



NetApp®

技术报告

适用于集群模式 Data ONTAP 的 SnapMirror 配置和最佳实践指南

NetApp 公司 Vincent Goveas
2015 年 4 月 | TR-4015

SnapMirror 配置和最佳实践

本文档介绍有关在 NetApp® 集群模式 Data ONTAP® 中配置复制的信息和最佳实践。

目录

1 简介	6
1.1 目的及目标受众	6
1.2 SnapMirror 的用途和优势	6
2 概述	10
2.1 SnapMirror 技术概述	10
2.2 集群模式 Data ONTAP 概述	11
3 在不同集群之间进行复制所需的网络配置	12
3.1 集群间网络连接	13
3.2 创建集群对等关系	14
3.3 集群对等要求	14
3.4 集群间多路径和网络冗余	15
3.5 集群间 SnapMirror 的网络连接	18
3.6 确定使用共享端口还是专用端口进行复制	19
3.7 共享要进行集群间复制的数据端口	20
3.8 使用专用集群间端口进行集群间复制	22
3.9 配置集群对等	26
3.10 集群间 SnapMirror 的带宽限制	27
3.11 集群间 SnapMirror 的防火墙要求	27
4 SVM 对等关系	28
4.1 SVM 对等要求	28
4.2 配置 SVM 对等关系	28
5 SnapMirror 数据保护关系	29
5.1 SnapMirror 数据保护关系	31
5.2 计划 SnapMirror 更新	33
5.3 将 SnapMirror 关系转换为 SnapVault 关系	34
6 使用 NetApp OnCommand System Manager 管理 SnapMirror 数据保护关系	36
6.1 在 System Manager 中创建 SnapMirror 关系	36
6.2 使用 System Manager 管理 SnapMirror 关系	47
7 SnapMirror 负载共享镜像关系	49
7.1 管理负载共享镜像	49
7.2 访问负载共享镜像卷	50
7.3 SVM 命名空间根卷的负载共享镜像	50
7.4 只读工作负载的负载共享镜像	51
7.5 负载共享镜像关系的故障转移	51

8 版本灵活复制	53
8.1 使用案例	53
8.2 默认策略	53
8.3 配置版本灵活复制	53
8.4 将默认 SnapMirror 转换为版本灵活复制	53
9 SnapMirror 和 Data ONTAP 的功能交互	56
9.1 SnapMirror 和 Snapshot 副本	56
9.2 SnapMirror 和 qtree	56
9.3 SnapMirror 和 FlexClone	56
9.4 SnapMirror 和无限卷	57
9.5 SnapMirror 和 NetApp 存储效率	59
9.6 SnapMirror 和卷移动	60
9.7 使用 SnapMirror 进行磁盘架故障保护	60
9.8 SnapMirror 和卷自动调整	60
9.9 SnapMirror 和网络数据管理协议	61
9.10 SnapMirror 和 Data ONTAP 的版本依赖关系	61
9.11 SnapMirror 和网络压缩	61
10 SnapMirror 规模估算建议	64
10.1 并发复制操作	64
10.2 建议的复制时间间隔	64
10.3 网络规模估算要求	64
11 故障排除技巧	65
11.1 集群对等关系的故障排除	65
11.2 SVM 对等关系故障排除	66
11.3 了解 SnapMirror 关系的状态	67
11.4 SnapMirror 关系故障排除	68
12 通过配置和故障转移实现灾难恢复	70
12.1 环境的故障转移的要求和假设前提	70
12.2 灾难恢复配置的最佳实践	71
12.3 准备故障转移的目标系统	71
12.4 执行故障转移	73
12.5 故障转移后的卷配置	74
13 SnapMirror 的过渡	74
参考资料	74
版本历史	74

表格目录

表 1) 双跳卷 SnapMirror 的 Snapshot 副本传播。	32
表 2) 版本灵活复制策略。	53

插图目录

图 1) NetApp 集群模式 Data ONTAP 复制概览。	6
图 2) 使用 SnapMirror 进行灾难恢复。	7
图 3) 使用 NetApp FlexClone 进行灾难恢复测试的容量要求。	8
图 4) 用于开发和灾难恢复测试的 FlexClone 卷。	8
图 5) 使用 SnapMirror 进行数据分发。	9
图 6) 使用 SnapMirror 进行远程磁带归档。	9
图 7) 统一架构的灵活性。	10
图 8) 节点、HA 对、集群和 SVM。	11
图 9) 集群互连网络以及数据网络和管理网络。	12
图 10) 集群间网络。	12
图 11) 端口类型。	13
图 12) 在集群间网络上执行 SnapMirror。	13
图 13) 多集群对等关系。	14
图 14) 集群间 LIF 的完全连接。	15
图 15) 主动-被动多路径。	15
图 16) LIF 故障转移期间的主动-被动多路径功能。	16
图 17) 主动-主动多路径。	16
图 18) LIF 故障转移期间的主动-主动多路径。	17
图 19) 使用一个集群间 LIF 的 TCP 连接。	18
图 20) 使用两个集群间 LIF 的 TCP 连接。	18
图 21) 用于 SnapMirror 的集群间网络。	30
图 22) 用于集群间 SnapMirror 的集群互连。	30
图 23) 使用 SnapMirror 进行级联卷复制。	32
图 24) 将 SnapMirror 关系转换为 SnapVault 关系。	34
图 25) 从目标集群创建 SnapMirror 关系: 选择镜像。	37
图 26) 从目标集群创建 SnapMirror 关系: 选择源集群。	38
图 27) 从目标集群创建 SnapMirror 关系: 创建集群对等关系。	39
图 28) 从目标集群创建 SnapMirror 关系: 选择源 SVM。	40
图 29) 从目标集群创建 SnapMirror 关系: 选择源卷。	40
图 30) 从目标集群创建 SnapMirror 关系: 选择目标卷。	41
图 31) 从目标集群创建 SnapMirror 关系: 选择或创建 SnapMirror 策略和计划。	42
图 32) 从目标集群创建 SnapMirror 关系: 创建新的 SnapMirror 策略。	43
图 33) 从目标集群创建 SnapMirror 关系: 创建新的 SnapMirror 计划。	44
图 34) 从目标集群创建 SnapMirror 关系: 启动基线传输。	45

图 35) 从目标集群创建 SnapMirror 关系: SnapMirror 关系配置与状态摘要。	46
图 36) SnapMirror 基线传输的详细信息。	46
图 37) SnapMirror 关系列表。	47
图 38) Systems Manager 上下文菜单。	47
图 39) SnapMirror 状态屏幕。	48
图 40) 只读工作负载的 LS 镜像。	51
图 41) 在 SnapMirror 目标上创建 FlexClone 卷。	57
图 42) SnapMirror 网络压缩功能图。	62
图 43) 实现最佳性能需要考虑的因素: 丢包率 (%)、延迟 (毫秒) 和网络带宽 (Mbps)。	65
图 44) 传输的时间戳信息。	69
图 45) SnapMirror 日志。	70
图 46) 灾难恢复的卷布局。	72

1 简介

在硬件、软件甚至是站点出现故障的情况下，有几种方法可提高数据可用性。通过备份可从归档介质（磁带或磁盘）中恢复丢失的数据。冗余硬件技术也有助于减少由硬件问题或故障导致的损害。镜像可作为提高数据可用性和最大限度地减少停机时间的第三种机制。NetApp SnapMirror® 技术为通过局域网 (LAN) 和广域网 (WAN) 执行数据镜像或复制提供了一款快速而灵活的企业解决方案。SnapMirror 是企业数据保护 (DP) 策略的重要组成部分。

1.1 目的及目标受众

本文档面向负责管理、安装或支持集群模式 Data ONTAP 操作系统的人员，以及计划配置和使用 SnapMirror 技术进行数据复制的人员。

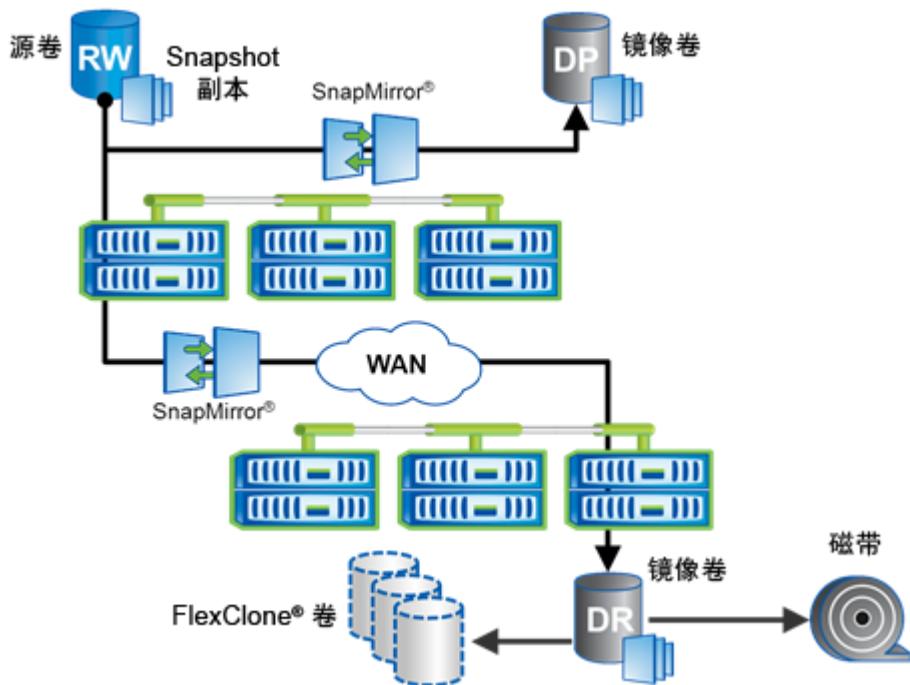
本文档假定读者已经了解以下流程和技术：

- 具备使用集群模式 Data ONTAP 操作系统操作流程进行存储系统管理的实践知识
- 具备使用 NetApp 的各种功能（如 NetApp Snapshot® 副本、NetApp FlexVol® 卷、NetApp FlexClone® 卷和 NetApp 无限卷）进行存储系统管理的实践知识
- 具备灾难恢复 (DR) 和数据复制解决方案方面的一般常识
- 熟悉 NetApp 支持站点上的《[Data ONTAP 8.3 集群模式数据保护指南](#)》

1.2 SnapMirror 的用途和优势

您可以在同一集群内复制数据，也可以将数据远程复制到另一个集群。NetApp Data ONTAP 可利用集成的数据复制技术来创建可用于灾难恢复 (DR) 的副本，以减轻主存储执行磁带备份过程的负担，将数据集分布到其他位置，并为测试和开发环境创建读/写克隆。有关集群模式 Data ONTAP 复制的概览，请参见图 1。

图 1) NetApp 集群模式 Data ONTAP 复制概览。



集成数据保护

NetApp Data ONTAP 操作系统内部集成了多种数据保护 (Data Protection, DP) 功能。NetApp SnapMirror 可与 NetApp Snapshot 技术相集成，提供一种不需要创建实际副本操作便可快速高效为数据创建磁盘副本或时间点副本的方法。

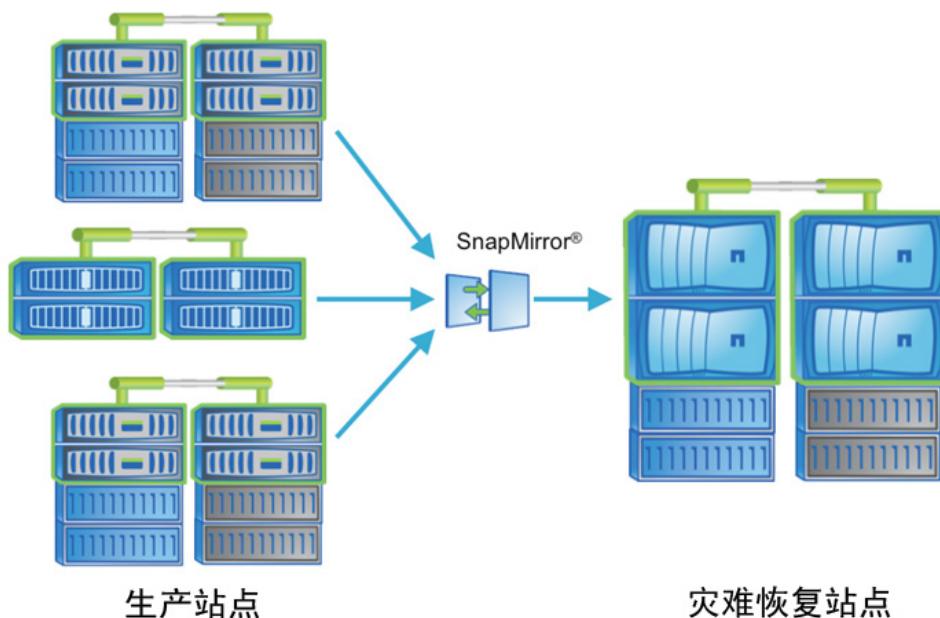
您可以使用 NetApp 集成数据保护功能在磁盘上为应用程序一致的 Snapshot 副本创建可快速访问的历史记录，从而消除传统备份时间间隔的概念。随后，NetApp SnapMirror 可将 Snapshot 副本的历史记录复制到可用于备份、灾难恢复或测试和开发的目标卷。

SnapMirror 复制很高效，因为它仅复制自上次更新以来更改过的或新添加的 4 KB 数据块。如果将 SnapMirror 与 NetApp 存储效率技术相结合，复制效率将进一步提升。如果主存储上使用了光纤连接存储 (Fabric-Attached Storage, FAS) 重复数据删除功能，则只会向灾难恢复站点复制非重复数据。如果源卷上启用了数据压缩功能，目标卷上也将保持数据压缩状态。由于数据已进行复制，因此不会解压缩。利用这些技术，可以节省电信费用，并大幅节省存储容量。

使用 SnapMirror 进行灾难恢复

SnapMirror 技术是灾难恢复计划不可或缺的组成部分。如果将关键数据复制到不同的物理位置，发生严重灾难时便不会导致数据长时间不可用。在修复因灾难造成的损坏之前，客户端可以通过网络访问复制的数据。恢复站点的应用程序服务器可以访问复制的数据来恢复业务关键型应用程序的运行，直到生产站点恢复为止。恢复操作可能包括从数据损坏、生产站点发生的自然灾害、意外删除等事件中进行恢复。

图 2) 使用 SnapMirror 进行灾难恢复。



如果因发生灾难而需要故障转移，而主存储未完全丢失，则可借助 SnapMirror 提供的一种高效方法来重新同步主站点和灾难恢复站点。主站点恢复联机后，SnapMirror 会重新同步两个站点，通过反转 SnapMirror 关系，仅将更改过的数据或新数据从灾难恢复站点传输回主站点。当主生产站点恢复正常应用程序操作后，会继续向灾难恢复设施进行 SnapMirror 传输，而无需再次执行完整的数据传输。

使用 NetApp FlexClone 技术进行灾难恢复测试和应用程序测试/开发

NetApp FlexClone 技术可用于快速创建 SnapMirror 目标 FlexVol 卷的读写副本，从而不再需要其他数据副本。例如，一个 10 GB 的 FlexClone 卷不需要再使用另一个 10 GB 的 FlexClone 卷来定义，只需要必要的元数据即可定义该 FlexClone 卷。FlexClone 卷仅会存储创建克隆之后写入或更改的数据。

图 3) 使用 NetApp FlexClone 进行灾难恢复测试的容量要求。

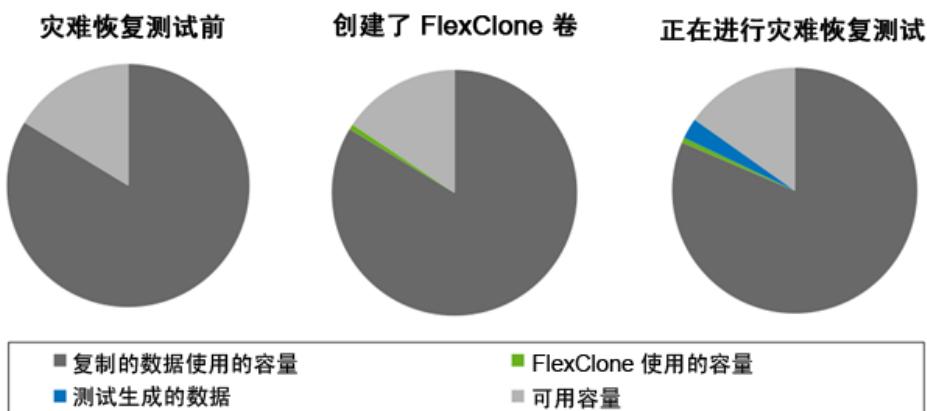
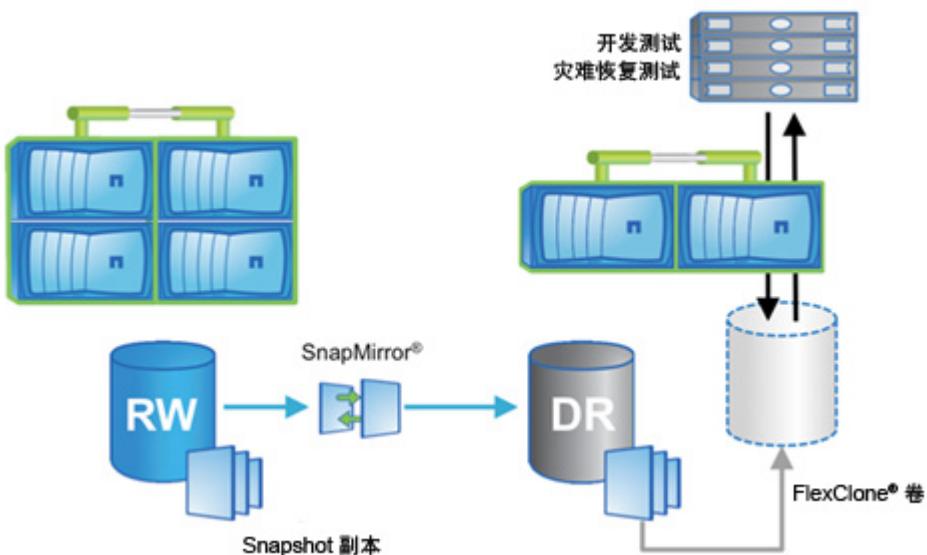


图 4 说明了 FlexClone 卷是如何在与其父 FlexVol 卷共享公共数据块的情况下表现为独立卷的。在后台向父 FlexVol 卷复制数据的同时，可以执行灾难恢复环境测试，甚至执行时间会更长。

