Fsas Technologies Inc.

P3AG-5642-06ENZ0

# ETERNUS AX series All-Flash Arrays, ETERNUS AC series All-Flash Arrays, ETERNUS HX series Hybrid Arrays

# SnapMirror Configuration and Best Practices Guide for ONTAP 9

# Table of Contents

# List of Figures

# List of Tables

# Preface

This document describes information and best practices related to configuring SnapMirror replication and has been updated to include changes introduced in ONTAP 9.16.1.

Sixth Edition
June 2025

## Trademarks

Third-party trademark information related to this product is available at:
https://www.fujitsu.com/global/products/computing/storage/eternus/trademarks.html

Trademark symbols such as ™ and ® are omitted in this document.

## About This Manual

### Intended Audience

This manual is intended for system administrators who configure and manage operations of the ETERNUS AX/AC/HX series, or field engineers who perform maintenance. Refer to this manual as required.

### Related Information and Documents

The latest information for the ETERNUS AX/AC/HX series is available at:
https://www.fujitsu.com/global/support/products/computing/storage/manuals-list.html

### Document Conventions

- Notice Symbols

The following notice symbols are used in this manual:

| | |
|---|---|
| Caution | Indicates information that you need to observe when using the ETERNUS AX/AC/HX series. Make sure to read the information. |
| Note | Indicates information and suggestions that supplement the descriptions included in this manual. |

# 1. Solution Overview

Businesses can use several approaches to increase data availability in the face of hardware, software, or site failures. Data protection is one of the most critical aspects of data management because any loss of data translates directly into lost revenue and time. Data protection is the process of taking data located in one location or repository and making a copy of it in a different location or repository to provide access and data resiliency.

Historically, data resiliency has supported the recovery of lost data from an archival medium (tape, drive, or the cloud), but mirroring that data to another location is becoming a more popular data resiliency mechanism because it supports lower data loss and faster recovery times than traditional archival mediums. SnapMirror technology offers a fast and flexible enterprise solution for mirroring or replicating data over LAN or WAN networks. The main advantages of using SnapMirror are as follows:

- **Robust enterprise technology**
  SnapMirror is a mature feature of ONTAP storage systems that has been enhanced and improved over time. SnapMirror can recover from update failures, use concurrent processes for replication processing, throttle the network bandwidth used for transfer operations, and much more.
- **Speed and efficiency**
  Block-level logical, incremental data transfer makes sure that only the data that has changed is sent to the destination replica. SnapMirror can reduce data bandwidth during replication by replicating data with all the storage efficiencies implemented on the source volume. It also uses network compression to possibly further compress data as it leaves the source and decompress it at the destination, thereby improving transfer performance.
- **Flexibility**
  SnapMirror supports different synchronization schedules on each protected volume to better meet data protection requirements. SnapMirror supports changing the direction of synchronization if there is a problem with the primary repository. It also supports a variety of replication topologies such as fan-out, in which a single volume replicates to many secondary systems, and cascade, in which the destination volume is itself synchronized to a tertiary system.
- **Testability**
  SnapMirror destination volumes can be instantly cloned as writable volumes in a space-efficient manner by using FlexClone technology, regardless of their size, and without needing to stop data replication from the source. This is invaluable for performing disaster recovery (DR) tests or supporting secondary applications that are dependent on the replicated data.
- **Failover and failback**
  If a DR system must be brought online, the SnapMirror relationship can be broken, making the destination volumes readable, writable, and ready for critical applications. After the DR event is resolved, SnapMirror supports resynchronization of all data changes made at the destination during the DR event back to the source volume and then reestablish the original SnapMirror relationship.
- **Ease of use**
  SnapMirror is integrated into ONTAP System Manager enabling storage administrators to perform operations with simplified workflows and wizard-guided walkthroughs. System Manager also supports monitoring and managing all SnapMirror replication relationships in one place.
- **Secure**
  SnapMirror relationships can be encrypted natively end-to-end using TLS 1.2 AES 256-bit encryption.
- **Cloud enabled**
  SnapMirror supports volume and storage virtual machine (SVM) replication to solutions in the cloud such as Amazon FSx for NetApp ONTAP and Cloud Volumes ONTAP offered by major cloud providers.

# Purpose and Intended Audience

This document is intended for individuals who administer, install, or support ONTAP systems and who intend to configure and use SnapMirror technology for data replication.

This document assumes that the reader understands the following processes and technologies:

- A working knowledge of ONTAP operations
- A working knowledge of features such as Snapshot copy technology, FlexVol or FlexGroup volumes, and FlexClone technology
- General knowledge of DR and data replication solutions
- Familiarity with the ETERNUS AX/AC/HX series Data Protection Power Guide in the manual site

# What's New for SnapMirror?

In an ongoing effort to provide better ONTAP features and capabilities, Table 1 lists the new features and capabilities for SnapMirror Asynchronous for volumes, and Table 2 lists the new features for SVM Disaster Recovery (SVM DR) for ONTAP 9.14.1.

Table 1 New features in SnapMirror asynchronous for individual volumes

| Functional area | New features |
|---|---|
| Disaster recovery | Ensuring that your DR plan works is critical. Typically, this involved a complex multi-step process that included stopping the production volumes, failing over to the DR site replica where the admin would perform specific data access tests. If those tests altered the data on the DR site replica, those changes might need to be resynchronized back to the production volume before reenabling the production site volume.<br><br>ONTAP System Manager now provides an automated workflow to perform DR site data integrity testing without the need to halt the production SnapMirror replication relationship. This ensures that production applications can continue to access critical data from the production site volume while DR data access testing is performed on the DR site replica. |
| Scalability | SnapMirror fan-out scalability has been increased to 20 relationships per source volume on high-end systems. |
| Storage efficiency | SnapMirror now supports quick-resync on FlexVol volumes containing snapshots with FlexClone volume dependencies. |

Table 2 New features in SnapMirror SVM disaster recovery

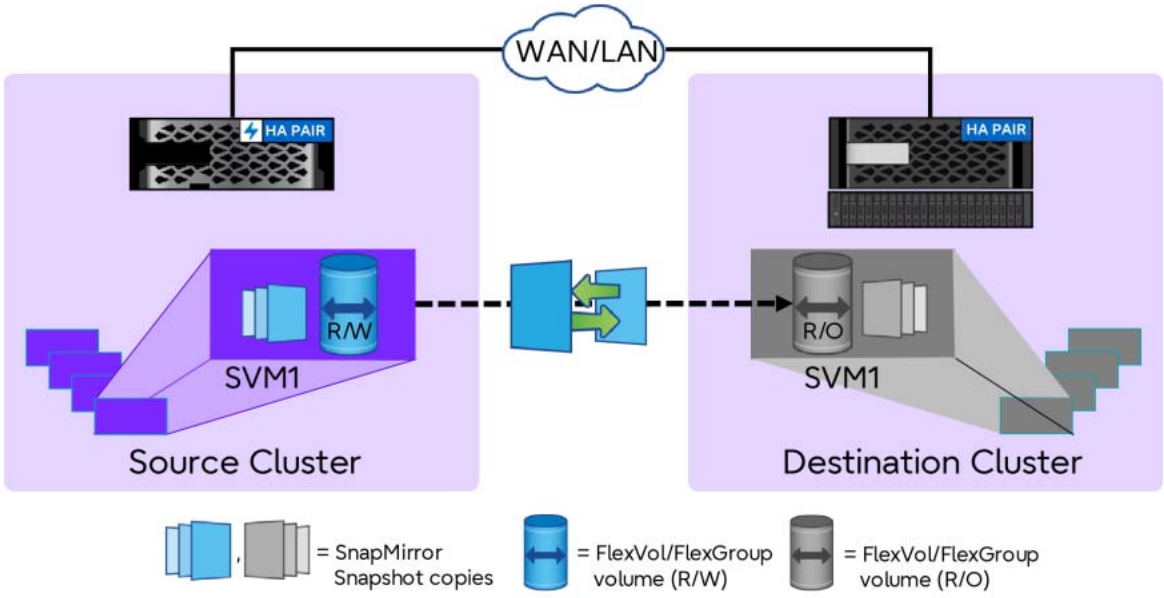
| Functional area | New features |
|---|---|
| Consistency groups | SnapMirror SVM DR now supports creating protection relationships for SVMs containing single-tier consistency groups (CGs). This is an extension of SnapMirror Asynchronous volume-scoped CG support. |
| Disaster recovery | Administrators can now track the progress of resync and reverse-resync operations using the `snapmirror show` command. |

# SnapMirror Overview

SnapMirror technology is a replication solution built into ONTAP for backup or archive and DR purposes. SnapMirror is configured through a data protection relationship between data volumes (FlexVol volumes or FlexGroup volumes) or consistency groups (CGs) consisting of FlexVol volumes on the primary and secondary storage systems. SnapMirror periodically updates the replica to keep it up to date with changes that have been written to the primary by using a schedule and replication policy.

This replica or mirror of enterprise data can be created on the same storage cluster or on a secondary storage cluster at a geographically remote site or in the cloud (by using Cloud Volumes ONTAP). Storage administrators can initiate a failover and serve data from the secondary site in the event of a catastrophe at the primary cluster site. After the error condition at the primary site is rectified, SnapMirror replicates any data changes back to the primary site volumes and restarts serving clients from the primary site. With SnapMirror, TCO can be reduced, making it easier to justify the DR investment by leveraging data in the DR site for other active business uses. For an overview of SnapMirror replication, see Figure 1.

Figure 1 SnapMirror replication overview




Data protection capabilities are an integral part of ONTAP. SnapMirror integrates tightly with Snapshot copy technology to create on-drive replicas or point-in-time, space-efficient copies of data quickly and efficiently.

Integrated data protection can be used to create a quickly accessible on-drive history of application-consistent snapshots that eliminate the concept of a traditional backup window. SnapMirror then replicates these snapshots to the destination, which can then be used for backup, DR, or test and development.

SnapMirror replication is efficient because it only replicates native data blocks that have been changed or added since the previous update. Additional efficiency is gained when SnapMirror is combined with storage efficiency technologies, such as compression and data deduplication technologies, resulting in significant telecommunication and storage capacity savings.

# Use Case Summary

## Near-Line Backup

One of the primary use cases for SnapMirror is data backup. Since the early days of enterprise data storage, data backup has been the purview of tape. Tape backup created some challenges for responsive data recovery, foremost of which is that restore operation simply failed and at the best of times it would take hours to recover from a disaster scenario.

SnapMirror can be used as a primary backup tool by replicating data within the same cluster or to remote targets. Using SnapMirror, storage administrators can restore single files or an entire storage configuration quickly.

## Disaster Recovery

SnapMirror technology also enables comprehensive disaster recovery (DR) plans. If critical data is replicated to a different physical location, a serious disaster does not have to cause extended periods of unavailable data for business-critical applications. Clients can access replicated data across the network until the recovery of the production site.

In the case of failback to the primary site, SnapMirror provides an efficient means of resynchronizing the DR site with the primary site by transferring only changed or new stored data back to the primary site by simply reversing the SnapMirror relationship. After all changes have been restored, the primary production site resumes normal application operations, and SnapMirror continues the transfer to the DR site without requiring another baseline transfer.

## DR Testing and Application Testing and Development

FlexClone technology quickly creates a copy of a SnapMirror destination volume enabling read-write access of the secondary copy to allow administrators to confirm if all the production data is available and verify that applications can operate normally from the DR site.

## Data Distribution and Remote Data Access

SnapMirror technology can be used to distribute large amounts of data throughout an enterprise, enabling access to data at remote locations. Remote data access provides faster access by clients in remote locations. It also allows more efficient and predictable use of expensive network and server resources because storage administrators can replicate production data at a specific time to minimize overall network utilization.

## Backup Offloading and Remote Tape Archiving

SnapMirror technology can also be used for backup consolidation and for offloading tape backup overhead from production servers. This approach facilitates centralized backup operations and reduces backup administrative requirements at remote locations. Snapshot technology eliminates the traditional backup window on the primary storage system.

# Unified Architecture Flexibility

SnapMirror provides data protection across a broad set of platforms to meet many different user requirements. SnapMirror started as a native feature of ONTAP for data center environments but has expanded in recent updates by embracing new cloud-centric platforms for private, hybrid, and public cloud deployments, as shown in Figure 2.

## Data Center Deployments

SnapMirror can be used on any ONTAP platform in the data center to meet a broad set of performance and scalability needs from ONTAP arrays. These deployments can be managed by using CLI, REST APIs, or the web-based System Manager graphical user interface.

## Private Cloud Deployments

SnapMirror private cloud deployments are architected similarly to data center deployments, with all the same platform flexibility, with the added management flexibility offered by REST APIs to support cloud-centric management and operational platforms such as Kubernetes and VMware vRealize.

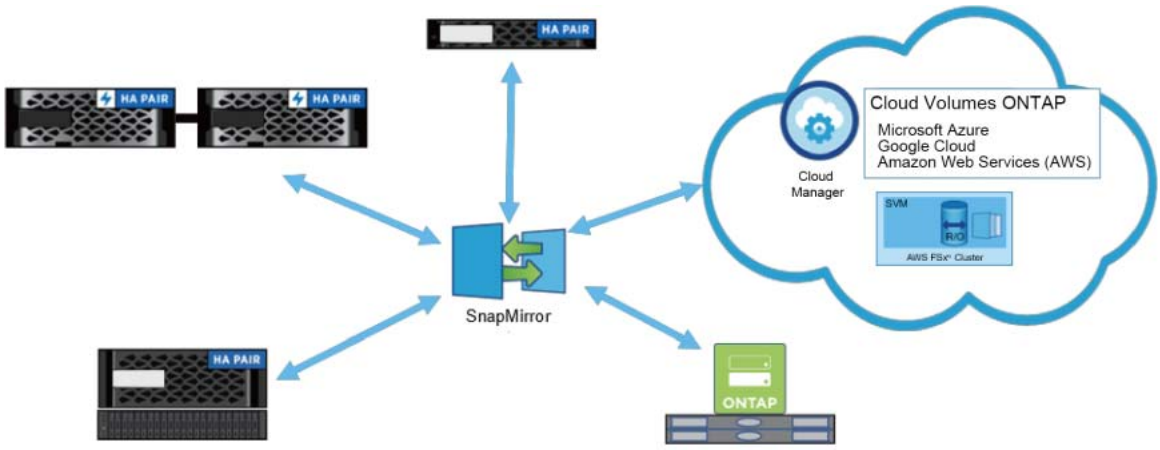
## Hybrid Cloud Deployments

SnapMirror provides support for data replication between ONTAP on-premises and ONTAP public-cloud implementations such as Cloud Volumes ONTAP deployments on major cloud provider environments using the Cloud Manager service and Amazon FSx for NetApp ONTAP service.

## Public Cloud Deployments

SnapMirror can support ONTAP deployments completely hosted in the cloud through solutions such as Cloud Volumes ONTAP deployments on major cloud provider environments, and Amazon FSx for NetApp ONTAP (Figure 2). These ONTAP solutions provide access to SnapMirror functionality to support primary and DR cloud-based deployments and can use System Manager, CLI, or REST APIs for management. In addition, through Cloud Manager, SnapMirror can be configured to replicate data between Cloud Volumes for ONTAP clusters within the same geographical region or in different regions for added protection.

Figure 2 Unified architecture flexibility



# 2. ONTAP Networking Basics

This section addresses only SnapMirror specific networking requirements.

## Common SnapMirror Networking Terms

Table 3 lists the basic terminology used in ONTAP and used by SnapMirror.

Table 3 ONTAP networking terminology

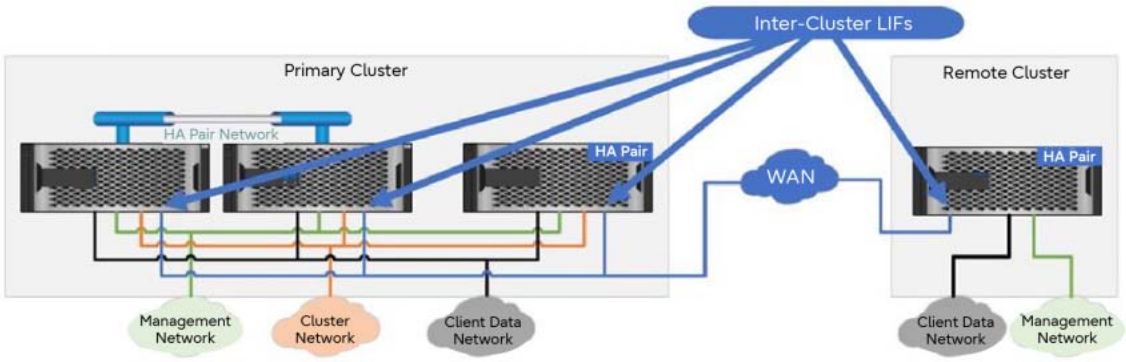
| Term | Definition |
|---|---|
| Node | A single device offering ONTAP storage services. A node can be standalone (ETERNUS AX/AC/HX series and ASA chassis, or Cloud Volumes ONTAP) or integrated into the same physical chassis as an HA pair. Typically, ONTAP storage clusters consist of one or more 2-node HA pairs, though some cloud deployments can be single-node deployments. |
| High-availability (HA) pair | Two ONTAP storage nodes configured in a pair for high availability. Each node can take ownership of storage resources from its paired node in the event of a node failure. |
| Cluster | One or more HA pairs that are interconnected and managed as a single storage solution. ONTAP clusters can consist of up to 6 SAN or 12 NAS HA pairs. |
| IPspace | An IPspace defines a distinct IP address space in which ONTAP arrays can participate. IPspaces offer a distinct routing table and enable ONTAP support for multitenancy deployments where tenants might use a common IP address range. |
| Broadcast domain | A broadcast domain is a group of physical network ports in the same IPspace that can communicate through a single layer-2 network protocol. |
| Intercluster network | The network used for communication and replication between different clusters. Intercluster networks are used by SnapMirror and other data protection solutions within the ONTAP environment. |
| Physical port | A physical network port such as `e0e` or `e0f` Ethernet or `0c` or `0e` FC ports. Physical ports can support Ethernet, FC, or provide unified protocol support. Physical ports host virtual ports and/or logical interfaces (LIFs). |
| Interface group (ifgrp) | A collection of physical ports combined to create one logical port used for link aggregation. An interface group can offer expanded throughput, redundancy or both. Interface groups comply with the IEEE 802.3ad and 802.1AX network standards. |
| Virtual LAN (VLAN) | A virtual LAN is an IEEE 802.1Q standard protocol that subdivides a physical network into distinct broadcast domains. As a result, traffic is completely isolated between VLANs unless a router (layer 3) is used to connect VLANs.<br>In ONTAP, VLANs subdivide a physical port into several separate virtual ports, allowing for one of the key components of our secure multitenant messaging—isolation of data. |
| Virtual port | A virtual port is a logical network interface that can come in various forms:<br>• Interface group<br>• VLAN |

| Term | Definition |
|---|---|
| Logical interface (LIF) | A LIF is an IP address or a worldwide port name (WWPN) that is associated with a port. LIFs can be attached to physical ports, interface groups or VLANs. LIFs have associated attributes such as failover rules, role and firewall rules. |
| Intercluster LIF | A LIF used to connect to an intercluster network.<br>An intercluster LIF must be created on each cluster node before a cluster peering relationship can be established. Intercluster LIFs can only fail over to ports in the same node. |
| Failover group | A failover group is a cluster-scoped list of physical ports within a cluster and in the same IPspace and same broadcast domain that can host a logical interface in the event of a physical port failure. |
| Cluster peer | Cluster peers are participants in an intercluster relationship. An intercluster peer relationship must be created before any data movement through ONTAP data protection services can be performed. The act of creating this intercluster relationship is called cluster peering. |
| Storage virtual machine (SVM) | An SVM is a logical storage server that provides data access to LUNs and/or a network-attached storage (NAS) namespace from one or more data LIFs. |
| SVM peer | SVM peers are participants in an SnapMirror relationship. An inter-SVM peer relationship must be created before any data movement through ONTAP data protection services can be performed. The act of creating this inter-SVM relationship is called SVM peering. |
| Volume | A data storage construct that refers to ONTAP native FlexVol volumes or FlexGroup volumes. |
| Consistency group | An ONTAP container that associates multiple volumes such that they can be simultaneously protected in an application consistent way. SnapMirror uses consistency groups to take application consistent snapshots for replication purposes. |

# SnapMirror Networking Requirements

There are multiple types of networks in ONTAP, as shown in Figure 3. Cluster management, inter-node communications and client data access each have dedicated networks. SnapMirror uses the intercluster LIF type to connect all nodes in all clusters that will participate in data replication by using SnapMirror and replicate data between SnapMirror endpoints.

Figure 3 Intercluster LIFs and relationship to other ONTAP networks




# Intercluster Networking Requirements

Intercluster LIFs must meet the following requirements:

- All intercluster LIFs for a given SnapMirror relationship must be in the same IPspace. Different IPspaces can be used to peer with different clusters by using different intercluster LIFs.
- At least one intercluster LIF must be configured on every node in the source cluster and every node in the destination cluster participating in the SnapMirror relationship.
- The IP addresses assigned to intercluster LIFs can reside in the same subnet as a cluster's data LIFs or in a different subnet.
- Intercluster LIFs can be assigned to ports, interface groups, or VLANs.
- Intercluster LIFs can reside on the same physical ports as data LIFs.
- Intercluster LIFs are node scoped. Therefore, when the port hosting an intercluster LIF fails, the LIF can only fail over to another intercluster-capable port on the same node, as defined by the LIF's failover policy.
- All intercluster LIFs must have consistent settings (the same maximum transmission units [MTUs], flow control, TCP options, and so on) using the same IPspace.
- SnapMirror replication over an FC network is not available in ONTAP.

For additional information regarding intercluster networking, see Cluster and SVM peering overview with the CLI.

# Intercluster Multipathing

It is recommended to configure multipathing for intercluster LIFs used for a SnapMirror relationship. Multipathing provides data path redundancy, bandwidth aggregation, or both depending on the multipath configuration used. This section discusses the use of interface groups (interface groups) and failover groups by highlighting the benefits of each and when to use each to meet the solution requirements.

The decision on which type of multi-pathing solution to use will depend on the capabilities provided by the network switches to which the cluster nodes connect. Interface groups require switches that support the IEEE 802.3ad link aggregation protocol. Failover groups are completely managed by ONTAP and do not require any additional switch features to implement.

## Intercluster Multipathing Using Interface Groups

Interface groups use the IEEE 802.3ad link aggregation (LAG) protocol and the 802.1AX link aggregation control protocol (LACP) standards to provide logical network interfaces that consist of two or more physical ports. Interface groups can host LIFs or VLANs to provide path redundancy or path redundancy with bandwidth aggregation depending on selected interface group type. It is important that all ports added to an interface group have the same link-speeds, duplex settings, allowed VLANs (if it is layer-2), native VLAN, and so on settings.

To learn more about interface groups and how to create and manage them, see Combine physical ports to create interface groups.

The following recommendations are suggested when using interface groups for SnapMirror intercluster replication:

- Use a dedicated broadcast domain (VLAN) for SnapMirror to simplify identification of upstream links in the network for troubleshooting and added network security.
- Use interface groups that are configured as type `-mode multimode_lacp` if the switch infrastructure supports LACP.
- If switches are capable, always configuring ISLs as multi-chassis control (vPC, MLAG, and so on) version and connect interface group ports on each node to different physical switches.
- Select constituent ports from different network interface controllers (NICs) that have the same physical network characteristics such as throughput, frame size, duplex settings, etc. and are in the same broadcast domain.
- Create interface groups with either 2, 4, or 8 constituent physical ports. This ensures the hash algorithm has an opportunity to evenly distribution connections between links. The hash algorithm always returns a result (bucket) between 0 and 7 with each bucket assigned to one physical link in the interface group at interface group initialization. Sizing the interface group with a number of physical links that are evenly divisible by eight ensures that each link will get the same number of hash buckets. Assuming there is variability in the input values used by the hash algorithm, network sessions should get evenly distributed.
- If source or destination clusters are small or single node (as may be possible in a cloud deployment) consider creating multiple intercluster LIFs per interface group to allow the hash algorithm to more evenly distribute connections across all physical ports. A maximum of eight intercluster LIFs per node can be configured.
- Use the `port` distribution function for intercluster LIFs. This provides the hash algorithm with the broadest set of values to use for the hash algorithm resulting in a more even distribution of connections mapped to the eight hash result buckets.

### Best Practice

When using interface groups, use the following parameters to the `network port interface group create` command:

- Use `-mode multimode_lacp`
- Use `-distr-func port`

## Intercluster Multipathing Using Failover Groups

Failover groups provide a non-switch dependent method of link failure protection. By default, every cluster creates a failover group for each broadcast domain. Custom failover groups can be defined to optimize the failover behavior of a LIF.

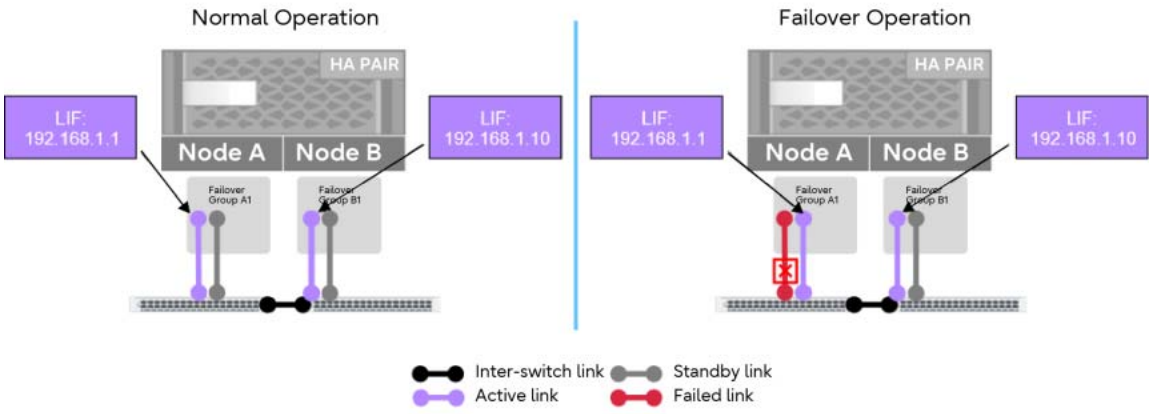
Intercluster LIFs can be assigned to failover groups that span cluster nodes, but intercluster LIFs will only failover to ports within the local node of the home port for that LIF. Intercluster LIFs are defined with the `-role intercluster` or `-service-policy default-intercluster` parameters to the `interface create` command.

Depending on how LIFs are assigned to the ports in a failover group, the failover group can operate in one of two modes:

- Failover mode
- Multiplexing mode

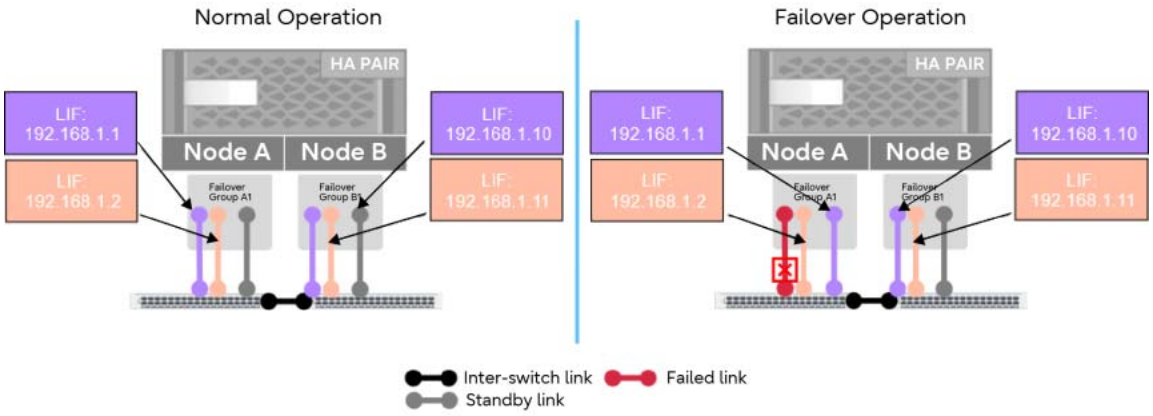
Once the failover group is created, LIFs are assigned to individual ports in the failover group. The port a LIF is initially assigned to is considered that LIF's home port. All other ports in the failover group are considered standby ports for that LIF. If a single intercluster LIF is assigned to a failover group, then the failover group operates in failover mode, as shown in Figure 4.

