



# SnapMirror Configuration and Best Practices Guide for ONTAP 9

Updated for ONTAP 9.13.1



## Abstract

This document describes information and best practices related to configuring Lenovo® SnapMirror® replication and has been updated based on changes introduced in Lenovo ONTAP® 9.13.1.

**Second edition (October 2023)**

**© Copyright Lenovo 2023.**

**LIMITED AND RESTRICTED RIGHTS NOTICE: If data or software is delivered pursuant to a General Services Administration (GSA) contract, use, reproduction, or disclosure is subject to restrictions set forth in Contract No. GS-35F-05925**

## TABLE OF CONTENTS

<b>Solution overview</b> .....	<b>6</b>
Purpose and intended audience .....	6
What's new for SnapMirror? .....	7
Lenovo SnapMirror overview .....	7
Use case summary .....	8
Unified architecture flexibility .....	9
<b>ONTAP networking basics</b> .....	<b>10</b>
Common SnapMirror networking terms .....	10
SnapMirror networking requirements .....	12
Intercluster networking requirements .....	12
Intercluster multipathing .....	13
How SnapMirror uses intercluster LIFs to replicate traffic .....	16
Firewall requirements .....	17
<b>Lenovo replication basics</b> .....	<b>17</b>
Licensing .....	17
Lenovo SnapMirror Asynchronous technology .....	17
SnapMirror Synchronous .....	21
SnapMirror for Lenovo cloud volume platforms .....	21
<b>SnapMirror configuration</b> .....	<b>21</b>
Cluster peering .....	21
SVM peering .....	23
SnapMirror data protection relationship .....	23
Protection policies .....	27
SnapMirror schedules .....	34
Create a SnapMirror relationship .....	35
Baseline transfer during initialization of SnapMirror relationship .....	36
Manual update to the SnapMirror relationship .....	37
<b>Converting between different data protection modes</b> .....	<b>38</b>
Convert a legacy DP SnapMirror relationship to an XDP SnapMirror relationship .....	38
Convert SnapMirror to unified replication .....	41
<b>SnapMirror and ONTAP feature interaction</b> .....	<b>47</b>
SnapMirror and Snapshot copies .....	47

SnapMirror and qtrees .....	48
SnapMirror and FlexGroup volumes .....	48
SnapMirror and FlexClone technologies .....	50
SnapMirror and Lenovo storage efficiency .....	50
SnapMirror and volume move .....	51
SnapMirror for disk shelf failure protection .....	51
SnapMirror and volume autosize .....	51
SnapMirror and NDMP .....	52
SnapMirror and FabricPool .....	52
<b>SnapMirror for consistency groups .....</b>	<b>52</b>
Overview of SnapMirror with CGs .....	53
Creating a SnapMirror CG relationship .....	53
Scalability of SnapMirror relationships for CGs .....	54
Managing SnapMirror manages protection of CGs .....	54
Converting the existing volume relationships to CG relationships .....	55
Changing the composition of a CG involved in a SnapMirror relationship .....	55
Interoperability with other ONTAP features .....	56
<b>SnapMirror for SVM DR .....</b>	<b>57</b>
SVM DR scalability .....	58
SVM DR advanced topics .....	59
SVM DR interoperability .....	60
<b>Performance .....</b>	<b>61</b>
Calculate SnapMirror throughput for performance .....	61
SnapMirror and network compression .....	62
SnapMirror throttling .....	64
How to change TCP receive buffer size .....	66
Concurrent replication operations .....	66
Network sizing requirements .....	66
<b>S3 SnapMirror .....</b>	<b>67</b>
<b>Interoperability .....</b>	<b>67</b>
<b>Troubleshooting tips .....</b>	<b>68</b>
Troubleshooting cluster peer relationships .....	68
Troubleshooting SVM peer relationships .....	68
Understanding SnapMirror relationship status .....	69

Troubleshooting SnapMirror relationships .....	69
<b>Best practices for DR configurations .....</b>	<b>71</b>
<b>Configuration and failover for DR .....</b>	<b>72</b>
Environment failover requirements and assumptions .....	72
Preparing the destination for failover .....	73
Performing a failover .....	75
Post failover volume configuration .....	75
<b>Where to find additional information .....</b>	<b>75</b>
<b>Version history .....</b>	<b>76</b>

#### LIST OF TABLES

Table 1) New features in SnapMirror asynchronous for individual volumes .....	7
Table 2) New features in SnapMirror SVM disaster recovery .....	7
Table 3) ONTAP networking terminology .....	10
Table 4) Shared versus dedicated interface groups .....	15
Table 5) Minimum RPO for SnapMirror replication of FlexVol volumes and FlexGroup volumes .....	25
Table 6) SnapMirror policy types .....	27
Table 7) SVM migration limitation summary .....	49
Table 8) SnapMirror interoperability with other ONTAP features .....	56
Table 9) Differences between SVM DR and SnapMirror .....	57
Table 10) SVM DR scalability .....	58
Table 11) TCP receive buffer windows .....	66

#### LIST OF FIGURES

Figure 1) Lenovo SnapMirror replication overview .....	8
Figure 2) Unified architecture flexibility .....	10
Figure 3) Intercluster LIFs and relationship to other ONTAP networks .....	12
Figure 4) Failover group used in failover mode .....	14
Figure 5) Failover group used in multiplex mode .....	15
Figure 6) TCP stream distribution examples .....	16
Figure 7) SnapMirror custom protection policy create .....	20
Figure 8) Add custom SnapMirror protection policy .....	20
Figure 9) SnapMirror fan-out and fan-in .....	25
Figure 10) SnapMirror cascade .....	26
Figure 11) SnapMirror Asynchronous policy definition .....	28

Figure 12) DailyBackup asynchronous SnapMirror policy definition. ....	29
Figure 13) DPDefault asynchronous SnapMirror policy definition. ....	30
Figure 14) MirrorAllSnapshots asynchronous SnapMirror policy definition. ....	31
Figure 15) MirrorLatest asynchronous SnapMirror policy definition. ....	31
Figure 16) MirrorAndVault asynchronous SnapMirror policy definition. ....	32
Figure 17) Unified7year asynchronous SnapMirror policy definition. ....	33
Figure 18) XDPDefault policy definition. ....	33
Figure 19) SnapMirror schedules can be listed and created in Storage Manager. ....	34
Figure 20) SnapMirror schedules can be listed and created using CLI. ....	34
Figure 21) Start a SnapMirror relationship update. ....	37
Figure 22) Relationship update dialog. ....	37
Figure 25) FlexGroup volumes used in SnapMirror cascade and fan-out configurations. ....	48
Figure 26) SnapMirror behavior during FlexGroup rebalance operation. ....	49
Figure 27) Creating a FlexClone volume at a SnapMirror destination. ....	50
Figure 23: Comparing ONTAP CGs with SnapMirror CG support. ....	53
Figure 24: SnapMirror for consistency groups. ....	53
Figure 28) SVM DR fan-out example. ....	59
Figure 29) SVM DR cascade example. ....	60
Figure 30) SVM DR with MetroCluster. ....	60
Figure 31) SnapMirror network compression functional diagram. ....	63
Figure 32) ONTAP S3 SnapMirror overview. ....	67
Figure 33) Volume layout for DR. ....	73

## 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, disk, 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. Lenovo 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 Lenovo 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 of 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 Lenovo 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 ThinkSystem Storage Manager enabling storage administrators to perform operations with simplified workflows and wizard-guided walkthroughs. Storage Manager also supports monitoring and managing all SnapMirror replication relationships in one place.
- **Secure.** From ONTAP 9.6 onwards, SnapMirror relationships can be encrypted natively end-to-end using TLS 1.2 AES 256-bit encryption. For details about ONTAP security, see [Security Hardening Guide](#).
- **Cloud enabled.** SnapMirror supports volume and storage virtual machine (SVM) replication to Lenovo solutions in the cloud such as 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 Lenovo features such as Lenovo Snapshot™ copy technology, Lenovo FlexVol® or FlexGroup® volumes, and Lenovo FlexClone technology
- General knowledge of DR and data replication solutions
- Familiarity with the [ONTAP Data Protection Guide](#).

## 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, and Table 2 lists the new features for SVM Disaster Recovery (SVM DR) for ONTAP 9.13.1.

**Table 1) New features in SnapMirror asynchronous for individual volumes**

Functional area	ONTAP 9.13.1 changes
Consistency group support	<p>Lenovo SnapMirror relationships can be created by using a consistency group (CG) as the source. Consistency groups provide a method to create data-consistent Snapshot copies of multiple volumes to support applications that require it, such as databases.</p> <p>CG replication management can be performed using ONTAP CLI, Storage Manager, or REST API.</p> <p>See the details of SnapMirror with consistency groups later in this report.</p>
Improved restore process	SnapMirror will validate all restore steps before starting the restore process and report any issues prior to execution.
Improved DR rehearsal recovery times	To support limited DR rehearsal time windows, SnapMirror default to using quick resync when reverting volumes back to production cluster (flip resync). Quick resync bypasses a complete data warehouse rebuild, unless there are Flexclone® volumes associated with the protected volumes.
Array interoperability	SnapMirror asynchronous volume replication has been qualified between all current AFA (DM-series and DG-series) and Hybrid systems.
Quick resync support for FlexGroup volumes	SnapMirror will now perform quick resync operations on FlexGroup volumes to support faster return to operational configuration.

**Table 2) New features in SnapMirror SVM disaster recovery.**

Functional area	ONTAP 9.13.1 changes
Fan-out	SnapMirror SVM DR now supports fan-out for SVMs hosting FlexGroup volumes.
Array interoperability	SVM DR has been qualified between all current AFA (DM-series and DG-series) and Hybrid systems.
Improved DR rehearsal support	SVM DR now supports data warehouse rebuild indicator

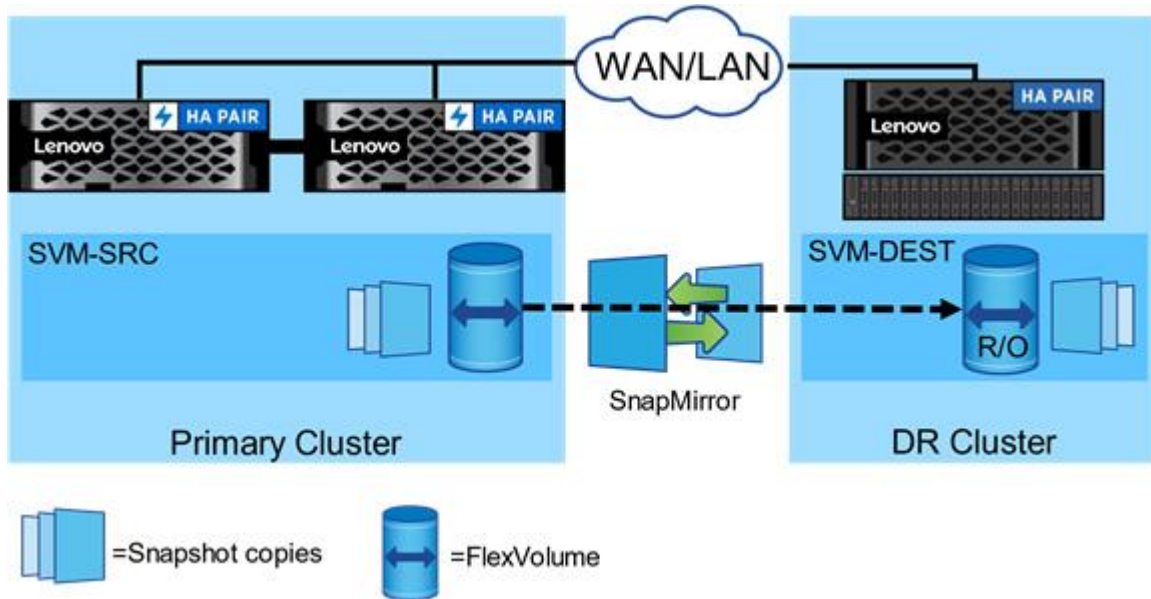
## Lenovo SnapMirror overview

Lenovo 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 either data volumes (FlexVol volumes or FlexGroup volumes) or consistency groups 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 Lenovo 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 again. 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 Lenovo SnapMirror replication, see Figure 1.

**Figure 1) Lenovo SnapMirror replication overview.**



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

Lenovo-integrated data protection can be used to create a quickly accessible on-disk history of application-consistent Snapshot copies that eliminate the concept of a traditional backup window. Lenovo SnapMirror then replicates these Snapshot copies 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 Lenovo 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**

Lenovo 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. Lenovo 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 Lenovo ONTAP arrays to software-defined storage within VMware vSphere infrastructures. These deployments can be managed by using CLI, REST APIs, or the web-based Storage 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, Ansible, and VMware vRealize.

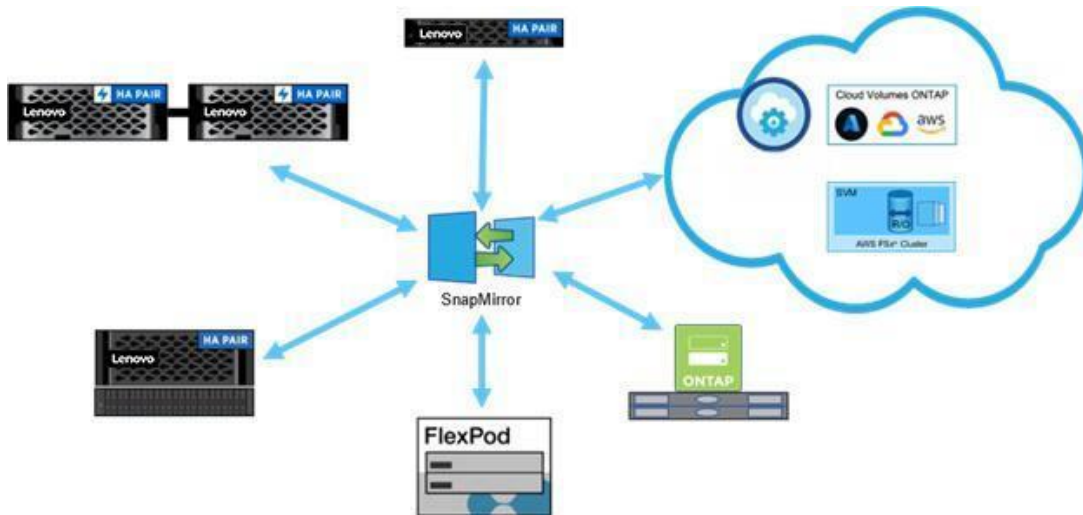
## Hybrid cloud deployments

SnapMirror provides support for data replication between ONTAP on-premises and ONTAP public-cloud implementations such as Lenovo Cloud Volumes ONTAP deployments on major cloud provider environments.

## 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. These ONTAP solutions provide access to SnapMirror functionality to support primary and DR cloud-based deployments and can use Storage Manager, CLI, or REST APIs for management.

Figure 2) Unified architecture flexibility.