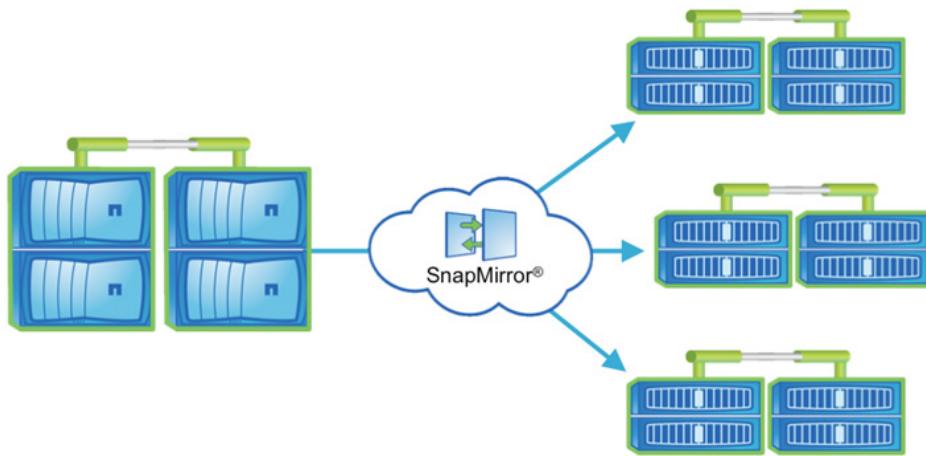
图 4) 用于开发和灾难恢复测试的 FlexClone 卷。



数据分发和远程数据访问

SnapMirror 技术可用于在企业范围内分发大量数据（如图 5 中所示），从而实现以只读方式访问远程位置的数据。通过远程数据访问，客户端便能以更快的速度访问远程位置的数据；而且由于是在预定的复制时间使用 WAN，因此会提高使用昂贵的网络和服务器资源的效率和可预测性。存储管理员可以在特定时间复制生产数据，从而最大限度地降低整个网络的占用率。

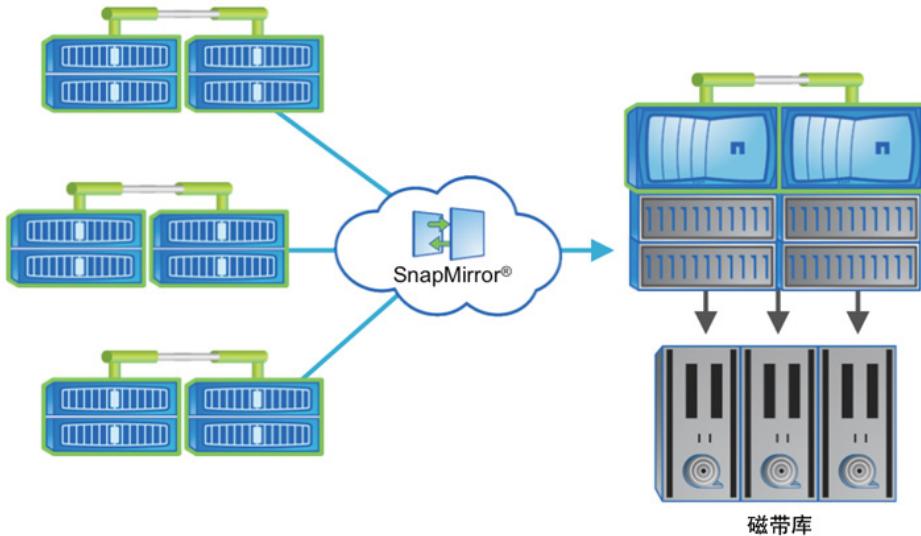
图 5) 使用 SnapMirror 进行数据分发。



备份负载分流及远程磁带归档

SnapMirror 技术还可用来整合备份，并减少生产服务器上的磁带备份开销。这样做有利于进行集中化备份操作，从而减少了远程位置的备份管理要求。由于使用 NetApp Snapshot 技术不再需要主存储系统上进行传统的备份，因此将磁带备份负载分流到 SnapMirror 目标（如图 6 中所示）会极大地降低生产存储系统上的备份操作开销。

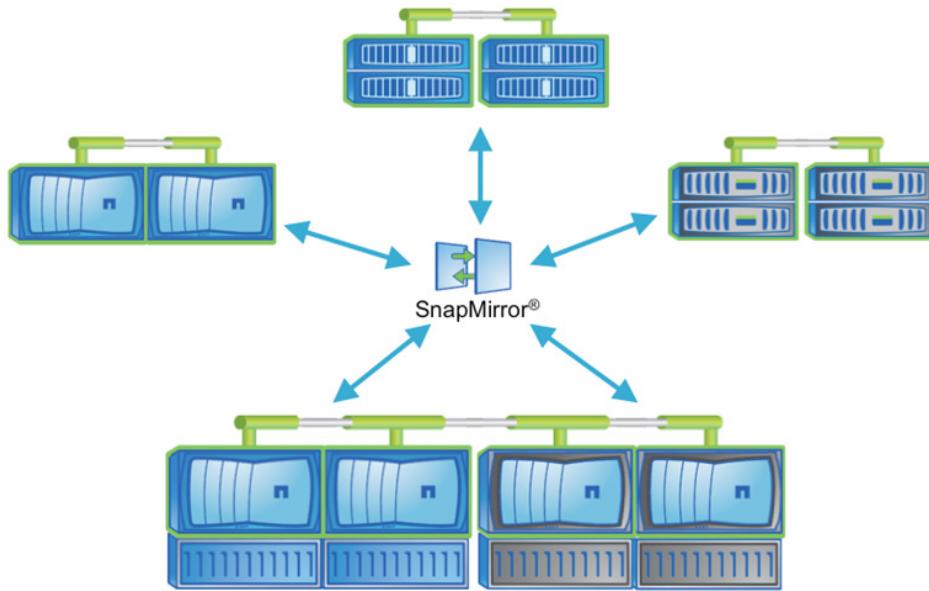
图 6) 使用 SnapMirror 进行远程磁带归档。



统一架构的灵活性

从集群模式 Data ONTAP 8.1 操作系统开始，您可以在 NetApp FAS 和（或）V 系列存储系统之间使用 SnapMirror 技术。在主站点和灾难恢复站点上，可以部署性能特性和成本各不相同的系统。例如，根据所需的功能，灾难恢复站点可能会部署一个型号较低的存储系统，使用的是 SATA 磁盘而不是光纤通道 (Fibre Channel, FC) 磁盘，或者使用的是 iSCSI 或以太网光纤通道 (Fibre Channel over Ethernet, FCoE) 协议而不是 FC。图 7 说明了统一架构的灵活性。

图 7) 统一架构的灵活性。



管理员还可以通过统一架构（从低端平台到高端平台）了解并使用一种管理和监控模式。

2 概述

2.1 SnapMirror 技术概述

集群模式 Data ONTAP 操作系统中的 SnapMirror 技术可根据所配置的复制更新间隔执行异步卷级复制。SnapMirror 会在复制过程中使用 NetApp Snapshot 技术。

集群模式 Data ONTAP 8.1 操作系统以及更高版本提供了以下复制功能：

- **数据保护镜像。**可通过复制在同一集群中创建备份副本（集群内复制）或在不同集群中创建灾难恢复副本（集群间复制）。
- **负载共享镜像。**可从一个卷复制到同一集群中的多个卷，以便在集群中分布只读工作负载。

SnapMirror 复制的基本知识

当计划程序触发复制更新时，将执行以下操作：

在源卷上创建一个新的 Snapshot 副本。

确定新的 Snapshot 副本与上次复制的 Snapshot 副本之间的块级差别，然后将块级差别传输到目标卷。期间会传输上次复制的 Snapshot 副本与新副本之间创建的其他 Snapshot 副本。

完成传输后，新的 Snapshot 副本将位于目标卷上。

如果 SnapMirror 目标卷使用通用互联网文件系统 (Common Internet File System, CIFS) 协议进行共享并使用网络文件系统 (Network File System, NFS) 协议导出，则可以对其进行只读访问。所复制的卷中的逻辑单元号 (Logical Unit Number, LUN) 可供能够连接到只读 LUN 的客户端使用。

复制在卷级进行。可以在集群模式 Data ONTAP 操作系统中创建 qtree，并随复制的卷一起复制这些 qtree。但是，不能单独复制各个 qtree。

进行故障转移后，无需重新复制整个卷，即可在任一方向上重新同步灾难恢复关系。如果反向重新同步某个关系，则只会将上次成功同步 Snapshot 副本后写入的新数据发送回目标卷。

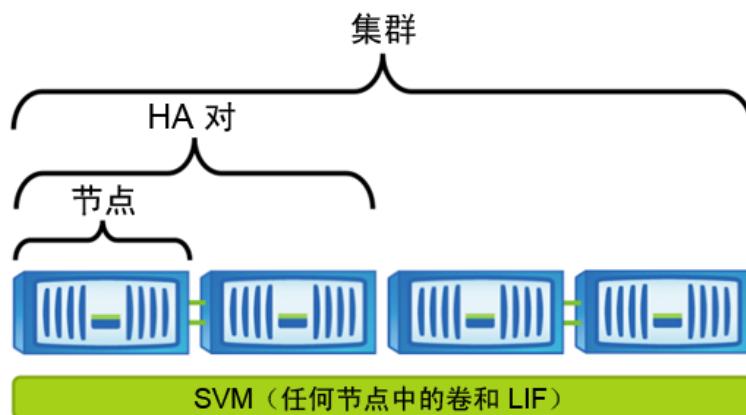
集群模式 Data ONTAP 8.1 操作系统中的 SnapMirror 技术关系必须由集群管理员管理，不能将该管理工作委派给 Storage Virtual Machine (SVM) 管理员。从集群模式 Data ONTAP 8.2 操作系统开始，集群管理员可以将 SnapMirror 关系的管理工作委派给 SVM 管理员。

2.2 集群模式 Data ONTAP 概述

集群模式 Data ONTAP 8.1 操作系统以及更高版本中使用的部分基本术语如下：

- **集群**。包含一个或多个相互连接并作为单一系统进行管理的节点。
- **集群互连**。一种专用的高速、低延迟私人网络，用于在同一集群的各个节点之间进行通信和复制。
- **集群模式 Data ONTAP 操作系统**。一种 Data ONTAP 操作模式，该模式支持将节点相互连接在一起并组成集群。
- **数据网络**。客户端访问数据所使用的网络。
- **HA 互连**。一个高可用性 (HA) 对中两个节点之间的专用互连。
- **HA 对**。成对配置的两个节点，用于实现高可用性 (HA)。
- **Ifgrp**。一系列物理端口的集合，这些端口组合在一起形成一个逻辑端口以用于链路聚合；一个集成组。
- **LIF**。一种分配了 IP 地址的逻辑接口，用于为集群中的特定节点提供以太网访问点。
- **集群间 LIF**。一种仅用于集群间复制且仅分配给一个节点的逻辑接口 (LIF)。
- **集群间网络**。用于在不同集群间进行通信和复制的网络。
- **管理网络**。用于管理集群、SVM 和节点的网络。
- **节点**。单个 NetApp 控制器，高可用性对中的一个控制器。
- **端口**。物理端口（如 e0e 或 e0f）或逻辑端口（如虚拟 LAN [VLAN] 或接口组 [ifgrp]）。
- **Storage Virtual Machine (SVM)**。一种逻辑存储服务器，用于提供从一个或多个逻辑接口 (LIF) 到 LUN 和（或）网络连接存储 (NAS) 命名空间的数据访问。

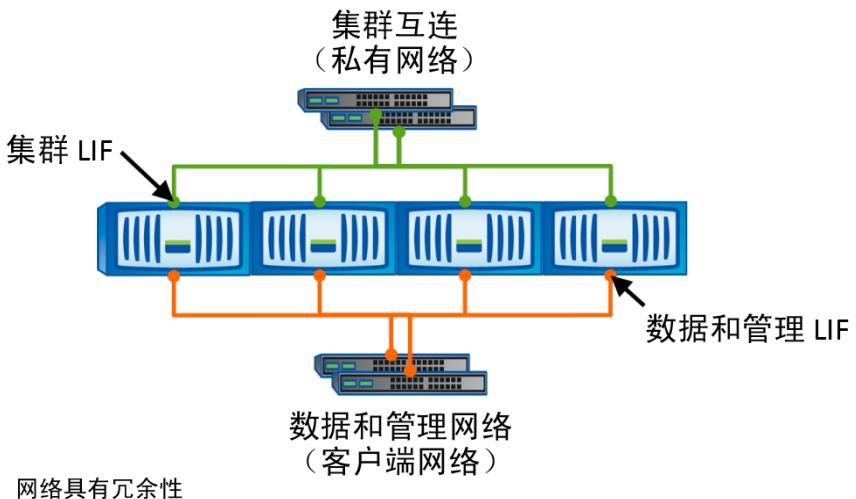
图 8) 节点、HA 对、集群和 SVM。



一个集群模式 Data ONTAP 解决方案中有多种类型的网络，如图 9 和图 10 中所示。了解每种网络类型的用途非常重要。

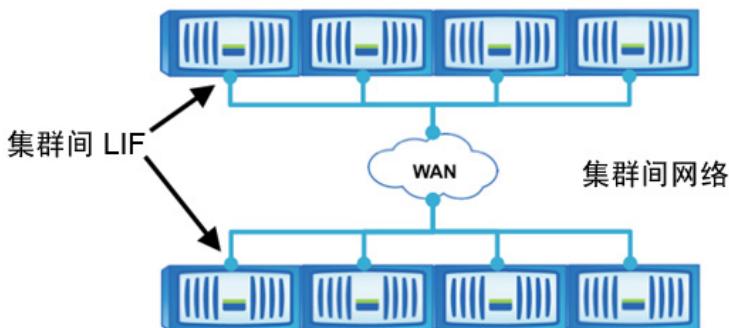
集群互连网络是一种专用的高速、低延迟私有网络，用于在同一集群内的各个节点之间进行通信和复制。这是一个冗余的后端网络，不能供客户端访问数据使用或进行共享，也不能用于管理集群、节点或 SVM。客户端通过数据网络访问数据。集群、节点和 SVM 则通过管理网络进行管理。数据网络和管理网络可能会共享相同的端口或物理网络，但它们必须是与集群互连网络不同的其他物理网络。

图 9) 集群互连网络以及数据网络和管理网络。



集群间网络可以使运行集群模式 Data ONTAP 操作系统的两个不同集群彼此通信并进行数据复制，如图 10 中所示。此网络可以是包含专用物理端口的网络，也可以是与数据网络和（或）管理网络共享端口的网络。下一节将详细介绍论集群间网络。

图 10) 集群间网络。



3 在不同集群之间进行复制所需的网络配置

集群模式 Data ONTAP 8.1 操作系统是首款可在不同集群之间进行复制的版本，可提供跨站点灾难恢复功能。集群模式 Data ONTAP 8.1 操作系统引入了一些新功能，用于满足以下用途：

- **创建集群对等关系。**连接两个集群以便在其间进行复制的操作。
- **集群间逻辑接口。**用于集群间通信的逻辑网络接口。
- **集群间端口。**专用于集群间复制的端口。

必须先将集群加入对等关系，然后才可在不同的集群之间进行复制。创建集群对等关系是一次性操作，必须由集群管理员执行。在不同集群间复制需要执行以下步骤：

1. 清楚地了解集群对等关系的创建。
2. 确定是否共享端口用来进行数据访问和集群间复制。
3. 如果使用专用端口，则指定用于集群间复制的端口。
4. 在集群中的每个节点上创建集群间 LIF。
5. 创建集群对等关系。

必须创建集群对等关系，因为它会定义在不同集群间进行所有复制所使用的网络。此外，自集群模式 Data ONTAP 8.2 操作系统起，要想在不同的 SVM 之间进行复制，必须先将 SVM 加入对等关系。SVM 对等关系将在第 4 节中介绍。

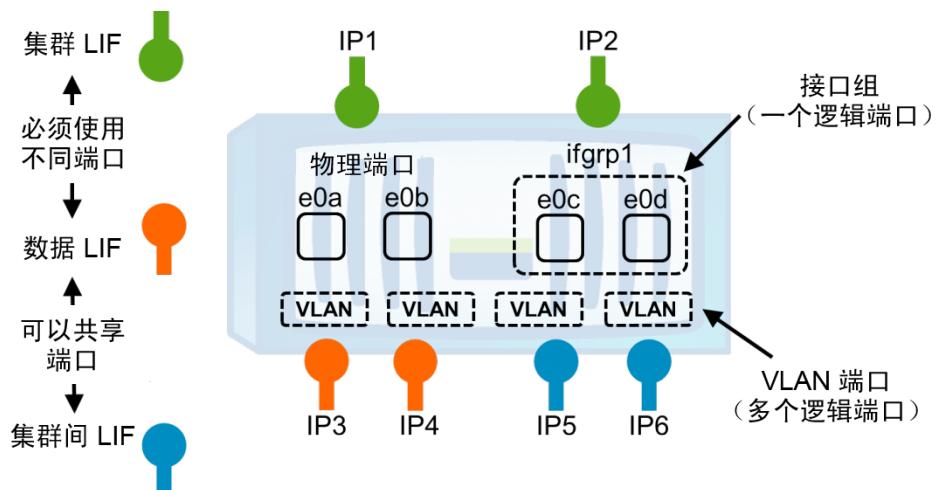
3.1 集群间网络连接

创建集群对等关系需要通过集群间逻辑接口 (LIF) 在两个集群之间进行通信和复制。集群间 LIF 必须创建于具有集群间能力的端口上，即分配了集群间角色或数据角色的端口。用于集群内集群互连的端口不可用于集群间复制。

您可以将集群间 LIF 分配给具有数据角色的端口，即用于 CIFS、NFS 或 iSCSI 访问的端口。但是，要使某些端口专用于集群间复制，只需将集群间 LIF 分配给用于集群间的专用端口，因为集群间端口仅可用于复制。本文档后面的内容会介绍各种方法以及如何确定使用哪种方法。

了解 LIF 与端口的不同非常重要。LIF 是一种分配了 IP 地址的逻辑接口，用于为集群中的特定节点提供访问点。LIF 会分配给端口，而端口类型多种多样。端口可以是 NetApp 控制器上已插入缆线的物理端口，如 e0e 或 e0f，也可以是逻辑端口，如 VLAN 或 ifgrp。Ifgrp 用于以太网链路聚合。图 11 显示了不同类型的端口。

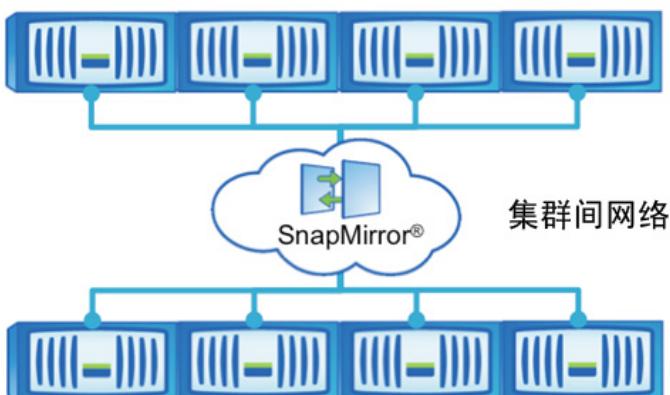
图 11) 端口类型。



系统中的任何以太网端口类型都可以具有数据角色或集群间角色，并可用于集群间复制，其中包括诸如 e0e 或 e0f 等物理端口以及诸如虚拟 LAN (VLAN) 或 ifgrp 等逻辑端口。