Figure 4 Failover group used in failover mode



If more than one intercluster LIF is assigned to ports within the failover group, then those LIFs enable users of those LIFs – such as SnapMirror – to multiplex across the active LIFs within that failover group. It is recommended that these LIFs be assigned to different ports within the failover group to provide additional bandwidth for SnapMirror replication, as shown in Figure 5. For each active LIF in the failover group, the port initially assigned to that LIF is considered its home port and all other ports – even if hosting other LIFs – is considered that LIF's standby ports. If a LIF's active link goes down, it is possible that a port in the failover group could host multiple LIFs.

Figure 5 Failover group used in multiplex mode




## Best Practice

- Create custom failover groups on each node specifically for intercluster LIFs.
- Use a dedicated broadcast domain for SnapMirror to simplify identification of upstream links in the network for troubleshooting and added network security.
- Create at least two intercluster LIFs and assign one LIF to different port in the failover group.
- Use the following parameters when creating intercluster LIFs:
  - `-service-policy default-intercluster`
  - `-data-protocol none`
  - `-failover-group <custom failover group name>`

## Share or Dedicate Failover Groups?

Intercluster LIFs can be assigned to failover groups that contain other types of LIFs or can be assigned to ports in failover groups specifically defined for intercluster communications. Table 4 highlights some considerations for each.

Table 4 Shared versus dedicated interface groups

| Shared interface group | Dedicated interface group |
|---|---|
| • Replication performance is not critical.<br>• Higher bandwidth (25–100GbE) ports are available and SnapMirror replication workload is not enough to saturate links.<br>• Node interface available bandwidth is not the bottleneck...the WAN link is.<br>• Replication requirements support a high RPO (for example, I only need to replicate once a day).<br>• Data change rate on the source volume is low.<br>• Port availability on each node is low. | • DR is a primary goal for SnapMirror.<br>• Lower bandwidth node ports that might easily get saturated with SnapMirror replication workload.<br>• Replication requirements need to support a very small RPO<br>(dictating the need for different MTU).<br>• Data change rate on the source volume is very high<br>(dictating the need for different maximum MTU).<br>• Port availability on each node supports having a dedicated failover group. |

### Best Practice

Although it is not required, the following are recommendations for using failover groups for SnapMirror:

- Configure at least two intercluster LIFs per node using a failover group containing at least two physical ports with the same network characteristics (speed, duplex mode, and so on)
- If the network utilization generated by the data protocols (SMB, NFS, iSCSI, or NVMe) is above 50%, then a dedicated failover group for intercluster communication is recommended.
- When using dedicated failover groups for intercluster communication, consider the following:
  - Remove the physical ports from other failover groups, including the default failover group associated with the cluster broadcast domain.
  - Use a dedicated broadcast domain for additional security.
  - Consider setting MTU size to jumbo frames to maximize network efficiency (consult network engineer for viability of this option).
- Use a standardized naming convention for intercluster LIFs. For example, `node_name_icl#` or `node-name-ic#`, depending on preference.

# How SnapMirror Uses Intercluster LIFs to Replicate Traffic

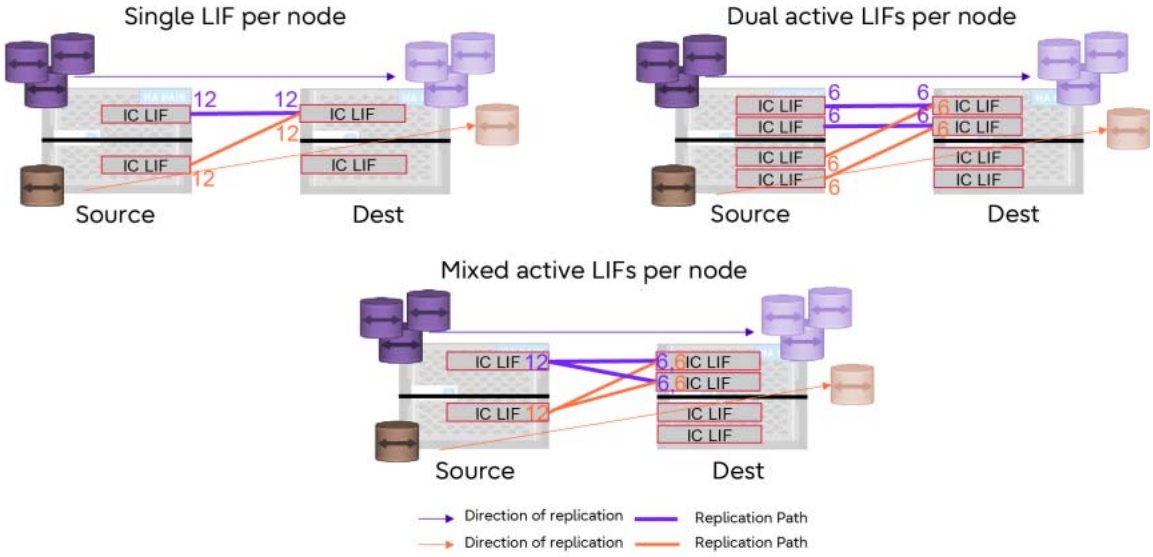
ONTAP provides a centralized network session manager to provide optimized network traffic management for various functions involving intercluster communications such as SnapMirror. For SnapMirror, this cluster session manager provides up to 12 TCP send streams and 12 TCP receive streams for each source-destination node peer relationship. For discussion purposes, only the 12 send sessions will be used, but there are also 12 receive sessions that are managed similarly.

These 12 sessions are controlled using the following rules:

- A set of 12 sessions is created between each source-destination node interaction between clusters that have been peered.
- On each node, the 12 sessions are distributed across the available IC LIFs for that source-destination node relationship.
- All volumes replicating between a given source-destination node pair share the 12 sessions.

Using these rules, Figure 6 illustrates how these 12 TCP sessions are distributed.

Figure 6 TCP stream distribution examples




**Note**

It is not possible to select a specific LIF pair to use for a replication event. All session management is performed automatically by the ONTAP session manager.

# Firewall Requirements

SnapMirror uses the typical socket, bind, listen, and accept sequence on a TCP port. The firewall and the intercluster firewall policy must allow the following protocols:

- TCP ports 11104 and 11105 for intercluster control and data respectively
- TCP port 10000 for NDMP backup services
- Optionally, TCP port 443 (HTTPS) in each direction between the intercluster LIFs for managing ONTAP arrays using System Manager or REST APIs

**Note**

HTTPS is not required to set up cluster peering using the CLI.

# 3. Replication Basics

## Licensing

A SnapMirror license is required for each cluster that will participate in SnapMirror replication. This license is included in the Data Protection Bundle and ONTAP One license. If the SnapMirror source and destination are on different clusters, a SnapMirror license must be enabled on each cluster. All nodes in each cluster must have a license.

Starting with ONTAP 9.14.1P6 for the ETERNUS AC series controllers and ONTAP 9.13.1 for all other ETERNUS AX/HX series controllers, SnapMirror is included as part of ONTAP One license sold with every ONTAP controller.

**Note**

- The license must be present on both source and destination.
- ONTAP One is available for clusters operating older versions of ONTAP. Contact your sales representative for more information.

## SnapMirror Asynchronous Technology

SnapMirror replicates data from a source volume or consistency group in a cluster to an equivalent volume or consistency group in a destination cluster by using snapshots. SnapMirror performs the following operations:

**Procedure ▶▶▶**

1. A snapshot of the data on the source is created. This snapshot will have a SnapMirror label of `sm_created`.
2. The snapshot is copied to the destination during baseline synchronization. This process creates a destination that is online, read-only, and contains the same data as the source at the time of the most recent common snapshot.
3. Ongoing replication is performed based on the SnapMirror policy schedule. What gets replicated is based on the SnapMirror policy type:
   - **Async-Mirror**
     A new `sm_created` snapshot is created and replicated to the destination cluster.
     If `mirror-all-snapshots` is part of the async-mirror policy rules, all snapshots – manually created and policy created – are replicated to the destination cluster regardless of SnapMirror label.
   - **Vault**
     Snapshots created manually or using a snapshot policy that have a SnapMirror label matching any SnapMirror policy rules will be replicated to the destination cluster.
   - **Mirror-Vault**
     A new `sm_created` snapshot is created and replicated along with any snapshots that have a SnapMirror label matching any SnapMirror policy rules will be replicated to the destination cluster.

◀◀◀

When a SnapMirror data protection relationship is established (`snapmirror initialize`), the destination volumes are identical replicas of the source volumes, including snapshots, volume settings, and ONTAP space efficiency features that are accessible as a read-only volumes. Breaking the SnapMirror relationship (`snapmirror break`) makes the destination volumes writable and is used to perform a failover for async-mirror and mirror-vault relationships to recover from a failure on the source volume. SnapMirror is sophisticated enough to identify the data changed at the failover site and replicating that changed data back to the primary system. The original SnapMirror relationship can then be reestablished (`snapmirror resync`).

SnapMirror may use either of two different replication engines to create replicas – LRSE (XDP) and BRE (DP). Although both engines operate at the volume level, they have different characteristics:

- **Logical replication with storage efficiency (LRSE)**
  LRSE, also called as XDP, uses block-level metadata and knowledge of the file system to determine differences between snapshots at the indirect pointer level. LRSE organizes the transfer of data from the source to the destination in two streams.
  - The data stream consists of data blocks that are transferred with specific volume block number (vvbn#) within the destination volume. This vvbn# helps identify the block number at which the data is stored on the source FlexVol volume, but without specifying a file context. On the destination, the data is written to the data warehouse (DW) file with a file block number (fbn#) which corresponds to the vvbn#.
  - The user files are transferred by reference using the user file inodes, which share blocks with the data warehouse file and do not use buffer trees that require parsing to reach a specific object. LRSE makes explicit requests to the block-sharing infrastructure of the DW blocks (the donors) with user files (recipients) while replication transfer is in progress.

  The mirror has a structure of logical block pointers to the original data set that has a completely different on-drive physical layout relative to the source.

  LRSE preserves space efficiency over the wire and on the destination when replicating data in storage-efficient source volumes. Storage efficiency is an important part of LRSE because features such as block sharing and compression allow a volume to effectively hold far more data than the space used. This efficiency must be preserved during replication to avoid the replica growing to an intolerably large size, not to mention the time needed to transfer it. LRSE also allows enablement of storage efficiency features on the secondary, independent of the primary storage settings. For more information, see Use deduplication, data compression and data compaction to increase storage efficiency – overview.

  In addition to asymmetric storage efficiency on primary and secondary storage, LRSE enables version flexibility where the destination version can be different than the source. It also supports asymmetric snapshots where the destination can support a greater number of snapshots than the source. All the files and directories in the source file system are created in the destination file system. Therefore, SnapMirror can replicate data between a storage system running an older version of ONTAP and a storage system running a newer version. This approach allows reduced downtime because the controllers on either side can be upgraded at any time while reducing overhead and managing complex topologies (fan-in, fan-out, and cascade).

  The SnapMirror relationship is created with `-type XDP` using the SnapMirror policy type `async-mirror`, `vault` or `mirror-vault`.

- **(DEPRECATED) Block Replication Engine (BRE)**
  BRE, also called as DP, replicates the on-drive layout from a source volume to a destination volume either as a whole or as an incremental update with 4K blocks. The BRE uses knowledge of the file system to determine differences between snapshots at the block-allocation level and replicates only those changed blocks. Therefore, the copy of data created on the destination has an identical structure of physical block pointers to the original data set on the source. The BRE replicates volumes using volume block (vvbn#) read and write operations.

  This SnapMirror relationship is created with `-type DP` using the SnapMirror policy type `async-mirror`. BRE only supports volume mirroring use cases and does not support vault use cases.

  **Caution**

  - As of 9.11.1, BRE (DP) SnapMirror relationships are used for legacy data protection policies only. All newly created SnapMirror relationships default to using LRSE (see section Logical replication with storage efficiency).
  - As of ONTAP 9.12.1, BRE is no longer supported and systems that still host BRE SnapMirror relationships will not be able to upgrade to ONTAP 9.12.1 until those BRE (DP) relationships are converted to LRSE (XDP). For more information, see Convert an existing DP-type relationship to XDP.

  The performance characteristics of LRSE are similar to those of BRE because the replication engine only transfers the difference between two snapshots from the primary to the secondary. This incremental-only transfer leads to savings in terms of storage and network bandwidth. Starting with ONTAP 9.3, SnapMirror XDP mode replaces SnapMirror DP mode as the SnapMirror default. More details can be found at XDP replaces DP as the SnapMirror default.

  SnapMirror can also be integrated with SnapCenter to replicate application consistent snapshots, such as those used for enterprise database applications. snapshots are created in coordination with the application to guarantee that no in-flight I/O operations cause inconsistencies in the snapshot. After creating an application consistent snapshot, SnapCenter can then trigger a SnapMirror replication of these application consistent snapshots to the secondary storage system.

## Unified Data Protection

The XDP relationship type enables SnapMirror to provide a single, unified replication engine to perform both mirroring (for disaster recovery) and vaulting (for drive-based backup and archiving) use cases. Moving forward, the use cases previously referred to as SnapVault will be subsumed into the SnapMirror standard nomenclature as SnapMirror with a policy type of `vault` and `mirror-vault`.

Overall, unified replication with SnapMirror provides powerful data management capabilities for virtualization, protecting critical data while providing the flexibility to move data between locations and storage tiers, including cloud service provider hosted SnapMirror instances. The relationship is created with relationship type of XDP, and a policy type of `mirror-vault`. We provide a predefined SnapMirror policy called `Asynchronous` that uses these settings. The policy can always be modified to include custom rules for backing up specific snapshots. In addition, this functionality reduces the number of secondary snapshots needed on the destination.

The major benefits for SnapMirror unified replication are as follows:

- Only one baseline copy of a volume is needed to perform mirroring and archiving to the secondary volume.
- Less network traffic is required between the primary and secondary (a single baseline plus fewer snapshots over time).
- The flexibility to replicate between storage systems running different ONTAP releases. XDP relationships are version independent, while DP relationships are not.

- To avoid corrupting replication from primary to secondary, unified replication makes it possible to recover the primary volume from available snapshots.
- An XDP relationship removes the limitation of the destination controller requiring an ONTAP major version number equal to or later than the major version of the source controller.

In System Manager for ONTAP 9.8 and later, SnapMirror Asynchronous uses the Asynchronous (mirror-vault) protection policy by default. A custom policy is required to modify any policy parameters.

The following example shows how unified replication can be configured with the Asynchronous policy from the CLI:

```
cluster02::> snapmirror create -source-path snap_src1:Source -destination-path
svm_dst1:Source_dest -type XDP -policy Asynchronous
```

● Best Practice

Use a mirror-vault relationship whenever possible. There is no performance or space impact in using this policy type even when archive is not a requirement of your data protection plan. By doing so, it is easy to modify your data protection plan to include archive in the future without requiring additional volume baseline images.

To create a custom policy, navigate to Protection > Overview > Local Policy Settings > Protection Policies > Add (Figure 7 and Figure 8).

Figure 7 SnapMirror custom protection policy create

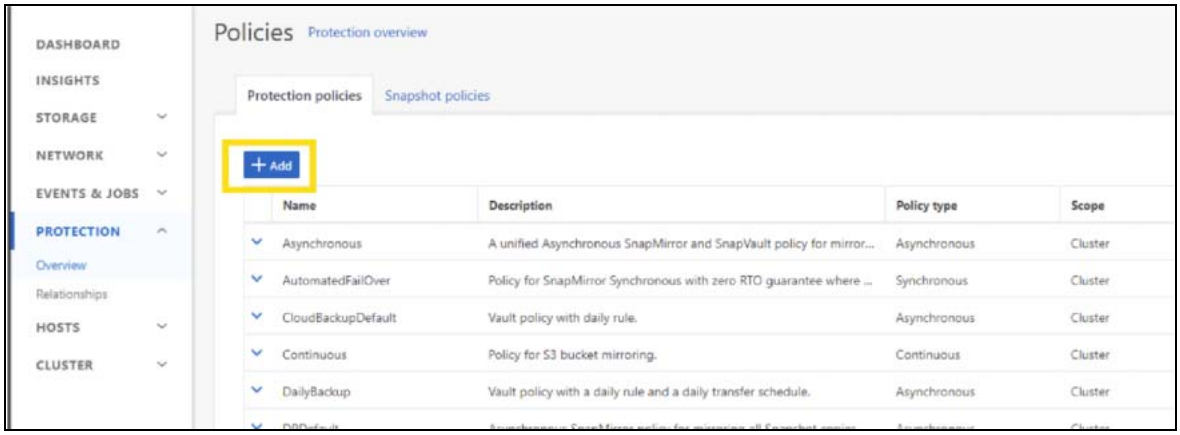
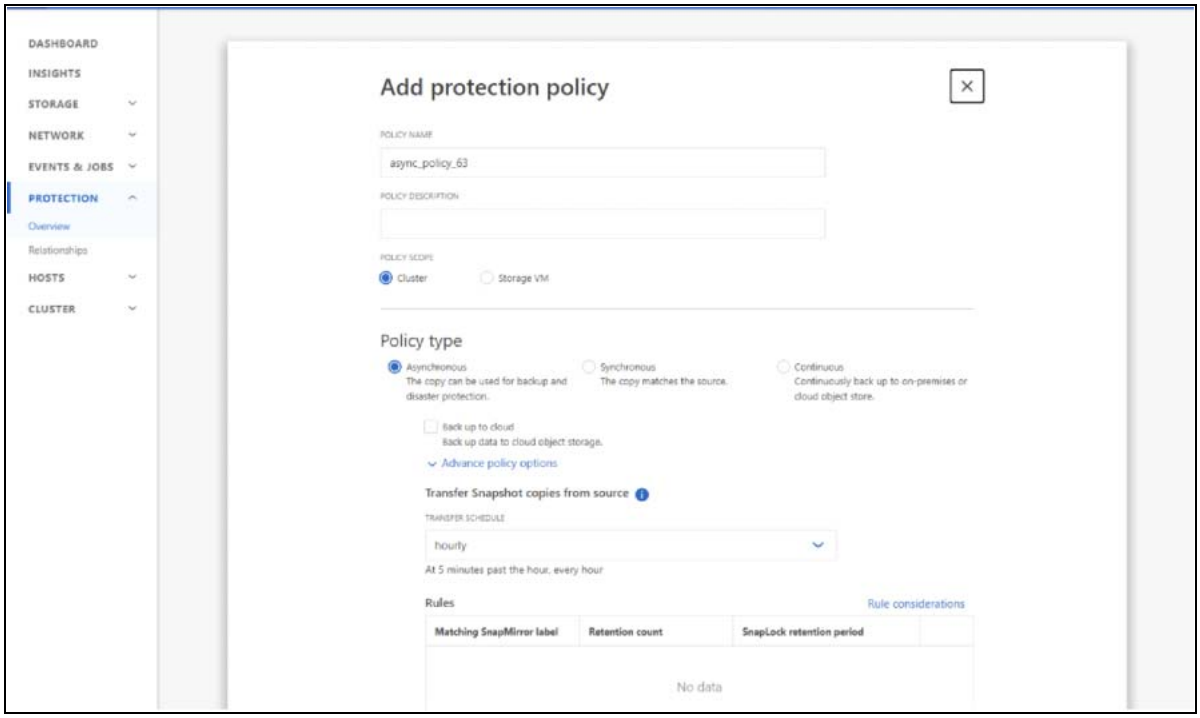


Figure 8 Add custom SnapMirror protection policy




## Load-Sharing Mirror

Every SVM in a NAS environment has a unique namespace. The SVM root volume is the entry point to this namespace hierarchy. For clusters consisting of two or more HA pairs, load-sharing mirrors (LSMs) of SVM root volume should be considered to ensure the namespace remains accessible to clients in the event of both nodes of an HA pair failing. Load-sharing mirrors are not suitable for clusters consisting of a single HA pair and are not suitable for a MetroCluster environment.

**Caution**

- Load-sharing mirrors have been deprecated for data volumes and are only supported for SVM root volumes.
- SnapMirror load-sharing mirrors are only capable of supporting NAS (CIFS/NFSv3). Load-sharing mirrors do not support NFSv4 clients or SAN client protocol connections (FC, FCoE, or iSCSI). However, NFSv4 and load-sharing mirrors can coexist in the same environment.

● Best Practice

LSMs are of limited value in modern ONTAP cluster solutions due to the multiple hardware redundancies integrated into the hardware and cluster software. For most deployments, LSMs are not recommended, rather it is recommended to use SVM disaster recovery as an alternative solution.

For additional information concerning load-sharing mirrors, see Managing SnapMirror Root Volume Replication.

# SnapMirror Synchronous (SM-S)

SM-S is an easy-to-use solution that replicates data synchronously between a source volume and a destination volume over a LAN or metropolitan area network (MAN). SM-S provides high data availability and rapid DR for business-critical applications without loss of data due to primary site or cluster failure.

SM-S uses a different mechanism than SnapMirror Asynchronous for transferring volume data and is covered in "SnapMirror Synchronous Configuration and Best Practices" in the manual site.

# SnapMirror for Cloud Volume Platforms

SnapMirror can be used to provide asynchronous data protection of on-premises or cloud-hosted ONTAP volumes and SVMs. Working in cooperation with Google, Amazon, and Microsoft, we offer a broad range of ONTAP cloud offerings that can be used as an overall data management fabric. The following sections describe each offering that is supported for use with SnapMirror.

## Cloud Volumes ONTAP

Cloud Volumes ONTAP enables customers to use performance and cost-optimized cloud storage in conjunction with on-premises ONTAP data management. Cloud Volumes ONTAP offers a full array of ONTAP APIs and data management services built upon the native cloud-provider compute, storage, and network products. Cloud Volumes ONTAP offerings are available from Amazon Web Services (AWS), Microsoft Azure, and Google Cloud.

To use a Cloud Volumes ONTAP instance as a SnapMirror endpoint, it must first be created using the cloud provider management tools or Cloud Manager. Once created, a SnapMirror asynchronous relationship can be created between an on-premises ONTAP cluster and the Cloud Volumes ONTAP instance using any ONTAP management interface – CLI, System Manager, REST APIs or BlueXP.

## Amazon FSx for NetApp

Amazon FSx for NetApp ONTAP is a native AWS storage service that offers fully managed ONTAP as a service that can be administered through BlueXP or the AWS suite of management tools such as AWS Management Console, AWS CLI, or AWS REST APIs. After an Amazon FSx for NetApp ONTAP cluster is created, it can be used as a source or destination cluster in a SnapMirror Asynchronous relationship with other Amazon FSx for NetApp ONTAP instances, Cloud Volumes ONTAP instances, or on-premises ONTAP clusters.

# 4. SnapMirror Configuration

## Cluster Peering

The cluster peering feature of ONTAP allows administrators of independent clusters to establish a peer relationship between them. SnapMirror leverages the ONTAP core networking infrastructure and depends on intercluster LIFs and cluster peering to transfer replication data between volumes or SVMs. For complete details on the cluster peering process, please review ONTAP Cluster and SVM peering overview in the online documentation.

- Before setting up cluster peering, confirm that the connectivity, port, IP address, subnet, firewall, and cluster-naming requirements are met. Cluster peering requirements include the following:
- The time on the clusters must be in sync to within 300 seconds (five minutes) for peering to be successful. Cluster peers can be in different time zones.
- At least one intercluster LIF must be created on every node in the cluster.
- Every intercluster LIF in the local cluster IPspace must be able to communicate with every intercluster LIF in the remote cluster IPspace.
- Every intercluster LIF requires an IP address dedicated to intercluster replication.
- The MTU settings of the ports must be consistent. The default value of 1,500 is correct for most environments.
- All paths on a node used for intercluster replication should have equal performance characteristics.

**Caution**

SnapMirror does not support network address translation (NAT).

Starting with ONTAP 9.7, cluster peering uses encrypted communication, which means any SnapMirror relationship that is created uses an additional layer of security through TLS encryption.

### Best Practice

- If using `hosts` files of ONTAP instead of DNS for name resolution of cluster nodes, the name and IP address of the source system must be in the `hosts` file of the destination system and vice versa.
- Use at least one intercluster IP address from each node in the remote cluster, so that the peer relationship remains available in the event of a single node failure.

## SVM Peering

SVM peering connects two SVMs to allow replication to occur between them, which requires cluster peering first. SVM peering enables granularity of access or the delegation of various replication operations to the SVM admin.

**Caution**

A peer relationship is not required to mirror data between two SVMs in the same cluster or between two volumes in the same SVM.

For more information about SVM peering, see ONTAP Cluster and SVM peering overview in the online documentation.

# SnapMirror Data Protection Relationship

A relationship created between the source object (for example, a FlexVol or FlexGroup volume, or a consistency group) in primary storage and the destination object in secondary storage is called a data protection relationship.

SnapMirror relationships have the following characteristics:

- SnapMirror relationships are created and managed on the destination cluster.
- SnapMirror relationship transfers are triggered by the scheduler on the destination cluster.
- Destination volumes must be created with the volume type of DP (`-type DP`) for SnapMirror initialization to succeed. After volumes have been created, the volumes' type cannot be changed.
- Destination volumes in a SnapMirror relationship are read-only until a manual failover is initiated by a storage administrator. Use the `snapmirror break` command to initiate a failover to the secondary copy and make the destination volume writable. The `snapmirror break` command must be performed separately for each volume or consistency group.
- The destination volumes can be mounted into an SVM namespace as read-only, but only after the initial transfer is complete.
- Destination volumes in a SnapMirror relationship between two separate clusters cannot be mounted in the same NAS namespace as the source volumes. However, the destination volumes in a SnapMirror relationship configured within a cluster can be mounted in the same namespace as the source volume if both the source and destination volumes exist in the same SVM. However, they cannot be mounted to the same mount point.
- LUNs contained in mirror destination volumes can be mapped to initiator groups (igroups) and connected to clients. However, the client must be able to support connection to a read-only LUN.
- SnapMirror relationships can be managed using the ONTAP CLI, ONTAP System Manager, REST API, Active IQ Unified Manager, or BlueXP.
- If an in-progress transfer is interrupted by a network outage or aborted by an administrator, a subsequent restart of that transfer can automatically continue from a saved restart checkpoint.
- Both the source and destination SVM can have different language types, but the source and destination volumes must have the same language type.
- The destination aggregate must have adequate space to host the replicated volumes and any vault snapshots retained per the configured protection policy.
- If the SnapMirror source is a consistency group, then the destination cluster must be running an ONTAP version of 9.13.1 or later.

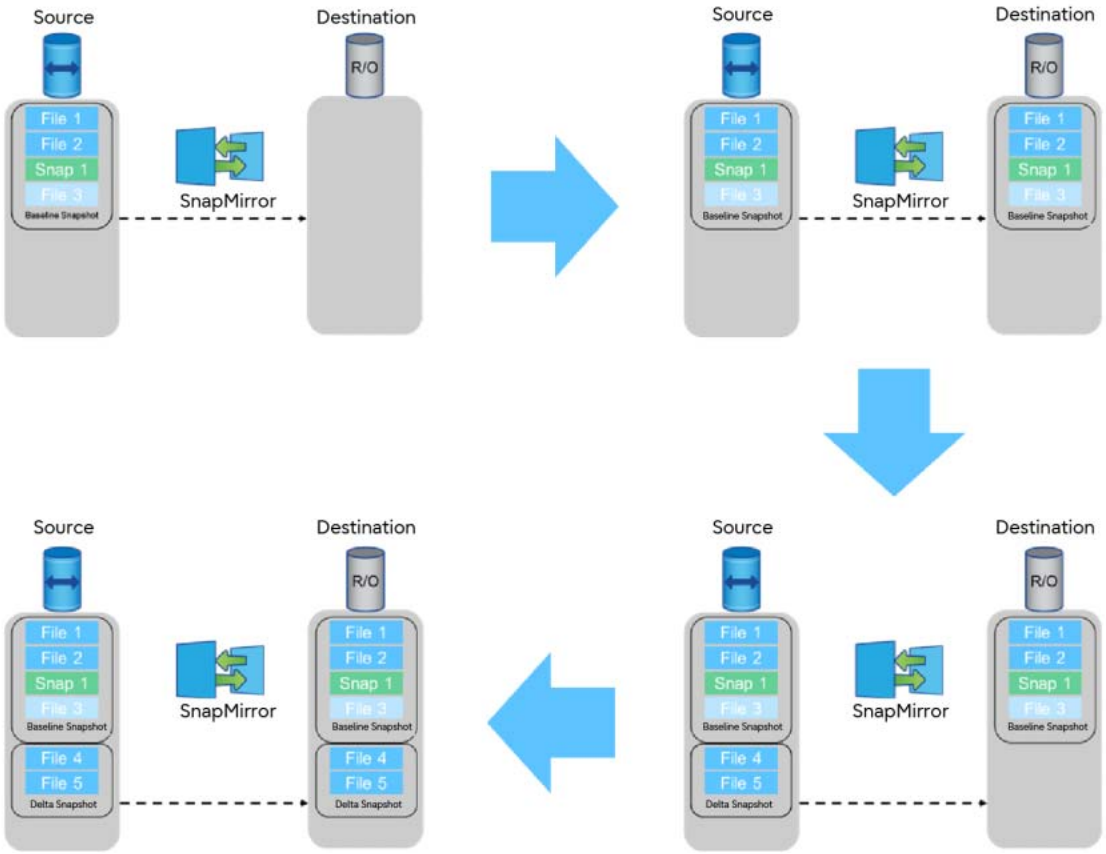
## How SnapMirror Replicates Volumes and Volume Changes

SnapMirror Asynchronous replicates data using native ONTAP snapshots to create a static image of the volume and/or changes since the last SnapMirror snapshot. This ensures that only data not previously replicated to the DR volume gets replicated.

