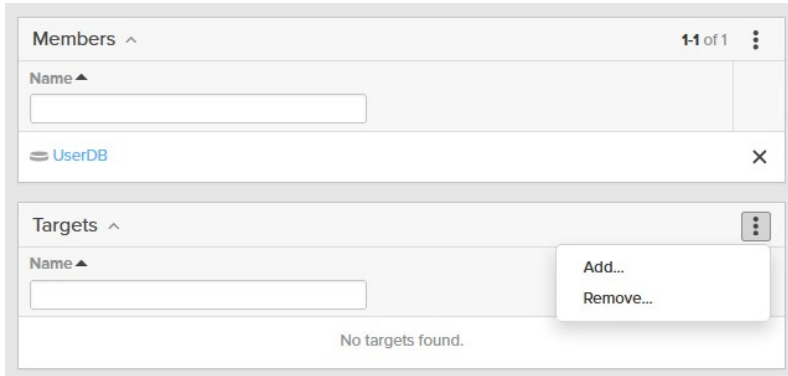


Figure 19. The targets section of the protection group

In the Add Targets dialog, select the targets to add and then select **Add**. This is where the snapshots will be replicated to.

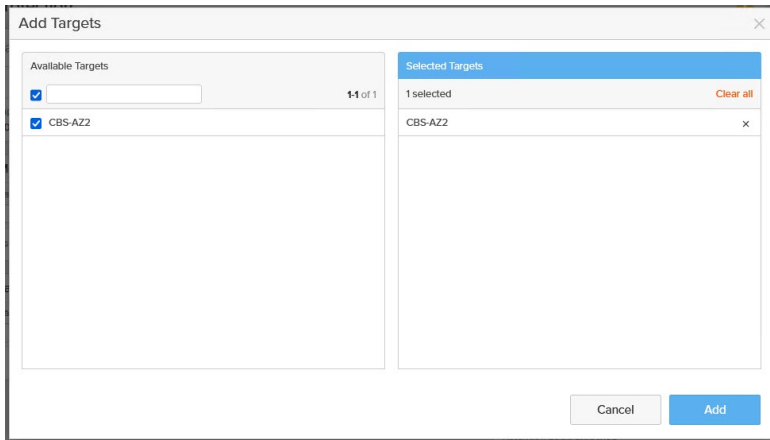


Figure 20. The Add Targets dialog

At this point, the protection group should consist of members and targets to replicate volume snapshots. Other aspects of the protection group that can be customized are the schedules. There are schedules for creating snapshots at regular intervals, and there are separate schedules for replicating those snapshots to a target. This can be customized to meet different RPO requirements as low as five minutes.



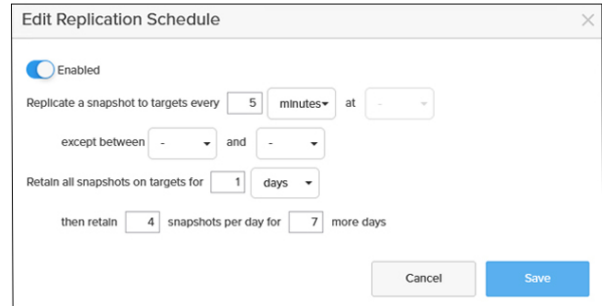
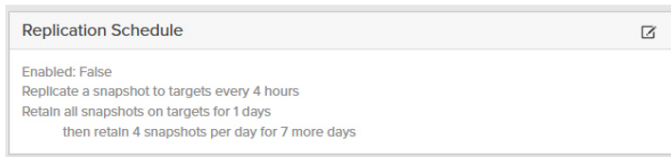
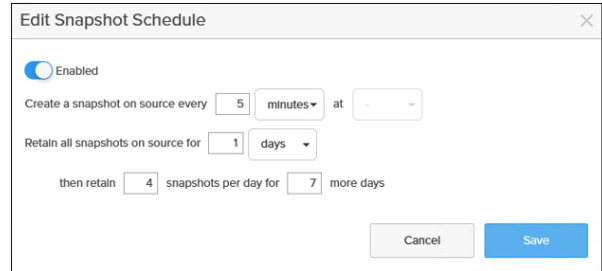
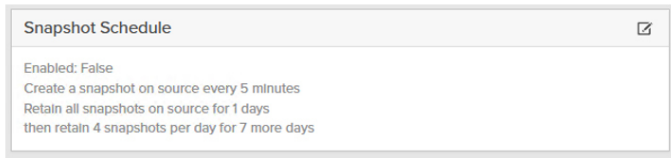


Figure 21. The various snapshot and replicate options for a protection group

It is possible to create and replicate a volume snapshot manually. This can be done in the management view for the protection group, in the Protection Group Snapshots section. Selecting the + will begin the manual snapshot creation process.

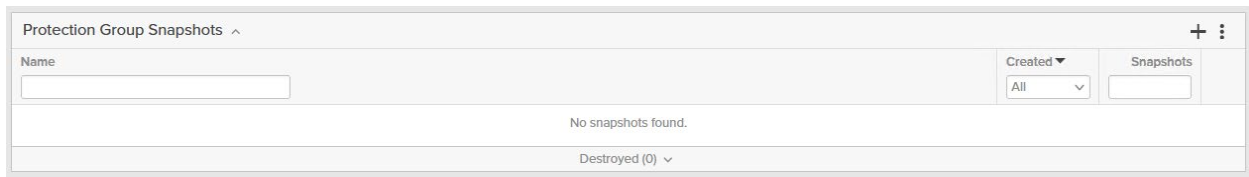


Figure 22. The Protection Group Snapshots section

In the Create Snapshot dialog which appears, provide an **optional suffix** to identify this protection group snapshot and select **Replicate Now** to ensure it is replicated to the target.

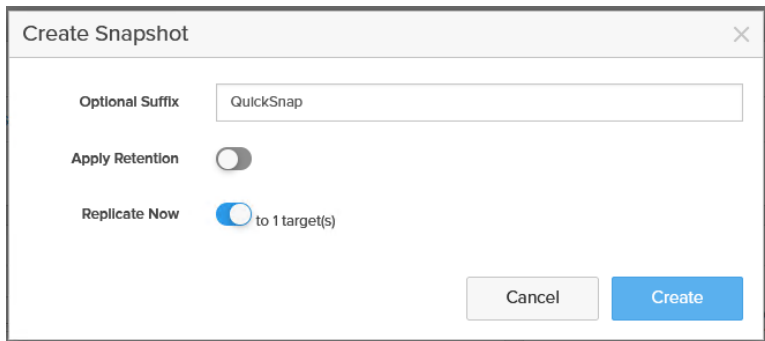


Figure 23. The create snapshot dialog

### Disaster Recovery Process

The disaster recovery process involves restoring a snapshot to a new or existing volume. This volume is then connected to the relevant Microsoft SQL Server and the database attached to the instance on it. All operations are performed using the target instance.

On the target instance, navigate to protection groups in the protection section. Identify the relevant target protection group.

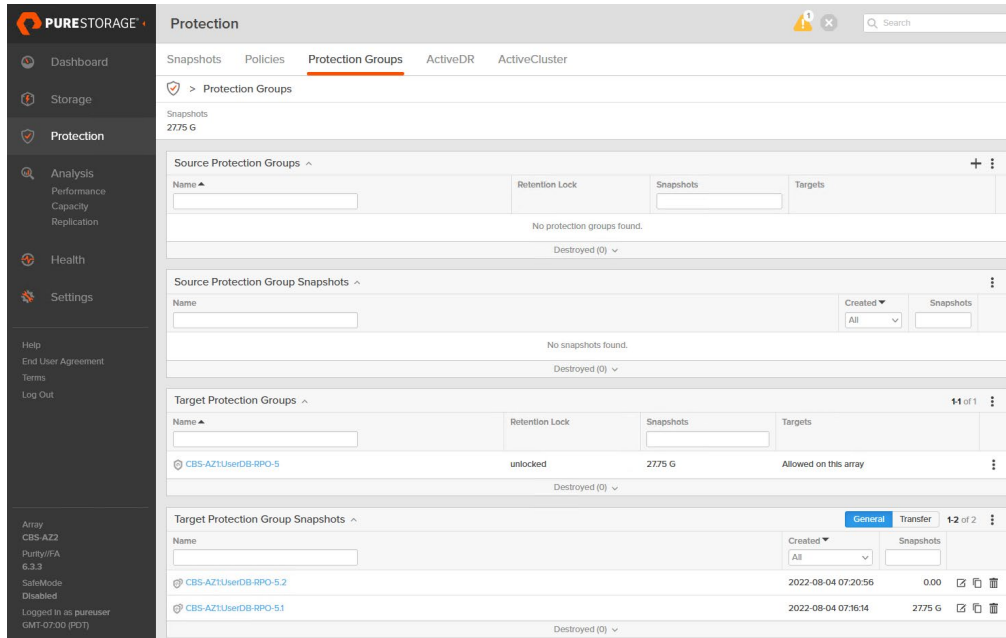


Figure 24. Protection Groups on the target instance

Once the relevant protection group management view has been brought up, select an entry from the Protection Group Snapshots. This entry will contain the recovery point that needs to be restored.

