Technical Report

SnapMirror Configuration and Best Practices Guide for Clustered Data ONTAP

Amit Prakash Sawant, NetApp
December 2013 | TR-4015

SnapMirror Configuration and Best Practices

This document describes information and best practices related to configuring replication in NetApp® clustered Data ONTAP®.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

There are several approaches to increasing data availability in the face of hardware, software, or even site failures. Backups provide a way to recover lost data from an archival medium (tape or disk). Redundant hardware technologies also help mitigate the damage caused by hardware issues or failures. Mirroring provides a third mechanism to facilitate data availability and minimize downtime. NetApp SnapMirror® technology offers a fast and flexible enterprise solution for mirroring or replicating data over local area networks (LANs) and wide area networks (WANs). SnapMirror is a key component in enterprise data protection (DP) strategies.

1.1 Purpose and Intended Audience

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

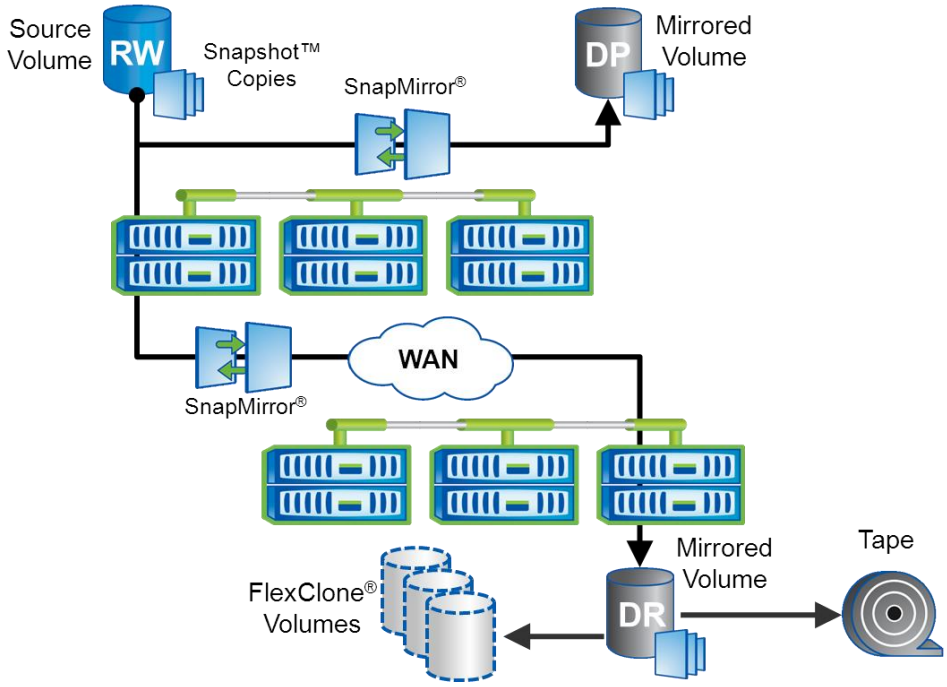
This document assumes that the reader has an understanding of the following processes and technologies:

- Storage systems administration working knowledge of clustered Data ONTAP operational processes
- Storage systems administration working knowledge of NetApp features such as NetApp Snapshot™ copies, NetApp FlexVol® volumes, NetApp FlexClone® volumes, and NetApp Infinite Volumes
- General knowledge of disaster recovery (DR) and data replication solutions
- Familiarity with the Data ONTAP 8.2 Cluster-Mode Data Protection Guide on the NetApp Support (formerly NOW®) site

1.2 SnapMirror Uses and Benefits

Replication can be performed within the same cluster or remotely to another cluster. NetApp Data ONTAP provides integrated data replication technologies for creating replica copies that can be used for DR, to offload tape backup processes from the primary, to distribute datasets to other locations, and to create read/write clones for test and development environments. For an overview of clustered Data ONTAP replication, refer to Figure 1.

Figure 1) NetApp clustered Data ONTAP replication overview.



Integrated Data Protection

DP capabilities are integrated within the NetApp Data ONTAP operating system. NetApp SnapMirror is integrated with NetApp Snapshot technology, which provides a method for quickly and efficiently creating on-disk replicas or point-in-time copies of data that do not require an actual copy operation to create.

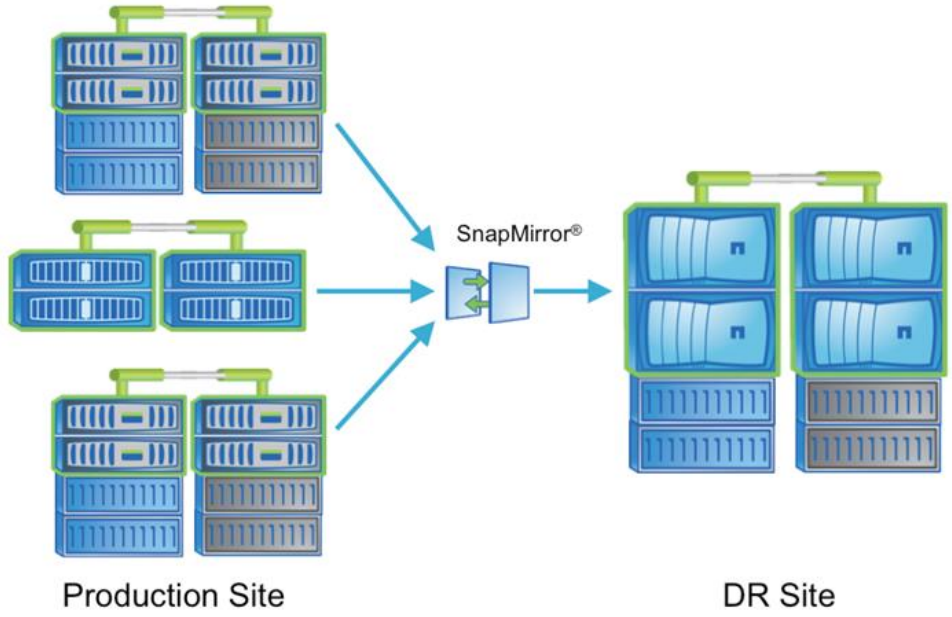
NetApp Integrated Data Protection can be used to create an on-disk, quickly accessible history of application-consistent Snapshot copies that eliminates the concept of traditional backup windows. NetApp SnapMirror then replicates the history of Snapshot copies to the destination volumes that can be used for backup, DR, or test and development.

SnapMirror replication is efficient because it only replicates the 4KB blocks that have changed or have been added since the previous update. Additional efficiency is gained when SnapMirror is combined with NetApp Storage Efficiency technologies. When fabric-attached storage (FAS) deduplication is used on the primary storage, only unique data is replicated to the DR site. If compression is enabled on the source, then compression is maintained on the destination. Data is not uncompressed because it is replicated. These technologies can result in telecommunication savings and significant storage capacity savings.

SnapMirror for Disaster Recovery

SnapMirror is an integral part of DR plans. If critical data is replicated to a different physical location, a serious disaster does not have to result in extended periods of unavailable data. Clients can access replicated data across the network until the damage caused by the disaster is repaired. Application servers at the recovery site can access replicated data to restore operations for business-critical applications for as long as necessary to recover the production site. Recovery might include recovery from corruption, natural disaster at the production site, accidental deletion, and so on.

Figure 2) SnapMirror for DR.



In cases in which a disaster requiring a failover occurs and the primary storage is not completely lost, SnapMirror provides an efficient means of resynchronizing the primary and DR sites. When the primary site is back online, SnapMirror resynchronizes the two sites, transferring only changed or new data back to the primary site from the DR site by simply reversing the SnapMirror relationships. After the primary production site resumes normal application operations, SnapMirror transfers to the DR facility resume without requiring another complete data transfer.

NetApp FlexClone for Disaster Recovery Testing and Application Test/Development

NetApp FlexClone technology can be used to quickly create a read-write copy of a SnapMirror destination FlexVol volume, eliminating the need for additional copies of the data. For example, a 10GB FlexClone does not require another 10GB FlexClone; it only requires the metadata needed to define the FlexClone. FlexClone volumes only store data that is written or changed after a clone is created.

Figure 3) Capacity requirements for DR testing with NetApp FlexClone.

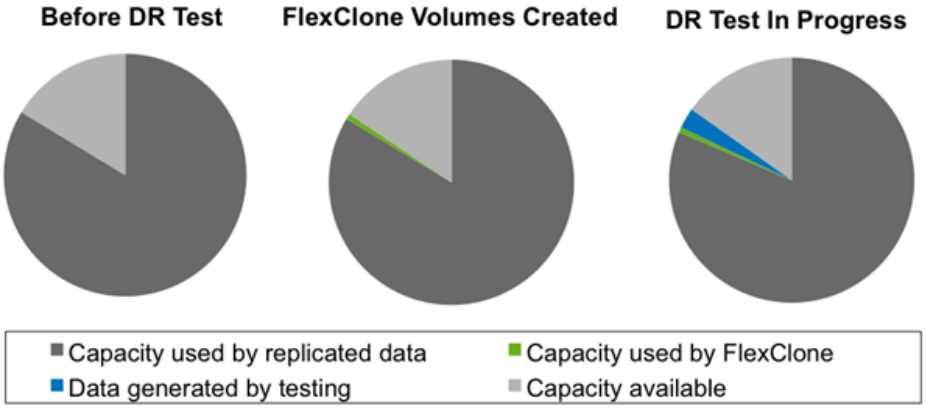
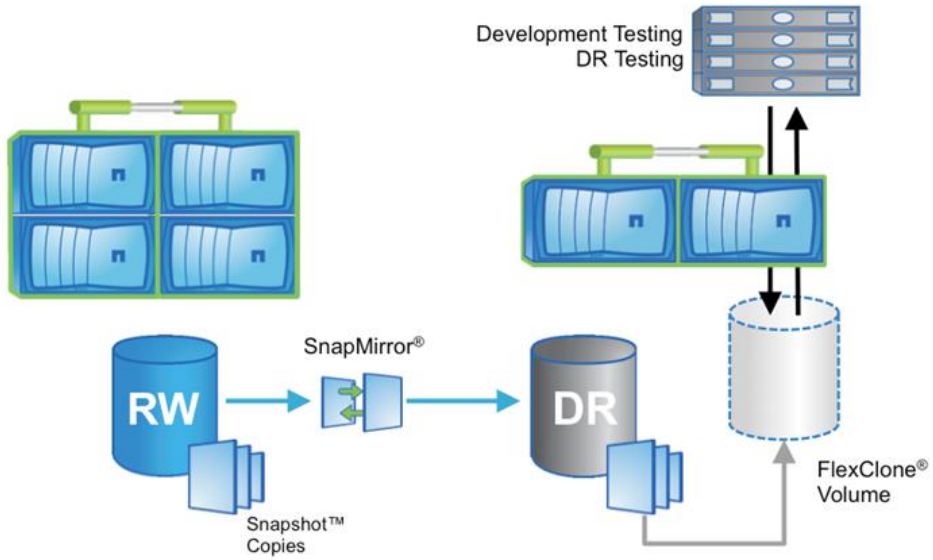


Figure 4 illustrates how FlexClone volumes share common data blocks with their parent FlexVol volumes but behave as independent volumes. DR environment testing can be performed, even for an extended period of time, while replication to the parent FlexVol volume is still occurring in the background.

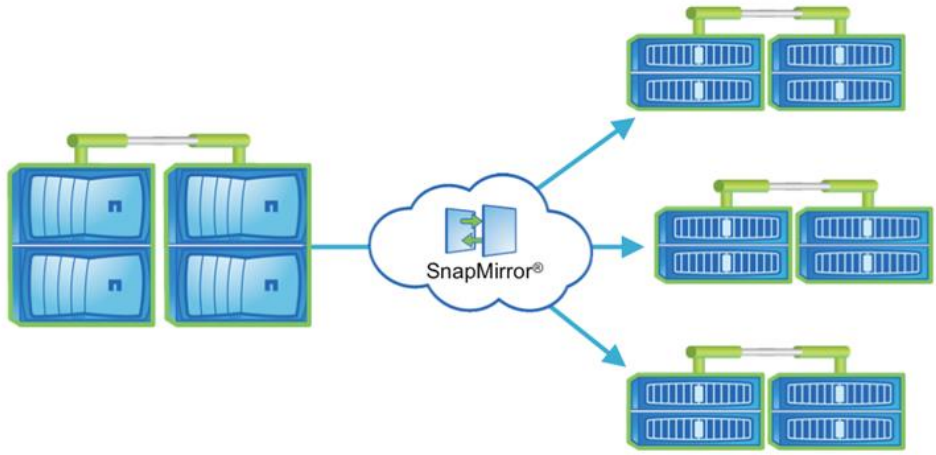
Figure 4) FlexClone volumes for development and DR testing.



Data Distribution and Remote Data Access

SnapMirror can be used to distribute large amounts of data throughout the enterprise, as shown in Figure 5, enabling read-only access to data at remote locations. Remote data access provides faster access to data by clients in the remote locations; it also allows more efficient and predictable use of an 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.

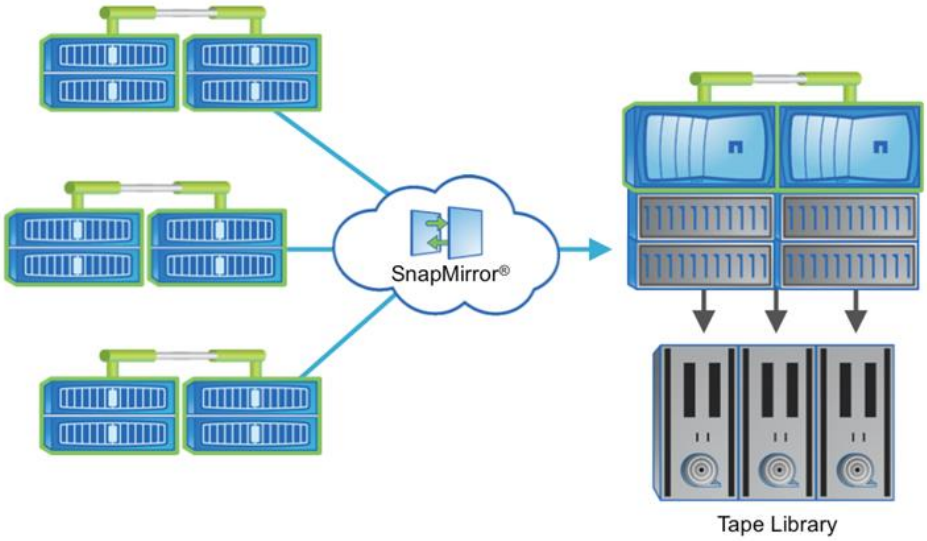
Figure 5) SnapMirror for data distribution.



Backup Off-loading and Remote Tape Archiving

SnapMirror can also be used for backup consolidation and for off-loading tape backup overhead from production servers. This facilitates centralized backup operations, reducing backup administrative requirements at remote locations. Because NetApp Snapshot technology eliminates the traditional backup window on the primary storage system, off-loading tape backup to a SnapMirror destination, as shown in Figure 6, dramatically reduces the overhead of backup operations on production storage systems.

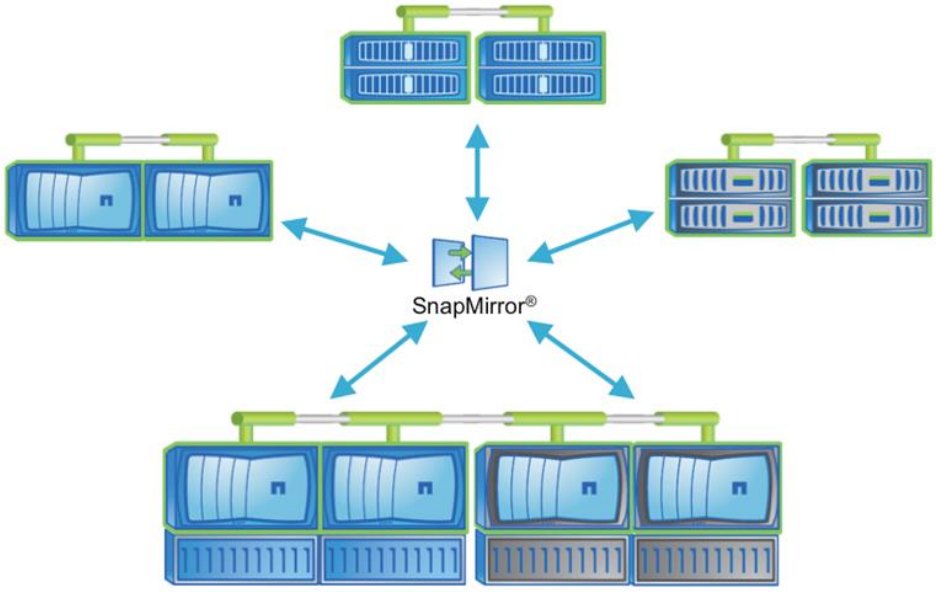
Figure 6) SnapMirror for remote tape archiving.



Unified Architecture Flexibility

Starting clustered Data ONTAP 8.1, SnapMirror can be used between NetApp FAS and/or V-Series storage systems. Systems with different performance characteristics and different costs can be deployed at the primary and DR sites. For example, depending on the capabilities required, the DR site might contain a lower-model storage system, SATA disk versus Fibre Channel (FC) disk, or the iSCSI or Fibre Channel over Ethernet (FCoE) protocol versus FC. Figure 7 illustrates the flexibility within a unified architecture.

Figure 7) Unified architecture flexibility.



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

2 Overview

2.1 SnapMirror Overview

SnapMirror in clustered Data ONTAP provides asynchronous volume-level replication based on a configured replication update interval. SnapMirror uses NetApp Snapshot technology as part of the replication process.

Clustered Data ONTAP 8.1 onward provides the following replication capabilities:

- **Data protection mirrors.** Replication to create a backup copy within the same cluster (intracluster) or to create a DR copy in a different cluster (intercluster).
- **Load-sharing mirrors.** Replication from one volume to multiple volumes in the same cluster to distribute a read-only workload across a cluster.

Basics of SnapMirror Replication

When the scheduler triggers a replication update, the following operations are performed:

1. A new Snapshot copy is created on the source volume.
2. The block-level difference between the new Snapshot copy and the last replication Snapshot copy is determined and then transferred to the destination volume. This transfer includes other Snapshot copies that were created between the last replication Snapshot copy and the new one.
3. When the transfer is complete, the new Snapshot copy exists on the destination volume.

A SnapMirror destination volume is available for read-only access if it is shared using Common Internet File System (CIFS) protocol, exported using Network File System (NFS) protocol. A logical unit number (LUN) in the replicated volume can be made available to a client that supports connection to read-only LUNs.

Replication occurs at the volume level. Qtrees can be created in clustered Data ONTAP and replicated along with the replicated volume; however, individual qtrees cannot be separately replicated.

DP relationships can be resynchronized in either direction after a failover without recopying the entire volume. If a relationship is resynchronized in the reverse direction, only new data written since the last successful synchronization snapshot will be sent back to the destination.

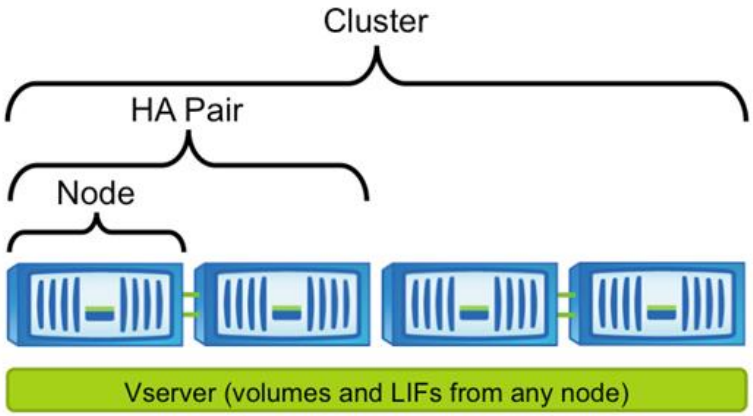
SnapMirror relationships in clustered Data ONTAP 8.1 must be managed by a cluster administrator; administration cannot be delegated to a Storage Virtual Machine administrator. Starting with clustered Data ONTAP 8.2, a cluster administrator can delegate the management of SnapMirror relationships to a Storage Virtual Machine administrator.

2.2 Clustered Data ONTAP Overview

Some basic terms used in clustered Data ONTAP 8.1 onward include:

- **Cluster.** Consists of 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.
- **Clustered Data ONTAP.** The Data ONTAP operating mode that supports interconnection of nodes into a cluster.
- **Data network.** The network used by clients to access data.
- **HA interconnect.** The dedicated interconnect between two nodes in one high-availability (HA) pair.
- **HA pair.** Two nodes configured in a pair for HA.
- **Ifgrp.** A collection of physical ports combined to create one logical port used for link aggregation; an integration group.
- **LIF.** A logical interface that is assigned an IP address that provides an Ethernet access point to a particular node in the cluster.
- **Intercluster LIF.** A LIF used only for intercluster replication, assigned only to a node.
- **Intercluster network.** The network used for communication and replication between different clusters.
- **Management network.** The network used for administration of the cluster, Storage Virtual Machine, and nodes.
- **Node.** A single NetApp controller, one of a high-availability pair.
- **Port.** A physical port, such as `e0e` or `e0f`, or a logical port such as a virtual LAN (VLAN) or an interface group (ifgrp).
- **Storage Virtual Machine.** A logical storage server that provides data access to LUNs and/or a network-attached storage (NAS) namespace from one or more logical interfaces (LIFs).

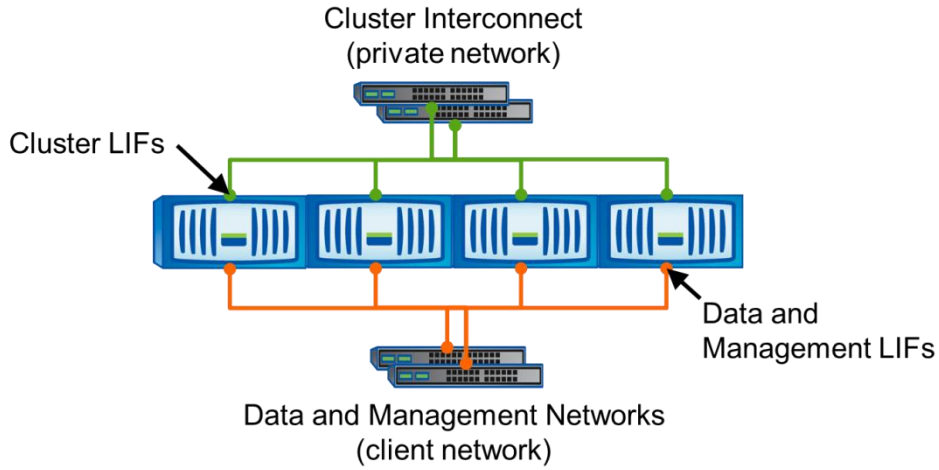
Figure 8) Node, HA pair, cluster, and Storage Virtual Machine.



There are multiple types of networks in a clustered Data ONTAP solution, as shown in Figure 9 and Figure 10. It is important to understand what each network type is used for.

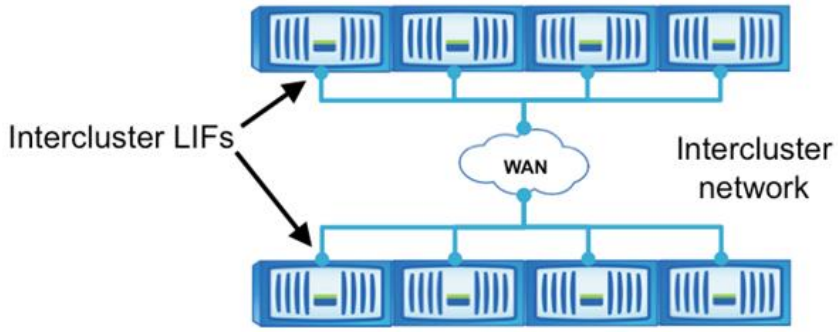
The cluster-interconnect network is a dedicated, high-speed, low-latency private network used for communication and replication between nodes in the same cluster. This is a redundant back-end network that cannot be used or shared for client access to data or for managing the cluster, nodes, or Storage Virtual Machines. Client access to data occurs on the data network. Management of the cluster, nodes, and Storage Virtual Machines 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 9) Cluster-interconnect and data and management networks.