## ONTAP networking basics

For a basic ONTAP networking overview, see [Network management](#). 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. Two nodes can be integrated into the same physical chassis as an HA pair. Typically, ONTAP storage clusters consist of 2-nodes (HA pair) or more.
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

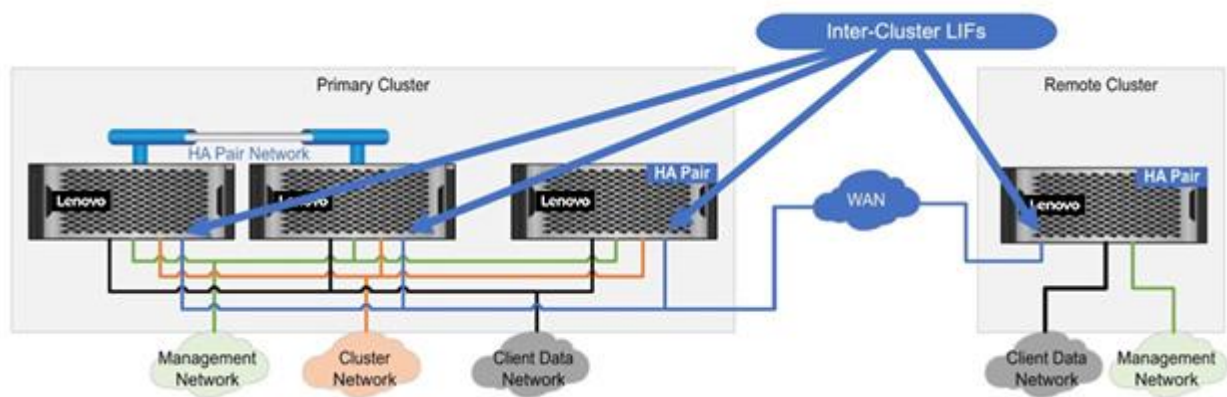
Term	Definition
	event of a node failure.
Cluster	One or more HA pairs that are interconnected and managed as a single storage solution. ONTAP clusters can consist of up to 6 SAN or 12 NAS HA pairs.
IPspace	An IPspace defines a distinct IP address space in which ONTAP arrays can participate. IPspaces 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. 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: <ul style="list-style-type: none"> <li>• Interface group</li> <li>• VLAN</li> </ul>
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. 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.

Term	Definition
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 Snapshot copies 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 [Network management](#) and the [Cluster Management using ThinkSystem Storage Manager](#).

## 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 distribute 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, 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 Lenovo cluster creates a failover group for each broadcast domain. Custom failover groups can be defined to optimize the failover behavior of a LIF.

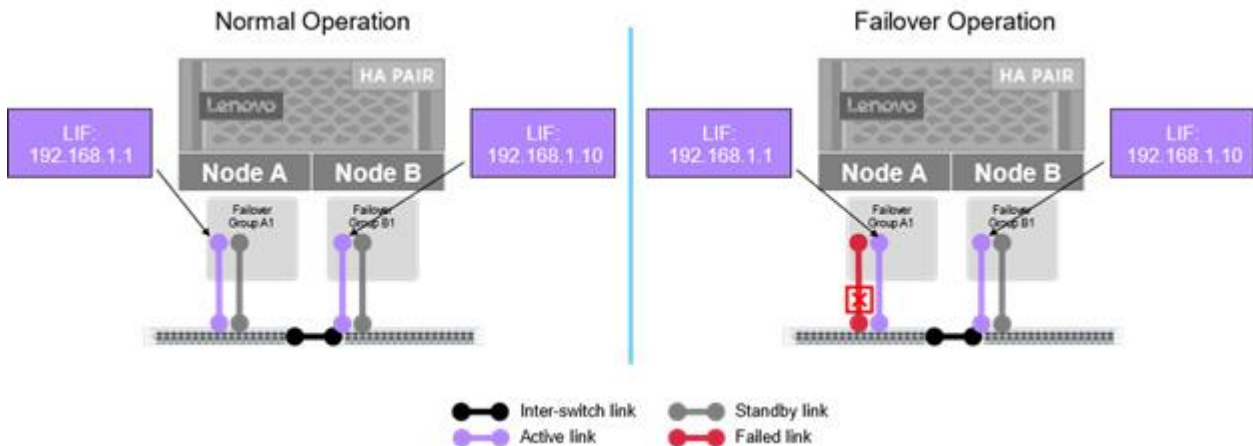
Intercluster LIFs can be assigned to failover groups that span cluster nodes, but intercluster LIFs will only failover to ports within the local node of the home port for that LIF. Intercluster LIFs are defined with the `-role intercluster` or `-service-policy default-intercluster` parameters to the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

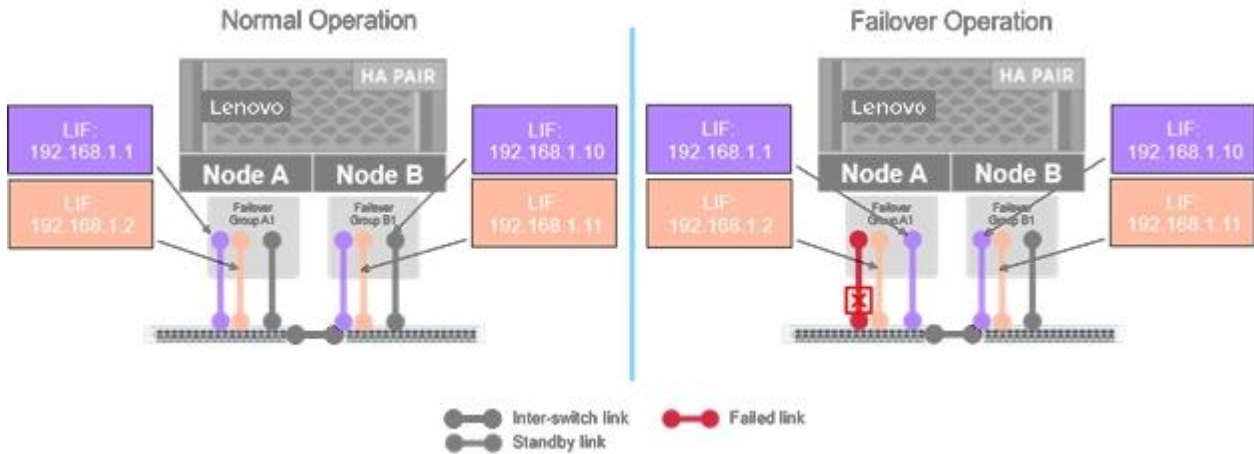
Once the failover group is created, LIFs are assigned to individual ports in the failover group. The port a LIF is initially assigned to is considered that LIF's home port. All other ports in the failover group are considered standby ports for that LIF. If a single intercluster LIF is assigned to a failover 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. Lenovo recommends 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
<ul style="list-style-type: none"> <li>• Replication performance is not critical.</li> <li>• Higher bandwidth (25–100GbE) ports are available and SnapMirror replication workload is not enough to saturate links.</li> <li>• Node interface available bandwidth is not the bottleneck...the WAN link is.</li> <li>• Replication requirements support a high RPO (for example, I only need to replicate once a day).</li> <li>• Data change rate on the source volume is low.</li> <li>• Port availability on each node is low.</li> </ul>	<ul style="list-style-type: none"> <li>• DR is a primary goal for SnapMirror.</li> <li>• Lower bandwidth node ports that might easily get saturated with SnapMirror replication workload.</li> <li>• Replication requirements need to support a very small RPO (dictating the need for different MTU).</li> <li>• Data change rate on the source volume is very high (dictating the need for different maximum MTU).</li> <li>• Port availability on each node supports having a dedicated failover group.</li> </ul>



## Best practice recommendations

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

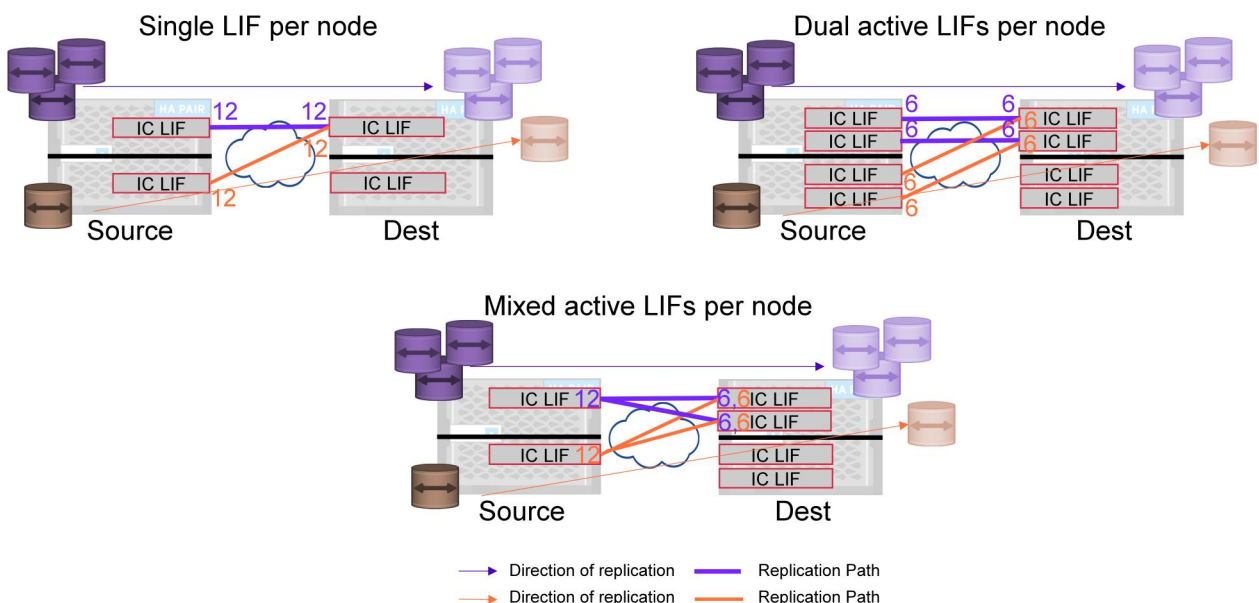
ONTAP provides a centralized network session manager to provide optimized network traffic management for various functions involving intercluster communications such as SnapMirror. For SnapMirror, this cluster session manager provides up to 12 TCP send streams and 12 TCP receive streams for each source-destination 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 Storage Manager or REST APIs

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

## Lenovo replication basics

### Licensing

A SnapMirror license is required on each cluster that will participate in SnapMirror replication. This license is included in the Unified Complete bundle. 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 9.12.1P1 for Lenovo AFA DG-Series controllers and ONTAP 9.13.1 for all other Lenovo Hybrid, and AFA DM-Series controllers, SnapMirror is included as part of Unified Complete or ONTAP One bundle license sold with every Lenovo ONTAP controller.

**Note:** The license must be present on both source and destination.

### Lenovo 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 Snapshot copies. SnapMirror performs the following operations:

1. A Snapshot copy of the data on the source is created. This Snapshot copy will have a SnapMirror label of `sm_created`.
2. The Snapshot copy 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 copy.
3. Additional Snapshot copies are created according to the SnapMirror schedule. Each Snapshot copy contains only the changes made to the source volumes since the last replicated Snapshot copy and is replicated to the destination volumes.

When a SnapMirror data protection relationship is established (`snapmirror initialize`), the destination volumes are identical replicas of the source, including snapshots, volume settings, and ONTAP space efficiency features that is accessible as a read-only volume. Breaking the SnapMirror relationship (`snapmirror break`) makes the destination volumes writable and is typically used to perform a failover when SnapMirror is used to synchronize data to a DR environment. 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. Although both engines operate at the volume level, they have different characteristics.

- **Block Replication Engine (BRE).** BRE replicates the on-disk 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 Snapshot copies 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 (VVB) read and write operations. The 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.
  - Note:** As of 9.11.1, BRE 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).
  - Note:** 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 relationships are converted to LRSE (XDP). For more information, see [Convert an existing DP-type relationship to XDP](#).
- **Logical replication with storage efficiency (LRSE).** LRSE uses block-level metadata and knowledge of the file system to determine differences between Snapshot copies 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-disk physical layout relative to the source. The SnapMirror relationship is created with `-type XDP` using the SnapMirror policy type `async-mirror, vault or mirror-vault`.

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 Snapshot copies where the destination can support a greater number of Snapshot copies 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 nondisruptively upgraded at any time while reducing overhead and managing complex topologies (fan-in, fan-out, and cascade).

The performance characteristics are also like those of the original block replication engine because the replication engine only transfers the difference between two Snapshot copies from the primary to the secondary. This incremental-only transfer leads to savings in terms of storage and network bandwidth.

Starting with ONTAP 9.4, SnapMirror XDP mode replaces SnapMirror data protection (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 Lenovo SnapCenter® to replicate application consistent snapshots, such as those used for enterprise database applications. Snapshot copies 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 copy, SnapCenter can then trigger a SnapMirror replication of these application consistent Snapshot copies to the secondary storage system.

## Unified data protection

Starting with ONTAP 9.4, SnapMirror supports the XDP relationship type for both the volume mirror use cases traditionally provided by SnapMirror DP relationship type, and disk-to-disk backup use cases traditionally referred to as SnapVault in legacy documentation. 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 providers. The relationship is created with type XDP, policy type `mirror-vault`, and the predefined policy `Asynchronous`. The policy can always be modified to include custom rules for backing up specific Snapshot copies. In addition, this functionality reduces the number of secondary Snapshot copies needed on the destination.

The major benefits for SnapMirror unified replication are as follows:

- Only one baseline copy of a volume is needed to the secondary storage.
- Less network traffic is required between the primary and secondary (a single baseline plus fewer Snapshot copies 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 Snapshot copies.
- XDP 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 Storage Manager for ONTAP 9.8 and later, SnapMirror Asynchronous uses the Asynchronous 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

Weigh the benefit of maintaining a full mirror against the advantages that unified replication offers by reducing the amount of secondary storage, limiting the number of baseline transfers, and decreasing network traffic. The key factor in determining the appropriateness of unified replication is the rate of change in the active file system. A traditional mirror might be better suited to a volume holding hourly Snapshot copies of database transaction logs, for example.

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.

