SnapMirror Configuration and Best Practices Guide for ONTAP 9

Updated for ONTAP 9.7

Abstract

This document describes information and best practices related to configuring replication in Lenovo® ONTAP.

First edition (March 2021)

TABLE OF CONTENTS

LIST OF FIGURES

1 Overview

Businesses can use several approaches to increase data availability in the face of hardware, software, or site failures. Data protection (DP) is one of the most critical aspects because any loss of data translates directly into lost money and time. Data protection is the process of taking data located in one location and making a copy of it in a different location to serve two use cases:

- **Backup.** The objective is to restore from the secondary to the primary with no intention of failing over to the secondary. This implies that the primary purpose of the secondary is archival storage. Therefore, you might have more data in the secondary than in the primary.
- **Disaster recovery (DR).** An exact replica or copy is maintained in the secondary and used for failover from the primary to the secondary if there is failure at the primary site.

Although backups allow you to recover lost data from an archival medium (tape, disk, or the cloud), mirroring is the most popular data availability mechanism for business continuity and DR, especially if you would like to minimize downtime. SnapMirror® technology offers a fast and flexible enterprise solution for mirroring or replicating data over local area networks (LANs) and wide area networks (WANs). The main advantages to using SnapMirror are as follows:

- **Robust enterprise technology.** SnapMirror is a mature feature of ONTAP storage systems that has been enhanced and improved over time. SnapMirror can recover from update failures, use concurrent processes for replication processing, throttle the network bandwidth used for transfer operations, and much more.
- **Speed and efficiency.** Block-level logical incremental data transfer makes sure that only the data that has changed is sent to the destination replica. SnapMirror can reduce data volume further with various storage efficiencies such as network compression to compress data as it leaves the source and decompress it at the destination, thereby improving transfer performance.
- **Flexibility.** SnapMirror allows you to define different synchronization schedules to better meet your system's needs. Using SnapMirror, you also can change the direction of the synchronization if there is a problem with the primary repository. SnapMirror can also be used to create a variety of replication topologies. Options include 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 using Lenovo FlexClone technology, irrespective of their size, and in a space-efficient manner, without needing to stop data being replicated from the source, and is invaluable for performing DR tests, for example.
- **Failover and failback.** If DR systems must be brought online, SnapMirror relationship can be broken, which makes the destination volumes read/write and ready to use. SnapMirror allows you to resynchronize the original source with the changes made at the destination and then re-establish the original SnapMirror relationship.
- **Ease of use.** With ThinkSystem Storage Manager for DM Series, you can perform operations with simplified workflows and wizard-guided walkthroughs. You can also monitor and manage all SnapMirror replication relationships in one place.
- **Secure.** From ONTAP 9.4 onwards, SnapMirror relationships can be encrypted natively end-to-end.

1.1 Purpose and Intended Audience

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

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

- A working knowledge of ONTAP operations

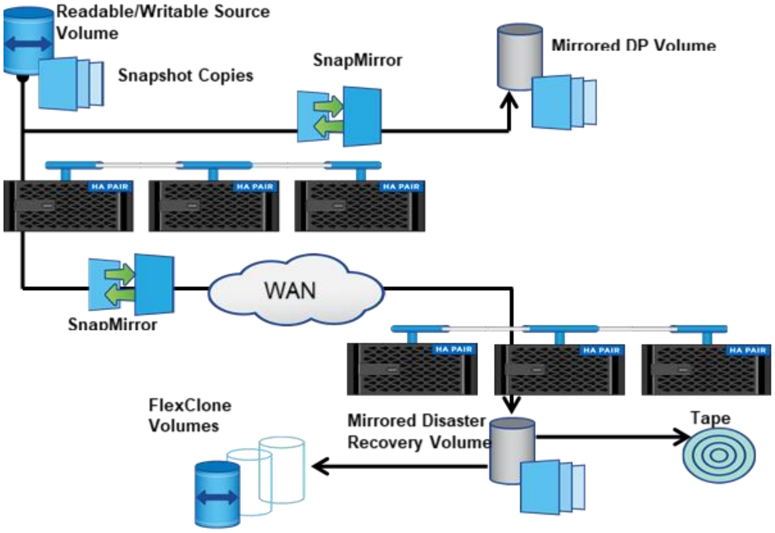
- A working knowledge of features such as Snapshot™ technology, FlexVol® or FlexGroup® volumes, and FlexClone® technology
- General knowledge of DR and data replication solutions
- Familiarity with the ONTAP Data Protection Guide on the Lenovo Data Center Support site

1.2 SnapMirror Overview

SnapMirror is a replicatiofn solution built into ONTAP for business continuity and DR purposes. SnapMirror is configured through a DP relationship between data volumes (FlexVol or FlexGroup) on 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.

This replica or mirror of enterprise data is created in a secondary storage system at a geographically remote site or in the cloud. You can fail over and serve data from the secondary in the event of a catastrophe at the primary site. After the error condition at the primary site is rectified, you can replicate any changes back to primary and start serving clients from the primary site again. With SnapMirror, you can reduce the total cost of ownership (TCO), making it easier to justify the DR investment by putting your DR site to active business use. For an overview of SnapMirror replication, see Figure 1.

Figure 1) SnapMirror replication overview.



Integrated Data Protection

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

Lenovo integrated DP can be used to create a quickly accessible on-disk history of application-consistent Snapshot copies that eliminates the concept of a traditional backup window. SnapMirror then replicates the history of Snapshot copies to the destination, which can be used for backup, DR, or test and development.

SnapMirror replication is efficient because it only replicates blocks that have changed or added since the previous update. Additional efficiency is gained when SnapMirror is combined with Lenovo storage efficiency technologies. Compression and data deduplication technologies can result in significant telecommunication and storage capacity savings.

SnapMirror for Disaster Recovery

SnapMirror technology is also used as a part of 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 from corruption, accidental deletion, natural disaster, or what have you.

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

Disaster Recovery Testing and Application Testing and Development

FlexClone technology can be used to quickly create a read-write copy of a SnapMirror destination FlexVol volume in case you want to have read-write access of the secondary copy to confirm if all the production data is available.

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 WAN usage occurs at a predetermined replication time. Storage administrators can replicate production data at a specific time to minimize overall network utilization.

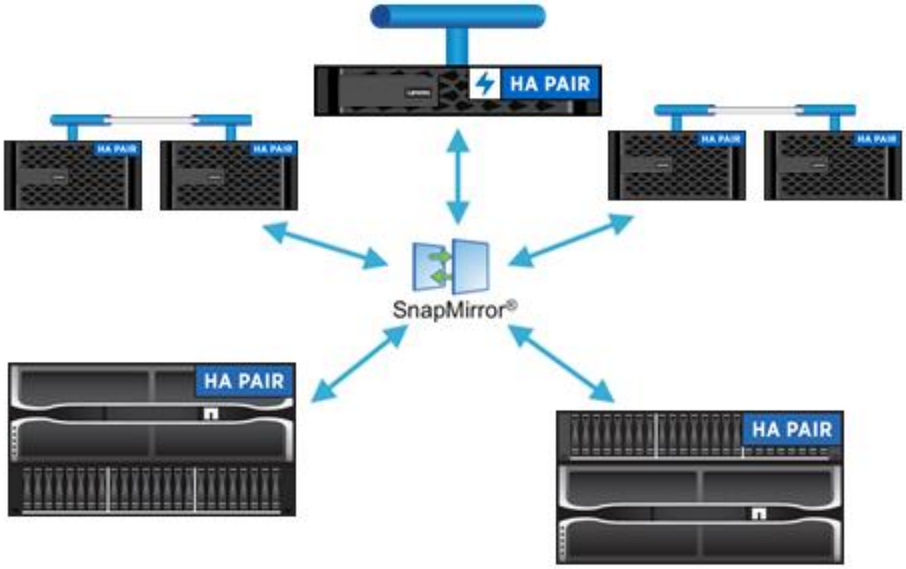
Backup Offloading and Remote Tape Archiving

SnapMirror technology can also be used for backup consolidation and for offloading tape backup overhead from production servers. This approach facilitates centralized backup operations and reduces backup administrative requirements at remote locations. Snapshot technology eliminates the traditional backup window on the primary storage system. Therefore, offloading tape backups to a SnapMirror destination (Figure 2) dramatically reduces the overhead of backup operations on production storage systems.

Unified Architecture Flexibility

SnapMirror technology can be used between ONTAP platforms with different performance characteristics (and different costs) at the primary and DR sites. For example, depending on the capabilities required, the DR site might contain a lower-end platform, with SATA disks rather than SAS or SSD disks, or the iSCSI protocol rather than FC. Figure 2 illustrates the flexibility within a unified architecture.

Figure 2) Unified architecture flexibility.



A unified architecture, from low-end platforms to high-end platforms, also allows system administrators to learn and use the same management and monitoring paradigm.

2 Lenovo Networking Basics

2.1 Lenovo ONTAP Overview

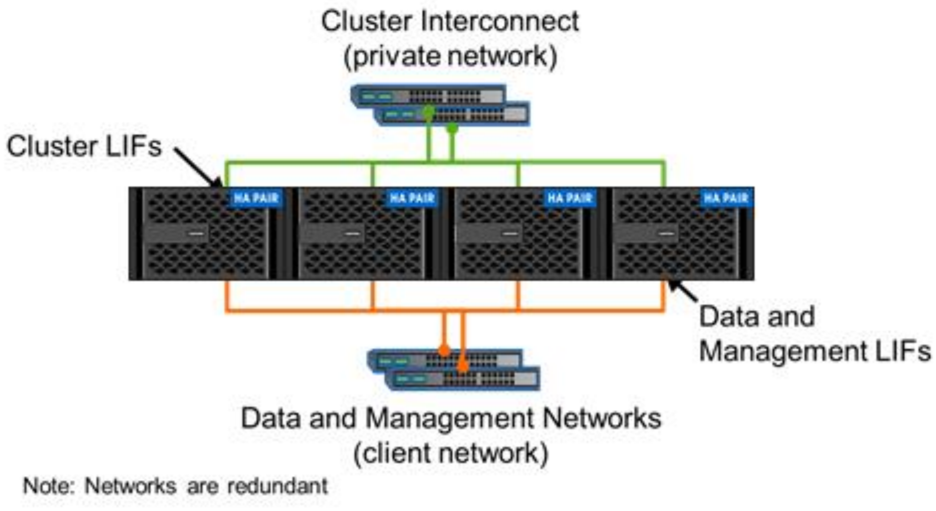
Some basic terms used in ONTAP include the following:

- **Node.** A single Lenovo controller in a high-availability (HA) pair.
- **Cluster.** One or more nodes that are interconnected and managed as a single system.
- **Cluster interconnect.** A dedicated high-speed, low-latency, private network used for communication and replication between nodes in the same cluster.
- **Data network.** The network used by clients to access data.
- **Management network.** The network used for administration of the cluster, SVM, and nodes.
- **HA interconnect.** The dedicated interconnect between two nodes in one HA pair.
- **HA pair.** Two nodes configured in a pair for HA.
- **Physical port.** A physical port such as `e0e` or `e0f` or a logical port such as a virtual LAN (VLAN) or an interface group (ifgrp).
- **Virtual port:**
  - **Ifgrp.** A collection of physical ports combined to create one logical port used for link aggregation.
  - **VLAN.** A VLAN 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 the networks. 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.
- **Logical interface (LIF).** A LIF is an IP address or a worldwide port name (WWPN) that is associated with a port. It is associated with attributes such as failover groups, failover rules, and firewall rules. A LIF communicates over the network through the port (physical or virtual) to which it is currently bound.