An intercluster network is a network that allows communication and replication between two different clusters operating in clustered Data ONTAP, as shown in Figure 10. This network might be a network consisting of dedicated physical ports but could also be a network sharing ports with the data and/or management networks. The intercluster network is discussed in detail in the following section.

Figure 10) Intercluster network.



3 Network Configuration for Replication Between Different Clusters

Clustered Data ONTAP 8.1 is the first release that allows replication between different clusters, providing cross-site DR capabilities. New capabilities have been introduced in clustered Data ONTAP 8.1 for the following purposes:

- **Cluster peering.** The act of connecting two clusters to allow replication to occur between them.
- **Intercluster logical interfaces.** Logical network interfaces used for intercluster communication.
- **Intercluster ports.** Ports dedicated to intercluster replication.

Clusters must be joined in a peer relationship before replication between different clusters is possible. Cluster peering is a one-time operation that must be performed by the cluster administrators. The following steps are required to allow replication between different clusters:

1. Have a clear understanding of cluster peering.
2. Determine whether or not to share ports for data access and intercluster replication.
3. Designate ports for intercluster replication if dedicating ports.
4. Create intercluster LIFs on each node in the clusters.
5. Peer the clusters together.

Cluster peering must be performed because this defines the network on which all replication between different clusters occurs. Additionally, starting in clustered Data ONTAP 8.2, Storage Virtual Machines must be joined in a peer relationship before replication between different Storage Virtual Machines is possible. Storage Virtual Machine peering is discussed in section 4.

3.1 Intercluster Networking

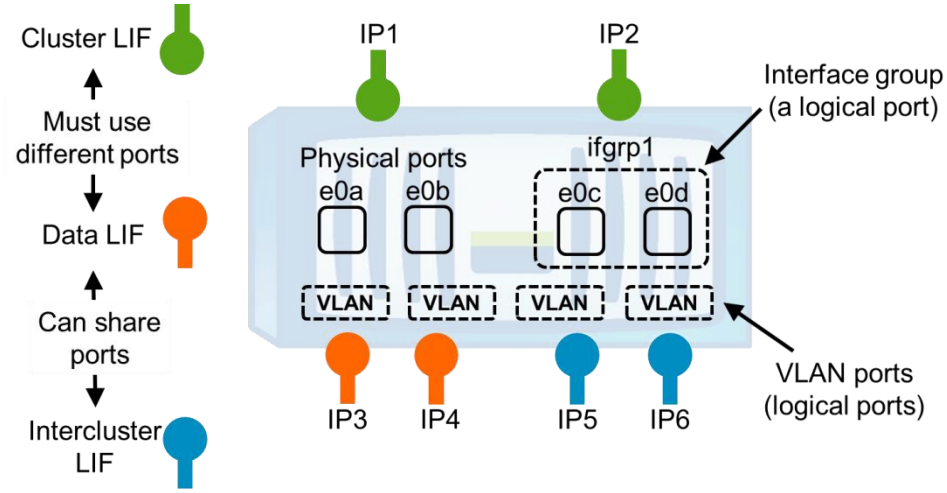
Cluster peering requires intercluster LIFs for communication and replication between two clusters. An intercluster LIF must be created on an intercluster-capable port, which is a port assigned the role of intercluster or a port assigned the role of data. Ports that are used for the intracluster cluster interconnect may not be used for intercluster replication.

It is possible to assign intercluster LIFs to ports that have the role of data, which are the same ports used for CIFS, NFS, or iSCSI access. However, to dedicate certain ports for intercluster replication, simply assign intercluster LIFs to dedicated ports that have the role of intercluster, because intercluster ports can only be used for replication. Each method and how to determine which method to use are discussed later in this document.

Understanding the difference between LIFs and ports is important. A LIF is a logical interface that is assigned an IP address and provides an access point to a particular node in the cluster. LIFs are

assigned to ports and there are different types of ports. A port can be a physical port where a cable is plugged in, such as `e0e` or `e0f`, on a NetApp controller. A port can also be a logical port, such as a VLAN or an ifgrp. Ifgrps are used for Ethernet link aggregation. Figure 11 shows these different types of ports.

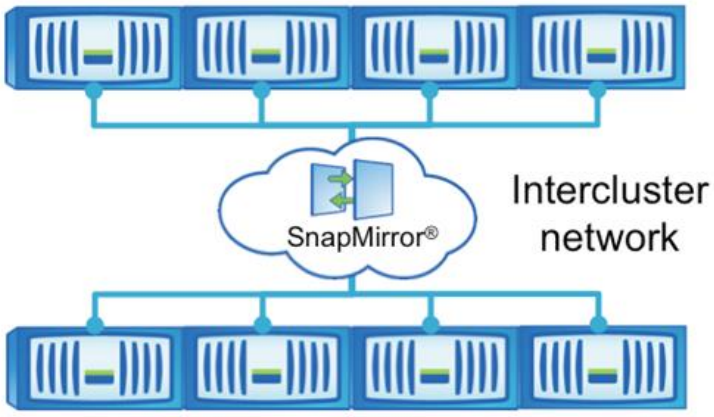
Figure 11) Port types.



Any Ethernet port type in the system can have the role of data or intercluster and be used for intercluster replication, which includes physical ports, such as `e0e` or `e0f`, and logical ports, such as a virtual LAN (VLAN) or an ifgrp.

At least one intercluster LIF must be created on every node in the cluster. Every intercluster LIF requires an IP address, meaning at least one IP address per node must be dedicated to performing intercluster replication. Provisioning intercluster LIFs only on some nodes of the cluster is not a supported configuration. Configuring an intercluster LIF on every node is required to allow intercluster SnapMirror transfers to continue when volume move operations move volumes between nodes, as shown in Figure 12. SnapMirror relationships do not have to be modified because volumes are moved to other nodes in the cluster.

Figure 12) SnapMirror on the intercluster network.



Creating intercluster LIFs defines the intercluster network on which replication occurs between two different clusters. Replication between two clusters can only occur on the intercluster network, even if the intercluster network is on the same subnet as a data network in the same cluster. It is not possible for other protocols such as CIFS, NFS, iSCSI, FC, or FCoE to use intercluster LIFs or ports. The IP addresses assigned to intercluster LIFs can reside in the same subnet as data LIFs or in a different

subnet. When an intercluster LIF is created, an intercluster routing group is automatically created on that node. If the source and destination clusters must use different subnets for intercluster replication, then it is necessary to define a gateway address for the intercluster routing group.

Intercluster LIFs are node scoped; therefore, when the port hosting an intercluster LIF fails, the LIF can fail over only 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 (same MTU's, flow control, tcp options and so on). 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. In clustered Data ONTAP 8.2, the same transfer will not automatically continue after the storage failover (SFO) of the destination. If SFO happens on the source, it will. However, replication as such will continue automatically from the surviving node.

Replication between different clusters in Data ONTAP 8.1 onward operating in clustered Data ONTAP requires IP connectivity between the clusters. SnapMirror replication over an FC network is not available in clustered Data ONTAP.

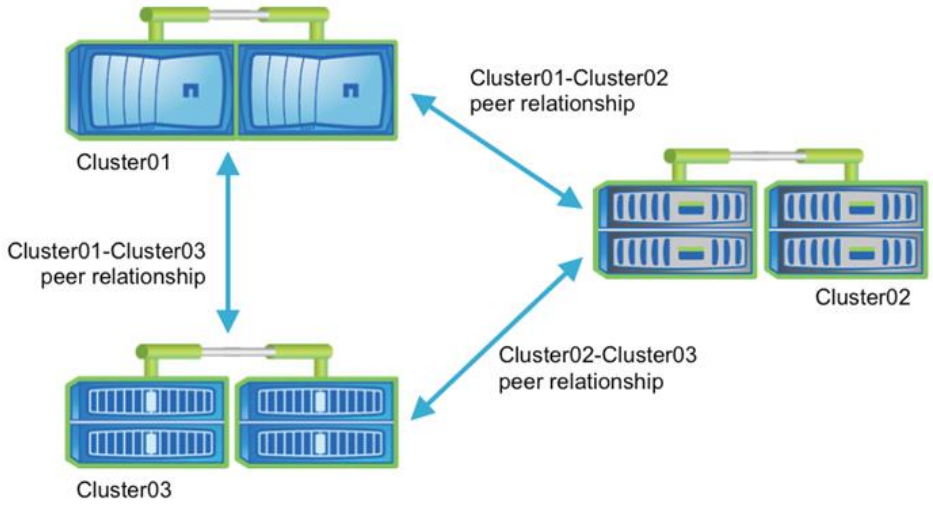
3.2 Cluster Peering

After the intercluster LIFs have been created and the intercluster network has been configured, cluster peers can be created. A cluster peer is a cluster that is allowed to replicate to or from another cluster.

Establishing cluster peering is a one-time operation that must be performed by the cluster administrators. 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 intercluster IP addresses of the other cluster.

A cluster can be in a peer relationship, as shown in Figure 13, with up to eight clusters, allowing multiple clusters to replicate between each other.

Figure 13) Multiple cluster peer relationships.

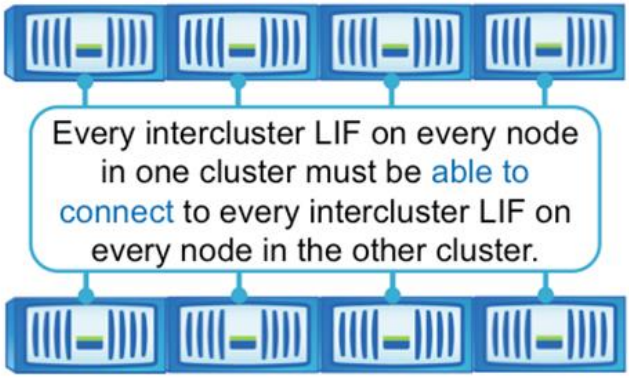


3.3 Cluster Peer Requirements

Cluster peer requirements include the following:

- The time on the clusters must be in sync 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 requires an IP address dedicated for intercluster replication.
- The correct maximum transmission unit (MTU) value must be used on the network ports that are used for replication. The network administrator can identify which MTU value to use in the environment. 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, as shown in Figure 14.

Figure 14) Full connectivity of intercluster LIFs.



3.4 Intercluster Multipathing and Network Redundancy

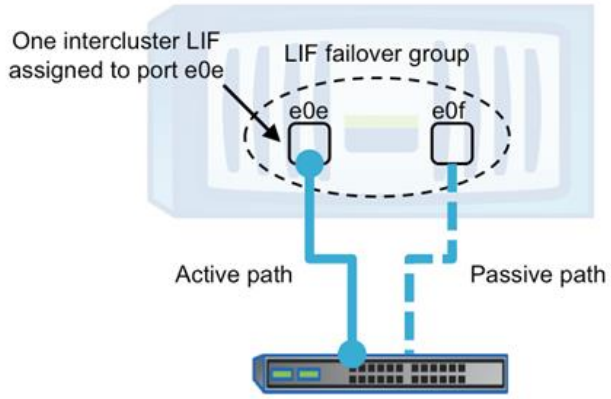
NetApp clustered Data ONTAP 8.1 provides the following capabilities to configure two kinds of multipathing for intercluster SnapMirror replication:

- **Active-passive.** Replication in which a particular path is used unless that path fails, then a different path is used.
- **Active-active.** Replication in which multiple paths are actively used at the same time. If one path fails, the surviving paths remain active and all replication transfers continue.

Active-Passive Intercluster Multipathing in Data ONTAP

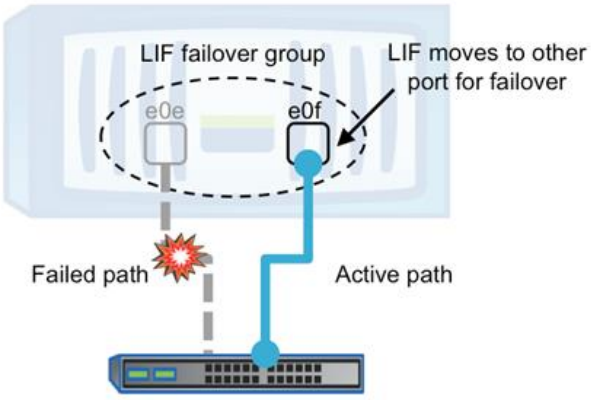
In many ways an intercluster LIF behaves in the same way as a LIF used for CIFS or NFS in terms of active-passive failover, except that an intercluster LIF cannot fail over to a port in a different node. The initial placement of a LIF on a specific port determines which port is used by that LIF. If the ports and LIFs have been configured such that the LIFs are redundant because there are other ports the LIFs might failover to on the same node, then the port the LIF is initially placed on is the active path and any port that LIF might fail over to is the passive path. So, it can be said that a properly configured redundant LIF provides active-passive multipathing, as shown in Figure 15.

Figure 15) Active-passive multipathing.



Communication on an intercluster LIF occurs only 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 16) Active-passive multipathing during LIF failover.

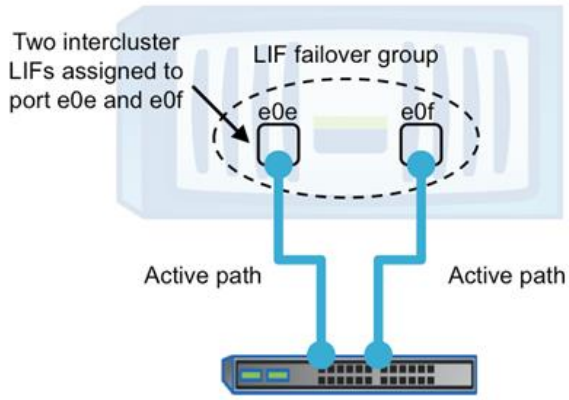


To configure active-passive multipathing, assign an intercluster LIF to an intercluster-capable port and make sure that another intercluster-capable port is configured that is capable of supporting that connection. Make sure that the LIF's failover policy is configured such that the LIF's failover group contains the necessary ports to allow failover, as shown in Figure 16.

Active-Active Intercluster Multipathing in Data ONTAP

Active-active multipathing 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 17.

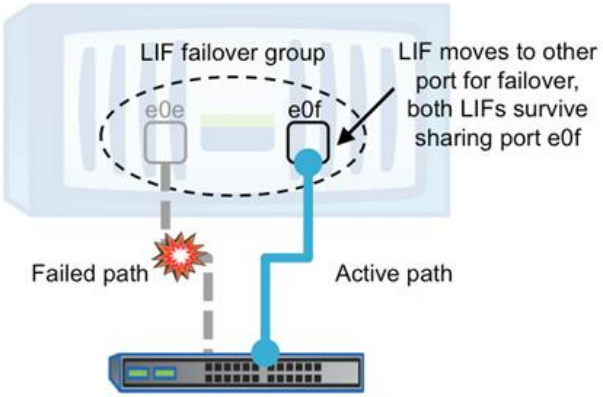
Figure 17) Active-active multipathing.



In clustered Data ONTAP 8.1 it is important that all paths on a node used in an active-active configuration for intercluster replication have equal performance characteristics. Configuring multipathing in such a way that one intercluster LIF is on a slow path and one on a fast path adversely affects performance because data is multiplexed across the slow and fast paths simultaneously. Starting in clustered Data ONTAP 8.2, 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, as shown in Figure 18.

Figure 18) Active-active multipathing during LIF failover.



To configure two-path active-active multipathing for SnapMirror, configure two intercluster-capable ports (role type intercluster or role type data), create two intercluster LIFs, and assign one LIF to each port. Make sure that each LIF's failover policy is configured such that the LIF's failover group contains the necessary ports to allow failover.

Depending on the replication workload between any given pair of source and destination nodes, it might be necessary to configure multiple paths on the source and destination node. There are no special configuration settings necessary to apply to each SnapMirror relationship to make use of the multipath

connection. All SnapMirror relationships are automatically multiplexed across the available LIFs on the source and destination nodes.

Switch-Based Link Aggregation for Multipathing

As mentioned earlier in this document, an intercluster LIF can be assigned to any kind of port in the system, including a logical port like 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 would have 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 particular source node.

If switch-based link aggregation is used to allow multiple physical paths in the ifgrp to be used when replicating between two particular nodes, additional intercluster LIFs can be configured on either of the two nodes. Data ONTAP automatically establishes a connection between every LIF on the source and destination node for SnapMirror. This 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 particular nodes. This would likely not be 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 particular nodes might not be beneficial, since many nodes must share the WAN bandwidth anyway.

Best Practice

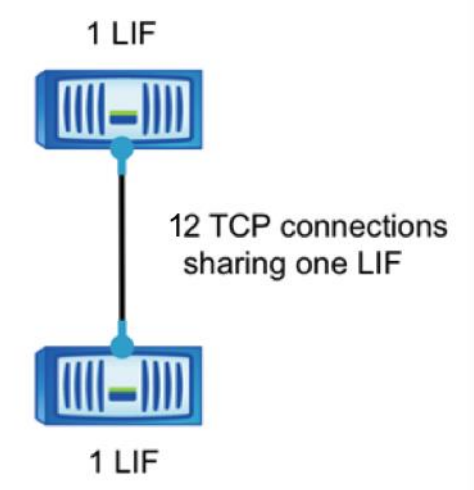
When using switch-based link aggregation, create the ifgrp with a `multimode_lacp` mode and set the distribution function of the ifgrp to `port`. Using the port value for the distribution function configures the ifgrp to distribute connections across paths by hashing the source/destination IP address,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

3.5 Network Connections for Intercluster SnapMirror

In clustered Data ONTAP, the number of intercluster LIFs determines the number of transmission control protocol (TCP) connections established between the source and destination node for SnapMirror. TCP connections are not created per volume or per relationship.

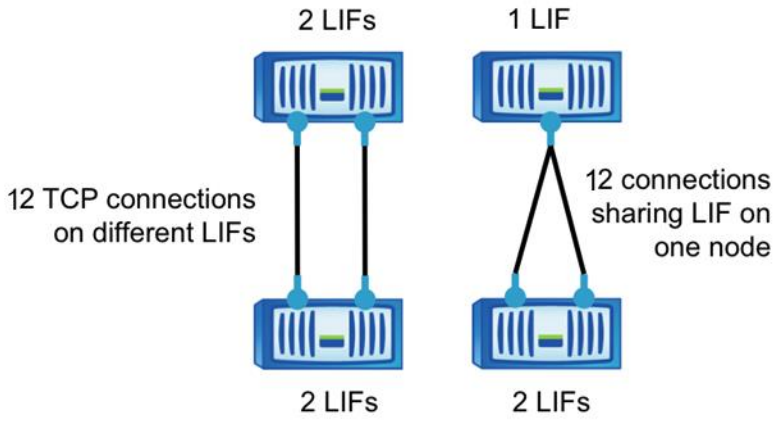
Remember that starting in clustered Data ONTAP 8.2, Data ONTAP establishes at least 12 intercluster TCP connections for sending data. A minimum of 12 TCP connections are created for sending data, as shown in Figure 19. This is true even if both the source and destination nodes have only one intercluster LIF and enough connections are created so that all intercluster LIFs on both the source and destination nodes are used.

Figure 19) TCP connections with one intercluster LIF.



If the source node, destination node, or both nodes are configured with 2 intercluster LIFs, then Data 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 20. This example shows different combinations of intercluster LIFs that produce 12 intercluster TCP connections. It is not possible to select a specific LIF pair to use for a certain TCP connection; they are managed automatically by Data ONTAP.

Figure 20) TCP connections with two intercluster LIFs.



After scaling past 12 intercluster LIFs on a node, Data ONTAP creates additional intercluster TCP connections, creating enough so that all intercluster LIFs are used.

The creation of additional intercluster TCP connections continues as more intercluster LIFs are added to either the source or the destination node. A maximum of 24 intercluster connections are currently supported for SnapMirror on a single node in Data ONTAP.

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, as described in the section titled "Switch-Based Link Aggregation for Multipathing."

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 1GbE link.

3.6 Determining Whether to Share or Dedicate Ports for Replication

There are a number of configurations and requirements to consider when determining whether to share or dedicate ports for replication; they include:

- **LAN type.** 1GbE or 10GbE connectivity.
- **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.
- **Replication interval.** Replication during nonproduction hours may have an irrelevant impact on the data network.
- **Change rate.** The amount of data required for replication may not interfere with client data access.
- **Number of ports used by the solution.** Dedicating ports for replication requires additional switch ports and cable runs.

Intercluster LIF failover policies can be manually configured so that intercluster LIFs only use a subset of data ports. However, when dedicating specific ports for intercluster by assigning ports the role of intercluster, Data ONTAP automatically configures LIF failover groups so that only intercluster ports are used for replication. It does not allow any data protocols to fail over or migrate to the intercluster ports. This is a more rigid and automated method for dedicating ports to intercluster replication.

Consider the following when sharing ports for intercluster replication:

- For a high-speed network such as 10GbE, a sufficient amount of local LAN bandwidth might be available to perform replication on the same 10GbE ports that are used for data access. In many cases, the available WAN bandwidth is far less than 10GbE, which reduces the LAN network utilization to only that which the WAN is capable of supporting.
- All nodes in the cluster may have to replicate data and share the available WAN bandwidth, making data port sharing more acceptable.
- Sharing ports for data and replication eliminates the extra port counts required to dedicate ports for replication.
- If the replication interval is set to perform only after hours, when minimal to no client activity exists, then using data ports for replication during this time is acceptable, even without a 10GbE LAN connection.
- Consider the data change rate and replication interval and whether or not the amount of data that must be replicated on each interval requires enough bandwidth that it might cause contention with data protocols if sharing data ports.
- When data ports for intercluster replication are shared, the intercluster LIFs can be migrated to any other intercluster-capable port on the same node to control the specific data port that is used for replication.

Consider the following when dedicating ports for intercluster replication:

- If the amount of available WAN bandwidth is similar to that of the LAN ports and the replication interval is such that replication occurs while regular client activity exists, then dedicate Ethernet ports for intercluster replication to avoid contention between replication and the data protocols.
- If the network utilization generated by the data protocols (CIFS, NFS, iSCSI) is such that network utilization is above 50%, then dedicate ports for replication to allow for nondegraded performance in the event of a node failover.
- When physical 10GbE ports are used for data and replication, VLAN ports can be created for replication and the logical ports dedicated for intercluster replication.
- Consider the data change rate and replication interval and whether or not the amount of data that must be replicated on each interval requires enough bandwidth that it might cause contention with data protocols if sharing data ports.
- If the replication network requires configuration of an MTU size that differs from the MTU size used on the data network, then physical ports must be dedicated for replication because the MTU size can only be configured on physical ports.

3.7 Sharing Data Ports for Intercluster Replication

This section demonstrates how to create intercluster LIFs that share data ports. In this example, a two-node cluster exists in which each node has two data ports, `e0c` and `e0d`, which are shared for intercluster replication.

Configuring Intercluster LIFs to Share Data Ports

Before completing the following steps, replace the ports, networks, IP addresses, subnet masks, and subnets with those specific to the environment.