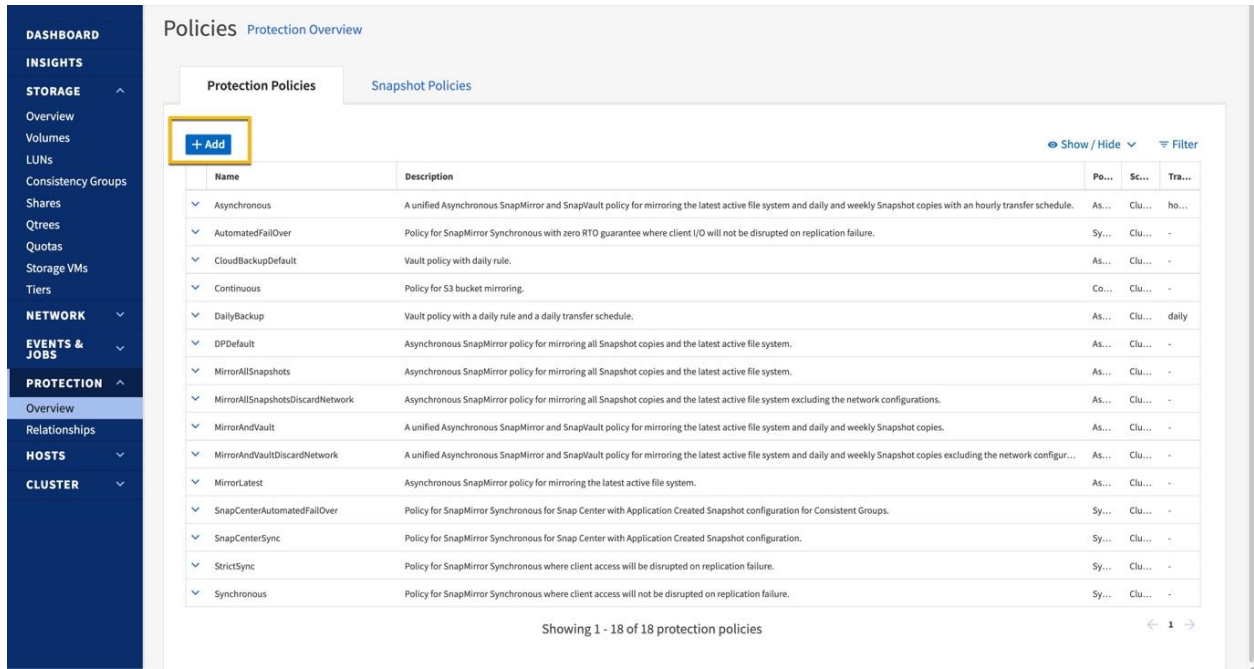
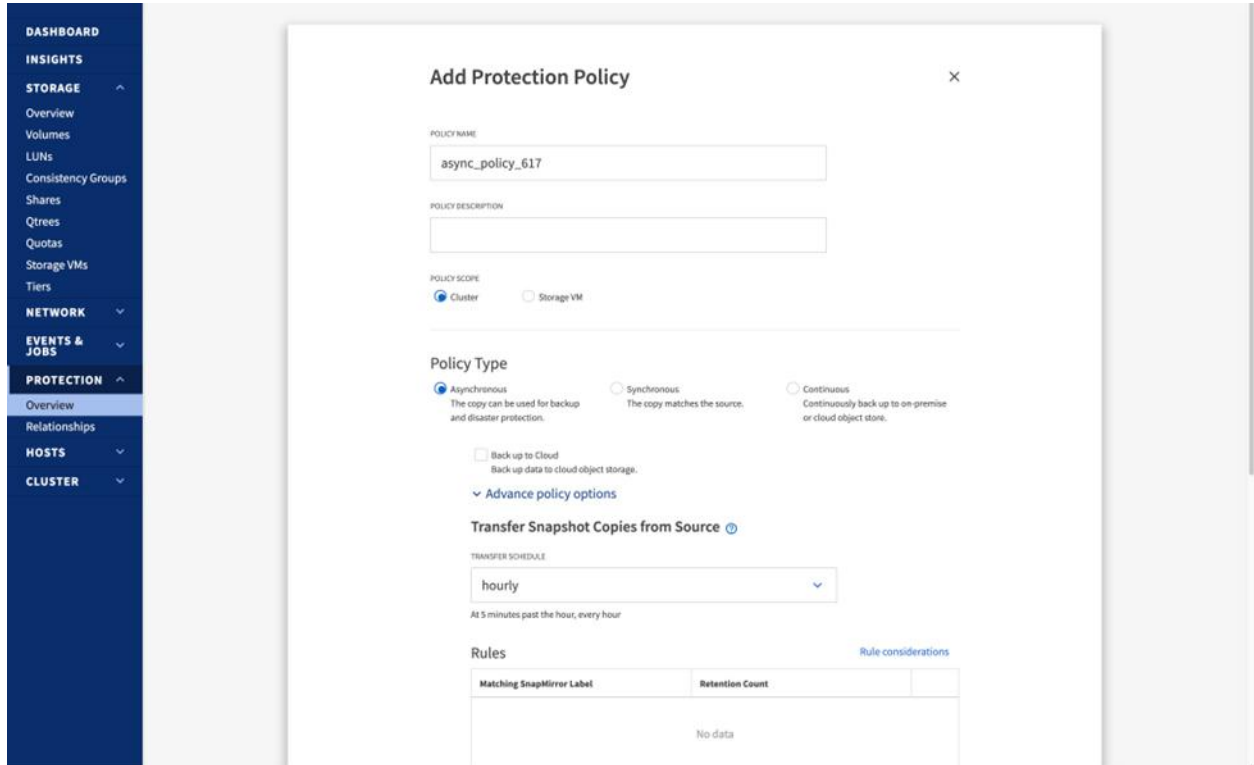


Figure 8) Add custom SnapMirror protection policy.



## 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 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 is not suitable for a MetroCluster environment.

**Note:** Starting with ONTAP 9.4, load-sharing mirrors have been deprecated for data volumes and are only supported for SVM root volumes.

**Note:** 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 or iSCSI). However, NFSv4 and load-sharing mirrors can coexist in the same environment.

### Best practice

Create a load-sharing mirror of an SVM root volume only on a node of a second HA-pair of the cluster.

For additional information concerning load-sharing mirrors, see [Managing SnapMirror Root Volume Replication](#).

## SnapMirror Synchronous

SnapMirror Synchronous (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.

## SnapMirror for Lenovo 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, Lenovo offers 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.

### Lenovo 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.

## SnapMirror configuration

### Cluster peering

The cluster peering feature of ONTAP allows administrators of independent clusters to establish a peer relationship between them. They can use an intercluster network that defines the network connections that enable clusters to securely exchange application data, configuration information, and coordinate operations. The intercluster LIF role is the interface to handle intercluster traffic. Administrators must define a range of addresses on their networks for use in cross-cluster communications, arrange routing

for these addresses (for example, by routing group), and assign the LIFs in this role to their intercluster or data ports. After the intercluster LIFs have been created and the intercluster network has been configured, cluster peers can be created to enable replication to or from another cluster using SnapMirror. Peer relationships can be configured for a single cluster with up to eight remote clusters.

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.

**Note:** For more information about cluster peering requirements, see [Cluster Management using ThinkSystem Storage Manager](#).

**Note:** SnapMirror does not support network address translation (NAT).

Establishing cluster peering is a one-time operation that must be performed by the cluster administrators. A cluster peer relationship is really nothing more than two corresponding collections of configuration objects in different clusters. Therefore, the relationship contained within one cluster is only half of the relationship. For a relationship to be considered complete and for it to function correctly, each cluster must share some of its configuration with its peer. A cluster peer relationship exists exactly between two clusters.

A peer relationship can be created in several ways:

- In one method, a peer relationship is created by a cluster administrator who has security credentials (a cluster admin login and password) for the remote cluster.
- Another method allows two administrators who do not want to exchange cluster admin passwords to peer their clusters. In this method, each administrator enters the `cluster peer create` command specifying the intercluster IP addresses of the other cluster.
- Beginning in ONTAP 9.4, the `-generate-passphrase` feature can be used to create a peer relationship with a cluster whose intercluster LIF IP addresses are not known in advance. This eliminates the need for the initiating cluster to authenticate itself with the remote cluster.

A cluster can be in a peer relationship with up to 255 clusters, allowing multiple clusters to replicate amongst themselves.

Starting with ONTAP 9.6, cluster peering uses encrypted communication, which means any SnapMirror relationship that is created uses an additional layer of security through TLS encryption.

### Best practice

- The name and IP address of the source system must be in the `hosts` file of the destination system and vice versa, or they must be resolvable through network-based DNS servers.
- Use at least one intercluster IP address from each node in the remote cluster, so that the peer relationship remains stable in the event of a node failure. As intercluster LIFs become available or unavailable, the list of active IP addresses can change. The discovery of active IP addresses is automatic in certain events, such as when a node reboots. The `-peer-addr` option requires the provision of only one remote cluster address. However, if the node hosting that address is down and becomes unavailable, then the cluster peer relationship might not be rediscovered. Therefore

## 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.

### Best practice

Name an SVM with a unique fully qualified domain name (FQDN): for example, `dataVserver.HQ` or `mirrorVserver.Offsite`. SVM peering requires unique SVM names, and using the FQDN naming style makes it much easier establish uniqueness.

For more information about cluster peering requirements, see [Cluster Management using ThinkSystem Storage Manager](#).

## 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. The creation of a SnapMirror relationship does not depend on SVM host name to IP address resolution. However, the cluster names are resolved through the cluster peer relationship, and the SVM names are internally resolved through the clusters. The host names of the source and destination SVM and cluster are used to create SnapMirror relationships in ONTAP. It is not necessary to use the IP address of an intercluster LIF.

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

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 has 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, because intercluster relationships are to a different cluster and therefore to a different SVM, which is a different NAS namespace. 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, ThinkSystem Storage Manager, REST API, and Unified Manager.
- 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.

Complete the following requirements before creating an intercluster SnapMirror relationship:



- Configure the source and destination nodes for intercluster networking.
- Configure the source and destination clusters in a cluster peer relationship.
- Both the source and destination SVM can have different language types, but the source and destination volumes must have the same language type.
- Configure the source and destination SVM in an SVM peer relationship.
- The destination aggregate must have adequate space to host the replicated volumes and any vault Snapshot copies retained per the configured protection policy.
- Both clusters must be configured and set up appropriately to meet the requirements of the network and security environments for user access, authentication, and client access.
- If the SnapMirror source is a consistency group, then the destination cluster must be running an ONTAP version of 9.13.1 or later.
- Create destination volumes with `-type DP` parameter, with adequate space to host the replicated volumes and any vault Snapshot copies retained per the configured protection policy.
- Assign a schedule to the SnapMirror relationship in the destination cluster to perform periodic updates. If any of the existing schedules are not adequate, a new schedule entry **can** be created.
- Assign a protection policy (built-in or custom) to the SnapMirror relationship.

SnapMirror relationship can be created in one of two ways:

- **Intercluster.** Replication between volumes in two different SVMs in different clusters operating in ONTAP. They are primarily used for providing DR to another site or location.
- **Intracluster.** Replication between two volumes in different SVMs in the same cluster or between two volumes in the same SVM. They are primarily used for maintaining a local backup copy.

### 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 Snapshot copies that SnapMirror creates in the source volume before copying the data to the destination. The most recent SnapMirror Snapshot copy is referred to as the newest common Snapshot copy (NCS). Incremental changes to the destination depend on the NCS. If SnapMirror cannot find the required Snapshot copy 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 Snapshot copies, 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 event 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.**

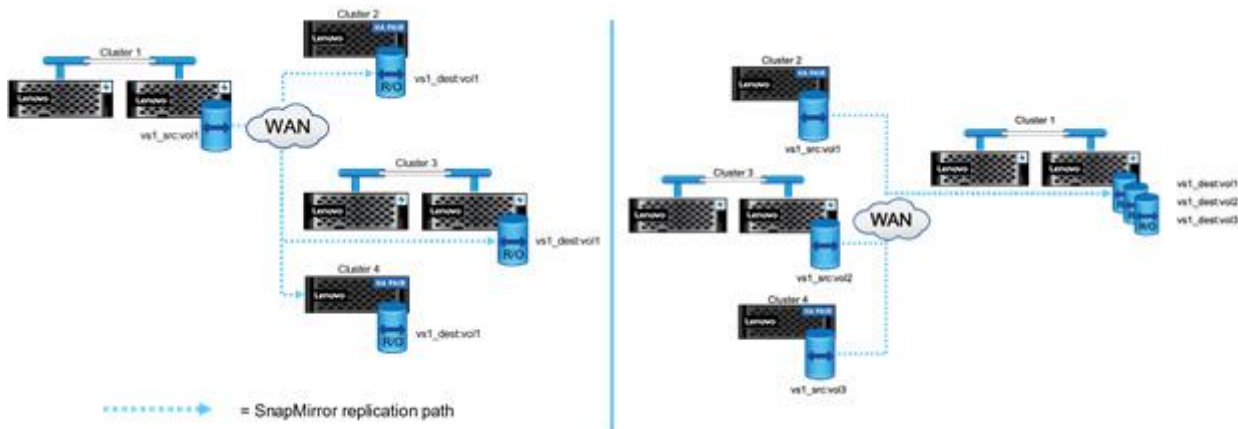
SnapMirror replication source	Minimum supported RPO
FlexVol volumes	5 minutes
FlexGroup volumes	30 minutes
Consistency groups	30 minutes

### 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 sources can be replicated to a single destination (fan-in), as shown in Figure 9.

**Note:** As of ONTAP 9.12.1, SnapMirror supports fan out to a maximum of 16 destination volumes from a single source volume depending on type of source and destination controllers. See [Lenovo Press](#) for fan-out limits for specific controller models.

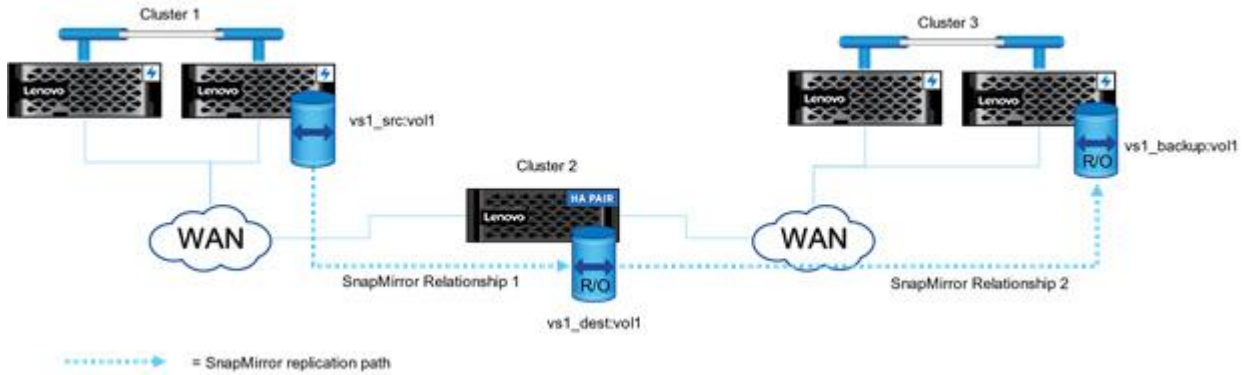
**Figure 9) SnapMirror fan-out and fan-in.**



### Cascade relationship

SnapMirror can replicate data from a SnapMirror destination to another 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. An example cascade configuration with two hops is shown in Figure 10.

Figure 10) 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.

Snapshot copies behave in the following ways:

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

SnapMirror supports a different relationship for each leg of the cascade. For any relationship in the cascade chain, that relationship can be a mirror or vault. Therefore, in long-chain cascades, the following combinations of data protection relationships can be used between any three clusters in the chain:

<b>Mirror</b>	<b>Mirror</b>	One cluster has a mirror relationship with a second cluster and that second cluster has a mirror relationship with a third cluster.
<b>Mirror</b>	<b>Vault</b>	One cluster has a mirror relationship with a second cluster and that second cluster has a vault relationship with a third cluster.
<b>Vault</b>	<b>Vault</b>	One cluster has a vault relationship with a second cluster and that second cluster has a vault relationship with a third cluster.

**Note:** For any cascade configuration, only the initial data protection relationship can be synchronous (`sync` or `strictsync`). All subsequent mirror relationships in the cascade must be asynchronous.

### Best practice

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 copy does not exist on the source and destination volumes or consistency groups. Use the [snapmirror snapshot-owner create](#) command to preserve a labeled Snapshot copy on the secondary in a mirror-vault deployment. Doing so provides a common Snapshot copy for the update of the vault relationship.

Some sample use cases for cascade SnapMirror relationships include:

- A backup admin can offload certain Snapshot copies to tertiary storage and thus retain more than the actual number of Snapshot copies supported by a single volume or consistency group (currently 1023).

- A backup admin can tier multiple backup copies. More frequent Snapshot copies (daily and weekly) can be retained on secondary storage (B), whereas monthly and yearly Snapshot copies can be retained on tertiary storage (C).
- Data can be moved from SSDs to SATA disk drives.
- Data can be distributed globally across locations without any hindrance to ongoing operations. It can be further moved to the cloud as archival storage.

**Note:** Creation of SnapMirror cascade relationships is not supported by ThinkSystem Storage 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.

## Protection policies

ONTAP relies on policies to dictate when to create Snapshot copies 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 copy created by SnapMirror at initialization. At each update, SnapMirror creates another Snapshot copy of the source. It then transfers the changes from this Snapshot copy and the previously transferred Snapshot copy (mirror) along with any new Snapshot copies that have labels matching the labels defined in the Snapshot copy rules defined on the SnapMirror policy if the policy is a vault policy. ONTAP comes with several predefined protection policies.

## Policy types

Each SnapMirror protection policy (standard or custom) is one of several different policy types. The policy types are described in Table 6.