After a SnapMirror replication relationship is created, the relationship must be initialized. During this initialization phase, an initial snapshot is created of the volume. Since this snapshot is the very first SnapMirror related snapshot created on the volume, this snapshot captures the entire volume's contents. This snapshot – and thus the entire volume's contents – are replicated to the DR volume. This "baseline" is used as the starting point for scheduled SnapMirror snapshots subsequently created as part of its replication policy schedule.

Once the baseline data has been replicated, SnapMirror goes into a replication mode where it regularly creates new snapshots – based on the requirements described in the SnapMirror replication policy – on a regular schedule defined by the `-schedule` parameter supplied during the SnapMirror creation. This is illustrated in Figure 9.

Figure 9 SnapMirror Asynchronous replication in action




When a snapshot is replicated to the DR volume, that latest SnapMirror snapshot will be located on both the source volume and the DR volume. This is called the newest common snapshot (NCS). The NCS is used for the next scheduled SnapMirror replication, and this continues as required by the supplied schedule.

When a failure event occurs on the source volume, any data written to the DR site volume during the event will need to be replicated to the production site volume once the failure event is rectified. The NCS is used as a reference point to how far back data must be resynchronized after a failure event once the production site is back in operation.

## Best Practices

- Do not reuse a destination volume from a previously existing SnapMirror relationship. Always use a newly created volume to start a new SnapMirror relationship.
- Do not delete snapshots that SnapMirror creates in the source volume before copying the data to the destination. The most recent SnapMirror snapshot is referred to as the newest common snapshot (NCS). Incremental changes to the destination depend on the NCS. If SnapMirror cannot find the required snapshot on the source, it cannot perform incremental changes to the destination.
- To avoid unnecessary autosize changes for the data protection destination FlexGroup volume, make sure to specify that the total size of the FlexGroup volume at time of volume creation is the same as the primary FlexGroup volume.

- Do not restrict or take the destination volume offline while SnapMirror is configured to transfer. Taking the destination offline prevents SnapMirror from performing updates to the destination.
- The number of constituents on the primary FlexGroup directly relates to the number of aggregate entries that need to be specified in the `-aggr-list` parameter. When choosing which aggregate is specified in the `-aggr-list`, make sure that the aggregates have enough space for the constituents.
- To ensure an efficient SnapMirror operation of FlexGroup volumes, make sure that for each FlexGroup hosted by the same set of aggregates, use a different order in the `-aggr-list` parameter. One recommendation is to rotate the aggregates in a round-robin fashion.
- Make sure that the size of each destination constituent is such that it can ingest data from the primary constituent. Otherwise, SnapMirror operations fail when they run out of space.

## Replication Intervals

SnapMirror updates must establish a communication session between the source and destination nodes, creating and deleting snapshots, and determining which blocks of data to send to the destination. SnapMirror uses the built-in ONTAP scheduler functionality to manage how often data is replicated. The period between replication events should be considered the volume's or consistency group's recovery point objective (RPO), or the acceptable amount of data loss in the event of a catastrophic failure of the primary volume or consistency group. Although the ONTAP scheduler supports creating schedules that run every minute, SnapMirror has minimum supported replication intervals listed in Table 5.

Table 5 Minimum RPO for SnapMirror replication of FlexVol volumes and FlexGroup volumes

| SnapMirror replication source | Minimum supported RPO |
|---|---|
| FlexVol volumes | 5 minutes |
| FlexGroup volumes | 30 minutes |
| Consistency groups | 30 minutes |

## Fan In and Fan Out

It is possible for a volume or consistency group in the source cluster to be replicated to multiple different destinations (fan-out) or volumes or consistency groups from different source SVMs can be replicated to separate SVMs in a single destination cluster (fan-in), as shown in Figure 10. Table 6 lists the fan-out limits for SnapMirror.

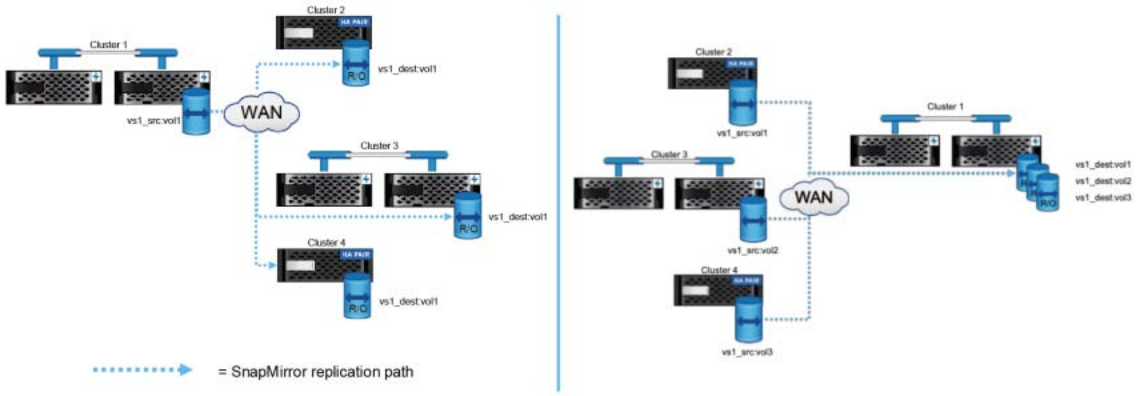
Table 6 Maximum fan out for a SnapMirror source volume

| ONTAP version | Maximum fan-out |
|---|---|
| ONTAP 9.11.1 and earlier | 8 |
| ONTAP 9.12.1 and later | 8 or 16 |
| ONTAP 9.14.1 and later | 8 or 20 |

**Note**

Fan out to greater than eight destination volumes from a single source volume depends on type of source and destination controllers.

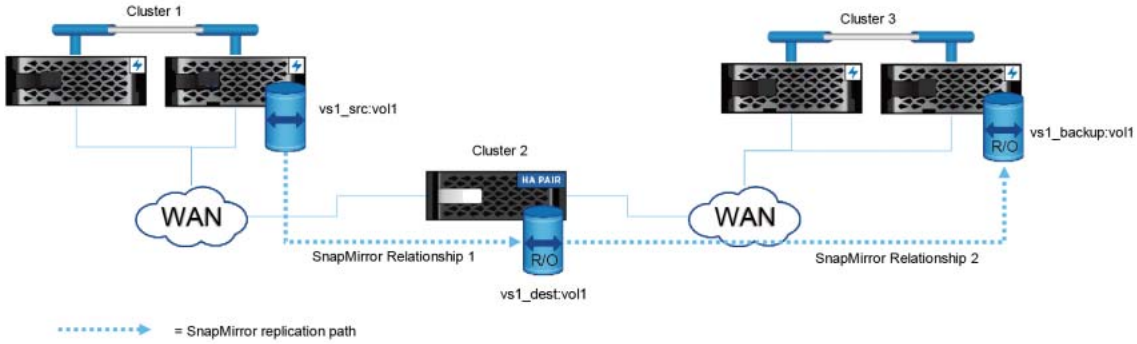
Figure 10 SnapMirror fan-out and fan-in




## Cascade Relationship

SnapMirror can replicate data from a SnapMirror destination to another destination system. Therefore, a system that is a destination for one SnapMirror relationship can act as the source for another SnapMirror relationship. This is useful for distributing data from one site to multiple sites. This is referred to as cascading. In a cascade topology, intercluster networks need to be created between the primary and secondary clusters and between the secondary and tertiary clusters. An intercluster network between the primary and the tertiary cluster is not needed. While there is no limit on the number of "hops" for a sing e source volume, there may be practical limits in the ability to schedule each downstream replication without being interfered with by an upstream replication. An example cascade configuration with two hops is shown in Figure 11.

Figure 11 SnapMirror cascade




The function of this deployment is to make a uniform set of data available on a read-only basis to users from various locations throughout a network and to enable the updating of that data uniformly at regular intervals.

Snapshots behave in the following ways:

- SnapMirror creates a soft lock on the snapshot of the source volume (tag) or consistency group.
- The destination system carries an extra snapshot.

SnapMirror supports a different relationship for each leg of the cascade. Cascade relationships can also be created using either SnapMirror asynchronous or synchronous relationships. For cascade relationships that include SnapMirror synchronous, the synchronous relationship must be the first relationship in the cascade chain. All other SnapMirror relationships in the cascade chain must be asynchronous.

## Cascade Relationships that Use Only SnapMirror Asynchronous Relationships

For any relationship in the cascade chain, that relationship can be a mirror or a vault. Therefore, in long-chain cascades, the following combinations of data protection relationships can be used between any three clusters in the chain as described in Table 7:

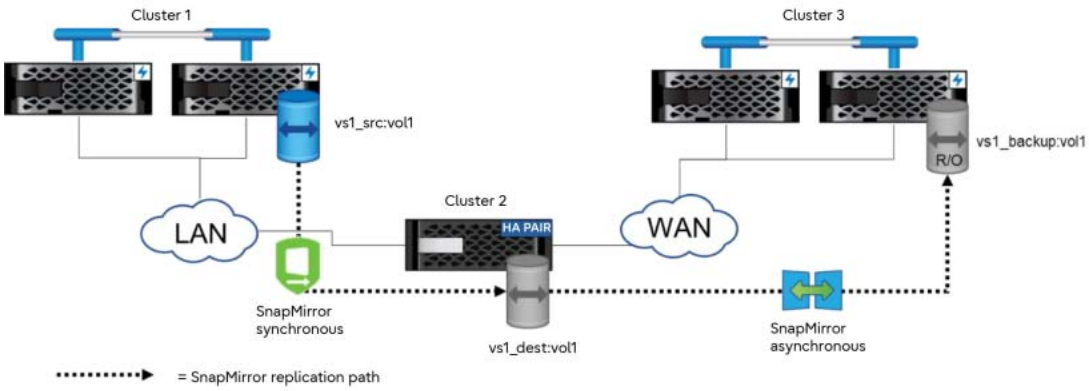
Table 7 Cascade relationships types between any three volumes in a SnapMirror relationship

| Cluster 1-Cluster 2 | Cluster 2-Cluster 3 | Description |
|---|---|---|
| Mirror | Mirror | One cluster has a mirror relationship with a second cluster and that second cluster has a mirror relationship with a third cluster. |
| Mirror | Vault | One cluster has a mirror relationship with a second cluster and that second cluster has a vault relationship with a third cluster. |
| Vault | Vault | One cluster has a vault relationship with a second cluster and that second cluster has a vault relationship with a third cluster. |

## Cascade Relationships that Use SnapMirror Asynchronous and Synchronous Relationships

For cascade relationships can be configured with the root source volume relationship that uses a SnapMirror synchronous relationship and all subsequent SnapMirror relationships in the cascade chain using SnapMirror asynchronous as shown in Figure 12.

Figure 12 Cascade relationships involving SnapMirror synchronous




The asynchronous relationship that uses the SnapMirror synchronous DP volume as the source volume (the secondary relationships in any long-chain cascade that includes SnapMirror synchronous) should be configured as an async-mirror with MirrorAllSnapshots. The resulting volume on the tertiary site (Cluster 3 in Figure 12) will depend on the ONTAP version used in the clusters involved in the SnapMirror cascade.

- ONTAP 9.13.1 and older
  The replica volume on Cluster 3 will have snapshots available that have been exported by the SnapMirror synchronous common snapshot (exported snapshot). This means that while the DP volume on Cluster 2 will have all application created snapshots from Cluster 1 on Cluster 2, those snapshots that were created/replicated since the last exported snapshot will not be visible. The asynchronous relationship between Cluster 2 and Cluster 3 will not see these snapshots until the next scheduled SnapMirror synchronous relationship's common snapshot creation.

- ONTAP 9.14.1 and later
  The replica volume on Cluster 3 will reflect all application created snapshots that have been created/replicated from Cluster 1 to Cluster 2 regardless of the current exported common snapshot.

Any cascade relationships downstream from Cluster 3 can be created as described for those cascades not involving synchronous relationships.

### Best Practice

- Keep the number of cascades as small as possible to ensure that each leg has the time to perform its scheduled replication without interfering with upstream or downstream relationships. Make sure that all the legs of the cascaded relationships complete successfully to make sure that subsequent SnapMirror updates do not fail with a `snapmirror busy` error.
- If using a combination mirror-vault, fan-out, or cascade deployment, keep in mind that updates fail if a common snapshot does not exist on the source and destination volumes or consistency groups. Use the snapmirror snapshot-owner create command to preserve a labeled snapshot on the secondary in a mirror-vault deployment. Doing so provides a common snapshot for the update of the vault relationship.
- If the cascade includes an initial SnapMirror synchronous relationship and the clusters are running ONTAP 9.13.1 or earlier, ensure that the synchronous relationship's common snapshot schedule matches the schedule duration of the asynchronous relationship between Cluster 2 and Cluster 3. This will ensure that both relationships will result in all snapshots being replicated to all three clusters.

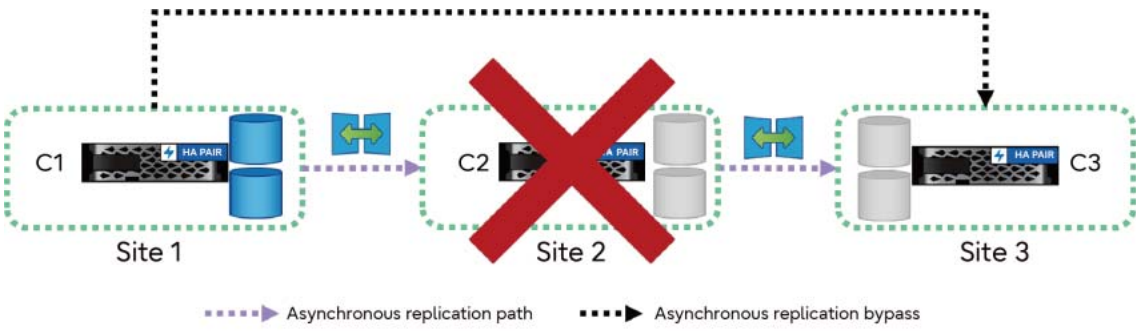
**Note**

Creation of SnapMirror cascade relationships is not supported by ONTAP System Manager and must be created using the ONTAP CLI or REST APIs.

## Dual-Hop Volume SnapMirror

This configuration involves volume or consistency group SnapMirror replication among three clusters, which consists of a chain of relationships in which a source volume or consistency group is mirrored to a secondary volume or consistency group, and the secondary volume or consistency group is mirrored to a tertiary volume or consistency group. If the secondary volume or consistency group becomes unavailable, SnapMirror can synchronize the relationship between the primary and tertiary volumes or consistency groups without performing a new baseline transfer if there can be a common snapshot copy identified on primary and tertiary clusters (Figure 13).

Figure 13 Bypassing a leg in a SnapMirror cascade infrastructure



# Protection Policies

ONTAP relies on policies to dictate when to create snapshots and how many copies are retained and/or replicated as a part of the relationship. Additionally, the policy helps determine the type of relationship that exists between the source and destination. SnapMirror replication limits the contents of the baseline transfer to the snapshot created by SnapMirror at initialization. At each update, SnapMirror creates another snapshot of the source. It then transfers the changes from this snapshot and the previously transferred snapshot (mirror) along with any new snapshots that have labels matching the labels defined in the snapshot rules defined on the SnapMirror policy if the policy is a vault policy. ONTAP comes with several predefined protection policies.

## SnapMirror Asynchronous Policy types

Each SnapMirror protection policy (standard or custom) is one of several different policy types. The policy types are described in Table 8.

Table 8 SnapMirror policy types

| Policy type | Definition |
|---|---|
| Async-mirror | The async-mirror policy type is used by SnapMirror to transfer snapshots of two types:<br>• Those source snapshots created by the SnapMirror engine. (rule label = `sm_created`)<br>• All source snapshots created on the volume from other Snapshot copy policies or manually created on the source volume. (rule label = `all_source_snapshots`)<br>**Caution**<br>No other rules can be applied to a protection policy of type async-mirror. |
| Vault | The vault policy type is used by SnapMirror to copy only source volume snapshots that match the labels provided with each rule.<br>The Vault policy type replaces SnapVault functionality.<br>This policy does not replicate snapshots created by the SnapMirror relationship (label = `sm_created`). |
| Mirror-vault | The mirror-vault policy type is used by SnapMirror to transfer snapshots created by the SnapMirror replication engine (snapshot label = `sm_created`) and any desired source snapshots that match the snapshot label defined in each rule.<br>The mirror-vault protection policy can have multiple rules defined to match the desired data protection requirements. |

## Standard Asynchronous Protection Policies

The following protection policy variants are available for the SnapMirror Asynchronous relationship creation:

- **Asynchronous**
  This is an asynchronous SnapMirror policy of the mirror-vault policy type. As such, the asynchronous protection policy is for mirroring the latest file system on an hourly schedule and retaining seven snapshots with the daily label and retaining 52 snapshots with the weekly label from the source volume or consistency group. This is the default policy for the SnapMirror relationship creation and replaces DPDefault used as the default protection policy in previous versions of ONTAP (Figure 14).

  This policy consists of the following settings:
  - Policy type is `mirror-vault`.
  - Create snapshot is set to `true`.
  - There are three rules:
    - `sm_created` replicates the changes on the source volume since the last SnapMirror generated snapshot.
    - `daily` keeps seven daily snapshots.
    - `weekly` and 52 weekly snapshots.

Figure 14 SnapMirror asynchronous policy definition

```
cluster_dst::> snapmirror policy show -policy Asynchronous -instance

                     Vserver: vs0
       SnapMirror Policy Name: Asynchronous
       SnapMirror Policy Type: mirror-vault
                 Policy Owner: cluster-admin
                  Tries Limit: 8
              Transfer Priority: normal
    Ignore accesstime Enabled: false
      Transfer Restartability: always
  Network Compression Enabled: false
              Create Snapshot: true
                      Comment: A unified Asynchronous SnapMirror and SnapVault policy for
mirroring the latest active file system and daily and weekly Snapshot copies with an hourly
transfer schedule.
        Total Number of Rules: 3
                  Total Keep: 60
       Transfer Schedule Name: hourly
                    Throttle: unlimited
 Rules:
 SnapMirror Label             Keep Preserve Warn Schedule Prefix
 ---------------------------- ---- -------- ---- -------- ----------
 sm_created                      1 false       0 -        -
 daily                           7 false       0 -        -
 weekly                         52 false       0 -        -
```

- **DailyBackup**
  This policy is an asynchronous SnapMirror policy of the vault policy type. It can be used to create an archive of the source volume snapshots that have a label = daily and will retain the last seven snapshots on the data protection volume. The Create Snapshot field is set to False, which indicates that this policy does not create any SnapMirror-related snapshots on the source volume. The included protection policies listed here are considered legacy policies but can be very useful in supporting additional data protection strategies. In System Manager, these policies are only shown if the Show Legacy Policies option is selected when creating or editing a protection relationship (Figure 15).

  This policy consists of the following settings:
  - The policy type is set to `vault`.
  - The Create Snapshot value is set to `false`, which means the policy does not create a snapshot when an update is triggered.
  - There is one rule:
    - Keep seven daily snapshots.

Figure 15 DailyBackup asynchronous SnapMirror policy definition

```
cluster_dst::> snapmirror policy show -policy DailyBackup -instance

                     Vserver: vs0
      SnapMirror Policy Name: DailyBackup
      SnapMirror Policy Type: vault
                Policy Owner: cluster-admin
                 Tries Limit: 8
           Transfer Priority: normal
   Ignore accesstime Enabled: false
     Transfer Restartability: always
  Network Compression Enabled: false
              Create Snapshot: false
                      Comment: Vault policy with a daily rule and a daily transfer schedule.
      Transfer Schedule Name: daily
                    Throttle: unlimited
       Total Number of Rules: 1
                  Total Keep: 7
      Transfer Schedule Name: daily
                    Throttle: unlimited
 Rules:
 SnapMirror Label             Keep Preserve Warn Schedule Prefix
 ---------------------------- ---- -------- ---- -------- ----------
 daily                           7 false       0 -        -
```

- **DPDefault**
  This is an asynchronous SnapMirror policy for mirroring all snapshots and the latest active file system from the source to the destination.

> **Note**
>
> This policy is available for legacy SnapMirror relationships. Administrators should use the newer MirrorAllSnapshots policy for new SnapMirror relationships.

With this configuration, the SnapMirror engine creates a snapshot and then replicates the difference between the new SnapMirror snapshot and the previous one and all the other snapshots. If the relationship is being initialized, then a snapshot is taken and everything before it is replicated. After the update is complete, the older snapshot is deleted, leaving just one common SnapMirror snapshot in place (Figure 16).

This policy consists of the following settings:

- Policy type is `async-mirror`.
- Create snapshot is set to `true`.
- There are two rules:
  - `sm_created` replicates the changes on the source volume since the last SnapMirror-generated snapshot.
  - `all_source_snapshots` keep one copy of each unique snapshot from the source volume.

Figure 16 DPDefault asynchronous SnapMirror policy definition

```
cluster_dst::> snapmirror policy show -policy DPDefault -instance

                     Vserver: vs0
       SnapMirror Policy Name: DPDefault
       SnapMirror Policy Type: async-mirror
                Policy Owner: cluster-admin
                 Tries Limit: 8
            Transfer Priority: normal
    Ignore accesstime Enabled: false
      Transfer Restartability: always
  Network Compression Enabled: false
              Create Snapshot: true
                      Comment: Asynchronous SnapMirror policy for mirroring all
Snapshot copies and the latest active file system.
        Total Number of Rules: 2
                   Total Keep: 2
       Transfer Schedule Name: -
                     Throttle: unlimited
 Rules:
 SnapMirror Label              Keep Preserve Warn Schedule Prefix
 ----------------------------- ---- -------- ---- -------- ----------
 sm_created                       1 false       0 -        -
 all_source_snapshots             1 false       0 -        -
```

- **MirrorAllSnapshots**
  This also is an asynchronous policy for mirroring all snapshots and the latest active file system from the primary to the secondary. This policy is similar to DPDefault (Figure 17).

  This policy consists of the following settings:
  - Policy type is `async-mirror`.
  - Create snapshot is set to `true`.
  - There are two rules:
    - `sm_created` replicates the changes on the source volume since the last SnapMirror-generated snapshot.
    - `all_source_snapshots` keep one copy of each unique snapshot from the source volume.

Figure 17 MirrorAllSnapshots asynchronous SnapMirror policy definition

```
cluster_dst::> snapmirror policy show -policy MirrorAllSnapshots -instance

                    Vserver: vs0
      SnapMirror Policy Name: MirrorAllSnapshots
      SnapMirror Policy Type: async-mirror
                Policy Owner: cluster-admin
                 Tries Limit: 8
           Transfer Priority: normal
   Ignore accesstime Enabled: false
     Transfer Restartability: always
 Network Compression Enabled: false
             Create Snapshot: true
                     Comment: Asynchronous SnapMirror policy for mirroring all snapshots
                              and the latest active file system.
       Total Number of Rules: 2
                  Total Keep: 2
      Transfer Schedule Name: -
                    Throttle: unlimited
Rules:
SnapMirror Label     Keep  Preserve Warn Schedule Prefix
----------------     ----  -------- ---- -------- ------
sm_created              1  false       0 -        -
all_source_snapshots    1  false       0 -        -
```

- **MirrorLatest**
  This is an asynchronous policy for mirroring the latest active file system from the primary to the secondary. Using this policy, the SnapMirror engine creates a snapshot and then replicates the difference between the new SnapMirror snapshot and the previous one. If the relationship is being initialized, then a snapshot is taken and everything before it is replicated. After the update is complete, the older snapshot is deleted, leaving just one common SnapMirror snapshot in place (Figure 18).

  This policy consists of the following settings:
  - Policy type is `async-mirror`.
  - Create snapshot is set to `true`.
  - There is one rule:
    - `sm_created` replicates the changes on the source volume since the last SnapMirror-generated snapshot.

Figure 18 MirrorLatest asynchronous SnapMirror policy definition

```
cluster_dst::> snapmirror policy show -policy MirrorLatest -instance

                     Vserver: vs0
       SnapMirror Policy Name: MirrorLatest
       SnapMirror Policy Type: async-mirror
                 Policy Owner: cluster-admin
                  Tries Limit: 8
            Transfer Priority: normal
    Ignore accesstime Enabled: false
      Transfer Restartability: always
  Network Compression Enabled: false
              Create Snapshot: true
                      Comment: Asynchronous SnapMirror policy for mirroring the latest active
file system.
               Total Number of Rules: 1
                   Total Keep: 1
       Transfer Schedule Name: -
                     Throttle: unlimited
 Rules:
 SnapMirror Label              Keep Preserve Warn Schedule Prefix
 ----------------------------- ---- -------- ---- -------- ----------
 sm_created                       1 false       0 -        -
```

- **MirrorAndVault**
  This is a unified SnapMirror policy for mirroring the latest active file system and daily and weekly snapshots. Starting with ONTAP 9.5, MirrorAndVault is the new default policy when no data protection mode is specified or when XDP mode is specified as the relationship type (Figure 19).

  This policy consists of the following settings:
  - Policy type is `mirror-vault`.
  - Create snapshot is set to `true`.
  - There are three rules:
    - `sm_created` replicates the changes on the source volume since the last SnapMirror-generated snapshot.
    - `daily` keeps seven daily snapshots.
    - `weekly` keeps 52 weekly snapshots.

Figure 19 MirrorAndVault asynchronous SnapMirror policy definition

```
cluster_dst::> snapmirror policy show -policy MirrorAndVault -instance

                     Vserver: vs0
       SnapMirror Policy Name: MirrorAndVault
       SnapMirror Policy Type: mirror-vault
                 Policy Owner: cluster-admin
                  Tries Limit: 8
              Transfer Priority: normal
    Ignore accesstime Enabled: false
      Transfer Restartability: always
  Network Compression Enabled: false
              Create Snapshot: true
                      Comment: A unified Asynchronous SnapMirror and SnapVault policy for
mirroring the latest active file system and daily and weekly Snapshot copies.
        Total Number of Rules: 3
                   Total Keep: 60
       Transfer Schedule Name: -
                     Throttle: unlimited
 Rules:
 SnapMirror Label             Keep Preserve Warn Schedule Prefix
 ---------------------------- ---- -------- ---- -------- ----------
 sm_created                      1 false       0 -        -
 daily                           7 false       0 -        -
 weekly                         52 false       0 -        -
```

- **Unified7year**
  This policy provides a `monthly` rule to transfer monthly snapshots and retain them for seven years (Figure 20).

  This policy consists of the following settings:
  - Policy type is `mirror-vault`.
  - Create snapshot is set to `true`.
  - There are four rules:
    - `sm_created` replicates the changes on the source volume since the last SnapMirror-generated snapshot.
    - `daily` keeps seven daily snapshots.
    - `weekly` keeps 52 weekly snapshots.
    - `monthly` keeps 84 monthly (7 years) snapshots.