在集群中的每个节点上必须至少创建一个集群间 LIF。每个集群间 LIF 都需要 IP 地址，这表示每个节点必须至少有一个 IP 地址专用于执行集群间复制。不支持仅在集群的部分节点上配置集群间 LIF 的做法。必须在每个节点上配置一个集群间 LIF，这样，当卷移动操作在节点间移动卷时，才能继续进行集群间 SnapMirror 传输，如图 12 中所示。SnapMirror 关系不必进行修改，因为卷会移至集群中的其他节点。

图 12) 在集群间网络上执行 SnapMirror。



创建集群间 LIF 即会定义集群间网络，通过该网络可在两个不同集群之间进行复制。两个集群之间的复制只能在集群间网络上进行，即使集群间网络与同一集群中的数据网络位于相同子网上也是如此。其他协议（如 CIFS、NFS、iSCSI、FC 或 FCoE）无法使用集群间 LIF 或端口。分配给集群间 LIF 的 IP 地址可以与数据 LIF 位于同一子网，也可以位于不同子网。创建集群间 LIF 时，会在该节点上自动创建一个集群间路由组。如果源集群和目标集群必须使用不同的子网进行集群间复制，则必须为该集群间路由组定义一个网关地址。

集群间 LIF 的范围仅限于节点内，因此，在托管集群间 LIF 的端口出现故障时，根据 LIF 的故障转移策略，LIF 只能故障转移至同一节点上另一个支持集群间操作的端口。每个节点至少具有 1 个集群间 LIF，才能在集群之间进行复制。请保持各个集群间 LIF 设置一致（相同的 MTU、流量控制、TCP 选项等）。如果正在进行集群间 SnapMirror 传输时某个节点发生故障，则会自动使用高可用性 (HA) 对中未受故障影响的节点上的集群间 LIF 继续进行传输。在集群模式 Data ONTAP 8.2 操作系统中，在对目标进行存储故障转移 (Storage Failover, SFO) 后，相同的传输不会自动继续进行。如果在源上发生 SFO，则会继续传输。但是，复制操作照样会自动继续在未受故障影响的节点上进行。

从集群模式 Data ONTAP 模式下运行的集群模式 Data ONTAP 8.1 开始，在不同集群之间进行复制需要在集群间建立 IP 连接。集群模式 Data ONTAP 操作系统不支持通过 FC 网络进行 SnapMirror 复制。

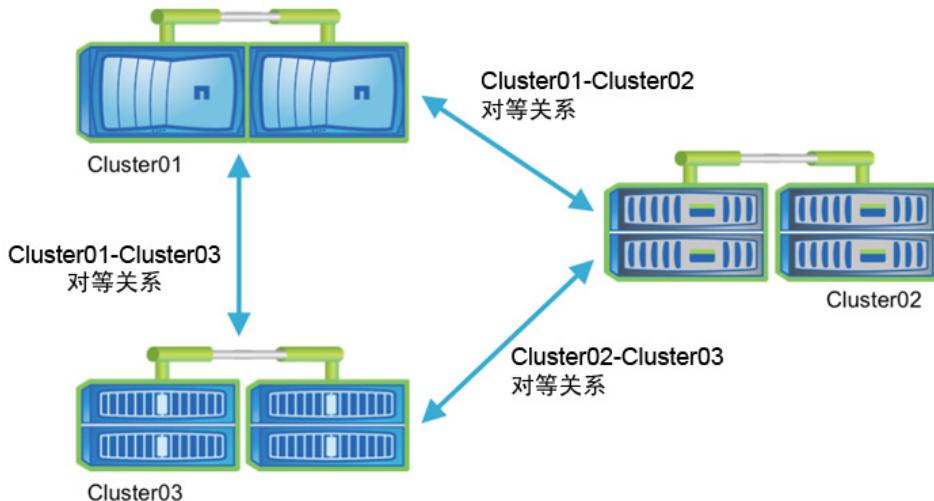
3.2 创建集群对等关系

创建集群间 LIF 并配置集群间网络之后，便可以创建集群对等。集群对等方也是一个集群，可支持与另一个集群之间进行数据复制。

创建集群对等关系是一次性操作，必须由集群管理员执行。对等关系可以通过两种方法来创建。一种方法是由拥有另一集群的安全凭据（集群管理登录名和密码）的集群管理员来创建对等关系。另一种方法则允许两位管理员在不交换集群管理密码的情况下为其集群建立对等关系。使用这种方法时，两位管理员需各自输入 `cluster peer create` 命令，指定对方集群的集群间 IP 地址。

如图 13 中所示，一个集群最多可以与八个集群建立对等关系，从而使多个集群能够彼此复制数据。

图 13) 多集群对等关系。



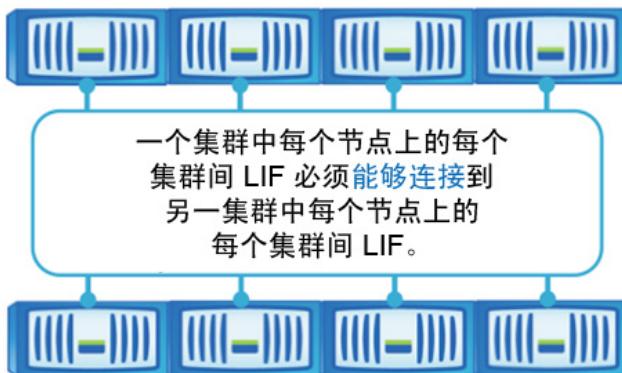
3.3 集群对等要求

集群对等包括以下要求：

- 集群上的时间必须在 300 秒（即五分钟）内同步才可使对等成功。对等的集群可以位于不同的时区。
- 在集群中的每个节点上必须至少创建一个集群间 LIF。
- 每个集群间 LIF 都需要专用于集群间复制的 IP 地址。
- 必须在用于复制的网络端口上使用正确的最大传输单元 (MTU) 值。网络管理员可以确定要在具体环境中使用的 MTU 值。默认值为 1,500，适用于大多数环境。

- 用于集群间复制的某个节点上的所有路径都应具有相同的性能特性。
- 集群间网络必须在集群对等中所有节点上的所有集群间 LIF 之间建立连接。集群中每个节点上的每个集群间 LIF 必须能够连接到对等集群中每个节点上的每个集群间 LIF，如图 14 中所示。

图 14) 集群间 LIF 的完全连接。



3.4 集群间多路径和网络冗余

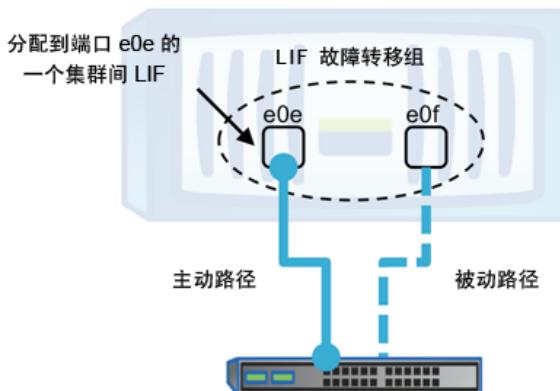
NetApp 集群模式 Data ONTAP 8.1 操作系统可通过以下功能为集群间 SnapMirror 复制配置两种多路径模式：

- **主动-被动。** 使用一个特定路径进行复制，只有在该路径发生故障时，才使用另一个路径。
- **主动-主动。** 同时主动使用多个路径进行复制。当一个路径发生故障时，无故障的路径仍然处于活动状态，所有复制传输仍可继续进行。

Data ONTAP 中的主动-被动集群间多路径

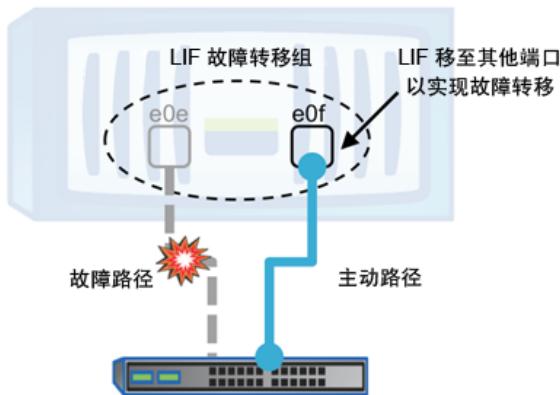
在主动-被动故障转移方面，集群间 LIF 的行为与 CIFS 或 NFS 所使用的 LIF 有很多相同之处，不同之处则在于集群间 LIF 不能故障转移到不同节点上的某个端口。最初选择将 LIF 放在哪个特定端口，决定了该 LIF 使用哪个端口。如果对端口和 LIF 进行配置，使 LIF 可以故障转移到同一节点上的其他端口，从而让 LIF 具有冗余功能，则最初放置 LIF 的端口为主动路径，而 LIF 可以故障转移到的任何端口为被动路径。因此，可以说经过适当配置的冗余 LIF 可提供主动-被动多路径功能，如图 15 中所示。

图 15) 主动-被动多路径。



集群间 LIF 上的通信只会在分配了该 LIF 的端口上进行，只有该端口发生故障时，该 LIF 才会移至其故障转移组中的另一个无故障端口。

图 16) LIF 故障转移期间的主动-被动多路径功能。

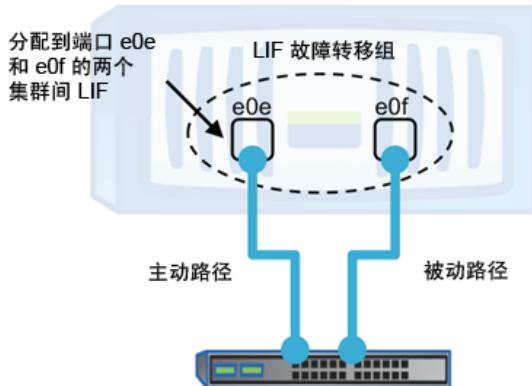


要配置主动-被动多路径功能，请将一个集群间 LIF 分配给一个具有集群间能力的端口，然后确保再配置一个具有集群间能力且支持该连接的端口。请确保对 LIF 的故障转移策略进行配置，使 LIF 的故障转移组包含进行故障转移所需的端口，如图 16 中所示。

Data ONTAP 中的主动-主动集群间多路径功能

要实现主动-主动多路径功能，要求在节点上配置另外的集群间 LIF。SnapMirror 使用源节点和目标节点上所有可用的集群间 LIF 来为这两个节点之间所有传输数据的 SnapMirror 关系发送和接收数据。如果配置了两个集群间 LIF，并且可以使两个端口进行集群间通信，则可以为每个端口分配一个 LIF，而 SnapMirror 将同时使用这两个端口，如图 17 所示。

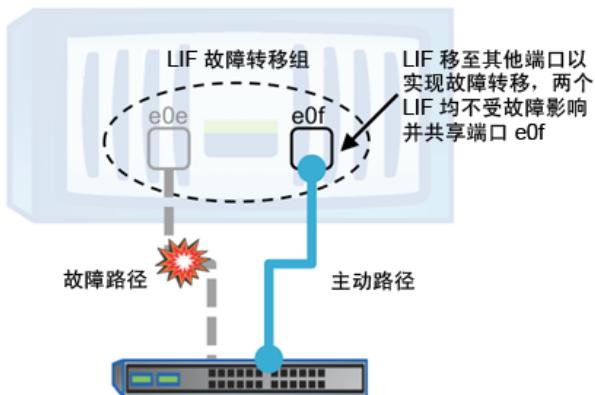
图 17) 主动-主动多路径。



在集群模式 Data ONTAP 8.1 操作系统中，节点上采用主动-主动配置为集群间复制提供的所有路径具有同等的性能特性，这一点非常重要。配置多路径功能时，如果使一个集群间 LIF 位于慢速路径上，一个位于快速路径上，则会对性能产生不利影响，因为这样数据会同时在慢速和快速路径上进行多路传输。从集群模式 Data ONTAP 8.2 操作系统开始，可使用不同网络类型和速度配置 SnapMirror 多路径功能，而不会对较快端口上的复制性能产生不利影响。

由于每个端口都分配了一个集群间 LIF，因此通信可以同时在两个端口上进行。如果一个端口发生故障，故障端口上的 LIF 将移至其故障转移组中的另一无故障端口。根据故障转移组中的端口数量，多个 LIF 现在可共享一个端口，如图 18 中所示。

图 18) LIF 故障转移期间的主动-主动多路径。



要为 SnapMirror 配置采用两个路径的主动-主动多路径功能, 请配置两个具有集群间能力的端口 (具有“集群间”或“数据”角色), 创建两个集群间 LIF, 然后为每个端口分配一个 LIF。请确保对 LIF 的故障转移策略进行配置, 使 LIF 的故障转移组包含进行故障转移所必要的端口。

根据给定的任意一对源节点与目标节点之间的复制工作负载, 可能有必要在该源节点和目标节点上配置多个路径。要使用多路径连接, 无需为每个 SnapMirror 关系应用特殊的配置设置。所有 SnapMirror 关系将自动在源节点和目标节点上的可用 LIF 之间进行多路传输。

通过基于交换机的链路聚合实现多路径功能

如本文档前面所述, 可以将集群间 LIF 分配给系统中任何类型的端口, 包括 ifgrp 这样的逻辑端口。ifgrp 支持基于交换机的链路聚合。您可以将多个物理端口配置成一个 ifgrp, 然后将集群间 LIF 分配给该 ifgrp 端口。之后, 可以通过链路聚合技术将多个交换机端口组合在一起, 以提供多路径和(或)冗余功能。

基于交换机的链路聚合并不能保证 ifgrp 中的多个物理路径会同时使用。例如, 假定在源节点和目标节点上配置了一个集群间 LIF; 此时, 每个节点都会具有一个用于集群间通信的 IP 地址和一个双端口 ifgrp。如果该 ifgrp 使用基于 IP 哈希的负载平衡方法, 则只会在一对源 IP 地址和目标 IP 地址上执行负载平衡哈希。链路可能会通过该端口组中的同一条路径在两个节点间进行所有连接。

请注意, 复制可能会发生在多个节点之间; 例如, 一个节点可能会向远程集群中的不同节点复制不同的卷。每个节点都有不同的集群间 LIF, 这些集群间 LIF 拥有不同的源 IP 地址和目标 IP 地址对, 以使链路内的多个路径能够用于这一特定源节点。

如果使用基于交换机的链路聚合, 以便在两个特定节点之间进行复制时使用 ifgrp 中的多个物理通道, 则可以在两个节点中的任意一个节点上配置其他集群间 LIF。Data ONTAP 会自动在 SnapMirror 的源节点和目标节点上的每个 LIF 之间建立一个连接。这样可以为负载平衡哈希提供其他源 IP 地址与目标 IP 地址的组合, 这些组合可放置在链路内的不同路径上。但是, 在此示例中, 在一个节点上配置多个 LIF 的目的是, 使任意两个特定节点之间的复制操作可以使用多个路径。在许多 WAN 复制方案中, 可能无需执行此操作, 因为 WAN 带宽可能远远小于 ifgrp 中合并链路的带宽。在两个特定节点之间启用多个路径可能并无益处, 因为无论如何, 都会有许多节点共享 WAN 带宽。

最佳实践

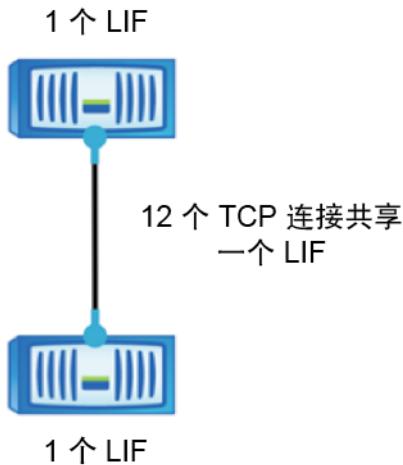
使用基于交换机的链路聚合时, 请使用 `multimode_lacp` 模式创建 ifgrp, 并将 ifgrp 的分布功能设置为 `port`。使用 `port` 值实现分布功能会对 ifgrp 进行配置, 使其通过计算源/目标 IP 地址以及所用端口的哈希在各个路径间分布连接。这种做法并不保证将在 ifgrp 中的所有路径间平均分布连接, 但确实允许使用 ifgrp 中的多个物理链路。

3.5 集群间 SnapMirror 的网络连接

在集群模式 Data ONTAP 操作系统中，集群间 LIF 的数量决定了 SnapMirror 源节点与目标节点之间建立的传输控制协议 (Transmission Control Protocol, TCP) 连接的数量。系统并不按每个卷或每个关系来创建 TCP 连接。

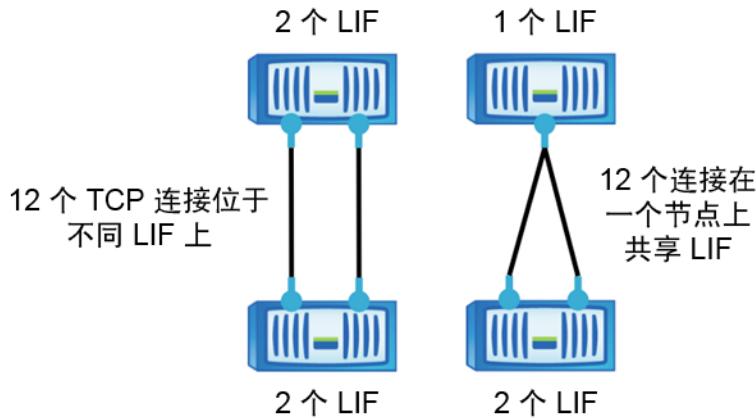
请记住，从集群模式 Data ONTAP 8.2 操作系统开始，Data ONTAP 至少会建立 12 个集群间 TCP 连接用于发送数据。如图 19 中所示，至少会创建 12 个 TCP 连接用于发送数据。即使源节点和目标节点均只有一个集群间 LIF 也是如此，并且会创建足够多的连接，以便能够使用源节点和目标节点上的所有集群间 LIF。

图 19) 使用一个集群间 LIF 的 TCP 连接。



如果为源节点和（或）目标节点配置了 2 个集群间 LIF，则 Data ONTAP 会建立 12 个 TCP 连接用于发送数据；但是，其中两个连接并不会使用相同的 LIF，而是一个连接使用一个 LIF 对，另一个连接使用另一个 LIF 对，如图 20 中所示。此示例显示了不同的集群间 LIF 组合，这些组合会产生 12 个集群间 TCP 连接。您无法选择某个 TCP 连接所使用的特定 LIF 对；Data ONTAP 会自动对其进行管理。

图 20) 使用两个集群间 LIF 的 TCP 连接。



当节点上的集群间 LIF 的规模超过 12 个以后，Data ONTAP 会创建另外的集群间 TCP 连接，并且数量足以保证所有集群间 LIF 都能得以利用。

随着更多集群间 LIF 添加到源节点或目标节点，会继续创建更多集群间 TCP 连接。目前在 Data ONTAP 中，一个节点最多支持 24 个集群间连接用于 SnapMirror。

最佳实践

虽然不是必需的，但是为了操作的一致性，您可以同时在源节点和目标节点上配置相同数量的集群间 LIF。如“用于多路径的基于交换机的链路聚合”一节所述，可以创建多个集群间 LIF，以便在多个物理路径间启用主动-主动多路径功能。