- **Intercluster LIF.** Logical network interfaces are used for intercluster communication, backup, and replication. You must create an intercluster LIF on each node in the cluster before a cluster peering relationship can be established. These LIFs can only fail over to ports in the same node. They cannot be migrated or failed over to another node in the cluster.
- **Intercluster network.** The network used for communication and replication between different clusters.
- **Intercluster ports.** Ports dedicated to intercluster replication.
- **Cluster peering.** The act of connecting two clusters to allow replication to occur between them.
- **SVM.** A logical storage server that provides data access to LUNs and/or a network-attached storage (NAS) namespace from one or more LIFs.

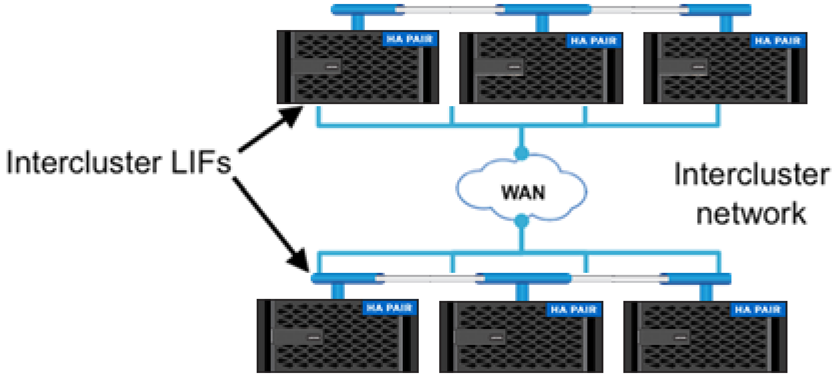
There are multiple types of networks in ONTAP, as shown in Figure 3 and Figure 4. It is important to understand what each network type is used for. The cluster interconnect network is a dedicated, high-speed, low-latency private network used for communication and replication between nodes in the same cluster. This configuration is a redundant back-end network that cannot be used or shared for client access to data or for managing the cluster, nodes, or SVMs. Client access to data occurs on the data network. Management of the cluster, nodes, and SVMs occurs on the management network. The data and management networks might share the same ports or physical network. However, the data and management networks must be a different physical network than the cluster interconnect network.

Figure 3) Cluster interconnect, data, and management networks.



The intercluster network must be configured, to enable cluster peering for replication of data from one geographical location to another as shown in Figure 4. The Intercluster network uses logical interfaces, or LIFs, that correspond to IP addresses and represent network access points to a node. Intercluster LIFs are assigned to ports as part of the cluster peer configuration process.

Figure 4) InterCluster network.



2.2 Intercluster Networking

The following are requirements for intercluster LIFs:

- At least one intercluster LIF must be configured on every node in the local cluster and every node in the remote cluster. Provisioning intercluster LIFs on only some nodes of the cluster is not supported.
- The IP addresses you assign to intercluster LIFs can reside in the same subnet as data LIFs or in a different subnet. Intercluster LIFs use routes that belong to the system SVM to which they are assigned. ONTAP automatically creates a system SVM for cluster-level communications within an IPspace.
- The cluster peering topology should use full-mesh connectivity. Full-mesh connectivity means that all the intercluster LIFs of one peer cluster can communicate with all the intercluster LIFs of the other peer cluster. All ports that are used to communicate with a given remote cluster must be in the same IPspace. You can use multiple IPspaces to peer with multiple clusters. Pairwise, full-mesh connectivity is required only within an IPspace. Also, consider using custom IPspaces to isolate replication traffic.
- 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 that node, as defined by the LIF's failover policy. At least one intercluster LIF is required per node for replication between clusters. Maintain consistent settings between the intercluster LIFs (the same maximum transmission units [MTUs], flow control, Transmission Control Protocol [TCP] options, and so on).
- SnapMirror replication over an FC network is not available in ONTAP.
- If a node fails while an intercluster SnapMirror transfer is in progress, the transfer automatically continues using an intercluster LIF on the surviving node of the HA pair. The same transfer does not automatically continue after the storage failover (SFO) of the destination. If SFO happens on the source, the transfer continues. However, replication as such continues automatically from the surviving node.

For additional information regarding intercluster networking, see the ONTAP 9 Data Protection Power Guide.

2.3 Intercluster Multipathing and Network Redundancy

You might want more than one physical path for a SnapMirror relationship. SnapMirror supports up to two paths for a SnapMirror relationship. When using multiple paths, you need to set up the configuration in one of the following ways:

- Set up static routes to ensure different routes are used for different IP connections.
- Use different subnets for the two connections.

The two paths can be used in one of these two modes:

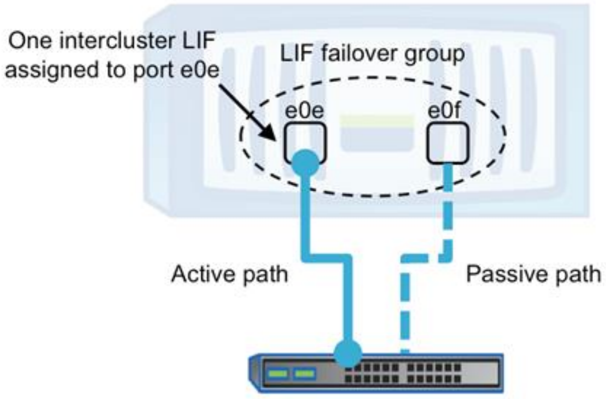
- **Failover mode.** SnapMirror uses the first specified path as the desired path and uses the second specified path only after the first path fails.
- **Multiplexing mode.** SnapMirror uses both paths at the same time, essentially load balancing the transfers. If one path fails, the transfers occur on the remaining path. After the failed path is repaired, the transfers resume using both paths.

Failover Mode

In many ways, an intercluster LIF behaves in the same way as a LIF used for CIFS or NFS in terms of failover, except that an intercluster LIF cannot fail over to a port in a different node. The initial placement of a LIF on a specific port determines which port is used by that LIF. If ports are redundant for failover on the same node, then the active path is the port where the initial LIF was placed. The passive path is any port where the LIF might fail over.

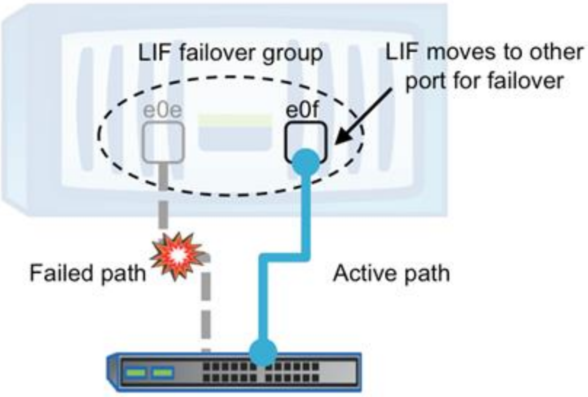
Therefore, it can be said that a properly configured redundant LIF provides failover multipathing, as shown in Figure 5:

Figure 5) Failover multipathing.



Communication on an intercluster LIF only occurs on the port to which the LIF is assigned unless that port fails, which causes the LIF to move to another surviving port in that LIF's failover group.

Figure 6) Failover multipathing during LIF failover.



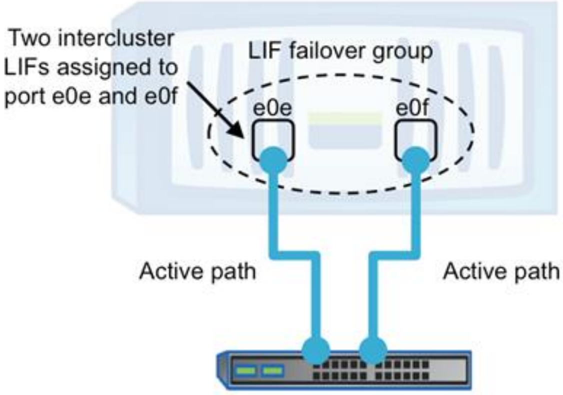
Best Practice

Assign an intercluster LIF to an intercluster-capable port and make sure that another intercluster-capable port is configured to support that connection. Make sure that the failover policy for the LIF is configured with the failover group containing the necessary ports to enable successful failover.

Multiplexing mode

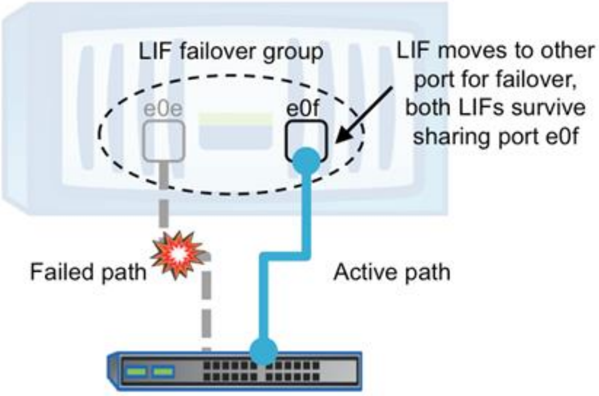
Multiplexing mode requires the configuration of additional intercluster LIFs on a node. SnapMirror uses all available intercluster LIFs on the source and destination nodes to send and receive data for all transferring SnapMirror relationships between those two nodes. If two intercluster LIFs are configured and two ports are available for intercluster communication, then one LIF can be assigned to each port, and SnapMirror simultaneously uses both ports, as shown in Figure 7:

Figure 7) Multiplexing mode.