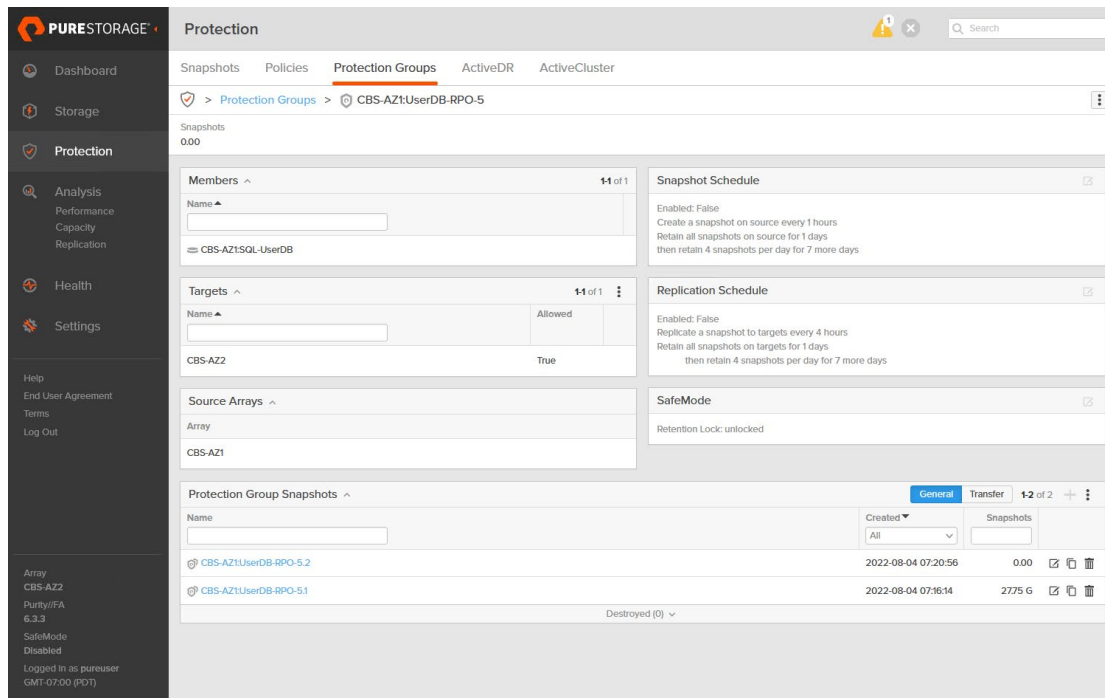


Figure 25. Identify the relevant protection group snapshot to restore from

Selecting a protection group snapshot will bring up a dialog showing the volume snapshots contained within it. These volume snapshots will be restored to a new or existing volume.





Figure 26. The volume snapshots contained within a protection group snapshot

Once the relevant volume snapshot has been identified, either restore it to a new volume or overwrite an existing volume with its contents. After the volume has been connected to the relevant target host, discover it and ensure the filesystem is mounted to a relevant location. For the purposes of this demonstration, the path allocated to the newly connected volume is C:\SQL\UserDB.

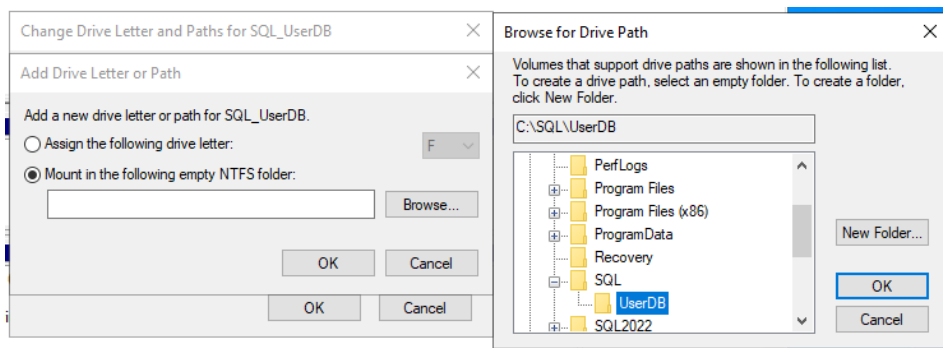


Figure 27. The location at which the filesystem is mounted on the target system

Using SQL Server Management Studio connected to the target instance, select **Databases** from the management tree and select **Attach**.

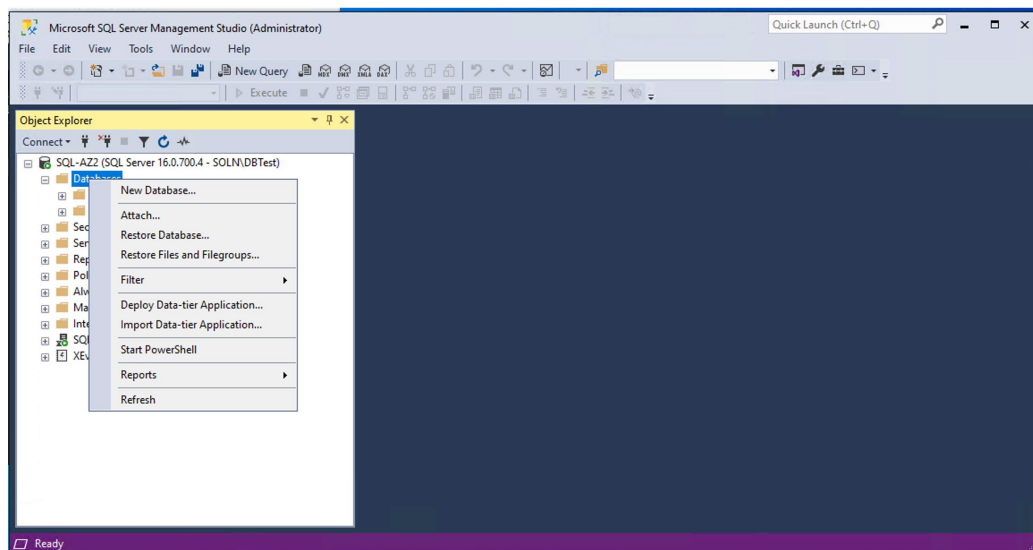


Figure 28. The management tree for the target SQL Server instance in Microsoft SQL

In the **Attach Databases** dialog, select **Add** to find the master data file (MDF) file for the database.



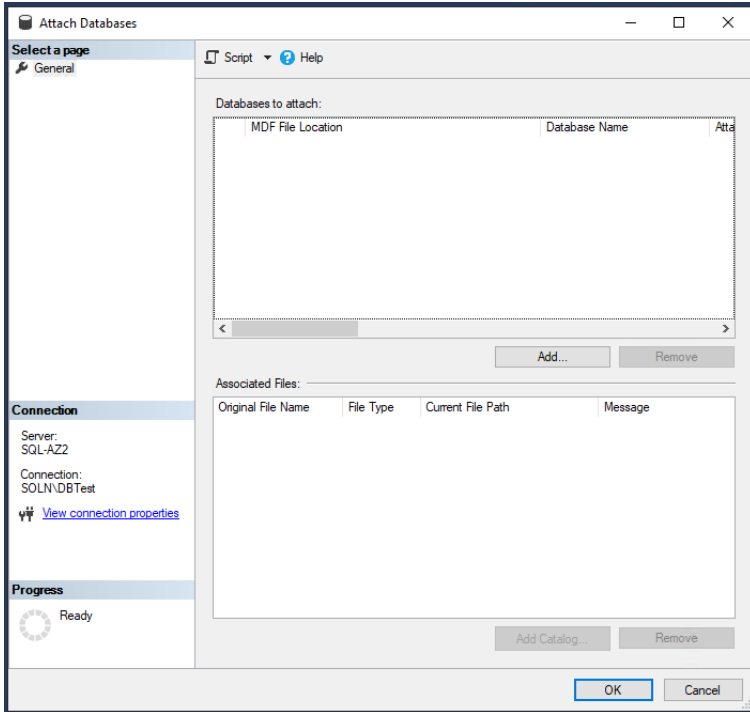


Figure 29. The Attach Databases SSMS dialog

In the additional dialog that is shown, navigate to the newly mounted volume and select the .mdf file for the database on it, select it, and then select **OK**.

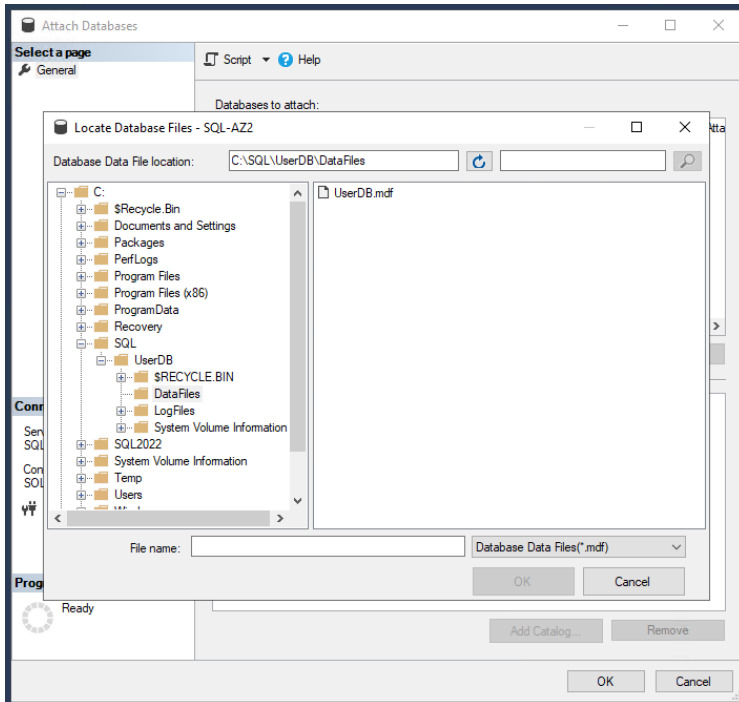


Figure 30. The Locate Database Files for instance attach dialog



Ensure that each of the files for the database has been resolved in the database details section of the dialog. It may be necessary to navigate to and select the location of each file prior to completing this process. Once ready to complete the attach process, select **OK** in the Attach Databases dialog.