例如，如果一个节点配置了四个千兆以太网 (GbE) 端口用于集群间复制，则需要四个集群间 LIF，每个端口分配一个，以确保使用所有路径提供超出仅使用一个 1GbE 链路时的带宽。

3.6 确定使用共享端口还是专用端口进行复制

确定使用共享端口还是专用端口进行复制时，需要考虑一系列配置和要求，包括：

- **LAN 类型。** 1GbE 还是 10GbE 连接。
- **可用的 WAN 带宽（与 LAN 带宽相比）。** 如果可用的 WAN 带宽远低于 LAN 带宽，则 WAN 可以充当带宽限制。
- **复制时间间隔。** 在非生产时段进行的复制对数据网络的影响可能无关紧要。
- **更改率。** 需要复制的数据量不得妨碍客户端的数据访问。
- **该解决方案使用的端口数。** 要使用专用端口进行复制，需要使用额外的交换机端口，并增加缆线布线。

可以手动配置集群间 LIF 故障转移策略，以便集群间 LIF 仅使用一部分数据端口。但是，在通过为端口分配集群间角色来使用特定专用端口进行集群间操作时，Data ONTAP 会自动配置 LIF 故障转移组，以便仅使用集群间端口进行复制。此时不允许任何数据协议故障转移或迁移到集群间端口。这是使端口专用于集群间复制的更为严格且自动执行的方法。

在为集群间复制使用共享端口时，考虑以下内容：

- 对于诸如 10GbE 这样的高速网络，在用于数据访问的相同 10GbE 端口上可能具有足够的本地 LAN 带宽量来执行复制。在许多情况下，可用的 WAN 带宽远低于 10GbE，这需要将 LAN 网络利用率降至 WAN 能够支持的水平。
- 集群中的所有节点都可能需要复制数据，并共享可用 WAN 带宽，因此，更多的人会支持数据端口共享。
- 为数据和复制使用共享端口可消除使用专用复制端口所需的额外端口数。
- 如果设置了复制间隔，使其在工作时间之外进行复制，则在客户端活动很少或没有客户端活动时，可以使用数据端口进行复制，即使没有 10GbE LAN 连接也毫无影响。
- 如果共享数据端口，则需考虑数据变更率和复制时间间隔，以及每个时间间隔内必须复制的数据量所要求的带宽是否会导致与数据协议之间出现争用情况。
- 共享用于集群间复制的数据端口时，集群间 LIF 可以迁移至同一节点上其他任何具有集群间能力的端口，控制用于复制的特定数据端口。

在为集群间复制使用专用端口时，考虑以下内容：

- 如果可用的 WAN 带宽量与 LAN 端口的带宽量相似，并且设置了复制间隔，使其在进行常规客户端活动时执行复制，请使用专用以太网端口执行集群间复制，从而避免复制协议和数据协议之间出现争用情况。
- 如果数据协议 (CIFS、NFS、iSCSI) 产生的网络利用率高于 50%，则使用专用端口进行复制，以便在节点故障转移时不会降低性能。
- 将物理 10GbE 端口用于数据和复制时，可以创建 VLAN 端口用于复制，并创建逻辑端口专用于集群间复制。
- 如果共享数据端口，则需考虑数据变更率和复制时间间隔，以及每个时间间隔内必须复制的数据量所要求的带宽是否会导致与数据协议之间出现争用情况。
- 如果复制网络要求配置与数据网络上使用的 MTU 大小不同的 MTU，则必须将物理端口专用于复制，因为只能在物理端口上配置 MTU 大小。

3.7 共享要进行集群间复制的数据端口

本部分演示如何创建共享数据端口的集群间 LIF。在此示例中，存在一个双节点集群，其中每个节点具有两个数据端口 e0c 和 e0d，这两个端口共享用于集群间复制。

配置集群间 LIF 以共享数据端口

在完成以下步骤之前，将端口、网络、IP 地址、子网掩码和子网更换为特定于环境的设置。

1. 检查集群中端口的角色。

```
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
```

2. 在 cluster01 中的每个节点上创建一个集群间 LIF。此示例对集群间 LIF 使用的 LIF 命名约定为 <节点名>_ic1#。

```
cluster01::> network int create -vserver cluster01-01 -lif cluster01-01_ic101 -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_ic101 -role
intercluster -home-node cluster01-02 -home-port e0d -address 10.12.12.2 -netmask
255.255.255.0
```

最佳实践

集群间 LIF 的范围仅限于节点内（它们只能故障转移到同一节点上的其他端口）。因此，请按以下命名约定来命名集群间 LIF：在节点名称后加上 ic 或 icl 字样以表示集群间 LIF；例如，node_name_ic1# 或 node-name-ic#，具体取决于您的首选设置。

3. 验证是否已正确创建集群间 LIF。

```
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_ic101
          up/up      10.12.12.1/24    cluster01-01  e0c      true
cluster01-02
          cluster01-02_ic101
          up/up      10.12.12.2/24    cluster01-02  e0c      true
```

4. 验证集群间 LIF 是否配置为冗余。

```
cluster01::> network int show -role intercluster -failover
          Logical      Home          Failover      Failover
Vserver   Interface  Node:Port  Group Usage  Group
-----  -----
cluster01-01
          cluster01-01_ic101  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

```

注：本示例中的 LIF 在每个节点上都分配了 e0c 端口。如果 e0c 端口出现故障，LIF 可以故障转移至 e0d 端口，因为 e0d 也具有数据角色。集群间 LIF 会分配给一个数据端口，因此会自动为集群间 LIF 创建一个故障转移组，其中包含该节点上具有数据角色的所有端口。集群间故障转移组是专为特定节点提供的，如果需要进行更改，则必须对每个节点的故障转移组进行管理，因为不同节点可能使用不同的端口进行复制。

最佳实践

验证所有必要的端口是否能够访问必要的网络或 VLAN，以便在端口故障转移后进行通信。集群间 LIF 只能根据故障转移组中的定义故障转移至同一节点上的其他集群间端口（在本示例中为其他数据端口），而无法故障转移至其他节点上的端口。为集群间复制使用专用端口时，最好的做法是在每个节点上配置两个集群间端口。

5. 系统将为集群间 LIF 自动创建集群间路由组。运行 `net routing-group show` 命令来显示路由组。

注：集群间路由组以 i 开头。

```

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

```

6. 集群间网络可能需要集群间路由。显示集群中的路由。

注：没有集群间路由可用。

```

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

```

7. 如果不同集群中集群间 LIF 之间的通信需要路由，则创建集群间路由。集群间网络的范围限定在各个节点内，因此，需要在每个节点上创建集群间路由。在此示例中，10.12.12.1 是 10.12.12.0/24 网络的网关地址。

注：仅当两个集群中的集群间地址不在同一子网上并且需要特定路由以在集群之间进行通信时，才需要创建路由。

```

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

```

注: 如果目标指定为 0.0.0.0/0, 则它将成为集群间网络的默认路由。

8. 显示新创建的路由。

```
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
```

注: 尽管并未向集群间路由分配角色, 但集群间路由会分配给路由组 i10.12.12.0/24, 而该路由组分配有集群间角色, 如该 net routing-group show 命令的输出内容中所示。这些路由仅用于集群间通信。

9. 重复本部分中的步骤在其他集群中配置集群间网络。

3.8 使用专用集群间端口进行集群间复制

本部分演示如何将特定端口指定为集群间端口, 以及如何将集群间 LIF 分配给这些端口。此示例提供了一个双节点集群, 其中每个节点都有两个数据端口 e0e 和 e0f, 这两个端口专用于集群间复制。在完成以下步骤之前, 请将端口、网络、IP 地址、子网掩码和子网替换为适用于您环境的设置。

配置集群间 LIF 以使用专用集群间端口

在完成本节中的步骤之前, 请查看“集群对等要求”一节, 并确定在您的环境中是使用共享端口还是专用端口来进行复制。

1. 检查集群中端口的角色。

```
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. 确定是否有任何 LIF 正在使用专用于复制的端口。

```
cluster01::> network int show
```

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

3. 如果某个数据 LIF 正在使用一个专用于复制的端口，则需要将该 LIF 迁移到另一端口，因为集群间端口不能托管数据 LIF。此迁移过程会无中断进行，并且假定已正确配置其他数据端口，以使客户端能够在迁移后访问该 LIF。

注：SAN LIF 不会在节点间迁移。如果当前端口配置了 iSCSI LIF，请确保 iSCSI 客户端中的多路径软件配置正确，能够不受 iSCSI LIF 故障的影响。然后，取消配置现有的 iSCSI LIF 并将其删除，然后在另一端口上重新创建该 LIF。iSCSI 客户端的多路径软件负责确保此重新配置操作不会造成客户端中断，因为在删除 LIF 后又会创建一个新的 LIF。

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

```
cluster01::> network int show
```

Vserver	Logical Interface	Status Admin/Oper	Network Address/Mask	Current Node	Current Port	Is Home
cluster01 vs1	cluster_mgmt	up/up	10.228.xx.xx/24	cluster01-01	e0c	true
	vs1_lif1	up/up	10.12.12.5/24		e0d	false

4. 新迁移的 LIF 可能需要修改 LIF 主端口，从而反映 LIF 应位于的新端口。

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

```
cluster01::> network int show
```

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

5. 从专用于复制的端口迁移出所有 LIF 后，将每个节点上使用的端口的角色更改为集群间。

```
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. 在两个集群之间成功创建集群对等关系后，创建集群间 SnapMirror 关系。

Node	Port	Role	Link	Auto-Negot			Duplex	Speed (Mbps)
				MTU	Admin/Oper	Admin/Oper		
cluster01-01	e0e	intercluster	up	1500	true/true	full/full	auto/1000	auto/1000
	e0f	intercluster	up	1500	true/true	full/full	auto/1000	auto/1000
cluster01-02	e0e	intercluster	up	1500	true/true	full/full	auto/1000	auto/1000
	e0f	intercluster	up	1500	true/true	full/full	auto/1000	auto/1000

7. 在 cluster01 中的每个节点上创建一个集群间 LIF。该示例中对集群间 LIF 使用的 LIF 命名约定为 <节点名>_ic1#。

```
cluster01::> network int create -vserver cluster01-01 -lif cluster01-01_ic101 -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_ic101 -role
intercluster -home-node cluster01-02 -home-port e0e -address 10.12.12.2 -netmask
255.255.255.0
```

最佳实践

集群间 LIF 的范围仅限于节点内（它们只能故障转移到同一节点上的其他端口）。因此，请按以下命名约定来命名集群间 LIF：在节点名称后加上 ic 或 ic1 字样以表示集群间 LIF；例如，node-name_ic1# 或 node-name-ic#，具体取决于您的首选设置。

8. 验证集群间 LIF 是否配置为冗余。

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

      Logical          Home          Failover          Failover
Vserver  Interface    Node:Port    Group Usage    Group
-----  -----
cluster01-01
      cluster01_ic101  cluster01-01:e0e  system-defined
                           Failover Targets: cluster01-01:e0e,
                                         cluster01-01:e0f
cluster01-02
      cluster01-02_ic101  cluster01-02:e0e  system-defined
                           Failover Targets: cluster01-02:e0e,
                                         cluster01-02:e0f
```

注：此示例中的 LIF 在每个节点上都分配有 e0e 主端口。如果 e0e 端口发生故障，LIF 可以故障转移到 e0f 端口，因为 e0f 端口也分配了集群间角色。集群间 LIF 分配给集群间端口，因此，会自动创建故障转移组，其中包含该节点上具有集群间角色的所有端口。在此示例中，故障转移组不含任何数据端口。集群间故障转移组是特定于节点的，因此，如果需要进行更改，必须对各个节点单独进行管理，因为不同的节点可能使用不同的端口进行复制。

最佳实践

验证所有必要的端口是否能够访问必要的网络或 VLAN，以便在端口故障转移后进行通信。集群间 LIF 只能根据故障转移组中的定义故障转移至同一节点上其他集群间端口（在本示例中为其他集群间端口），而无法故障转移至其他节点上的端口。为集群间复制使用专用端口时，最好的做法是在每个节点上配置两个集群间端口。

9. 验证是否已正确创建集群间 LIF。

```
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_ic101  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_ic101  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. 集群间网络可能需要集群间路由。系统将为集群间 LIF 自动创建集群间路由组。运行 `net routing-group show` 命令来显示路由组。

注：集群间路由组以 `i` 开头。

cluster01::> network routing-group show -role intercluster				
Vserver	Routing Group	Subnet	Role	Metric
<hr/>				
cluster01-01	i10.12.12.0/24	10.12.12.0/24	intercluster	40
<hr/>				
cluster01-02	i10.12.12.0/24	10.12.12.0/24	intercluster	40

11. 显示集群中的路由。

注：没有集群间路由可用。

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

12. 如果不同集群中集群间 LIF 之间的通信需要路由，则创建集群间路由。集群间网络的范围限定在各个节点内，因此，需要在每个节点上创建集群间路由。在此示例中，10.12.12.1 是 10.12.12.0/24 网络的网关地址。

注：仅当两个集群中的集群间地址不在同一子网上并且需要特定路由以在集群之间进行通信时，才需要创建路由。

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

注：如果目标指定为 0.0.0.0/0，则它将成为集群间网络的默认路由。

13. 显示新创建的路由。

cluster01::> network routing-group route show				
Vserver	Routing Group	Destination	Gateway	Metric
<hr/>				
cluster01	c10.288.225.0/24	0.0.0.0/0	10.288.225.1	20
<hr/>				
cluster01-01	n10.288.225.0/24	0.0.0.0/0	10.288.225.1	10
<hr/>				
cluster01-02	i10.12.12.0/24	0.0.0.0/0	10.12.12.1	40

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

注：尽管集群间路由没有分配角色，但集群路由会分配给路由组 `i10.12.12.0/24`，而该路由组分配有集群间角色，如输出内容中所示。这些路由仅用于集群间通信。

14. 重复本部分中的步骤在其他集群中配置集群间网络。在一个集群中使用专用端口进行复制并不要求在所有集群中都使用专用端口；可以在一个集群中使用专用端口，而在其他集群中使用共享数据端口进行集群间复制。在两种配置中，验证端口是否能够访问适当的子网、VLAN 等。

3.9 配置集群对等

两个集群中的节点都配置集群间 LIF 后，可以创建集群对等关系，以便在集群之间建立复制关系。

可以将一个集群与多达八个远程集群创建对等关系。每个集群对等关系必须单独创建。创建对等关系后，可以在处于该对等关系的两个集群相互间执行镜像。

在此示例中，`cluster01` 与名为 `cluster02` 的远程集群对等在一起。`cluster02` 还是一个双节点集群，每个节点具有一个集群间 LIF。在 `cluster02` 中创建的集群间 LIF 的 IP 地址为 `10.12.12.3` 和 `10.12.12.4`，它们会在以下 `cluster peer create` 命令示例中使用。但是，如果配置了 DNS 来解析集群间 IP 地址对应的主机名，则可以在 `-peer-addrs` 选项中使用主机名。集群间 IP 地址不太可能频繁更改，但是，如果使用主机名，则在更改集群间 IP 地址后不必修改集群对等关系。

在完成以下步骤之前，使用特定于您环境的设置替换集群名称和 IP 地址。

1. 运行 `cluster peer create` 命令创建集群对等关系，并在 `-peer-addrs` 选项中为远程集群中每个节点上至少一个集群间 LIF 指定 IP 地址，多个地址以逗号分隔。当系统提示时，提供远程集群管理员用户名和密码。

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

注：`cluster peer create` 操作所使用的凭据不会存储在远程系统中，系统不需要使用这些凭据来保持对等关系。这些凭据用于验证对等请求的身份，它们会以基于哈希的消息身份验证代码 MD5 (HMAC-MD5) 格式发送，而不是以明文格式发送。在完成身份验证并创建集群对等关系后，这些凭据将被销毁，不会再在集群间进行交换。

2. 显示新创建的集群对等关系。

```
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
```

注：在 `-peer-addrs` 选项中使用的 IP 地址作为远程集群间地址列出。这些地址用于发现能够在集群之间进行通信的所有其他集群间 IP 地址，并作为活动 IP 地址列出。活动 IP 地址列表可能包含更多地址，具体取决于每个节点存在的集群间 LIF 数，因为它们是自动发现的。

最佳实践

随着集群间 LIF 变为可用或不可用状态，活动 IP 地址的列表可能会发生变化。在发生某些事件（如节点重新启动）时，会自动发现活动 IP 地址。`-peer-addrs` 选项仅要求提供一个远程集群地址，但是，如果托管该地址的节点关闭并变为不可用，则可能不会重新发现集群对等关系。因此，最好的做法是，至少使用远程集群中每个节点上的一个集群间 IP 地址，这样，在节点发生故障时，对等关系会保持稳定。

3. 查看集群对等关系的运行状况。

```
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
          Data: interface_reachable
          ICMP: interface_reachable true      true      true
cluster01-02
          cluster02      cluster02-01
          Data: interface_reachable
          ICMP: interface_reachable true      true      true
          Data: interface_reachable
          ICMP: interface_reachable true      true      true
```

3.10 集群间 SnapMirror 的带宽限制

要限制集群间 SnapMirror 使用的带宽量，可以对集群间 SnapMirror 关系应用带宽限制。在创建新关系时，可以在命令行中添加 `-throttle` 选项和以千字节为单位的值来设置带宽限制，也可以通过执行 `snapmirror modify` 命令修改现有关系来设置带宽限制。NetApp OnCommand® System Manager 3.0 目前不支持 SnapMirror 带宽限制管理。在以下示例中，使用 `snapmirror modify` 命令对一个现有关系应用了 10 MB 的带宽限制。

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

要更改活动 SnapMirror 关系的带宽限制，请终止现有传输，然后重新启动传输以使用新值。SnapMirror 将使用新的带宽限制值从上次的重新启动检查点重新启动传输，而不是从头开始传输。

注： 集群内 SnapMirror 关系使用集群互连，不允许设置带宽限制。但是，从集群模式 Data ONTAP 8.2.1 操作系统开始，支持集群内限制，它与集群间限制的工作原理完全相同。

3.11 集群间 SnapMirror 的防火墙要求

请打开用于集群间复制的所有源节点和目标节点之间的集群间网络上的以下端口：

- 端口 11104
- 端口 11105

注： 集群模式 Data ONTAP 操作系统使用端口 11104 管理集群间通信会话，并使用端口 11105 传输数据。

4 SVM 对等关系

创建 SVM 对等关系是指将两个 SVM 连接在一起，以便在彼此之间进行复制（此功能从集群模式 Data ONTAP 8.2 操作系统开始提供）。您必须配置集群对等关系，才能在不同集群之间进行复制。在集群模式 Data ONTAP 8.1 操作系统中，任何 SVM 都可以将数据复制到同一集群或任何对等集群中的任何其他 SVM。复制安全性只能在集群范围内进行控制。从集群模式 Data ONTAP 8.2 操作系统开始，可以更加精细地控制 SnapMirror 安全性。复制权限必须通过创建 SVM 对等关系来定义。