**Table 6) SnapMirror policy types.**

Policy type	Definition
Async-mirror	<p>The async-mirror policy type is used by SnapMirror to transfer Snapshot copies of two types:</p> <ul style="list-style-type: none"> <li>• Those source Snapshot copies created by the SnapMirror engine (rule label = sm_created)</li> <li>• All source Snapshot copies created on the volume from other Snapshot copy policies or manually created on the source volume. (rule label = all_source_snapshots)</li> </ul> <p><b>Note:</b> No other rules can be applied to a protection policy of type async-mirror.</p>
Vault	<p>The vault policy type is used by SnapMirror to copy only source volume Snapshot copies that match the labels provided with each rule.</p> <p>The Vault policy type replaces SnapVault functionality.</p> <p>This policy does not replicate Snapshot copies created by the SnapMirror relationship (label = sm_created)</p>
Mirror-vault	<p>The mirror-vault policy type is used by SnapMirror to transfer Snapshot copies created by the SnapMirror replication engine (snapshot label =</p>

Policy type	Definition
	<p><code>sm_created</code>) and any desired source Snapshot copies that match the Snapshot copy label defined in each rule.</p> <p>The mirror-vault protection policy can have multiple rules defined to match the desired data protection requirements.</p>
Sync-mirror	The sync-mirror policy is the default policy for SnapMirror Synchronous (SM-S). This policy supports the source and destination volumes being out of sync for short periods of time. In such conditions, if the write acknowledgement is not received from the destination data protection cluster, a write acknowledgement is still forwarded to the application.
Strict-sync-mirror	The strict-sync-mirror policy type is a policy for SM-S. This policy requires all write operations to be acknowledged by both clusters in the SnapMirror relationship before a write acknowledgement is sent to the application. If the write acknowledgement is not received from the data protection cluster, a write-fail is sent to the application and the volume is considered in a failed mode.
Continuous	The continuous policy type is used for S3 SnapMirror relationships. S3 SnapMirror is used to replicate ONTAP S3 buckets to another cluster running ONTAP S3, or cloud-based S3 buckets.
Automated-failover	The automated-failover policy type is used by SnapMirror-Business Continuity and is not designed for use in custom data protection policies for asynchronous data protection.

## 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 Snapshot copies with the daily label and retaining 52 Snapshot copies 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 11).

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 copy.
  - `daily` keeps seven daily Snapshot copies.
  - `weekly` and 52 weekly Snapshot copies.

**Figure 11) 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 Snapshot copies that have a label = daily and will retain the last seven Snapshot copies on the data protection volume. The Create Snapshot field is set to False, which indicates that this policy does not create any SnapMirror-related Snapshot copies 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 Storage Manager, these policies are only shown if the Show Legacy Policies option is selected when creating or editing a protection relationship (Figure 12).

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 copy when an update is triggered.
- There is one rule:
  - Keep seven daily Snapshot copies.

**Figure 12) 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 Snapshot copies 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 copy and then replicates the difference between the new SnapMirror Snapshot copy and the previous one and all the other Snapshot copies. If the relationship is being initialized, then a Snapshot copy is taken and everything before it is replicated. After the update is complete, the older Snapshot copy is deleted, leaving just one common SnapMirror Snapshot copy in place (Figure 13).

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 copy.
  - `all_source_snapshots` keep one copy of each unique Snapshot copy from the source volume.

**Figure 13) 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 Snapshot copies and the latest active file system from the primary to the secondary. This policy is similar to DPDefault (Figure 14).

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 copy.

- `all_source_snapshots` keep one copy of each unique Snapshot copy from the source volume.

**Figure 14) 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 copy 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 copy in place (Figure 15).

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 copy.

**Figure 15) 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 Snapshot copies. 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 16).

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 copy.
  - `daily` keeps seven daily Snapshot copies .
  - `weekly` keeps 52 weekly Snapshot copies.

**Figure 16) 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 Snapshot copies and retain them for seven years (Figure 17).

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 copy.
  - `daily` keeps seven daily Snapshot copies.

- weekly keeps 52 weekly Snapshot copies.
- monthly keeps 84 monthly (7 years) Snapshot copies.

**Figure 17) 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 Snapshot copies with labels of Daily and Weekly. This is a legacy vault-only policy (Figure 18).

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 copy when an update is triggered.
- There are two rules:
  - `daily` keeps seven daily Snapshot copies.
  - `weekly` keeps 52 weekly Snapshot copies

**Figure 18) 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:	Keep	Preserve	Warn	Schedule	Prefix
SnapMirror Label					
-----	-----	-----	-----	-----	-----
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 19 shows the pre-defined job schedules in ThinkSystem Storage Manager.

**Figure 19) SnapMirror schedules can be listed and created in Storage Manager.**

The screenshot shows the 'Schedules' page in the Storage Manager interface. The page has a left-hand navigation menu with categories like Overview, Applications, Volumes, LUNs, NVMe Namespaces, Shares, Buckets, Qtrees, Quotas, Storage VMs, Tiers, NETWORK, EVENTS & JOBS, PROTECTION, HOSTS, and CLUSTER. The 'PROTECTION' section is expanded, showing 'Overview' selected. The main content area displays a table of schedules with columns for Name, Type, and Schedule. A '+ Add' button is visible at the top left of the table area, and 'Show / Hide' and 'Filter' options are at the top right.

Name	Type	Schedule
5min	Time-based	At 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, and 55 minutes past the hour, every hour
8hour	Time-based	At 02:15 AM, 10:15 AM and 06:15 PM, every day
Application Templates ASUP Dump	Interval-based	Every 1 day
Auto Balance Aggregate Scheduler	Interval-based	Every 1 hour
Balanced Placement Model Cache Update	Interval-based	Every 7 minutes, 30 seconds
daily	Time-based	At 12:10 AM, every day
hourly	Time-based	At 5 minutes past the hour, every hour
monthly	Time-based	At 12:20 AM, on day 1 of the month, every day of the week
RepositoryBalanceMonitorJobSchedule	Interval-based	Every 10 minutes
weekly	Time-based	At 12:15 AM, only on Sunday, every month

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 Storage 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 20.

**Figure 20) 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
```

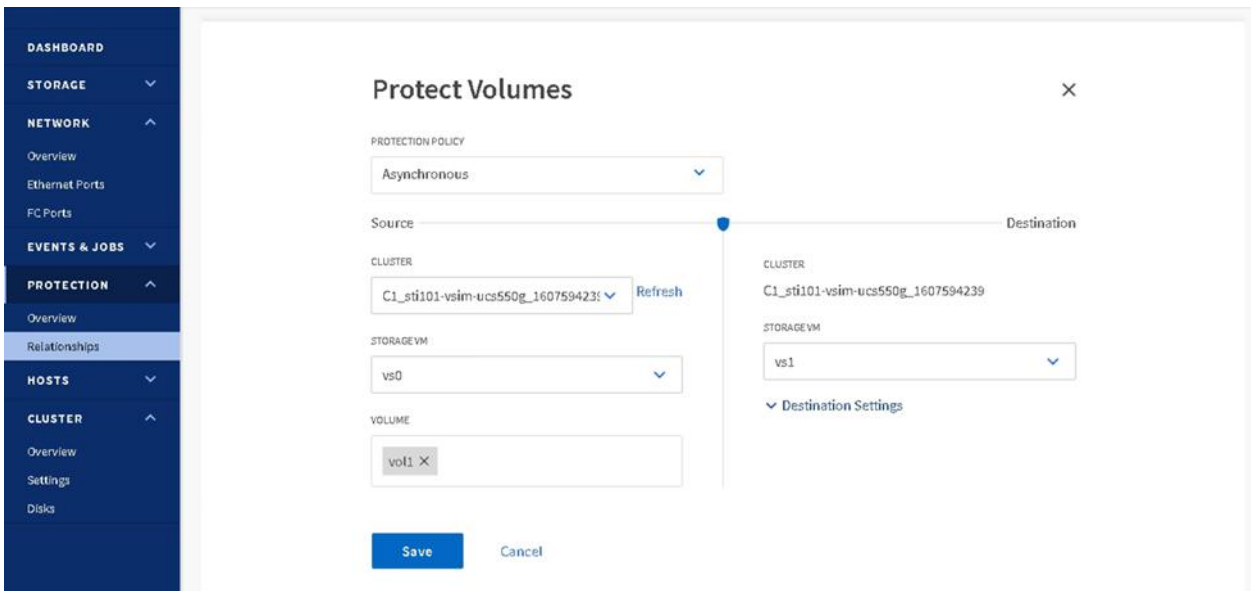
The schedule can then be applied to a SnapMirror relationship at the time of creation using the `-schedule` option or added to an existing relationship using the `snapmirror modify` command and the `-schedule` option. In this example, the `Hourly_SnapMirror` schedule is applied to an existing relationship.

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

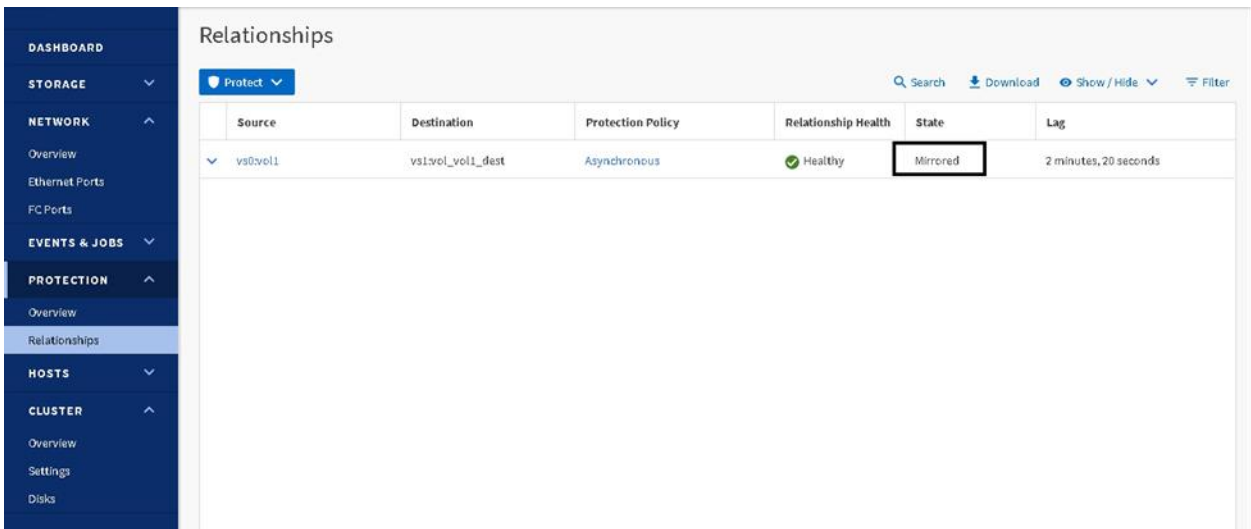
## Create a SnapMirror relationship

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

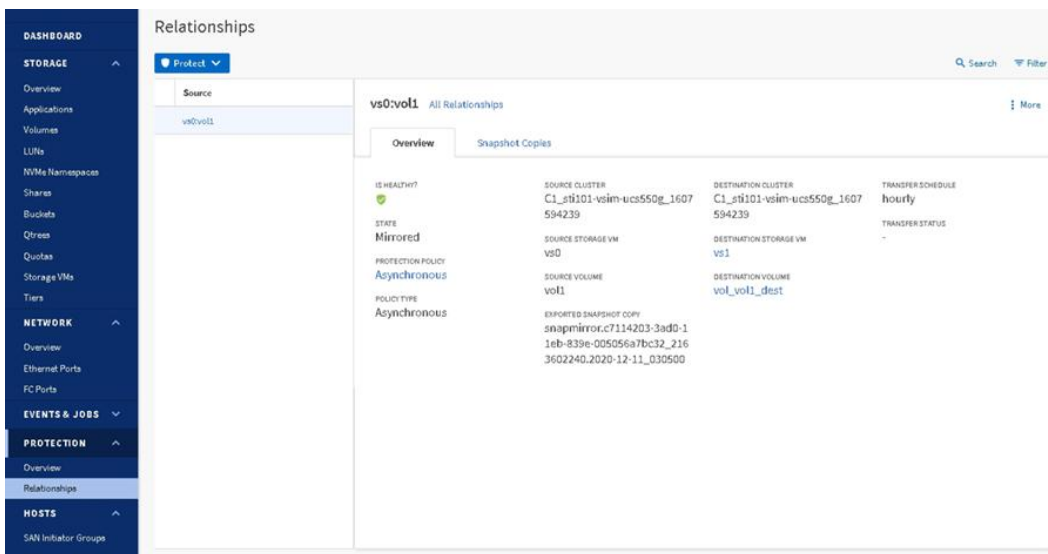
1. Click Relationships.
2. Select the volume or consistency group on which to create a SnapMirror relationship, and then click Save.



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 initialize a relationship:

1. Make a Snapshot copy of the source.
2. Transfer the Snapshot copy to the destination.
3. Depending on the SnapMirror policy attached to the relationship, it also transfers other Snapshot copies 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 Snapshot copies 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 copy or from a specific Snapshot copy to prevent data loss due to an upcoming power outage, scheduled maintenance, or data migration (Figure 21 and Figure 22).

Figure 21) Start a SnapMirror relationship update.

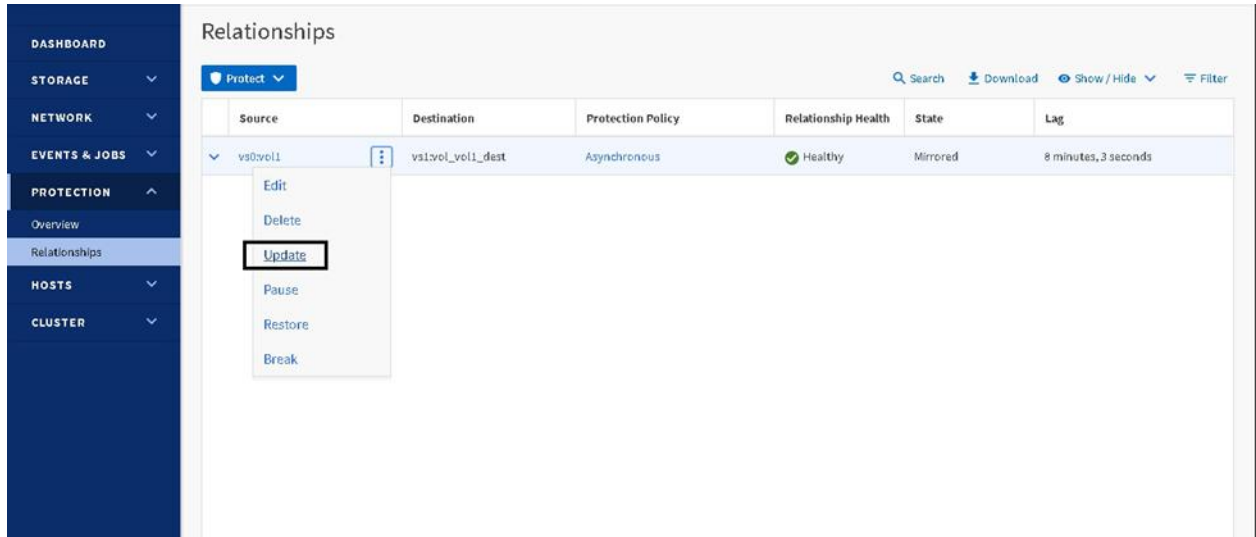
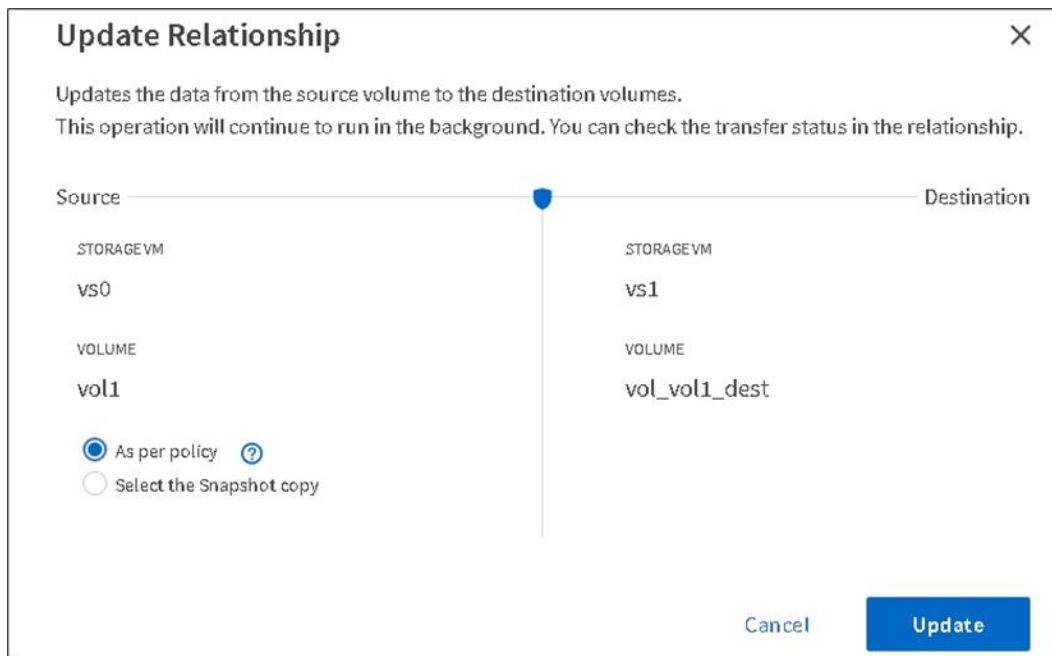


Figure 22) Relationship update dialog.



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

# Converting between different data protection modes

## Convert a legacy DP SnapMirror relationship to an XDP SnapMirror relationship

As of ONTAP 9.11.1, SnapMirror relationships of type DP are supported only for preexisting SnapMirror relationships within the cluster, and as of ONTAP 9.12.1, Snapmirror relationships of type DP are no longer supported. As of ONTAP 9.12.1, Lenovo requires that existing DP relationships be converted to new XDP relationships.

To perform this conversion, complete the following high-level steps:

1. Quiesce the SnapMirror relationship on the destination cluster.
2. Break the SnapMirror relationship on the destination cluster.
3. Delete this SnapMirror relationship.
4. Create a `-type XDP` relationship between the same endpoints.
5. Perform a resync between the endpoints. This resync converts a DR destination to a vault destination without another baseline. A metadata rebuild takes 10 to 12 minutes per TB of source data.

The following steps present this process in detail:

1. Quiesce the SnapMirror relationship