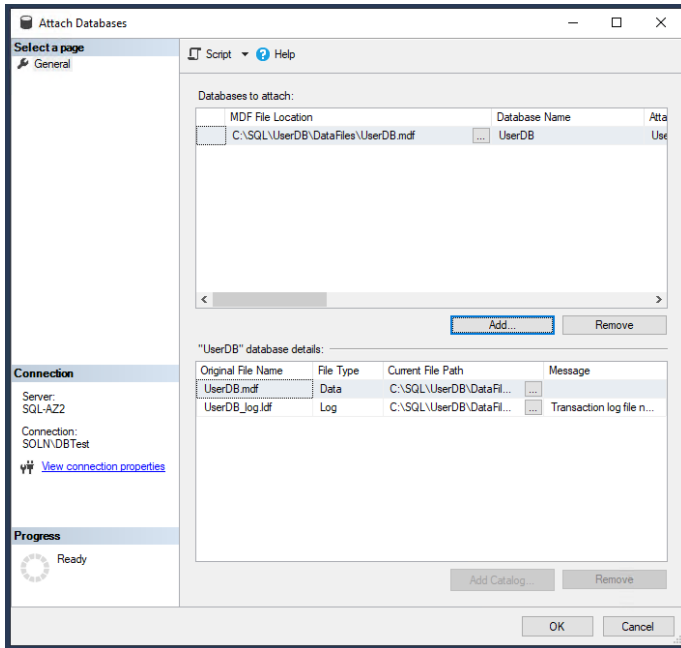


Figure 31. A database is ready to attach in the attach databases dialog

Once completed, the database will be attached and shown in the management tree of the object explorer.

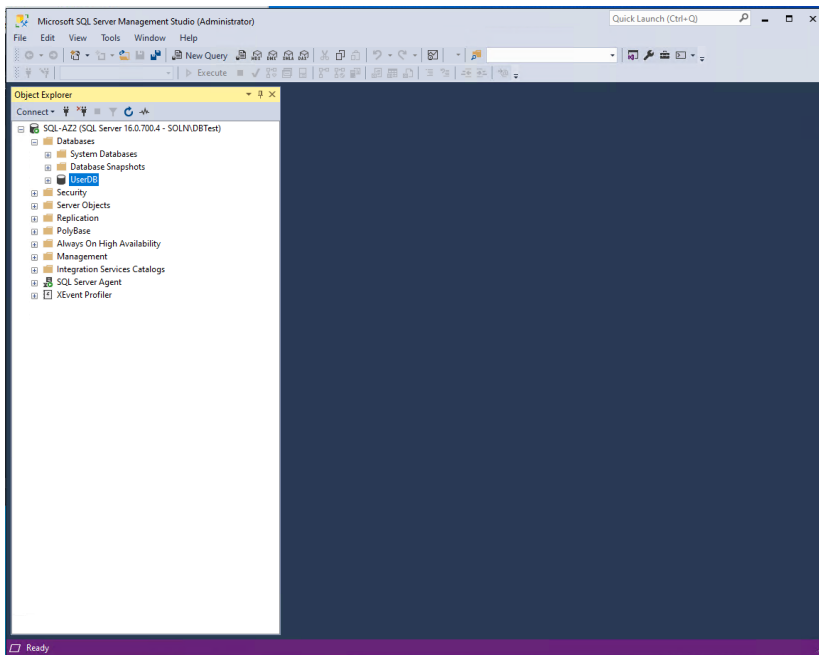


Figure 32. A running database in the target SQL Server instance



At this point, both the storage and SQL Server instance have been recovered. Note that some objects, such as server logins, SQL Server Agent Jobs, and linked Servers etc., that are external to this specific database, would not have been recovered and may require recreating or updating.

### ActiveDR for Near-zero Recovery Point Objectives

To achieve a near zero recovery point objective, the continuous replication (ActiveDR) of volumes between Pure Cloud Block Store instances is used. Volumes are added to a POD with the target instances as a replication target.

#### Initial Setup

The process in this initial setup of a pod and continuous replication will use a single volume, SQL-UserDB.

A Pod needs to be created prior to a volume being set up for continuous replication. To create a Pod in the graphical user interface, navigate to pods in the Storage view. In the section for Pods identify and select the + to create a new pod.

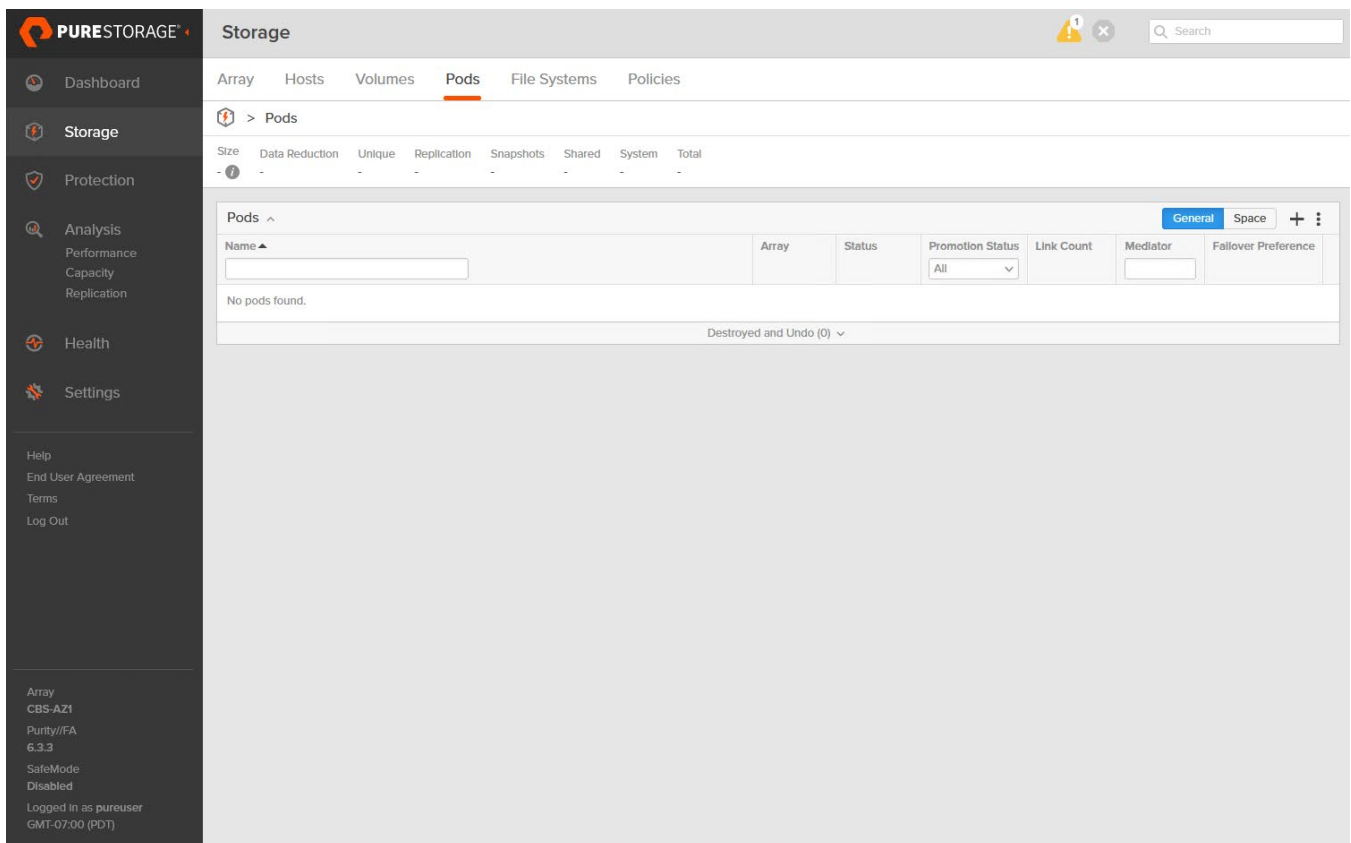


Figure 33. The pods view

Provide a name for the pod in the dialog and then select **Create**.



Figure 34. The create pod dialog

Once created, the new pod will show up in the list. Select the newly created pod to navigate to the management view for it.

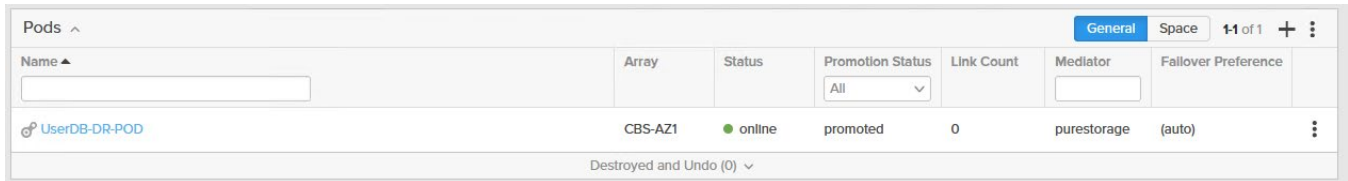


Figure 35. The new pod will be displayed in the list of pods.

In the pod management view, identify the volumes section. In this section, a new volume can be created, or an existing volume moved in. In this example, an existing volume is being moved into the Pod. Select the three ellipses next to volumes and then select **Move In**.