SnapMirror multipathing with different types and speeds of networks is supported without adversely affecting replication performance on the faster ports. Communication occurs on both ports because an intercluster LIF is assigned to each port. If a port fails, the LIF that was on the failed port moves to another surviving port in that LIF's failover group. Depending on the number of ports in the failover group, multiple LIFs can now share a port.

Figure 8) LIF failover in multiplexing mode.


LIF failover group
e0e
e0f
LIF moves to other port for failover, both LIFs survive sharing port e0f
Failed path
Active path

Best Practice
Create two intercluster LIFs and assign one LIF to each port. Make sure that each LIF failover policy is configured to use the LIFs failover group, which contains the necessary ports to allow failover.

Switch-Based Link Aggregation for Multipathing

The intercluster LIF can be assigned to any kind of port in the system, including a logical port such as an ifgrp. An ifgrp supports switch-based link aggregation. Multiple physical ports can be configured into an ifgrp, and then the intercluster LIF can be assigned to that ifgrp port. The switch ports can then be combined using link-aggregation technology as a method of providing multipathing and/or redundancy.

Switch-based link aggregation does not guarantee that multiple physical paths in the ifgrp are used simultaneously. For example, assume that a single intercluster LIF is configured on both the source and destinations nodes. Therefore, each node has one IP address to use for intercluster communication and a two-port ifgrp. If the ifgrp is using an IP hash-based method of load balancing, then there is only one pair of source and destination IP addresses on which to perform the load balancing hash. The link might place all connections between these two nodes on the same path within that port group.

Keep in mind that replication can take place between multiple nodes. For example, one node might replicate different volumes to different nodes in the remote cluster. Each node has different intercluster LIFs, which have different pairs of source and destination IP addresses that enable multiple paths within the link to be used for that source node.

If switch-based link aggregation is used to allow multiple physical paths in the ifgrp to be used when replicating between two nodes, additional intercluster LIFs can be configured on either of the two nodes. ONTAP automatically establishes a connection between every LIF on the source and destination node for SnapMirror. This approach provides additional combinations of source and destination IP addresses for the load-balancing hash, which could be placed on different paths within the link. However, in this example the purpose of configuring multiple LIFs on one node is to enable multiple paths to be used for replication between any two nodes. This precaution is likely not necessary in many WAN replication scenarios because WAN bandwidth might be significantly less than the bandwidth of the combined links in the ifgrp. Enabling multiple paths between two nodes might not be beneficial, because many nodes must share the WAN bandwidth anyway.

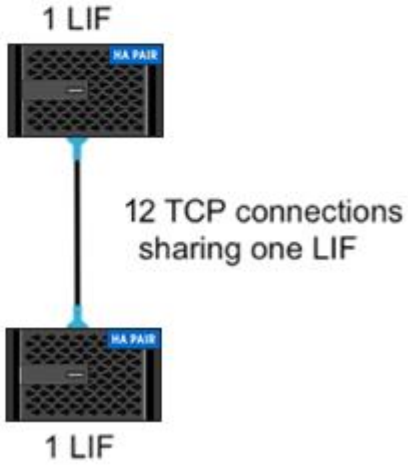
Best Practice
When using switch-based link aggregation, create the ifgrp with `multimode_lacp` mode and set the distribution function of the ifgrp to `port`. Using the port value for the distribution function configures the ifgrp to distribute connections across paths by hashing the source/destination IP address, as well as the port used. This practice does not guarantee that connections are evenly distributed across all paths in the ifgrp, but it does allow use of multiple physical links in the ifgrp.

2.4 Network Connections for Intercluster SnapMirror

In ONTAP, the number of intercluster LIFs determines the number of TCP connections established between the source and destination nodes for SnapMirror. TCP connections are not created per volume or per relationship.

ONTAP establishes at least 12 intercluster TCP connections for sending data, as shown in Figure 9. This is true even if both the source and destination nodes have only one intercluster LIF and enough connections are created so that all intercluster LIFs on both the source and destination nodes are used.

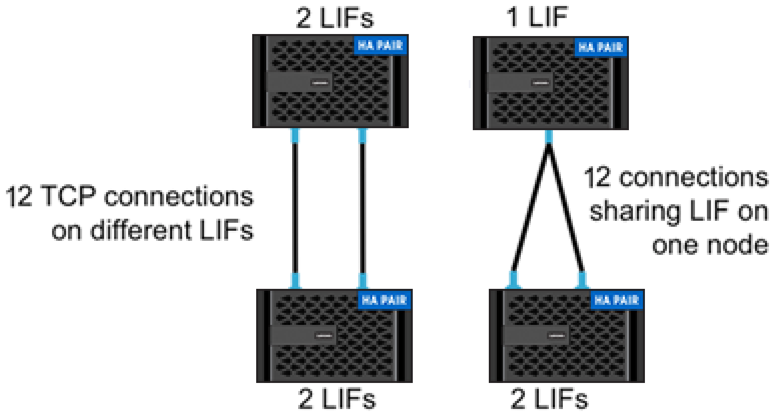
Figure 9) TCP connections with one intercluster LIF.



If the source node, destination node, or both nodes are configured with two intercluster LIFs, then ONTAP establishes 12 TCP connections for sending data. However, instead of both connections using the same LIFs, one connection uses one LIF pair, and the other connection uses the other LIF pair, as shown in Figure 10. This example shows different combinations of intercluster LIFs that produce 12 intercluster TCP connections.

Note: It is not possible to select a specific LIF pair to use for a certain TCP connection. They are managed automatically by ONTAP.

Figure 10) TCP connections with two intercluster LIFs.



Best Practice

Although it is not required, the same number of intercluster LIFs can be configured on both the source and destination nodes for operational consistency. Multiple intercluster LIFs can be created to enable active-active multipathing across multiple physical paths.

For example, if a node is configured with four 1 Gigabit Ethernet (GbE) ports for intercluster replication, then four intercluster LIFs are required, one assigned to each port to make sure all paths are used to provide bandwidth beyond just one GbE link.

2.5 Share or Dedicate Ports?

You can use dedicated ports for intercluster communication, or share ports used by the data network. Configuring intercluster LIFs to use dedicated data ports allows greater bandwidth than using shared data ports. Although configuring intercluster LIFs to share data ports enables you to use existing data ports, it does not physically isolate this network from the clients. If configured this way, the administrator should take care that routing rules or data center firewalls (external to the cluster) are set up such that general clients cannot reach the IP addresses used on the intercluster LIFs or view the intercluster traffic.

There are several configurations and requirements to consider when determining whether to share or dedicate ports for replication. They include the following:

- **LAN type.** If you have a high-speed network, such as 10GbE, 25GbE, 40GbE, and 100GbE, you might have enough local LAN bandwidth to perform replication using the same 10GbE ports used for data access. You should compare your available WAN bandwidth to your LAN bandwidth. If the available WAN bandwidth is significantly less than 10GbE, you might be limited to the network utilization that the WAN can support.
- **Available WAN bandwidth (compared to LAN bandwidth).** The WAN can act as a throttle if there is significantly less available WAN bandwidth than LAN bandwidth. If the available WAN bandwidth is significantly less than 10GbE, you might need to use dedicated ports.

  Note: The one exception to this rule might be when all or many nodes in the cluster replicate data, in which case bandwidth utilization is typically spread across nodes.
- **Replication interval.** Consider how your available bandwidth will handle the level of client activity during the replication interval. Replication during nonproduction hours might have an irrelevant effect on the data network. If replication takes place in off-peak hours, you should be able to use data ports for replication, even without a 10GbE LAN connection. However, if replication takes place during normal business hours, you need to consider the amount of data to be replicated and whether it requires so much bandwidth that it could cause contention with data protocols. If network utilization by data protocols (SMB, NFS, or iSCSI) is above 50%, then you should use dedicated ports for intercluster communication to allow for nondegraded performance if node failover occurs.
- **Change rate.** Consider the amount of data to be replicated in each interval and whether it requires so much bandwidth that it could cause contention with data protocols for shared data ports. If you use the peer relationship for replication and replication is set to occur only when minimal to no client activity occurs, you might be able to use data ports for intercluster replication successfully, even without a 10GbE LAN connection.
- **Number of ports.** If you determine that replication traffic is interfering with data traffic, you can migrate intercluster LIFs to any other intercluster-capable shared port on the same node. You can also dedicate VLAN ports for replication. The bandwidth of the port is shared between all VLANs and the base port. However dedicating ports for replication require additional switch ports and cable runs.

  Note: If you decide to dedicate ports for intercluster communication, it is a best practice to configure at least two intercluster ports per node. An intercluster LIF cannot fail over to a port on a different node; its failover group contains only intercluster-capable ports on the same node. If you use intercluster ports, ONTAP uses only intercluster ports in the failover group for an intercluster LIF. Therefore, if you use intercluster ports, you should configure at least two intercluster ports per node so that there is a port to which the intercluster LIF can fail over.

  Note: If you are not using dedicated ports, the maximum transmission unit (MTU) size of the replication network should typically be the same as the MTU size of the data network.

Best Practice
• If the network utilization generated by the data protocols (CIFS, NFS, or iSCSI) is above 50%, then you should dedicate ports for intercluster communication to allow for nondegraded performance if a node failover occurs.

- Intercluster LIFs are node scoped (they only fail over to other ports on the same node). Therefore, use a naming convention for intercluster LIFs that includes the node name followed by `ic` or `icl` for the intercluster LIF: for example, `node_name_icl#` or `node-name-ic#`, depending on your preference.
- Verify that all relevant ports have access to the necessary networks or VLANs to allow communication after port failover.
- 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–addrs` 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 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.

2.6 Firewall Requirements

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

- TCP to the IP addresses of all the intercluster LIFs over the ports 10000, 11104, and 11105. Data ONTAP uses port 11104 to manage intercluster communication sessions and port 11105 to transfer data.
- Bidirectional HTTPS between the intercluster LIFs.
- Although HTTPS is not required when you set up cluster peering using the CLI, HTTPS is required later if you use ThinkSystem Storage Manager for DM Series to configure DP.