在一对 SVM 之间创建任何 SnapMirror 关系之前，必须在这对 SVM 之间建立 SVM 对等关系。这些 SVM 可以位于本地（集群内），也可以位于远程位置（集群间）。创建 SVM 对等关系是一种基于权限的机制，该操作只需执行一次，并且必须由集群管理员来执行。要在不同 SVM 之间配置复制，需要执行以下步骤：

1. 清楚地了解如何建立 SVM 对等关系。
2. 创建 SVM 对等关系。
3. 在不同 SVM 之间创建 NetApp SnapMirror 关系。

最佳实践

使用唯一的完全限定域名 (FQDN) 来命名 SVM：例如，dataVserver.HQ 或 mirrorVserver.Offsite。创建 SVM 对等关系要求 SVM 名称必须唯一，而使用 FQDN 命名方式更便于确保这种唯一性。

4.1 SVM 对等要求

SVM 对等要求包括几项：

- 在为两个集群创建任何 SVM 对等关系之前，必须先为其创建集群对等关系。如果这些 SVM 位于同一集群内，则不受此限制。
- 如果这些 SVM 位于不同集群上，则执行“vserver peer create”和“vserver peer accept”。如果这些 SVM 位于同一集群上，则仅执行“vserver peer create”。
- SVM 对等关系中使用的 SVM 名称必须是唯一的。
- 两个 SVM 的语言必须相同（在集群模式 Data ONTAP 8.1 操作系统中）。从集群模式 Data ONTAP 8.1.1 操作系统开始，源卷和目标卷的语言类型必须相同。

4.2 配置 SVM 对等关系

可以使用 SVM 对等关系基础架构向 SVM 管理员委派 SnapMirror 关系管理中的创建/使用/删除任务。

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

SVM 的名称在集群间必须是唯一的。

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

创建 SVM 对等关系后，需要在目标集群上接受请求才能完成 SVM 对等关系的创建过程。使用 vserver peer show 命令可查看任何待定请求并验证 vserver peer create 命令是否有效。

```
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 数据保护关系

从集群模式 Data ONTAP 8.1 操作系统开始，可以使用两种类型的 SnapMirror 关系：DP 镜像和负载共享 (LS) 镜像。本节将讨论 DP 镜像；LS 镜像将在后面的小节中进行介绍。

DP 镜像可以在集群间进行，也可以在集群内进行。

- **集群间 DP 镜像。** 在运行集群模式 Data ONTAP 操作系统的所有集群中两个不同 SVM 上的卷之间进行复制。这类镜像主要用于为另一站点或位置提供灾难恢复。
- **集群内 DP 镜像。** 在同一集群中不同 SVM 上的两个卷之间进行复制，或者在同一 SVM 上的两个卷之间进行复制。这类镜像主要用于维护本地备份副本。

DP 镜像关系具有相同特征，无论复制是在集群内还是集群间进行。这些特征包括：

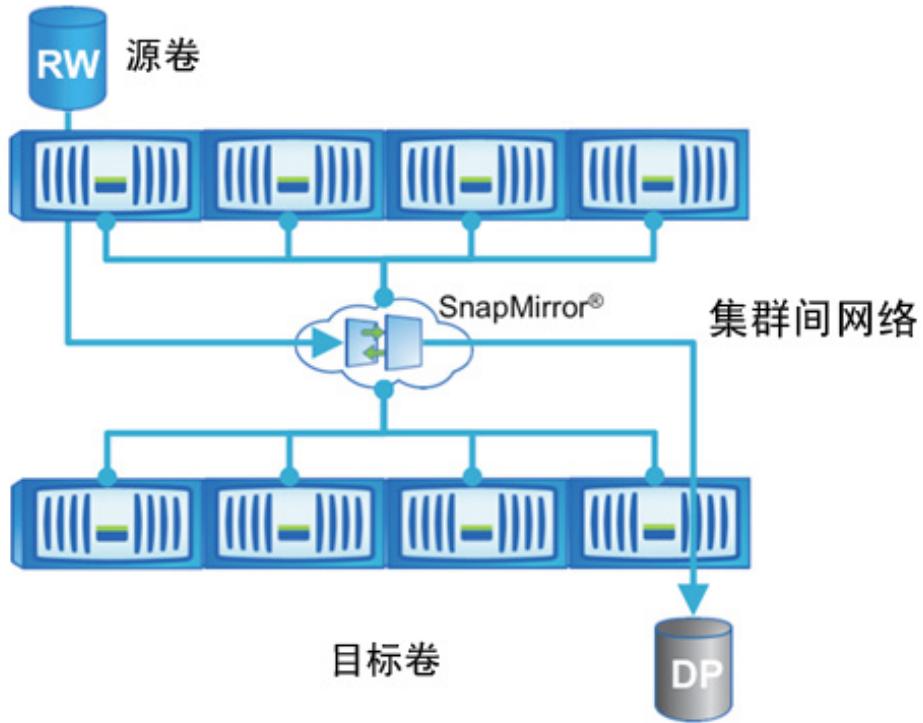
- DP 镜像关系在目标集群上创建和管理。
- DP 镜像关系传输由目标集群中的计划程序触发。
- 每个 DP 镜像目标卷都是一个单独的 SnapMirror 关系，独立于其他 DP 镜像卷执行；但是，不同的 DP 镜像关系可以使用同一个集群模式 Data ONTAP 操作系统计划条目。
- 创建 DP 和 LS 镜像的目标卷时，使用的卷类型必须为 DP 类型（-type 选项）。创建卷之后，存储管理员不能更改卷的 -type 属性。
- DP 镜像目标卷在故障转移前为只读状态。
- 可以使用 SnapMirror 中断操作对 DP 镜像目标卷进行故障转移，以使目标卷可写。必须分别对每个卷执行 SnapMirror 中断操作。
- 可以将 DP 镜像目标卷挂载到 SVM 命名空间中，同时仍保持只读状态，但只能在首次传输完成之后才能挂载。
- 不能将集群间 DP 镜像目标卷与源卷挂载在同一命名空间中，因为集群间 DP 镜像关系指向的是不同的集群，因此指向不同的 SVM，即不同的命名空间。
- 可以将集群内 DP 镜像目标卷与源卷挂载在同一命名空间中，前提是源卷和目标卷位于同一 SVM 中；但是，不能将它们挂载到同一挂载点。
- 可以将 DP 镜像目标卷中的 LUN 映射到 igrup 并连接到客户端；但是，客户端必须支持连接到只读 LUN。
- 可以使用集群模式 Data ONTAP 操作系统命令行界面 (CLI)、NetApp OnCommand System Manager 3.0 和 NetApp OnCommand Unified Manager 6.0 管理 DP 镜像关系。
- 如果网络故障导致进行中的传输中断或者管理员中止了传输，之后重新启动该传输时可自动从已保存的重新启动检查点继续传输。

从集群模式 Data ONTAP 8.2 操作系统开始，可以使用另外一种 SnapMirror 关系：XDP 存储。有关集群模式 Data ONTAP 8.2 操作系统中 SnapVault® 的详细信息，请参见 TR-4183。

SnapMirror 数据保护关系使用的网络

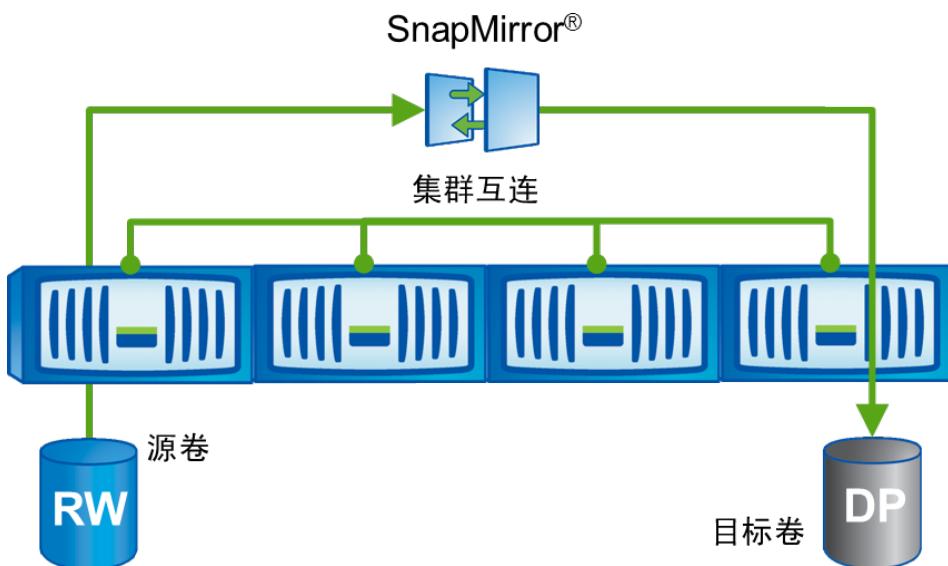
集群间和集群内 DP SnapMirror 关系在用于发送数据的网络方面有所不同。集群间 DP SnapMirror 关系使用集群间 LIF 定义的集群间网络。图 21 说明了用于 SnapMirror 的集群间网络。

图 21) 用于 SnapMirror 的集群间网络。



集群内 DP 镜像关系使用集群互连，这是用于在同一集群内的节点之间进行通信的私人连接。图 22 说明了用于集群间 SnapMirror 的集群互连。

图 22) 用于集群间 SnapMirror 的集群互连。



5.1 SnapMirror 数据保护关系

在两个集群之间成功创建集群对等关系和 SVM 对等关系之后，可以创建集群间 SnapMirror 关系。在同一集群中的两个 SVM 之间或同一 SVM 中的两个卷之间镜像数据并不需要建立对等关系。

源 SVM 和目标 SVM 必须具有相同的语言类型设置，才能在它们之间进行复制（在集群模式 Data ONTAP 8.1 操作系统中）。从集群模式 Data ONTAP 8.1.1 操作系统开始，源卷和目标卷的语言类型必须相同。一旦创建 SVM 就无法再更改其语言类型。

集群间 SnapMirror 关系主要用于在另一站点或位置提供灾难恢复功能。如果已使用 SnapMirror 将所有必要的卷复制到灾难恢复站点，则可以执行恢复操作，以便从灾难恢复站点恢复运行。

在集群模式 Data ONTAP 操作系统中创建 SnapMirror 关系并不需要将 SVM 主机名解析为 IP 地址。集群名称通过对等关系进行解析，而 SVM 名称则通过集群在内部进行解析。在集群模式 Data ONTAP 中可以使用源和目标 SVM 的主机名以及集群来创建 SnapMirror 关系，而无需使用 LIF 的 IP 地址。

集群间 SnapMirror 的要求

在创建集群间 SnapMirror 关系之前，满足以下要求：

- 为集群间网络配置源节点和目标节点。
- 配置对等关系中的源集群和目标集群。
- 创建一个与源 SVM 语言类型相同的目标 SVM；在集群模式 Data ONTAP 操作系统中，卷必须具有 SVM（集群模式 Data ONTAP 8.1 操作系统）。
- 源 SVM 和目标 SVM 可以具有不同的语言类型，但是源卷和目标卷“必须”具有相同的语言类型。从集群模式 Data ONTAP 8.1.1 操作系统开始，SVM 的语言类型只能在创建 SVM 时设置。
- 配置源 SVM 和目标 SVM 的对等关系。
- 创建一个类型为数据保护 (DP) 的目标卷，该卷需大于或等于源卷。
- 在目标集群中为 SnapMirror 关系分配一个计划，以执行定期更新。如果任何现有计划不充分，则必须创建新计划条目。

SVM 的扇出和扇入

可以在不同的 SVM 之间扇出或扇入卷。例如，在复制源集群中某个 SVM 上的多个不同卷时，可以将每个卷各自复制到目标集群的不同 SVM 中，这种方式称为扇出。或者，也可以将源集群中不同 SVM 上的多个不同卷复制到目标集群的一个 SVM 中，这种方式称为扇入。

最佳实践

要通过复制实现灾难恢复，可以将源集群中给定 SVM 上所有必要的卷镜像到目标集群中与之匹配的特定 SVM。另外，设计时还应考虑将应驻留在相同 SVM 中的一组给定卷镜像到灾难恢复站点上的相似 SVM 中。要使不同的卷可以从同一命名空间访问，这些卷必须位于同一个 SVM 中（一个 SVM 就是一个命名空间）。

卷的扇出和扇入

对于 SnapMirror DP 关系，一个 NetApp FlexVol 卷最多可以复制到五个不同的目标卷。各个目标卷可以分别位于不同的 SVM 中，也可以全部位于同一个 SVM 中；这种方式称为卷的扇出。无法执行卷扇入（将多个不同的卷复制到同一目标卷）。

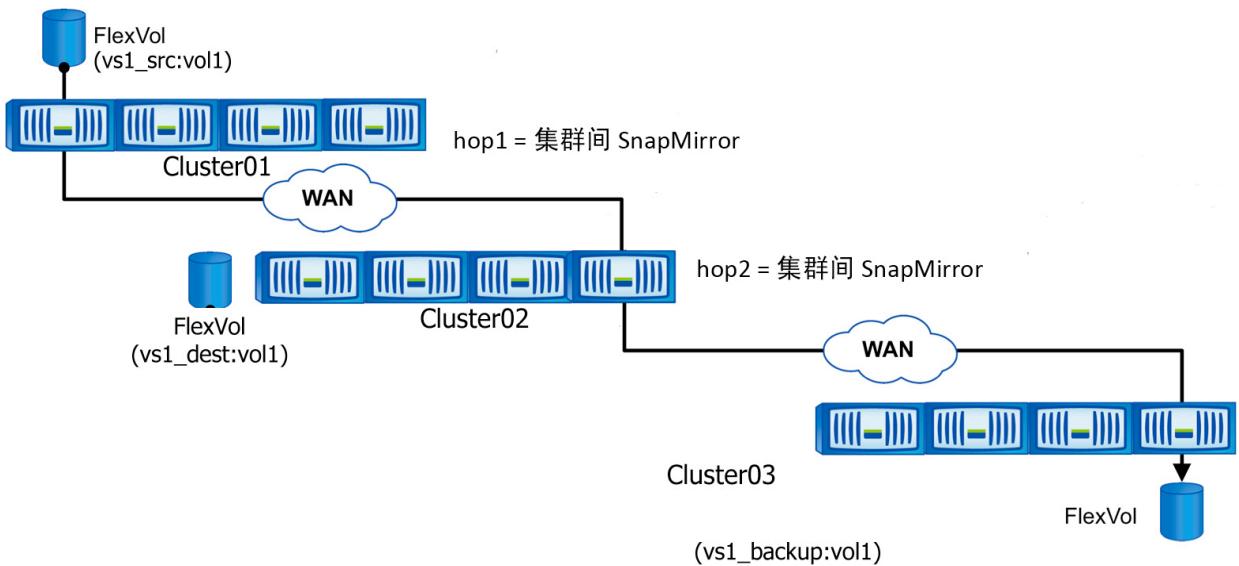
级联关系或多跳复制

从集群模式 Data ONTAP 8.2 操作系统开始，SnapMirror 关系可以采取级联形式。但是，在级联配置中，只能有一个关系是 SnapVault 关系。

级联是指从已生成的副本进行复制。假设有三个存储系统：A、B 和 C。从 A 复制到 B，再从 B 复制到 C 被视为级联配置。

图 23 中显示了一个包含两跳的级联配置示例。

图 23) 使用 SnapMirror 进行级联卷复制。



这种部署方案的功能是，使网络上不同位置的用户能够以只读方式访问一组统一的数据，并可定期统一更新该数据。

注： Snapshot 副本的行为：

1. SnapMirror 会对源卷的 Snapshot 副本创建一个软锁定 (snapmirror 标记)。

目标系统包含一个额外的 Snapshot 副本。

双跳卷 SnapMirror

此配置涉及在三个集群间进行卷 SnapMirror 复制。

vs1_src:vol1 → vs1_dest:vol1 → vs1_backup:vol1

注： 在以上配置中，vs1_src:vol1 到 vs1_dest:vol1 的传输和 vs1_dest:vol1 到 vs1_backup:vol1 的传输可同时进行。

表 1) 双跳卷 SnapMirror 的 Snapshot 副本传播。

时间表	cluster01 上的 Snapshot 副本	cluster02 上的 Snapshot 副本	cluster03 上的 Snapshot 副本
1) 在 cluster02 上进行卷初始化后	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1	
2) 在 cluster02 上进行卷 SnapMirror 更新	hourly.2013-02-26_1505 snapmirror.cd20c2a0v2	hourly.2013-02-26_1505 snapmirror.cd20c2a0v1 snapmirror.cd20c2a0v2	
3) 在 cluster03 上进行卷初始化后	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) 在 cluster02 上进行卷 SnapMirror 更新	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) 在 cluster03 上进行卷 SnapMirror 更新	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

需注意的 Snapshot 副本的使用情况：

- 在第一次 SnapMirror 更新（步骤 2）之后，cluster02（目标）上有一个额外的 Snapshot 副本。
- 由于 cluster02 与 cluster03 系统之间存在卷 SnapMirror 关系，因此步骤 3 之后，cluster03 也具有与 cluster02 相同数量的 Snapshot 副本。
- 步骤 3 之后，cluster02 上存在一个新的软锁定，因为 cluster02 现在是 cluster03 的卷 SnapMirror 的源集群。
- 步骤 4 之后，源集群 cluster01 包含两个 SnapMirror Snapshot 副本。这是因为，要为 cluster03 执行 SnapMirror 更新必须使用 Snapshot 副本 snapmirror.cd20c2a0v2，因此 cluster02 锁定了该副本。如果 cluster02 系统发生灾难，cluster01 系统上的此 Snapshot 副本也用于为 cluster03 系统执行 SnapMirror 重新同步。
- 在 cluster03 上执行更新（步骤 5）后，对最新的 SnapMirror Snapshot 副本 snapmirror.cd20c2a0v3 进行软锁定，因为该副本是 cluster02 和 cluster03 系统之间的新基线 SnapMirror Snapshot 副本。

为集群间 SnapMirror 关系传播数据

术语“传播数据”是指新创建的 SnapMirror 关系的初始数据传输。从集群模式 Data ONTAP 8.2 操作系统开始，您可以使用 SnapMirror 到磁带 (SMTape) 为 SnapMirror 和 SnapVault 关系传播数据。

使用 snapmirror create 命令创建新的 SnapMirror 关系时，不会自动执行初始传输。create 命令只能创建关系以及用于定义该关系的元数据。使用 snapmirror create 命令并随后使用 snapmirror initialize 命令可执行初始传输。或者，也可以在创建关系后立即单独使用 snapmirror initialize 命令执行初始传输。如果 SnapMirror 关系不存在，则 initialize 命令将创建关系并执行初始传输。

NetApp OnCommand System Manager 3.0 可以使用 SnapMirror 关系创建向导来对关系进行初始化。本文档稍后将介绍使用 System Manager 管理 SnapMirror 的信息。

5.2 计划 SnapMirror 更新

集群模式 Data ONTAP 操作系统内置了一个类似于 cron 的计划引擎。可以通过在目标集群中为 SnapMirror 关系分配计划在集群模式 Data ONTAP 操作系统中安排定期复制更新。您可以通过命令行使用 job schedule cron create 命令创建计划。下面的示例说明了创建名为 Hourly_SnapMirror 的计划的过程，该计划会在每小时开始时（即，每小时的第零分钟）运行。