Figure 20 Unified7year asynchronous SnapMirror policy definition

```
cluster_dst::> snapmirror policy show -policy Unified7year -instance

                     Vserver: vs0
      SnapMirror Policy Name: Unified7year
      SnapMirror Policy Type: mirror-vault
                Policy Owner: cluster-admin
                 Tries Limit: 8
           Transfer Priority: normal
   Ignore accesstime Enabled: false
     Transfer Restartability: always
  Network Compression Enabled: false
              Create Snapshot: true
                      Comment: Unified SnapMirror policy with 7year retention.
       Total Number of Rules: 4
                  Total Keep: 144
      Transfer Schedule Name: -
                    Throttle: unlimited
 Rules:
 SnapMirror Label              Keep Preserve Warn Schedule Prefix
 ----------------------------- ---- -------- ---- -------- ----------
 sm_created                       1 false       0 -        -
 daily                            7 false       0 -        -
 weekly                          52 false       0 -        -
 monthly                         84 false       0 monthly  -
```

- **XDPDefault**
  This is an asynchronous vault policy for mirroring all snapshots with labels of Daily and Weekly. This is a legacy vault-only policy (Figure 21).

  This policy consists of the following settings:
  - The policy type is set to `vault`.
  - The Create Snapshot value is set to `false`, which means the policy does not create a snapshot when an update is triggered.
  - There are two rules:
    - `daily` keeps seven daily snapshots.
    - `weekly` keeps 52 weekly snapshots.

Figure 21 XDPDefault policy definition

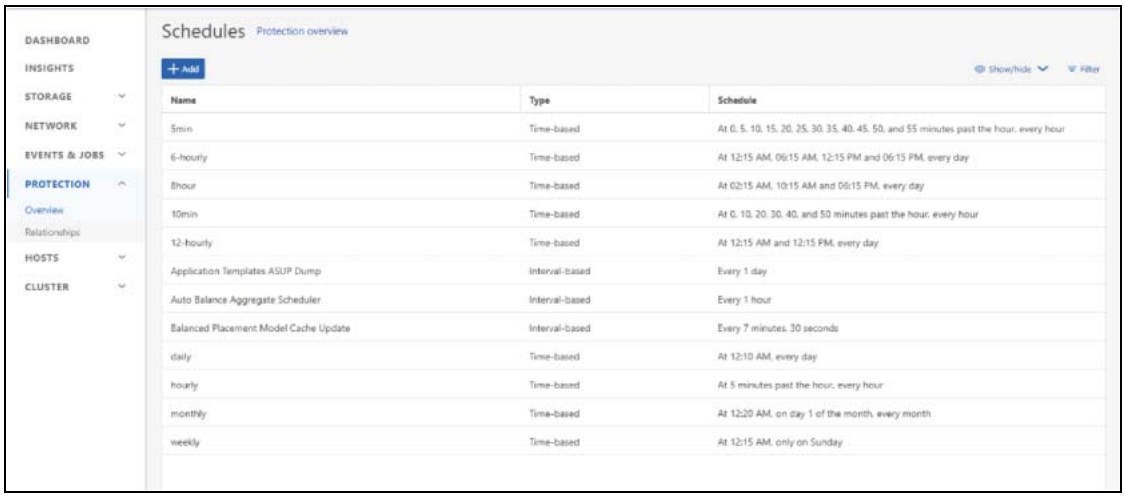
```
cluster_dst::> snapmirror policy show -policy XDPDefault -instance

                     Vserver: vs0
       SnapMirror Policy Name: XDPDefault
       SnapMirror Policy Type: vault
                 Policy Owner: cluster-admin
                  Tries Limit: 8
            Transfer Priority: normal
   Ignore accesstime Enabled: false
     Transfer Restartability: always
  Network Compression Enabled: false
              Create Snapshot: false
                      Comment: Vault policy with daily and weekly rules.
       Total Number of Rules: 2
                  Total Keep: 59
       Transfer Schedule Name: -
                    Throttle: unlimited
 Rules:
 SnapMirror Label             Keep Preserve Warn Schedule Prefix
 ---------------------------- ---- -------- ---- -------- ----------
 daily                           7 false       0 -        -
 weekly                         52 false       0 -        -
```

# SnapMirror Schedules

A SnapMirror policy requires at least one SnapMirror job schedule be assigned to the SnapMirror relationship. The schedule can be a pre-defined schedule, or a custom schedule can be created to run periodic asynchronous replication by assigning a schedule to a SnapMirror relationship in the destination cluster. Figure 22 shows the pre-defined job schedules in ONTAP System Manager.

Figure 22 SnapMirror schedules can be listed and created in System Manager




Alternatively, use the `job schedule cron create` command to create a schedule through the command line or create a new schedule by clicking the +Add button in System Manager. The following example demonstrates the creation of a schedule called `Hourly_SnapMirror` that runs at the top of every hour (on the zero minute of every hour) as shown in Figure 23.

Figure 23 SnapMirror schedules can be listed and created using CLI

```
cluster02::> job schedule cron create Hourly_SnapMirror -minute 0
cluster02::> job schedule cron show
Name                Description
----------------    ----------------------------------------------------
5min                @:00,:05,:10,:15,:20,:25,:30,:35,:40,:45,:50,:55
8hour               @2:15,10:15,18:15
Hourly_SnapMirror   @:00
avUpdateSchedule    @2:00
daily               @0:10
hourly              @:05
weekly              Sun@0:15
```

The schedule can then be applied to a SnapMirror relationship at the time of creation using the `-schedule` option or added to an existing relationship using the `snapmirror modify` command and the `-schedule` option. In this example, the `Hourly_SnapMirror` schedule is applied to an existing relationship.

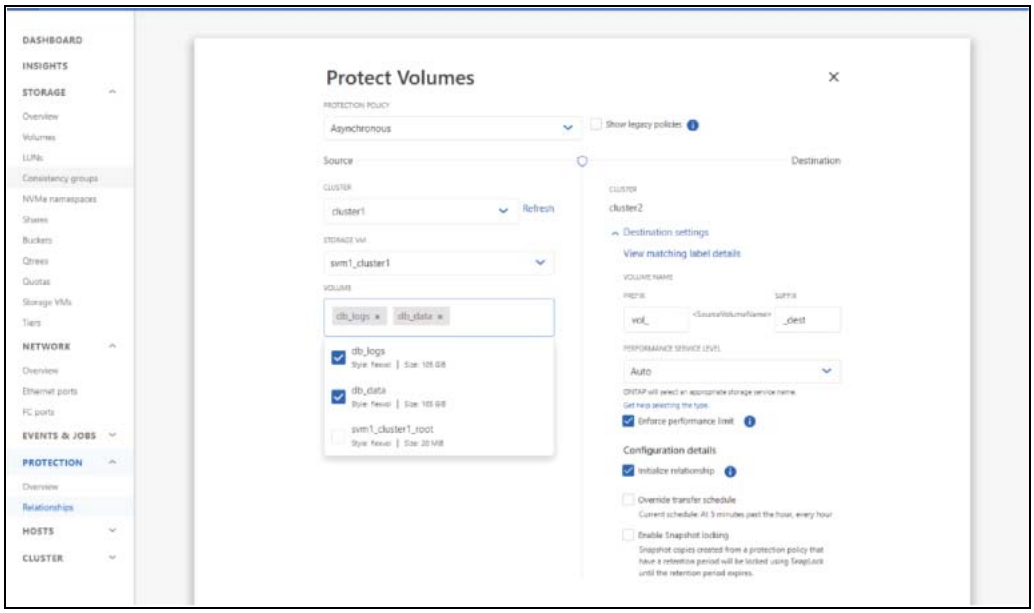
```
cluster02::> snapmirror modify -destination-path cluster02://vs1/vol1 -schedule Hourly_SnapMirror
```

# Create a SnapMirror Relationship

SnapMirror relationships are created between the source volume on one cluster and a destination volume on the peered cluster enabling data protection. From ONTAP System Manager on the source cluster, complete the following steps:

**Procedure ▶▶▶**

1. Click Relationships.
2. Select the volume or consistency group on which to create a SnapMirror relationship, and then click Save.




3. After the relationship is created and initialized, verify that the relationship status of the SnapMirror relationship is in the Mirrored state.


4. Select the SnapMirror relationship between the source and the destination volumes, and then verify the status in the Details tab.

5 The Details tab displays the health status of the SnapMirror relationship and shows the transfer errors and lag time.

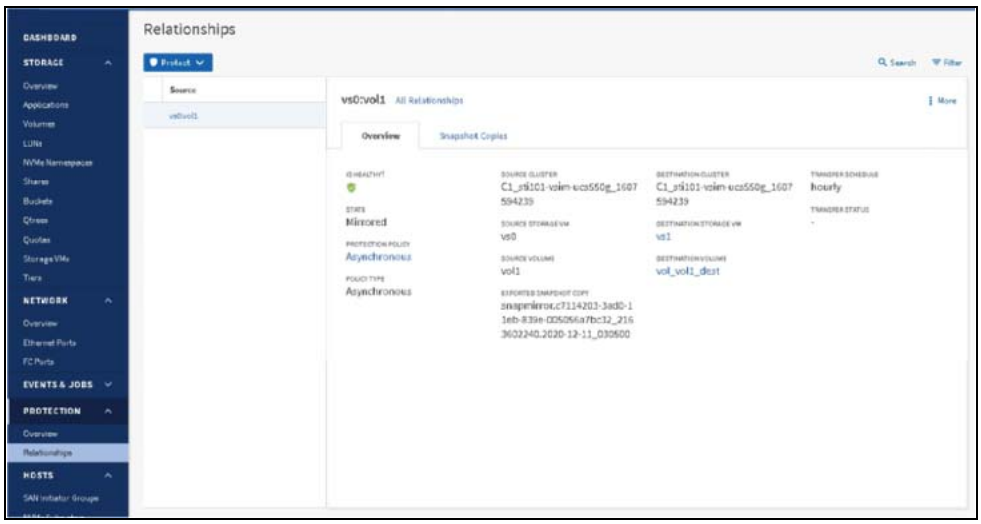
6 The Is Healthy field must display Yes.

7 For most SnapMirror data transfer failures, the field displays No. In some failure cases, however, the field continues to display Yes. This might require checking the transfer errors in the Details section to verify that no data transfer failure occurred.

8 The State field must display Mirrored.

9 The lag time must be no more than the transfer schedule interval. For example, if the transfer schedule is hourly, then the lag time must not be more than an hour.

10 Also, navigate to the Volumes window, and then select the desired volume. Double-click the volume to view the volume details and the data protection status.




# Baseline Transfer During Initialization of SnapMirror Relationship

When a new SnapMirror relationship is created, it establishes the relationship and the metadata that defines it. Optionally, select Initialize the Relationship to perform a baseline transfer from the source to the destination based on the SnapMirror policy that defines the content of the baseline and any updates.
SnapMirror uses the following process to initialized a relationship:

**Procedure**

1 Make a snapshot of the source.

2 Transfer the snapshot to the destination.

3 Depending on the SnapMirror policy attached to the relationship, it also transfers other snapshots from the source to the destination.

4 After baseline transfer, updates to this relationship occur according to the schedule assigned to the SnapMirror relationship.

The destination is a volume or consistency group that is already created and marked restricted. After SnapMirror finishes transferring the data, it brings the destination volumes online in a read-only state. While the initial data transfer is taking place, the destination is marked invalid in the output of a `vol status` command. The volumes become valid and go online after the initial transfer is complete. Now, the files and snapshots in the source volumes should be available on the destination.

# Manual Update to the SnapMirror Relationship

The SnapMirror relationship might need a manual update either from the newest snapshot or from a specific Snapshot to prevent data loss due to an upcoming power outage, scheduled maintenance, or data migration (Figure 24 and Figure 25).

Figure 24 Start a SnapMirror relationship update

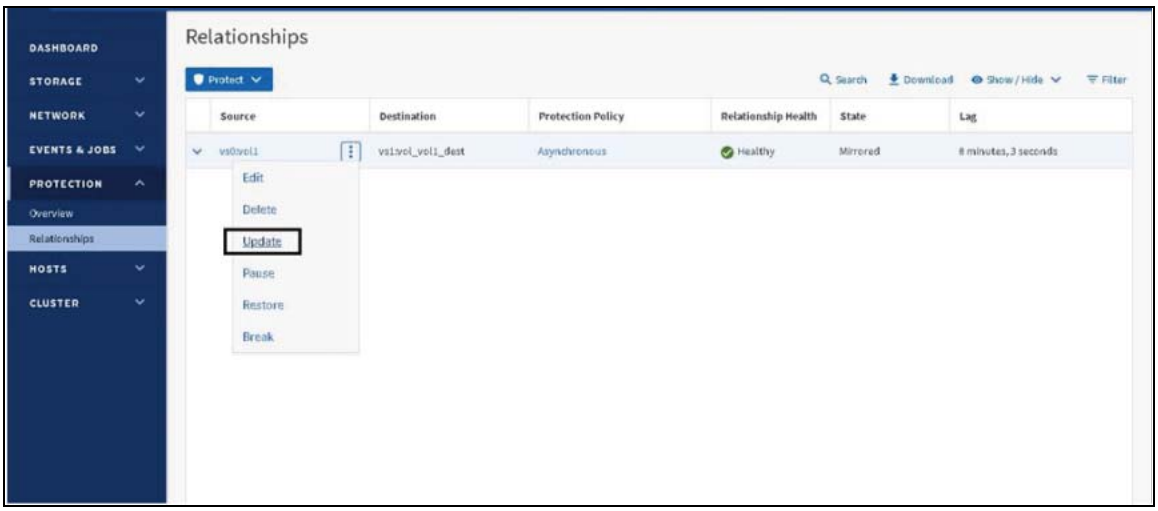
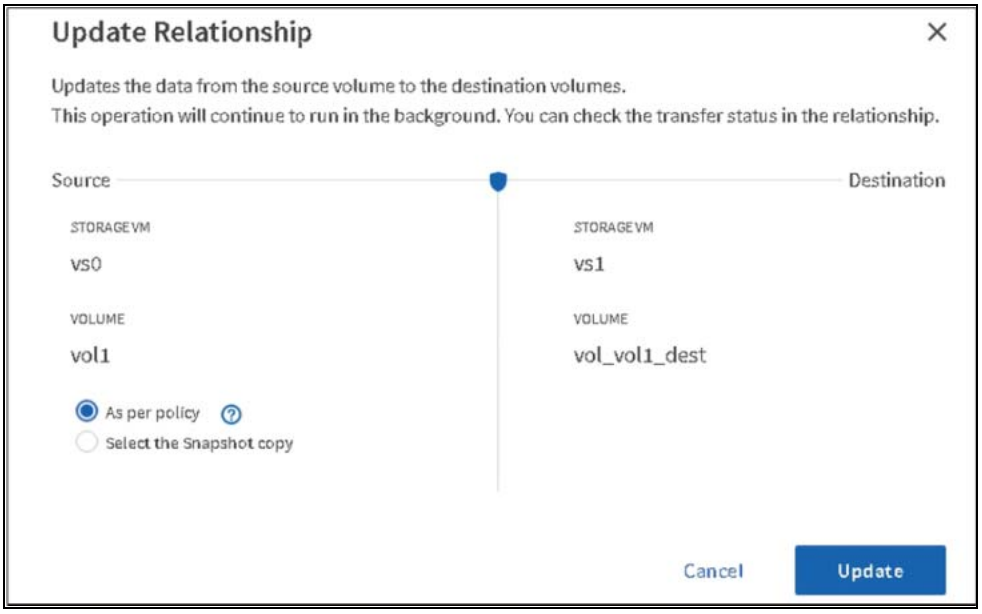


Figure 25 Relationship update dialog




After the update completes, the Transfer State field changes from Transferring to Idle.

# 5. Converting a Legacy DP SnapMirror Relationship to an XDP SnapMirror Relationship

As of ONTAP 9.11.1, DP type SnapMirror relationships are supported only for pre-existing SnapMirror relationships within the cluster, and as of ONTAP 9.12.1, DP type SnapMirror relationships are no longer supported, so existing DP relationships must be converted to the new XDP relationships. A summary of the process includes:

**Procedure ▶▶▶**

1. Break the SnapMirror volume relationship from the destination cluster.
2. Delete the SnapMirror volume relationship.
3. Create a new SnapMirror relationship using one of the built-in policies (all built-in policies use XDP relationship type), or create a custom policy with the `-type XDP` parameter, between the same endpoints with one of the default SnapMirror unified replication policies.
4. Perform a resync operation between the endpoints. This resync converts the relationship to a SnapMirror unified replication configuration without having to perform a re-baseline.

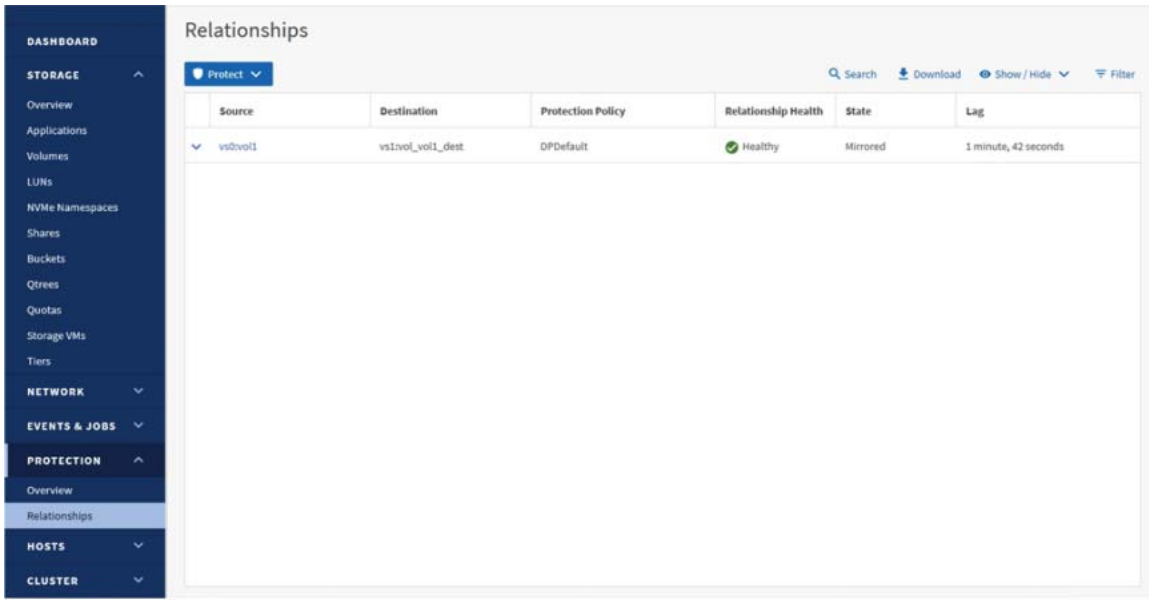
◀◀◀

## SnapMirror Relationship Conversion Details

The following steps present this process in detail:

**Procedure ▶▶▶**

1. View the status of the SnapMirror relationship. The SnapMirror relationship state is shown as Mirrored.

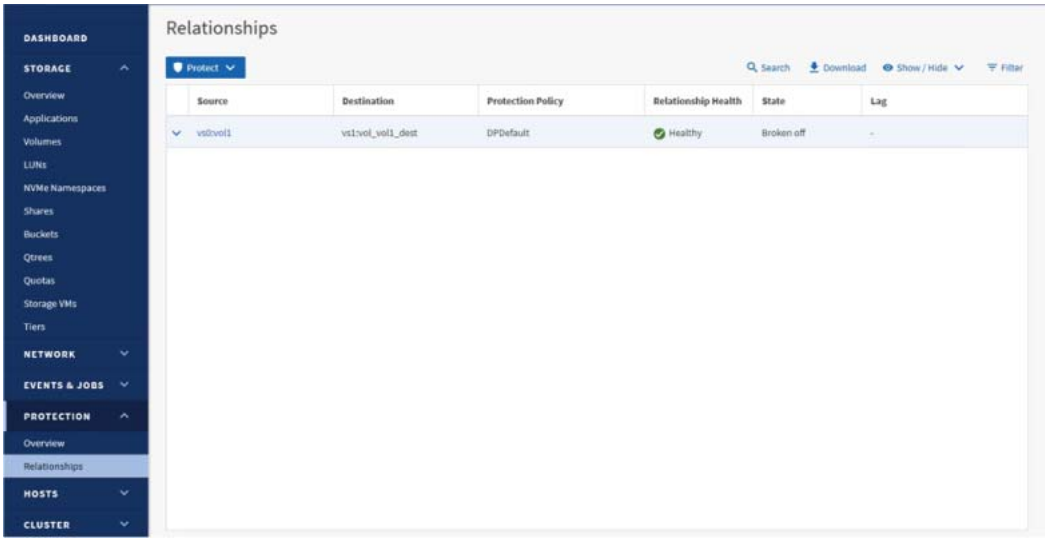


**2** Perform a SnapMirror Break operation:

a Using the CLI, run the following commands:

```
Remote::> snapmirror break -destination-path vs1:vol_vol1_dr
[Job 128] Job succeeded: SnapMirror Break Succeeded
```

b Using System Manager, click the desired relationship then click the More > Break command. Upon completion, System Manager shows the relationship to be Broken Off in the State column.

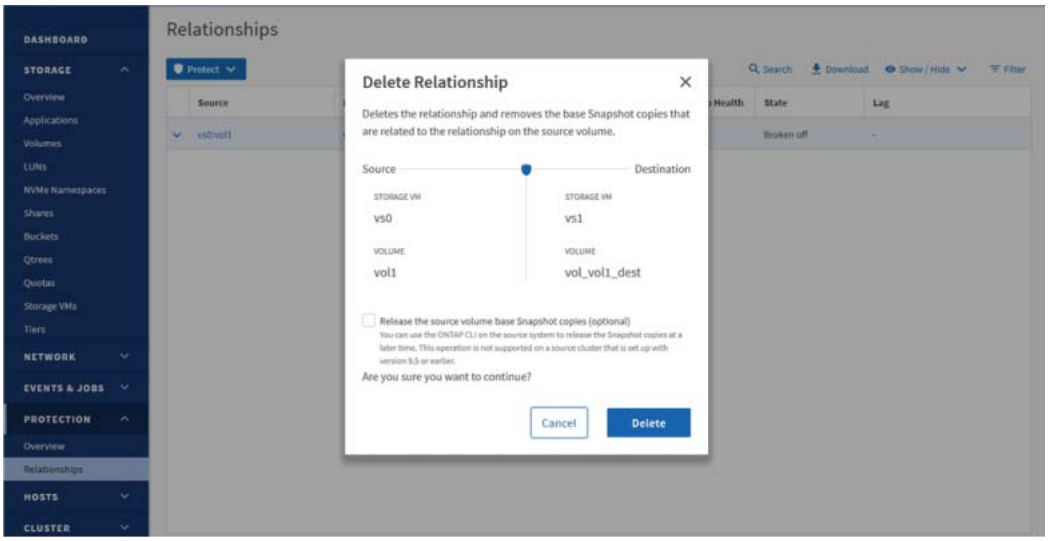



**3** Delete the SnapMirror relationship:

a Using the CLI: Run the `snapmirror delete` command.

```
Remote::> snapmirror delete -destination-path vs1:vol_vol1_dr -relationship-info-only true
Operation succeeded: snapmirror delete the relationship with destination vs1:vol_vol1_dr.
```

b Using System Manager, click the More > Delete command. Ensure that the Release the Source Volume Base Snapshot Copies field is unchecked before clicking Delete.



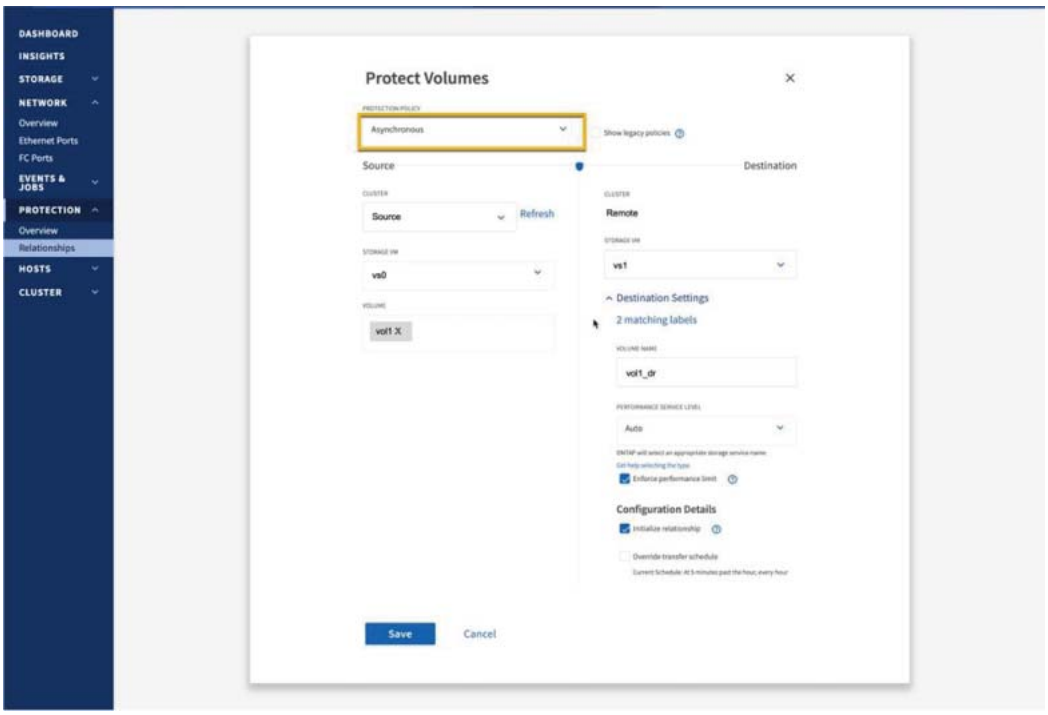

This relationship disappears from the destination cluster under Protection > Relationships.

**4** Create the unified replication relationship:

a Using the CLI, run the `snapmirror create` command with a `-policy Asynchronous` parameter.

```
Remote::> snapmirror create -source-path vs0:vol1 -destination-path vs1:vol_vol1_dr -type XDP
-policy Asynchronous
Operation succeeded: snapmirror create the relationship with destination svm_dst1:Source_dest.
```

b Using System Manager, click the Protect > Volumes option and select the Protection Policy Asynchronous from the drop-down menu.

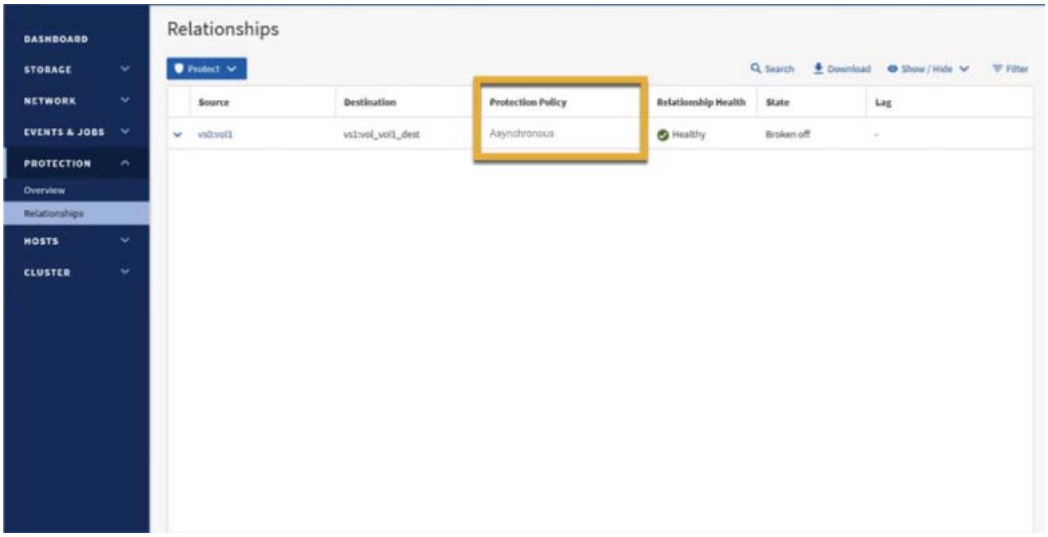



**5** Verify that the new relationship is in place:

a Using the CLI, run the `snapmirror show` command.

```
Remote::> snapmirror show
                                                                    Progress
Source            Destination  Mirror  Relationship  Total            Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
vs0:vol1
            XDP  vs1:vol_vol1_dr
                              Broken-off
                                      Idle           -         true    -
```

b Using System Manager, verify that the Protection Policy is listed as Asynchronous.