6. Check the role of the ports in the cluster.

```
cluster01::> network port show
data sheet
                                   Auto-Negot  Duplex     Speed(Mbps)
Node   Port   Role         Link   MTU Admin/Oper  Admin/Oper Admin/Oper
------ ------ ------------ ---- ----- ----------- ---------- ----------
cluster01-01
       e0a    cluster      up    1500  true/true  full/full   auto/1000
       e0c    data         up    1500  true/true  full/full   auto/1000
       e0d    cluster      up    1500  true/true  full/full   auto/1000
cluster01-02
       e0a    cluster      up    1500  true/true  full/full   auto/1000
       e0c    data         up    1500  true/true  full/full   auto/1000
       e0d    cluster      up    1500  true/true  full/full   auto/1000
```

7. Create an intercluster LIF on each node in `cluster01`. This example uses a LIF naming convention of `<nodename>_icl#` for the intercluster LIF.

```
cluster01::> network int create -vserver cluster01-01 -lif cluster01-01_icl01 -role
intercluster -home-node cluster01-01 -home-port e0d -address 10.12.12.1 -netmask
255.255.255.0
cluster01::> network int create -vserver cluster01-02 -lif cluster01-02_icl01 -role
intercluster -home-node cluster01-02 -home-port e0d -address 10.12.12.2 -netmask
255.255.255.0
```

Best Practice

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 intercluster LIF; for example, `node_name_icl#` or `node-name-ic#`, depending on your preference.

8. Verify that the intercluster LIFs were created properly.

```
cluster01::> network int show -role intercluster
            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
cluster01-01
            cluster01-01_icl01
                         up/up    10.12.12.1/24      cluster01-01  e0c     true
cluster01-02
            cluster01-02_icl01
                         up/up    10.12.12.2/24      cluster01-02  e0c     true
```

9. Verify that the intercluster LIFs are configured to be redundant.

```
cluster01::> network int show -role intercluster -failover

         Logical         Home                  Failover        Failover
Vserver  Interface       Node:Port             Group Usage     Group
-------- --------------- --------------------- --------------- --------
cluster01-01
         cluster01-01_icl01 cluster01-01:e0d   system-defined
                            Failover Targets: cluster01-01:e0c,
                                              cluster01-01:e0d

cluster01-02
         cluster01-02_icl01 cluster01-02:e0d   system-defined
                            Failover Targets: cluster01-02:e0c,
                                              cluster01-01:e0d
```

Note: The LIFs in this example are assigned the `e0c` port on each node. If the `e0c` port fails, the LIF can fail over to the `e0d` port because `e0d` also has the role of data. The intercluster LIF was assigned to a data port; therefore, a failover group for the intercluster LIF was automatically created containing all ports with the role of data on that node. Intercluster failover groups are node specific; if changes are required they must be managed for each node because different nodes might use different ports for replication.

Best Practice

Verify that all necessary ports have access to the necessary networks or VLANs to allow communication after port failover. Intercluster LIFs can only fail over to other intercluster-capable ports on the same node, as defined in the failover group (in this example, other data ports); they cannot fail over to ports on other nodes. It is a best practice to configure two intercluster ports per node when dedicating ports for intercluster replication.

10. An intercluster routing group is automatically created for the intercluster LIFs. Run the `net routing-group show` command to display the routing groups.

Note: The intercluster routing groups begin with `i`.

```
cluster01::> network routing-group show -role intercluster

          Routing
Vserver   Group     Subnet          Role         Metric
--------- --------- --------------- ------------ -------
cluster01-01
          i10.12.12.0/24
                    i10.12.12.0/24  intercluster     40
cluster01-02
          i10.12.12.0/24
                    i10.12.12.0/24  intercluster     40
```

11. The intercluster network might require intercluster routes. Display the routes in the cluster.

    Note: No intercluster routes are available.

```
cluster01::> network routing-group route show
          Routing
Vserver   Group     Destination     Gateway         Metric
--------- --------- --------------- --------------- ------
cluster01
          c10.228.225.0/24
                    0.0.0.0/0       10.228.225.1     20
cluster01-01
          n10.228.225.0/24
                    0.0.0.0/0       10.228.225.1     10
cluster01-02
          n10.228.225.0/24
                    0.0.0.0/0       10.228.225.1     10
```

12. If communication between intercluster LIFs in different clusters requires routing, then create an intercluster route. The intercluster networks are scoped to each node; therefore, create an intercluster route on each node. In this example, `10.12.12.1` is the gateway address for the `10.12.12.0/24` network.

    Note: Creating a route is required only if the intercluster addresses in both clusters are not on the same subnet and a specific route is needed for communication between the clusters.

```
cluster01::> network routing-groups route create -server cluster01-01 -routing-group
i10.12.12.0/24 -destination 0.0.0.0/0 -gateway 10.12.12.1 -metric 40

cluster01::> network routing-groups route create -server cluster01-02 -routing-group
i10.12.12.0/24 -destination 0.0.0.0/0 -gateway 10.12.12.11 -metric 40
```

    Note: If the destination is specified as `0.0.0.0/0`, then it becomes the default route for the intercluster network.

13. Display the newly created routes.

```
cluster01::> network routing-group route show
          Routing
Vserver   Group     Destination     Gateway         Metric
--------- --------- --------------- --------------- ------
cluster01
          c10.228.225.0/24
                    0.0.0.0/0       10.228.225.1     20
cluster01-01
          n10.228.225.0/24
                    0.0.0.0/0       10.228.225.1     10
          i10.12.12.1/24
                    0.0.0.0/0       10.12.12.1       40
cluster01-02
          n10.228.225.0/24
                    0.0.0.0/0       10.228.225.1     10
```

```
          i10.12.12.1/24
                    0.0.0.0/0       10.12.12.1       40
```

Note: Although the intercluster routes do not have an assigned role, they are assigned to the routing group `i10.12.12.0/24`, which is assigned the role of intercluster, as shown in this net routing-group show command output. These routes are only used for intercluster communication.

14. Repeat the steps in this section to configure intercluster networking in the other cluster.

3.8 Dedicating Intercluster Ports for Intercluster Replication

This section demonstrates how to designate specific ports as intercluster ports and assign intercluster LIFs to those ports. In this example, a two-node cluster exists in which each node has two data ports, `e0e` and `e0f`, which are dedicated for intercluster replication. Before completing the following steps, replace the ports, networks, IP addresses, subnet masks, and subnets with those specific to your environment.

Configuring Intercluster LIFs to Use Dedicated Intercluster Ports

Before completing the steps in this section, review the section titled "Cluster Peer Requirements" and determine whether or not to share or dedicate ports for replication in your environment.

1. Check the role of the ports in the cluster.

```
cluster01::> network port show
                                        Auto-Negot  Duplex     Speed(Mbps)
Node   Port   Role         Link   MTU  Admin/Oper  Admin/Oper Admin/Oper
------ ------ ------------ ---- ----- ----------- ---------- ----------
cluster01-01
       e0a    cluster      up    1500  true/true  full/full   auto/1000
       e0b    cluster      up    1500  true/true  full/full   auto/1000
       e0c    data         up    1500  true/true  full/full   auto/1000
       e0d    data         up    1500  true/true  full/full   auto/1000
       e0e    data         up    1500  true/true  full/full   auto/1000
       e0f    data         up    1500  true/true  full/full   auto/1000
cluster01-02
       e0a    cluster      up    1500  true/true  full/full   auto/1000
       e0b    cluster      up    1500  true/true  full/full   auto/1000
       e0c    data         up    1500  true/true  full/full   auto/1000
       e0d    data         up    1500  true/true  full/full   auto/1000
       e0e    data         up    1500  true/true  full/full   auto/1000
       e0f    data         up    1500  true/true  full/full   auto/1000
```

2. Determine whether any of the LIFs are using ports that are dedicated for replication.

```
cluster01::> network int show

            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
cluster01
            cluster_mgmt up/up    10.228.xx.xx/24    cluster01-01  e0c     true
vs1
            vs1_lif1     up/up    10.12.12.5/24      cluster01-01  e0e     true
```

3. If a data LIF is using one of the ports to be dedicated to replication, then migrate the LIF to another port because intercluster ports cannot host data LIFs. This migration is nondisruptive, assuming that the other data ports were configured properly so that clients are able to access the LIF after migration.

   Note: SAN LIFs are not migrated between nodes. If the current port is configured with an iSCSI LIF, then make sure that the multipathing software in the iSCSI client is properly configured to

survive the outage of the iSCSI LIF. Then, unconfigure the existing iSCSI LIF, remove it, and recreate it on another port. The iSCSI client's multipathing software is responsible for making this reconfiguration nondisruptive for the clients because the LIF is removed and a new one is created.

```
cluster01::> network int migrate -vserver vs1 -lif vs1_lif1 -dest-node cluster01-01 -
dest-port e0d

cluster01::> network int show

            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
cluster01
            cluster_mgmt up/up    10.288.xx.xx/24    cluster01-01  e0c     true
vs1
            vs1_lif1     up/up    10.12.12.5/24      cluster01-01  e0d     false
```

4. The newly migrated LIFs might need modification to the LIF home port to reflect the new port where the LIF should reside.

```
cluster01::> network int modify -vserver vs1 -lif vs1_lif1 -home-node cluster01-01 -
home-port e0d

cluster01::> network int show

            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
vs1
            vs1_lif1     up/up    192.168.0.151/24   cluster01-01  e0d     true
```

5. After all LIFs have been migrated off the ports dedicated for replication, change the role of the port used on each node to intercluster.

```
cluster01::> network port modify -node cluster01-01 -port e0e -role intercluster

cluster01::> network port modify -node cluster01-01 -port e0f -role intercluster

cluster01::> network port modify -node cluster01-02 -port e0e -role intercluster

cluster01::> network port modify -node cluster01-02 -port e0f -role intercluster
```

6. After the cluster peer relationship has been successfully created between the two clusters, create the intercluster SnapMirror relationships.

```
cluster01::> network port show –role intercluster
                                      Auto-Negot  Duplex     Speed(Mbps)
Node   Port   Role         Link   MTU Admin/Oper  Admin/Oper Admin/Oper
------ ------ ------------ ---- ----- ----------- ---------- ----------
cluster01-01
       e0e    intercluster up    1500  true/true  full/full   auto/1000
       e0f    intercluster up    1500  true/true  full/full   auto/1000
cluster01-02
       e0e    intercluster up    1500  true/true  full/full   auto/1000
       e0f    intercluster up    1500  true/true  full/full   auto/1000
```

7. Create an intercluster LIF on each node in `cluster01`. This example uses the LIF naming convention `<nodename>_icl#` for intercluster LIF.

```
cluster01::> network int create -vserver cluster01-01 -lif cluster01-01_icl01 -role
intercluster -home-node cluster01-01 -home-port e0e -address 10.12.12.1 -netmask
255.255.255.0
```

```
cluster01::> network int create -vserver cluster01-02 -lif cluster01-02_icl01 -role
intercluster -home-node cluster01-02 -home-port e0e -address 10.12.12.2 -netmask
255.255.255.0
```

Best Practice

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 intercluster LIF; for example, `node-name_icl#` or `node-name-ic#`, depending on your preference.

8. Verify that the intercluster LIFs are configured for redundancy.

```
cluster01::> network int show -role intercluster –failover

         Logical         Home                  Failover        Failover
Vserver  Interface       Node:Port             Group Usage     Group
-------- --------------- --------------------- --------------- --------
cluster01-01
         cluster01-01_icl01 cluster01-01:e0e   system-defined
                            Failover Targets: cluster01-01:e0e,
                                              cluster01-01:e0f
cluster01-02
         cluster01-02_icl01 cluster01-02:e0e   system-defined
                            Failover Targets: cluster01-02:e0e,
                                              cluster01-02:e0f
```

Note: The LIFs in this example are assigned the `e0e` home port on each node. If the `e0e` port fails, the LIF can fail over to the `e0f` port because `e0f` is also assigned the role of intercluster. The intercluster LIF is assigned to an intercluster port; therefore, a failover group is automatically created containing all ports with the intercluster role on that node. In this example, the failover group does not include any data ports. Intercluster failover groups are node specific; therefore, if changes are required, they must be managed for each node because different nodes might use different ports for replication.

Best Practice

Verify that all the necessary ports have access to the necessary networks or VLANs to allow communication after port failover. Intercluster LIFs can only fail over to other intercluster-capable ports on the same node, as defined in the failover group (in this example, other intercluster ports); they cannot fail over to ports on other nodes. It is a best practice to configure two intercluster ports per node when dedicating ports for intercluster replication.

9. Verify that the intercluster LIFs were created properly.

```
cluster01::> network int show

            Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
cluster01
            cluster_mgmt up/up    10.288.xx.xx/24    cluster01-01  e0c     true
cluster01-01
            cluster01-01_icl01
                         up/up    10.12.12.1/24      cluster01-01  e0e     true
            clus1        up/up    169.254.xx.xx/24   cluster01-01  e0a     true
            clus2        up/up    169.254.xx.xx/24   cluster01-01  e0b     true
            mgmt1        up/up    10.228.225.xx/24   cluster01-01  e0c     true
cluster01-02
```

```
            cluster01-02_icl01
                         up/up    10.12.12.2/24      cluster01-02  e0e     true
            clus1        up/up    169.254.xx.xx/24   cluster01-02  e0a     true
            clus2        up/up    169.254.xx.xx/24   cluster01-02  e0b     true
            mgmt1        up/up    10.288.225.xx/24   cluster01-02  e0c     true
```

10. The intercluster network might require intercluster routes. An intercluster routing group is automatically created for the intercluster LIFs. Run the `net routing-group show` command to display the routing groups.

    Note: The intercluster routing groups begin with `i`.

```
cluster01::> network routing-group show -role intercluster

          Routing
Vserver   Group     Subnet          Role         Metric
--------- --------- --------------- ------------ -------
cluster01-01
          i10.12.12.0/24
                    10.12.12.0/24   intercluster      40
cluster01-02
          i10.12.12.0/24
                    10.12.12.0/24   intercluster      40
```

11. Display the routes in the cluster.

    Note: No intercluster routes are available.

```
cluster01::> network routing-group route show
          Routing
Vserver   Group     Destination     Gateway         Metric
--------- --------- --------------- --------------- ------
cluster01
          c10.288.225.0/24
                    0.0.0.0/0       10.288.225.1    20
cluster01-01
          n10.288.225.0/24
                    0.0.0.0/0       10.288.225.1    10
cluster01-02
          n10.288.225.0/24
                    0.0.0.0/0       10.288.225.1    10
```

12. If communication between intercluster LIFs in different clusters requires routing, then create an intercluster route. The intercluster networks are scoped to each node; therefore, create an intercluster route on each node. In this example, `10.12.12.1` is the gateway address for the `10.12.12.0/24` network.

    Note: Creating a route is required only if the intercluster addresses in both clusters are not on the same subnet and a specific route is needed for communication between the clusters.

```
cluster01::> network routing-groups route create -server cluster01-01 -routing-group
i10.12.12.0/24 -destination 0.0.0.0/0 -gateway 10.12.12.1 -metric 40

cluster01::> network routing-groups route create -server cluster01-02 -routing-group
i10.12.12.0/24 -destination 0.0.0.0/0 -gateway 10.12.12.1 -metric 40
```

    Note: If the destination is specified as `0.0.0.0/0`, then it becomes the default route for the intercluster network.

13. Display the newly created routes.

```
cluster01::> network routing-group route show
          Routing
Vserver   Group     Destination     Gateway         Metric
```

```
--------- --------- --------------- --------------- ------
cluster01
          c10.288.225.0/24
                    0.0.0.0/0       10.288.225.1    20
cluster01-01
          n10.288.225.0/24
                    0.0.0.0/0       10.288.225.1    10
          i10.12.12.0/24
                    0.0.0.0/0       10.12.12.1      40
cluster01-02
          n10.288.225.0/24
                    0.0.0.0/0       10.288.225.1    10
          i10.12.12.0/24
                    0.0.0.0/0       10.12.12.1      40
```

Note: Although the intercluster routes do not have an assigned role, they are assigned to the routing group `i10.12.12.0/24`, which is assigned the role of intercluster, as shown in the output. These routes are only used for intercluster communication.

14. Repeat the steps in this section to configure intercluster networking in the other cluster. Dedicating ports for replication in one cluster does not require dedicating ports in all clusters; one cluster might use dedicated ports while the other cluster shares data ports for intercluster replication. In both configurations, verify that the ports have access to the proper subnets, VLANs, and so on.

3.9 Configuring Cluster Peers

After the nodes in both clusters are configured with intercluster LIFs, the clusters can be peered together to allow the creation of replication relationships between the clusters.

A single cluster can be peered with up to eight remote clusters. Each cluster peer relationship must be created separately. Once a peer relationship is created, mirroring can be performed in either direction between the two clusters in that peer relationship.

In this example, `cluster01` is peered with a remote cluster named `cluster02`. `Cluster02` is also a two-node cluster that has one intercluster LIF per node. The IP addresses of the intercluster LIFs created in `cluster02` are `10.12.12.3` and `10.12.12.4`, they are used in the following `cluster peer create` command example. However, if DNS is configured to resolve hostnames for the intercluster IP addresses, hostnames in the `-peer-addrs` option can be used. It is not likely that intercluster IP addresses frequently change; however, using hostnames allows intercluster IP addresses to be changed without having to modify the cluster peer relationship.

Before completing the following steps, replace the cluster name and IP addresses with those specific to your environment.

1. Run the `cluster peer create` command to peer the clusters, specifying the IP address of at least one intercluster LIF per node from the remote cluster in the `-peer-addrs` option, separated by commas. Provide the remote cluster administrator user name and password when prompted.

```
cluster01::> cluster peer create -peer-addrs 10.12.12.3,10.12.12.4 -username admin

Password: *********
```

Note: The credentials used for the cluster peer create operation are not stored in the remote system and are not required by the systems to maintain the peer relationship. They are used to authenticate the peer request and are sent in a Hash-based Message Authentication Code-MD5 (HMAC-MD5), not in clear text. Once authentication is complete and the cluster peer relationship has been created, the credentials are destroyed and are no long exchanged between the clusters.

2. Display the newly created cluster peer relationship.

```
cluster01::> cluster peer show -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
```

Note: The IP addresses used in the `-peer-addrs` option are listed as remote intercluster addresses. These addresses are used to discover all other intercluster IP addresses capable of communication between the clusters, listed as active IP addresses. The list of active IP addresses might be longer, depending on how many intercluster LIFs exist per node, because they are automatically discovered.

Best Practice

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 only one remote cluster address to be provided; however, in the event that the node hosting that address is down and it becomes unavailable, then the cluster peer relationship might not be rediscovered. Therefore, it is a best practice to use at least one intercluster IP address from each node in the remote cluster, so, in the event of a node failure, the peer relationship remains stable.

3. Review the health of the cluster peer relationship.

```
cluster01::> cluster peer health show

Node       cluster-Name                Node-Name
             Ping-Status               RDB-Health Cluster-Health  Available
---------- --------------------------- ---------  --------------- ---------
cluster01-01
           cluster02                   cluster02-01
             Data: interface_reachable
             ICMP: interface_reachable true       true            true
                                       cluster02-02
             Data: interface_reachable
             ICMP: interface_reachable true       true            true
cluster01-02
           cluster02                   cluster02-01
             Data: interface_reachable
             ICMP: interface_reachable true       true            true
                                       cluster02-02
             Data: interface_reachable
             ICMP: interface_reachable true       true            true
```

3.10 Intercluster SnapMirror Throttle

To limit the amount of bandwidth used by intercluster SnapMirror, apply a throttle to intercluster SnapMirror relationships. When creating a new relationship, a throttle can be set through the command line by adding the `-throttle` option and a value in kilobytes, by modifying an existing relationship with the `snapmirror modify command`. NetApp OnCommand® System Manager 3.0 does not currently support SnapMirror throttle management. 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
```

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 SnapMirror relationships, which use the cluster interconnect, do not allow a throttle to be set. But, starting clustered Data ONTAP 8.2.1, intra-cluster throttle is supported, and it works exactly the same way as inter-cluster throttle.

3.11 Firewall Requirements for Intercluster SnapMirror

Open the following ports on the intercluster network between all source and destination nodes for intercluster replication:

- Port 11104
- Port 11105

Note: Clustered Data ONTAP uses port 11104 to manage intercluster communication sessions; it uses port 11105 to transfer data.

4 Storage Virtual Machine Peering

Storage Virtual Machine peering is the act of connecting two Storage Virtual Machines to allow replication to occur between them (starting in clustered Data ONTAP 8.2). Cluster peering must be configured to allow any replication to occur between different clusters. In clustered Data ONTAP 8.1 any Storage Virtual Machine could replicate data to any other Storage Virtual Machine in the same cluster or any cluster peer. Control of replication security could only be maintained at a clusterwide level. Starting in clustered Data ONTAP 8.2, more granularity in SnapMirror security is provided. Replication permission must be defined by peering Storage Virtual Machines together.

Before creating any SnapMirror relationships between a pair of Storage Virtual Machines, you must have a Storage Virtual Machine peer relationship between the pair of Storage Virtual Machines. These Storage Virtual Machines can be local (intracluster) or remote (intercluster). Storage Virtual Machine peering is a permission-based mechanism and is a one-time operation that must be performed by the cluster administrators. The following steps are required to configure replication between different Storage Virtual Machines:

1. Have a clear understanding of Storage Virtual Machine peering.
2. Peer the Storage Virtual Machines together.
3. Create NetApp SnapMirror relationships between different Storage Virtual Machines.

Best Practice
Name a Storage Virtual Machine with a unique fully qualified domain name (FQDN): for example, dataVserser.HQ or mirrorVserver.Offsite. Storage Virtual Machine peering requires unique Storage Virtual Machine names, and using FQDN naming style makes it much easier to make sure of uniqueness.

4.1 Storage Virtual Machine Peer Requirements

Storage Virtual Machine peer requirements include the following:

- A cluster peering relationship must exist before any Storage Virtual Machine peer relationships involving two clusters can be created. This is not required if the Storage Virtual Machines reside in the same cluster.
- If Storage Virtual Machines are on different clusters, then execute "vserver peer create" and "vserver peer accept". If Storage Virtual Machines are on the same cluster, then execute ONLY "vserver peer create".
- Storage Virtual Machine names involved in Storage Virtual Machine peering relationships must be unique.
- The languages of the two Storage Virtual Machines must be the same.

4.2 Configuring Storage Virtual Machine Peers

Storage Virtual Machine peering infrastructure forms the basis for delegation of SnapMirror relationship management creation/use/deletion to the Storage Virtual Machine administrator.

```
cluster01::> vserver peer create -vserver vs1_src -peer-vserver vs1_dest -applications
snapmirror -peer-cluster cluster02
```

Storage Virtual Machine name must be unique across clusters.