```
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
```

然后，可以使用 -schedule 选项在创建 SnapMirror 关系时将该计划应用于该关系，或者使用 snapmirror modify 命令和 -schedule 选项将该计划应用于现有关系。在此示例中，Hourly_SnapMirror 计划会应用于一个现有关系。

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

您也可以使用 NetApp OnCommand System Manager 3.0 来管理计划并将计划应用于 SnapMirror 关系。

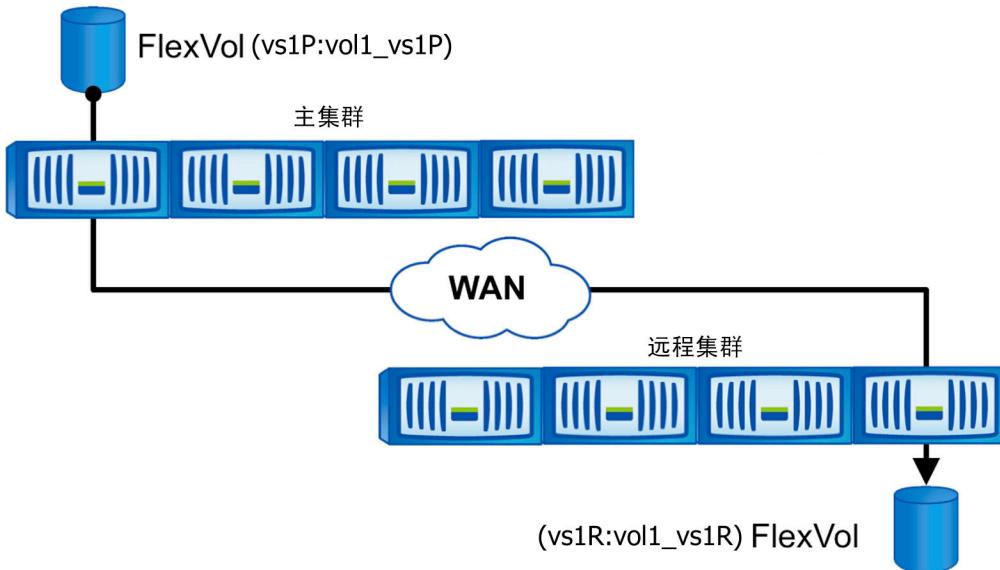
5.3 将 SnapMirror 关系转换为 SnapVault 关系

有时，您可能希望将现有的 SnapMirror 关系转换为 SnapVault 关系：一家现有客户正在集群模式 Data ONTAP 8.1 操作系统中使用 SnapMirror，希望在集群模式 Data ONTAP 8.2 操作系统中使用 SnapVault 来延长保留期限。

在这种情况下，可以将源集群和目标集群升级到集群模式 Data ONTAP 8.2 操作系统。现有的 SnapMirror 关系仍会保持在集群范围内，并且与在集群模式 Data ONTAP 8.1 操作系统中运行方式相同。提高可扩展性不会有所帮助，除非删除并重新创建这些关系。但是，集群模式 Data ONTAP 8.1 和集群模式 Data ONTAP 8.2 操作系统均使用块级引擎进行镜像，并且需要注意的是，无需进行基线传输，而只需要重新同步即可。

下面将举例说明详细过程，如图 24 中所示。目前已创建集群对等关系和 SVM 对等关系。

图 24) 将 SnapMirror 关系转换为 SnapVault 关系。



转换过程包括以下步骤：

1. 删除镜像 (DR) 关系。
2. 中断镜像目标。
3. 在相同端点间创建 XDP (vault) 关系。
4. 在端点间执行重新同步。这样，不必重新进行基线传输，即可将 DR 目标转换为存储目标。

在主集群上创建卷

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

在远程集群上创建 DP 卷

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

在主集群与远程集群上的卷之间创建 SnapMirror 关系