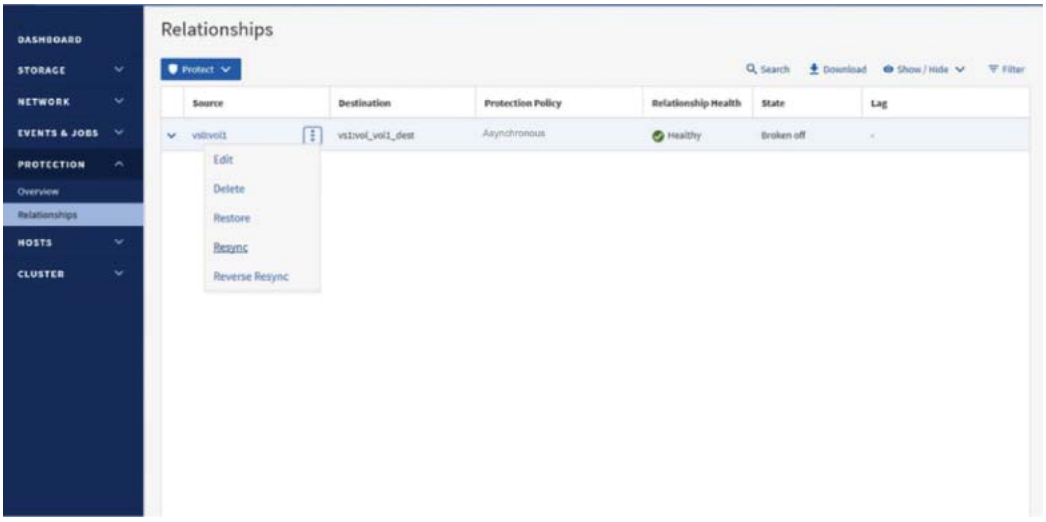



**6** Resync the SnapMirror relationship:

a Using the CLI, run the `snapmirror resync` command Remote::> snapmirror resync -destination-path `vs1:vol_vol1_dr`.

```
Warning: All data newer than Snapshot copy snapmirror.12ceb7f0-b078-11e8-baec-
005056b013db_2160175149.2020-01-24_091316 on volume
         vs1:vol_vol1_drwill be deleted.
Do you want to continue? {y|n}: y
Operation is queued: initiate snapmirror resync to destination "vs1:vol_vol1_dr".
```

b Using System Manager, select More > Resync.

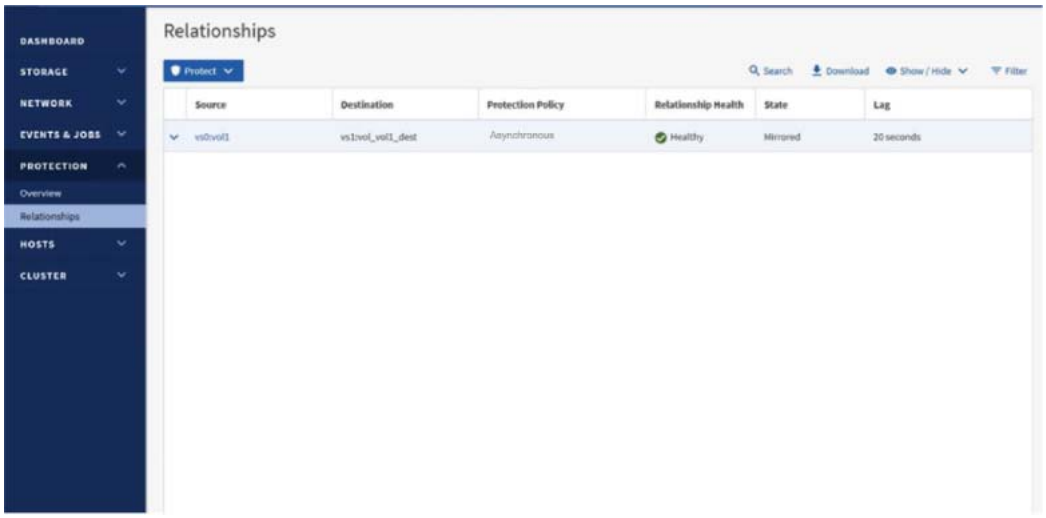



**7** Verify the relationship status:

a Using the CLI, run the `snapmirror show` command and verify the Relationship Status column shows Idle.

```
Remote::> snapmirror show
                                                                    Progress
Source            Destination  Mirror  Relationship  Total            Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
vs0:vol1
            XDP  vs1:vol_vol1_dr
                              Snapmirrored
                                      Idle           -         true    -
```

b Using System Manager, verify the State column shows Mirrored and the Relationship Health column shows Healthy.

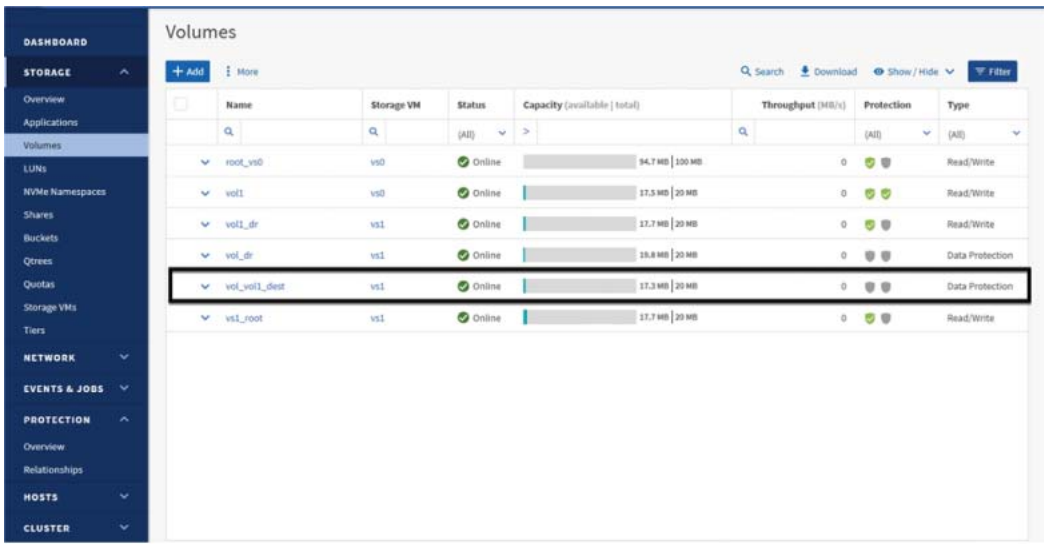



**8** Verify the volume changes from type `rw` to type `dp`:

a Using the CLI, run the `volume show` command and verify that the volume type is listed as `DP`.

```
Remote::> volume show -volume vol1
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
vs1       vol1_dr       data_01      online     DP        250GB    217.8GB   12%
```

b Using System Manager, verify the Type field shows Data Protection.



# 6. SnapMirror Asynchronous ONTAP Feature Interoperability

## SnapMirror and Snapshot Copies

SnapMirror creates a snapshot before it performs a replication update. A SnapMirror snapshot is created on the source volume and a snapshot label of `sm_created` is applied. The new snapshot is then compared to the previous SnapMirror snapshot that was replicated to the data protection volume. Any data changes between the new SnapMirror snapshot and the previous one (including all snapshots on the volume between the two SnapMirror snapshots and all data in those snapshots) is replicated to the destination volume. After the SnapMirror update is complete, the new SnapMirror snapshot is exported on the destination system. SnapMirror maintains a history of one SnapMirror snapshot on the source volume and two on the destination volume.

- Best Practice

  Verify that SnapMirror updates are not scheduled to occur on the source volume at the same time as other snapshots.

ONTAP maintains locks on snapshots created by SnapMirror to prevent them from being deleted by mistake because these snapshots are required to perform scheduled updates. If the snapshots created by SnapMirror must be deleted, the volumes can still be resynchronized. A full baseline is not required if other common snapshots between the two volumes still exist on the volumes.

In the following example, a SnapMirror resync is performed on a volume where all snapshots created by SnapMirror were deleted and uses the hourly snapshot as the base for the resync.

```
remote::> snapmirror resync -source-path cluster01://vs1/vol1 -destination-path remote://vs2/vol1
Warning: All data newer than Snapshot copy hourly.2011-12-06_1805 on volume remote://vs2/vol1
will be deleted.
Do you want to continue? {y|n}: y
[Job 1364] Job is queued: snapmirror resync to destination cluster02://vs2/vol1.
```

## SnapMirror and Locked Snapshot Copies

Introduced in ONTAP 9.12.1, locked Snapshot copies can provide indelible protection for individual snapshots without the need for a dedicated SnapLock volume. SnapMirror replicates snapshots that are marked as tamper-proof along with other snapshots based on the SnapMirror replication policy rules. The replicated locked snapshots will be replicated with either:

- The same retention time of the source tamper-proof snapshot,

or

- The default retention time of the destination cluster,

whichever is longer.

**Note**

Though tamper-proof snapshots do not require the configuration of a SnapLock volume, it does require a SnapLock license on each cluster that implements this feature. Contact your sales representative for details.

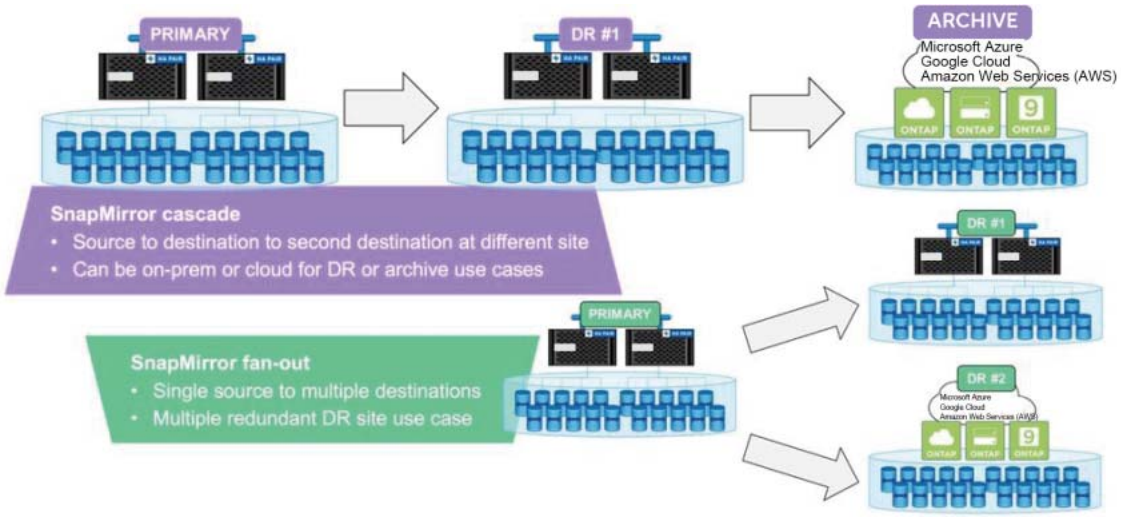
# SnapMirror and Qtrees

Qtrees are special directories that allow the application of file system quotas for NAS. ONTAP allows creation of qtrees, and qtrees can exist in volumes that are replicated with SnapMirror. However, SnapMirror does not allow replication of individual qtrees or qtree-level replication because SnapMirror replications operate only at the volume level.

# SnapMirror and FlexGroup Volumes

Starting with ONTAP 9.9.1, SnapMirror supports FlexGroup volumes as source and destination in cascade and fan-out configurations (Figure 26). Destinations can be on-premises or cloud-hosted Cloud Volumes ONTAP clusters.

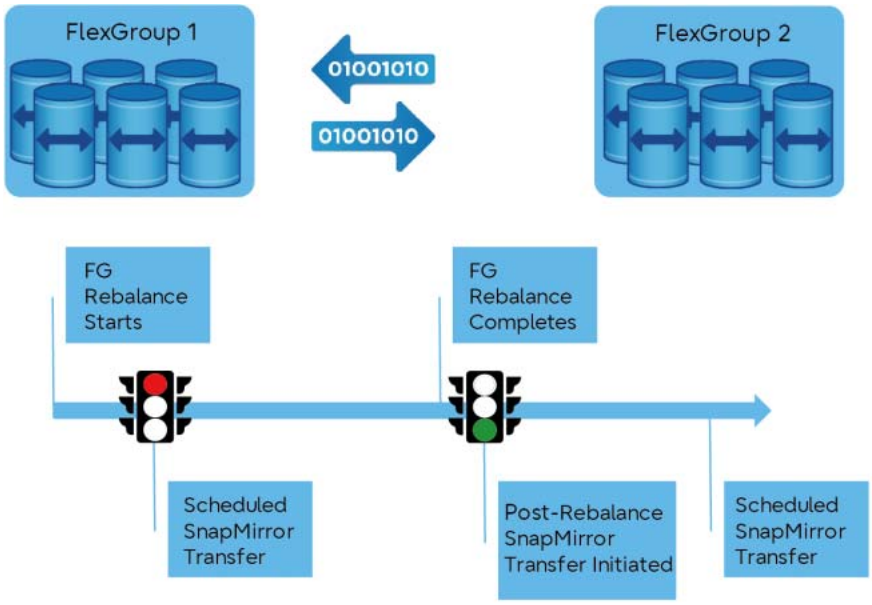
Figure 26 FlexGroup volumes used in SnapMirror cascade and fan-out configurations



## SnapMirror Behavior During FlexGroup Rebalance Operations

Beginning with ONTAP 9.12.1, SnapMirror behavior is changed to execute the last scheduled SnapMirror operation immediately following the completion of the FlexGroup volume rebalance operation. This ensures that the RPO is minimized due to the rebalance operation suspending the scheduled SnapMirror operation and restarting the scheduled SnapMirror operation after the rebalance operation is completed. This updated process is illustrated in Figure 27.

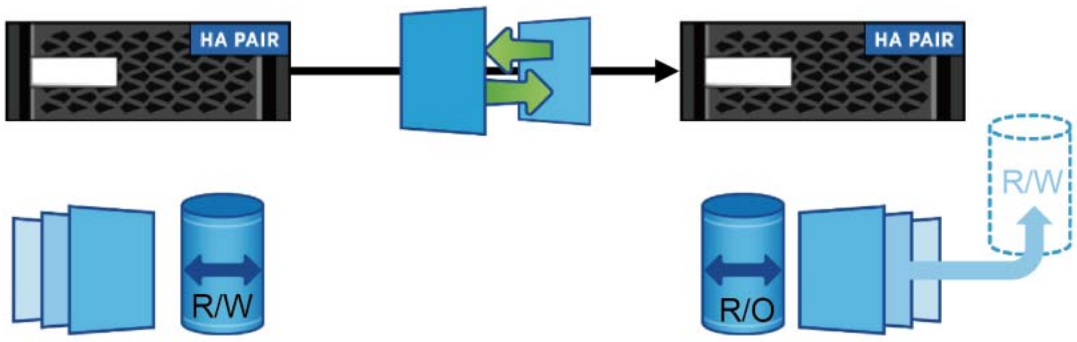
Figure 27 SnapMirror behavior during FlexGroup rebalance operation




# SnapMirror and FlexClone Technologies

FlexClone technology allows storage administrators to create a writable volume from a read-only SnapMirror destination volume without interrupting the SnapMirror replication process. Although a SnapMirror relationship can be created using a FlexClone volume as the source, the SnapMirror destination volume cannot be a FlexClone volume. Figure 28 illustrates the creation of a FlexClone volume at the SnapMirror destination.

Figure 28 Creating a FlexClone volume at a SnapMirror destination



SnapMirror replicates the snapshot history from a source to a destination volume. If a snapshot is removed from the source volume, the next SnapMirror update removes that snapshot from the destination volume. If that snapshot is the basis for a FlexClone volume, then the SnapMirror update will fail. The only way for a SnapMirror update to proceed is to delete the FlexClone volume or split it to remove the snapshot dependency.

To avoid this issue when creating FlexClone volumes on SnapMirror destination, manually create a base snapshot required by the FlexClone volume on the source system, then replicate that snapshot to the destination system and use that snapshot as the base for the FlexClone volume, as shown in Figure 28. Using a snapshot specifically created for the FlexClone volume in this manner prevents the SnapMirror update from failing due to an automatically created snapshot being removed from the source system. The label associated with this snapshot should also not use any labels that are associated with any SnapMirror replication policy rules in effect. This ensures that the retention policies for snapshots using the preexisting labels do not attempt to delete the snapshot associated with the FlexClone volume.

# SnapMirror and Storage Efficiency

In general, SnapMirror will always replicate data to the destination in the same state as the source data with respect to storage efficiency (SE) regardless of source and destination default SE settings. SnapMirror maintains all storage efficiency benefits in replicated volumes including deduplication, compression, and compaction. This ensures that ONTAP maximizes resource utilization for data at rest as well as over the network.

SnapMirror creates a snapshot before performing an update transfer. Any blocks in the snapshot are locked and cannot be deduplicated. Therefore, if maximum space savings from deduplication are required, run the dedupe process before performing SnapMirror updates.

- Best Practice

  Make sure that deduplication and SnapMirror operations do not run at the same time. Start SnapMirror transfers of a deduplicated volume after the deduplication operation is complete. This prevents any effects on replication performance while deduplication is in progress and sending of non-deduplicated data and additional temporary deduplication metadata files over the network.

One aspect of ONTAP SE that can change depending on the source and destination array models is compression. Several modern array controller models support enhanced compression techniques that may not be available on the older array controller model. Once a DP volume is converted to read/write state using the `snapmirror break` command, SE for new updates and writes may change to the default for the destination array model depending on the ONTAP version for all type XDP relationships. All-flash array controllers can have temperature-sensitive SE (TSSE) enabled (ETERNUS AX/AC series has TSSE enabled by default).

## Storage Compression for XDP SnapMirror Relationships

Table 9 provides guidance on SE behavior for SnapMirror protected volumes when the source does not have TSSE enabled, but the destination volume is on a controller that has TSSE enabled as a default (for example, ETERNUS AX series with TSSE enabled or an ETERNUS AC series controller).

Table 9 Replication behavior when source volume is on a non-TSSE capable controller and destination volume is on a TSSE enabled controller

| Destination ONTAP versions | Source volume compression | Replicated data (Pre-Break) | Post-break | | |
|---|---|---|---|---|---|
| | | | Replicated data | Hot data (new data only) | Cold data (new and replicated data) |
| <-ONTAP 9.11.1P10<br><-ONTAP 9.12.1P3 | No compression | No compression | No compression | Standard compression | No TSSE-enabled cold data compression |
| | Compression enabled | Source compression | Source compression | | |
| ONTAP 9.11.1P11-><br>ONTAP 9.12.1P4-><br>ONTAP 9.13.1-> | No compression | No compression | No compression | TSSE compression | |
| | Compression enabled | Source compression | Source compression | | |

Table 10 provides guidance on SE behavior for SnapMirror protected volumes hosted on an ETERNUS AC series or ETERNUS AX series with TSSE enabled on the protected volume replicating to a non-TSSE capable array model.

Table 10 Compression behavior when source volume has TSSE enabled, and destination volume controller is not on a TSSE capable

| Destination ONTAP versions | Source volume compression | Replicated data (Pre-Break) | Post-Break | | |
|---|---|---|---|---|---|
| | | | Replicated data | Hot data (new data only) | Cold data (new and replicated data) |
| All | TSSE | TSSE | TSSE | **ETERNUS HX series**: Uncompressed<br><br>**ETERNUS AX/AC series (TSSE not enabled)**: File compression | NA |

Table 11 provides guidance on SE behavior for SnapMirror protected volumes hosted on an ETERNUS AC series or ETERNUS AX series with TSSE enabled on the protected volume when replicating to another TSSE capable array controller (ETERNUS AC series or ETERNUS AX series).

Table 11 Compression behavior when source volume has TSSE enabled, and destination volume controller is TSSE capable

| Destination ONTAP versions | Source volume compression | Replicated data (Pre-Break) | Post-Break | | |
|---|---|---|---|---|---|
| | | | Replicated data | Hot data (new data only) | Cold data (new and replicated data) |
| All | TSSE | TSSE compression | TSSE compression | TSSE compression | |

## Storage Compression for DP SnapMirror Relationships

If SnapMirror is using legacy (`-type dp`) relationships, storage efficiency configurations cannot differ between the source and destination volumes. For example, it is not possible to compress or deduplicate the SnapMirror destination volume alone without enabling compression or deduplication on the SnapMirror source volume.

# SnapMirror and Volume Move

The volume-move capability allows volumes to be moved nondisruptively between nodes in the cluster using the volume move command. The SnapMirror relationship does not have to be reconfigured or modified on the source or destination when a volume move is performed. If a volume in an intercluster SnapMirror relationship is moved, the node to which the volume is moved must have an intercluster LIF and be connected to the intercluster network to successfully perform SnapMirror updates.

The effect a volume move has on a SnapMirror relationship depends on whether the source volume or the destination volume is being moved. If a SnapMirror transfer is currently in progress and the SnapMirror source volume is being moved, then both the SnapMirror transfer, and the volume move transfer can run simultaneously. However, when the volume move cutover occurs (the moment ONTAP redirects I/O to the new volume), the active SnapMirror transfer is then momentarily interrupted and automatically continues from the source volume's new location.

For more information about volume move, see the documentation in the manual site.

# SnapMirror for Drive Shelf Failure Protection

SnapMirror can mirror volumes to nodes in a different HA pair on the same cluster. Mirroring a volume to a different HA pair ensures that the other volume is always placed on a different drive shelf. If mirroring to a different drive shelf on the same node, then the mirror must be on a different aggregate. There is still a risk that an aggregate might have a constituent drive from any drive shelf due to drive failure and having a spare assigned. This configuration avoids having a single point of failure and provides protection against drive shelf failure.

One caveat is that the configuration does not failover automatically. The storage administrator must manually break the SnapMirror relationship, unmount the clients, remount the clients on the destination volumes, and change the NFS export policies.

# SnapMirror and Volume Autosize

When using SnapMirror XDP relationships, it is possible to mirror a larger volume from a source to a smaller volume on the destination due to the integrated data efficiencies native to the LRSE replication process used by XDP relationships. It is recommended that destination volumes be similar in size (or larger) than the source volume and enable the Autosize option.

## Best Practice

- Keep the source and destination volumes the same size or slightly larger with the Auto Grow option enabled on the destination volume.
- If Autosize is enabled on the source, Autosize is recommended to be enabled on the destination to ensure adequate capacity for the SnapMirror transfers.

When autosize increases the size of the source volume of a SnapMirror relationship, the destination volume also automatically increases in size.

# SnapMirror and NDMP

NDMP backups can be performed from either a SnapMirror source or destination volume. When a SnapMirror destination is backed up to tape with the dump engine, only the data in the volume is backed up. However, if a SnapMirror destination is backed up to tape using SMTape, then the metadata is also backed up. The SnapMirror relationships and the associated metadata are not backed up to tape. Therefore, during restore, only the data on that volume is restored, but the associated SnapMirror relationships are not restored. There are advantages to performing NDMP backups from SnapMirror destination volumes rather than from source volumes, including the following:

- SnapMirror transfers can happen quickly and with less effect on the source system. Use snapshots and perform SnapMirror replication from a primary system as a first stage of backup to significantly shorten or eliminate backup windows. Then perform NDMP backup to tape from the secondary system.
- SnapMirror source volumes are more likely to be moved to optimize the production environment than the DR volumes on the destination.

# SnapMirror and FabricPool

SnapMirror supports replication of volumes hosted on FabricPool-enabled aggregates. When replicating volumes on a FabricPool aggregate, the replication interval should be set to a lower value than the FabricPool tiering policy to make sure that all data is protected.

**Note**

- FabricPool alone does not represent a data protection strategy.
- Starting with ONTAP 9.12.1, SnapMirror SVM DR supports replication of SVMs hosting both FlexGroup volumes and FabricPool volumes.

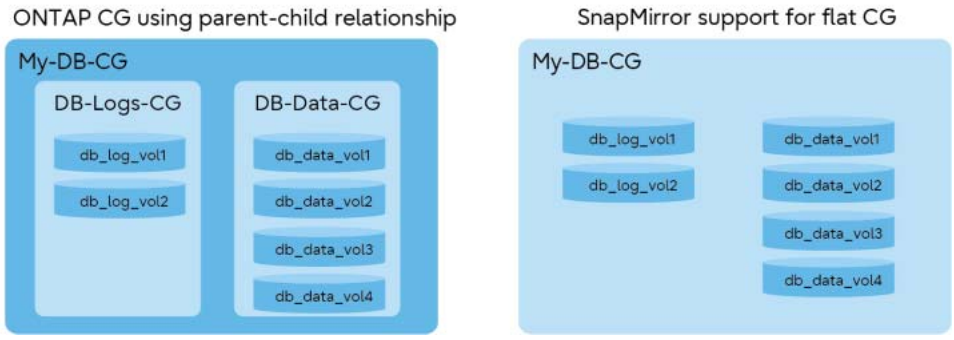
# SnapMirror and Consistency Groups (CGs)

ONTAP supports the use of CGs to group multiple volumes such that operations can be applied to the group of volumes in a coherent way. This ensures that all volumes in a CG are write consistent with each other. CGs are typically used for backup of more advanced applications – such as relational databases – that utilize multiple volumes to store data for varying use cases (such as logs and data table storage) but must be in lockstep. The ONTAP implementation of CGs provides the ability to create CGs that contain volumes in a parent-child CG configuration. This feature is used primarily for snapshot creation and SnapMirror Business Continuity configurations.

Starting with ONTAP 9.13.1, SnapMirror provides support for replicating volumes in consistency groups. For this release, only single-tier – what is referred to as flat – consistency groups are supported for SnapMirror asynchronous replication. Creating SnapMirror relationships that target CGs that contain other CGs (parent-child CGs) will result in an error. Other than this parent-child CG restriction, SnapMirror relationships targeting CG sources will support the same functionality currently offered for SnapMirror relationships created against individual volume sources unless otherwise noted in this section.

Figure 29 compares an example of a parent-child CG to a flat CG supported by SnapMirror for asynchronous replication.

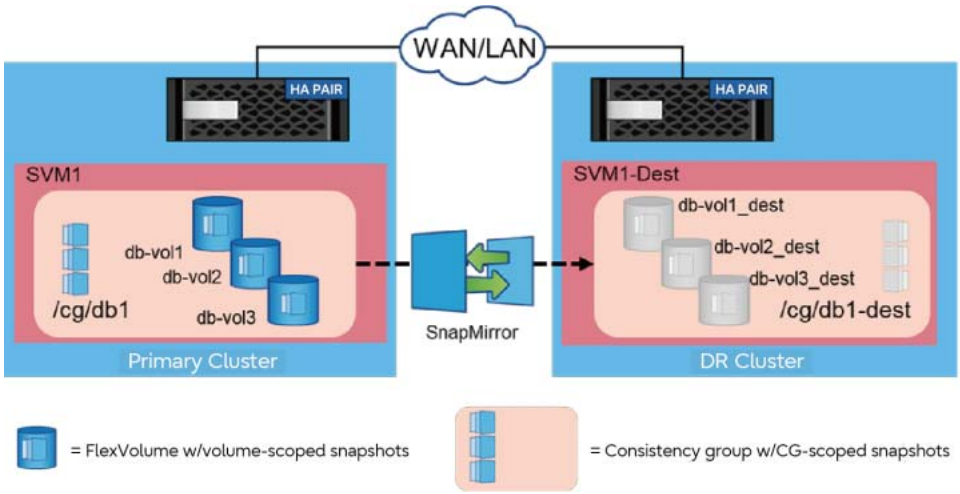
Figure 29 Comparing ONTAP CGs with SnapMirror CG support



## Overview of SnapMirror with CGs

The establishment of SnapMirror relationships that target CGs is like that for creating SnapMirror relationships for individual volumes. The difference is associated with referencing source and destination CGs (instead of individual volumes) and providing a volume map between source CG component volumes and destination CG component volumes (Figure 30).

Figure 30 SnapMirror for Consistency Groups




SnapMirror relationships targeting CGs support all SnapMirror policy types. What snapshots will be replicated will depend on the ONTAP version and the SnapMirror replication policy. In all ONTAP versions, SnapMirror supports all SnapMirror replication policies (async-mirror, vault, and mirror-vault). For vault, mirror-vault relationships and async-mirror relationships using MirrorAllSnapshots, what snapshots get replicated will vary depending on ONTAP version.