```
Remote::> snapmirror quiesce -destination-path vs1:vol_voll_dr
Operation succeeded: snapmirror quiesce for destination "vs1:vol_voll_dr".
```

2. Perform a SnapMirror break operation.

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

The status of SnapMirror relationship is displayed as `Broken Off`.

```
Remote::> snapmirror show -destination-path vs1:voll_voll_dr -fields state
source-path          desitnation-path      state
-----
Vs0:vs0voll          vs1:voll_voll_dr      broken off
```

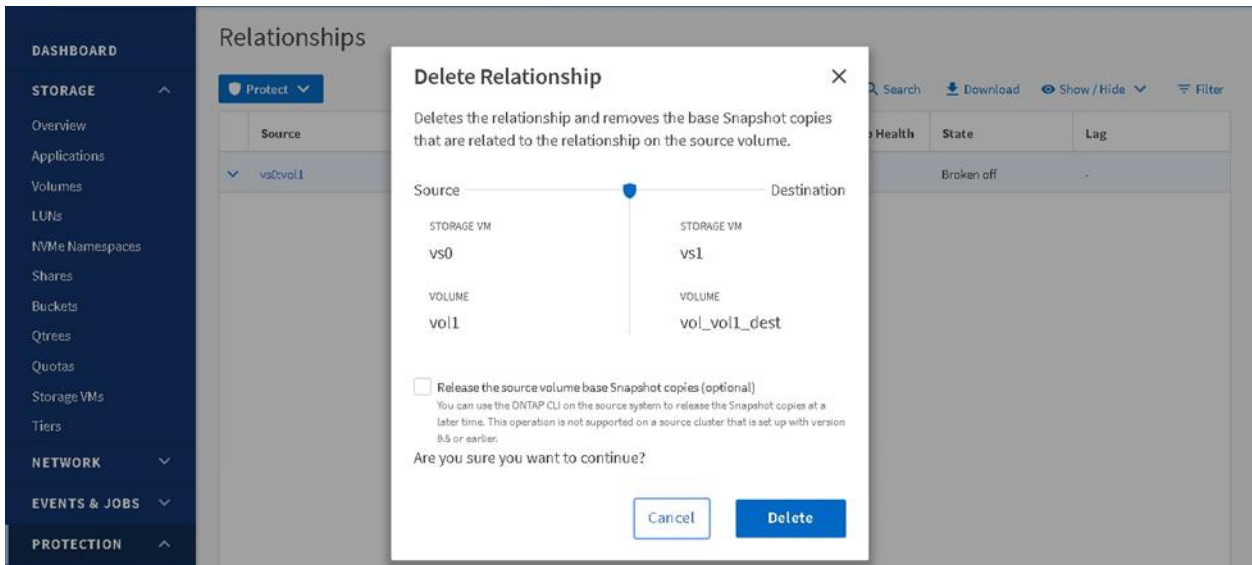
As a result of the break operation, the destination volume changes from DP to RW:

```
Remote::> volume show -volume voll_voll_dr -fields type
vserver              volume                  type
-----
Vs1                   voll_voll_dr           RW
```

3. Run the `snapMirror delete` command.

```
Remote::> snapmirror delete -destination-path vs1:vol_voll_dr
Operation succeeded: snapmirror delete the relationship with destination vs1:vol_voll_dr.
```

**Note:** ThinkSystem Storage Manager includes the release operation by deleting the relationship without releasing the base Snapshot copies. Uncheck the Release the Source Volume Base Snapshot Copies option to ensure that a new baseline is not created. This base Snapshot copy is needed so that another baseline is not required to create the XDP relationship.



#### 4. Create the XDP relationship.

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

#### 5. Verify that the relationship has been created and the Mirror State is Broken-off.

```
Remote::> snapmirror show
```

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

#### 6. Perform a snapMirror resync operation.

```
Remote::> snapmirror resync -destination-path vs1:vol_vol1_dr
```

Warning:

```
All data newer than Snapshot copy snapmirror.c7114203-3ad0-11eb-839e-005056a7bc32_2163602240.2020-12-11_032052 on volume vs1:vol_vol1_dest will be deleted.
Do you want to continue? {y|n}: y
Operation is queued: initiate snapmirror resync to destination "vs1:vol_vol1_dest".
```

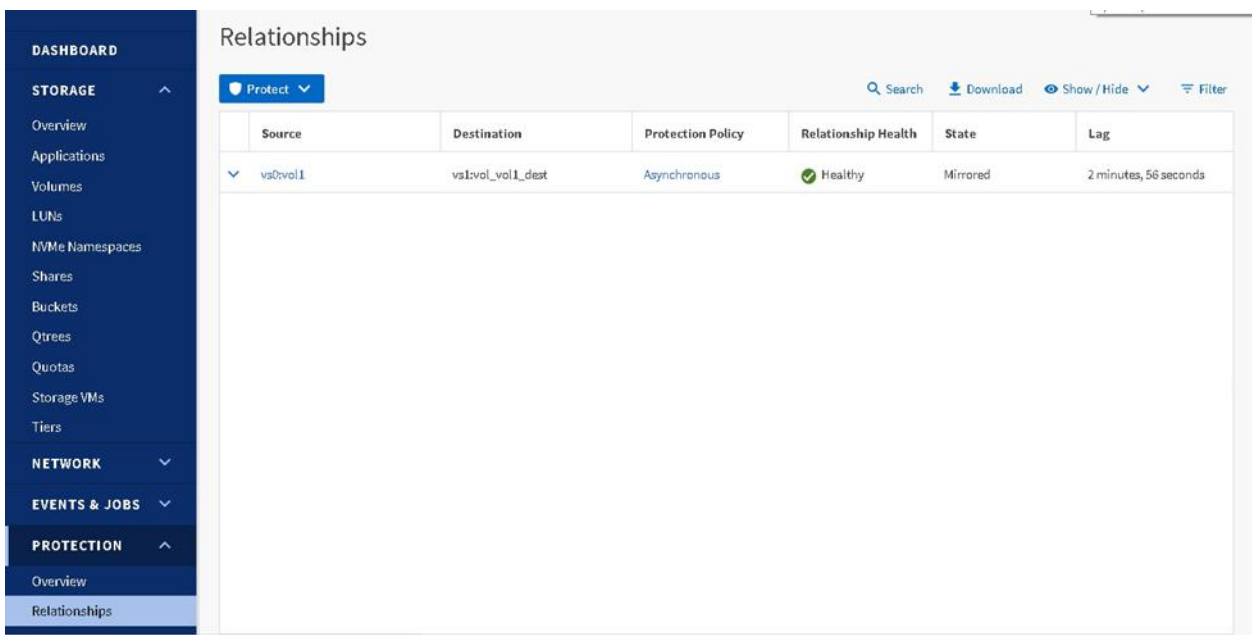
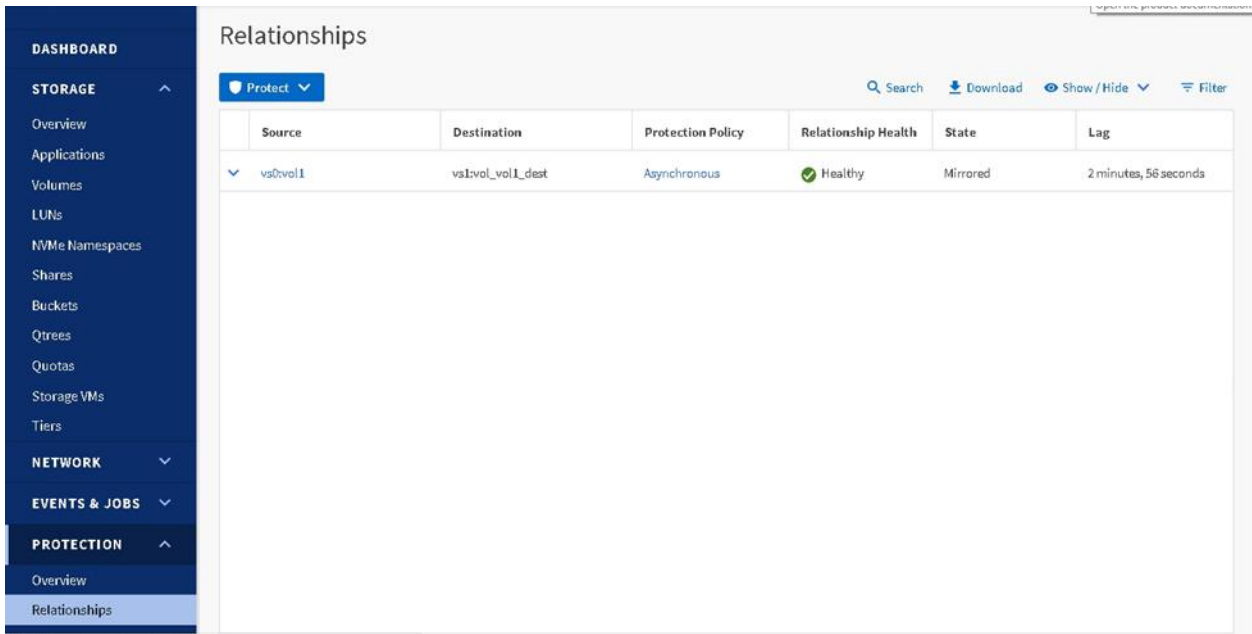
#### 7. Run the snapmirror show command to view the relationship type and status.

```
Remote::> snapmirror show
```

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

#### 8. Verify the relationship status is Snapmirrored in Storage Manager.





9. The SnapMirror destination volume is of type Data Protection for use as a DR volume.