2.7 Licensing

You must purchase and enable a SnapMirror license. If the SnapMirror source and destination are on different systems, you must enable a SnapMirror license on each system.

Note: You can use a SnapMirror license to enable SnapVault® technology without the need for a separate SnapVault license.

A DP_Optimized (DPO) license, which enables SnapMirror protection for increased numbers of volumes and peer relationships, is required on both the source and destination depending on the platform. A DPO license is also available as a tiered license or a standalone license for ONTAP secondary clusters.

Note: A DPO license allows you to scale up to 2,500 FlexVol volumes per node or 5,000 FlexVol volumes in failover mode.

3 Lenovo replication basics

3.1 SnapMirror Technology

SnapMirror replicates data from a source FlexVol or FlexGroup volume to a destination cluster by using Snapshot copies. SnapMirror performs the following operations:

1. A Snapshot copy of the data on the source is 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 update.
3. The destination is updated to reflect incremental changes on the source according to the schedule you specify.

When a SnapMirror relationship is established, the destination volume is an identical replica of the source, including snapshots, volume settings, and ONTAP space efficiency features. Breaking the SnapMirror relationship makes the destination volume 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 allow the data changed at the failover site to be efficiently resynchronized back to the primary system when it comes back online. The original SnapMirror relationship can then be re-established.

SnapMirror can use either of the replication engines to create replicas. While 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. In other words, BRE uses knowledge of the file system to determine differences between snapshots at the block-allocation level and replicates only those changed blocks. Therefore, the copy of data created on the destination has an identical structure of physical block pointers to the original data set on the source. BRE replicates volumes using volume block (VVBN) read and write operations. The SnapMirror relationship is created with `-type DP` using the SnapMirror policy type `async-mirror`.
- **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 FlexVol. This number 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 which require you to parse down 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 you to apply storage efficiencies on the secondary, independent of the primary storage settings.

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, you can replicate data between a storage system running an older version of ONTAP and a storage system running a newer version. This approach allows for 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 extended data protection (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 SnapCenter® technology 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.

Load-Sharing Mirror

Every SVM in a NAS environment has a unique namespace. The SVM root volume, containing the operating system and related information, is the entry point to the namespace hierarchy. To make sure that data remains accessible to clients in the event of a node outage or failover, you should create a load-sharing mirror copy of the SVM root volume. This step is not suitable for a MetroCluster environment.

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

Note: SnapMirror LS mirrors are only capable of supporting NAS (CIFS/NFSv3). LS mirrors do not support NFSv4 clients or SAN client protocol connections (FC, FCoE, or iSCSI). However, you can use NFSv4 and LS mirrors in the same environment. NFSv4 never uses an LS mirror; instead, it always uses the source volume.

Best Practice
Create a load-sharing mirror of an SVM root volume only on the partner node in the HA-pair of the cluster.

For additional information concerning LS mirrors, see “Managing SnapMirror Root Volume Replication” in the ONTAP 9 Documentation Center.

3.2 SnapVault Technology

SnapVault is an archiving technology designed for disk-to-disk Snapshot copy replication for standard compliance and other governance-related purposes. In contrast to SnapMirror, in which the destination usually contains only the Snapshot copies currently in the source volume, SnapVault typically retains point-in-time Snapshot copies created over a much longer period. You might want to keep monthly Snapshot copies of your data over a 20-year span, for example, to comply with government accounting regulations for your business. Since there is no requirement to serve data from vault storage, you can use slower, less expensive disks on the destination system. SnapVault technology also uses LRSE which provides the flexibility of version-independent backup replication referred to as extended data protection (XDP) relationship, combined with a policy type of `vault` and policy `XDPDefault` (predefined). A SnapMirror license must be installed on both source and destination clusters to enable the SnapVault feature on your Lenovo system.

Backing up volumes to a SnapVault backup requires the following steps:

1. **Start the baseline transfer.** As with SnapMirror, SnapVault performs a baseline transfer the first time you invoke it. The SnapMirror policy for the relationship defines the contents of the baseline and any updates. Baseline transfer under the default SnapVault policy (`XDPDefault`) makes a Snapshot copy of the source volume, and then transfers that copy and the data blocks it references to the destination volume. Unlike SnapMirror, SnapVault does not include older Snapshot copies in the baseline.
2. **Perform scheduled incremental transfers.** Updates are asynchronous, following the schedule you configure. The rules you define in the policy for the relationship identify which new Snapshot copies to include in updates and how many copies to retain. The labels defined in the policy (monthly, for example) must match one or more labels defined in the Snapshot policy on the source. Otherwise, replication fails. At each update under the `XDPDefault` policy, SnapMirror transfers the Snapshot

copies that were made since the last update, provided they have labels matching the labels defined in the policy rules.

3. **Update the SnapVault common Snapshot copy.** At the end of each Snapshot copy transfer session, which can include transferring multiple Snapshot copies, the most recent incremental Snapshot copy in the SnapVault backup is used to establish a new common base between the primary and secondary volumes and is exported as the active file system.
4. **Restore data upon request.** If data must be restored to the primary volume or to a new volume, the SnapVault secondary transfers the specified data from the SnapVault backup.

3.3 Unified Data Protection

SnapMirror Unified Replication allows you to protect mission-critical business data with simple, efficient replication by bringing together the powerful capabilities of SnapMirror with the same (unified) logical replication engine as SnapVault technology, for the purpose of DR and archiving to the same destination. A unified relationship type is designated as XDP and provides single-baseline functionality, drastically reducing storage and network bandwidth, which translates immediately into cost savings. The major benefits for SnapMirror Unified Replication are as follows:

- Only one baseline copy of a volume is needed to the secondary storage (without Unified Replication, SnapMirror and SnapVault each need their own baseline copy).
- 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 of different ONTAP releases. Previously, you could only replicate from higher to lower releases causing operational complexities. With Unified Replication, you can replicate from lower to higher and higher to lower if both sides are ONTAP 9.4 or higher.
- To avoid corrupting replication from primary to secondary, Unified Replication makes it possible to recover the primary volume from available Snapshot copies.

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 `MirrorAndVault`. The policy can always be modified to include custom rules for backing up specific application-consistent Snapshot copies. XDP removes the limitation of the destination controller requiring an ONTAP major version number equal to or higher than the major version of the source controller, which enables nondisruptive upgrades. In addition, this functionality reduces the number of secondary Snapshot copies needed on the destination.

The following example shows how Unified Replication can be configured with the `MirrorAllSnapshots` policy from the CLI:

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

Best Practice

- You must 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.

4 SnapMirror Configuration

4.1 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 you set up cluster peering, you should 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 on the local cluster must be able to communicate with every intercluster LIF on the remote cluster.
- 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.
- The intercluster network must provide connectivity among all intercluster LIFs on all nodes in the cluster peers. Every intercluster LIF on every node in a cluster must be able to connect to every intercluster LIF on every node in the peer cluster.

Note: For more information about cluster peering requirements, see the Cluster Administration Guide in the ONTAP 9 documentation.

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 two 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 other cluster. The other 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. A cluster can be in a peer relationship with up to 256 clusters, allowing multiple clusters to replicate amongst themselves.

Beginning in ONTAP 9.4, you can use the `-generate-passphrase` feature to create a peer relationship with a cluster whose intercluster LIF IP addresses you do not know in advance. This eliminates the need for the initiating cluster to authenticate itself with the remote cluster.

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 `vserver services dns hosts` file of the destination system and vice versa, or they must be resolvable through the DNS.

4.2 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.

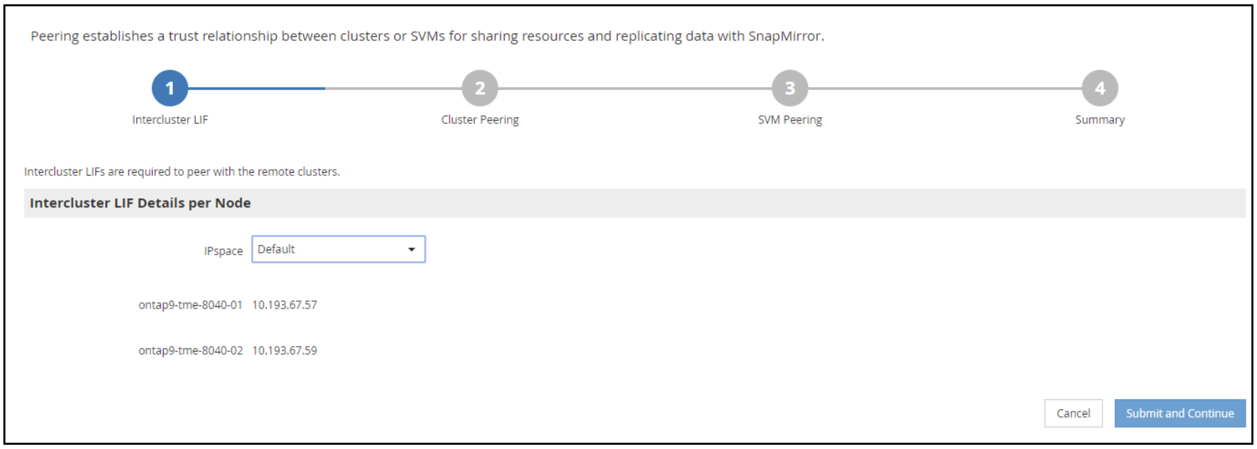
Best Practice

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 additional information regarding SVM peering, see the Cluster and SVM Peering Power Guide and the Cluster Administration Guide in the ONTAP 9 documentation.

In ThinkSystem Storage Manager for DM Series for ONTAP 9.4 and later, peering establishes a trust relationship between clusters or SVMs as shown in Figure 11) SVM peering.:

Figure 11) SVM peering.



Peering establishes a trust relationship between clusters or SVMs for sharing resources and replicating data with SnapMirror.

Intercluster LIFs are required to peer with the remote clusters.

Intercluster LIF Details per Node

IPspace: Default

ontap9-tme-8040-01 10.193.67.57

ontap9-tme-8040-02 10.193.67.59

4.3 SnapMirror Data Protection Relationship

A relationship created between the source (for example, a FlexVol or FlexGroup volume) in primary storage and the destination in secondary storage is called a data protection relationship. The creation of SnapMirror relationship does not depend on SVM host name to IP address resolution. However, the cluster names are resolved through the 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 a 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.