For ONTAP 9.13.1, SnapMirror will replicate only those snapshots associated with the CG that meet the policy type and any SnapMirror snapshot label rules. It will not replicate any member volume-scoped snapshots that may meet the SnapMirror CG policy definitions.

Starting with ONTAP 9.14.1, SnapMirror will replicate all snapshots associated with the CG and the member volumes that have snapshots matching snapshot label rules that are part of the SnapMirror CG policy definition.

## Creating a SnapMirror CG Relationship

To create a SnapMirror relationship between two CGs, the source and destination CGs must be identified along with a mapping of the volumes in the source CG to the volumes in the destination CG. Prior to creating the SnapMirror relationship, make sure all volumes and CGs are created on both the source and destination clusters.

**Note**

CGs cannot be created, managed, or deleted using ONTAP CLI. Acceptable interfaces for CG management are ONTAP REST API and ONTAP System Manager.

### Creating SnapMirror Relationships for CGs using ONTAP CLI

Using Figure 30 as a reference, the following CLI command executed from the destination cluster will create an hourly SnapMirror replication of the source CG to the destination CG.

```
Dest::> snapmirror create -source-path SVM1:/cg/db1 -destination-path SVM1-Dest:/cg/db1-dest -cg-item-mappings db-vol1:@db-vol1_dest, db-vol2:@db-vol2_dest, db-vol3:@db-vol3_dest -policy MirroAndVault -schedule hourly
```

### Creating SnapMirror Relationships for CGs using ONTAP REST APIs

ONTAP REST APIs can be used to create a SnapMirror relationship from either the source cluster or the destination cluster. REST API commands can be executed from the source or destination cluster.

If the destination CG does not exist, the REST API operation will automatically create the CG. As part of provisioning the destination ONTAP CG, destination constituent volumes matching the source CG constituent volumes are created on the destination cluster and then the destination CG is created using these new constituent volumes. The operation also establishes the SVM peering relationship, if it does not already exist, then creates the CG Async SnapMirror relationship and then optionally initializes the relationship. This operation can be initiated either on the source or on the destination cluster.

Using Figure 30 as a reference, the following REST command executed from the source cluster will create a SnapMirror replication of the source CG to the destination CG.

```
POST /api/snapmirror/relationships/
'{"source":      { "path": "SVM1:/cg/db1", "consistency_group_volumes": "db-vol1, db-vol2, db-vol3"},
  "destination": { "cluster.name": "dest_cluster", "path": "SVM1-Dest:/cg/db1-dest", "consistency_group_volumes": "db-vol1_dest, db-vol2_dest, db-vol3_dest"},
  "policy":      "MirrorAllSnapshots",
"create_destination": { "enabled": "true", "storage_service": { "enabled": "true", "name": "extreme", "enforce_performance": "true" } }'
```

### Creating SnapMirror Relationships for CGs using ONTAP System Manager

ONTAP System Manager can be used to create SnapMirror relationships for CG. The process for creating a SnapMirror relationship for a CG using System Manager will be the similar to the process for creating a volume scoped SnapMirror relationship except you start creating the relationship on consistency group object rather than a volume object. When System Manager detects the source of the relationship is a CG, the user will be offered a full set of pre-created or custom asynchronous SnapMirror protection policies alongside any SnapMirror Business Continuity (SM-BC) protection policies. SnapMirror will create the appropriate relationship type based on the selected protection policy.

Once the clusters and SVMs are peered, the user can protect a CG through the System Manager Consistency Group page by selecting Protect consistency group menu option from the destination cluster.

## Scalability of SnapMirror Relationships for CGs

SnapMirror can scale to support up to 50 CGs per cluster with a maximum of 16 FlexVol member volumes per CG. The maximum number of total member volumes supported per cluster is 400 Flexvol volumes. Also, the total number of CGs and FlexGroup volumes cannot exceed 50 for any cluster.

# Managing SnapMirror Protection of CGs

CGs add an additional layer of complexity to the data protection process since it must coordinate the replication of multiple member volumes with a single SnapMirror relationship. Each of the following sections describes how SnapMirror behaves for creation and recovery scenarios.

## SnapMirror Replication

When a SnapMirror scheduled replication executes, the following actions take place:

**Procedure ▶▶▶**

1. All constituent volumes within the CG will be fenced from all I/O operations.
2. A CG snapshot of each member volume is created.
3. The fence is then removed to permit continued I/O operations.
4. SnapMirror will replicate the changes captured in the member of volumes' CG snapshots based on the SnapMirror policy as shown in Table 12.

◀◀◀

Table 12 lists the which type of snapshots are replicated based on the SnapMirror policy when replicating CGs.

Table 12 What snapshots get replicated as part of a SnapMirror replication of a consistency group

| SnapMirror policy type | Replicated snapshots |
|---|---|
| Async-mirror (only `sm_created` rule) | • The scheduled `sm_created` CG snapshot on each member volume |
| Async-mirror (`sm_created` + `all_source_snapshots` rules) | • The scheduled `sm_created` CG snapshot on each member volume<br>• All CG-scoped snapshots since the previous CG-scoped SnapMirror scheduled replication.<br>• All volume-scoped snapshots since previous CG-scoped SnapMirror scheduled replication. |
| Mirror-vault | • The scheduled `sm_created` CG snapshot on each member volume<br>• The CG-scoped snapshots since the previous CG-scoped SnapMirror scheduled replication that have `snapmirror-label` values matching the SnapMirror policy rules.<br>• The volume-scoped snapshots since the previous CG-scoped SnapMirror scheduled replication that have `snapmirror-label` values matching the SnapMirror policy rules. |
| Vault | • The CG-scoped snapshots since the previous CG-scoped SnapMirror scheduled replication that have `snapmirror-label` values matching the SnapMirror policy rules.<br>• The volume-scoped snapshots since the previous CG-scoped SnapMirror scheduled replication that have `snapmirror-label` values matching the SnapMirror policy rules. |

If the snapshot schedules are very short when compared to SnapMirror replication schedules, it may be possible that not all snapshtos will be available on the secondary due to `-keep` parameter settings. The SnapMirror policy's `-keep` setting for any rule is applied to the CG-scoped snapshots and the volume-scoped snapshots independently.

**Note**

Prior to ONTAP 9.14.1, snapshots independently created on each member volume, using a member volume's snapshot policy, are not replicated as part of the SnapMirror CG relationship. Only snapshots created as part of the SnapMirror CG relationship policy are replicated.

### SnapMirror Restore

The following behavior is true for restore operations to the original source CG or for a restore to a new CG (redirection).

- During a SnapMirror restore operation, all CG member volumes must be restored using the same CG snapshot.
- If this is not possible, the member volumes that cannot be restored will remain in a fenced state until all volumes are restored from the same CG snapshot.
- This fenced state for each failed member volume restore will remain until all member volumes get restored from the requested CG snapshot.

## Converting Existing Volume Relationships to CG Relationships

Consider a scenario where a set of individual volumes each has a SnapMirror asynchronous relationship. You now wish to combine these volumes into a CG and then protect the entire CG using a SnapMirror asynchronous relationship.

Converting these individual relationships into a single CG relationship is possible without requiring a new baseline replication. The following high-level steps can be used to convert the SnapMirror configuration:

**Procedure ▶▶▶**

1. Delete each volume SnapMirror relationship.
2. Release each volume SnapMirror relationship with the `-relationship-info-only true` parameter.
3. If the source volumes are not already in a CG, then create a new CG with the desired volumes.
4. If the destination volumes are not already in a CG, create a new destination CG with the desired volumes.
5. Create the new SnapMirror relationship using the source and destination CGs.
6. Resync the new SnapMirror relationship.

◀◀◀

## Changing the Composition of a CG Involved in a SnapMirror Relationship

In general, SnapMirror relationships for CGs support the following lifecycle functions:

- Modifying CG snapshot policies.
- Adding volumes to source CG.
- Adding or deleting LUNs on protected constituent volumes.
- Adding or deleting NVMe namespaces on protected constituent volumes.
- Retrieving information about the SnapMirror relationship or the protected CG.
- Modifying volume attributes while being part of the protected CG.
- Resizing volumes that are constituent members of a protected CG.

### Interchanging or Removing Constituent Volumes

Consider the scenario where it may be necessary to remove member volumes or exchange (delete and add) member volumes in an existing CG that has a SnapMirror relationship in place. The following steps can be used to accomplish this task:

**Procedure ▶▶▶**

1. Delete the current SnapMirror relationship for the CG on the destination cluster.
2. Release the SnapMirror relationship with the `-relationship-info-only true` parameter on the source cluster.
3. Add and/or remove the desired constituent volumes from the CG.
4. Recreate the new SnapMirror relationship using the source and destination CGs.
5. Resync the new SnapMirror relationship.

◀◀◀

**Note**

The success of the resync operation depends on the availability of a common, valid CG snapshot between the new source and destination CGs.

### Adding Constituent Volumes to a CG

Consider the scenario where additional volumes need to be added to an existing CG with an existing SnapMirror relationship. Volumes can be added to the source CG without interrupting the SnapMirror replication relationship on the source CG. Once the new volumes have been added to the source CG, SnapMirror will perform the following activities automatically in the next scheduled or manual SnapMirror update operation:

**Procedure ▶▶▶**

1. SnapMirror detects the difference in CG composition between the source and the destination.
2. Destination cluster creates new volume(s) to match the added source CG volumes.
3. Adds the newly created volumes to the destination CG.

4 Execute a resync operations which will:

- Perform a baseline replication on new volumes.
- Resync original volumes to get them all to same consistency point.

In case there is a need to failover to the destination before the non-disruptive expand resync completes, the CG snapshot prior to the ONTAP CG expand will be available. The new CG snapshots on the expanded ONTAP CG will be available after the successful resync operation.

# SnapMirror for CGs Interoperability with Other ONTAP Features

Table 13 defines how SnapMirror relationships for CGs interoperates with other ONTAP features.

Table 13 SnapMirror interoperability with other ONTAP features

| Category | Feature | Definition |
|---|---|---|
| Mobility | Volume move | Constituent volumes can be moved to another aggregate while in protected CG. |
| | FlexClone volumes | Constituent volumes can be cloned while in a protected CG. |
| | Volume rehost | Constituent volumes cannot be rehosted to another SVM while in a protected CG. |
| | LUN move | Supported |
| | LUN clone | Supported |
| Data Protection | Snapshot copies | Snapshots associated with the SnapMirror CG are replicated. Starting with ONTAP 9.14.1, Snapshots created on constituent volumes independently of the CG are replicated if they match the SnapMirror policy rules. |
| | Volume SnapMirror | Constituent volumes can have separate volume-scoped asynchronous SnapMirror relationships while in a protected CG. |
| | SVM DR | Starting with ONTAP 9.14.1, SVM DR relationships can be defined for SVMs containing SnapMirror compliant CGs. |
| | MetroCluster | Supported |
| | SnapLock | SnapLock volumes cannot be members of a protected CG. |
| Lifecycle Management | FlexGroup volumes | FlexGroup volumes cannot be members of a protected CG. |
| | FabricPool | Supported |
| | FlexCache | FlexCache is not compatible with protected CGs. |
| | Volume encryption | Member volumes can be encrypted at the volume level. |
| | Aggregate encryption | Member volume can reside on encrypted aggregates. |

# 7. SVM DR

SnapMirror can be used to protect entire SVMs and is called SVM DR. SVM DR replicates FlexVol volumes, FlexGroup volumes, and consistency groups owned by the SVM as well as the SVM configuration and identity information to a remote destination for NAS protocols. Although SVM DR uses the same SnapMirror replication technology as SnapMirror for volumes, there are differences (listed in Table 14).

Table 14 Differences between SVM DR and SnapMirror

| SVM DR | SnapMirror |
|---|---|
| Works at SVM-level granularity | Works at volume-level granularity |
| Protects FlexVol, FlexGroup, and CG data containers as well as SVM configuration | Protects only data stored in FlexVol, CG, and FlexGroup volumes |
| Minimum supported RPO:<br>• SVMs containing FlexVol volumes: 15 min<br>• SVMs containing FlexGroup volumes: 30 min<br>• SVMs containing consistency groups: 30 min | Minimum supported RPO:<br>• FlexVol volumes: 5 min<br>• FlexGroup volumes: 15 min<br>• Consistency groups: 30 min |

Common use cases for SVM DR include, but are not limited to, the following:

- DR of entire SVMs in case of loss of access to a cluster, site, or data center.
- Provide load balancing of data traffic between various clusters for content consumption.
- Development and testing purposes on an idle secondary site.
- Ensuring all data volumes are available for a specific tenant or application.
- Automatic protection of new FlexVol volumes, consistency groups, or FlexGroup volumes created in the protected SVM.

**Note**

SVM DR, when used for DR use cases, supports replication between clusters running the same version of ONTAP software. SVM DR used for use cases that do not have an expectation of recovery back to the original source cluster, such as using SVM DR for migration of SVMs and contained data volumes as part of a technology refresh, supports replication of SVMs from a source cluster up to two major versions lower than the destination cluster.

## Best Practice

Use SVM data mobility instead of SVM DR for migrating an SVM between clusters for reasons other than ongoing data protection. SVM data mobility is the preferred tool for the following use cases:

- Manual workload balancing for resource management
- One-time migration of an SVM for technology refresh

# Defining What SVM DR Will Protect

## SVM Data Replication

SVM DR can be used to protect volume data organized as FlexVol volumes, FlexGroup volumes, or consistency groups (ONTAP 9.14.1 or later) as described in "SnapMirror and Consistency Groups (CGs)" (page 63).

There are some additional limitations associated with SVMs containing CG:

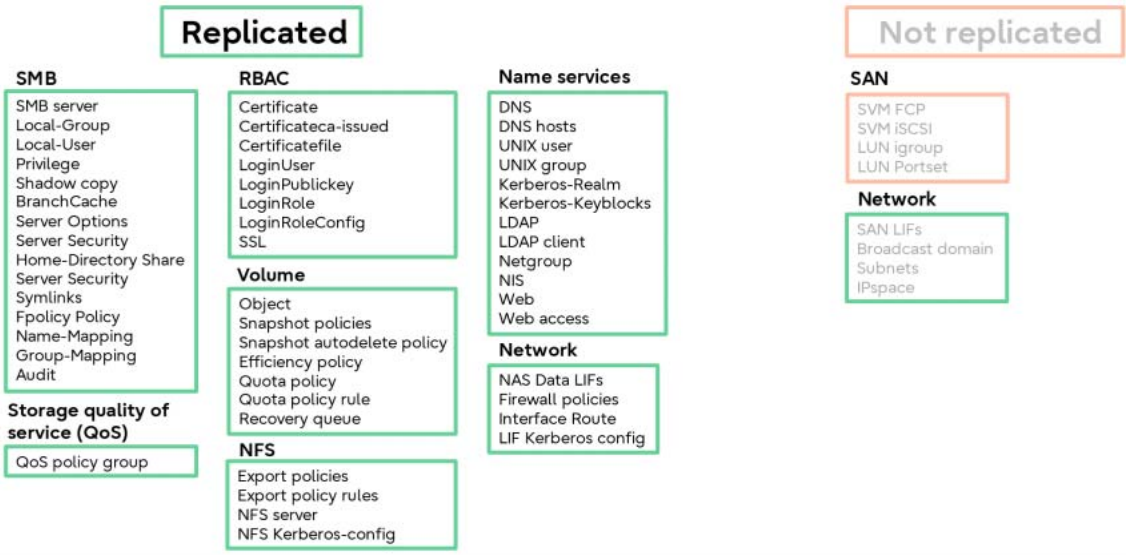
- CG-scoped asynchronous SnapMirror relationships cannot be converted to SVM DR relationships as is possible for FlexVol volume relationships, but it can be accomplished by performing some manual steps using the ONTAP CLI:
  1. Delete the CG relationship and release the CG async relationship with relationship-info-only.
  2. Rename the destination CG volumes to that of the source SVM CG volume names.
  3. Create and initialize/resync the SVM DR relationship.
  4. Verify the SVM DR relationship is in SnapMirrored, Idle with healthy set as true.
- SVM DR fan-out relationships are not supported if the source SVM contains a CG.
- SVM DR relationships from source SVMs residing on a sync-source SVM in a MetroCluster infrastructure are not supported.
- CGs cannot be excluded from SVM DR protection.
- The CGs in an SVM protected by SVM DR cannot have member volumes removed from the protected CGs.
- SVM DR will automatically create required DP member volumes and CG on the destination cluster. These volumes, and their containing CG, will have the same name as the source volumes and CG respectively.

## SVM Configuration Information

When creating an SVM DR relationship, SnapMirror can protect just volume or CG data or it can protect both the data and the SVM NAS configuration information. By default, SVM configuration information is not replicated.

For DR use cases, where it is desirable to enable NAS clients to reconnect to the NFS exports after the DR failover event, SnapMirror must be configured to replicate the SVM configuration information along with the SVM's volumes. To configure SnapMirror to replicate the SVM configuration information, use the `-identity-preserve true` parameter to the `snapmirror create` command. Figure 31 lists what is and is not replicated when `-identity-preserve` is true.

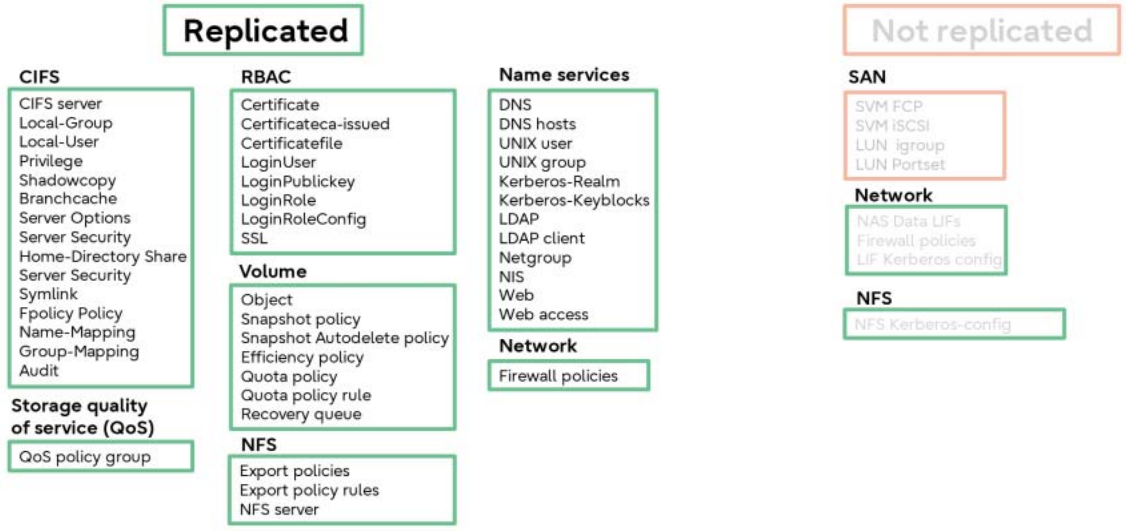
Figure 31 Replicated information for SVM DR using -identity-preserve true




It is also possible to replicate SVM configuration information without any of the network configuration information. This is desirable when the destination SVM is in a different IP subnet than the source SVM. To replicate SVM configuration information without network settings, a custom SnapMirror policy must be created that uses the `-discard-configs network` parameter. This parameter is only supported with SnapMirror policy type async-mirror. Figure 32 lists the additional information not replicated when the replication policy uses `-discard-configs network`.

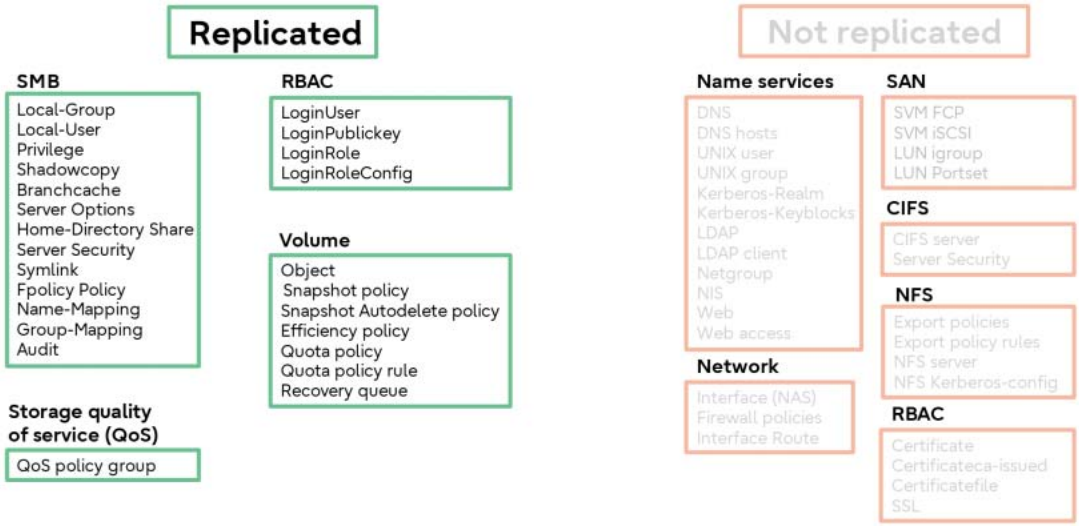
```
remote::>snapmirror policy modify -vserver vs0 -policy MySVMDRPolicy -type async-mirror -discard-configs network
```

Figure 32 Replicated information for SVM DR using -identity-preserve true with -discardconfigs network policy




By default, SVM DR does not replicate the SVM configuration information. This is the equivalent of using `-identity-preserve false` during the SVM DR relationship creation. Figure 33 lists what is and is not replicated by SVM DR when `-identity-preserve` is false.

Figure 33 Replicated information for SVM DR using `-identity-preserve false`




For details regarding which SVM configuration information is replicated, see About SnapMirror SVM replication.

# Creating an SVM DR Relationship

Creating an SVM DR relationship uses the same `snapmirror create` commands as volume scoped SnapMirror. Instead of providing source and destination volume parameters, SVM DR requires only providing the source and destination SVM names. This example demonstrates how to create an SVM DR relationship that replicates data volumes and SVM configuration information.

```
remote::>snapmirror create -source-cluster clust1 -source-vserver vs0 -destination-cluster remote
-destination-vserver vs1 -policy Asynchronous -schedule daily -type XDP
```

Prior to creating the SVM DR relationship, a target SVM must be created and designated as a data protection SVM. To designate an SVM as a data protection SVM, use the `-subtype dp-destination` parameter of the `vserver create` command.

```
remote::>vserver create -source-cluster clust1 -vserver vs1 -subtype dp-destination
```

# SVM DR Scalability

SVM DR has different scalability limits than volume scoped SnapMirror relationships. Table 15 lists the scalability limits of SVM DR data protection relationships.

Table 15 SVM DR scalability

| Parameter | Source SVM contains... | Limits |
|---|---|---|
| Number of SVM DR relationships per HA pair | FlexVol volumes only | ONTAP 9.9.1 and earlier: 32<br>ONTAP 9.10.1: 64<br>ONTAP 9.11.1 and later: 128 |
| | FlexGroup volumes | ONTAP 9.9.1 and later: 32 (*1) |
| | Consistency groups | ONTAP 9.14.1 and later: 32 (*1) |
| Number of volumes per SVM | | 300 |
| Number of volumes per HA pair | | 1,000 |
| Number of CG constituent volumes | | 16 |
| Number of FlexGroup constituent volumes | | 16 |

*1: When using FlexGroup volumes and/or CGs the total number combined cannot exceed 32.

# SVM DR Advanced Topics

## Limiting the Volumes Replicated as Part of SVM DR Relationship

By default, SVM DR replicates all data volumes contained within the source SVM. This has several advantages for SVM data and identity protection. Because SVM DR has a primary use case of ensuring access to the SVM's data in the case of a source SVM failure event, it might be important to ensure that all the data volumes owned by the SVM are replicated. This default behavior also enables automatic replication of any new volumes created in the protected SVM without any further administrative steps.

There might be a scenario where it is not desirable to replicate all volumes contained within an SVM. In these cases, ONTAP supports excluding one or more data volumes from the SVM DR replication relationship.

To exclude a volume from SVM DR replication, use the `volume modify` command with the `-vserver-dr-protection unprotected` parameter for each volume to be excluded. See Exclude volumes from SVM replication for more information.

**Note**

CGs and any member volumes cannot be excluded from replication.

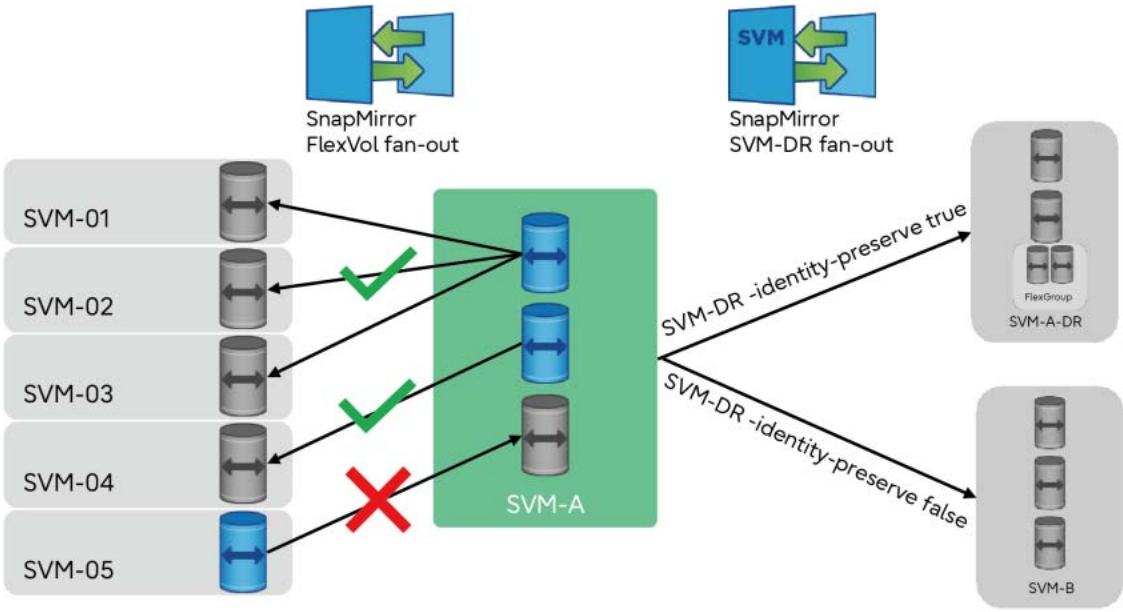
## SVM DR Fan-in and Fan-out

SVM DR supports limited fan-out capabilities. In general, a single source SVM can be replicated to up to two destination SVMs. In this configuration one of the SVM DR relationships can be configured with the `-identity-preserve true` parameter, and the second SVM DR relationship for the source SVM must use the `-identity-preserve false` parameter as shown in Figure 34.

Additionally, the following replication rules apply:

- Individual FlexVol volumes that are contained in an SVM DR source SVM can be a source volume for additional SnapMirror relationships (e.g., volume in SVM-A to volume in SVM-04).
- Individual FlexVol volume relationships can be fan-out relationships (e.g., volume in SVM-A to volumes SVM-01/02/03).
- SVM DR does not support replication of DP volumes (e.g., volume in SVM-05 to volume in SVM-A).
- There are additional restrictions for FlexGroup volumes and consistency groups.

Figure 34 SVM DR fan-out limitations for SVMs




SVM DR does not support fan-in of multiple source SVMs to the same destination SVM. An ONTAP cluster can be the destination for multiple SVM DR relationships assuming all SVMs have different names and configurations.

- SVM DR fan-out for FlexGroup volumes

  Starting with ONTAP 9.13.1, SVM DR supports SVM replication fan-out for SVMs containing FlexGroup volumes. Both the source and destination clusters must be running ONTAP 9.13.1 or later.

- SVM DR fan-out for consistency groups

  Fan-out of SVMs using SVM DR is not supported when the source SVM contains consistency groups.