```
Remote::> volume show -volume vol1_vol1_dr -fields type
vserver          volume          type
-----
Vs1              vol1_vol1_dr   DP
```

The screenshot shows the 'Volumes' page in the ONTAP management interface. The left sidebar contains navigation options: DASHBOARD, STORAGE (Overview, Applications, Volumes, LUNs, NVMe Namespaces, Shares, Buckets, Qtree, Quotas, Storage VMs, Tiers), NETWORK, EVENTS & JOBS, PROTECTION (Overview, Relationships), HOSTS, and CLUSTER. The main content area displays a table of volumes with columns: Name, Storage VM, Status, Capacity (available | total), IOPS, Latency (ms), Throughput (MB/s), and Protection. Below the table, a detailed view for a volume shows properties like STATUS (Online), EXPORT POLICY (default), TYPE (Data Protection), STORAGE EFFICIENCY (Enabled), SNAPSHOT COPIES (LOCAL), and SNAPSHOT POLICY (none).

Name	Storage VM	Status	Capacity (available   total)	IOPS	Latency (ms)	Throughput (MB/s)	Protection
root_vs0	vs0	Online	53.5 MB   160 MB	0	0	0	Protected
vol1	vs0	Online	18.4 MB   20 MB	0	0	0	Protected
vol1_dr	vs1	Online	18.7 MB   20 MB	0	0	0	Protected
vol_dr	vs1	Online	19.8 MB   20 MB	0	0	0	Protected
vol1_dest	vs1	Online	20.4 MB   22 MB	0	0	0	Protected
vs1_root	vs1	Online	18.2 MB   20 MB	0	0	0	Protected

## Convert SnapMirror to unified replication

Consider the following scenario: an existing customer using SnapMirror wants to use SnapMirror unified replication to use a single destination volume for DR and backup or archival. Complete the following steps for the conversion:

1. Break the SnapMirror volume relationship from the destination cluster.
2. Delete the SnapMirror volume relationship.
3. Create a unified relationship (`mirror-vault`) 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 rebaseline.

The following steps present this process in detail:

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

**Relationships**

Protect

Search Download Show/Hide Filter

Source	Destination	Protection Policy	Relationship Health	State	Lag
vs0:vol1	vs1:vol_vol1_dest	DPDefault	Healthy	Mirrored	1 minute, 42 seconds

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

**Relationships**

Protect

Search Download Show/Hide Filter

Source	Destination	Protection Policy	Relationship Health	State	Lag
vs0:vol1	vs1:vol_vol1_dest	DPDefault	Healthy	Broken off	-

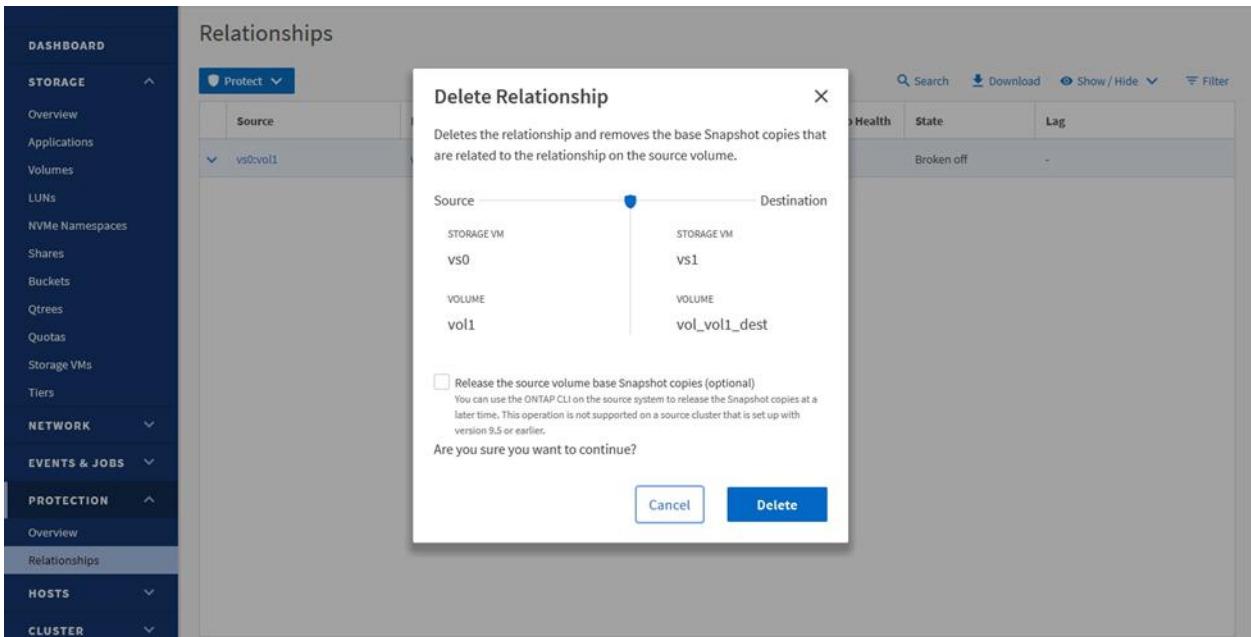
3. Delete the SnapMirror relationship:

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

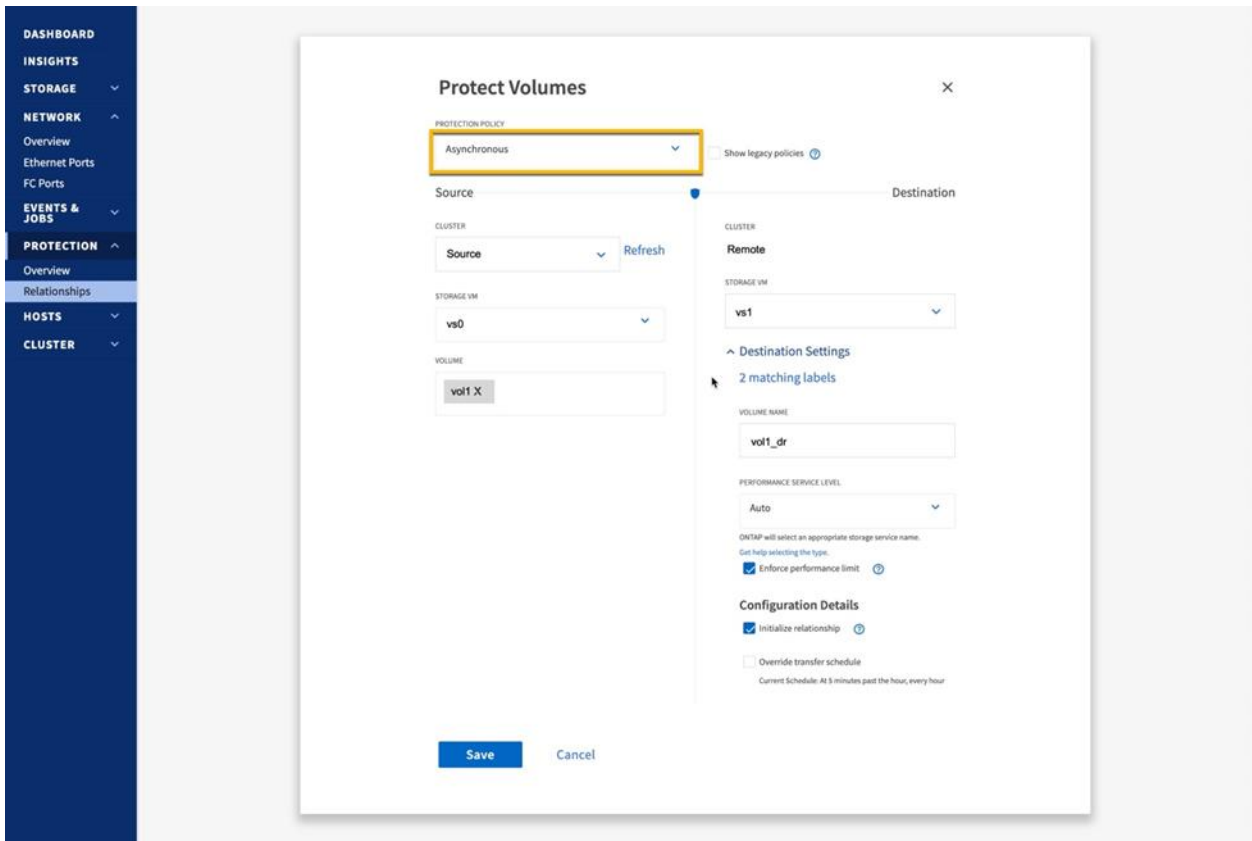


This relationship disappears from the destination cluster under Protection > Relationships.

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

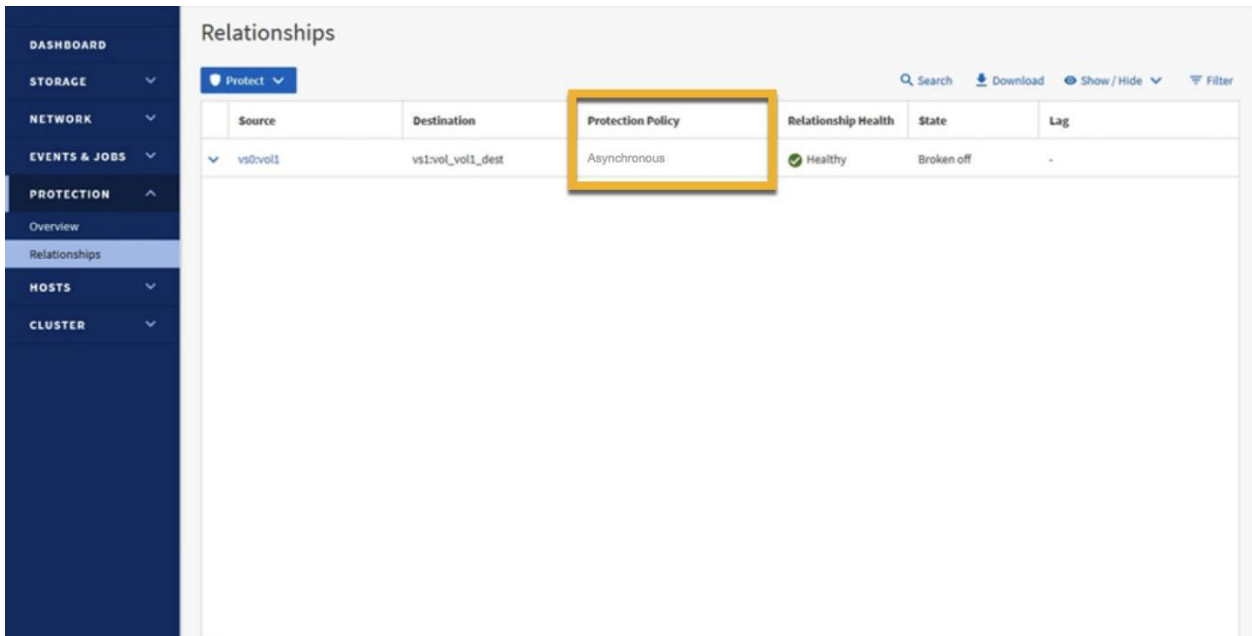


5. Verify that the new relationship is in place:
  - a. Using the CLI, run the `snapmirror show` command.

```
Remote:::> snapmirror show
```

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

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

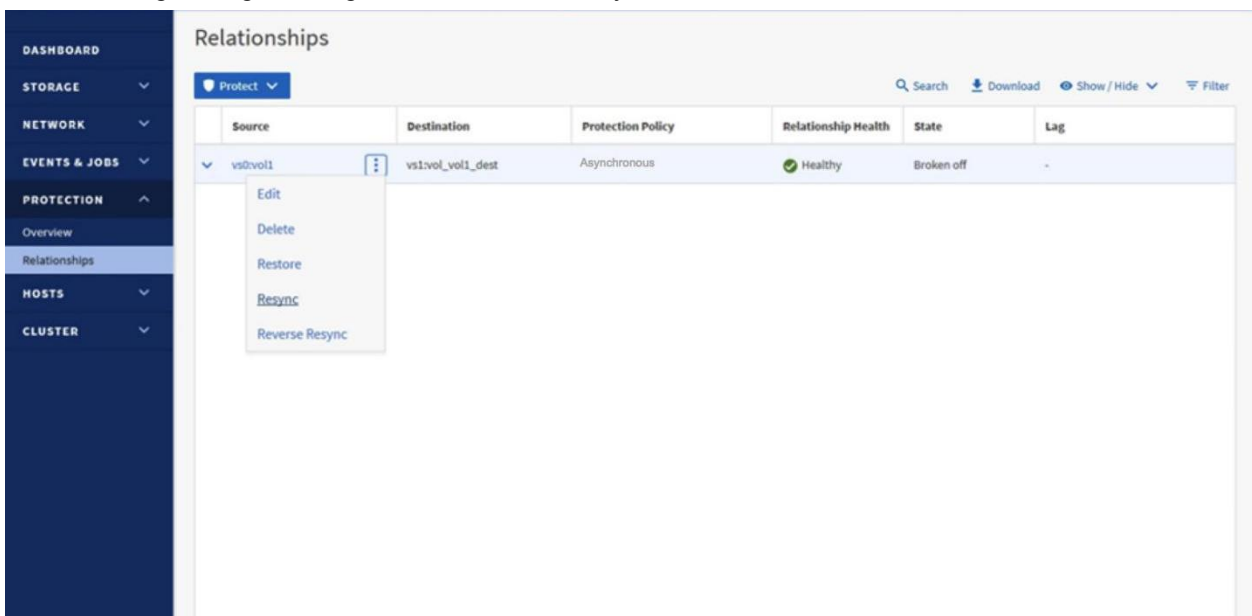


6. Resync the SnapMirror relationship:

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

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

- b. Using Storage Manager, select More > Resync.



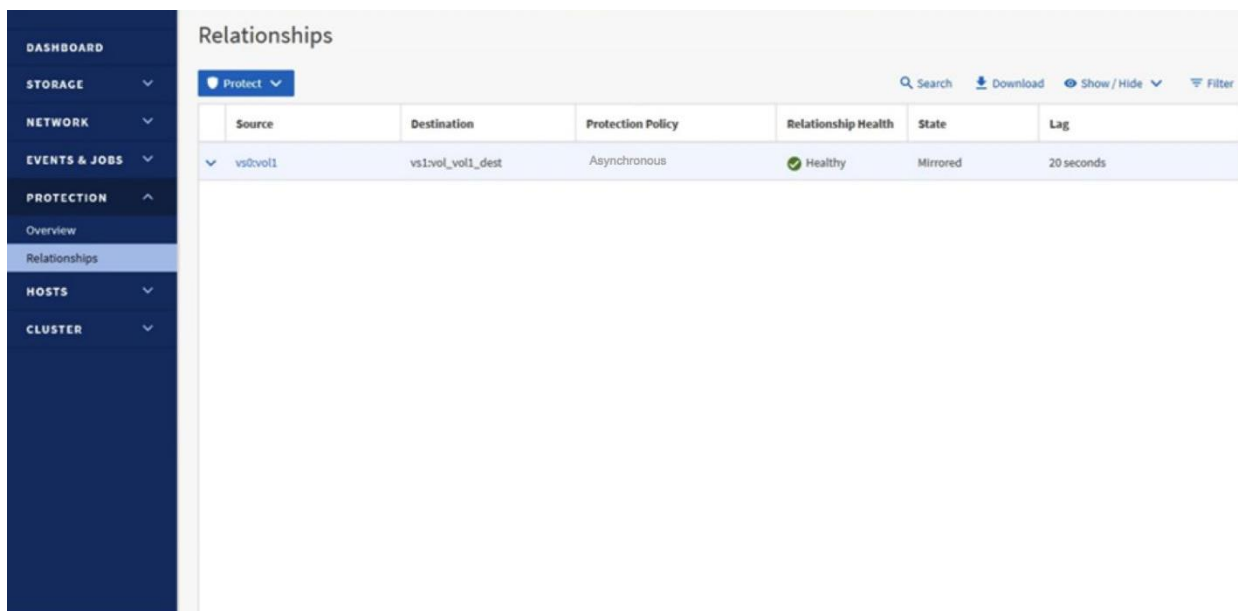
7. Verify the relationship status:

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

```
Remote:::> snapmirror show
```

Source Path	Type	Destination Path	Mirror State	Relationship Status	Total Progress	Healthy	Progress Last Updated
vs0:vol1	XDP	vs1:vol_voll_dr	Snapmirrored	Idle	-	true	-

- b. Using Storage 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 -volume voll` command and verify that the volume type is listed as `DP`.