```
cluster01::> vserver peer show
            Peer        Peer
Vserver     Vserver     State
----------- ----------- ------------
vs1_src     vs1_dest    initiated
```

After the Storage Virtual Machine peer relationship has been created, you will have to accept the request on the destination cluster to complete the Storage Virtual Machine peering. Using `vserver peer show` will allow you to see any pending requests and verify that `vserver peer create` worked.

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

cluster02::> vserver peer accept -vserver vs1_dest -peer-vserver vs1_src

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

5 SnapMirror Data Protection Relationships

Clustered Data ONTAP 8.1 onward provides two types of SnapMirror relationships: DP mirrors and load-sharing (LS) mirrors. DP mirrors are discussed in this section; LS mirrors are discussed in a later section.

DP mirrors can be performed as intercluster or intracluster.

- **Intercluster DP mirrors.** Replication between volumes in two different Storage Virtual Machines in different clusters operating in clustered Data ONTAP. They are primarily used for providing DR to another site or location.
- **Intracluster DP mirrors.** Replication between two volumes in different Storage Virtual Machines in the same cluster, or between two volumes in the same Storage Virtual Machine. They are primarily used for maintaining a local backup copy.

DP mirror relationships have the same characteristics regardless of whether intracluster or intercluster is being replicated. These characteristics include:

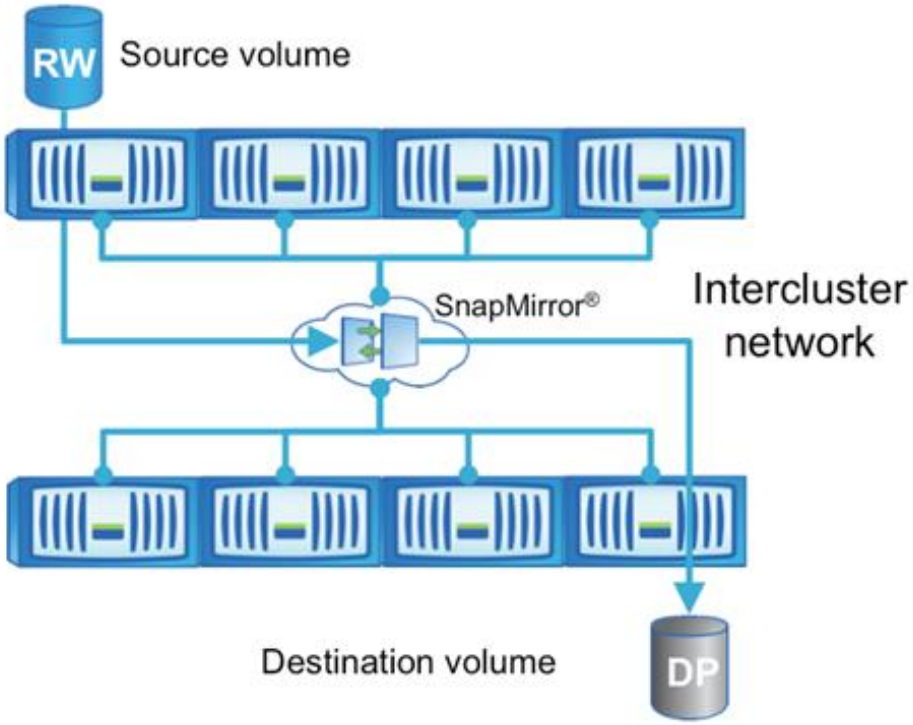
- DP mirror relationships are created and managed on the destination cluster.
- DP mirror relationship transfers are triggered by the scheduler in the destination cluster.
- Each DP mirror destination volume is a separate SnapMirror relationship that is performed independently of other DP mirror volumes; however, the same clustered Data ONTAP schedule entry can be used for different DP mirror relationships.
- Destination volumes for both DP- and LS-type mirrors must be created with a volume type (`-type` option) of DP. The storage administrator cannot change the volume `-type` property after the volume has been created.
- DP mirror destination volumes are read-only until failover.
- DP mirror destination volumes can be failed over using the SnapMirror break operation, making the destination volume writable. The SnapMirror break must be performed separately for each volume.
- DP mirror destination volumes can be mounted into a Storage Virtual Machine namespace while still read-only, but only after the initial transfer is complete.
- An intercluster DP mirror destination volume cannot be mounted in the same namespace as the source volume, because intercluster DP mirror relationships are to a different cluster and therefore to a different Storage Virtual Machine, which is a different namespace.
- An intracluster DP mirror destination volume can be mounted in the same namespace as the source volume if both the source and destination volumes exist in the same Storage Virtual Machine; 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 clustered Data ONTAP command line interface (CLI), NetApp OnCommand System Manager 3.0, and NetApp OnCommand Unified Manager 6.0.
- 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.

Clustered Data ONTAP 8.2 onward provides an additional SnapMirror relationship: XDP vault. For more information on SnapVault® in clustered Data ONTAP 8.2, refer to TR-4183.

Networks Used for SnapMirror Data Protection Relationships

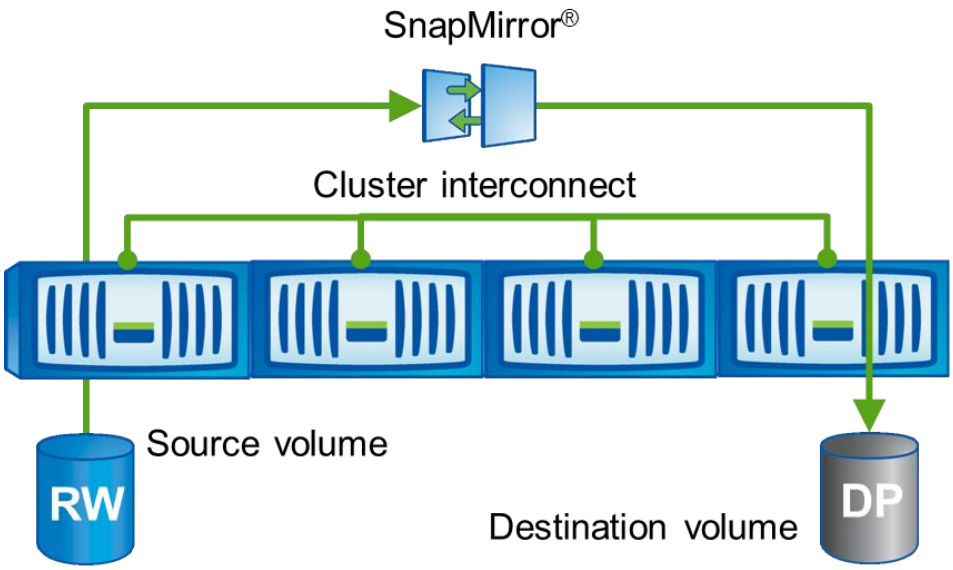
Intercluster and intracluster DP SnapMirror relationships are different based on the network that is used for sending data. Intercluster DP SnapMirror relationships use the intercluster network defined by intercluster LIFs. Figure 21 illustrates an intercluster network for SnapMirror.

Figure 21) Intercluster network for SnapMirror.



Intracluster DP mirror relationships use the cluster interconnect, which is the private connection used for communication between nodes in the same cluster. Figure 22 illustrates a cluster interconnect for intercluster SnapMirror.

Figure 22) Cluster interconnect for intercluster SnapMirror.



5.1 SnapMirror Data Protection Relationships

After the cluster peer relationship and Storage Virtual Machine peer relationship have been successfully created between the two clusters, create the intercluster SnapMirror relationships. A peer relationship is not required to mirror data between two Storage Virtual Machines in the same cluster or between two volumes in the same Storage Virtual Machine.

Both the source and destination Storage Virtual Machines must have the same language type setting to be able to replicate between them. A Storage Virtual Machine language type cannot be changed after it has been created.

Intercluster SnapMirror relationships are primarily used to provide DR capability in another site or location. If all necessary volumes have been replicated to a DR site with SnapMirror, then a recovery can be performed so that operations can be restored from the DR site.

The creation of SnapMirror relationships in clustered Data ONTAP does not depend on Storage Virtual Machine hostname to IP address resolution. While the cluster names are resolved through the peer relationship, the Storage Virtual Machine names are internally resolved through the clusters. The host names of the source and destination Storage Virtual Machine and cluster are used to create SnapMirror relationships in clustered Data ONTAP; it is not necessary to use the IP address of a LIF.

Intercluster SnapMirror Requirements

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 Storage Virtual Machine that has the same language type as the source Storage Virtual Machine; volumes cannot exist in clustered Data ONTAP without a Storage Virtual Machine.
- Configure the source and destination Storage Virtual Machine in a peer relationship.
- Create a destination volume with a type of 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.

Storage Virtual Machine Fan-Out and Fan-In

It is possible to fan out or fan in volumes between different Storage Virtual Machines. For example, multiple different volumes from a single Storage Virtual Machine in the source cluster might be replicated with each volume replicating into a different Storage Virtual Machine in the destination cluster, referred to as fan-out. Alternatively, multiple different volumes might also be replicated, each existing in a different Storage Virtual Machine in the source cluster, to a single Storage Virtual Machine in the destination cluster, referred to as fan-in.

Best Practice

When replicating to provide DR capabilities, mirror all required volumes from a given Storage Virtual Machine in the source cluster to a particular matching Storage Virtual Machine in the destination cluster. Design considerations that determine that a given set of volumes should reside in the same Storage Virtual Machine should also apply to keeping those same volumes in a like Storage Virtual Machine at a DR site. In order for different volumes to be accessible in the same namespace, they must exist in the same Storage Virtual Machine (a Storage Virtual Machine is a namespace).

Volume Fan-Out and Fan-In

For SnapMirror DP relationships, a single NetApp FlexVol volume can be replicated to up to five different destination volumes. Each destination volume can exist in a different Storage Virtual Machine or all can exist in the same Storage Virtual Machine; this is referred to as volume fan-out. Volume fan-in, which is replication of multiple different volumes into the same destination volume, is not possible.

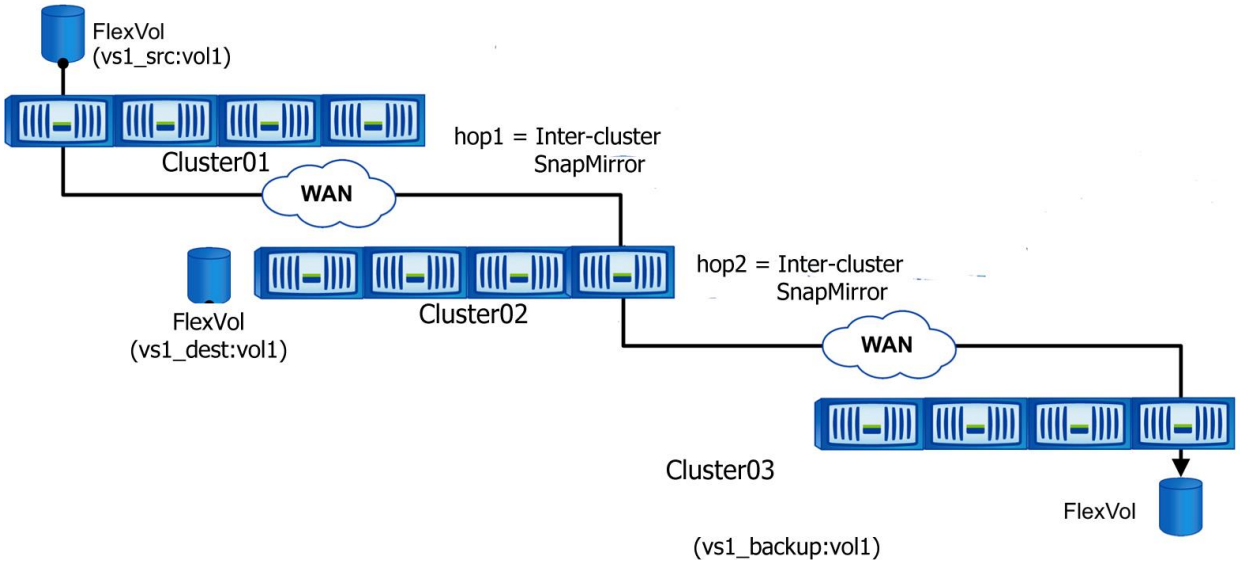
Cascade Relationships or Multihop Replication

Starting in clustered Data ONTAP 8.2, SnapMirror relationships can be cascaded. However, only one of the relationships in the cascade configuration can be a SnapVault relationship.

Cascading is defined as replicating from established replicas. Suppose there are three storage systems, A, B, and C. Replicating from A to B and from B to C is considered a cascade configuration.

An example cascade configuration with two hops is shown in Figure 23.

Figure 23) Cascaded volume replication using SnapMirror.



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 allow updating that data uniformly at regular intervals.

Note: Snapshot copy behaviors:

1. SnapMirror creates a soft lock on the Snapshot copy of the source volume (`snapmirror` tag).
2. Destination system carries an extra Snapshot copy.

Dual-Hop Volume SnapMirror

This configuration involves volume SnapMirror replication among three clusters.

vs1_src:vol1 → vs1_dest:vol1 → vs1_backup:vol1

Note: 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.

Table 1) Snapshot copy propagation for dual-hop volume SnapMirror.

Timeline	Snapshot Copies on cluster01	Snapshot Copies on cluster02	Snapshot Copies on cluster03
1) After volume initialization on cluster02	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1	
2) Volume SnapMirror update on cluster02	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1 snapmirror.cd20c2a0v2	
3) After volume initialization on cluster03	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1 snapmirror.cd20c2a0v2	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1 snapmirror.cd20c2a0v2
4) Volume SnapMirror update on cluster02	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2 snapmirror.cd20c2a0v3	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2 snapmirror.cd20c2a0v3	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1 snapmirror.cd20c2a0v2
5) Volume SnapMirror update on cluster03	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2 snapmirror.cd20c2a0v3	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2 snapmirror.cd20c2a0v3	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2 snapmirror.cd20c2a0v3

Snapshot copy behaviors to note:

1. There is an extra Snapshot copy on cluster02 (destination) after the first SnapMirror update (step 2).
2. Cluster03 also has the same number of Snapshot copies as cluster02 after step 3 because there is a volume SnapMirror relationship between cluster02 and cluster03 systems.
3. A new soft lock exists on cluster02 after step 3 because cluster02 is now the volume SnapMirror source for cluster03.
4. After step 4, the source cluster, cluster01, contains two SnapMirror Snapshot copies. This is because the Snapshot copy 'snapmirror.cd20c2a0v2' is locked by cluster02 as it is required to continue to perform SnapMirror updates with cluster03. This Snapshot copy on cluster01 system is also used to perform SnapMirror resync with cluster03 system in case cluster02 system meets disaster.
5. After an update is performed on cluster03 (step 5), the soft lock now exists on the latest SnapMirror Snapshot copy 'snapmirror.cd20c2a0v3' because this is the new baseline SnapMirror Snapshot copy between cluster02 and cluster03 systems.

Seeding Intercluster SnapMirror Relationships

The term seeding refers to the initial transfer of data for a newly created SnapMirror relationship. In clustered Data ONTAP, the initial transfer for all SnapMirror relationships must be performed over the network.

When a new SnapMirror relationship is created using the `snapmirror create` command, an initial transfer is not automatically performed. The `create` command simply establishes the relationship and the metadata that defines it. Follow the `snapmirror create` command with the `snapmirror initialize` command to perform the initial transfer. Alternatively, use the `snapmirror initialize` command alone to perform the initial transfer as soon as the relationship is created. If the SnapMirror relationship does not exist, then the `initialize` command creates the relationship and performs the initial transfer.

NetApp OnCommand System Manager 3.0 provides the option of initializing a relationship using the SnapMirror Relationship Create Wizard. Managing SnapMirror with System Manager is described later in this document.

5.2 Scheduling SnapMirror Updates

Clustered Data ONTAP has a built-in scheduling engine similar to cron. Periodic replication updates in clustered Data ONTAP can be scheduled 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. This 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
```

Schedules can also be managed and applied to SnapMirror relationships using NetApp OnCommand System Manager 3.0.

5.3 Converting a SnapMirror Relationship to a SnapVault Relationship

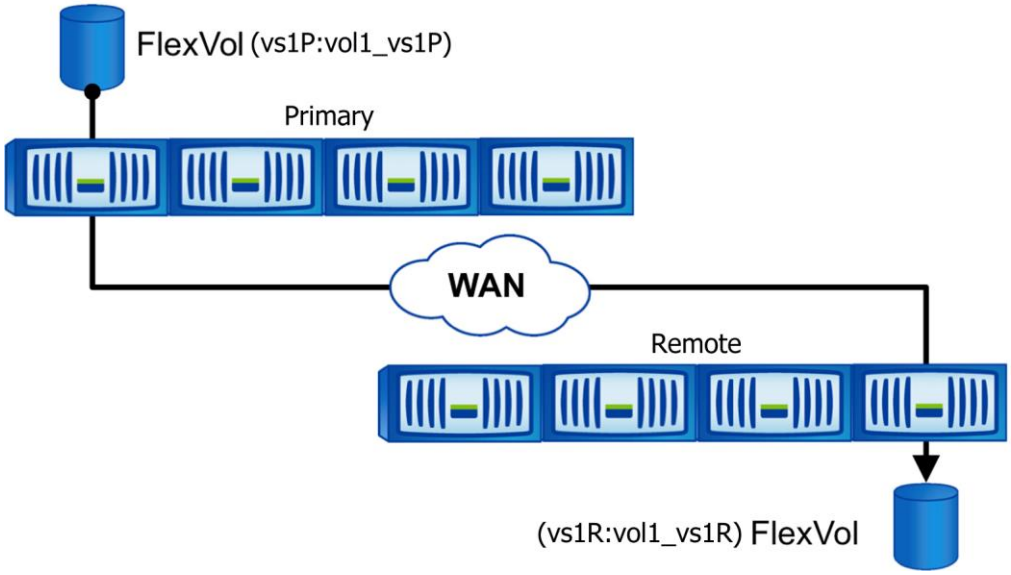
One scenario where you would want to convert an existing SnapMirror relationship to a SnapVault relationship: An existing customer using SnapMirror in clustered Data ONTAP 8.1 wants to make use of SnapVault in clustered Data ONTAP 8.2 for longer retention.

Upgrade your source and destination clusters to clustered Data ONTAP 8.2. Your existing SnapMirror relationships will continue to remain cluster scope and will behave as they did in clustered Data ONTAP 8.1. They will not benefit from the scalability improvements unless they are deleted and recreated. However, both clustered Data ONTAP 8.1 and clustered Data ONTAP 8.2 use the block-level engine for mirrors, and it is important to note that no rebaseline will be required, only resync.

Here are the details based on an example as shown in Figure 24:

Cluster peering and Storage Virtual Machine peering have already been done.

Figure 24) Conversion of a SnapMirror relationship to a SnapVault relationship.



It consists of the following steps:

1. Delete mirror (DR) relationship.
2. Break the mirror destination.
3. Create a XDP (vault) relationship between the same endpoints.
4. Perform resync between the endpoints. This will convert a DR destination to a vault destination without having to do a rebaseline.

Create a Volume on the Primary Cluster

```
Primary::> vol create -vserver vs1P -volume vol1_vs1P -aggregate aggr1_Primary_01
-size 10GB (volume create)
[Job 81] Job succeeded: Successful
```

Create a DP Volume on the Remote Cluster

```
Remote::> vol create -vserver vs1R -volume vol1_vs1R -aggregate aggr1_Remote_01
-size 10GB -type DP              (volume create)
[Job 81] Job succeeded: Successful
```

Create a SnapMirror Relationship Between the Volumes on the Primary and the Remote Clusters

```
Remote::> snapmirror create -source-path vs1P:vol1_vs1P -destination-path
vs1R:vol1_vs1R -type DP -schedule daily
Operation succeeded: snapmirror create the relationship with destination
vs1R:vol1_vs1R.
```

```
Remote::> snapmirror show
                                                                    Progress
Source            Destination  Mirror  Relationship  Total            Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
vs1P:vol1_vs1P
            DP   vs1R:vol1_vs1R
                              Uninitialized
                                      Idle           -         true    -
```

```
1 entries were displayed.
```

Initialize the SnapMirror Relationship

```
Remote::> snapmirror initialize -destination-path vs1R:vol1_vs1R
Operation is queued: snapmirror initialize of destination vs1R:vol1_vs1R.

Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
vs1P:vol1_vs1P
            DP   vs1R:vol1_vs1R
                              Snapmirrored
                                      Idle           -         true    -
1 entries were displayed.
```

Conversion of SnapMirror to SnapVault

1. SnapMirror Delete

```
Remote::> snapmirror delete -destination-path vs1R:vol1_vs1R
Operation succeeded: snapmirror delete the relationship with destination
vs1R:vol1_vs1R.
```

2. SnapMirror Break

```
Remote::> snapmirror break -destination-path vs1R:vol1_vs1R
[Job 128] Job succeeded: SnapMirror Break Succeeded
```

3. SnapVault Create

```
Remote::> snapmirror create -source-path vs1P:vol1_vs1P -destination-path
vs1R:vol1_vs1R -type XDP
Operation succeeded: snapmirror create the relationship with destination
vs1R:vol1_vs1R.
```

```
Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
vs1P:vol1_vs1P
            XDP  vs1R:vol1_vs1R
                              Broken-off
                                      Idle           -         true    -
```

4. SnapMirror Resync

```
Remote::> snapmirror resync -destination-path vs1R:vol1_vs1R

Warning: All data newer than Snapshot copy
         snapmirror.3fd9730b-8192-11e2-9caa-123478563412_2147484699.2013-02-28_1
         10732 on volume vs1r:vol1_vs1r 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
```

```
Remote::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
vs1P:vol1_vs1P
            XDP  vs1R:vol1_vs1R
                              Snapmirrored
                                      Idle           -         true    -
```

After completing the preceding steps, you would adjust the schedules and policies accordingly to keep desired Snapshot copies on the vault destination. Also, you cannot make a SnapVault destination volume read/write for use as a DR volume.

6 Managing SnapMirror Data Protection Relationships with NetApp OnCommand System Manager

NetApp OnCommand System Manager 3.0 can be used for creating and managing SnapMirror DP relationships. System Manager includes a wizard used to create SnapMirror DP relationships, create schedules to assign to relationships, and create the destination volume, all within the same wizard.

However, the capability to create and manage LS mirrors and manage SnapMirror throttle settings is not available in System Manager 3.0.

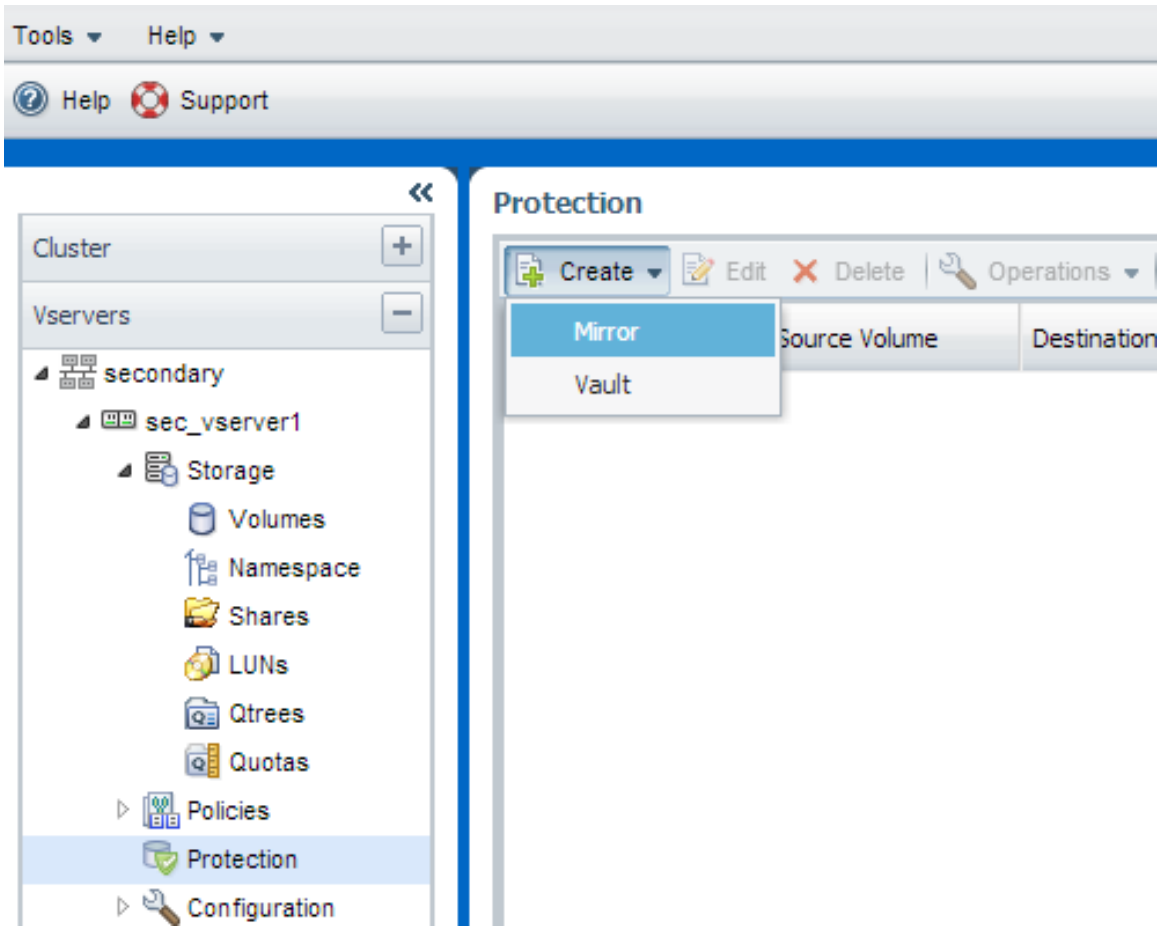
Note: SnapMirror relationships operating in clustered Data ONTAP must be managed by a cluster administrator; administration cannot be delegated to a Storage Virtual Machine administrator. Starting with clustered Data ONTAP 8.2, a cluster administrator can delegate the management of SnapMirror relationships to Storage Virtual Machine administrator.