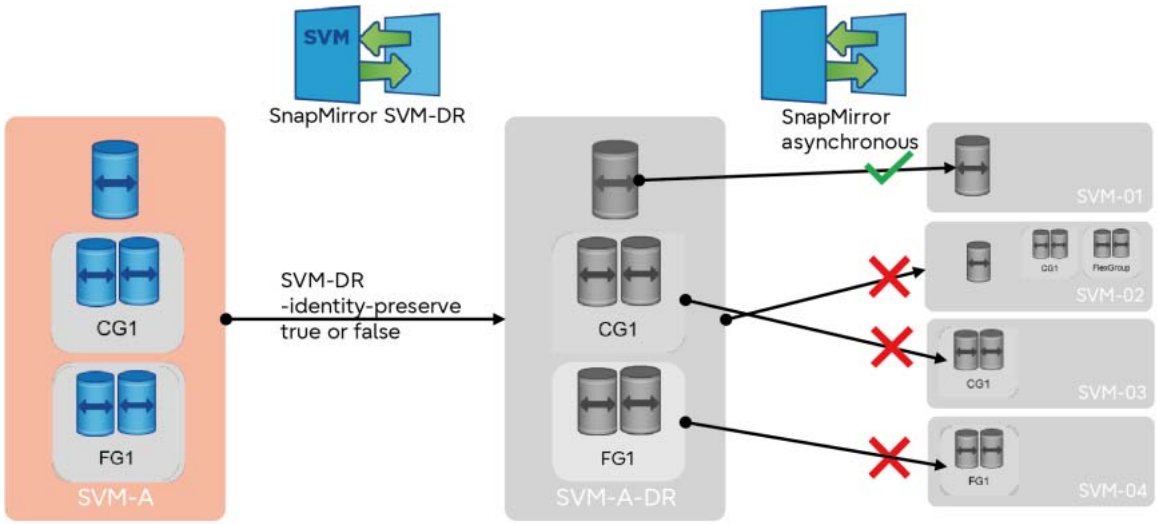
## SVM DR Cascade

Configuring a cascade relationship of a target SVM in an SVM DR relationship using SVM DR is not supported. Individual DP FlexVol volumes in an SVM DR destination SVM can be replicated SnapMirror asynchronous relationships. DP CGs and DP FlexGroup volumes are not allowed as sources for SnapMirror asynchronous relationships (Figure 35).

The following restrictions are imposed for secondary SnapMirror relationships from a dp-destination SVM containing CGs:

- Individual volume-scoped SnapMirror relationships are supported (e.g., single volume in SVM-A-DR to dp volume in SVM-01).
- SVM DR does not support cascade (e.g., SVM-A-DR to SVM-02).
- CG scoped SnapMirror relationships are not supported (e.g., CG1 in SVM-A-DR to CG1 in SVM-03).
- Flexgroup volume SnapMirror relationships are not supported (e.g., FG1 in SVM-A-DR to FG1 in SVM-04).

Figure 35 SVM DR cascade support




Volumes contained in the destination SVM of an SVM DR relationship that is using `-identity-preserve true` are not accessible because the destination SVM is not running as its configuration would conflict with the source SVM on the network (due to the SVM configuration data being the same on both the source and destination SVMs). This prevents SnapMirror from accessing the volumes as a source for a downstream replication relationship.

# SVM DR Interoperability

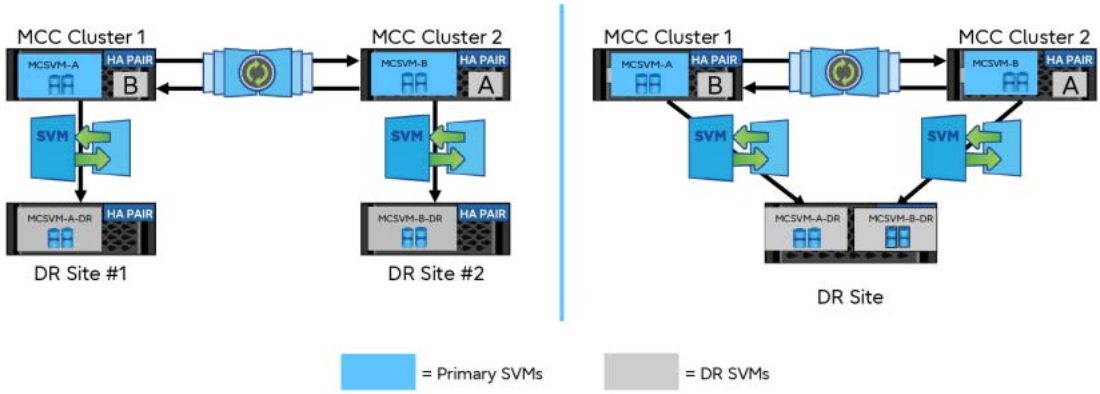
## SVM DR and SAN Protocols

SVM DR supports replication of iSCSI and FC LUNs and NVMe namespaces hosted on replicated volumes. SAN configuration information (initiator IDs, LUN IDs, igroups or SAN LIFs) are not replicated as part of the replicated SVM configuration data set. Additional configuration on the destination cluster, and potentially on clients accessing the namespaces or LUNs, is required.

## SVM DR and MetroCluster

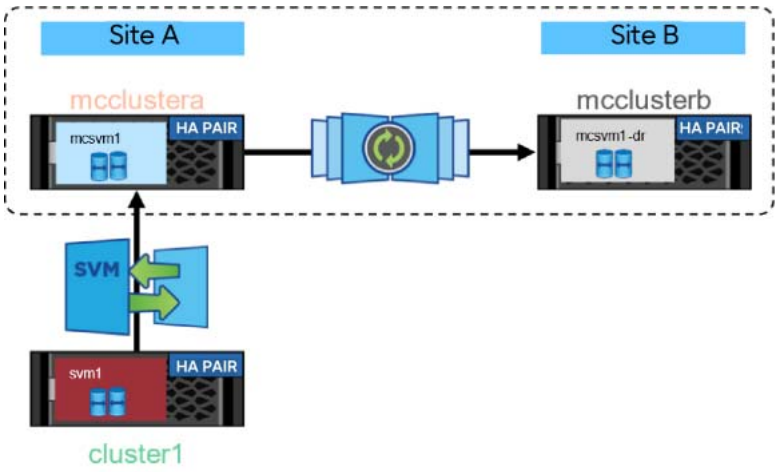
As of ONTAP 9.11.1, SVMs residing on either end of a MetroCluster relationship can be used as a source SVM for SVM DR. The combination of MetroCluster and SVM DR can be used to build a three-site DR topology that provides continuous availability at local or metropolitan distances and DR capabilities at longer distances with the ability maintain the SVM identity across all three sites, as shown in Figure 36.

Figure 36 SVM DR with MetroCluster




SVM DR can also be used to migrate an SVM from a stand-alone HA cluster to a MetroCluster infrastructure sync-source cluster (Figure 37).

Figure 37 SVM DR used to migrate an SVM from an HA cluster to a MetroCluster sync-source cluster




### Best Practice

- If all clusters are running ONTAP 9.16.1 or later, and the SVM contains only FlexVol volumes, the SVM data mobility tool (vserver migrate) can now be used to move an SVM in the following scenarios:
  - HA cluster to MetroCluster
  - MetroCluster to HA cluster
  - MetroCluster to MetroCluster
- Use SVM DR for migration of SVMs when the new destination is part of a MetroCluster source cluster and the ONTAP version is older than 9.16.1 on any cluster involved in the migration.

## SVM DR Support for FlexGroup Volumes

Starting with ONTAP 9.9.1, SVM DR can be used with FlexGroup volumes. With this support, SnapMirror SVM relationships transferred to the destination cluster will retain full awareness of FlexGroup volumes and properly mount those volumes in a DR scenario (Figure 38).

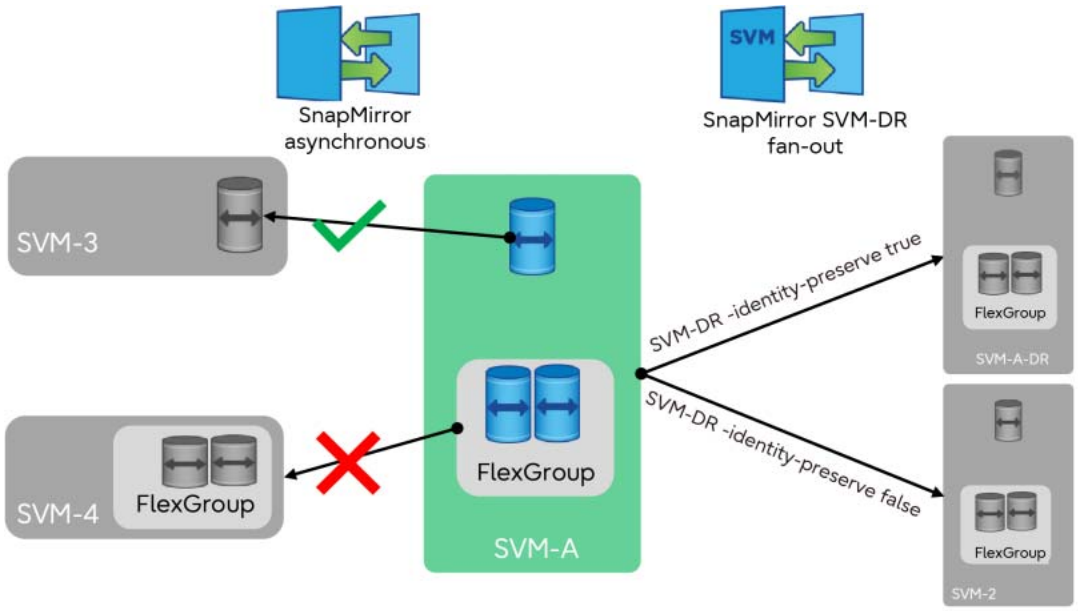
The following features are not supported for SVM DR using FlexGroup volumes:

- FlexClone volumes on source or destination clusters
- SnapMirror cascade configurations
- Converting FlexVol volumes to FlexGroup volumes

Starting with ONTAP 9.12.1, SnapMirror SVM DR supports replication of SVMs hosting both FlexGroup volumes and FabricPool volumes.

Starting with ONTAP 9.13.1, SnapMirror supports SVM DR fan-out configurations for source SVMs containing FlexGroup volumes.

Figure 38 SVM DR support for FlexGroup volumes




When a FlexGroup volume in a source SVM is modified, such as when a FlexGroup volume is expanded or contracted (add or remove constituent volumes), the changes are replicated to the destination SVM by the SVM DR relationship.

SVM DR uses a centralized ONTAP volume balanced placement algorithm to determine the location of replicated FlexVol volumes and FlexGroup volumes. The constituent volumes of a FlexGroup are distributed across all nodes and available suitable aggregates within the destination cluster.

FlexGroup volumes in a source SVM can be flagged for non-protection similar to FlexVol volumes using the `-vserverdr-protection` parameter of the FlexGroup volume.

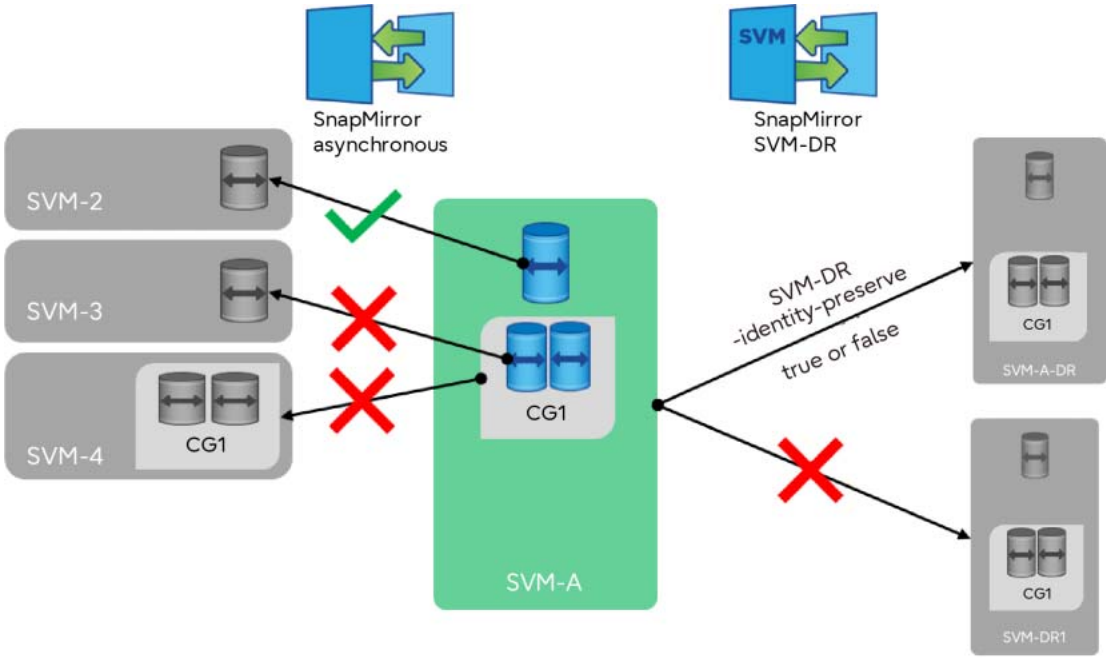
## SVM DR Support for Consistency Groups

Starting with ONTAP 9.14.1, SVM DR can be used to protect SVMs containing consistency groups (CGs).

The following limitations apply to CG-scoped SnapMirror relationships (Figure 39):

- SVM DR fan-out is not supported for SVMs containing CGs (e.g., SVM-A to SVM-DR1).
- CGs in an SVM DR protected source SVM cannot be replicated using a separate CG-scoped SnapMirror asynchronous relationships (e.g. volume in SVM-A replicating to volume in SVM-2).
- The member volumes in a CG that is in an SVM DR protected source SVM cannot be replicated using a separate volume-scoped SnapMirror relationship (member volume of CG1 to volume in SVM-3).

Figure 39 SVM DR support for consistency groups




## SVM DR and Cloud Target Interoperability

SVMs and SnapMirror are supported to some extent by Cloud Volumes ONTAP implementations offered by major cloud providers, the extent of SVM DR support for the Amazon FSx for NetApp service might be limited. It is imperative that customers work with their cloud provider to understand the full extent of support for SVMs, SnapMirror, and SVM DR.

SVM DR does not work with SnapMirror Cloud and the Cloud Backup Service.

# 8. Performance

There are multiple factors that can affect the performance of replication:

- Node CPU utilization
  CPUs will be shared between various data operations such as application data access and data protection operations.
- Number of concurrent SnapMirror operations
  Each transfer operation takes additional CPU cycles and network bandwidth to move data. The fewer concurrent transfers occurring at a given time, the faster each transfer operation will complete. The supported number of concurrent transfers supported will depend on node model and ONTAP version.
- Type of transfer: initialization or update
  A new SnapMirror relationship requires a baseline snapshot to be transferred that include all data in the volume at the time of relationship initialization. Subsequent updates will only transfer the differential data changes since the previous SnapMirror snapshot creation.
- Node hardware type
  The node model, configuration (including drive types, number of drives in the aggregate, number of volumes in the aggregate, and intercluster network physical port type) will directly affect SnapMirror performance.

# Calculate SnapMirror Throughput for Performance

Throughput for a relationship can be determined based on the amount of data moved over a set period. To determine throughput, the fields to note are the Transfer Size and Transfer Duration. To find the transfer throughput, divide the transfer size by the transfer duration.

```
cluster::> snapmirror show -destination-path vs3:dst -instance

                           Source Path: vs1:src_test
                      Destination Path: vs3:dst
                     Relationship Type: DP
                Relationship Group Type: none
                    SnapMirror Schedule: -
                 SnapMirror Policy Type: async-mirror
                      SnapMirror Policy: DPDefault
                            Tries Limit: -
                     Throttle (KB/sec): unlimited
                           Mirror State: Snapmirrored
                    Relationship Status: Transferring
                File Restore File Count: -
                 File Restore File List: -
                     Transfer Snapshot: snapmirror.89659724-bd35-11e4-9f11-
000c299bf0b8_2147484674.2015-03-02_134417
                     Snapshot Progress: 0B
                        Total Progress: 0B
              Network Compression Ratio: 2:1
                    Snapshot Checkpoint: 0B
                        Newest Snapshot: snapmirror.89659724-bd35-11e4-9f11-
000c299bf0b8_2147484674.2015-02-25_134212
            Newest Snapshot Timestamp: 02/25 13:22:08
                     Exported Snapshot: snapmirror.89659724-bd35-11e4-9f11-
000c299bf0b8_2147484674.2015-02-25_134212
           Exported Snapshot Timestamp: 02/25 13:22:08
                               Healthy: true
                    Unhealthy Reason: -
               Constituent Relationship: false
                Destination Volume Node: vsim
                        Relationship ID: d8b4cbc8-bd36-11e4-9f11-000c299bf0b8
                   Current Operation ID: 46da2fc6-c125-11e4-9f1a-000c299bf0b8
                          Transfer Type: update
                         Transfer Error: -
                     Current Throttle: unlimited
              Current Transfer Priority: normal
                     Last Transfer Type: initialize
                    Last Transfer Error: -
                      Last Transfer Size: 240GB
Last Transfer Network Compression Ratio: 3.1:1
                  Last Transfer Duration: 02:13:32
                     Last Transfer From: vs1:src_test
            Last Transfer End Timestamp: 02/25 13:42:15
                  Progress Last Updated: 03/02 13:44:17
                Relationship Capability: 8.2 and above
                                Lag Time: 120:22:10
          Number of Successful Updates: 0
              Number of Failed Updates: 0
          Number of Successful Resyncs: 0
              Number of Failed Resyncs: 0
           Number of Successful Breaks: 0
                Number of Failed Breaks: 0
                   Total Transfer Bytes: 245760
        Total Transfer Time in Seconds: 3
```

# SnapMirror and Network Compression

With increasing network bandwidth costs and increasing data growth, customers must do more with less. As the amount of data to be protected increases, more network bandwidth is needed to maintain the same RPO. Otherwise, replication times increase as the amount of data sent over the network to the DR site increases.

The SnapMirror native network compression feature can cut down on the amount of data replicated over the network. It also offers more flexibility and choices, as described in the following section.

## Maintaining the Same RPO Level

- **Challenge**
  Data replication needs are growing and additional bandwidth is needed to maintain the same level of RPO.
- **Solution**
  By using network compression, it is possible to maintain the same RPO without purchasing additional network bandwidth.

## Improve Your RPO without Buying Additional Bandwidth

- **Challenge**
  Network bandwidth is fully utilized. However, the customer wants to reduce their exposure to data loss and improve their RPO.
- **Solution**
  Enabling network compression improves the RPO without purchasing more network bandwidth.

## Use the Network Bandwidth for Other Purposes

- **Challenge**
  Due to other applications and services also require network bandwidth, all available network bandwidth required for data replication is insufficient.
- **Solution**
  By using network compression, it is possible to reduce the bandwidth consumed by SnapMirror without sacrificing RPO, thereby freeing up network bandwidth for other purposes.

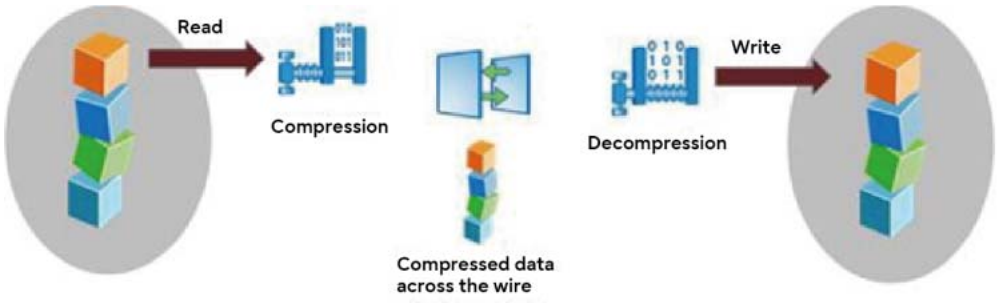
## Speeding Up the Initial Transfers

- **Challenge**
  Initial SnapMirror transfers can be large and therefore can take a long time to complete under bandwidth constraints.
- **Solution**
  Network compression can speed up the initial SnapMirror transfers.

# What Is SnapMirror Network Compression?

Network compression is natively built into SnapMirror to enable increased efficiency over the network for SnapMirror transfers. It does not, however, compress data at rest. SnapMirror network compression is not the same as volume compression. Figure 40 shows a very high-level flow of SnapMirror network compression.

Figure 40 SnapMirror network compression functional diagram




On the source system, the data blocks that must be sent to the destination system are handed off to the compression engine, which compresses the data blocks. The compression engine on the source system creates several threads, depending on the number of CPUs available on the storage system. These compression threads help to compress data in parallel. The compressed blocks are then sent over the network.

On the destination system, the compressed blocks are received and decompressed in parallel by using multiple threads. Decompressed data is then written to the appropriate volume.

## Enable and Disable Network Compression

SnapMirror network compression can be enabled or disabled by the `-is-network-compression-enabled` option in SnapMirror policy. It cannot be enabled for an active transfer. To enable compression for an existing transfer, first abort outstanding transfers, set the `-is-network-compression-enabled` option to `true` in the SnapMirror policy, and then resume the transfer.

### Best Practice

SnapMirror network compression increases resource utilization on both the SnapMirror source and destination systems. Therefore, evaluate the resource usage and benefits before deploying compression. For example, compression might not be useful for a high-bandwidth, low-latency connection. But it can be useful for connections that have relatively low bandwidth, such as WAN connections.

## Reporting the Compression Ratio

The SnapMirror network compression ratio is reported in the `snapmirror show -instance` output.

```
cluster::> snapmirror show -destination-path vs3:dst -instance

                            Source Path: vs1:src_test
                       Destination Path: vs3:dst
                      Relationship Type: DP
                Relationship Group Type: none
                    SnapMirror Schedule: -
                 SnapMirror Policy Type: async-mirror
                      SnapMirror Policy: DPDefault
                            Tries Limit: -
                      Throttle (KB/sec): unlimited
                           Mirror State: Snapmirrored
                    Relationship Status: Transferring
                File Restore File Count: -
                 File Restore File List: -
                      Transfer Snapshot: snapmirror.89659724-bd35-11e4-9f11-
000c299bf0b8_2147484674.2015-03-02_134417
                      Snapshot Progress: 0B
                         Total Progress: 0B
              Network Compression Ratio: 2:1
                     Snapshot Checkpoint: 0B
                         Newest Snapshot: snapmirror.89659724-bd35-11e4-9f11-
000c299bf0b8_2147484674.2015-02-25_134212
               Newest Snapshot Timestamp: 02/25 13:22:08
                       Exported Snapshot: snapmirror.89659724-bd35-11e4-9f11-
000c299bf0b8_2147484674.2015-02-25_134212
            Exported Snapshot Timestamp: 02/25 13:22:08
                                Healthy: true
                      Unhealthy Reason: -
               Constituent Relationship: false
                Destination Volume Node: vsim
                        Relationship ID: d8b4cbc8-bd36-11e4-9f11-000c299bf0b8
                   Current Operation ID: 46da2fc6-c125-11e4-9f1a-000c299bf0b8
                          Transfer Type: update
                         Transfer Error: -
                     Current Throttle: unlimited
               Current Transfer Priority: normal
                     Last Transfer Type: initialize
                    Last Transfer Error: -
                     Last Transfer Size: 240KB
Last Transfer Network Compression Ratio: 1:1
                 Last Transfer Duration: 0:0:3
                     Last Transfer From: vs1:src_test
             Last Transfer End Timestamp: 02/25 13:42:15
                   Progress Last Updated: 03/02 13:44:17
                Relationship Capability: 8.2 and above
                                Lag Time: 120:22:10
           Number of Successful Updates: 0
               Number of Failed Updates: 0
           Number of Successful Resyncs: 0
               Number of Failed Resyncs: 0
            Number of Successful Breaks: 0
                Number of Failed Breaks: 0
                   Total Transfer Bytes: 245760
         Total Transfer Time in Seconds: 3
```

Compression ratio is only shown in transferring state.

# SnapMirror Throttling

The SnapMirror throttle setting is used to throttle the network bandwidth consumed, which limits the amount of bandwidth used by intercluster SnapMirror. In other words, SnapMirror throttle does not control network bandwidth. Rather, it works by limiting the blocks that WAFL can use for SnapMirror transfers.

**Note**

All replication throttles in ONTAP are in kilobytes per second.

SnapMirror throttle can be set on a per relationship basis when creating a new relationship by using the `-throttle` option and by modifying an existing relationship with the `snapmirror modify` command. In this example, a 10MB throttle is applied to an existing relationship by using the `snapmirror modify` command.

```
cluster02::> snapmirror modify -destination-path vs1:vol1 -throttle 10240
```

**Caution**

- To change the throttle of an active SnapMirror relationship, terminate the existing transfer and restart it to use the new value. SnapMirror restarts the transfer from the last restart checkpoint by using the new throttle value, rather than restarting from the beginning.
- Intracluster throttle is supported, and it works the same way as intercluster throttle.

ONTAP 9 introduces global SnapMirror throttling available for each node in a cluster to perform SnapMirror transfer at a fixed maximum bandwidth for outgoing and incoming transfers. SnapMirror global throttling restricts the bandwidth used by incoming and/or outgoing SnapMirror transfers. The restriction is enforced cluster wide on all nodes in the cluster. This capability is in addition to the throttle for each SnapMirror relationship as described earlier. Each node has a global throttle for sender-side (outgoing) transfers as well as receiver-side (incoming) transfers and an option to enable or disable this throttling. The per-transfer throttle is capped at the node-level throttle if it exceeds the global node throttle value. Otherwise, the transfers take place at the specified value.

Global throttling works with the per-relationship throttle feature for SnapMirror transfers. The per-relationship throttle is enforced until the combined bandwidth of per-relationship transfers exceeds the value of the global throttle, after which the global throttle is enforced. A throttle value 0 implies that global throttling is disabled.

**Note**

Global throttling should not be enabled on clusters that have SnapMirror Synchronous relationships.

The minimum throttle bandwidth should be 4KBps, and the maximum can be up to 2TBps. A throttle bandwidth of 0 implies that the transfer is not throttled or that bandwidth is unlimited.

A new cluster-wide option to control throttling is as follows:

```
cluster::> options replication*

cluster
   replication.throttle.enable       on                  -
   replication.throttle.incoming.max_kbs
                                     4000                -
   replication.throttle.outgoing.max_kbs
                                     2000                -
3 entries were displayed.
```

Each entry can be edited individually. The enable option either enables or disables both the outgoing and incoming throttle.

```
cluster::> options replication.throttle.enable on
1 entry was modified.
```

Changing the outgoing and incoming throttle is reflected in the actual transfer only if the enable option is on. The outgoing and incoming throttle values can be changed irrespective of the enable option value.

```
cluster::> options replication.throttle.outgoing.max_kbs 8000
1 entry was modified.

cluster::> options replication.throttle.incoming.max_kbs 5000
1 entry was modified.
```

# How to Change TCP Receive Buffer Size

SnapMirror uses the network service `ctlopcp` with a tunable TCP receive buffer window for both intercluster (WAN network) and intra-cluster (LAN network) replication. The TCP receive buffer window is configured per cluster, and an increase in the TCP receive buffer size takes effect immediately with no requirement to reboot.

Table 16 TCP receive buffer windows

| | Default | Minimum | Maximum |
|---|---|---|---|
| Intercluster TCP receive buffer window | 2MB | 256KB | 7MB |
| Intracluster TCP receive buffer window | 256KB | 256KB | 7MB |

**Note**

The intercluster TCP receive buffer window autotunes by default. The window starts at 64KB per TCP stream and grow to a maximum of the configured value (default 2MB) to accommodate intercluster replications. Once grown, the receive buffer window does not shrink if the TCP stream remains open.

# Concurrent Replication Operations

The number of supported simultaneous SnapMirror operations is limited. This limit is per node and varies depending on the platform and version of ONTAP shown in Table 17.