ONTAP 9.7 introduces a new-look and feel for ThinkSystem Storage Manager for DM Series. Storage Manager is designed to simplify day-to-day operations by providing a more logical layout and reducing the number of clicks needed to perform tasks.

SnapMirror relationships have the following characteristics:

- SnapMirror relationships are created and managed on the destination cluster.
- SnapMirror relationship transfers are triggered by the scheduler in the destination cluster.
- A destination volume must be created with the volume type (`-type` option) of DP for SnapMirror initialization to succeed. You cannot change the volume `-type` property after the volume has been created.
- The destination volumes in a SnapMirror relationship are read-only until failover.
- You can failover to the secondary copy with the SnapMirror break operation, making the destination volume writable. The SnapMirror break must be performed separately for each volume.
- The destination volumes can be mounted into an SVM namespace while still read-only, but only after the initial transfer is complete.
- A destination volume in a SnapMirror relationship configured between two clusters cannot be mounted in the same namespace as the source volume, because intercluster relationships are to a different cluster and therefore to a different SVM, which is a different namespace. However, the destination volume 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 DP mirror destination volumes can be mapped to igroups and connected to clients. However, the client must be able to support connection to a read-only LUN.
- DP mirror relationships can be managed using the ONTAP CLI, ThinkSystem Storage Manager for DM Series, and ThinkSystem Intelligent Monitoring 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 peer relationship.
- Create a destination SVM that has the same language type as the source SVM. Volumes cannot exist in ONTAP without an SVM.
- 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 a peer relationship.
- The destination aggregate must have available space.
- Both clusters must be configured and set up appropriately to meet the requirements of your environment for user access, authentication, and client access.
- Create a destination volume of -type `dp`, with a size equal to or greater than that of the source volume.
- 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 must be created.
- Assign a policy (default 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 Practice

- 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 autogrow changes for the DP FlexGroup volume, make sure to specify that the total size of the DP 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. Also, the ordering of the aggregates on the `aggr-list` directly relates to the order in which the constituents get created.
- If you expand the primary FlexGroup, make sure to also expand the secondary DP FlexGroup to avoid failure of the next scheduled SnapMirror operation. Before performing the expand operation, always quiesce the SnapMirror relationship.
- 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 runs out of space.
- To delete a constituent in a primary FlexGroup volume, quiesce the SnapMirror relationship, and then perform the SnapMirror release command. Also make sure that the corresponding constituent from the secondary FlexGroup volume is also deleted before performing a SnapMirror resync.

Fan-Out and Fan-In

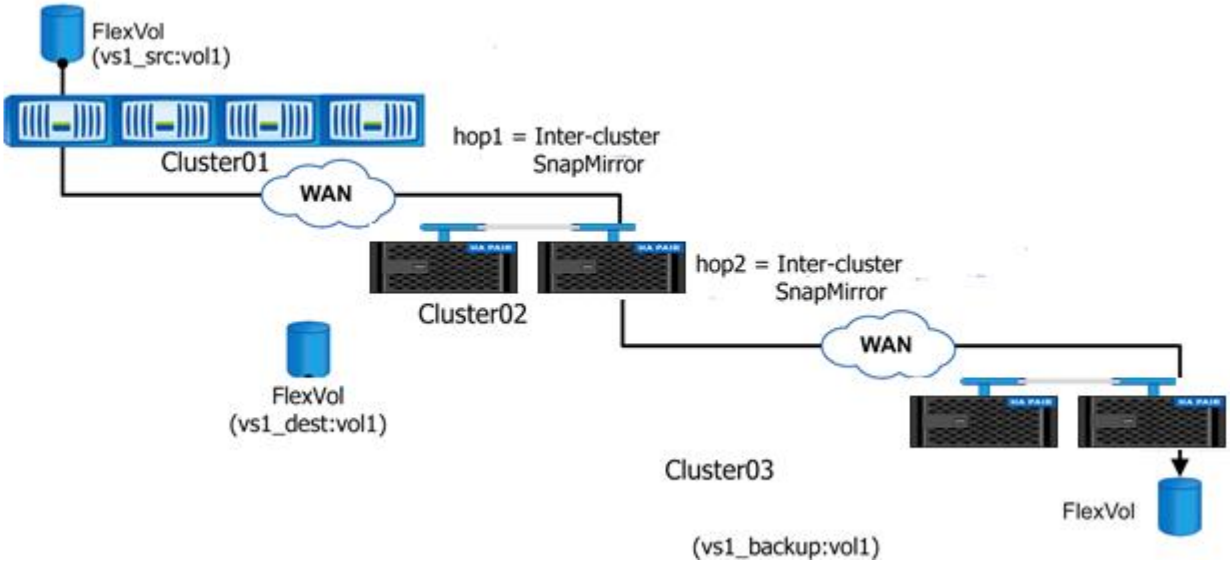
It is possible to fan out or fan in volumes between different SVMs. For example, different volumes in a single SVM in the source cluster can be replicated into different SVMs in the destination cluster; this is referred to as fan-out. Alternatively, multiple different volumes in different SVMs in the source cluster can also be replicated to a single SVM in the destination cluster; this is referred to as fan-in.

Note: You can fan out a maximum of eight destination volumes from a single source volume.

Cascade Relationship

You can replicate data from a SnapMirror destination to another system using SnapMirror. Therefore, a system that is a destination for one SnapMirror relationship can act as the source for another SnapMirror relationship. This is useful when you need to copy data from one site to many sites. Instead of replicating data from a single source to each of the destinations, you can replicate data from one destination to another destination in a series. This is referred to as cascading. SnapMirror supports mirror-mirror, mirror-vault, vault-mirror, and vault-vault cascade deployments. In a cascade topology, you need only create intercluster networks between the primary and secondary clusters and between the secondary and tertiary clusters. You need not create an intercluster network between the primary and the tertiary cluster. An example cascade configuration with two hops is shown in Figure 12) SnapMirror cascade..

Figure 12) 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).
- The destination system carries an extra Snapshot copy.

Best Practice

Make sure that all the legs of the relationship complete successfully to make sure that your SnapMirror update does not fail with Snapmirror busy error

If you use a combination mirror-vault, fan-out, or cascade deployment, you should keep in mind that updates fail if a common Snapshot copy does not exist on the source and destination volumes. You can 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.

Cascade support enables the following:

1. 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 (currently 255).
2. A backup admin can tier multiple backup copies. More frequent Snapshot copies (say, daily and weekly) can be retained on secondary storage (B), whereas monthly and yearly Snapshot copies can be retained on tertiary storage (C).
3. Data can be moved from SSDs to SATA disk drives.
4. 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: Fan-in, fan-out, and cascade configurations are not currently supported with FlexGroup volumes.

Dual-Hop Volume SnapMirror

This configuration involves volume SnapMirror replication among three clusters, which consists of a chain of relationships in which a source volume is mirrored to a secondary volume, and the secondary volume is mirrored to a tertiary volume. If the secondary volume becomes unavailable, you can synchronize the relationship between the primary and tertiary volumes without performing a new baseline transfer.

```
vs1_src:vol1 > vs1_dest:vol1 > vs1_backup:vol1
```

In the preceding configuration, `vs1_src:vol1` to `vs1_dest:vol1` and `vs1_dest:vol1` to `vs1_backup:vol1` transfers can occur at the same time.

4.4 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 that Snapshot copy and any new Snapshot copies that have labels matching the labels defined in the Snapshot policy rules based on the SnapMirror policy. ONTAP comes with several predefined protection policies.

The following policy variants are available for SnapMirror:

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

```
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
 Rules:
 SnapMirror Label              Keep Preserve Warn Schedule Prefix
 ----------------------------- ---- -------- ---- -------- ----------
 sm_created                       1 false       0 -        -
 all_source_snapshots             1 false       0 -        -
```

In this example, the policy type is set to `async-mirror`, and the Create Snapshot value is set to true. Rules include `sm_created` to allow crash-consistent Snapshot copy creation on the primary and `all_source_snapshots` to specify that all Snapshot copies must be transferred with a retention policy of 1.

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

- **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.

```
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
                       Rules: SnapMirror Label     Keep  Preserve Warn Schedule Prefix
                              ----------------     ----  -------- ---- -------- ------
                              sm_created              1  false       0 -        -
                              all_source_snapshots    1  false       0 -        -
```

- **MirrorLatest.** This also 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.

```
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
 Rules:
 SnapMirror Label              Keep Preserve Warn Schedule Prefix
 ----------------------------- ---- -------- ---- -------- ----------
 sm_created                       1 false       0 -        -
```

- **MirrorAndVault**. This is a unified SnapMirror and SnapVault 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.

```
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
 Rules:
 SnapMirror Label              Keep Preserve Warn Schedule Prefix
 ----------------------------- ---- -------- ---- -------- ----------
 sm_created                       1 false       0 -        -
 daily                            7 false       0 -        -
 weekly                          52 false       0 -        -
```

For this example:

- The policy type is `mirror-vault`.
- Create snapshot is set to `true`.
- There are three rules. `sm_created` creates a Snapshot copy and replicate any differences between the copy and the previous SnapMirror snapshot. The rules `daily` and `weekly` keep one week of daily snapshots and one year of weekly snapshots of your protected volume at the destination.

- **Unified7year.** In this policy, you have the additional rule of `monthly` to transfer monthly Snapshot copies and retains them for seven years.

```
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
 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.**

```
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
 Rules:
 SnapMirror Label              Keep Preserve Warn Schedule Prefix
 ----------------------------- ---- -------- ---- -------- ----------
 daily                            7 false       0 -        -
 weekly                          52 false       0 -        -
```

For this example:

- 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.
- Two rules, `daily` and `weekly`, replicate any existing Snapshot copies that have names matching the patterns in the “SnapMirror Label”, retaining 7 and 52 snapshots respectively.

4.5 SnapMirror Schedules

A SnapMirror policy requires at least one SnapMirror job schedule that can be created to run periodic asynchronous replication by assigning a schedule to a SnapMirror relationship in the destination cluster.

Create a schedule through the command line using the `job schedule cron create` command. 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).