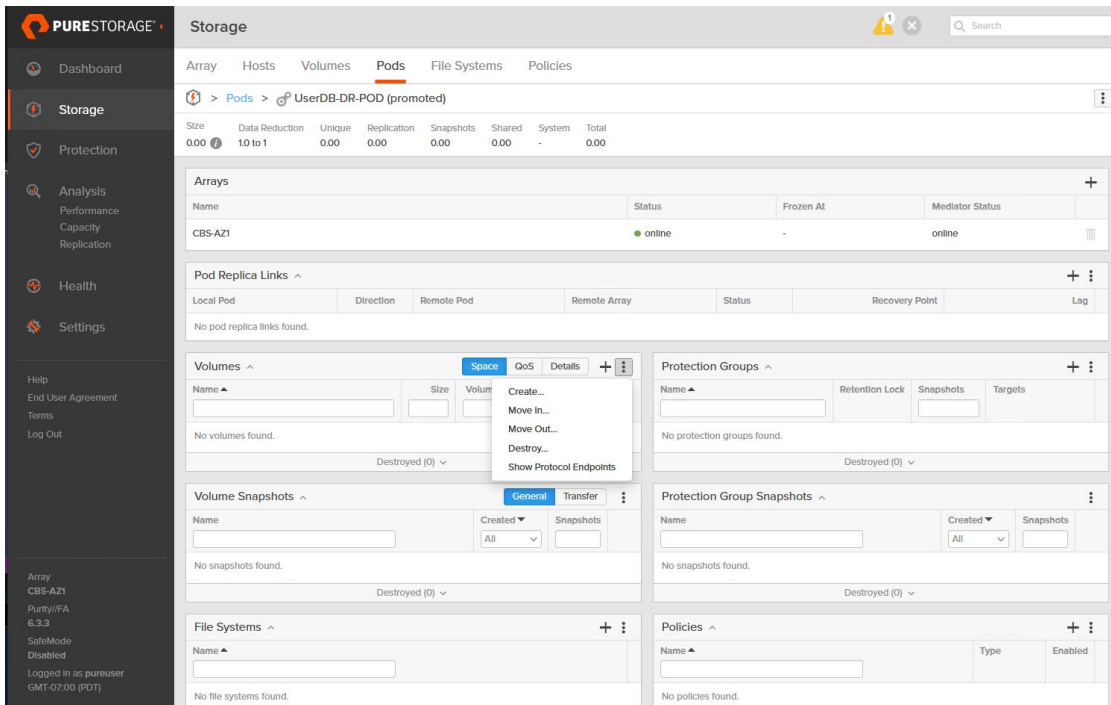


Figure 36. The Pods management view

Select the volume from the list of available volumes and then select **Move**.



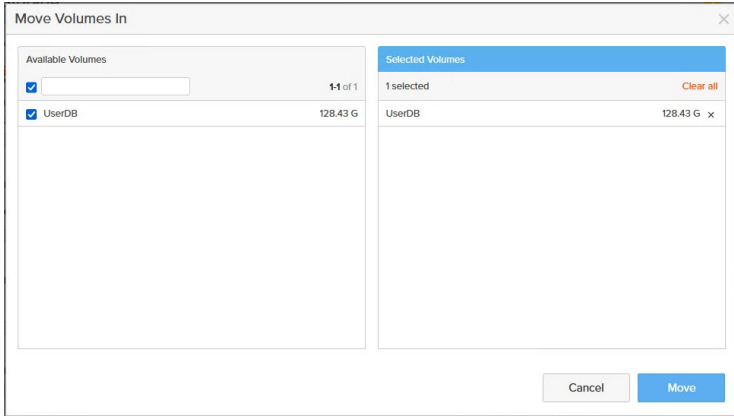


Figure 37. The Move Volumes In dialog

Once the volume has been moved into the pod, it will be shown in the list of volumes with a new naming convention: <Pod Name>::<Volume Name>.

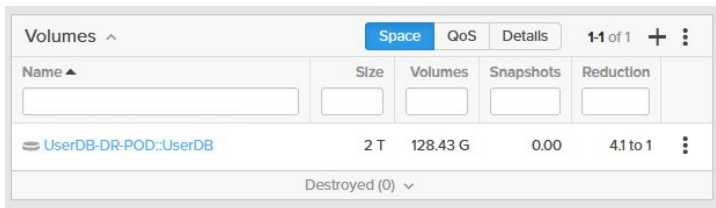


Figure 38. The list of volumes for the pod

To enable ActiveDR for the volumes that have been added, a replica link to a target instance needs to be created. This can be achieved by selecting the + or the three ellipses in Pod Replica Links.

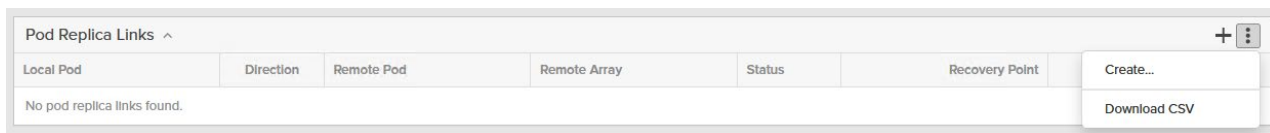


Figure 39. The pod replica links section

In the Create Replica Link dialog, select a remote array from the drop-down list, select or create a remote pod for ActiveDR to target, and then select **Create**.

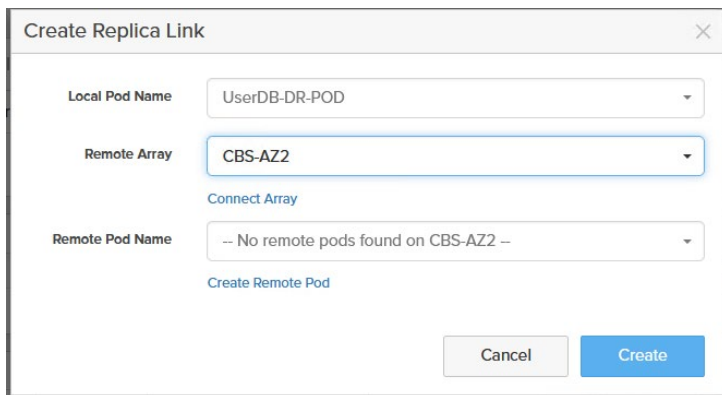


Figure 39. The Create Replica Link dialog

If a remote pod needs to be created, the following dialog will be shown. Provide a name for it and then select **OK**.

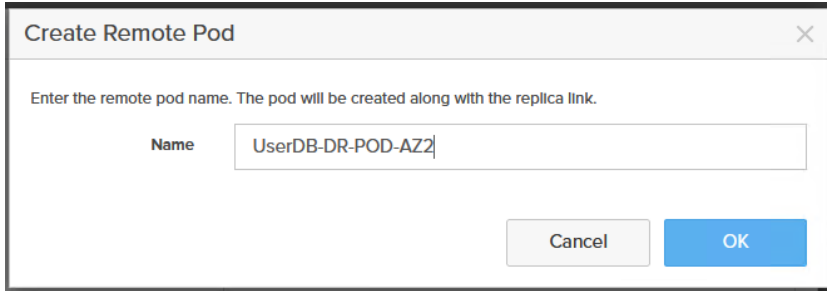


Figure 40. The Create Remote Pod target

Once the replica links have been created, the pod will begin an initial baseline of the volume. This involves an initial copy of volume data from the local pod to the remote pod.

Pod Replica Links ^							11 of 1 +	⋮
Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag		
UserDB-DR-POD (promoted)	→	UserDB-DR-POD-AZ2	CBS-AZ2	baselining	-	-		⋮

Figure 41. The pod replica link with a baselining status

Once the baselining is complete, the replica link should transition to a replicating status.

Pod Replica Links ^							11 of 1 +	⋮
Local Pod	Direction	Remote Pod	Remote Array	Status	Recovery Point	Lag		
UserDB-DR-POD (promoted)	→	UserDB-DR-POD-AZ2	CBS-AZ2	replicating	2022-08-05 04:02	1s		⋮

Figure 41. The pod replica link with a replicating status

Once in a replicating state, the pod on the source Pure Cloud Block Store instance is in a promoted state while the pod on the target is in a demoted state. A promoted pod allows for both read and write operations to occur while a demoted pod's volumes are read only.

A demoted pod's volumes on the target can be connected to a host but should not be mounted or set as offline until a recovery process is needed.



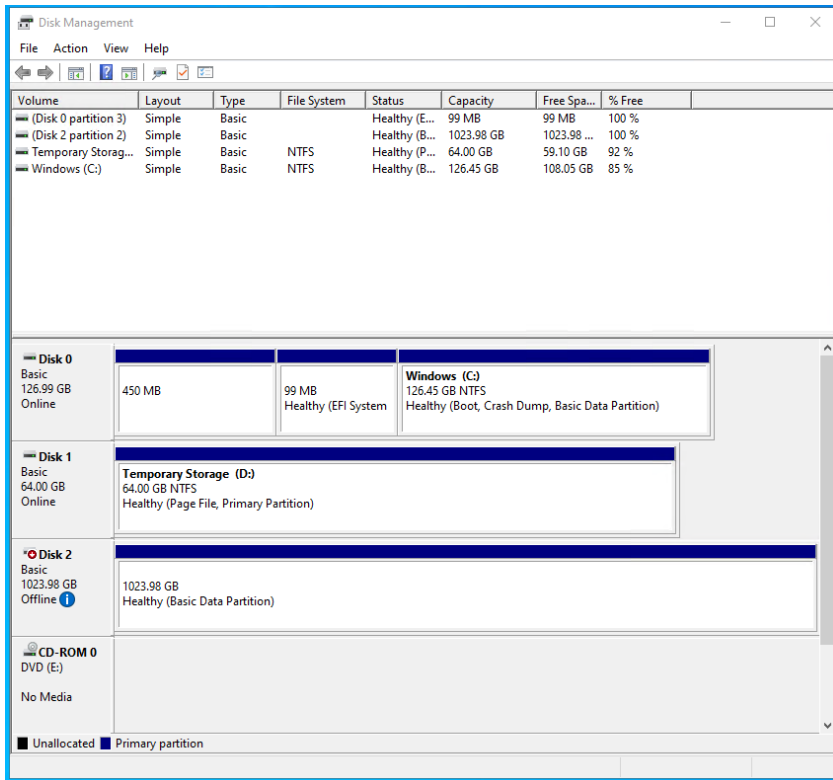


Figure 42. An example of a demoted pods volumes being set as offline in Microsoft Windows Disk Management

### Disaster Recovery Process

In the event of a disaster, the target pod needs to be promoted and the databases attached to a remote SQL Server instance.