Table 17 Maximum number of concurrent SnapMirror transfers per node by ONTAP version and controller model

| ONTAP version | ETERNUS AX series | | | | |
|---|---|---|---|---|---|
| | AX1100 | AX1200 | AX2100 | AX2200 | AX4100 |
| 9.7.x | 100 | – | 100 | – | 100 |
| 9.8.x | 100 | – | 100 | 100 | 100 |
| 9.9.x | 100 | – | 100 | 100 | 100 |
| 9.10.x | 100 | 100 | 100 | 100 | 100 |
| 9.11.x | 100 | 100 | 100 | 100 | 100 |
| 9.12.x | 100 | 100 | 100 | 100 | 100 |
| 9.13.x | 100 | 100 | 100 | 100 | 100 |
| 9.14.x | 100 | 100 | 100 | 100 | 100 |

| ONTAP version | ETERNUS AX series ASA | | | |
|---|---|---|---|---|
| | AX1200 ASA (*1) | AX2100 ASA | AX2200 ASA | AX4100 ASA |
| 9.7.x | – | 100 | – | – |
| 9.8.x | – | 100 | 100 | 100 |
| 9.9.x | – | 100 | 100 | 100 |
| 9.10.x | – | 100 | 100 | 100 |
| 9.11.x | – | 100 | 100 | 100 |
| 9.12.x | – | 100 | 100 | 100 |
| 9.13.x | 100 | 100 | 100 | 100 |
| 9.14.x | 100 | 100 | 100 | 100 |

*1: ONTAP 9.12P1 or later

| ONTAP version | ETERNUS AC series | ETERNUS AC series ASA |
|---|---|---|
| | AC2100 | AC2100 ASA |
| 9.7.x | – | – |
| 9.8.x | – | – |
| 9.9.x | – | – |
| 9.10.x | – | – |
| 9.11.x | 100 | 100 |
| 9.12.x | 100 | 100 |
| 9.13.x | 100 | 100 |
| 9.14.x | 100 | 100 |

| ONTAP version | ETERNUS HX series | | | |
|---|---|---|---|---|
| | HX2100 | HX2200 | HX2300 | HX6100 |
| 9.7.x | 100 | 100 | – | 100 |
| 9.8.x | 100 | 100 | – | 100 |
| 9.9.x | 100 | 100 | – | 100 |
| 9.10.x | 100 | 100 | – | 100 |
| 9.11.x | 100 | 100 | – | 100 |
| 9.12.x | 100 | 100 | – | 100 |
| 9.13.x | 100 | 100 | 100 | 100 |
| 9.14.x | 100 | 100 | 100 | 100 |

- Best Practices
  - When planning concurrent operations, it is a best practice to consider the frequency of volume move and volume copy operations in the environment in addition to SnapMirror replications.
  - Size the system correctly with enough CPU headroom to allow the CPU workload to execute.

ONTAP provides a greater level of scalability by allowing expansion of a cluster beyond two nodes. Each node in the cluster provides CPU and memory resources that are used for replication of volumes owned by that node.

- Best Practice

  To optimize replication, distribute replicated volumes across different nodes in the clusters rather than placing all volumes requiring replication on a single node. This best practice allows all nodes in the cluster to share replication activity.

# Network Sizing Requirements

When deploying SnapMirror, consider the round-trip travel time of a packet from the source to the destination storage system, because network distance causes write latency. A network with the appropriate bandwidth available to transfer the system data is required to support the desired replication interval, so that application performance is not affected. There are limitations on the network characteristics that are supported for intercluster replication.

## Network Sizing Requirements for Intercluster Replication

The intercluster network must be sized appropriately depending on the data change rate and the update interval to meet the RPO of the solution and individual node performance characteristics. Intercluster SnapMirror is supported across networks that have the following characteristics:

- A minimum bandwidth of 0.5Mbps
- A packet loss of 1%

### Best Practice

All paths used for intercluster replication must have equal performance characteristics. Configuring multipathing in such a way that a node has one intercluster LIF on a slow path and another intercluster LIF on a fast path degrades performance, because data is multiplexed across both paths simultaneously.

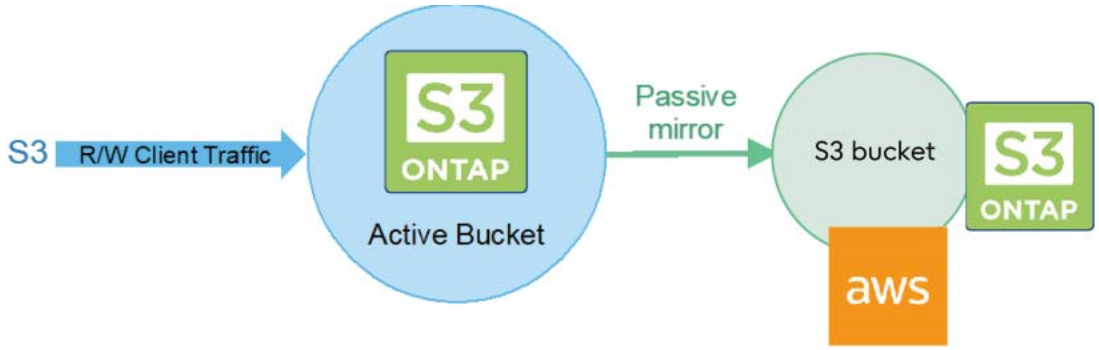
## Network Sizing Requirements for Intracluster Replication

All intracluster transfers, including SnapMirror, volume move, and volume copy operations, use the private cluster interconnect between nodes in the same cluster whose bandwidth is not configurable.

# 9. S3 SnapMirror

ONTAP 9.10.1 introduced ONTAP S3 SnapMirror. S3 SnapMirror provides customers with a native replication and backup solution for their ONTAP S3 object stores. S3 SnapMirror supports a wide variety of S3 targets for data protection and DR including another ONTAP S3 bucket, StorageGRID, ONTAP Cloud Volumes ONTAP S3 buckets and native cloud-provided S3 buckets such as AWS S3 (Figure 41).

Figure 41 ONTAP S3 SnapMirror overview




S3 SnapMirror uses a specialized replication engine different from the standard LRSE replication engine used by SnapMirror for FlexVolume and FlexGroup asynchronous replication. When setting up S3 SnapMirror relationships, the replication protection policy used is Continuous (`-type continuous`).

For more information on S3 SnapMirror, see S3 SnapMirror overview.

# 10. SVM Data Mobility

SVM data mobility (SVM migration) is a feature that allows a cluster administrator to move an SVM (including data and SVM configuration information) from one cluster to another. This feature is not dependent on having a previous SVM DR relationship configured for the SVM being moved.

SVM migration does not support SAN protocols.

Nondisruptive (NDO) SVM migration is supported between ETERNUS AX series HA pairs for NFS 3, NFS 4.1, NFS 4.2, and pNFS workloads.

Table 18 lists the maximum size of the source and destination cluster eligible for SVM data mobility use.

Table 18 SVM data mobility cluster scalability support

| ONTAP version | Maximum cluster size |
|---|---|
| ONTAP 9.10.1 and older | 1 HA pair |
| ONTAP 9.12.1 and later | 3 HA pairs |
| ONTAP 9.14.1 and later | 12 HA pairs |

Table 19 summarizes the support limits for SVM data mobility.

Table 19 SVM migration support summary

| Feature | SVM data mobility |
|---|---|
| Scale | ONTAP 9.13.1 and older: 100 FlexVol volumes<br>ONTAP 9.14.1 and later: 400 FlexVol volumes |
| Network | L2 with < 10ms RTT latency |
| Platform | ETERNUS AX/AC/HX series<br>ONTAP 9.13.1 and later |
| Protocols | NFS 3, NFS 4.1, NFS 4.2 |
| Data management | Snapshot copies<br>Storage efficiency |
| Data security | VE<br>OKM<br>External Key Management (EKM) |

# 11. Hardware Interoperability

Check with your sales representative to make sure that the versions of the product and features are supported. In addition, verify that the source and destination volumes are running compatible ONTAP versions before creating a SnapMirror data protection relationship.

# 12. Troubleshooting Tips

## Troubleshooting Cluster Peer Relationships

**Procedure ▶▶▶**

1. Run the `cluster peer show` command to verify the availability of the cluster peer relationship. This command displays all existing configured cluster peer relationships.

```
cluster01::> cluster peer show
Peer Cluster Name         Cluster Serial Number Availability
------------------------- --------------------- --------------
cluster02                 1-80-000013           Available
```

2. Add `-instance` to the command to view more detailed information about the cluster peers. Include `-cluster <cluster_name>` to view results for a specific cluster. The `-instance` option displays the remote addresses that are used for intercluster communication.

```
cluster01::> cluster peer show -cluster cluster02 -instance
            Peer Cluster Name: cluster02
Remote Intercluster Addresses: 10.12.12.3,10.12.12.4
                Availability: Available
          Remote Cluster Name: cluster02
          Active IP Addresses: 10.12.12.3,10.12.12.4
        Cluster Serial Number: 1-80-000013
```

3. Run the `cluster peer ping` command to view information about connectivity between each intercluster address, including RTT response times. For multiple configured cluster peers, use the `-cluster <cluster_name>` option to perform the ping for one specific peer relationship. The `cluster peer ping` command displays the results of a ping between intercluster interfaces. As mentioned earlier, when performing intercluster SnapMirror mirroring over multiple paths between the local and remote clusters, each path must have the same performance characteristics. In this example, the ping response times (RTTs) are comparatively equal to the pings to nodes where the destination cluster displays as `cluster02`.

```
cluster01::> cluster peer ping cluster02

Node: cluster01-01           Destination Cluster: cluster01
Destination Node IP Address       Count TTL  RTT(ms) Status
---------------- ---------------- ----- ---- ------- ------------------
cluster01-01     10.12.12.1       1     255  0.186   interface_reachable
cluster01-02     10.12.12.2       1     255  1.156   interface_reachable

Node: cluster01-01           Destination Cluster: cluster02
Destination Node IP Address       Count TTL  RTT(ms) Status
---------------- ---------------- ----- ---- ------- ------------------
cluster02-01     10.12.12.3       1     255  7.164   interface_reachable
cluster02-02     10.12.12.4       1     255  7.065   interface_reachable

Node: cluster01-02           Destination Cluster: cluster01
Destination Node IP Address       Count TTL  RTT(ms) Status
---------------- ---------------- ----- ---- ------- ------------------
cluster01-01     10.12.12.1       1     255  1.324   interface_reachable
cluster01-02     10.12.12.2       1     255  0.809   interface_reachable

Node: cluster01-02           Destination Cluster: cluster02
Destination Node IP Address       Count TTL  RTT(ms) Status
---------------- ---------------- ----- ---- ------- ------------------
cluster02-01     10.12.12.3       1     255  7.279   interface_reachable
cluster02-02     10.12.12.4       1     255  7.282   interface_reachable
```

◀◀◀

# Troubleshooting SVM Peer Relationships

Here is a list of common issues and how to troubleshoot them:

- SVM peer action failure for the intercluster environment:
  - Verify that the peer cluster is reachable.
  - Verify that both clusters support ONTAP versions with SVM peering capability enabled.
  - Verify that the peer SVM name is not associated with another cluster from peer SVM names in the SVM peering table.
  - Check `mgwd.log` and the console logs for error messages.
- SVM peer action failure for the intra-cluster or intercluster environment:
  - Verify that both clusters supported ONTAP versions, with SVM peering capability enabled. Verify that local and peer SVM names are not the same.
  - Check `mgwd.log` and the console logs for error messages.
- Run the `vserver peer show` command to verify the SVM peer relationship. This command displays all existing configured SVM peer relationships.

```
cluster02::> vserver peer show
            Peer        Peer
Vserver     Vserver     State
----------- ----------- ------------
vs1_dest    vs1_backup  peered
vs1_dest    vs1_src     peered
```

- Check for any notifications with the `vserver peer show-all` command.

```
cluster02::> vserver peer show-all
            Peer        Peer                                  Peering
Vserver     Vserver     State        Peer Cluster             Applications
----------- ----------- ------------ ------------------------ ---------------
vs1_dest    vs1_backup  peered       cluster03                snapmirror
vs1_dest    vs1_src     peered       cluster01                snapmirror
```

# Understanding SnapMirror Relationship Status

The Healthy column indicates the SnapMirror relationship status. This column is shown in the output of the `snapmirror show` command on the CLI and as the Healthy column in the displayed status of SnapMirror relationships in System Manager.

```
cluster02::> snapmirror show
                                                                    Progress
Source            Destination  Mirror  Relationship  Total            Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
vs1_src:vol1
            XDP   vs1_dest:vol1
                              Snapmirrored
                                      Transferring   128KB     true    02/25 15:43:53
```

The Mirror State column also displays if the destination volume is offline or if it cannot be reached.

# Troubleshooting SnapMirror Relationships

To determine when the last SnapMirror transfer for a specific relationship completed, see the Exported Snapshot Timestamp field for instance information.

```
cluster02::> snapmirror show -instance

                           Source Path: snap_src1:SMSource
                      Destination Path: svm_dst1:SMSource_dest
                     Relationship Type: XDP
                Relationship Group Type: none
                    SnapMirror Schedule: -
                 SnapMirror Policy Type: vault
                      SnapMirror Policy: XDPDefault
                            Tries Limit: -
                     Throttle (KB/sec): unlimited
                           Mirror State: Snapmirrored
                    Relationship Status: Idle
                File Restore File Count: -
                 File Restore File List: -
                      Transfer Snapshot: -
                      Snapshot Progress: -
                         Total Progress: -
            Network Compression Ratio: -
                  Snapshot Checkpoint: -
                      Newest Snapshot: snapmirror.12ceb7f0-b078-11e8-baec-0050
56b013db_2160175147.2020-01-24_043858
            Newest Snapshot Timestamp: 01/24 04:38:59
                     Exported Snapshot: snapmirror.12ceb7f0-b078-11e8-baec-0050
56b013db_2160175147.2020-01-24_043858
          Exported Snapshot Timestamp: 01/24 04:38:59
                              Healthy: true
                     Unhealthy Reason: -
              Constituent Relationship: false
               Destination Volume Node: cluster2-01
                       Relationship ID: 1a46a611-3e64-11ea-86bf-005056b013db
                  Current Operation ID: -
                         Transfer Type: -
                        Transfer Error: -
                     Current Throttle: -
            Current Transfer Priority: -
                   Last Transfer Type: resync
                  Last Transfer Error: -
                   Last Transfer Size: 0B
Last Transfer Network Compression Ratio: 1:1
               Last Transfer Duration: 0:0:1
                   Last Transfer From: snap_src1:SMSource
          Last Transfer End Timestamp: 01/24 04:45:16
                Progress Last Updated: -
              Relationship Capability: 8.2 and above
                             Lag Time: 5:27:1
          Identity Preserve Vserver DR: -
                Volume MSIDs Preserved: -
                Is Auto Expand Enabled: -
          Number of Successful Updates: 0
              Number of Failed Updates: 0
          Number of Successful Resyncs: 1
              Number of Failed Resyncs: 0
           Number of Successful Breaks: 0
                Number of Failed Breaks: 0
                   Total Transfer Bytes: 0
        Total Transfer Time in Seconds: 1
```

For SnapMirror relationship issues, review information about relationships in the event log. Use the –messagename option with the event log show command to filter the event log for messages related to SnapMirror, as shown in the following example. Specify the mgmt.snapmir* message name to filter the output and find only messages related to SnapMirror.

```
cluster01::> event log show -messagename mgmt.snapmir*
Time                Node             Severity      Event
------------------- ---------------- ------------- ---------------------------
12/6/2011 17:35     cluster02-01     ERROR         mgmt.snapmir.update.fail: Update from source
volume 'cluster01://vs1/vol03' to destination volume(s) 'cluster02://vs2/vol03' failed with error
'Failed to setup transfer. (Duplicate transfer specified. (Other error.))'. Job ID 1322.
12/6/2011 17:34:35  cluster02-01     DEBUG         mgmt.snapmir.abnormal.abort: Source Path
cluster01://vs1/vol01, Destination Path cluster02://vs2/vol01, Error Transfer failed.
(Destination volume cluster02://vs2/vol01 is smaller than the source volume.), Function
copySnapshot, line 5030, job ID 1355.
12/5/2011 05:15:45  cluster02-01     DEBUG         mgmt.snapmir.abnormal.abort: Source Path
cluster01://vs2/vol12, Destination Path cluster02://vs8/vol12, Error Failed to delete Snapshot
copy weekly.2011-12-04_0015 on volume cluster02://vs8/vol12. (Snapshot is in use.), Function
deleteSnapshot, line 4285, job ID 1215.
```

To find an error message about a specific volume, filter the message list further by specifying the name of the volume, enclosed in asterisks, with the –event option, as shown in the following example.

```
cluster01::> event log show -messagename mgmt.snapmir* -event *vol01*
Time                Node             Severity      Event
------------------- ---------------- ------------- ---------------------------
12/6/2011 17:34:35  cluster02-01     DEBUG         mgmt.snapmir.abnormal.abort: Source Path
cluster01://vs1/vol01, Destination Path cluster02://vs2/vol01, Error Transfer failed.
(Destination volume cluster02://vs2/vol01 is smaller than the source volume.), Function
copySnapshot, line 5030, job ID 1355.
```

All SnapMirror events are logged to the SnapMirror_audit.log and SnapMirror_error.log files on the node where the destination volume resides. This node might be different from the one where the command was issued. The node running the operation can be determined by running the snapmirror show -fields destination-volume-node command. System Manager allows viewing of the SnapMirror log files.

# 13. Best Practices for DR Configurations

## Best Practices

- **Replicate volumes hosted by a single SVM on the source to a single SVM on the destination.**
  An SVM hosts the root of a NAS namespace for NAS clients and a single storage target in SAN environments. If some NAS volumes are replicated from one SVM into different SVMs at the destination, then those volumes cannot be recovered into the same namespace. The same is true of volumes containing LUNs. If the volumes are replicated into different SVMs at the destination, then all the LUNs are not presented under the same SAN target.
- **Ensure that the source and destination SVMs are hosted within the same Active Directory, LDAP, or NIS domain.**
  This configuration is required so that access control lists (ACLs) stored in NAS files are not broken if a NAS volume is recovered into an SVM that cannot authenticate those ACLs. The process of changing file-level ACLs to reenable access from a different domain can be extremely difficult and time consuming. It is also important for authentication of tools running in SAN clients such as SnapCenter Plug-in for Windows.
- **Consider using DNS aliases to associate source and destination SVMs for volume-based DR use cases.**
  Volume-based SnapMirror replication requires that source and destination SVMs use different names to support the ability to access both SVMs as needed for secondary activities such as backup to tape or cloud from the destination SVM. In a DR failover scenario, clients must access the volumes from the secondary site. Although changing the DNS records for the destination SVM to reflect the source SVM's name is an option, creating DNS alias records is recommended. This practice makes sure that SMB shares are still accessible using the same Uniform Naming Convention (UNC) path name or NFS exports they did prior to the DR event.
- **Consider using SnapMirror SVM DR when an entire SMB or NFS namespace needs to be protected.**
  SVM DR using the `-identity-preserve true` parameter enables client access to the namespace by using the same client configuration in the event of a DR failover scenario. This also ensures that the namespace is identical because all replicated volumes within the SVM will have the same names.
- **When using volume replication, consider using the same volume names in both the source and destination SVMs.**
  Although using destination volume names that are the same as the source volume names is not required, this practice can make mounting destination volumes into the destination simpler to manage if the junction path where the volume is mounted also has the same name as the volume.
- **Many SAN clients cannot access a LUN that resides in a read-only container, such as a SnapMirror destination volume.**
  Generally, LUNs should be mapped to igroups and mounted by SAN clients after the SnapMirror break operation is performed.
- **Configure the destination SVMs ahead of time as described in the following section.**
  This approach can greatly speed up the storage system DR process, possibly reducing it to a few SnapMirror break operations and the update of some DNS aliases.
- **If not using SVM DR, as new volumes are created at the source site, SnapMirror volume relationships must be created to replicate those volumes.**
  Make configuration settings pertaining to those volumes in the DR site after the volumes are created and replicated so they can be ready in the event of a disaster.

# 14. Configuration and Failover for DR

Configuration and failover for DR is presented here in an overview of the DR process for intracluster SnapMirror data protection (async-mirror) replication. The process is presented in two sections. The first section provides steps that must be completed before a failover is required to prepare the destination for failover. These steps should be completed to prepare the DR site for a DR scenario. The second section provides the steps necessary to perform a failover.

Every environment has its own unique characteristics. Each environment can affect a DR plan. Depending on the type of DR solutions deployed each organization's DR situation can be very different. To enable success, proper planning, documentation, and a realistic walkthrough of a DR scenario are required.

## Environment Failover Requirements and Assumptions

To provide a successful DR experience, consider some general requirements and assumptions. The following is not an all-inclusive list:

- System administrator access to a workstation or server desktop session from which to administer the DR site and perform the failover.
- System administrators have all appropriate credentials, accounts, passwords, and so on required to access the systems.
- Connectivity to the DR network is available from wherever operations are performed.
- Certain infrastructure servers already exist in the DR site and are accessible. These systems provide basic services necessary for the administrators to work in the environment and execute the recovery plan.
  - DR site Active Directory or LDAP services to provide authentication.
  - DR site DNS services to provide name resolution.
  - DR site license servers to provide licensing services for all applications that require them.

**Note**

A server must be available at the DR site to perform the necessary Active Directory FS MO roles. For information regarding transferring roles to a surviving Active Directory server or seizing these roles from a failed server, see Microsoft KB 255504.

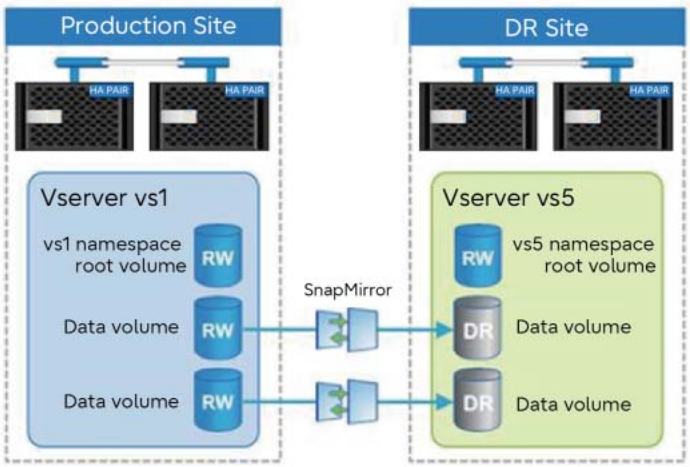
- The DR site has time synchronized to the same source as the primary site or a source in sync with the primary site.
- All required volumes are replicated using SnapMirror to the DR site.
- SnapMirror operations have been monitored and are up to date with respect to the designed RPO.
- The required capacity exists on the DR controller. This refers to the capacity required to support day-to-day operations that have been planned for in the DR environment.
- All DR site application servers have the proper connectivity configured to be able to connect to the DR storage arrays.
- A method exists to isolate or fence the failed primary network from the DR site. This approach is necessary if the event causing the disaster is temporary or intermittent in nature, such as an extended power outage. When the primary site systems restart, services might conflict with the recovered operations that are then running at the DR site.
- Plans have been made for providing users and applications access to the data and services at the DR site. For example, updating DNS records such that home directory mount requests to the primary site SVM are directed to the DR site SVM instead.

# Preparing the Destination for Failover

Many parts of a DR process can be prepared ahead of time prior to a DR event. For example, mounting volumes into the namespace, creating SMB shares, and assigning NFS export policies, can all be performed ahead of time. SnapMirror volume replication cannot be used to replicate configuration information that could be independent in the destination SVMs. These configurations include SVM domain membership, SMB configuration, NFS policies, Snapshot policy schedules, or storage efficiency policies.

Figure 42 illustrates volume layout for DR.

Figure 42 Volume layout for DR




After volumes have been replicated, complete the following steps to prepare the destination system for failover.

## NAS and SAN Environments

**Procedure ▶▶▶**

1. Configure the destination SVM membership into the appropriate Active Directory, LDAP, or NIS domain.
2. Determine that the destination SVM is a member of the same domain as the source SVM so that authentication is not broken for tools such as SnapCenter. This configuration also ensures that the same users can be authenticated against file-level ACLs that are replicated by SnapMirror.
3. Create any nondefault Snapshot copy policies needed in the destination cluster.

**Caution**

Configuring Snapshot copy policies in the destination cluster with the same schedules as those in the source is recommended. Snapshot copy policies must be applied after failover (`snapmirror break`) has completed.

4. Create storage efficiency policies in the destination SVM.

**Caution**

If storage efficiency policies are assigned to the volumes in the source SVM, an identical policy must be created in the destination SVM in order to schedule the dedupe process after failover at the DR site. These storage efficiency policies must be applied after failover (`snapmirror break`) has completed.

## NAS Only Environments

**Procedure**

1. Verify that all necessary volumes in the source SVM are being replicated to the destination SVM. Volumes can be mounted in subfolders or inside other volumes in the namespace. If this condition exists, it is important to make sure that all the volumes required to properly reconstruct the namespace at the destination are being replicated.
2. Verify the security style and permissions on the destination SVM root volume. The security style and permissions of the root of the destination SVM namespace must be set correctly, or the NAS namespace might be inaccessible after failover.
3. Mount the destination NAS volumes into the destination SVM namespace.
   SnapMirror does not replicate the SVM namespace junction path information. NAS volumes have no junction path, so they are not accessible after a SnapMirror break occurs unless they are premounted before failover or until they are mounted after failover.
   When mounting the volumes, mount them into the namespace by using the same junction path into which the source volume was mounted in the source SVM. This configuration is important so that paths in the recovered namespace are not different than paths that existed at the primary site. If the paths are different, then client mount points, links, shortcuts, and aliases might not be able to find the correct paths.

   **Note**

   Volumes cannot be mounted inside (nested in) other volumes that are still in a DP state. After using the `snapmirror break` command, any volume that has a mount point nested inside a replicated volume must be mounted, and any CIFS shares must be created.
4. Create CIFS shares on the destination SVM by using the same share names that were used at the source. Clients can access the CIFS shares. However, all data is read-only until the volume is failed over.
5. Apply the proper ACLs to the CIFS shares at the destination.
6. Create appropriate NFS export policies for the destination SVM.

7 Assign the NFS export policies to the destination volumes. Although clients can access the NFS exports, all data is read-only until the volume is failed over.

## SAN Only Environments

**Procedure**

1 If the destination SVMs use portsets, they can be configured as required before failover.

2 Configure igroups on the destination SVM.

Typically, there are different application servers that connect to the recovered storage at the DR site. The initiators from these servers can be preconfigured into appropriate igroups in the destination SVM.

**Note**

Because many host operating systems do not support connecting to LUNs in read-only containers, map LUNs to igroups after failover (`snapmirror break`) has completed.

# Performing a Failover

With most of the configuration necessary for DR performed prior to a failover, the actual steps required to fail over during a DR scenario are greatly reduced. They are as follows.

## NAS Environment

**Procedure**

1 Perform a SnapMirror break operation to fail over each volume. In ONTAP, wildcards can be used to perform a SnapMirror operation on multiple volumes with one command. The following example performs failover for all volumes in the destination SVM called `vs5`. It can be restricted to certain volumes by using part of the volume name in the command.

```
cluster02::> snapmirror break -destination-path cluster02://vs5/*
```

2 If the volumes have been mounted in the namespace and CIFS shares and NFS export policies have been created and applied, clients then have read-write access to the NAS data.

3 Redirect clients to the recovered storage.

It is a common practice to have a DR system with a different name than the source system. In DR failover scenarios, it is typical to change DNS name resolution or use DNS aliases to redirect clients to the name of the recovered storage systems. This approach enables CIFS share access using the same UNC path name, and NFS clients can also access the expected path. Alternatively, the failed source storage system can be removed from Active Directory. The recovery storage system can then be removed and added again to Active Directory by using the same name as the source system. However, it can take time for this change to propagate through a large Active Directory environment.

## SAN Environment

**Procedure**

1. Perform a SnapMirror break operation to fail over each volume. Wildcards can be used to perform a SnapMirror operation on multiple volumes with one command. The following example performs failover for all volumes in the destination SVM called `vs5`. It can be restricted to certain volumes by using part of the volume name in the command.

```
cluster02::> snapmirror break -destination-path cluster02://vs5/*
```

2. Make the LUNs in the volume available to the SAN clients at the DR site by mapping the LUN into the appropriate igroup.
3. On the SAN client, perform a storage rescan to detect the connected LUN.

# Post Failover Volume Configuration

Snapshot copy policies and storage efficiency policies cannot be assigned to volumes in a DP state, so they must be assigned after failover.

**Procedure**

1. If using an ONTAP snapshot schedule, assign a Snapshot copy policy to the recovered volumes. In SAN environments, snapshots are typically scheduled in the client.
2. If using storage efficiency technology, assign a storage efficiency policy to the recovered volumes.

ETERNUS AX series All-Flash Arrays, ETERNUS AC series All-Flash Arrays, ETERNUS HX series Hybrid Arrays
SnapMirror Configuration and Best Practices Guide for ONTAP 9

P3AG-5642-06ENZ0

Date of issuance: June 2025
Issuance responsibility: Fsas Technologies Inc.



Fsas Technologies Inc.