6.1 Creating a SnapMirror Relationship in System Manager

This section describes how to create a SnapMirror relationship using System Manager. In this example, a new relationship is created to mirror volume `vol1` from Storage Virtual Machine `vs1_src` in cluster `Cluster01` to Storage Virtual Machine `vs1_dest` in cluster `Cluster02`.

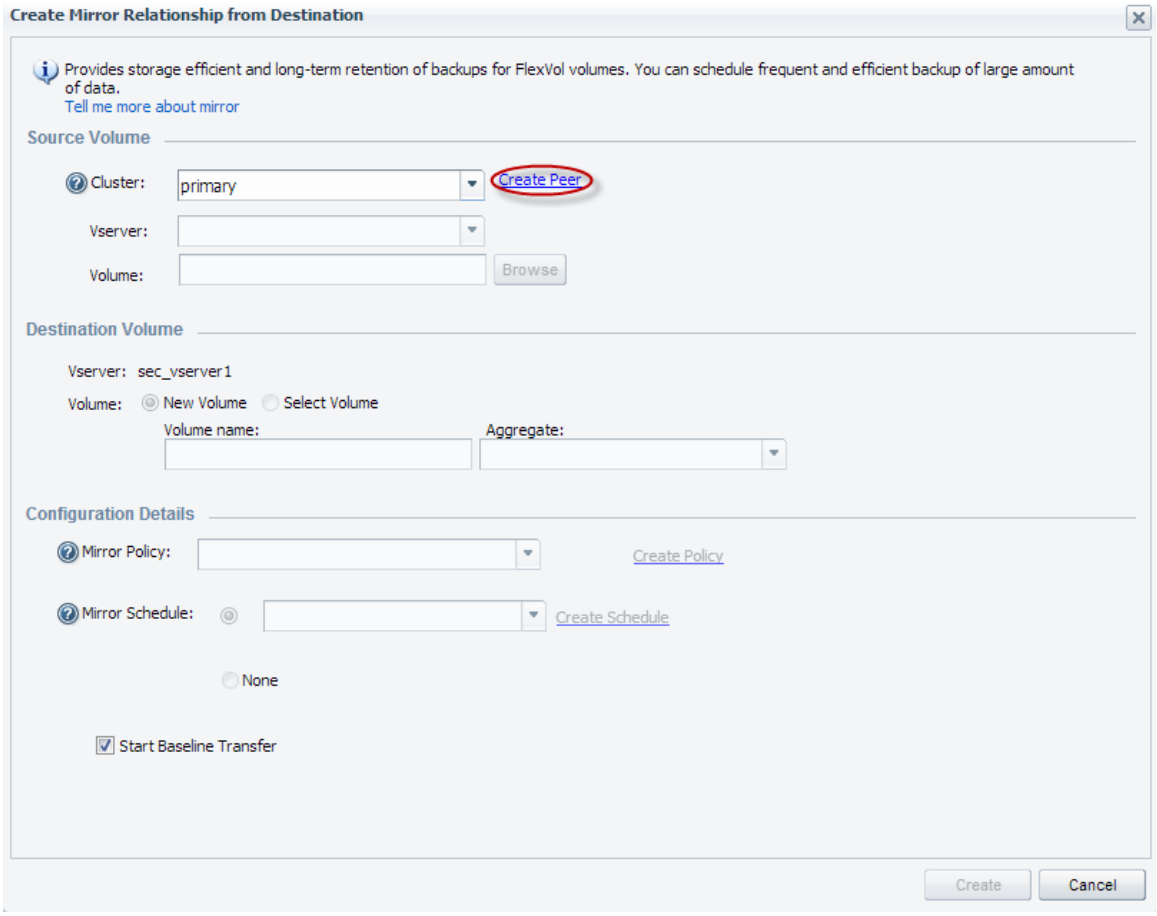
1. In System Manager 3.0, a new relationship can be created only from the destination cluster. In this example, the destination Storage Virtual Machine named `vs1_dest` is selected. Click Vserver > Protection > and then click Create.

Figure 25) Create SnapMirror relationship from destination - select Mirror.



2. Using System Manager, a relationship can be created only from the destination cluster; therefore, identify the Source cluster `cluster01`.

Figure 26) Create SnapMirror relationship from destination - select Source cluster.



3. Next, select the source cluster. If credentials for the source cluster have already been stored in System Manager, or if System Manager is already logged into that cluster, then the cluster credentials are automatically entered. Otherwise, enter the source cluster credentials. If cluster peering is not established, peer the clusters.

Figure 27) Create SnapMirror relationship from destination – cluster peering.

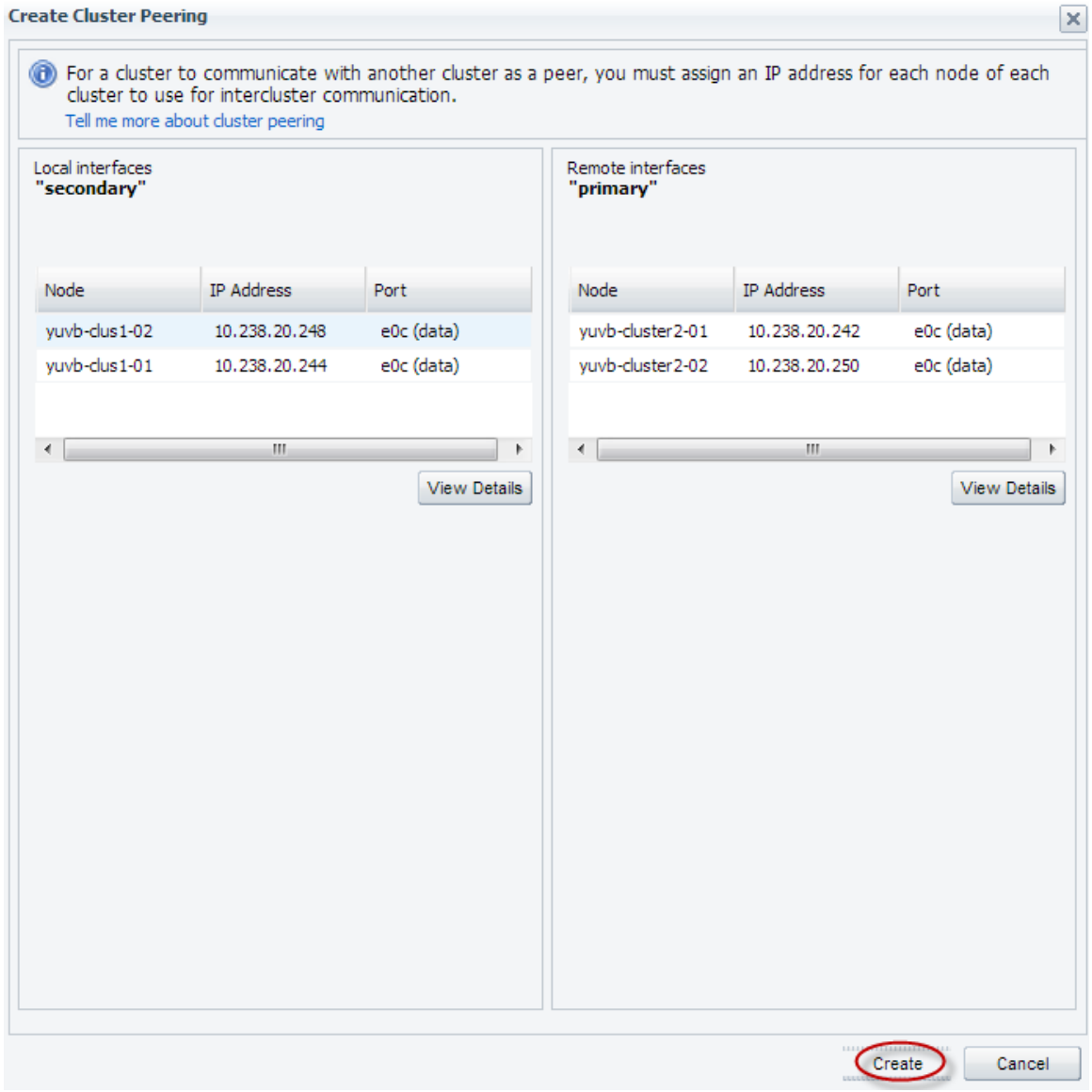
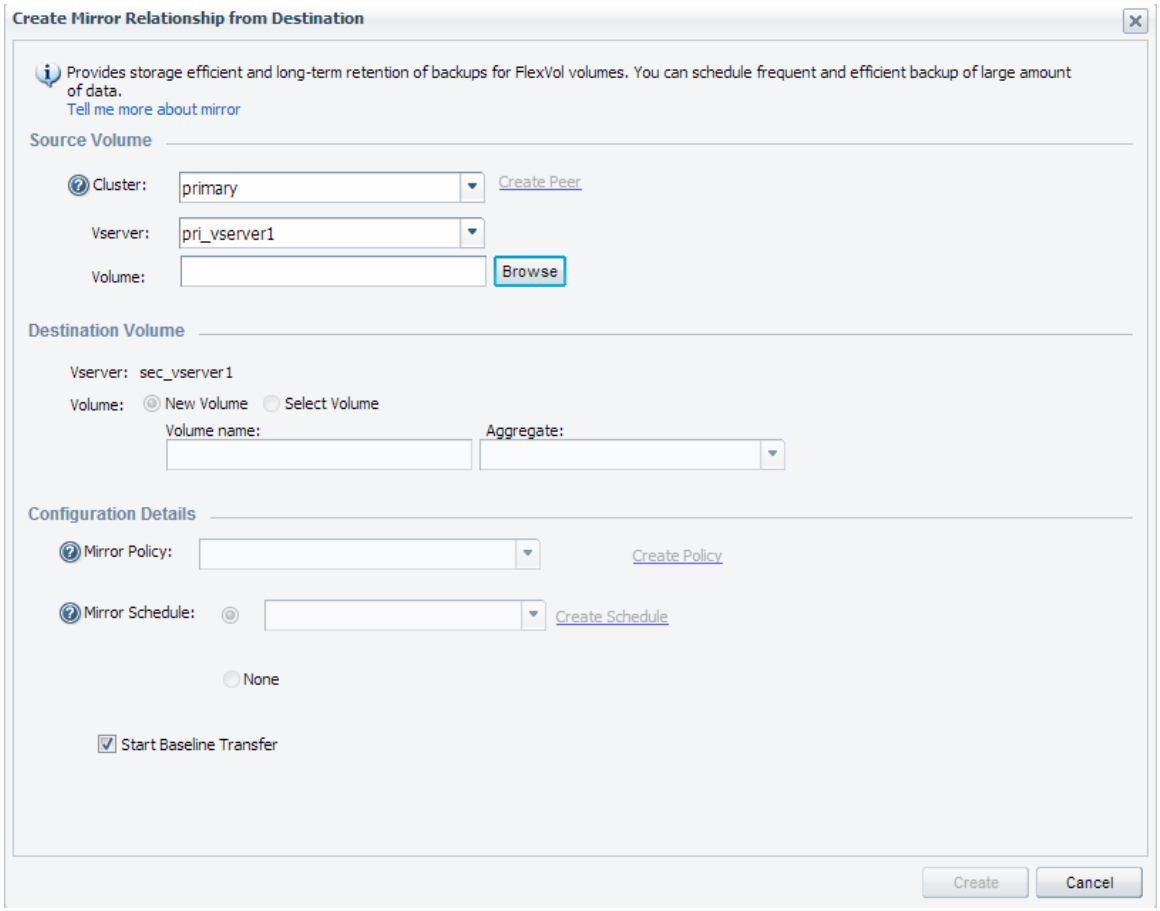
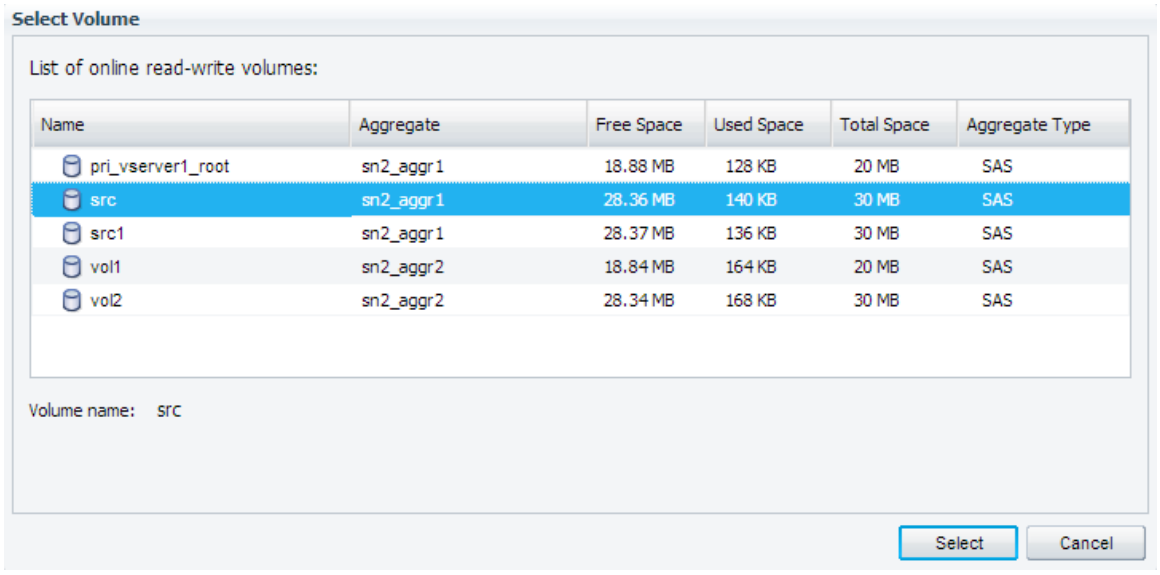


Figure 28) Create SnapMirror relationship from destination - select the source Storage Virtual Machine.



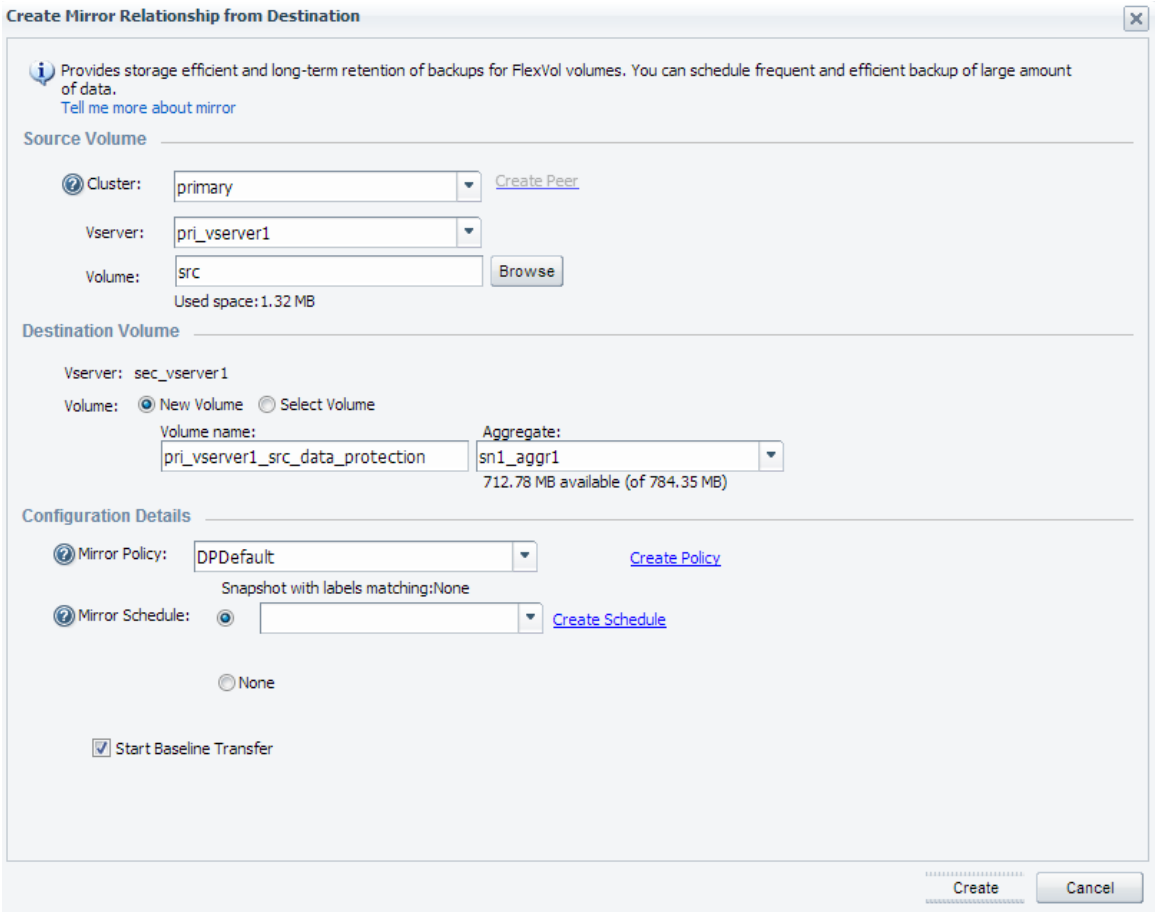
4. Select a source Storage Virtual Machine. If the source Storage Virtual Machine is not peered with the destination Vserver, then System Manager 3.0 will peer the two Storage Virtual Machine.

Figure 29) Create SnapMirror relationship from destination - select the source Volume.



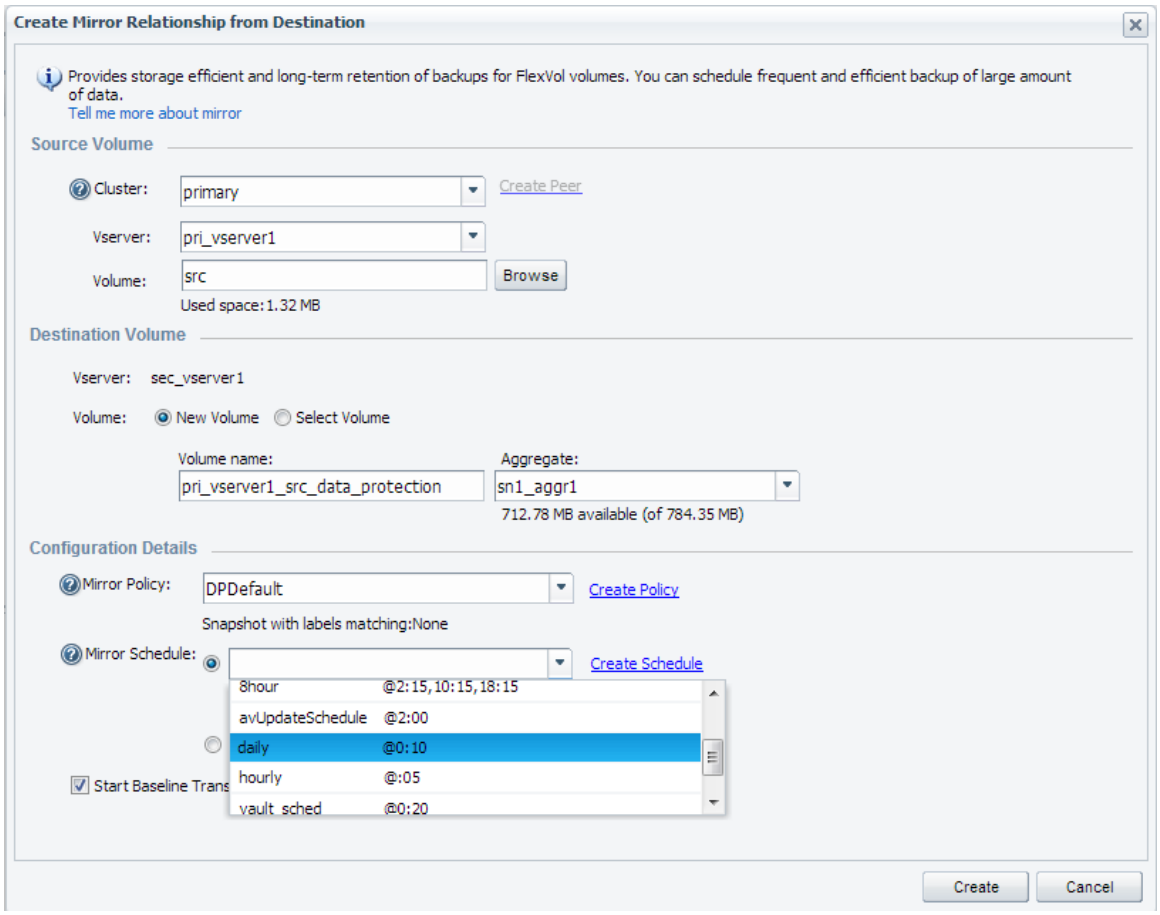
5. Select an existing volume on the source Storage Virtual Machine.

Figure 30) Create SnapMirror relationship from destination - select the destination Volume.



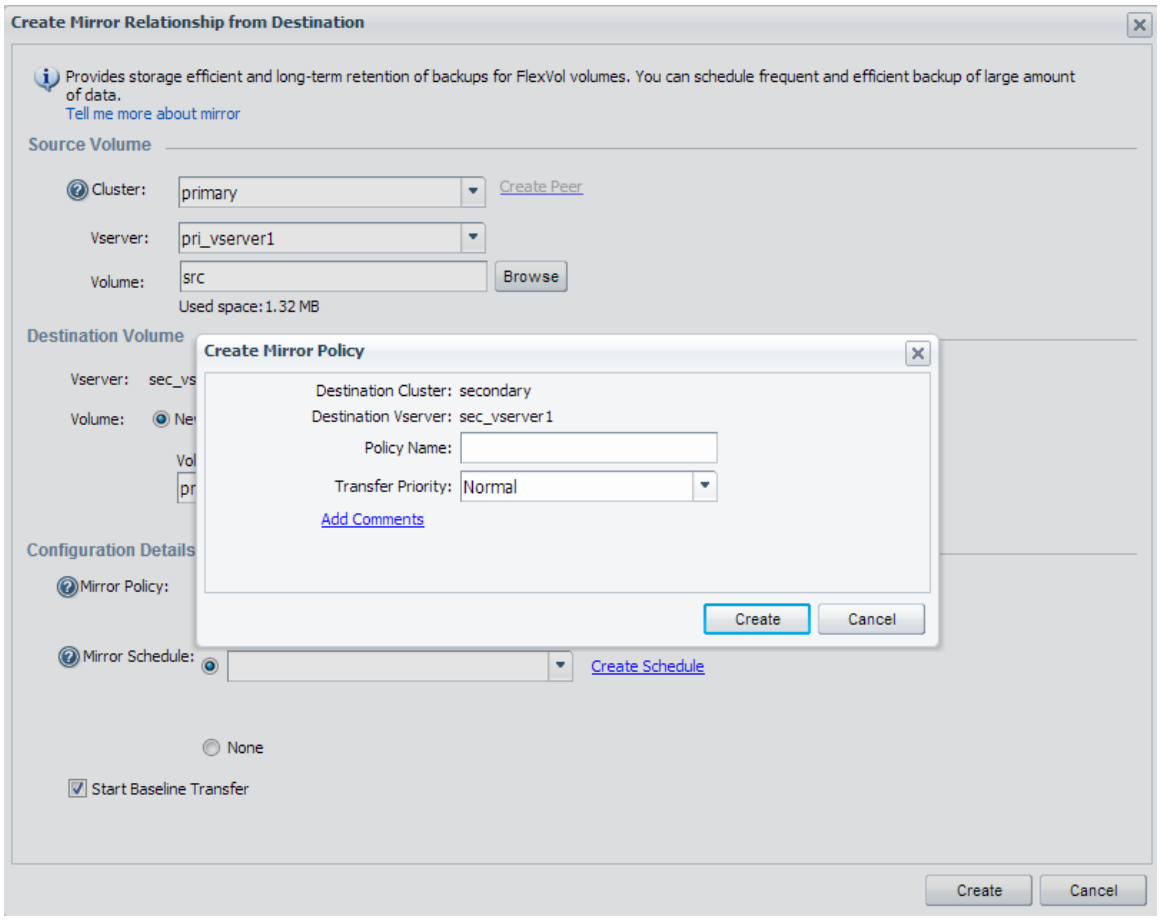
6. Select the destination Volume on the destination Storage Virtual Machine or create a destination volume on the destination Storage Virtual Machine.