In a planned failover, where the source cloud block store instance is still available and replicating, the pod on the source needs to be demoted first before promoting the pod on the target to ensure no data loss. This can be achieved through the selection of the three ellipses in the pod management view and selecting demote.





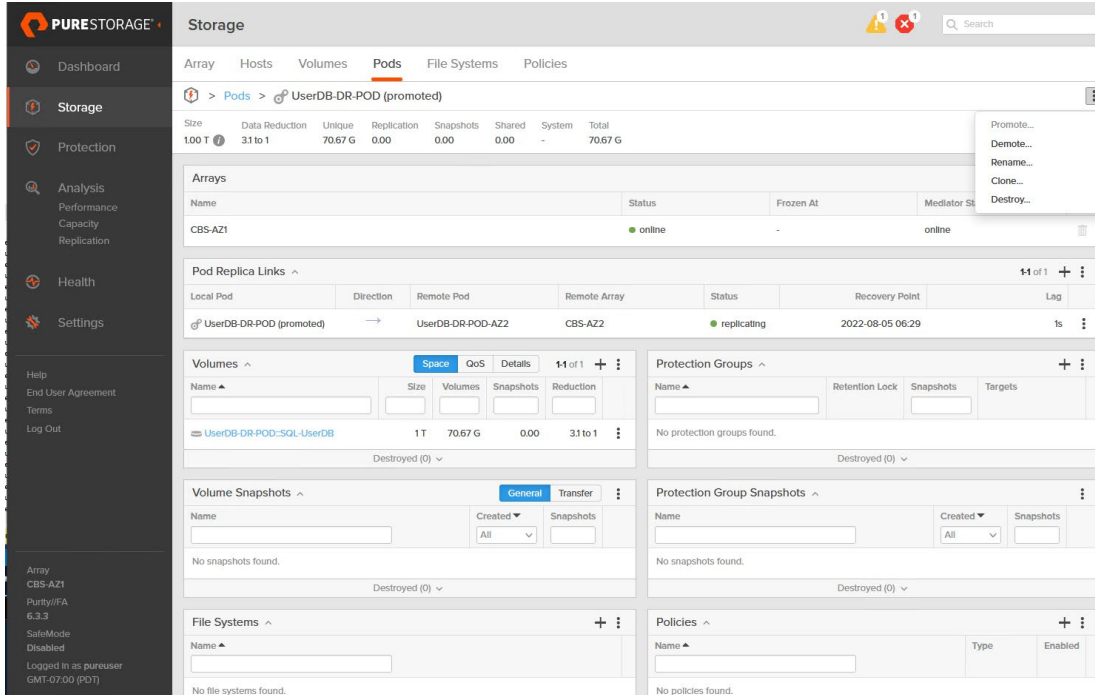


Figure 43. The pod management view on the source instance

In the Demote Pod dialog, ensure Quiesce is selected when it is required that all local data is replicated to the target pod.

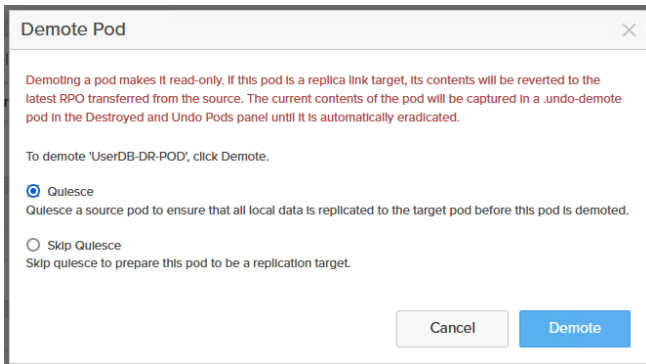


Figure 44. The Demote Pod dialog

In the event of an unplanned disaster, where the source Pure Cloud Block Store instance is offline, only the target pod needs to be promoted.

Promotion of the target pod can be achieved through the selection of the three ellipses in the pod management view and selecting **Promote**.



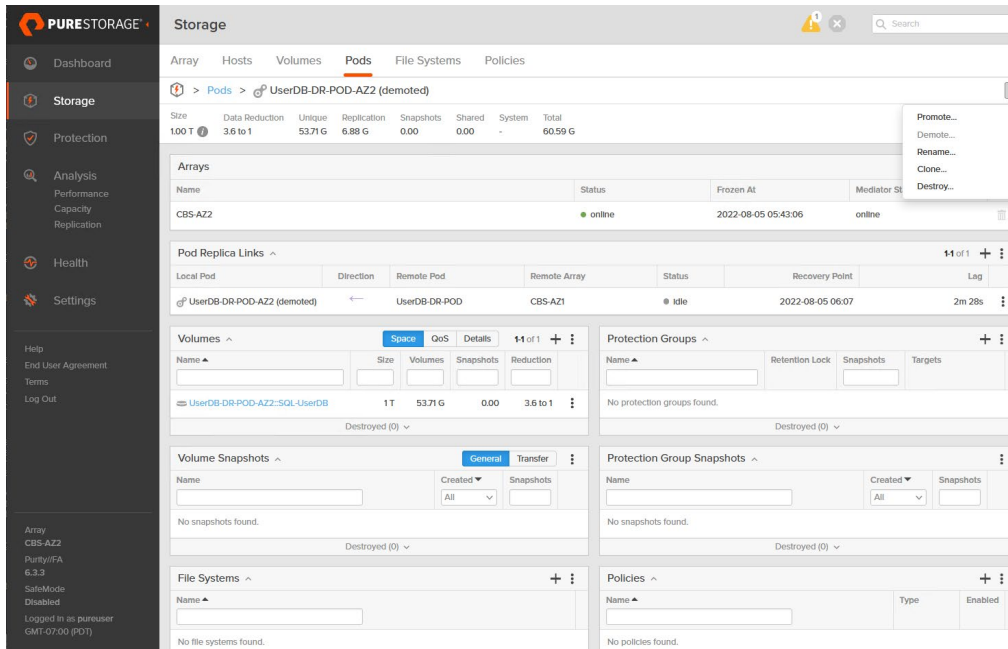


Figure 45. The pod management view on the target instance

A dialog will appear requesting confirmation for the pod promotion. Select **Promote** to continue.

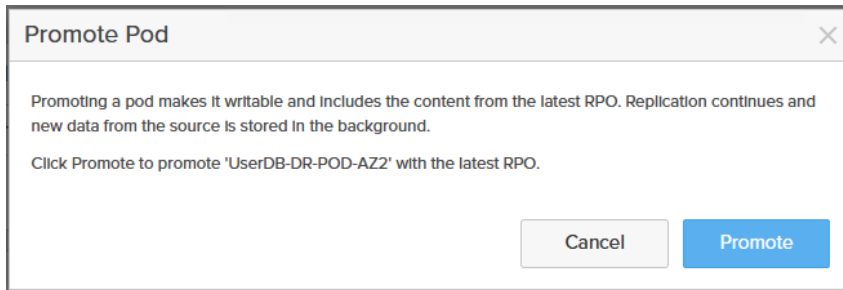


Figure 46. The promote pod dialog

Once the pod on the target instance has been promoted, it should be shown as such in the Pod Replica Links screen.

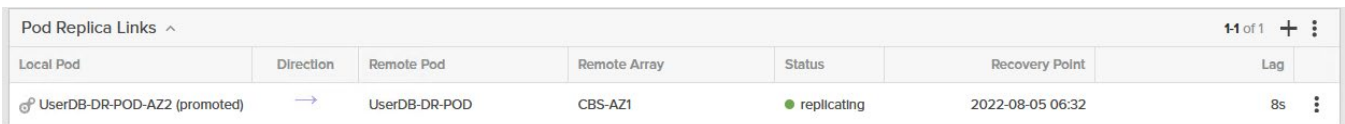


Figure 47. A promoted local pod in Pod Replica Links

If the pod on the source instance has been demoted, it will be displayed as demoted with the quiesced status.

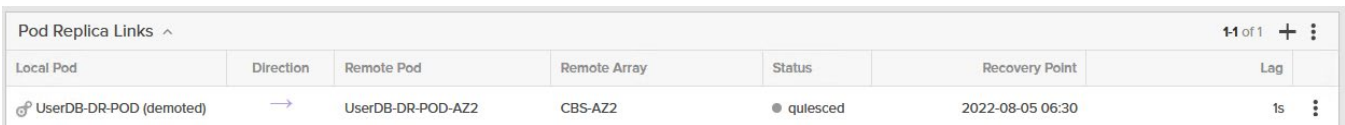


Figure 48. A demoted and quiesced pod on the source instance

At this point, once the volume has been promoted, it can be set to online on the target system containing the SQL Server instance and provided with a drive letter or path.