```
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 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
```

4.6 Create a SnapMirror Relationship

You must create a SnapMirror relationship between the source on one cluster and the destination on the peered cluster to enable data protection for replicating data for DR. From ThinkSystem Storage Manager for DM Series on the source cluster, complete the following steps:

1. Click Storage > Volumes.
2. Select the volume for which you want to create a mirror relationship, and then click Actions > Protect.

Note: The relationship type and the policy selected affects what is transferred in the relationship.

3. After the relationship is created and initialized, verify that the relationship status of the SnapMirror relationship is in the Snapmirrored state.
4. Depending on the Storage Manager version that you are running, perform one of the following steps:
   a. For ONTAP 9.4, click Protection > Relationships.
   b. Starting with ONTAP 9.5, click Protection > Volume Relationships.
5. Select the SnapMirror relationship between the source and the destination volumes, and then verify the status in the Details tab.
6. The Details tab displays the health status of the SnapMirror relationship and shows the transfer errors and lag time.
7. The Is Healthy field must display Yes.
8. For most SnapMirror data transfer failures, the field displays No. In some failure cases, however, the field continues to display Yes. You must check the transfer errors in the Details section to verify that no data transfer failure occurred.
9. The Relationship State field must display Snapmirrored.
10. 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.
11. Also, navigate to the Volumes window, and then select the volume for which you created the SnapMirror relationship. Double-click the volume to view the volume details and the data protection status.

4.7 Baseline Transfer during Initialization of SnapMirror Relationship

When a new SnapMirror relationship is created, it establishes the relationship and the metadata that defines it. You can 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. The process is as follows:

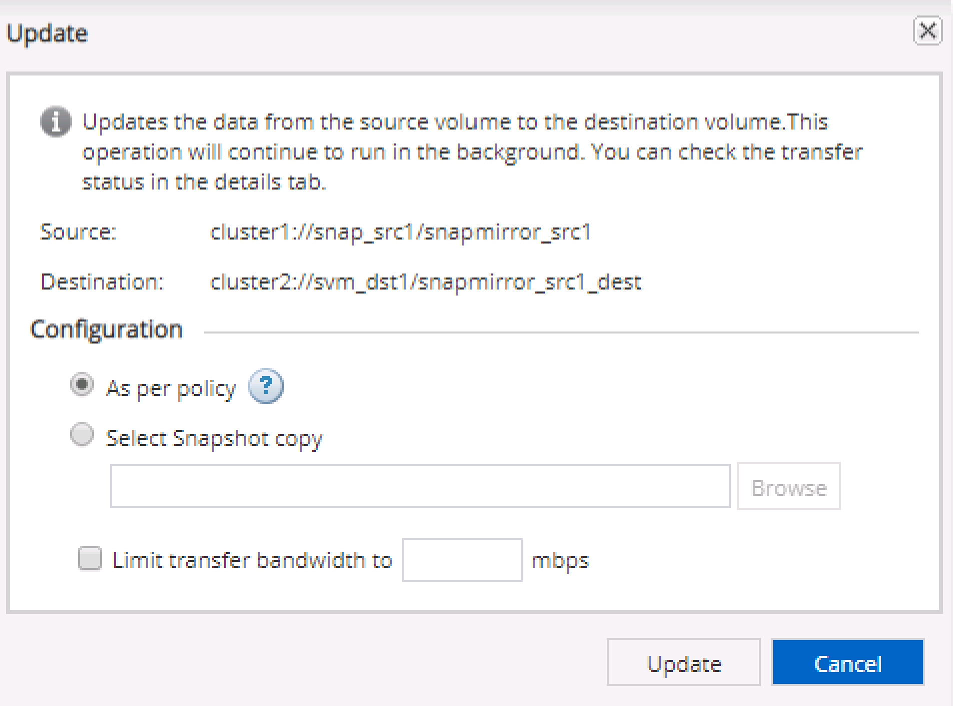
1. Make a Snapshot copy of the source.
2. Transfer the Snapshot copy and all the data blocks it references to the destination.
3. Depending on the SnapMirror policy attached to the relationship, it also transfers other Snapshot copies from the source to the destination.
4. After baseline transfer, updates to this relationship occur according to the schedule assigned to the SnapMirror relationship.

The destination is a volume that you have already created and marked restricted. After SnapMirror finishes transferring the data, it brings the destination 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 volume becomes valid and goes online after the initial transfer is complete. Now, the files and Snapshot copies in the source volume should be available on the destination.

4.8 Manual Update to the SnapMirror Relationship

You might need a manual update the SnapMirror relationship 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 13) Update.



Update

Updates the data from the source volume to the destination volume.This operation will continue to run in the background. You can check the transfer status in the details tab.

Source: cluster1://snap_src1/snapmirror_src1

Destination: cluster2://svm_dst1/snapmirror_src1_dest

Configuration

● As per policy

○ Select Snapshot copy

□ Limit transfer bandwidth to mbps

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

5 Conversion

5.1 Convert a SnapMirror Relationship to a SnapVault Relationship

One scenario in which you would want to convert an existing SnapMirror relationship to a SnapVault relationship is when you want a DR copy to be changed to long-term retention without another baseline. To perform this conversion, complete the following high-level steps:

1. Break the SnapMirror relationship on the destination cluster.
2. Delete this SnapMirror relationship.
3. Create a SnapVault relationship between the same endpoints.
4. 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. View the status of the existing SnapMirror relationship.

```
Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:SMSource
            DP   svm_dst1:SMSource_dest
                              Uninitialized
                                      Idle           -         true    -
1 entries were displayed.
```

2. Initialize the SnapMirror relationship.

```
Remote::> snapmirror initialize -destination-path svm_dst1:SMSource_dest
Operation is queued: snapmirror initialize of destination svm_dst1:SMSource_dest.

Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:SMSource
            DP   svm_dst1:SMSource_dest
                              Snapmirrored
                                      Idle           -         true    -
1 entries were displayed.
```

3. The relationship on the destination cluster can be viewed in ThinkSystem Storage Manager for DM Series.
4. Convert SnapMirror to SnapVault by first performing a SnapMirror break operation.

```
Remote::> snapmirror break -destination-path svm_dst1:SMSource_dest
[Job 128] Job succeeded: SnapMirror Break Succeeded
```

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

As a result of the break operation, the destination volume changes from `dp` to `rw`.

5. Run the `SnapMirror Delete` command.

```
Remote::> snapmirror delete -destination-path svm_dst1:SMSource_dest -relationship-info-only true
Operation succeeded: snapmirror delete the relationship with destination svm_dst1:SMSource_dest.
```

If performed from ThinkSystem Storage Manager for DM Series, the release operation is included in this operation by deleting the relationship without releasing the base snapshot copies to be deleted. This is done because you want these base Snapshot copies so that another baseline is not required when you create the SnapVault relationship.

6. Uncheck OK to Release the Base Snapshot Copies of the Relationship.
7. Create the SnapVault relationship.

```
Remote::> snapmirror create -source-path snap_src1:SMSource -destination-path
svm_dst1:SMSource_dest -type XDP
Operation succeeded: snapmirror create the relationship with destination svm_dst1:SMSource_dest.
```

The Relationship Type is shown to be Vault and the Relationship Status is Broken Off.

```
Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:SMSource
            XDP  svm_dst1:SMSource_dest
                              Broken-off
                                      Idle           -         true    -
```

8. Perform a SnapMirror resync operation.

```
Remote::> snapmirror resync -destination-path svm_dst1:SMSource_dest

Warning: All data newer than Snapshot copy
         snapmirror.3fd9730b-8192-11e2-9caa-123478563412_2147484699.2013-02-28_1
         10732 on volume svm_dst1:SMSource_dest will be deleted.
         Verify there is no XDP relationship whose source volume is
         "vs1R:vol1_vs1R".  If such a relationship exists then you are creating
         an unsupported XDP to XDP cascade.
Do you want to continue? {y|n}: y
[Job 133] Job succeeded: SnapMirror Resync Transfer Queued
```

9. Run the `snapmirror show` command to view the relationship type and status.

```
Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:SMSource
            XDP  svm_dst1:SMSource_dest
                              Snapmirrored
                                      Idle           -         true    -
```

The Relationship Type is shown as Vault, and the Relationship Status is Snapmirrored.

The SnapVault destination volume is of type `dp` for use as a DR volume.

5.2 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. Complete the following steps for the conversion:

1. Break the mirror relationship from the destination cluster.
2. Delete the mirror (DR) relationship.
3. Create a unified relationship (mirror and 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 do a re-baseline.

The following steps present this process in detail:

1. View the status of the SnapMirror relationship.

```

Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:Source
            DP   svm_dst1:Source_dest
                              Uninitialized
                                      Idle           -         true    -
1 entries were displayed.
```

The SnapMirror relationship is shown as Uninitialized.

2. Initialize the SnapMirror relationship.

```
Remote::> snapmirror initialize -destination-path svm_dst1:Source_dest
Operation is queued: snapmirror initialize of destination svm_dst1:Source_dest.

Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:Source
            DP   svm_dst1:Source_dest
                              Snapmirrored
                                      Idle           -         true    -
1 entries were displayed.
```

The SnapMirror relationship status is shown as Snapmirrored.

3. To convert to SnapMirror Unified Replication, perform a SnapMirror Break operation.

```
Remote::> snapmirror break -destination-path svm_dst1:Source_dest
[Job 128] Job succeeded: SnapMirror Break Succeeded
```

The SnapMirror relationship is shown to be Broken Off.

4. Run the `snapmirror delete` command.

```
Remote::> snapmirror delete -destination-path svm_dst1:Source_dest -relationship-info-only true
Operation succeeded: snapmirror delete the relationship with destination svm_dst1:Source_dest.
```

5. Delete the SnapMirror relationship without releasing the base Snapshot copies.

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

6. Create the Unified Replication relationship by running the `snapmirror create` command.

```
Remote::> snapmirror create -source-path snap_src1:Source -destination-path svm_dst1:Source_dest
-type XDP –policy MirrorAndVault
Operation succeeded: snapmirror create the relationship with destination svm_dst1:Source_dest.
```

7. Run the `snapmirror show` command to see the relationships created.

```
Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:Source
            XDP  svm_dst1:Source_dest
                              Broken-off
                                      Idle           -         true    -
```

The destination volume is still of type `rw`.

Run the `snapmirror resync` command.