```
Remote:::> volume show -volume voll
```

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
vs1	voll_dr	data_01	online	DP	250GB	217.8GB	12%

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

	Name	Storage VM	Status	Capacity (available   total)	Throughput (MB/s)	Protection	Type
▼	root_vs0	vs0	Online	94.7 MB   100 MB	0	🟢 🛡️	Read/Write
▼	vol1	vs0	Online	17.5 MB   20 MB	0	🟢 🟢	Read/Write
▼	vol1_dr	vs1	Online	17.7 MB   20 MB	0	🟢 🛡️	Read/Write
▼	vol_dr	vs1	Online	19.8 MB   20 MB	0	🛡️ 🛡️	Data Protection
▼	vol_vol1_dest	vs1	Online	17.3 MB   20 MB	0	🛡️ 🛡️	Data Protection
▼	vs1_root	vs1	Online	17.7 MB   20 MB	0	🟢 🛡️	Read/Write

## SnapMirror and ONTAP feature interaction

### SnapMirror and Snapshot copies

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

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

In the following example, a SnapMirror resync is performed on a volume where all Snapshot copies created by SnapMirror were deleted and uses the hourly Snapshot copy 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 tamper-proof Snapshot copies

Introduced in ONTAP 9.12.1, tamper-proof Snapshot copies can provide indelible protection for individual Snapshot copies without the need for a dedicated SnapLock volume. SnapMirror and SVM DR replicate Snapshot copies that are marked as tamper-proof along with other Snapshot copies based on the SnapMirror replication policy rules. The replicated tamper-proof Snapshot copies retain the same indelible characteristics of the original tamper-proof Snapshot copy.

**Note:** Though tamper-proof Snapshot copies do not require the configuration of a SnapLock volume, it does require a SnapLock license on each cluster that implements this feature.

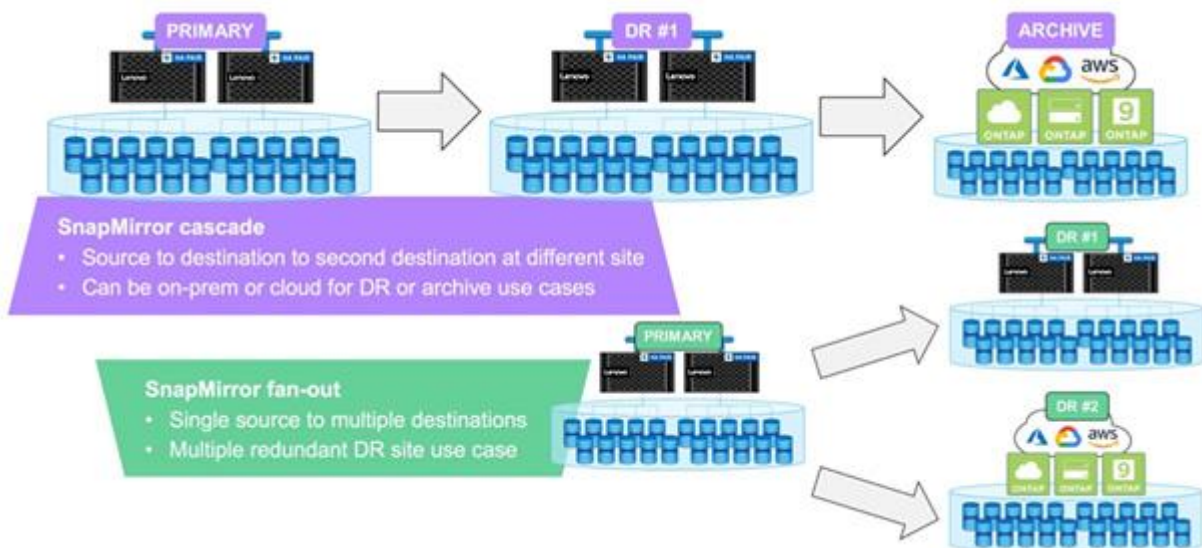
## SnapMirror and qtrees

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 operates 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 23). Destinations can be on-premises or cloud-hosted Cloud Volumes ONTAP clusters.

Figure 23) FlexGroup volumes used in SnapMirror cascade and fan-out configurations.

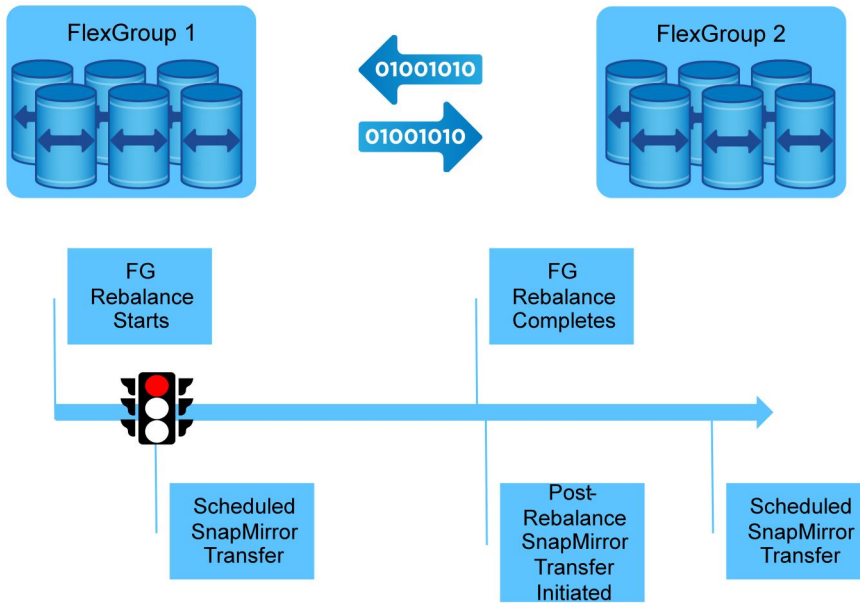


**Note:** Starting with ONTAP 9.12.1, SnapMirror SVM DR supports the replication of SVMs hosting both FlexGroup volumes and FabricPool aggregates.

## 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 preventing a scheduled SnapMirror operation from completing. This updated process is illustrated in Figure 24.

**Figure 24) SnapMirror behavior during FlexGroup rebalance operation.**



### SVM data mobility

SVM data mobility 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.

SVM migration supports up to 100 FlexVol volumes per SVM.

Nondisruptive (NDO) SVM migration is supported between AFA HA pairs for most NFS 3, NFS 4.1, and NFS 4.2 workloads.

Starting with ONTAP 9.11.1, SVM migration supports clusters with up to three HA pairs in the source or destination clusters. Prior to 9.11.1, SVM migration was only supported on clusters with a single HA pair.

SVMs in a data protection relationship can host up to 600 FlexVol volumes.

Table 7 summarizes the limitations as of 9.11.1.

**Table 7) SVM migration limitation summary.**

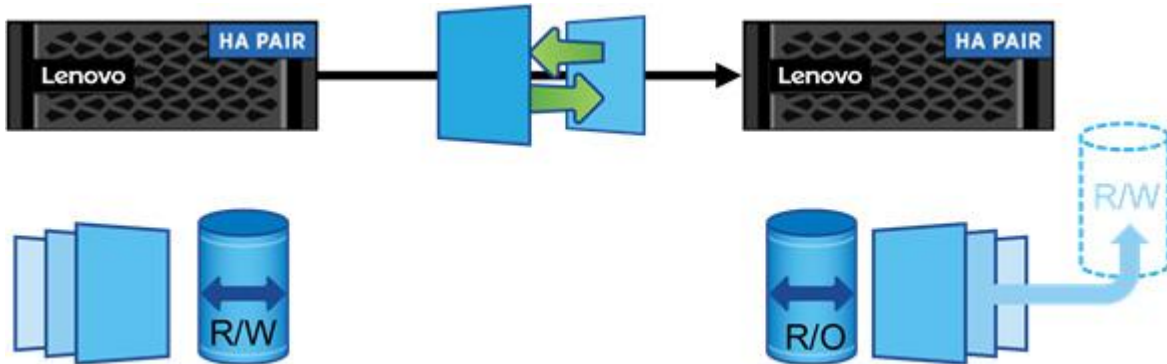
Feature	SVM data mobility
Scale	100 FlexVol volumes Six-node clusters
Network	L2 with < 2msec RTT latency
Platform	AFA only ONTAP version 9.10.1 or later
Protocols	NFS 3, NFS 4.1, NFS 4.2
Data management	Snapshot copies Storage efficiency
Data security	LVE OKM

Feature	SVM data mobility
	External Key Management (EKM)

## 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 25 illustrates the creation of a FlexClone volume at the SnapMirror destination.

Figure 25) Creating a FlexClone volume at a SnapMirror destination.



SnapMirror replicates the Snapshot copy history from a source to a destination volume. If a Snapshot copy is removed from the source volume, the next SnapMirror update removes that Snapshot copy from the destination volume. If that Snapshot copy 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 copy dependency.

To avoid this issue when creating FlexClone volumes on SnapMirror destination, manually create a base Snapshot copy required by the FlexClone volume on the source system, then replicate that Snapshot copy to the destination system and use that Snapshot copy as the base for the FlexClone volume, as shown in Figure 25. Using a Snapshot copy specifically created for the FlexClone volume in this manner prevents the SnapMirror update from failing due to an automatically created Snapshot copy being removed from the source system. The label associated with this Snapshot copy should also not use any labels that are associated with any SnapMirror replication policy rules in effect. This ensures that the retention policies for Snapshot copies using the preexisting labels do not attempt to delete the Snapshot copy associated with the FlexClone volume.

## SnapMirror and Lenovo storage efficiency

SnapMirror maintains storage efficiency benefits in replicated volumes. If the source volume is deduplicated, the destination volume is in a deduplicated state as well. SnapMirror does not inflate deduplicated data during a transfer. If the source volume is compressed, the destination volume is in a compressed state as well. Replication of compressed volumes does not decompress the source volume to read data for a transfer. Rather, data is replicated in a compressed state to the destination volume.

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 creates a Snapshot copy before performing an update transfer. Any blocks in the Snapshot copy 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 undeduplicated data and additional temporary deduplication metadata files over the network.

## 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 [ONTAP documentation on this subject](#).

## SnapMirror for disk shelf failure protection

SnapMirror can mirror volumes to nodes in a different HA pair on the same cluster. Mirroring to a different HA pair ensures that the other volume is always placed on a different disk shelf. If mirroring to a different disk 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 disk from any disk shelf due to drive failure and having a spare assigned. This configuration avoids having a single point of failure and provides protection against disk shelf failure.

One caveat is that the configuration does not fail over 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, Lenovo recommends that Autosize 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 Lenovo Snapshot copies 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.

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

## SnapMirror for consistency groups

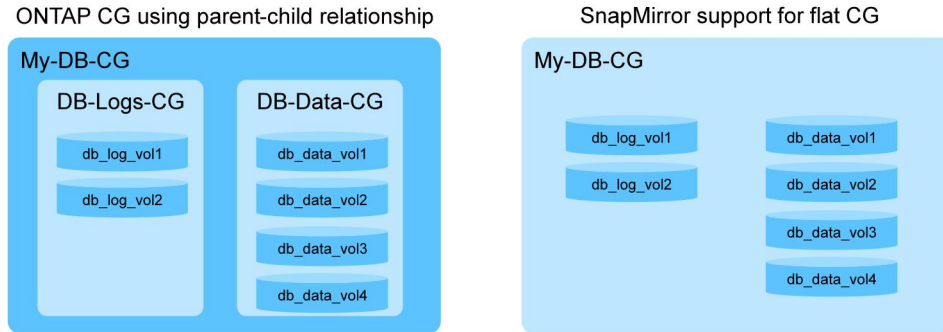
ONTAP supports the use of consistency groups (CGs) to group multiple volumes such that operations can be applied to the group of volumes in a coherent way. This ensures that all volumes in a CG are write consistent with each other. CGs are typically used for backup of more advanced applications – such 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 CG that contain volumes...a parent-child CG configuration. This feature is used primarily for Snapshot copy 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.

**Note:** SnapMirror SVM DR relationships will not replicate SVM volumes based on CGs. SVM DR will only replicate volumes individually.

Figure 26 compares parent-child CGs to flat CGs supported by SnapMirror for asynchronous replication.

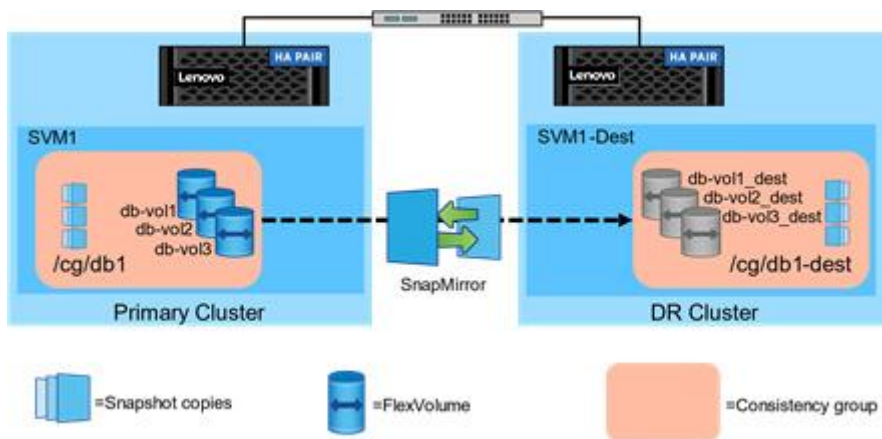
**Figure 26: Comparing ONTAP CGs with SnapMirror CG support**



## 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 27).

**Figure 27: SnapMirror for consistency groups**



## 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 ThinkSystem Storage Manager.

## Creating SnapMirror relationships for CGs using ONTAP CLI

Using Figure 27 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 27 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 ThinkSystem Storage Manager

ThinkSystem Storage Manager can be used to create SnapMirror relationships for CG. The process for creating a SnapMirror relationship for a CG using Storage Manager will be the same as for creating a volume scoped SnapMirror relationship. When Storage 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 Storage 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 volumes per CG. The maximum number of constituent volumes supported per cluster is 400 Flexvol volumes.

## Managing SnapMirror protection of CGs

CGs add an additional layer of complexity to the data protection process. 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:

1. All constituent volumes within the CG will be fenced from all I/O operations.
2. A CG Snapshot copy of each constituent volume is created.

3. The fence is then removed to permit continued I/O operations.
4. SnapMirror will replicate the changes captured in the constituent volumes' CG Snapshot copies to the DR site as a single replication operation.

**Note:** Independent snapshots created on each constituent volume using a constituent volume's snapshot policy are not replicated as part of the SnapMirror CG relationship. Only CG snapshots are replicated as part of the SnapMirror CG relationship.

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

## Converting the 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:

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 supports the following lifecycle functions:

- Modifying CG Snapshot copy 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 add or remove constituent volumes to or from an existing CG that has a SnapMirror relationship in place. The following steps can be used to accomplish this task:

1. Delete the current SnapMirror relationship for the CG on destination cluster
2. Release the SnapMirror relationship with the `-relationship-info-only true` parameter on the source cluster.
3. Add 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 copy between the new source and destination CGs.

## Adding constituent volumes to a CG

Consider the scenario where additional volumes need to be added to a 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 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:

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:
  - a. Perform a baseline replication on new volumes.
  - b. 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 copy prior to the ONTAP CG expand will be available. The new CG Snapshot copy on the expanded ONTAP CG will be available after the successful resync operation.

## Interoperability with other ONTAP features

Table 8 defines how SnapMirror relationships for CGs interoperates with other ONTAP features

**Table 8) 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. Snapshots created on constituent volumes

Category	Feature	Definition
		independently of the CG are not replicated.
	Volume SnapMirror	Constituent volumes can have separate volume-scoped asynchronous SnapMirror relationships while in a protected CG.
	SVM DR	SVM DR is not supported for SVMs containing CGs
	MetroCluster	Supported
	SnapLock®	SnapLock volumes cannot be constituent members of a protected CG
Lifecycle Management	FlexGroup volumes	FlexGroup volumes cannot be constituents of a protected CG
	FabricPool	Supported
	FlexCache	FlexCache is not compatible with protected CGs
	Volume encryption	Constituent volumes can be encrypted at the volume level
	Aggregate encryption	Constituent volume can reside on encrypted aggregates

## SnapMirror for SVM DR

SnapMirror can be used to protect entire SVMs and is called SVM DR. SVM DR replicates FlexVol volumes and FlexGroup volumes owned by the SVM as well as the SVM configuration and identity information to a remote destination. Although SVM DR uses the same SnapMirror replication technology as SnapMirror for volumes, there are differences (listed in Table 9).

**Table 9) Differences between SVM DR and SnapMirror.**

SVM DR	SnapMirror
Works at SVM-level granularity	Works at volume-level granularity
Protects FlexVol and FlexGroup data and SVM configuration	Protects only data stored in FlexVol and FlexGroup volumes
Minimum supported RPO is 15 minutes	Minimum supported RPO is 5 minutes

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 or FlexGroup volumes created in the protected SVM.

**Note:** SVM DR, when used for DR, supports replication between clusters running the same major 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 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.

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.

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.

When creating an SVM DR relationship, SnapMirror can be configured in two modes that determine whether the SVM configuration information is replicated or just the SVM's volumes are replicated. 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.

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 source -source-vserver vs0 -destination-cluster remote
-destination-vserver vs1 -policy Asynchronous -schedule daily -type XDP -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 located in a different subnet than the source SVM. To replicate SVM configuration information without network settings, a custom SnapMirror policy must be created using the `-discard-configs network` parameter. This parameter is only supported with SnapMirror policy type `async-mirror`.

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

For details regarding which SVM configuration information is replicated, see [About SnapMirror SVM replication](#).

## SVM DR scalability

SVM DR has different scalability limits than volume scoped SnapMirror relationships. Table 10 lists the scalability limits of SVM DR data protection relationships.

**Table 10) SVM DR scalability.**

Parameter	Limits
Number of SVM DR relationships per HA pair	ONTAP 9.9.1 and earlier: 32 ONTAP 9.10.1: 64 ONTAP 9.11.1 and later: 128  SVMs containing FlexGroup volumes are limited to 32 relationships per HA Pair.
Number of volumes per SVM	300
Number of volumes per HA pair	1,000

## 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.

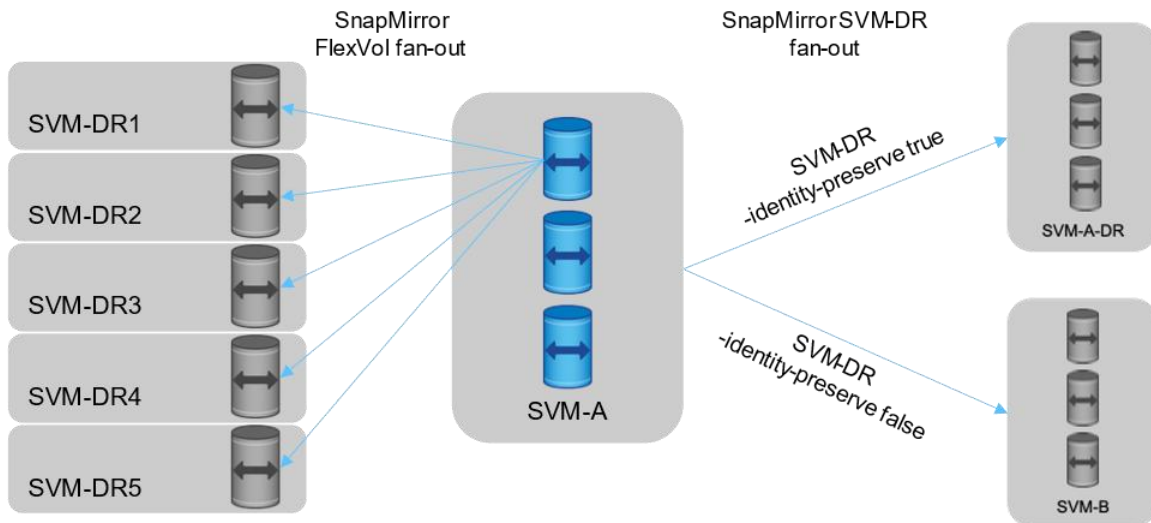
### SVM DR fan-in and fan-out

SVM DR supports limited fan-out capabilities. A single source SVM can be replicated to up to two destination SVMs. In this configuration, only one of the SVM DR relationships can be configured with the `-identity-preserve true` parameter. The second SVM DR relationship for the source SVM must use the `-identity-preserve false` parameter.

Volumes that are contained in an SVM DR source SVM cannot be the target for another volume scoped SnapMirror relationship. Volumes that are contained in an SVM DR source SVM can be the source for other SnapMirror relationships. Fan-out examples are illustrated in Figure 28.

Starting with ONTAP 9.13.1, SVM DR fan-out can be configured for SVMs containing FlexGroup volumes. Prior to ONTAP 9.12.1, SVM DR fan-out only supports FlexVol volumes.

Figure 28) SVM DR fan-out example.



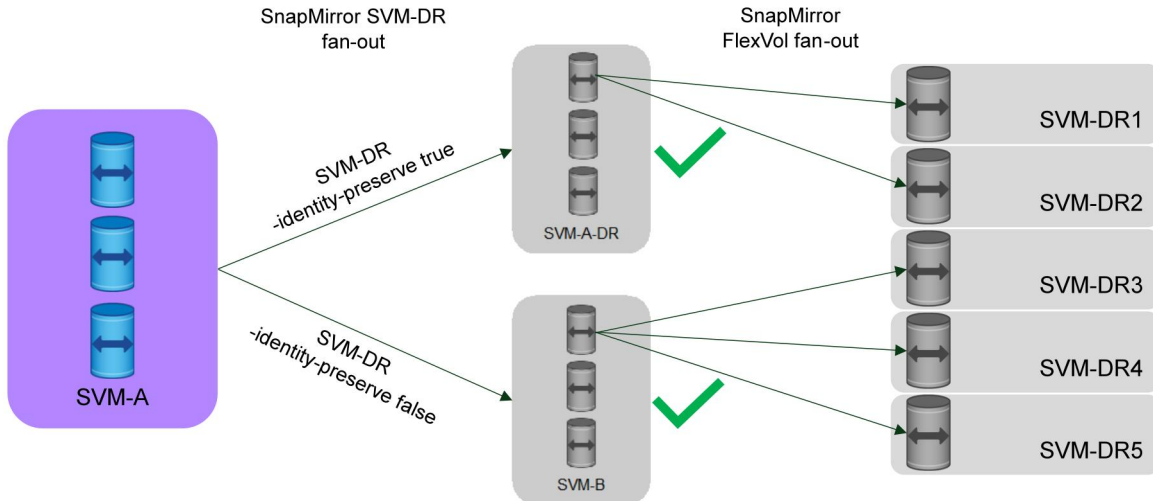
### SVM DR cascade

Configuring a cascade relationship of a target SVM in an SVM DR relationship is not supported.

SnapMirror does support configuring relationships for the DR SVM's constituent (DP).

Volumes contained in the destination SVM of an SVM DR relationship that is using `-identity-preserve true` are not accessible for read access by clients because the destination SVM is not running as its configuration would conflict with the source SVM on the network, but SnapMirror can replicate the volumes of the offline destination SVM's volumes as a source for a downstream replication relationship. Figure 29 highlights how SVM DR interacts with cascade architectures.

**Figure 29) SVM DR cascade example.**

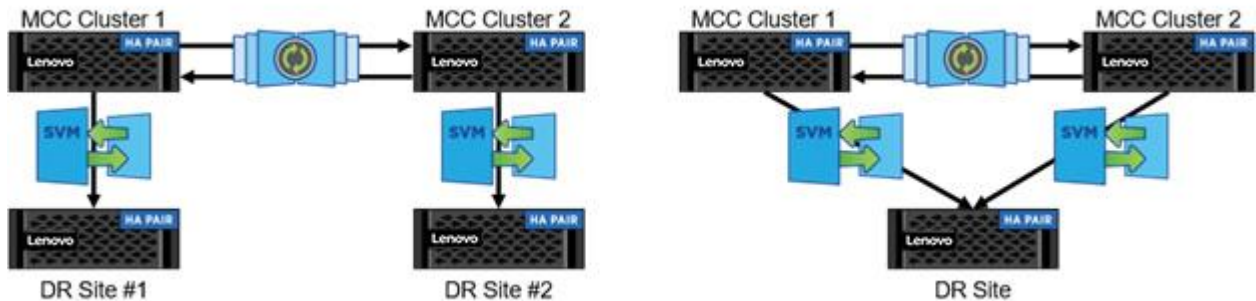


## SVM DR interoperability

### 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 30.

**Figure 30) SVM DR with MetroCluster.**



### 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.

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

- FlexClone volumes on source or destination clusters

- SnapMirror fan-out configurations
- SnapMirror cascade configurations
- Converting FlexVol volumes to FlexGroup volumes

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

## SVM DR and cloud target interoperability

SVMs and SnapMirror are supported to some extent by Lenovo Cloud Volumes ONTAP implementations offered by major cloud providers. 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 Lenovo Cloud Backup Service.

## 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. For more information, see [Lenovo Press](#).
- Type of transfer: initialization or update  
A new SnapMirror relationship requires a baseline Snapshot copy 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 copy creation.
- Node hardware type.  
The node model, configuration (including disk types, number of disks 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.** Data replication is consuming all available network bandwidth, but there are other applications and services being introduced that also require network bandwidth.
- **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.

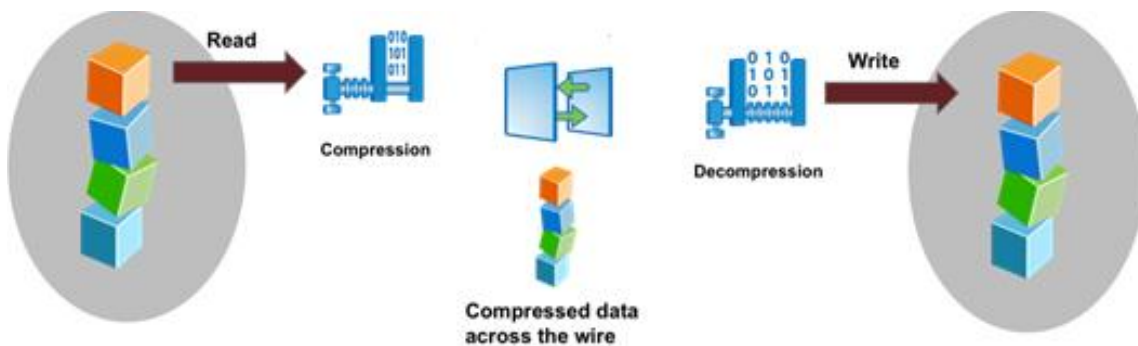
## Speeding up the initial transfers

- **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 31 shows a very high-level flow of SnapMirror network compression.

Figure 31) 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
```