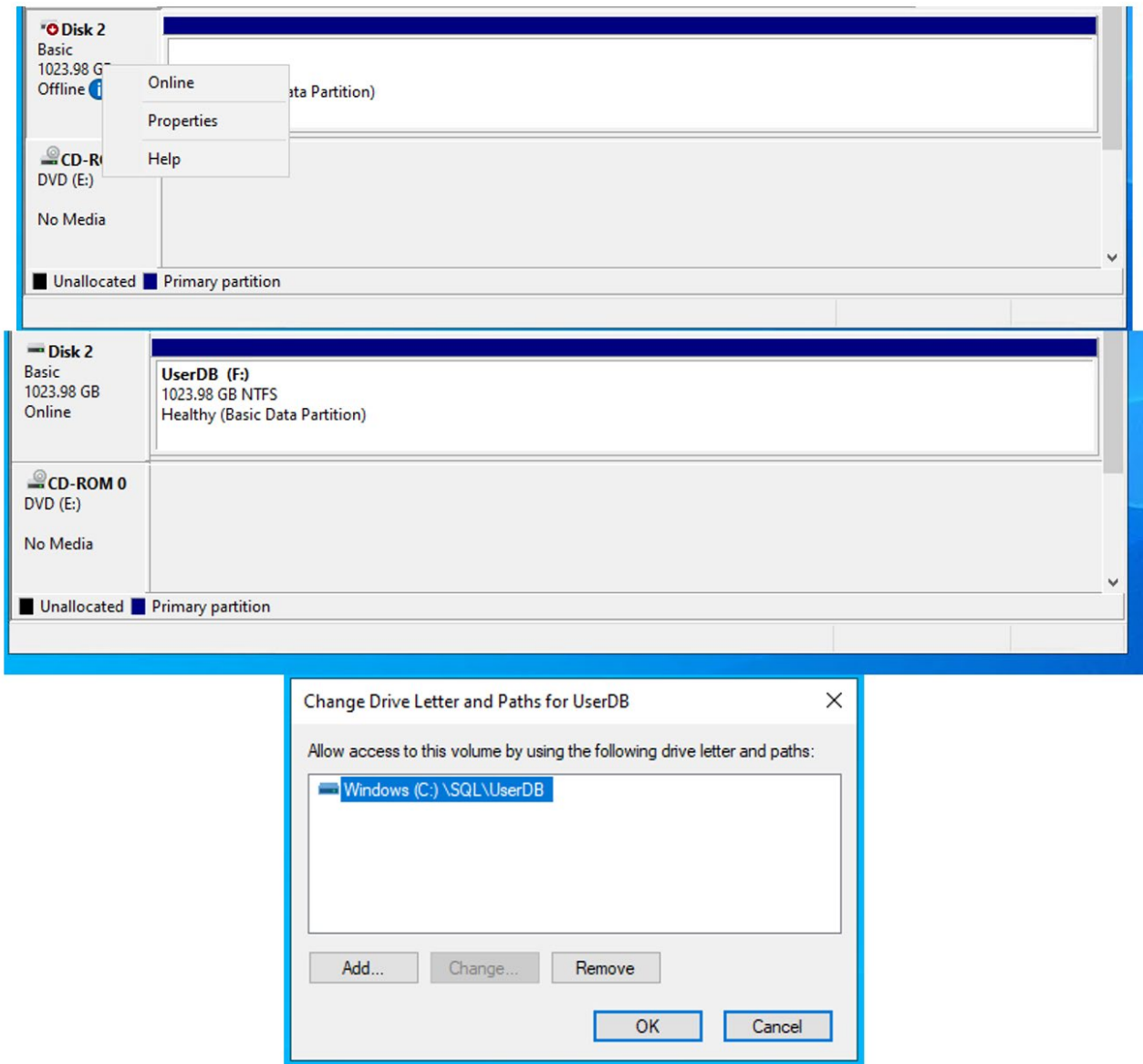


Figure 49. Setting a volume online in disk management and ensuring it has the correct drive letter or path

At this point, the database files can be attached to the target SQL Server instance and recovered. These steps have been set out within the section on Disaster Recovery for [Asynchronous Replication](#).

### Seeding Replicas Across Regions in an Always On Availability Group

[Always On availability groups](#) are a feature in SQL Server that provide a replicated environment for a discrete set of user databases. Availability groups can be created to provide high availability, read scale, or disaster recovery.

An availability group, created between SQL Server instances deployed across regions, could provide a way in which to ensure disaster recovery for specific databases using replicas connected with asynchronous SQL Server replication.



SQL Server instances, utilizing Always On availability groups, can be combined with Pure Cloud Block Store data services to lower costs and increase efficiency. Volume snapshots can be used to replace the automatic seeding and reseeding of always-on databases while providing a more efficient method of reestablishing high availability in the event of an outage.

Where two (2) Pure Cloud Block Store or more instances are deployed in different regions, asynchronous replication can be used as the data copy mechanism. As asynchronous replication only transfers unique data (post compression and deduplication) lower costs and transfer times can be achieved for inter-region data transfers.

This implementation guide for seeding replicas in an Always On availability group using Volume Snapshots only works for SQL Server 2022 and later as it utilizes the [Create a Transact-SQL snapshot backup](#) process. This can be implemented across availability zones or regions when using two (2) or more Pure Cloud Block Store instances or can be implemented for many SQL Server instances in the same availability zone using a single Pure Cloud Block Store instance.

In this implementation, there will be no automatic failover between instances and utilizes asynchronous SQL Server replication for communication between the replicas.

The Microsoft SQL Server documentation contains a [Getting Started with Always On Availability groups](#) guide for a detailed overview of the technology.

### Initial Setup

Prior to implementing an Always On availability group, the functionality must first be enabled. This can be accomplished using the [SQL Server Configuration Manager](#), navigating to the properties for the instance service and then selecting Enable Always On Availability Groups in the Always On Availability Groups Tab. Once complete, click **OK** or **Apply** and restart the service. This should be done on all instances to join the availability group.

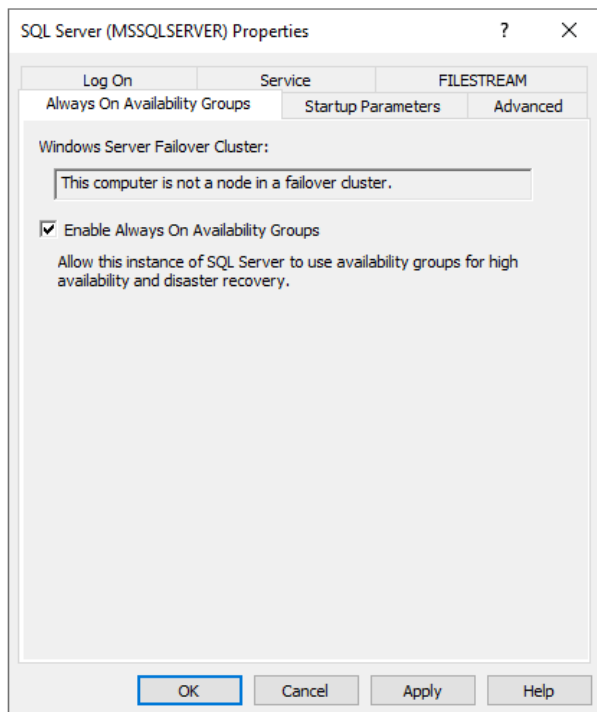


Figure 50. The instance properties for the SQL Server instance



The ordering of these steps assumes that the data synchronization preference is set to join only—where the database(s) intended for the availability group are already on the intended replica(s) and in the restoring state. To use this synchronization preference, skip initial database synchronization; the availability group database(s) can be restored after the availability group has been created.

**Taking a snapshot backup:** Prior to creating a volume snapshot and replicating it, the database needs to be suspended and prepared for a storage snapshot. To do this execute the following T-SQL command, substituting the database name:

```
ALTER DATABASE UserDB SET SUSPEND_FOR_SNAPSHOT_BACKUP = ON
```

A volume snapshot can now be created and/or replicated. This process can be found in the section for [Asynchronous Replication for a Five-minute Recovery Point Objective](#)—Initial Setup.

After the volume snapshot has been created, execute the following command, substituting database name and location for the metadata file (this file is required for recovery), to finish the snapshot backup process.

```
BACKUP DATABASE UserDB TO DISK='C:\SQL\UserDB-Replica.bkm' WITH METADATA_ONLY
```

**Recovering from a snapshot backup:** At this point, the snapshot backup should be restored either to a new or existing volume and mounted on the intended replica. Execute the following command, substituting the database name and metadata file location:

```
RESTORE DATABASE UserDB FROM DISK = 'C:\SQL\UserDB-Replica.bkm' WITH METADATA_ONLY, REPLACE,  
NORECOVERY
```

Once the database has been restored and is in a restoring state, the final step is to ensure that transaction logs on the replica and source system are identical. This can be done by backing up the transaction log with the following command, substituting the database name and intended location:

```
BACKUP LOG UserDB TO DISK = 'C:\SQL\UserDB-seed.trn' WITH FORMAT
```

To restore the transaction log to the intended replica, execute the following command, substituting the database name and transaction log backup location:

```
RESTORE LOG UserDB FROM DISK = 'C:\SQL\UserDB-seed.trn' WITH NORECOVERY
```

At this point, the creation of a new availability group or the joining of the database to an existing availability group can occur.

**Join only initial synchronization:** Using the join only option for the availability group, enter the New Availability Group Wizard in SQL Server Management Studio.



In the Specify Options section, provide a name for the availability group and specify a Cluster type and then select **Next**.

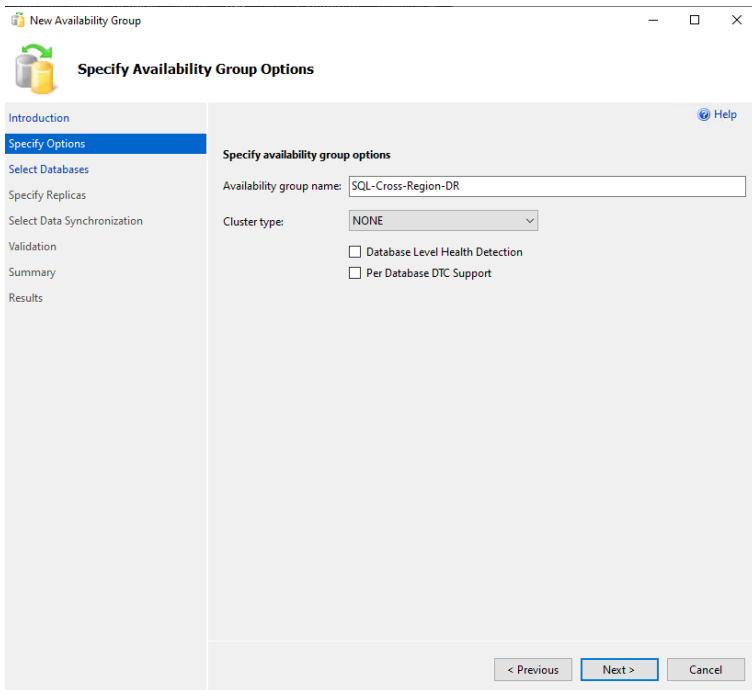


Figure 51. New Availability Group Wizard, Specify Availability Group Options

In the next option, Select Databases, select all the databases that will be added to the availability group and then select **Next**.