```
Remote::> snapmirror resync -destination-path svm_dst1:Source_dest

Warning: All data newer than Snapshot copy snapmirror.12ceb7f0-b078-11e8-baec-
005056b013db_2160175149.2020-01-24_091316 on volume
         svm_dst1:Source_dest will be deleted.
Do you want to continue? {y|n}: y
Operation is queued: initiate snapmirror resync to destination "svm_dst1:Source_dest".
```

8. Run the `snapmirror show` command to verify the relationship status.

```
Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship   Total             Last
Path        Type  Path         State   Status         Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
snap_src1:Source
            XDP  svm_dst1:Source_dest
                              Snapmirrored
                                      Idle           -         true    -
```

The SnapMirror relationship is now of type `Mirror and Vault`.

The destination volume changes from type `rw` to type `dp`.

Note: The SnapMirror unified replication relationship cannot be converted back to a default SnapMirror relationship unless a rebaseline is created to a new destination volume.

6 SnapMirror and ONTAP Feature Interaction

6.1 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 that Snapshot copy is then compared to the previous SnapMirror Snapshot copy that was replicated. All the delta of 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. You do not have to perform a full baseline if other common Snapshot copies between the two volumes still exist on the volumes.

In 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 of the resync.

```
cluster02::> snapmirror resync -source-path cluster01://vs1/vol1 -destination-path
cluster02://vs2/vol1
Warning: All data newer than Snapshot copy hourly.2011-12-06_1805 on volume cluster02://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.
```

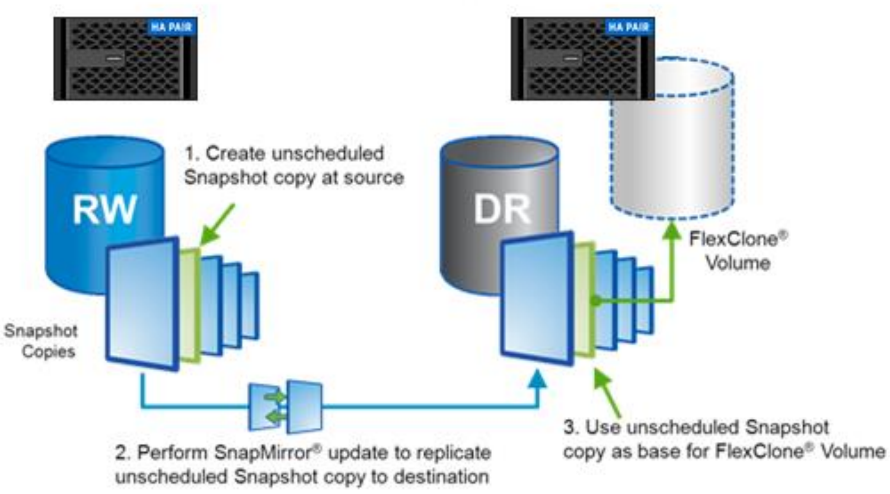
6.2 SnapMirror and Qtrees

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 at the volume level only.

6.3 SnapMirror and FlexClone

FlexClone technology makes it possible to create a writable volume from a read-only SnapMirror destination 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 14) Creating a FlexClone volume at a SnapMirror destination. illustrates the creation of a FlexClone volume at the SnapMirror destination.

Figure 14) Creating a FlexClone volume at a SnapMirror destination.


HA PAIR
HA PAIR
RW
DR
1. Create unscheduled Snapshot copy at source
Snapshot Copies
FlexClone® Volume
2. Perform SnapMirror® update to replicate unscheduled Snapshot copy to destination
3. Use unscheduled Snapshot copy as base for FlexClone® Volume

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 cannot be removed from the destination—for example, if the Snapshot copy is locked because it is the base Snapshot copy of a FlexClone volume—then the SnapMirror update fails. 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, create the 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 14) Creating a FlexClone volume at a SnapMirror destination.. 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.

6.4 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 still using legacy SnapMirror (of type `-dp`), you cannot enable different storage efficiency configurations for 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. You should start SnapMirror transfer 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.

6.5 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 that is 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 in order 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 in ONTAP 9.7, see the ONTAP documentation on this subject.

6.6 SnapMirror for Disk Shelf Failure Protection

If you have decided that you want to use SnapMirror to protect against disk shelf failure, you need to be aware of two things:

- You cannot mirror the volumes to be in the same HA pair.
- The configuration does not automatically fail over.

You can mirror the volumes to different nodes in a different HA pair on the same cluster. Mirroring to a different node makes sure that the other volume is always in a different shelf. If you try to mirror to a different shelf on the same node, then the mirror must be on a different aggregate. However, there is still the risk that an aggregate might have a disk in any shelf. Even if you try to configure otherwise (keeping aggregates on their own shelves), that can change because drives fail, and spares are used. This configuration avoids having a single point of failure and provides protection against disk shelf failure. The caveat here is that the configuration does not fail over automatically. You must manually break the SnapMirror relationship, unmount the clients, remount the clients on the destination volumes, and change the NFS export policies.

6.7 SnapMirror and Volume Autosize

The destination volume must be the same size as or larger than the source volume. SnapMirror updates fail if the destination volume is smaller than the source volume.

Best Practice
Keep the source and destination volumes the same size or slightly larger with the Auto Grow option enabled on the destination volume.

Note: If the source volume size is automatically increased by the volume autosize feature, or if it is manually increased, then the destination volume size must be increased to match the size of the source volume.

If the autosize feature increases the size of the source volume, to avoid having to manually resize the destination volume, size the destination volume so that it is at least as large as the source volume's maximum autosize value. To eliminate the need for the additional capacity required to guarantee the larger destination volume, the space guarantee can be disabled on the destination. However, keep in mind that the capacity of the destination system must be properly managed so that there is room for operations that generate data on the destination system.

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

6.8 SnapMirror and Network Data Management Protocol

Network Data Management Protocol (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 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 using the volume move capability for performance or capacity reasons. When a volume is moved to a different node, the NDMP backup job must be reconfigured to back up the volume from the new location. If backups are performed from the SnapMirror destination volume, these volumes are less likely to require a move. Therefore, it is less likely that the NDMP backup jobs need to be reconfigured.

6.9 SnapMirror and Fabric Pool

The SnapMirror replication interval should be set to a lower value that the Fabric Pool tiering policy to make sure that all data is protected. Fabric Pool alone does not represent a data protection strategy.

7 Performance

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

- Load on the system because CPU cycles are used for replication of this data as well.
- The number of transfers being attempted simultaneously—that is, the total number of SnapMirror transfers occurring at the same time. The fewer transfers occurring at a given time, the more resources and bandwidth are available to allow them to complete faster.
- The type of transfer: initialization or update.
- System configuration, including the storage system type, the disk types, the number of disks in the aggregate, or the number of volumes in the aggregate.

7.1 Calculate SnapMirror and SnapVault 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: 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: 9.4 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
```

7.2 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 recovery point objective (RPO). Otherwise, replication times increase as the amount of data sent over the network to the DR site increases. Differently put, if you do not want to or cannot increase the network bandwidth, you need to lower the replication frequency that is causing larger RPO values and thus increasing your exposure to larger data loss.

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

Maintaining the Same RPO Level

- **Challenge.** Your data replication needs are growing. You need more bandwidth 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.** You are using all your network bandwidth. However, your customer wants to reduce their exposure to data loss and improve their RPO.
- **Solution.** By using network compression, you can improve your RPO without purchasing more network bandwidth.

Use the Network Bandwidth for Other Purposes

- **Challenge.** Your replication is consuming all your bandwidth. You want to use the network bandwidth for other purposes such as client access or applications without purchasing additional 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.

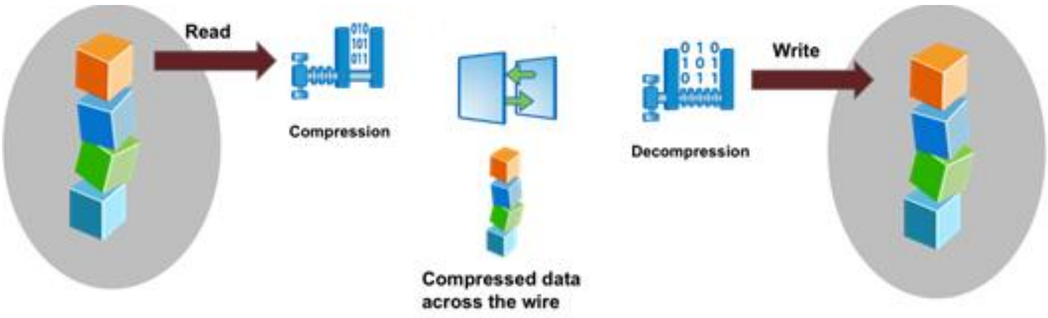
Speeding Up the Initial Transfers

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

What Is SnapMirror Network Compression?

Network compression is natively built into SnapMirror to enable data compression 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 15) SnapMirror network compression functional diagram. shows a very high-level flow of SnapMirror network compression.

Figure 15) 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 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, you must first abort the transfer, 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, you need to 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: -
```

Compression ratio is only shown in transferring state.

```
                     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: 9.4 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
```

7.3 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 using the `snapmirror modify` command.

```
cluster02::> snapmirror modify -destination-path cluster02://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 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.

Beginning from ONTAP 9.4, ONTAP 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 and SnapVault transfers. The restriction is enforced cluster wide on all nodes in the cluster. This capablility is in addition to the throttle for each SnapMirror relationship as described above. 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 and SnapVault 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 4 KBps, and the maximum can be up to 2 TBps. 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.
```

7.4 How to Change TCP Receive Buffer Size

SnapMirror uses the network service `ctlopcp` with a tunable TCP receive buffer window for both intercluster (WAN network) and intracluster (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 1) TCP receive buffer windows.

	Default	Min	Max
Intercluster TCP receive buffer window	2MB	256KB	7MB
Intracluster TCP receive buffer window	256KB	256KB	7MB

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

Note: Increasing the intercluster TCP receive buffer can increase the effective throughput when working with high capacity WAN connections. However, this is not typically necessary due to the increased number of active TCP streams used for replication.

The command to change the intercluster TCP receive buffer window can be done only in the diagnostics mode (`set diag`).

```
::*> network connections options buffer show
Service      Layer 4 Protocol  Network  Receive Buffer Size (KB) Auto-Tune?
-----------  ----------------  -------  ------------------------ ----------
ctlopcp      TCP               WAN      2048                     true
ctlopcp      TCP               LAN      256                      false
2 entries were displayed.