**Note:** 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.

**Note:** 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 `ctlopccp` 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 11) 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 grows 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. For information about the number of concurrent SnapMirror operations allowed per node model, see [Lenovo Press](#).

### 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 Lenovo 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. 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.

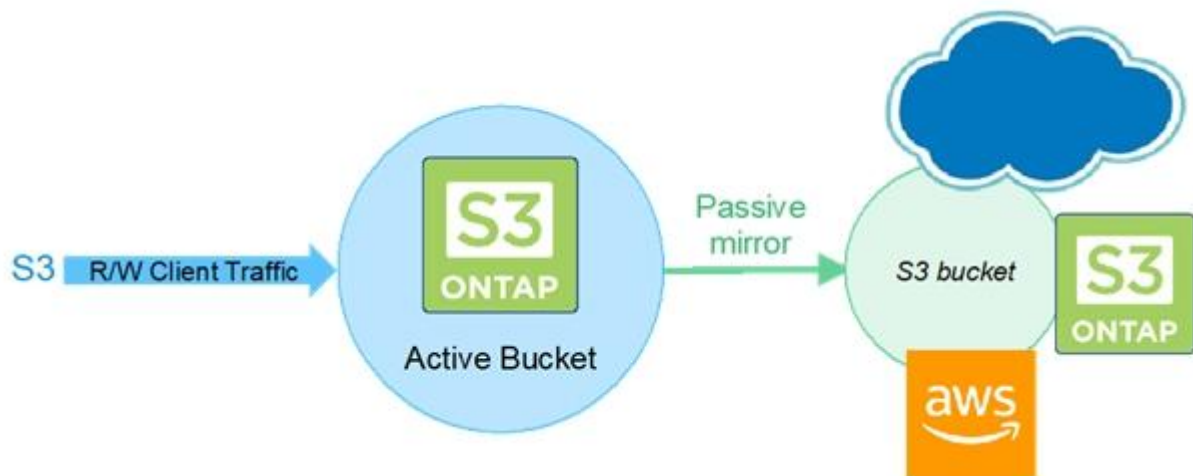
## Network sizing requirements for intracluster replication

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.

## 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, ONTAP Cloud Volumes ONTAP S3 buckets and native cloud-provided S3 buckets such as AWS S3 (Figure 32).

Figure 32) 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`).

## Interoperability

Refer to [Lenovo Data Center Support](#) to understand details about hardware and limits based on each platform. Also, refer to the [Lenovo Storage Interoperation Center \(LSIC\)](#) on the Lenovo Data Center Support Site to validate that the exact product and feature versions described in this document are supported. The Lenovo LSIC defines the product components and versions that can be used to construct configurations that are supported by Lenovo. Specific results depend on each customer's installation in accordance with the published specifications. Verify that the source and destination volumes are running [compatible ONTAP versions before creating a SnapMirror data protection relationship](#).

# Troubleshooting tips

## Troubleshooting cluster peer relationships

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
Vserver      Peer      Peer
            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
Vserver      Peer      Peer      Peering
            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 Storage Manager.

```
cluster02::> snapmirror show
Source      Destination  Mirror  Relationship  Total  Progress
Path       Type  Path      State  Status  Progress  Healthy  Last 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:SMSSource
Destination Path: svm_dst1:SMSSource_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:SMSSource
    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 troubleshooting issues, review information about relationships in the event log. Use the `-messagemname` 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 -messagemname 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. Storage Manager allows viewing of the SnapMirror log files.

## 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 is 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 reenforce 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 Lenovo 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.



## Best practices

- **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.

## 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 Lenovo 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.

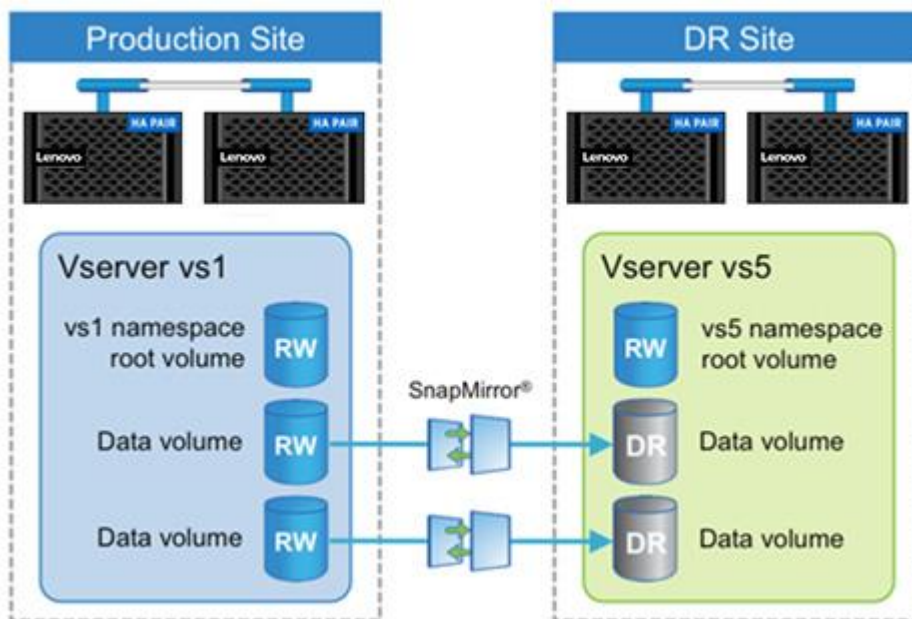
- The required capacity exists on the DR Lenovo controller. This refers to the capacity required to support day-to-day operations that have been planned for in the DR environment.
- All DR site application servers have the proper connectivity configured to be able to connect to the DR storage arrays.
- 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 Lenovo storage efficiency policies.

Figure 33 illustrates volume layout for DR.

**Figure 33) Volume layout for DR.**



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

## NAS and SAN environments

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 Lenovo 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.
 

**Note:** Lenovo recommends configuring Snapshot copy policies in the destination cluster with the same schedules as those in the source. Snapshot copy policies must be applied after failover (`snapmirror break`) has completed.
4. Create Lenovo storage efficiency policies in the destination SVM.
 

**Note:** If Lenovo 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 environments

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 environments

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

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

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

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

## Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- Lenovo Data Center Support Site  
<https://datacentersupport.lenovo.com/>
- ONTAP and ThinkSystem Storage Manager documentation resources  
[https://thinksystem.lenovofiles.com/storage/help/topic/ontap\\_software/overview.html](https://thinksystem.lenovofiles.com/storage/help/topic/ontap_software/overview.html)

- ONTAP command reference  
[https://thinksystem.lenovofiles.com/storage/help/topic/command-man-pages\\_9.13.1/index.html](https://thinksystem.lenovofiles.com/storage/help/topic/command-man-pages_9.13.1/index.html)

## Version history

Version	Date	Document Version History
Version 2.0	October 2023	Second release
Version 1.0	March 2021	Initial release

# Contacting Support

You can contact Support to obtain help for your issue.

You can receive hardware service through a Lenovo Authorized Service Provider. To locate a service provider authorized by Lenovo to provide warranty service, go to <https://datacentersupport.lenovo.com/serviceprovider> and use filter searching for different countries. For Lenovo support telephone numbers, see <https://datacentersupport.lenovo.com/supportphonenumber> for your region support details.

## Notice

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area.

Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document is not an offer and does not provide a license under any patents or patent applications. You can send inquiries in writing to the following:

*Lenovo (United States), Inc.*

*8001 Development Drive*

*Morrisville, NC 27560*

*U.S.A.*

*Attention: Lenovo Director of Licensing*

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT,

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk .

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

## **Trademark**

LENOVO, LENOVO logo, and THINKSYSTEM are trademarks of Lenovo. All other trademarks are the property of their respective owners. © 2023 Lenovo