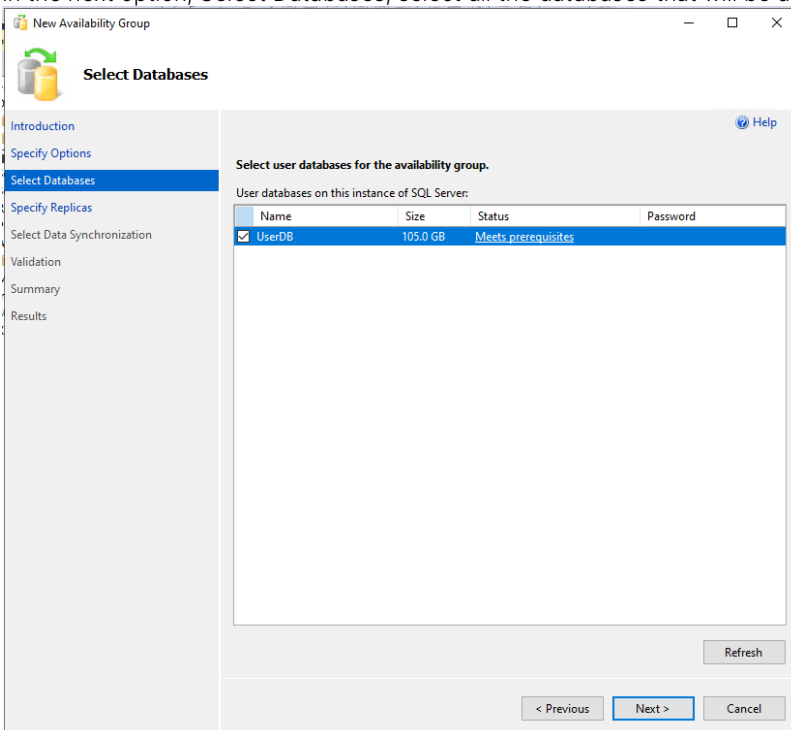


Figure 52. Selecting databases in the New Availability Group wizard



In Specify Replicas, select **Add Replica** and connect to the intended replica instances with the relevant user. Once the SQL service instances that will be used as availability replicas are all present, set the Availability Mode. For cross-region disaster recovery, it is recommended to set Asynchronous commit as **the Availability Mode**. Select **Next** to continue.

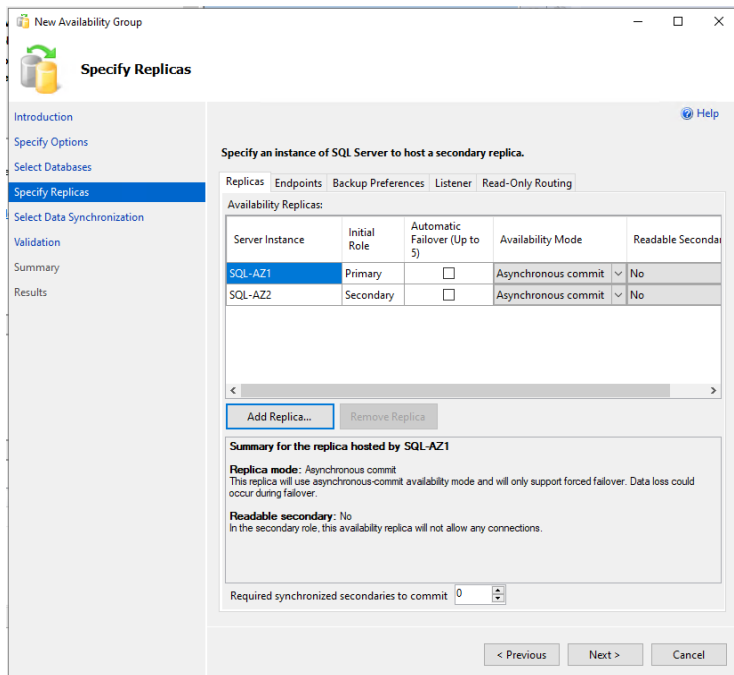


Figure 53. Selecting databases in the New Availability Group Wizard

On this screen, select **Join only** if the database has already been restored and is ready to join. Select **Skip initial data synchronization** if the database has not been restored and is not ready to join the availability group. Select **Next** to continue.

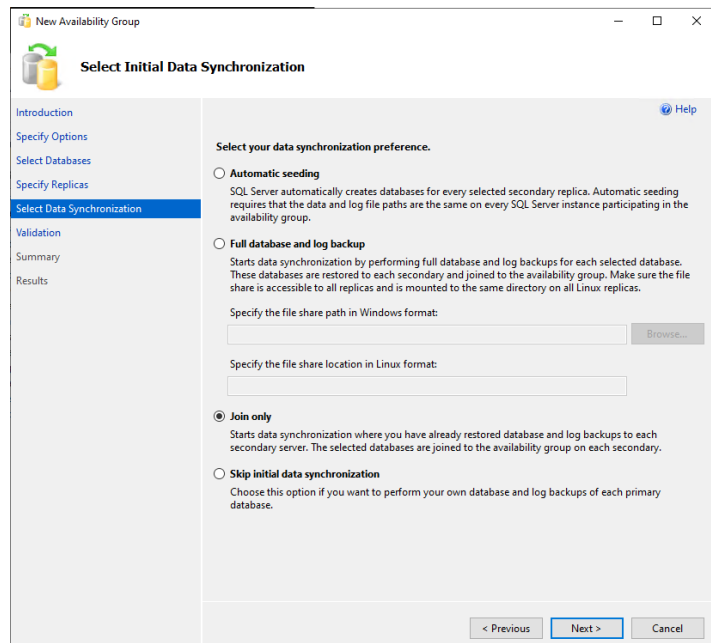


Figure 53. Initial data synchronization in the New Availability Group Wizard Initial Data Synchronization



Ensure all the critical availability group validation steps have a result of Success or Skipped. Warning results should be inspected prior to continuing. Select **Next** to proceed to the summary and check the summary view prior to continuing.

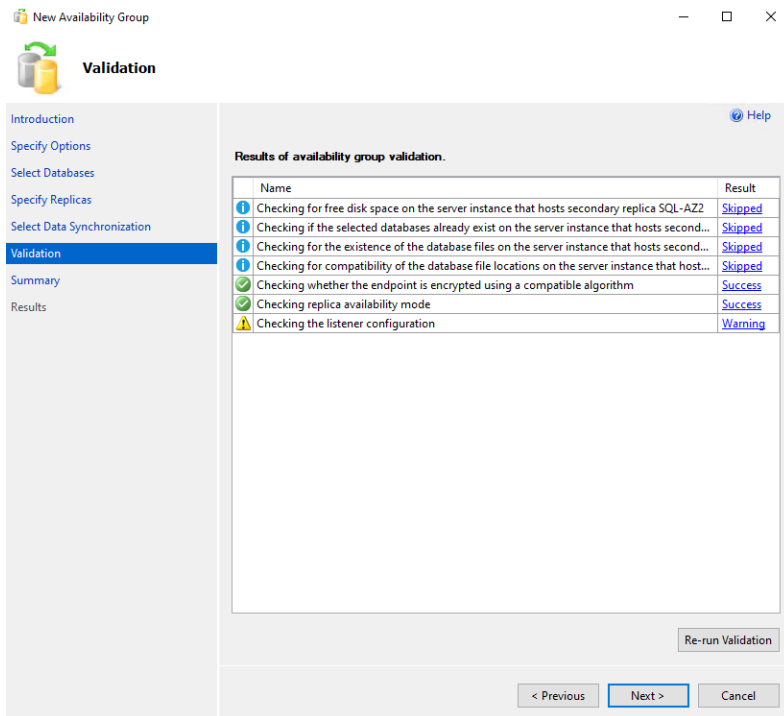


Figure 54. Validation in the New Availability Group Wizard

Ensure that in the results view that each step of the wizard completed successfully or was skipped, and the database(s) joined successfully.

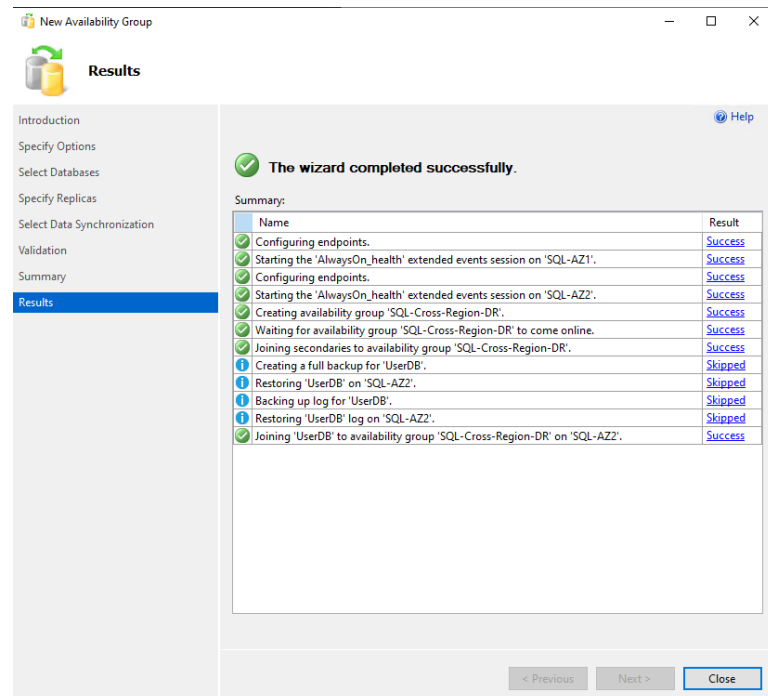


Figure 55. Skip initial data synchronization option



Using skip initial data synchronization, the availability group must already be created prior to the database being joined to it.