Figure 31) Create SnapMirror relationship from destination - select or create SnapMirror policy and schedule.



7. Select an existing SnapMirror policy or create a new policy without having to leave the wizard (default policy is *DPDefault*). Select an existing SnapMirror schedule or create a new schedule.

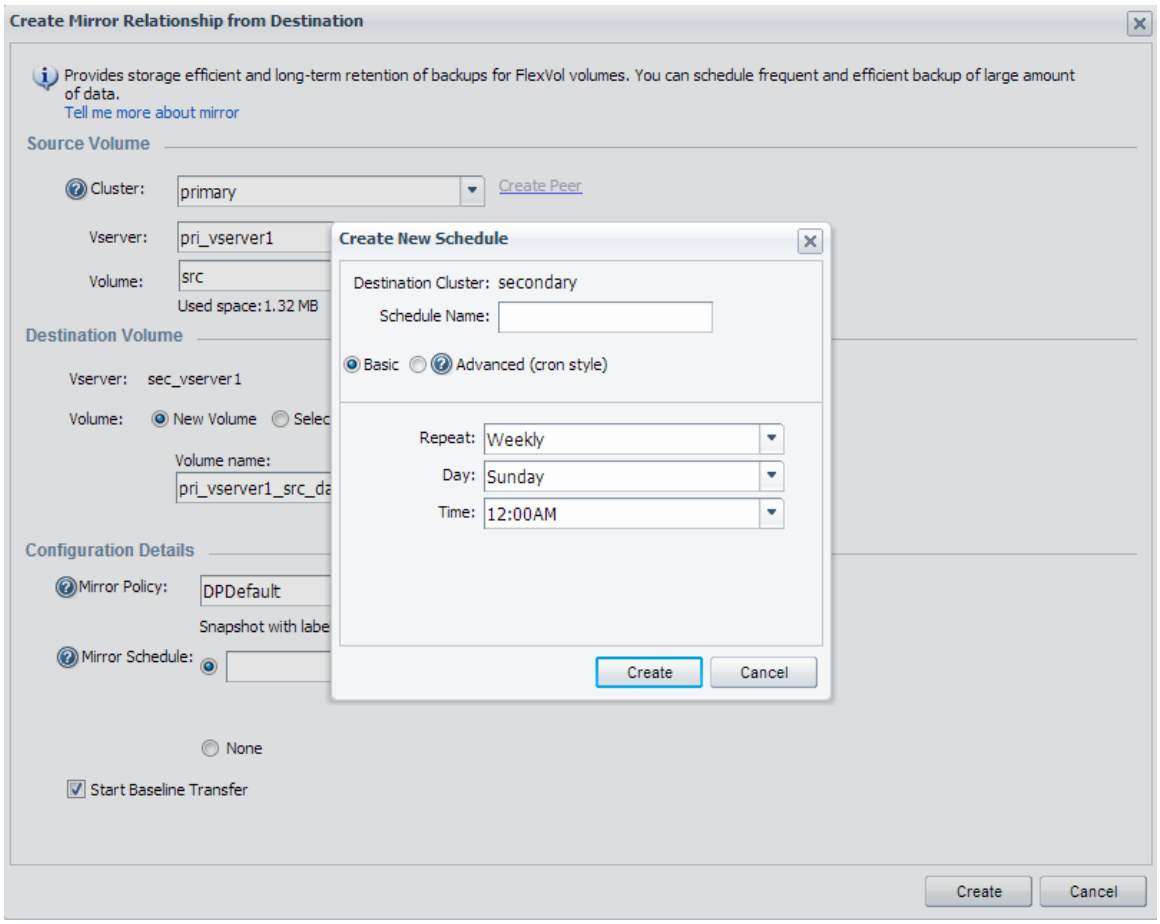
Figure 32) Create SnapMirror relationship from destination - create a new SnapMirror policy.



8. Create a new SnapMirror policy. Transfer priority has two levels – normal and low (default is normal). Transfer priority can be changed at any time; it affects the next operation. It comes into effect when there are enough operations pending on a node such that the meter* is filled. Low-priority operations get delayed by few minutes even if the meter is empty. They are delayed by one minute.

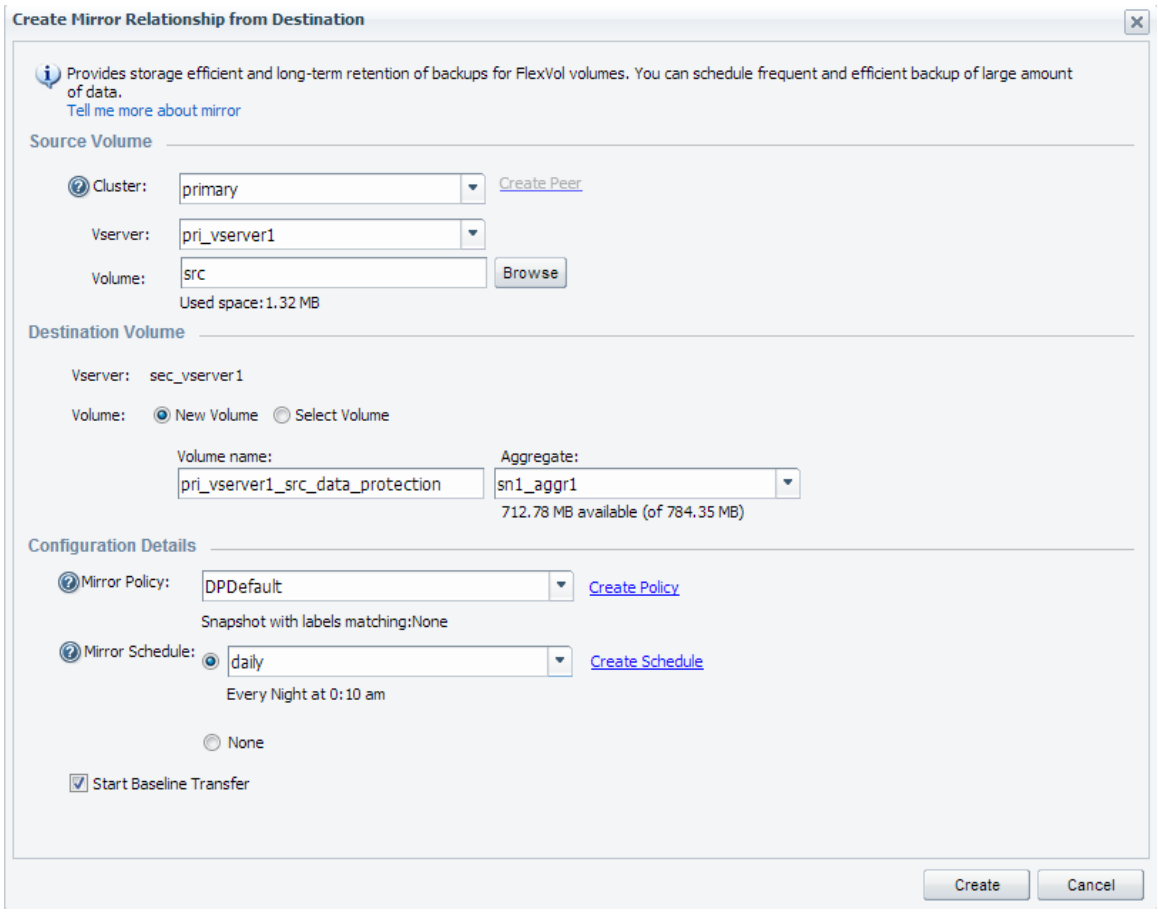
   Meter* - value depends on the platform and memory. The value for high-end platforms with more than 8GB of memory is 100 and for low-end and midrange platforms with less than or equal to 8GB of memory is 20. Part of the meter can be reserved for SnapMirror and part for SnapVault. By default there is no reservation.

Figure 33) Create SnapMirror relationship from destination - create a new SnapMirror schedule.



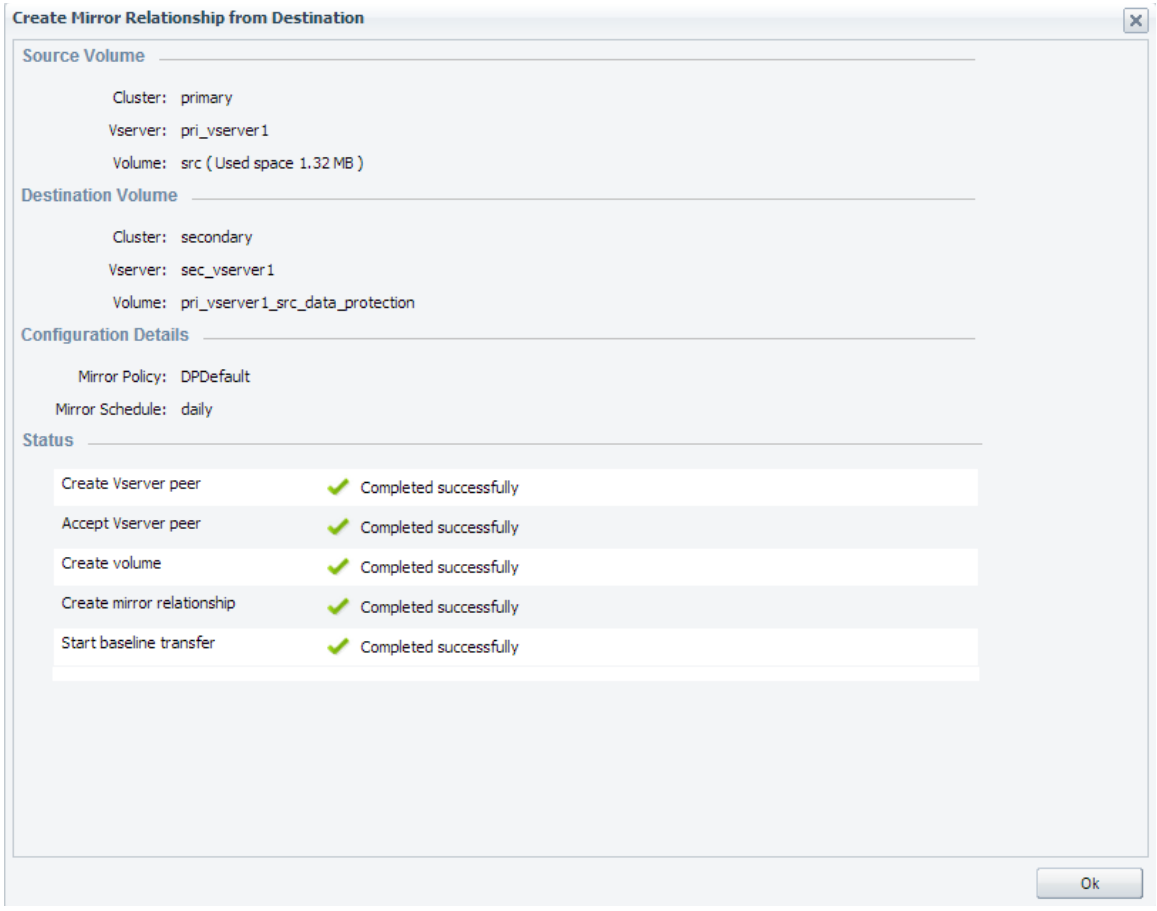
9. Create a new SnapMirror Schedule.

Figure 34) Create SnapMirror relationship from destination – start Baseline transfer.



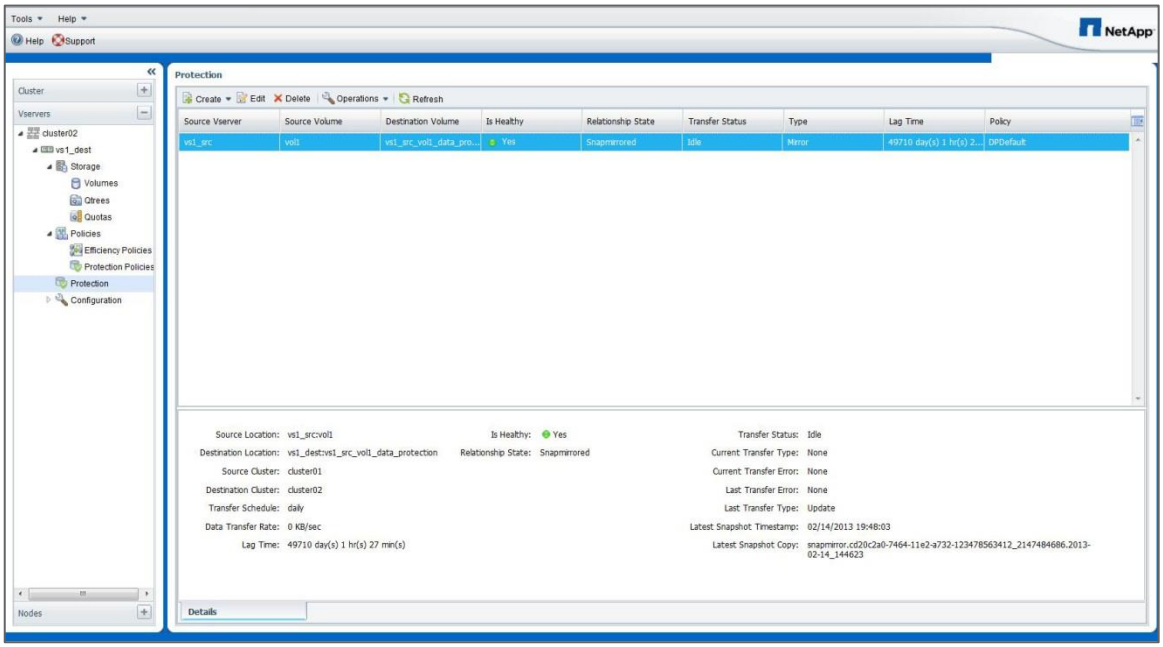
10. To automatically start the SnapMirror initialization (initial baseline copy) after the relationship is created, check the Start Baseline Transfer checkbox.

Figure 35) Create SnapMirror relationship from destination – summary of SnapMirror relationship configuration and status.



11. The wizard displays a summary of the SnapMirror relationship configurations, and the status of the SnapMirror relationship.

Figure 36) SnapMirror Baseline transfer details.

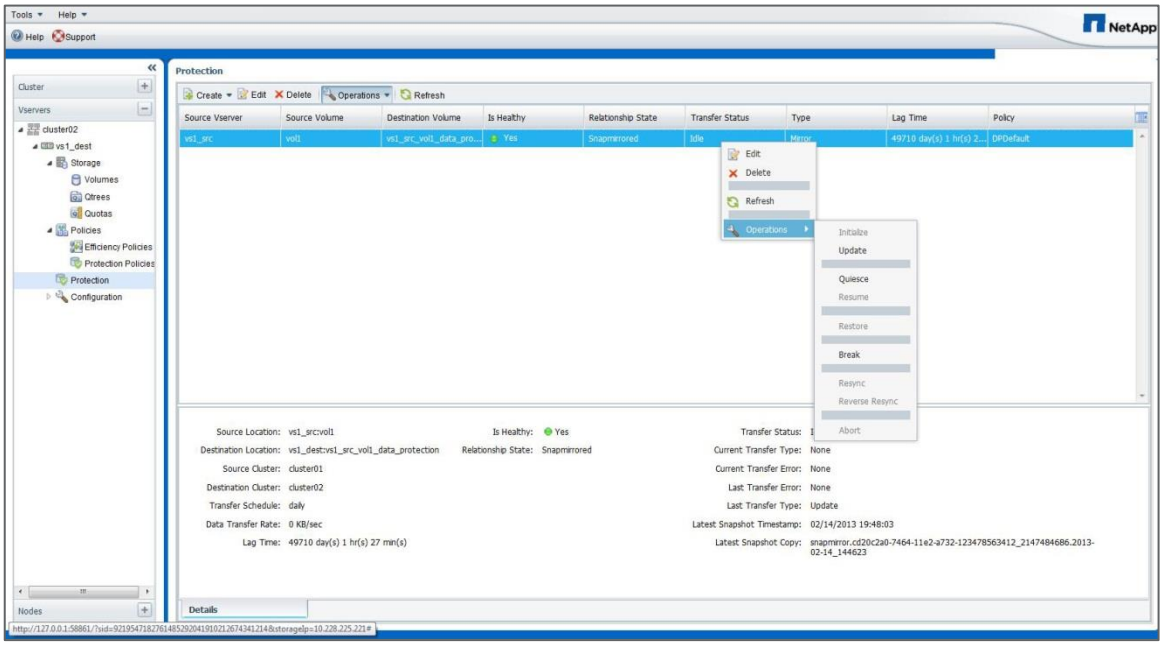


After the SnapMirror Relationship Create Wizard completes, the SnapMirror window opens. Click Refresh, if needed, after the initial baseline transfer is complete to view the finished status.

6.2 Managing SnapMirror Relationships with System Manager

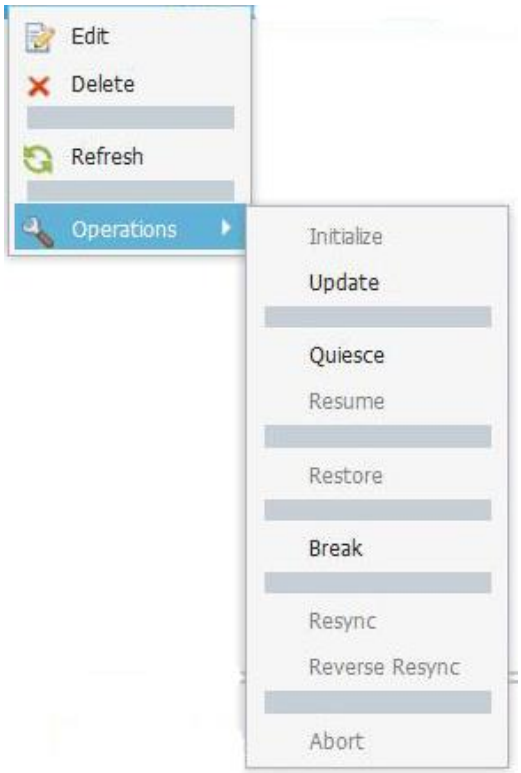
To manage SnapMirror DP relationships in System Manager, click the Operations menu at the top of the SnapMirror screen as shown in Figure 37 or right-click on a specific SnapMirror relationship, and open a context menu. Only operations that are currently allowed for that SnapMirror relationship are enabled in the context menu.

Figure 37) SnapMirror relationships list.



The context menu provides several other options. Grayed-out options are not available based on the current state of the selected SnapMirror relationship. Figure 38 shows all available operations that can be performed in the System Manager context menu.

Figure 38) Systems Manager Context Menu.



The operations listed in the SnapMirror context menu perform the following functions:

- **Edit.** Edits the schedule for the relationship.
- **Delete.** Deletes the SnapMirror relationship. This function does not delete the destination volume.
- **Initialize.** Performs the first initial baseline transfer of data to establish a new relationship.
- **Update.** Performs an on-demand update of the relationship, replicating any new data and Snapshot copies included since the last update to the destination.
- **Quiesce.** Prevents any further updates for a relationship.
- **Resume.** Resumes a relationship that was quiesced.
- **Restore.** Restore
- **Break.** Makes the destination volume read/write.
- **Resync.** Reestablishes a broken relationship in the same direction before the SnapMirror break occurred.

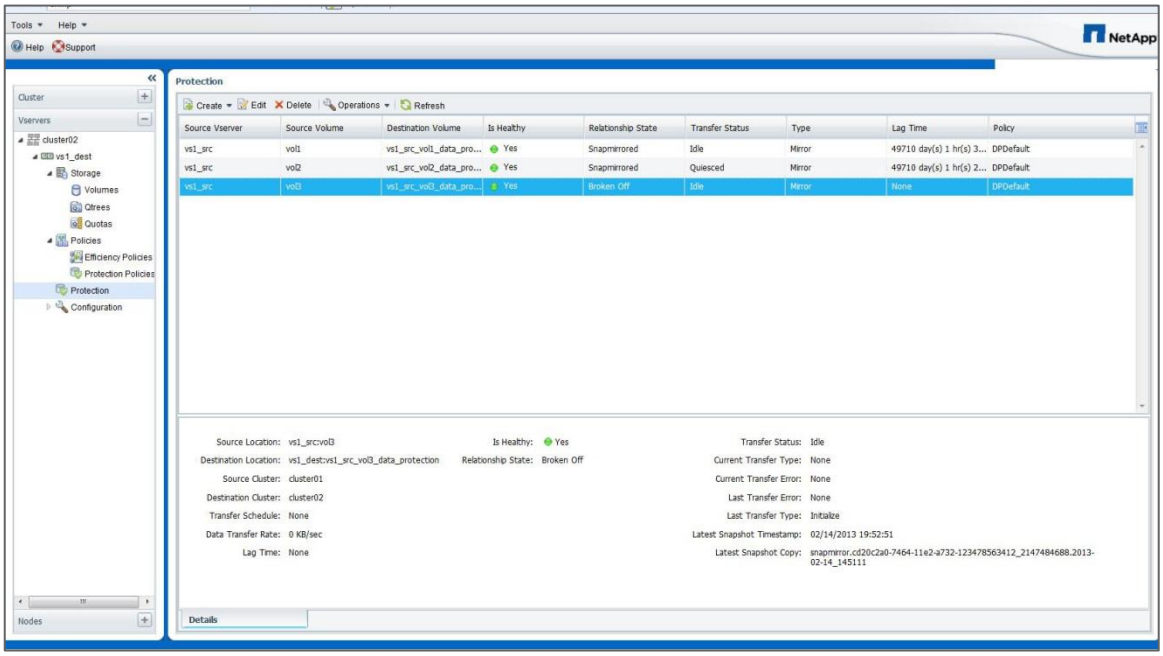
  Note: If a SnapMirror relationship is broken, deleted, and then recreated, perform a SnapMirror resync to resynchronize the volumes without having to rebaseline. This task requires that a common Snapshot copy exist on both volumes.

- **Reverse Resync.** Automates the necessary steps to reverse a SnapMirror relationship, recreating it and then resyncing it in the opposite direction. This can only be done if the existing relationship is in a broken state. Determine that clients are not using the original source volume, since the reverse/resync operation makes the original source volume read-only. The original source volume reverts to the most recent common Snapshot copy and resynchronizes with the destination. Any changes that are made to the original source volume since the last successful SnapMirror update are lost. Any changes that were made to, or new data written into, the current destination volume is sent back to the original source volume. When selecting this option, System Manager displays a confirmation screen explaining the operation that is being performed.
- **Abort.** Cancels a current transfer in progress. If a SnapMirror update is issued for an aborted relationship, the relationship continues with the last transfer from the last restart checkpoint that was created before the abort occurred.