```
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
Source          Destination  Mirror  Relationship  Total          Progress
Path           Type     Path      State   Status      Progress  Healthy  Last
-----  -----
vs1P:vol1_vs1P  DP       vs1R:vol1_vs1R
                           Uninitialized
                           Idle
                           -           true
1 entries were displayed.
```

初始化 SnapMirror 关系

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

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

将 SnapMirror 转换为 SnapVault

SnapMirror Delete

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

SnapMirror Break

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

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

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

完成以上步骤后，要相应调整计划和策略，以便在存储目标上保留需要的 Snapshot 副本。另外，您不能将 SnapVault 目标卷变为读/写卷以用作 DR 卷。

6 使用 NetApp OnCommand System Manager 管理 SnapMirror 数据保护关系

NetApp OnCommand System Manager 3.0 可用于创建和管理 SnapMirror DP 关系。System Manager 中包含一个向导，可用于创建 SnapMirror DP 关系、创建要分配给关系的计划以及创建目标卷，所有操作都在同一个向导中进行。

但是，System Manager 3.0 不提供创建和管理 LS 镜像以及管理 SnapMirror 带宽限制设置的功能。

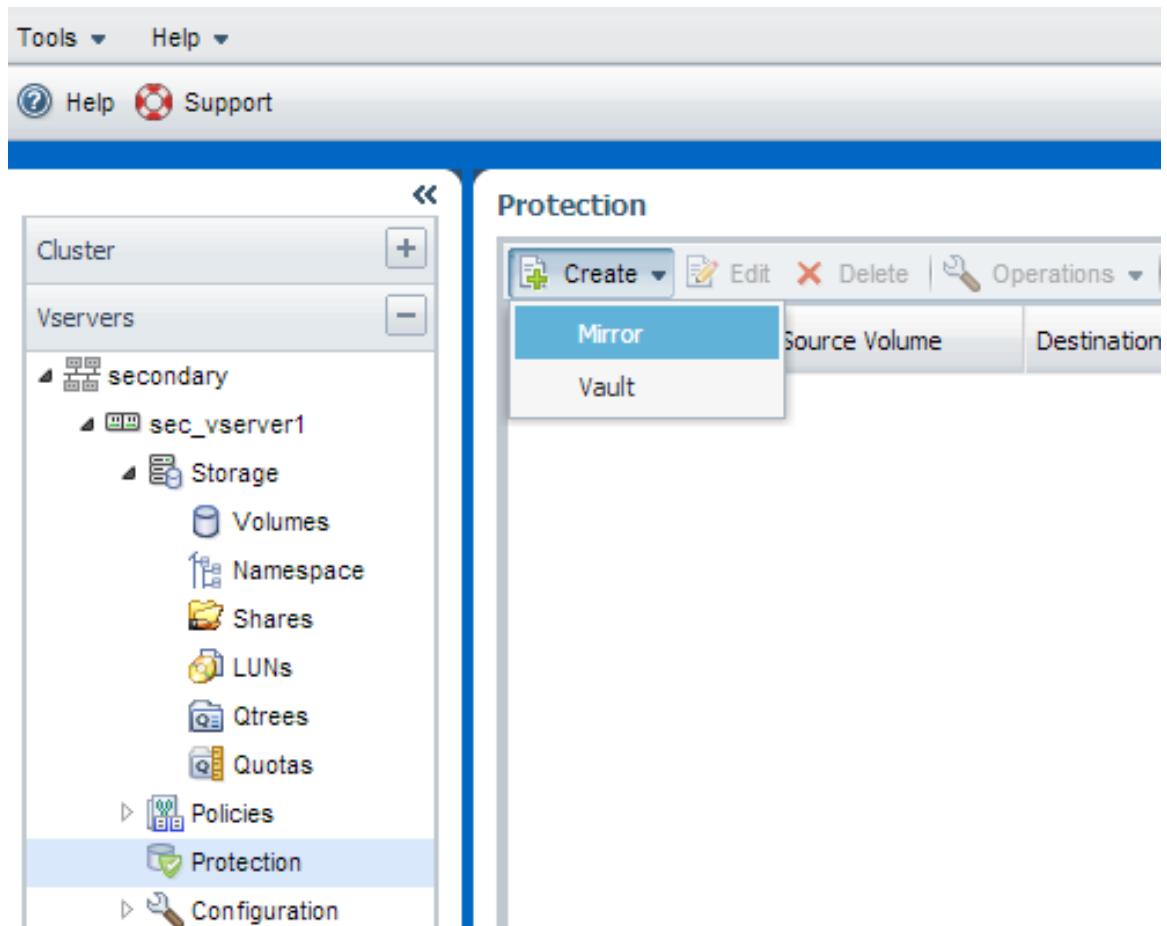
注：集群模式 Data ONTAP 操作系统中运行的 SnapMirror 关系必须由集群管理员管理，而不能委派给 SVM 管理员。从集群模式 Data ONTAP 8.2 操作系统开始，集群管理员可以将 SnapMirror 关系的管理工作委派给 SVM 管理员。

6.1 在 System Manager 中创建 SnapMirror 关系

本节介绍如何使用 System Manager 创建 SnapMirror 关系。在此示例中，我们创建了一个新的关系，用于将卷 vol1 从集群 Cluster01 中的 SVM vs1_src 镜像到集群 Cluster02 中的 SVM vs1_dest。

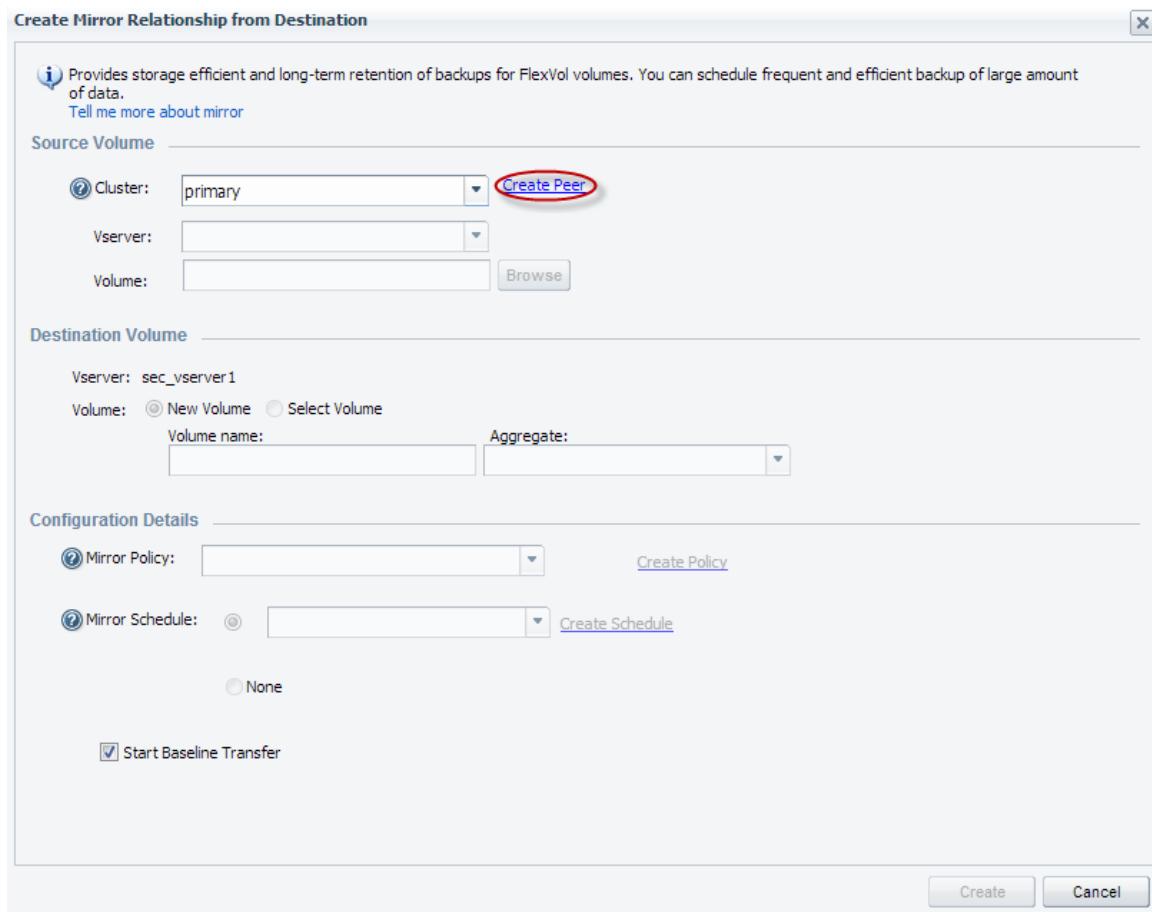
1. 在 System Manager 3.0 中，只能从目标集群创建新关系。在此示例中，我们选择的是名为 vs1_dest 的目标 SVM。依次单击“SVM”>“Protection”（保护）>“Create”（创建）。

图 25) 从目标集群创建 SnapMirror 关系: 选择镜像。



2. 使用 System Manager, 只能从目标集群创建关系; 因此, 请确定源集群 cluster01。

图 26) 从目标集群创建 SnapMirror 关系: 选择源集群。



3. 接下来, 选择源集群。如果已在 System Manager 中存储了源集群的凭据, 或者 System Manager 已登录到该集群, 则会自动输入集群凭据。否则, 请输入源集群的凭据。如果未创建集群对等关系, 请为集群创建该对等关系。

图 27) 从目标集群创建 SnapMirror 关系: 创建集群对等关系。

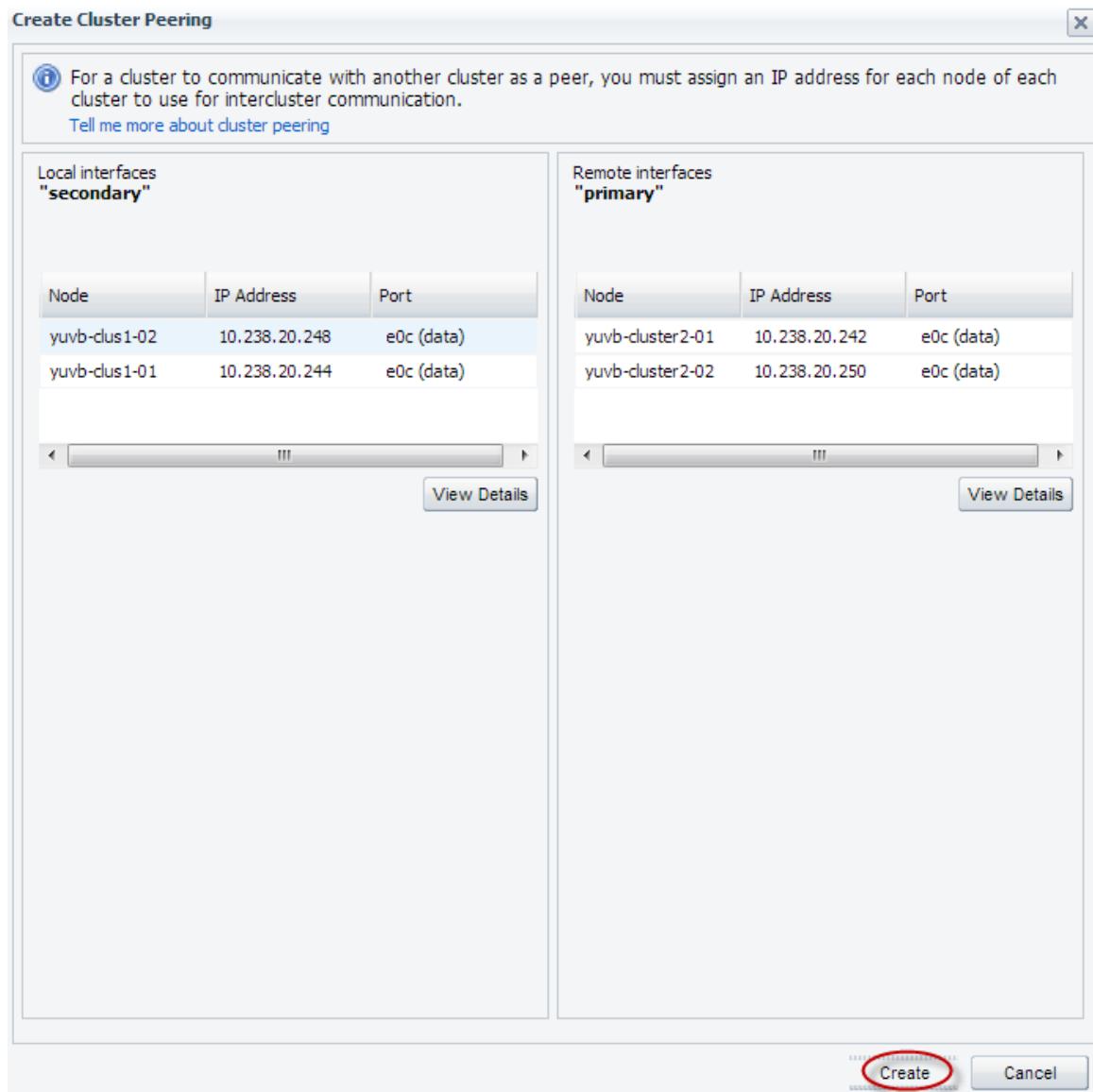
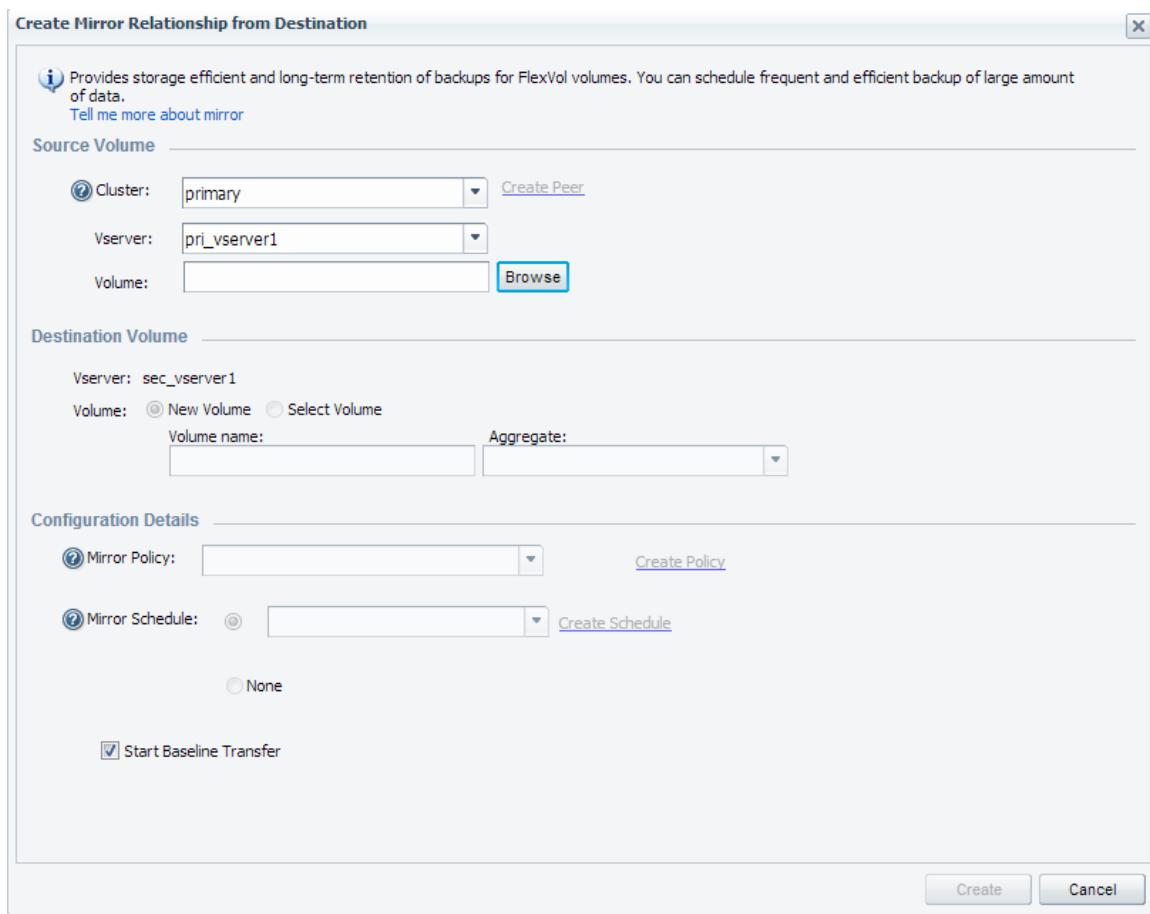
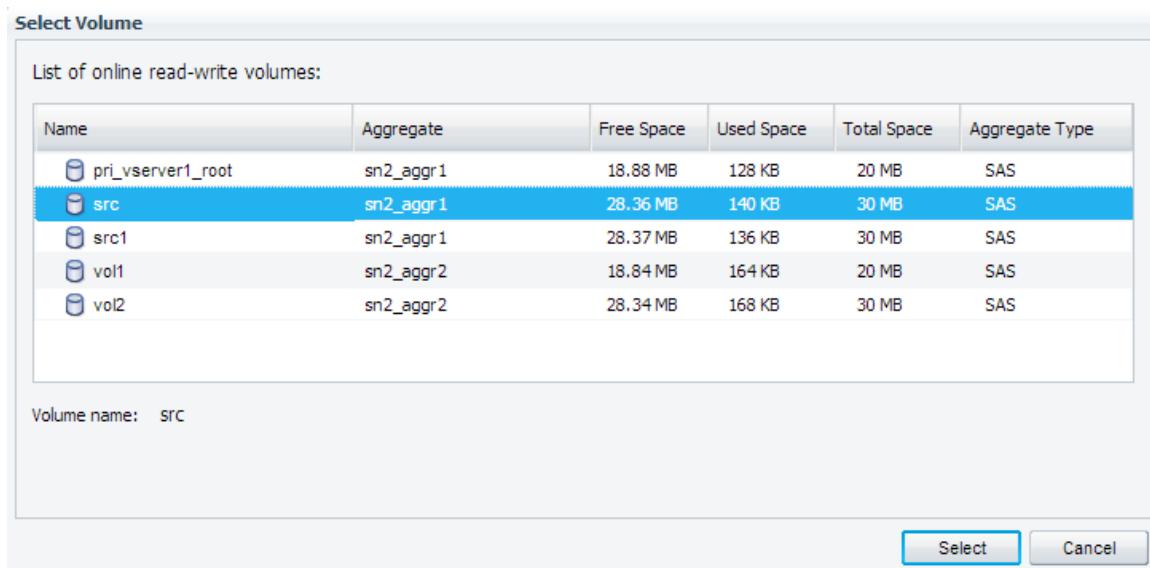


图 28) 从目标集群创建 SnapMirror 关系: 选择源 SVM。



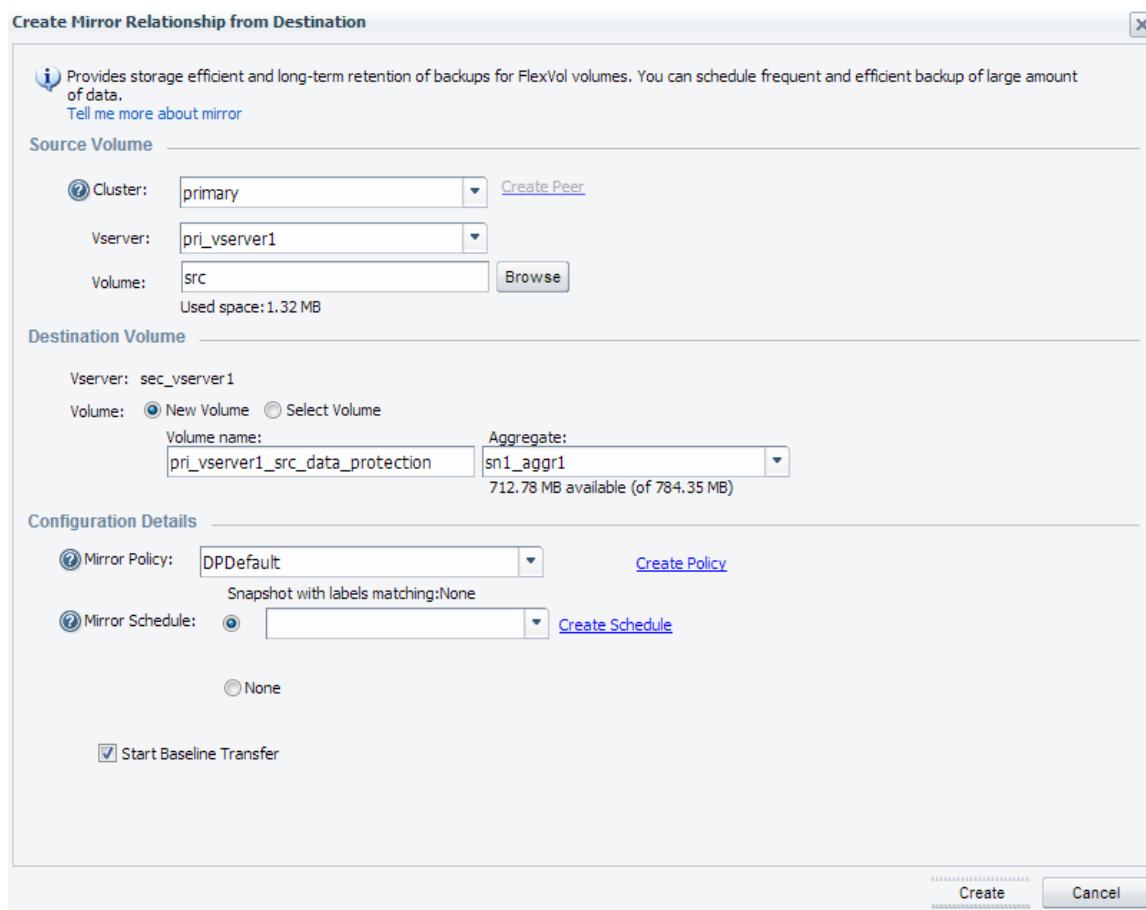
4. 选择源 SVM。如果源 SVM 未与目标 SVM 建立对等关系，则 System Manager 3.0 将为这两个 SVM 建立对等关系。

图 29) 从目标集群创建 SnapMirror 关系: 选择源卷。



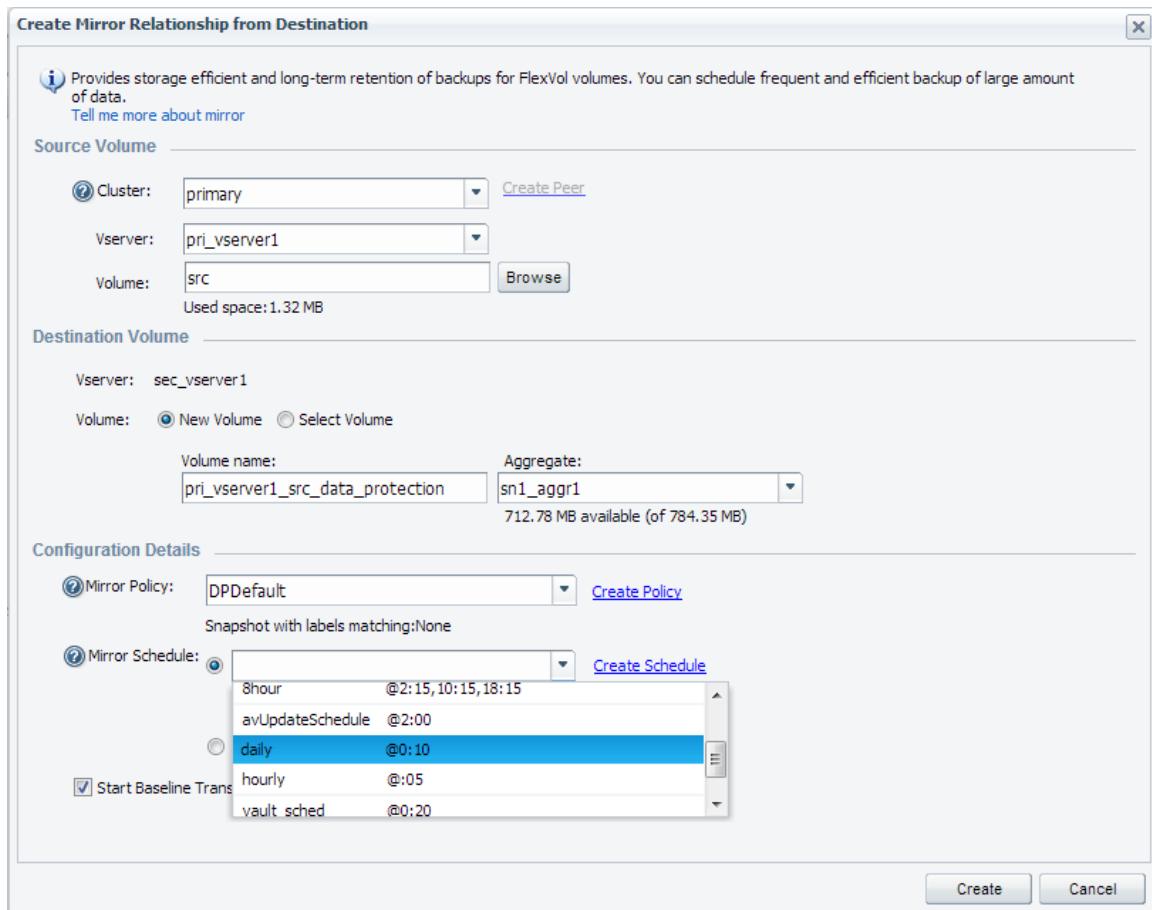
5. 选择源 SVM 上的现有卷。

图 30) 从目标集群创建 SnapMirror 关系: 选择目标卷。



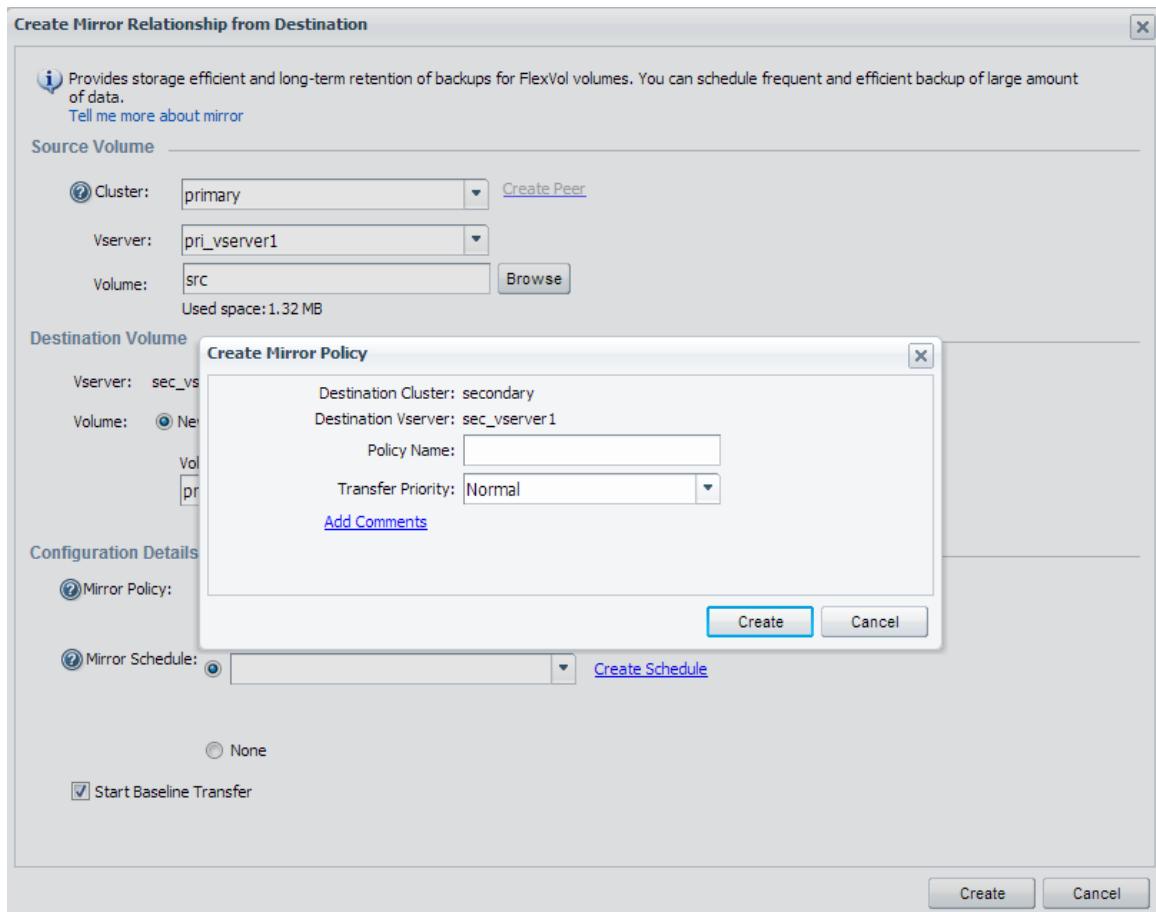
6. 选择目标 SVM 上的目标卷, 或者在目标 SVM 上创建目标卷。

图 31) 从目标集群创建 SnapMirror 关系: 选择或创建 SnapMirror 策略和计划。



7. 在无需退出向导的情况下选择一个现有 SnapMirror 策略或创建一个新策略（默认策略为 DPDefault）。选择一个现有 SnapMirror 计划或创建一个新计划。

图 32) 从目标集群创建 SnapMirror 关系: 创建新的 SnapMirror 策略。

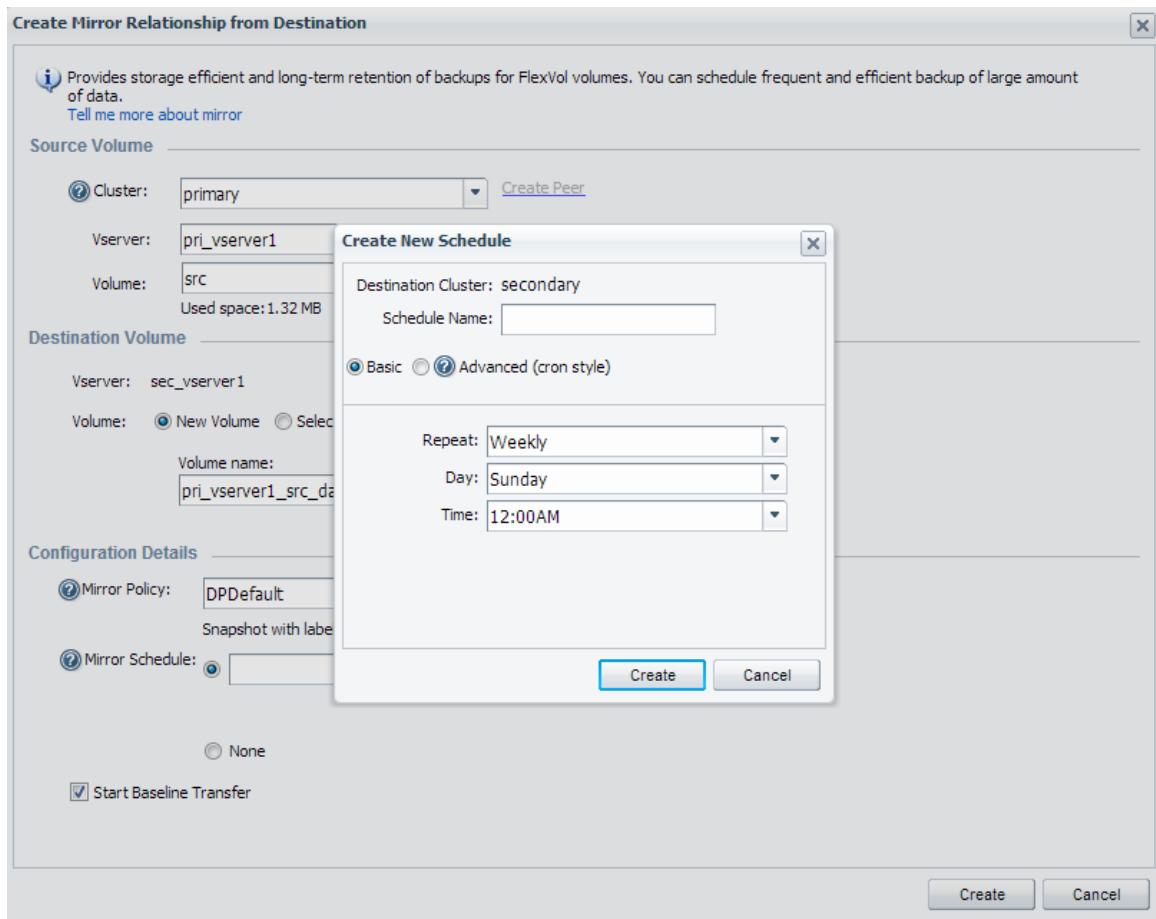