Once the database has been restored to the intended replica and is in a restoring state, it can be joined to the availability group. To join a database to an existing availability group using SQL Server Management Studio expand the availability groups until the availability databases are listed. Select a database and bring up the context menu for it, selecting **Join to Availability Group**.

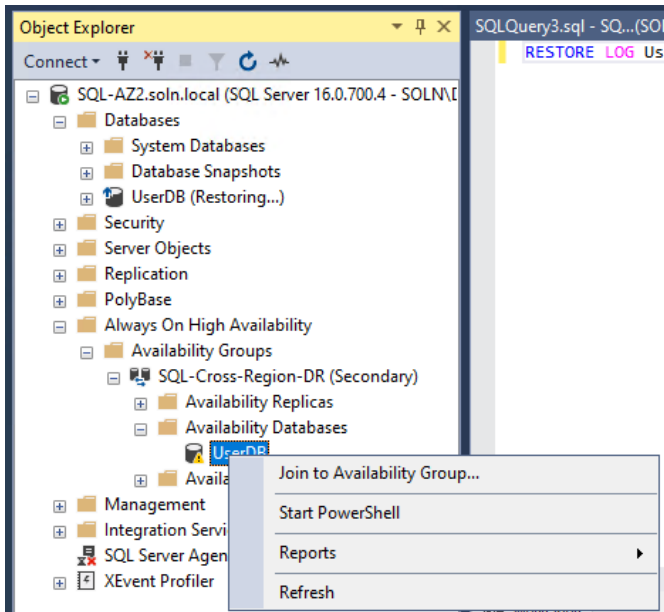


Figure 56. The Object Explorer in SQL Server Management Studio expanded to Availability Databases

In the Join Database to Availability Group dialog check **Stop executing after first error** and then select **OK**.

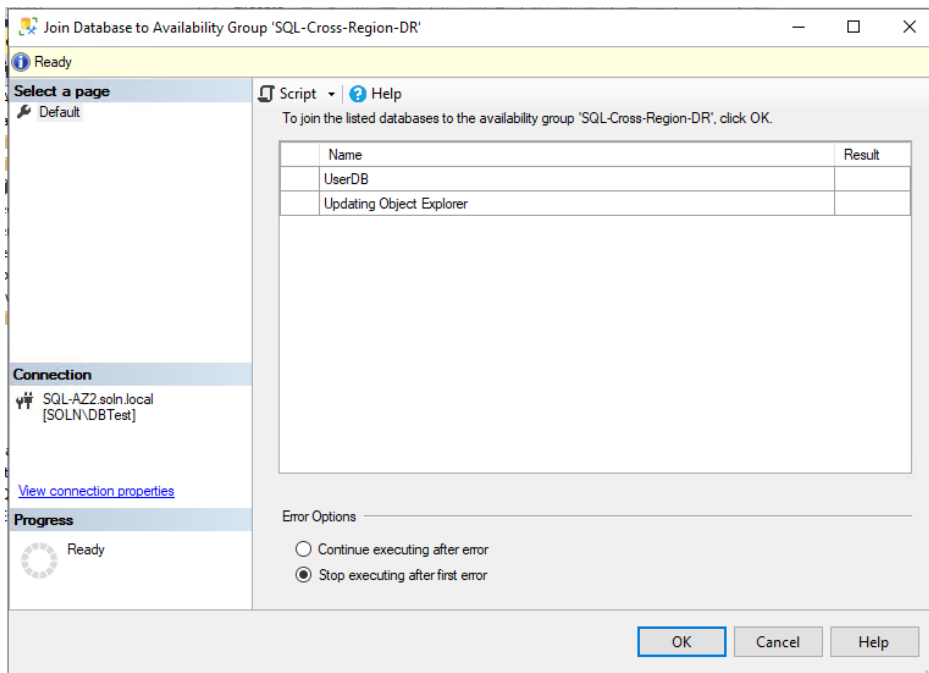


Figure 57. Join Database to Availability Group dialog



Once the database has been joined to the availability mode and is synchronized, this will be reflected in the availability groups on all replicas and set as “synchronized.”

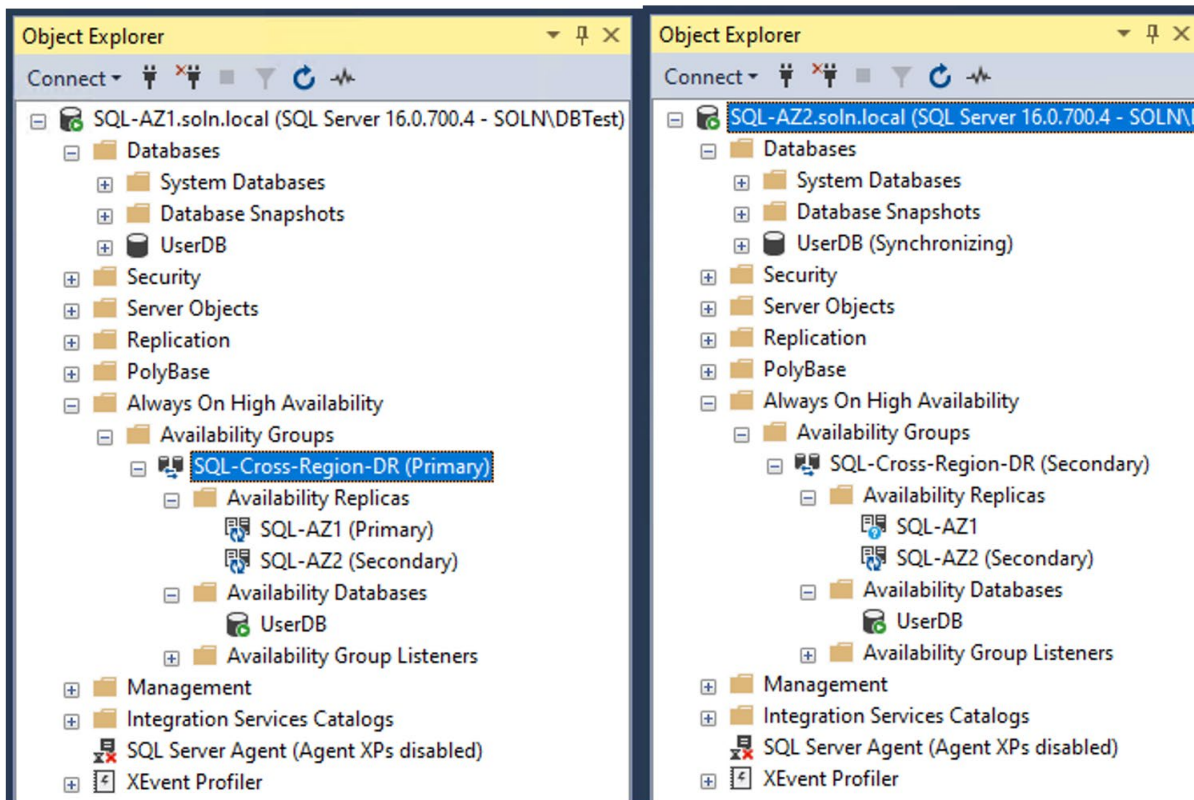


Figure 58. Two replicas in a synchronized state in SQL Server Management Studio

## Ransomware Remediation with SafeMode Snapshots

An additional layer of security can be added to Pure Cloud Block Store instances with the use of [SafeMode™](#).

SafeMode creates a separation of power between Pure Cloud Block Store administration functions and the eradication of data. SafeMode Snapshots apply to volume snapshots and can be combined with replication to offer superior resilience to data loss and integrity concurrently.

Volume Snapshots on Pure Cloud Block Store are immutable meaning, they cannot be changed. A new volume can be created from a snapshot, but the state of the snapshot will remain unchanged from the moment it was taken. This assists with ransomware remediation as data cannot be changed or affected by malicious software attempting to encrypt it and will thus provide a recovery point that organizations can rest assured has not been compromised.



## Conclusion

Pure Cloud Block Store provides several methods to implement efficient and cost-effective disaster recovery for Microsoft SQL Server. Different data services and software capabilities can be combined to achieve different recovery point objectives and mitigate against several disaster resolution scenarios. As organizations migrate to public cloud infrastructure, they need to be reassured that the tried and tested disaster recovery tools are available for their use, no matter where their database workloads are deployed. Pure Cloud Block Store provides this reassurance while also lowering the cost burden of data storage and movement.

The Pure Storage products and programs described in this documentation are distributed under a license agreement restricting the use, copying, distribution, and decompilation/reverse engineering of the products. No part of this documentation may be reproduced in any form by any means without prior written authorization from Pure Storage, Inc. and its licensors, if any. Pure Storage may make improvements and/or changes in the Pure Storage products and/or the programs described in this documentation at any time without notice.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. PURE STORAGE SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

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

[purestorage.com](https://purestorage.com)

800.379.PURE