Relationship Health in System Manager

SnapMirror relationships are primarily managed from the destination system; however, up-to-date information about a SnapMirror relationship can be reviewed using System Manager. When the SnapMirror status screen initially loads, authoritative information only displays for destination volume relationships that are on the selected Storage Virtual Machine, as shown in Figure 39.

Figure 39) SnapMirror status screen.



Relationships in which only the source volume is on the selected Storage Virtual Machine are initially shown with Health unknown. Click on that relationship and refresh the status from the destination to show the correct status in the Health column. Clicking on that relationship causes System Manager to contact the destination and collect authoritative information about that relationship and display the status. Clicking on the relationship also causes the detailed information pane at the bottom of the window to be updated with more information about that relationship, such as the last replication time stamp.

7 SnapMirror Load-Sharing Mirror Relationships

SnapMirror LS mirrors increase performance and availability for NAS clients by distributing a Storage Virtual Machine namespace root volume to other nodes in the same cluster and distributing data volumes to other nodes in the cluster to improve performance for large read-only workloads.

Note: SnapMirror LS mirrors are capable of supporting NAS only (CIFS/NFSv3). LS mirrors do not support NFSv4 clients or SAN client protocol connections (FC, FCoE, or iSCSI).

7.1 Administering Load-Sharing Mirrors

LS mirror relationships can only be managed by the Data ONTAP. Currently, LS mirror relationships cannot be managed using System Manager.

One way in which LS mirror relationships differ from DP relationships is that additional commands are provided to manage the LS mirror's `snapmirror initialize-ls-set`, `update-ls-set`, and `promote` commands. A group of LS mirror destination volumes that replicate from the same source volume is referred to as an LS mirror set.

When an LS mirror set is created, each destination volume must be created in the appropriate aggregate, creating the destination volumes with a type of DP. In this example, two volumes named `vs1_ls_a` and `vs1_ls_b` are created as LS mirror destination volumes for the Storage Virtual Machine root volume named `vs1`.

```
cluster01::> vol create -vserver vs1 -volume vs1_ls_a -aggregate aggr1 -size 20MB -
type DP

cluster01::> vol create -vserver vs1 -volume vs1_ls_b -aggregate aggr1 -size 20MB -
type DP
```

After all LS mirror destination volumes are created, each SnapMirror relationship can be created with a type of LS. In this example, an LS SnapMirror relationship is created for each of the destination volumes, `vs1_ls_a` and `vs1_ls_b`, with an hourly update schedule.

```
cluster01::> snapmirror create -source-path vs1:vs1 -destination-path vs1:vs1_ls_a -
type LS

cluster01::> snapmirror create -source-path vs1:vs1 -destination-path vs1:vs1_ls_b -
type LS -schedule hourly
```

LS mirror relationships can be updated manually or by setting the desired schedule in the `-schedule` option. For LS mirror relationships, this is done by setting the desired schedule on any one of the destinations in the LS mirror set. Data ONTAP automatically applies that schedule to all destinations in that LS mirror set. A later change to the update schedule for any of the destination volumes in the LS mirror set applies the new schedule to all volumes in that LS mirror set. Therefore, in the previous example, the `-schedule` option was used only in the creation of the last relationship, which applied the schedule to both relationships.

All destination volumes can then be initialized for a particular LS mirror set in one operation using the `snapmirror initialize-ls-set` command, as shown in the following example. When using this command, specify the source path to identify the LS mirror set instead of a destination path, because in an LS mirror set the source path is common to all relationships that are being initialized.

```
cluster01::> snapmirror initialize-ls-set -source-path cluster01://vs1/vs1
```

```
cluster01::> snapmirror show
                                                                    Progress
Source            Destination  Mirror  Relationship  Total            Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
cluster01://vs1/vs1
            LS   cluster01://vs1/vs1_ls_a
                              Snapmirrored
                                      Transferring   -         false   -
                 cluster01://vs1/vs1_ls_b
                              Snapmirrored
                                      Transferring   -         false   -
```

LS mirror relationships can be updated on demand using the `snapmirror update-ls-set` command, as shown in the following example. When using this command, specify the source path to identify the LS mirror set instead of a destination path, because in an LS mirror set the source path is common to all relationships that are being updated. Data ONTAP updates all destination volumes for the LS set in one operation.

```
cluster01::> snapmirror update-ls-set -source-path vs1:vs1
```

7.2 Accessing Load-Sharing Mirror Volumes

By default, all client requests for access to a volume in an LS mirror set are granted read-only access. Read-write access is granted by accessing a special administrative mount point, which is the path that

servers requiring read-write access into the LS mirror set must mount. All other clients will have read-only access. After changes are made to the source volume, the changes must be replicated to the rest of the volumes in the LS mirror set using the `snapmirror update-ls-set` command, or with a scheduled update.

Volumes can be mounted inside other volumes, also referred to as a nested volume. When a new volume is mounted inside a volume that is configured in an LS mirror set, clients cannot see the new mount point until after the LS mirror set has been updated. This can be performed on demand using the `snapmirror update-ls-set` command or when the next scheduled update of the LS mirror is set to occur.

Access to the volume in read-write mode is granted to different CIFS and NFS clients.

To allow a CIFS client to connect to the source volume with read-write access, create a CIFS share for the admin mount point by adding `/.admin` to the volume path. In the following example, a CIFS share called `report_data_rw` is created that allows read-write access to a volume called `report_data`, which is part of an LS mirror set.

Use the following path to access the read-write admin share of an LS mirror set using CIFS.

```
Cluster01::> vserver cifs share create -vserver vs1 -share-name report_data_rw -path
/.admin/report_data
```

Any CIFS client requiring read-write access must connect to the following path.

```
\\vs1\report_data_rw
```

To connect to the source volume of an LS mirror set with read-write access, from an NFS client, mount the NFS export path and add `/.admin` to the volume path, as shown in the following example.

```
nfs_client# mount -t nfs vs1:/.admin/report_data /client_rw_mountpoint
```

Any process or application running on the `nfs_client` system must use the path `/client_rw_mountpoint` for read-write access.

7.3 Load-Sharing Mirrors for Storage Virtual Machine Namespace Root Volumes

A namespace root volume is very small, containing only directories that are used as mount points, the paths where data volumes are junctioned (mounted) into the namespace. However, they are extremely important for NAS clients, which are not able to access data if the Storage Virtual Machine root volume is unavailable.

Note: SAN client connections (FC, FCoE, or iSCSI) do not depend on the Storage Virtual Machine root volume.

New volumes can be mounted into a namespace root volume that is configured in an LS mirror set. After mounting the volume, clients cannot see the new mount point until after the LS mirror set has been updated. This can be performed on demand using the `snapmirror update-ls-set` command, or when the next scheduled update of the LS mirror is set to occur.

As previously mentioned, LS mirror volumes are read-only. When the LS mirror volume is a namespace root volume, that volume is read-only; however, data volumes mounted in the namespace are read-write or read-only, depending on the individual volume characteristics and permissions set on files and folders within those volumes.

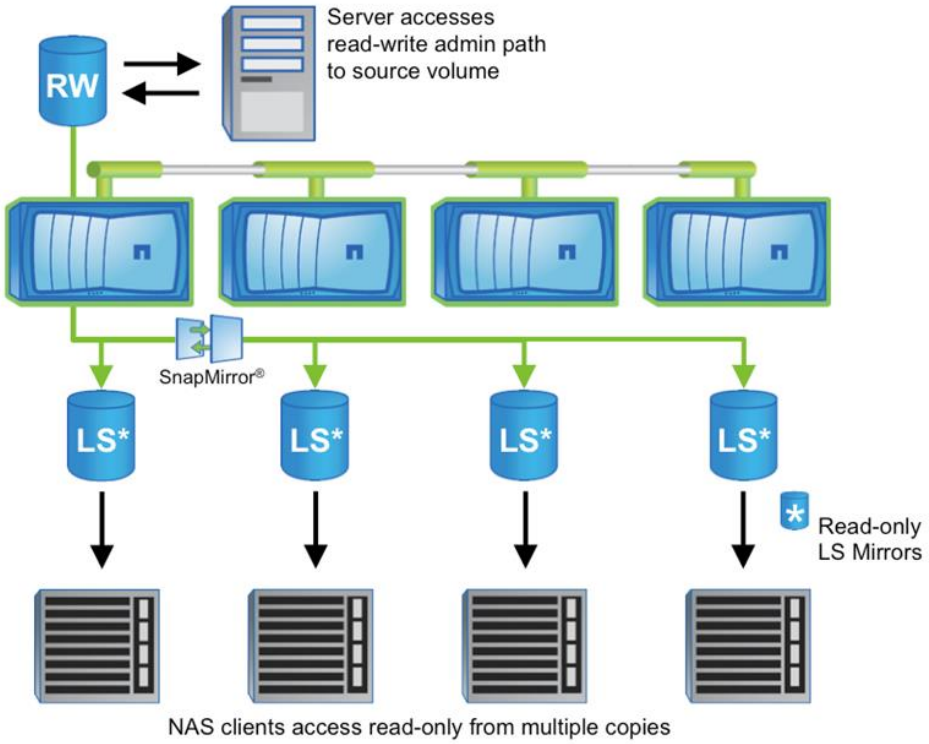
Best Practice
Create an LS mirror of a NAS Storage Virtual Machine namespace root volume on every node in the cluster so that the root of the namespace is available, regardless of node outages or node failovers. When a client requests access to a volume configured with an LS mirror set, Data ONTAP directs all client connections only to the LS mirror destination volumes; therefore, a destination volume on the same node where the source volume resides should be created, allowing the namespace to provide a direct data access path to data volumes on that node.

7.4 Load-Sharing Mirrors for Read-Only Workloads

LS mirrors can also be used to distribute data volumes to other nodes in the cluster to improve performance for read-only workloads. LS mirrors of data volumes are used in cases in which one or a few clients have read-write access to the dataset and many clients have read-only access. For example, a few servers that generate a large amount of test output data into the source volume and many servers that have read-only access to the test data can process it to output reports with increased performance because the read-only workload has been distributed across the cluster.

In the configuration shown in Figure 40, all volumes appear as one volume and are presented as read-only to all clients. In the same way that LS mirrors of a namespace root volume can distribute connections across a cluster, the read-only LS mirror volumes can be made available on every node, so that all clients can connect to the read-only volume by a direct data access path.

Figure 40) LS mirrors for read-only workloads.



7.5 Failover of Load-Sharing Mirror Relationships

Failover of a SnapMirror LS mirror relationship is performed differently than failover of a DP relationship. The `snapmirror break` command is not allowed for LS mirror destination volumes in Data ONTAP 8.1. Failover of LS mirror relationships uses the `snapmirror promote` command. This command promotes the failover target volume to be the source volume for the entire LS mirror set and it also deletes the original source volume. In this example, the Storage Virtual Machine root volume `vs1` is mirrored to volume `vs1_a` and `vs1_b`, and volume `vs1_b` is promoted to replace the source volume in the LS mirror set.

```
cluster01::> snapmirror show -type LS
Source             Destination  Mirror        Relationship   Total
Path          Type Path         State         Status         Progress   Healthy
------------ ---- ------------ ------------- -------------- ---------- -------
cluster01://vs1/vs1
             LS   cluster01://vs1/vs1_a
                               Snapmirrored  Idle           -          true
                  cluster01://vs1/vs1_b
                               Snapmirrored  Idle           -          true

cluster01::> snapmirror promote -destination-path cluster01://vs1/vs1_b

Warning: Promote will delete the read-write volume cluster01://vs1/vs1 and replace it
with cluster01://vs1/vs1_b.
Do you want to continue? {y|n}: y
[Job 1437] Job succeeded: SnapMirror: done
```

The destination volume that is the target of the `promote` command is now the source volume for the LS mirror set and all NFS file handles and CIFS connections are not interrupted. Remember that LS mirror volumes are read-only and that all client access requests are directed only to destination volumes, so there are no read-only connections to the source volume that was removed by the promote operation except for connections that might access the read-write admin share. NFS file handles and CIFS connections to the read-write admin share are nondisruptively transferred to the new source volume. Read-only connections to the promoted volume are nondisruptively transferred to other destination volumes in the LS mirror set.

The promote operation also deletes the original source volume, and the specific SnapMirror relationship for the promoted destination volume is removed.

```
cluster01::> snapmirror show -type LS
Source             Destination  Mirror        Relationship   Total
Path          Type Path         State         Status         Progress   Healthy
------------ ---- ------------ ------------- -------------- ---------- -------
cluster01://vs1/vs1_b
             LS   cluster01://vs1/vs1_a
                               Snapmirrored  Idle           -          true
```

Because the promoted volume would have had a different volume name than the original source volume, the new source volume can be renamed to retain that name for the source of the LS mirror set, as shown in the following example.

```
cluster01::> volume rename -vserver vs1 -volume vs1_b -newname vs1
[Job 1438] Job succeeded: Successful

cluster01::> snapmirror show -type LS
Source             Destination  Mirror        Relationship   Total
Path          Type Path         State         Status         Progress   Healthy
------------ ---- ------------ ------------- -------------- ---------- -------
cluster01://vs1/vs1
             LS   cluster01://vs1/vs1_a
```

```
                                    Snapmirrored  Idle            -           true
```

Best Practice

The promote operation deletes the original source volume; therefore, another LS mirror destination volume might need to be created on the node where the current source volume was located.

8 SnapMirror and Data ONTAP Feature Interaction

8.1 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 data between the new SnapMirror Snapshot copy and the previous one (including all Snapshot copies on the volume between those and all data in those Snapshot copies) is replicated to the destination volume. Once 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.

Data ONTAP maintains locks on Snapshot copies created by SnapMirror to prevent them from being deleted; these Snapshot copies are required to perform scheduled updates. If the SnapMirror-created Snapshot copies must be deleted, the volumes can still be resynchronized without having to perform a full baseline as long as other common Snapshot copies between the two volumes still exist on the volumes. In this example, a SnapMirror resync is performed on a volume where all SnapMirror-created Snapshot copies were deleted.

Note: The system specifies the name of an hourly Snapshot copy used for 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.
```

8.2 SnapMirror and Qtrees

Qtrees are special directories that allow the application of file system quotas for NAS. Clustered Data 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. All SnapMirror replication is at the volume level only.

8.3 SnapMirror and FlexClone

A NetApp FlexClone volume is a writable point-in-time clone of a FlexVol volume. A FlexClone volume shares data blocks with the parent volume, storing only new data or changes made to the clone. A FlexClone volume can also be split from its parent to create a new standalone volume.

A SnapMirror relationship can be created using a FlexClone volume as the source; however, a SnapMirror destination volume cannot be a FlexClone volume.

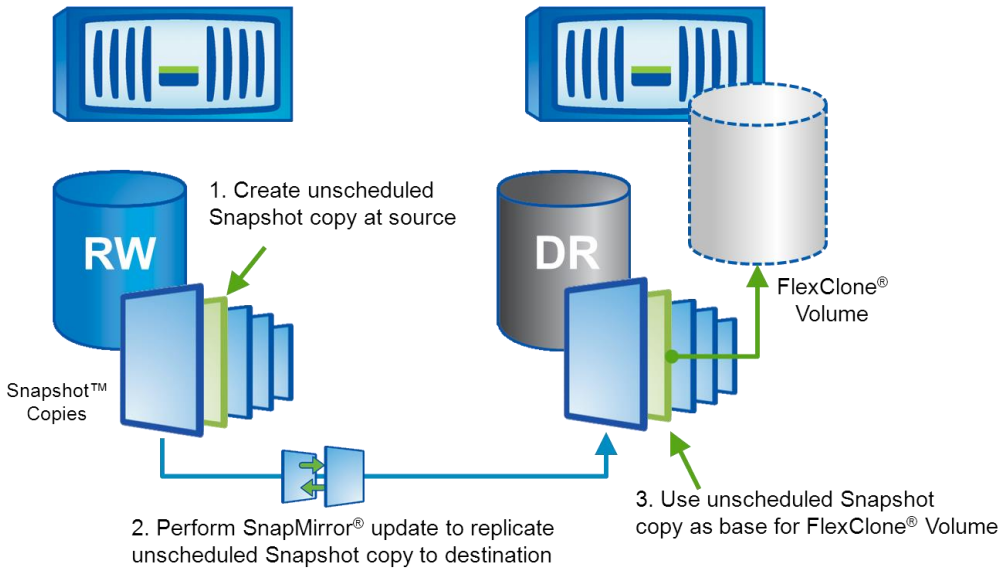
FlexClone technology also makes it possible to create a writable volume from a read-only SnapMirror destination without interrupting the SnapMirror replication process or the production operations. Figure 41 illustrates the creation of a FlexClone volume at the SnapMirror destination.

Best Practice

SnapMirror replicates Snapshot copy history from source to destination volumes. 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 destinations, create the base Snapshot copy required by the FlexClone volume on the source system and 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 41. 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.

Figure 41) Creating a FlexClone volume at the SnapMirror destination.



8.4 SnapMirror and Infinite Volume

SnapMirror works with Infinite Volume just like a FlexVol volume.

There are a few key differences:

- For namespace (NS) constituent volume, mirroring is restricted to intracluster only. Starting in clustered Data ONTAP 8.2, NS mirror is automatically created when you create the Infinite Volume. If you use SnapDiff, it will automatically create one NS mirror per node; and if you don't use SnapDiff then you will have only one NS mirror on the Infinite Volume.
- Only intercluster SnapMirror is supported for mirroring the entire Infinite Volume.

The process of creating an Infinite Volume is different compared to a FlexVol volume. But, apart from that the SnapMirror relationship setup is the same as with a FlexVol volume. This section walks through the SnapMirror lifecycle operations with Infinite Volume.

Creating an Infinite Volume

Create a Storage Virtual Machine and a volume on the source. Make sure the Storage Virtual Machine has `-is-repository` set to `true` to specify that the Storage Virtual Machine will be holding an Infinite Volume.

```
cluster01::> vserver create -vserver VS1 -rootvolume rootvol -aggregate aggrroot -ns-
switch file -nm-switch file -rootvolume-security-style mixed -is-repository true

cluster01::> volume create -vserver VS1 -volume IV1 -state online -type RW -policy
repos_namespace_export_policy -security-style unified -space-guarantee none
```

Also do the same on the destination. These Storage Virtual Machines will only be able to hold one Infinite Volume.

```
cluster02::> vserver create -vserver DVS1 -rootvolume rootvol -aggregate aggrroot -ns-
switch file -nm-switch file -rootvolume-security-style mixed -is-repository true

cluster02::> volume create -vserver DVS1 -volume DIV1 -state online -type DP -policy
repos_namespace_export_policy -security-style unified -space-guarantee none
```

Create a Storage Virtual Machine peer relationship between the source and destination Storage Virtual Machine.

SnapMirror Create

Create a SnapMirror relationship from destination cluster.

```
cluster02::> snapmirror show
This table is currently empty.

cluster02::> snapmirror create -source-path VS1:IV1 -destination-path DVS1:DIV1
```

```
cluster02::> snapmirror show
                                                                    Progress
Source            Destination  Mirror  Relationship  Total            Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
VS1:IV1       DP   DVS1:DIV1
                              Uninitialized
                                     Idle           -         true    -
```

SnapMirror Initialize

Initialize the relationship to start the baseline data transfer from the source to the destination.

```
cluster02::> snapmirror initialize -destination-path DVS1:DIV1

cluster02::>  snapmirror show
Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
cluster02::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
VS1:IV1       DP   DVS1:DIV1
                               Uninitialized
                                       Transferring    -         true    -

cluster02::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
VS1:IV1       DP   DVS1:DIV1
                               Snapmirrored
                                       Idle           -         true    -
```

SnapMirror Update

Perform an update to make the destination volume an up-to-date mirror of the source volume. This will make the source create a new Snapshot copy, which will be transferred over to the destination.

```
cluster02::> snapmirror update -destination-path DVS1:DIV1
```

SnapMirror Break

This command breaks the SnapMirror relationship between the source and the destination Infinite Volume and will cause the destination Infinite Volume to be a read-write volume.

```
cluster02::> snapmirror break -destination-path DVS1:DIV1

cluster02::> snapmirror show
                                                                       Progress
Source            Destination  Mirror  Relationship  Total             Last
Path        Type  Path         State   Status        Progress  Healthy Updated
----------- ---- ------------ ------- -------------- --------- ------- --------
VS1:IV1 DP   DVS1_:DIV1
                               Broken-off
                                       Idle           -         -       -
```

SnapMirror Resync

This command reestablishes mirroring where the destination mirror is broken. The destination volume is made a DP mirror, and the mirror can be manually updated or scheduled for updates.

```
cluster02::> snapmirror resync -destination-path DVS1:DIV1

Warning: All data newer than Snapshot copy
         snapmirror.c1463ecf-5f29-11e2-8552-123478563412_11_13.2013-02-28_1404
```

```
         on volume TestVS1_dest:DIV1 will be deleted.
Do you want to continue? {y|n}: y
```

SnapMirror Delete

Deleting a SnapMirror relationship removes the SnapMirror relationship between the source Infinite Volume and the destination Infinite Volume. It does not destroy the Infinite Volume.

```
cluster02::> snapmirror delete -destination-path DVS1:DIV1
```

SnapMirror Release

This command removes the DP relationship information from the source Storage Virtual Machine. It will unlock and clean up the Snapshot copies pertaining to the relationship.