::*> network connections options buffer modify -receive-buffer 7168 -network WAN -protocol
tcp -service ctlopcp
```

```
::*> network connections options buffer show
Service      Layer 4 Protocol  Network  Receive Buffer Size (KB) Auto-Tune?
-----------  ----------------  -------  ------------------------ ----------
ctlopcp      TCP               WAN      7168                     true
ctlopcp      TCP               LAN      256                      false
2 entries were displayed.

::*> network connections options buffer modify
Usage:
   [-service] <protocol service>       *Protocol Service
   [-protocol] {UDP|TCP}               *Layer 4 Protocol
   [-network] {WAN|LAN}                *Network Type
  [[-receive-buffer] {256..7168}]      *Size of Receive Buffer (KB)
  [ -receive-auto-tune {true|false} ]  *Automatically Size Receive Buffer
```

7.5 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, see the ONTAP 9 Data Protection Power Guide.

Best Practice

- 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 rather than placing all volumes requiring replication on a single node. This best practice allows all nodes in the cluster to share replication activity.

7.6 Recommended 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. Therefore, while the Lenovo scheduler supports creating schedules that run every minute, Lenovo does not recommend performing a SnapMirror update operation every minute.

7.7 Network Sizing Requirements

When deploying SnapMirror, you must 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 impacted. 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 recovery point objective (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.

8 Interoperability

You can always refer to Lenovo Press 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 for your specific environment. The Interoperability Matrix 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. You should verify that the source and destination volumes are running compatible ONTAP versions before creating a SnapMirror data protection relationship.

9 Troubleshooting Tips

9.1 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
```

9.2 Troubleshooting SVM Peer Relationships

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

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

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

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

```
cluster02::> vserver peer show-all
            Peer        Peer                                      Peering
Vserver     Vserver     State        Peer Cluster                 Applications
----------- ----------- ------------ -------------------------- ---------------
```

```
vs1_dest    vs1_backup  peered       cluster03                  snapmirror
vs1_dest    vs1_src     peered       cluster01                  snapmirror
```

9.3 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, in the cluster element manager web interface, and as the Healthy column in the displayed status of SnapMirror relationships in Storage Manager.

In this example, the `snapmirror show` command displays the Healthy column.

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

The Healthy column displays the health of the SnapMirror relationship. It also indicates whether the RPO is maintained without needing to determine the age of the last update in order to interpret the relationship's health. For example, the Healthy column displays `true` for a SnapMirror relationship scheduled for regular updates if the last update completed before a following update attempted to start, as shown in the first relationship in the output presented in this example.

If a scheduled update is in progress when the next scheduled update begins, the Healthy column displays `false` for that relationship. Additionally, if the previously scheduled or manual update fails, then the Healthy column also displays `false` for that relationship.

If a transfer is currently in progress, the Healthy column displays – and the Total Progress column displays the amount of progress for the currently running transfer.

The Healthy column also displays - when the relationship is in an uninitialized state, as shown in the third relationship. It also displays - if the relationship is in a broken state because the `snapmirror break` command was used.

The Healthy column displays – for the relationship on the source system. To view authoritative information about the health of a SnapMirror relationship, look at that relationship from the destination.

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

9.4 Troubleshooting SnapMirror Relationships

To determine when the last SnapMirror transfer for a specific relationship completed, see the exported Snapshot timestamp for instance information.

```
cluster02::> snapmirror show -instance

                            Source Path: snap_src1:SMSource
                       Destination Path: svm_dst1:SMSource_dest
                      Relationship Type: XDP
                Relationship Group Type: none
                    SnapMirror Schedule: -
                 SnapMirror Policy Type: vault
                      SnapMirror Policy: XDPDefault
                            Tries Limit: -
                      Throttle (KB/sec): unlimited
                           Mirror State: Snapmirrored
                    Relationship Status: Idle
                File Restore File Count: -
                 File Restore File List: -
                      Transfer Snapshot: -
                      Snapshot Progress: -
                         Total Progress: -
              Network Compression Ratio: -
                    Snapshot Checkpoint: -
                        Newest Snapshot: snapmirror.12ceb7f0-b078-11e8-baec-0050
56b013db_2160175147.2020-01-24_043858
              Newest Snapshot Timestamp: 01/24 04:38:59
                      Exported Snapshot: snapmirror.12ceb7f0-b078-11e8-baec-0050
56b013db_2160175147.2020-01-24_043858
            Exported Snapshot Timestamp: 01/24 04:38:59
                                Healthy: true
                       Unhealthy Reason: -
               Constituent Relationship: false
                Destination Volume Node: cluster2-01
                        Relationship ID: 1a46a611-3e64-11ea-86bf-005056b013db
                   Current Operation ID: -
                          Transfer Type: -
                         Transfer Error: -
                       Current Throttle: -
              Current Transfer Priority: -
                     Last Transfer Type: resync
                    Last Transfer Error: -
                     Last Transfer Size: 0B
Last Transfer Network Compression Ratio: 1:1
                 Last Transfer Duration: 0:0:1
                     Last Transfer From: snap_src1:SMSource
            Last Transfer End Timestamp: 01/24 04:45:16
                  Progress Last Updated: -
                Relationship Capability: 9.4 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
```

You can also view the last Snapshot timestamp information from ThinkSystem Storage Manager for DM Series.

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

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

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

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

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

You can also use Storage Manager to view the SnapMirror log separately from the rest of the event logs; navigate to Cluster > Diagnostics > Logs > SnapMirror Log. From the Select node drop-down list, select the node that owns the volume in which you are interested.

10 Best Practices for DR Configurations

Best Practices
• Volumes that belong to one SVM at the source site should be replicated to one SVM at the destination site. 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 all 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. • The destination SVM should be a member of the same Active Directory, LDAP, or NIS domain of which the source SVM is a member. 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 correct them for access from a different domain can be extremely difficult and time consuming. It is also important so that authentication of tools running in SAN clients such as Lenovo SnapCenter Plug-in for Windows and SnapDrive® for Windows can be done using the same credentials. • Because the destination SVM is a different SVM than the source, and because Lenovo recommends that it be a member of the same Active Directory domain, the destination SVM must be joined to the domain with a different SVM name. It is 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 practice makes sure that CIFS shares are still accessible using the same UNC path name and that NFS clients are also able to access the expected path. • Using destination volume names that are the same as the source volume names is not required. However, 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. • Construct the destination NAS namespace for an SVM such that it is identical in paths and directory structure as the source SVM. • Many SAN clients cannot access a LUN that resides in a completely 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. • As new volumes are created at the source site, SnapMirror relationships must be created to replicate those volumes. You should 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.

11 Configuration and Failover for Disaster Recovery

Configuration and failover for DR is presented here in an overview of the DR process for intracluster SnapMirror DP mirrors. 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.

11.1 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 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.

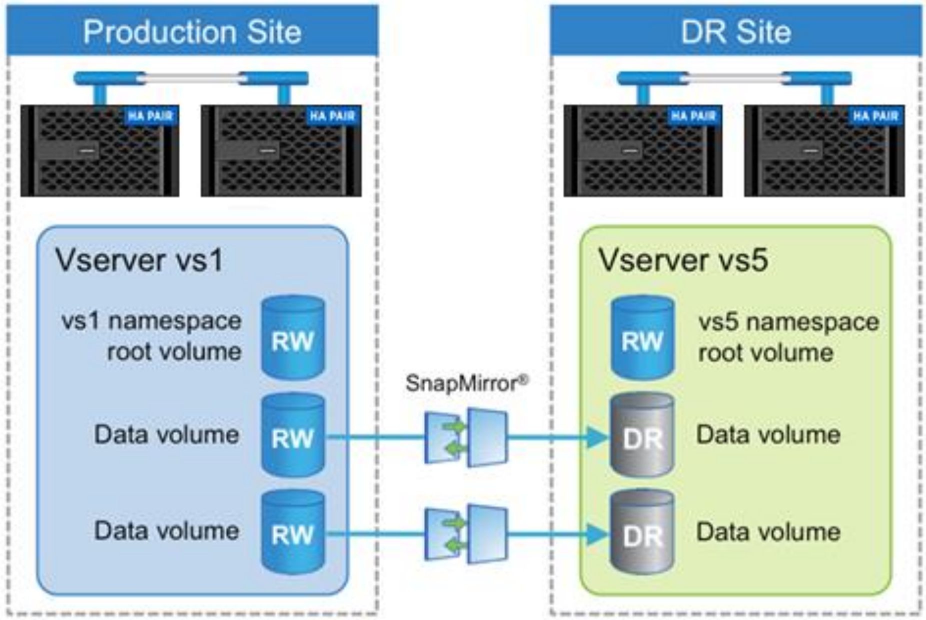
  Note: A server must be available at the DR site to perform the necessary Active Directory FSMO roles. For information regarding transferring roles to a surviving Active Directory server or seizing these roles from a failed server, see Microsoft KB 255504.
- The DR site has time synchronized to the same 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.
- The 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 the DNS such that home directory mount requests to the primary site SVM are directed to the DR site SVM instead.

11.2 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 CIFS shares, assigning NFS export policies, and other things can all be performed ahead of time. SnapMirror cannot be used to replicate configuration information that could be independent in the destination SVMs. These configurations include SVM domain membership, CIFS configuration, NFS policies, Snapshot policy schedules, or Lenovo storage efficiency policies.

Figure 16) Volume layout for DR. illustrates volume layout for DR.

Figure 16) 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 SnapDrive for Windows. 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 to DP volumes after failover.

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, a policy must be created in the destination SVM in order to schedule the dedupe process after failover at the DR site. Lenovo storage efficiency policies must be applied to DP volumes after failover.

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 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 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.

Because some hosts do not support connecting to LUNs in read-only containers, which is what a SnapMirror destination volume is, mapping LUNs to igroups is normally performed after failover.

11.3 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 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.

11.4 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 you are using Lenovo storage efficiency technology, assign a storage efficiency policy to the recovered volumes.

12 Where to Find Additional Information

The following references were used in this TR:

- ONTAP 9 Information Center
  https://thinksystem.lenovofiles.com/storage/help/nav/2_7

Contact 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/supportphonelist for your region support details.

Notices

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.

Trademarks