```
cluster01::> snapmirror release -destination-path DVS1:DIV1
```

For more information on Infinite Volume in clustered Data ONTAP 8.2, refer to TR-4178.

8.5 SnapMirror and NetApp 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 uncompress the source volume to read data for a transfer; data is replicated in a compressed state to the destination volume.

It is not possible to have different configurations of storage efficiency enabled between the source and destination volumes. For example, it is not possible to compress or deduplicate the SnapMirror destination volume alone without enabling compression or deduplication on the SnapMirror source volume.

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

8.6 SnapMirror and Volume Move

The volume move capability allows volumes to be moved nondisruptively between nodes in the cluster. DP mirror source or destination volumes can be moved 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 that the volume is moved to must have an intercluster LIF and be connected to the intercluster network in order to perform future 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 the clustered Data 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.

Note: In clustered Data ONTAP 8.1, for SnapMirror destination volumes, a SnapMirror transfer and a volume move transfer are mutually exclusive. A SnapMirror destination volume move cannot start while a SnapMirror transfer to that volume is in progress. A SnapMirror update transfer cannot be performed if the SnapMirror destination volume is currently in the process of being migrated with volume move. But, starting in clustered Data ONTAP 8.2, for SnapMirror destination volumes, a

SnapMirror transfer and a volume move transfer can run simultaneously, except during volume move cutover, when they will be mutually exclusive (brief duration of few seconds).

For more information on volume move, refer to TR-3975.

8.7 SnapMirror for Disk Shelf Failure Protection

If you 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.
- It will 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 would make 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 it has to be a different aggregate, but there's still the risk that an aggregate might have a disk in any shelf. Even if you try to set it up otherwise (keeping aggregates on their own shelves), that can change because drives fail and spares get used. This would avoid having a single point of failure and would provide protection against disk shelf failure. The caveat here is that it will not fail over automatically. You will have to manually break the SnapMirror relationship, unmount the clients, remount the clients on the destination volumes, and change the NFS export policies.

8.8 SnapMirror and Volume Autosize

The destination volume must be the same size 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; however, the destination volume can be slightly larger. The `-filesys-size-fixed` option makes sure that the file system size of a SnapMirror volume remains the same to allow a SnapMirror relationship to be reversed, even if the destination volume size is larger than the source.

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. Clustered Data ONTAP 8.1 does not automatically increase the size of the destination volume. Use the CLI or System Manager to increase the destination volume; the next SnapMirror update automatically replicates the value of the file system size to the destination volume to match that of the source.

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.

Starting in Data ONTAP 8.2, when autosize increases the size of the source volume of a SnapMirror relationship, the destination volume also automatically increases in size. This is applicable only to FlexVols, and not Infinite Volumes.

8.9 SnapMirror and Network Data management Protocol

Network Data Management Protocol (NDMP) backups can be performed from SnapMirror source or destination volumes. There are advantages to performing NDMP backups from SnapMirror destination volumes rather than from source volumes, they include:

- SnapMirror transfers can happen quickly and with less impact on the source system than that of NDMP backups. Use NetApp 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 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.

8.10 SnapMirror and Data ONTAP Version Dependencies

Replication for DP or DR is not possible between systems operating in 7-Mode and clustered Data ONTAP.

Several new replication capabilities have been implemented in SnapMirror in clustered Data ONTAP 8.1 such as block-level replication, support for NetApp Storage Efficiency, and the ability to replicate between clusters and break, reverse, and resync relationships. The Data ONTAP 8.1 implementation of SnapMirror is not compatible with the Data ONTAP 8.0 implementation. Replication between systems running clustered Data ONTAP 8.0 and 8.1 is not possible. For information about upgrading systems operating in clustered Data ONTAP 8.0 to 8.1, refer to the NetApp Data ONTAP 8.1 Cluster-Mode Upgrade and Revert/Downgrade Guide on the NetApp Support (formerly NOW) site.

Clustered Data ONTAP 8.2 introduces SnapVault to clustered Data ONTAP, supports SnapMirror cascading, SnapMirror to tape for seeding only, and is multi-tenancy ready with the ability for Storage Virtual Machine administrators to manage replication. Additionally, clustered Data ONTAP 8.2 improves scalability (number of concurrent transfers) and increases data transfer speeds.

The clustered Data ONTAP 8.2 implementation of SnapMirror is compatible with the clustered Data ONTAP 8.1 implementation. Replication between systems running clustered Data ONTAP 8.1 and 8.2 is possible. On an upgrade from clustered Data ONTAP 8.1 to 8.2, existing SnapMirror relationships will continue to remain cluster scope. The SnapMirror relationships will not benefit from the scalability improvements unless Storage Virtual Machine peering is established and the SnapMirror relationships are deleted and recreated once the source and destination nodes are both upgraded to clustered Data ONTAP 8.2.

Starting clustered Data ONTAP 8.2.1, on an upgrade from 8.1 to 8.2.1, the above procedure is automated. That is, the SnapMirror relationships are auto-converted to benefit from the scalability improvements once the destination node is upgraded to 8.2.1, source node is upgraded to 8.2 or above, and source and destination volumes are in the same Storage Virtual Machine or Storage Virtual Machine peering is established between the two Storage Virtual Machines that host the source and destination volumes. For information about upgrading systems operating in clustered Data ONTAP 8.1 to 8.2, refer to the NetApp Data ONTAP 8.2 Cluster-Mode Upgrade and Revert/Downgrade Guide on the NetApp Support (formerly NOW) site.

The following replication is supported between different versions of clustered Data ONTAP 8.1 and higher:

- Replication is allowed in either direction between minor versions. Clustered Data ONTAP 8.1.X is a minor version; therefore, replication is supported from 8.1.X to 8.1.Y or from 8.1.Y to 8.1.X.

- Replication is only allowed from older to newer major versions. Clustered Data ONTAP 8.1 is a major version; therefore, replication is allowed from 8.1 to a later major release (for example, 8.2). However, replication is not allowed from a later release (for example, 8.2) to an earlier major release, (for example, 8.1).

  Note: The clustered Data ONTAP 8.0 release family is excluded.

9 SnapMirror Sizing Recommendations

9.1 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 Data ONTAP. For information about the number of concurrent SnapMirror operations allowed per node, refer to the NetApp Data ONTAP Cluster-Mode Data Protection Management Guide on the NetApp Support (formerly NOW) site for the appropriate Data ONTAP release.

Best Practice
It is also important to understand which operations in clustered Data ONTAP constitute SnapMirror operations. Regularly scheduled SnapMirror updates of SnapMirror DP or LS relationships are SnapMirror operations. However, volume move and volume copy operations also use SnapMirror as the mechanism to move data from one aggregate to another. Therefore, when planning concurrent operations, it is a best practice to consider the frequency of volume move and volume copy operations in the environment.

Clustered Data ONTAP provides a greater level of scalability by allowing expansion of a NetApp 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
In order 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.

9.2 Recommended Replication Intervals

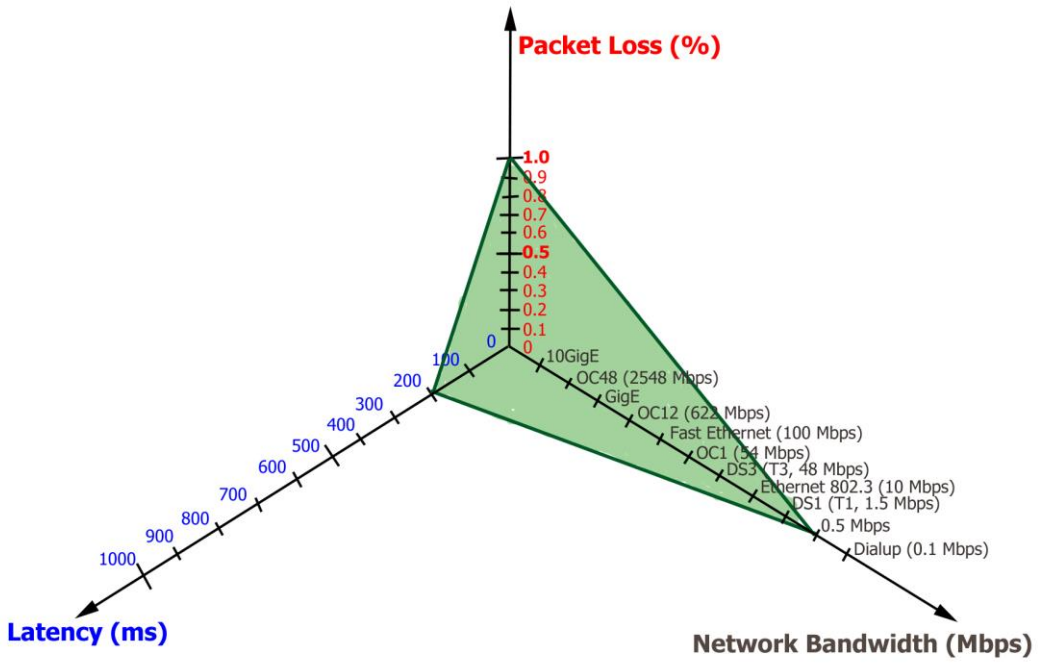
SnapMirror updates require establishing a communication session between the source and destination nodes, creating and deleting Snapshot copies, and determining which blocks of data to send to the destination. Therefore, while the Data ONTAP scheduler supports creating schedules that run every minute, NetApp does not recommend performing a SnapMirror update operation every minute. However, SnapMirror update operations in single digit are possible depending upon your environment. Please refer to SnapMirror Sizing Guide for Clustered Data ONTAP 8.2 for proper SnapMirror and system sizing guidelines.

9.3 Network Sizing Requirements

A network with the appropriate bandwidth available to transfer the system's data ingest rate is required to support the desired replication interval. There are limitations on the network characteristics that are supported for intercluster replication.

Network Sizing Requirements for Intercluster Replication

Figure 42) Factors to consider for optimum performance – packet loss (%), latency (ms) and network bandwidth (Mbps).



The intercluster network must be sized appropriately depending on the data change rate and 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 a minimum bandwidth of 0.5 megabits (Mbps), a maximum round-trip network latency of 200ms round-trip time (RTT), and a packet loss of 1% (volume covered by the green triangle in Figure 42).

Best Practice
It is important that all paths used for intercluster replication have equal performance characteristics. Configuring multipathing in such a way that a node has one intercluster LIF on a slow path and the same node has another intercluster LIF on a fast path adversely affects performance because data is multiplexed across the slow and fast paths simultaneously.

Network Sizing Requirements for Intracluster Replication

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

10 Troubleshooting Tips

10.1 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 in the document, when performing intercluster SnapMirror mirroring over multiple paths between the local and remote cluster, 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
```

10.2 Troubleshooting Storage Virtual Machine Peer Relationships

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

1. Storage Virtual Machine peer action failure for intercluster environment.
   a. Check if both the clusters are in same league.
   b. Check if peer cluster is reachable.

c. Check if both the clusters are running SN and Storage Virtual Machine peering capability is enabled.

d. Check if the peer Storage Virtual Machine name is not associated with another cluster from peer Storage Virtual Machine names in the Storage Virtual Machine peering table.

e. Check mgwd.log and console logs for error messages.

2. Storage Virtual Machine peer action failure for intracluster/intercluster environment.

a. Check if both the clusters are running SN and Storage Virtual Machine peering capability is enabled.

b. Check if local and peer Storage Virtual Machine names are not same.

c. Check mgwd.log and console logs for error messages.

3. Run the `vserver peer show` command to verify the Storage Virtual Machine peer relationship; this command displays all existing configured Storage Virtual Machine peer relationships.

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

4. Check for any notifications by `vserver peer show-all`.

```
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
```

10.3 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 OnCommand System 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, as shown in the second relationship.

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

In the fourth example, 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 if offline or if it cannot be reached.

10.4 Troubleshooting SnapMirror Relationships

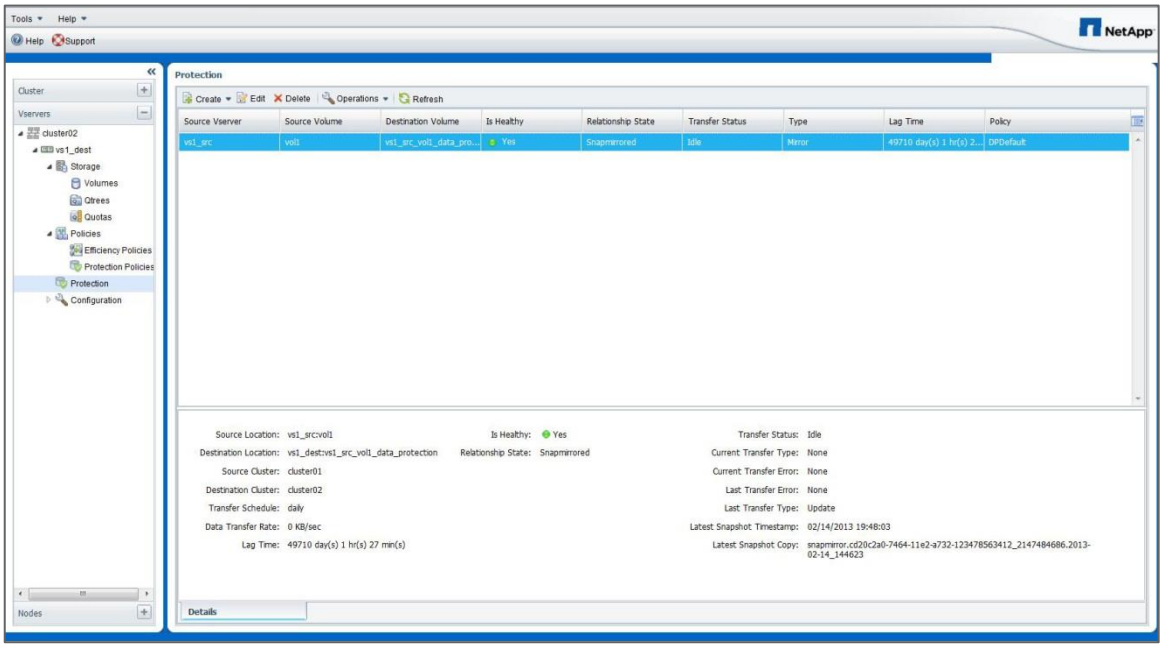
To determine when the last SnapMirror transfer for a specific relationship completed, refer to the Exported Snapshot Timestamp for instance information in clustered Data ONTAP 8.2.

```
cluster02::> snapmirror show -instance

                Source Path: vs1_src:vol1
          Destination Path: vs1_dest:vol1
          Relationship Type: DP
        SnapMirror Schedule: 8hour
                Tries Limit: -
          Throttle (KB/sec): unlimited
               Mirror State: Snapmirrored
        Relationship Status: Idle
          Transfer Snapshot: -
          Snapshot Progress: -
             Total Progress: -
        Snapshot Checkpoint: -
            Newest Snapshot: snapmirror.cd20c2a0-7464-11e2-a732-
123478563412_2147484690.2013-02-25_154353
  Newest Snapshot Timestamp: 02/25 20:45:36
          Exported Snapshot: snapmirror.cd20c2a0-7464-11e2-a732-
123478563412_2147484690.2013-02-25_154353
Exported Snapshot Timestamp: 02/25 20:45:36
                    Healthy: true
            Unhealthy Reason: -
   Constituent Relationship: false
    Destination Volume Node: cluster02-02
            Relationship ID: fdb1c700-7f5c-11e2-9caa-123478563412
              Transfer Type: -
             Transfer Error: -
           Current Throttle: -
  Current Transfer Priority: -
         Last Transfer Type: update
        Last Transfer Error: -
         Last Transfer Size: 206.1MB
     Last Transfer Duration: 0:0:3
         Last Transfer From: vs1_src:vol1
Last Transfer End Timestamp: 02/25 15:43:56
      Progress Last Updated: -
    Relationship Capability: 8.2 and above
                  Lag Time: 1193041:26:42
          SnapMirror Policy: DPDefault
```

Note: The last snapshot timestamp information also displays at the bottom of the System Manager interface, as shown in Figure 43.

Figure 43) Transfer timestamp information.



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 SnapMirror-related messages, as shown in the following example. Specify the `mgmt.snapmir*` message name to filter the output and find only SnapMirror-related messages.

```
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. OnCommand System Manager 3.0 allows viewing of the SnapMirror log files.

You can also use System Manager to view the SnapMirror log separately from the rest of the event logs: Cluster > Diagnostics > Logs > SnapMirror Log. From the Select node drop-down list, select the node that owns the volume you are interested in, as shown in Figure 44.

Figure 44) SnapMirror log.

SnapMirror logs cannot be viewed in System Manager 3.0. Under development.

11 Configuration and Failover for Disaster Recovery

Configuration and failover for DR is 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 have an effect on a DR plan. Depending on the type of DR solutions deployed, each organization's DR situation could 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. There are many variables to plan for depending on the configuration of each environment.

- Systems administrators have access to a workstation or server desktop session from which to administer the DR site and perform the failover.
- Systems 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 providing licensing services for all applications that require it

  Note: It is important that a server performing the necessary Active Directory FSMO roles is available at the DR site. For information regarding transferring roles to a surviving Active Directory server or seizing these roles from a failed server, refer to Microsoft KB 255504.http://support.microsoft.com/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 NetApp volumes are being 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 NetApp controller. This refers to 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 is necessary because, if the event causing the disaster is temporary or intermittent in nature, such as an extended power outage, when the primary site systems restart, and services might conflict with the recovered operations that are then running at the DR site.
- Plans have been made for providing users and applications access to the data and services at the DR site; for example, updating DNS such that home directory mount requests to the primary site Storage Virtual Machine are directed to the DR site Storage Virtual Machine instead.

11.2 Best Practices for DR Configurations

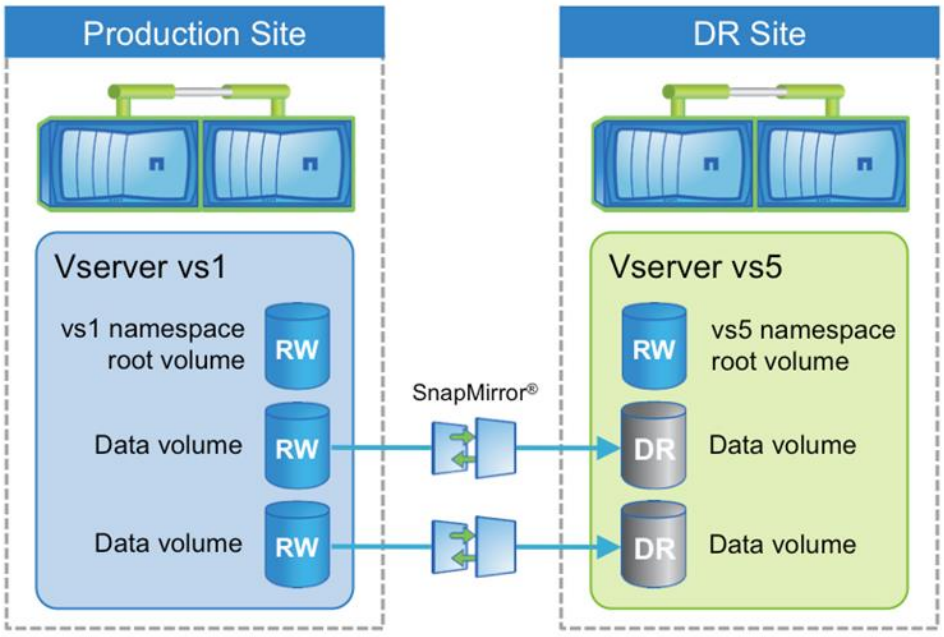
Best Practices
1. Volumes that belong to one Storage Virtual Machine at the source site should be replicated to one Storage Virtual Machine at the destination site. A Storage Virtual Machine is the root of a NAS namespace for NAS clients and a single storage target in SAN environments. If some NAS volumes are replicated from one Storage Virtual Machine into different Storage Virtual Machines at the destination, then all of those volumes cannot be recovered into the same namespace. The same is true of volumes containing LUNs; if the volumes are replicated into different Storage Virtual Machines at the destination, then all of the LUNs are not presented under the same SAN target.
2. The destination Storage Virtual Machine should be a member of the same Active Directory, LDAP, or NIS domain that the source Storage Virtual Machine is a member of. This is required so that access control lists (ACLs) stored within NAS files are not broken if a NAS volume is recovered into a Storage Virtual Machine that cannot authenticate those ACLs. The processes 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 like NetApp SnapDrive® for Windows® can be done using the same credentials.
3. Because the destination Storage Virtual Machine is a different Storage Virtual Machine than the source, and because NetApp recommends that it be a member of the same Active Director domain, the destination Storage Virtual Machine must be joined to the domain with a different Storage Virtual Machine 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 so that the CIFS shares are still accessible using the same UNC path name and NFS clients are also able to access the expected path.
4. Using destination volume names that are the same as the source volume names is not required but 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.
5. Construct the destination NAS namespace for a Storage Virtual Machine such that it is identical in paths and directory structure as the source Storage Virtual Machine.
6. 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.
7. Configure the destination Storage Virtual Machines ahead of time as described in the following section. This 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.
8. As new volumes are created at the source site, SnapMirror relationships must be created to replicate those volumes. Configuration settings pertaining to those volumes should be made in the DR site after the volumes are created and replicated so they can be ready in the event of a disaster.

11.3 Preparing the Destination for Failover

Many parts of a DR process for clustered Data ONTAP 8.1 onward can be prepared ahead of time, prior to a DR event. For example, mounting volumes into the namespace, creating CIFS shares, and assigning NFS export policies, as well as other things, can all be done ahead of time. SnapMirror cannot be used to replicate configuration information that could be independent in the destination Storage Virtual Machines, such as Storage Virtual Machine domain membership, CIFS configuration, NFS policies, Snapshot policy schedules, or NetApp Storage Efficiency policies.

Figure 45 illustrates volume layout for DR.

Figure 45) Volume layout for DR.



After volumes have been replicated, complete the following steps to prepare the destination system for failover, as show in Figure 45.

NAS and SAN Environments

1. Configure the destination Storage Virtual Machine membership into the appropriate Active Directory, LDAP, or NIS domain.
2. Determine that the destination Storage Virtual Machine is a member of the same domain as the source Storage Virtual Machine so that authentication is not broken for tools, such as NetApp SnapDrive for Windows, and so 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: NetApp 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 NetApp Storage Efficiency policies in the destination Storage Virtual Machine.

   Note: If NetApp Storage Efficiency policies are assigned to the volumes in the source Storage Virtual Machine, a policy must be created in the destination Storage Virtual Machine in order to schedule the dedupe process after failover at the DR site. NetApp Storage Efficiency policies must be applied to DP volumes after failover.

NAS Environments

1. Verify that all necessary volumes in the source Storage Virtual Machine are being replicated to the destination Storage Virtual Machine.

   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 security style and permissions on the destination Storage Virtual Machine root volume.

   The security style and permissions of the root of the destination Storage Virtual Machine namespace must be set correctly or the NAS namespace might be inaccessible after failover.

3. Mount the destination NAS volumes into the destination Storage Virtual Machine namespace.

   SnapMirror does not replicate the Storage Virtual Machine 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 that the source volume was mounted to in the source Storage Virtual Machine. This 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 of (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 Storage Virtual Machine using the same share names that were used at the source. Clients are able to 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 Storage Virtual Machine.
7. Assign the NFS export policies to the destination volumes. Clients are able to access the NFS exports; however, all data is read-only until the volume is failed over.

SAN Environments

1. If the destination Storage Virtual Machine use portsets, they can be configured as required before failover.
2. Configure igroups on the destination Storage Virtual Machine.

   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 Storage Virtual Machine.

   Since 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 done after failover.

SnapMirror ToolKit (for Clustered Data ONTAP 8.2)

The main goal of this tool (SnapMirror ToolKit) is to improve the user experience of setting up and running SnapMirror (and SnapVault) in Clustered Data ONTAP 8.2. The feedback from QA, the usability team and customers who participated in the Clustered Data ONTAP 8.2 Early Validation Program indicates that SnapMirror in Clustered Data ONTAP 8.2 is more complicated to set up and manage than 7-mode SnapMirror. We will improve the usability of SnapMirror in future releases of Clustered Data ONTAP, but these scripts provide immediate benefit. They are lightweight, portable and provide a simpler user experience than using the Clustered Data ONTP CLI. Furthermore, customers can build in-house automated tools using these scripts as a foundation.

You can download the SnapMirror ToolKit (SMTK) from the SE Utility Toolchest - http://support.netapp.com/NOW/download/tools/smtk.

11.4 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 clustered Data 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 Storage Virtual Machine 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 created and applied, clients then have read-write access to the NAS data.
3. Redirect clients to the recovered storage.

   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 enables CIFS shares to be accessible using the same UNC path name, and NFS clients can also access the expected path. Alternatively, the failed source storage system might be removed from Active Directory and the recovery storage system removed and readded 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. In clustered Data ONTAP 8.1, 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 Storage Virtual Machine 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.

11.5 Postfailover Volume Configuration

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

1. If using the Data ONTAP Snapshot copy schedule, assign a Snapshot copy policy to the recovered volumes. In SAN environments Snapshot copies are typically scheduled in the client.
2. If using NetApp Storage Efficiency technology, assign a Storage Efficiency policy to the recovered volumes.

12 SnapMirror Transition

There are existing 7-Mode customers who are using QSM and VSM (sync, semi-sync, and async). How would you transition those customers to clustered Data ONTAP? This will be covered in TR-4052: Clustered Data ONTAP Transition Guide.

13 References

The following references were used in this TR:

- Data ONTAP 8.1 Cluster-Mode Data Protection Guide on NetApp Support
- Data ONTAP 8.1 Cluster-Mode Upgrade and Revert/Downgrade Guide on NetApp Support
- Data ONTAP 8.2 Cluster-Mode Data Protection Guide on NetApp Support
- Data ONTAP 8.2 Cluster-Mode Upgrade and Revert/Downgrade Guide on NetApp Support
-
- TR-3975: DataMotion for Volumes Overview in Clustered Data ONTAP 8.2
- TR-4178: Infinite Volume Deployment and Implementation Guide
- TR-4183: SnapVault Best Practices Guide for Clustered Data ONTAP
- TR-4052: Clustered Data ONTAP Transition Guide

14 Version History

Version	Date	Document Version History
Version 2.2	November	Amit Prakash Sawant
Version 2.1	July 2013	Amit Prakash Sawant
Version 2.0	April 2013	Amit Prakash Sawant
Version 1.0	February 2012	Larry Touchette

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

www.netapp.com