8. 创建新的 SnapMirror 策略。传输优先级有两个级别: “Normal” (普通) 和 “Low” (低) (默认为 “Normal”)。

您可随时更改传输优先级; 下次操作时生效。节点上的待定操作数足以达到计量值* 时, 该设置即会生效。即使计量值为空, 低优先级操作也会延迟数分钟。它们的延迟时间为一分钟。

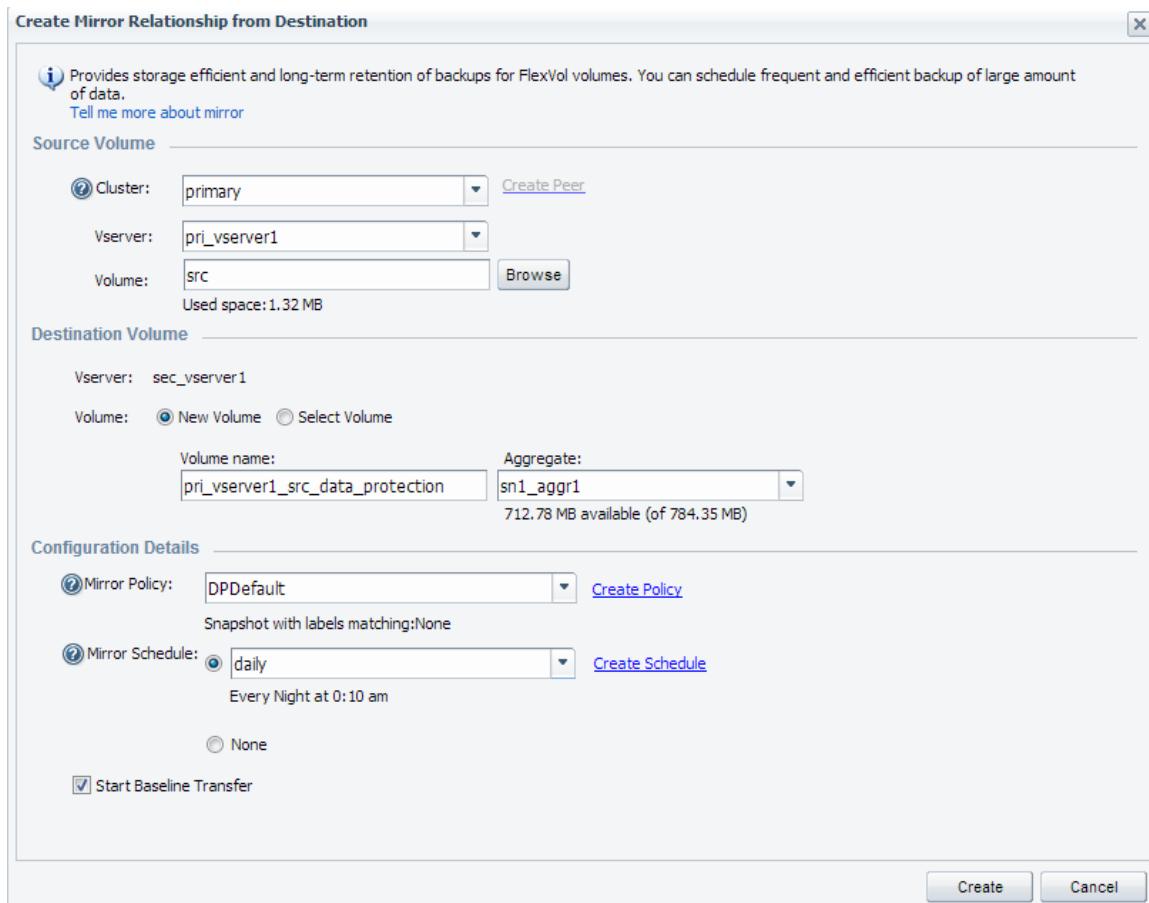
计量*值取决于平台和内存。对于内存超过 8 GB 的高端平台, 该值为 100; 对于内存等于或小于 8 GB 的中低端平台, 该值为 20。可以为 SnapMirror 预留部分计量值, 并为 SnapVault 预留部分计量值。默认情况下不进行预留。

图 33) 从目标集群创建 SnapMirror 关系: 创建新的 SnapMirror 计划。



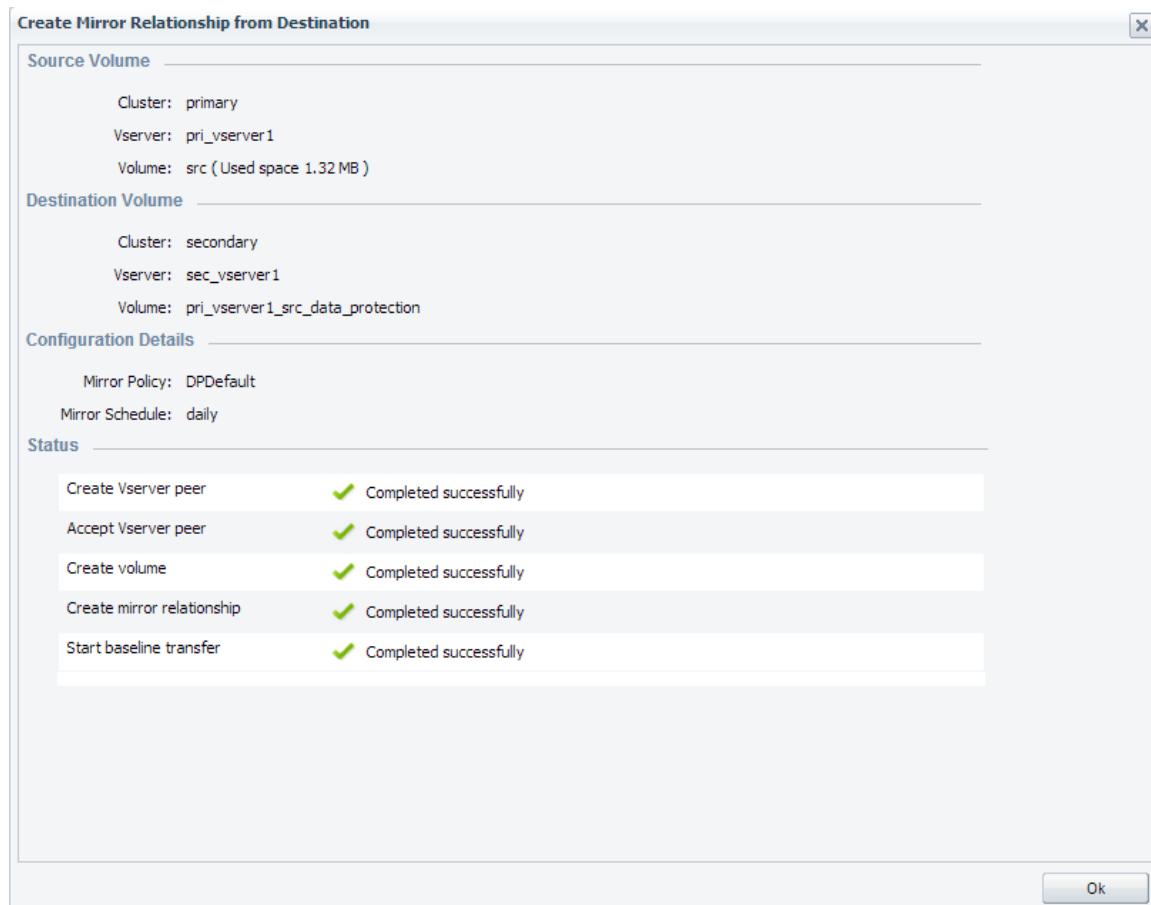
9. 创建新的 SnapMirror 计划。

图 34) 从目标集群创建 SnapMirror 关系: 启动基线传输。



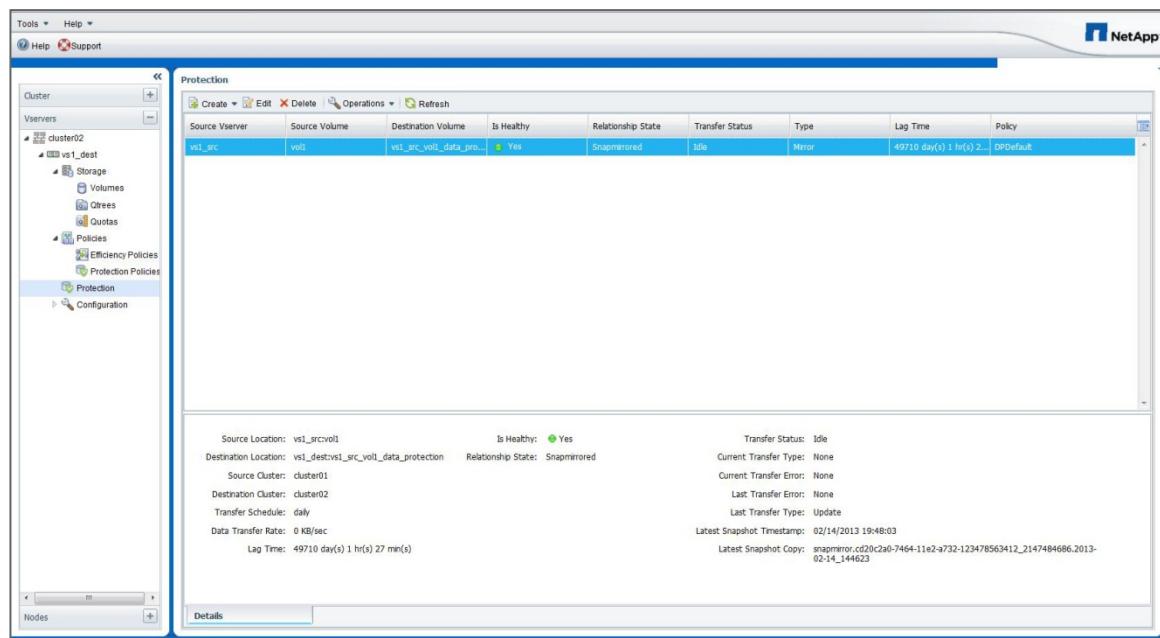
10. 要在创建关系后自动开始 SnapMirror 初始化（初始基线复制），请选中“Start Baseline Transfer”（启动基线传输）复选框。

图 35) 从目标集群创建 SnapMirror 关系: SnapMirror 关系配置与状态摘要。



11. 该向导将显示 SnapMirror 关系配置与状态的摘要。

图 36) SnapMirror 基线传输的详细信息。

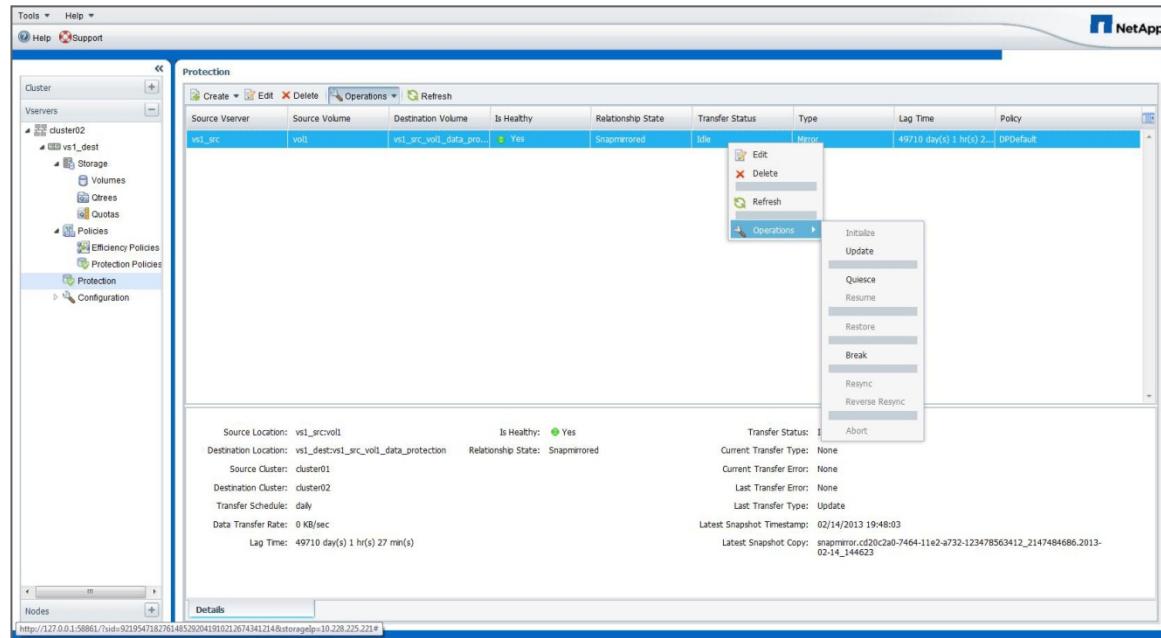


完成 SnapMirror 关系创建向导后, 将打开 SnapMirror 窗口。完成初始基线传输后, 如果需要, 可单击“Refresh”(刷新) 查看完成状态。

6.2 使用 System Manager 管理 SnapMirror 关系

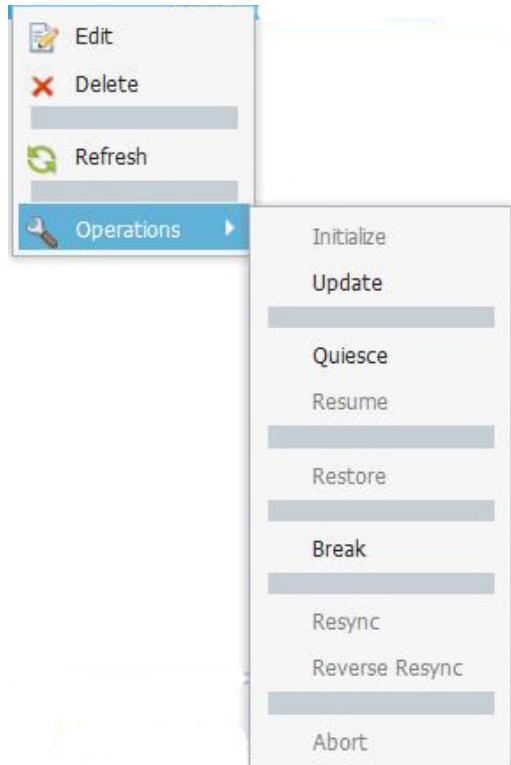
要在 System Manager 中管理 SnapMirror DP 关系, 请按图 37 中所示单击 SnapMirror 屏幕顶部的“Operations”(操作)菜单, 或者右键单击某个特定 SnapMirror 关系, 然后打开上下文菜单。在该上下文菜单中, 只会启用当前允许对该 SnapMirror 关系执行的操作。

图 37) SnapMirror 关系列表。



上下文菜单还提供其他多个选项。灰色的选项不可用, 具体取决于所选 SnapMirror 关系的当前状态。图 38 显示了 System Manager 上下文菜单中可以执行的所有可用操作。

图 38) Systems Manager 上下文菜单。



SnapMirror 上下文菜单中列出的操作可执行以下功能：

- **Edit (编辑)**。编辑关系的计划。
- **Delete (删除)**。删除 SnapMirror 关系。此功能并不删除目标卷。
- **Initialize (初始化)**。执行第一次初始基线数据传输以建立新关系。
- **Update (更新)**。为关系执行按需更新，将上次更新以来的所有新数据和 Snapshot 副本复制到目标位置。
- **Quiesce (暂停)**。阻止对关系进行任何进一步的更新。
- **Resume (恢复)**。恢复之前暂停的关系。
- **Restore (还原)**。将 Snapshot 副本从源卷还原至目标卷。
- **Break (中断)**。使目标卷可读/写。
- **Resync (重新同步)**。按照发生 SnapMirror 中断前的相同方向重新建立已中断的关系。

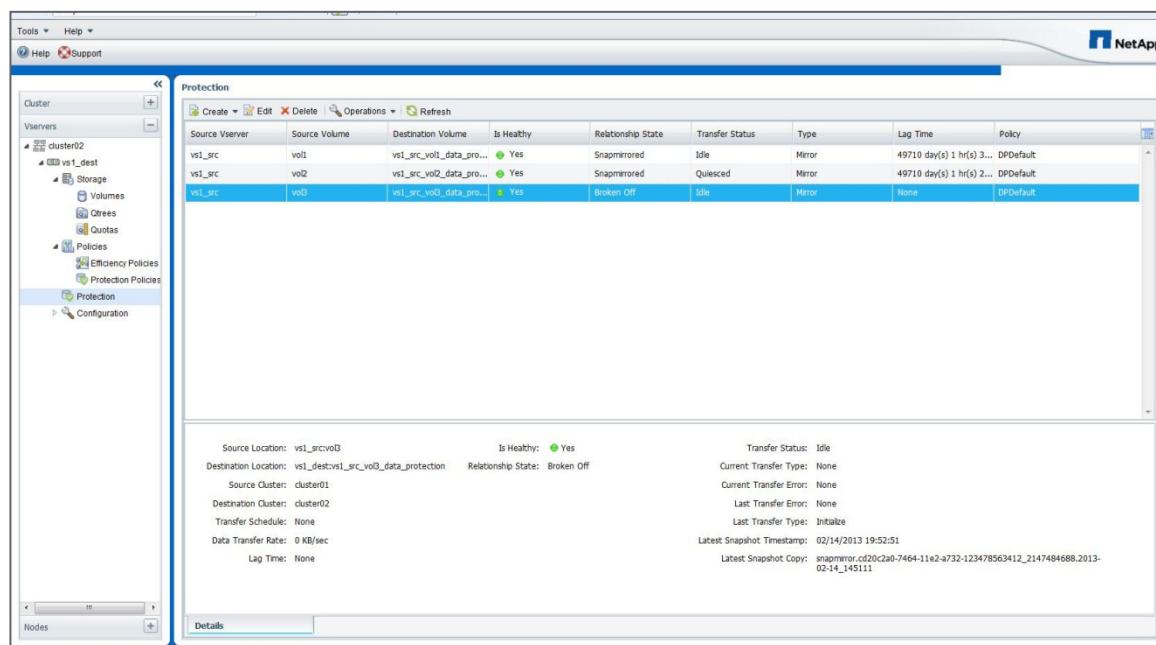
注：如果您在中断并删除 SnapMirror 关系后重新创建了该关系，则可以执行 SnapMirror 重新同步操作来重新同步各个卷，而无需重新执行基线传输。此任务要求两个卷上具有一个通用 Snapshot 副本。

- **Reverse Resync (反向重新同步)**。自动执行必要的步骤以反转 SnapMirror 关系，从而重新创建该关系并以相反方向重新同步。只有当现有关系处于中断状态时，才能执行此操作。确定客户端未在使用初始源卷，因为反转/重新同步操作会将初始源卷设置为只读。初始源卷会恢复为最新的通用 Snapshot 副本，然后与目标进行重新同步。自上次成功更新 SnapMirror 以来对初始源卷所做的任何更改都将丢失。对当前目标卷所做的任何更改或写入当前目标卷的新数据，将发送回初始源卷。选择此选项时，System Manager 会显示确认屏幕，说明正在执行的操作。
- **Abort (中止)**。取消当前正在进行的传输。如果为中止的关系发出 SnapMirror 更新命令，关系将从中止前创建的重新启动检查点继续进行上次传输。

System Manager 中的关系运行状况

SnapMirror 关系主要在目标系统中进行管理；但是，可以使用 System Manager 查看有关 SnapMirror 关系的最新信息。首次加载 SnapMirror 状态屏幕时，只会显示选定 SVM 上目标卷关系的权威信息，如图 39 中所示。

图 39) SnapMirror 状态屏幕。



如果关系中只有源卷位于选定 SVM 上，则该关系的最初运行状况会显示为未知。单击该关系并从目标卷刷新状态，即可在“Health”（运行状况）列中显示正确的状态。单击该关系后，System Manager 将访问目标卷，收集有关该关系的权威信息，然后显示相应状态。另外，单击该关系后，还会更新窗口底部的详细信息窗格，并显示有关该关系的更多信息，如上次复制的时间戳。

7 SnapMirror 负载共享镜像关系

SnapMirror LS 镜像可以将 SVM 命名空间根卷分发到同一集群中的其他节点，并将数据卷分发到该集群中的其他节点，以提高大型只读工作负载的性能，从而增强 NAS 客户端的性能和可用性。

注： SnapMirror LS 镜像仅能够支持 NAS (CIFS/NFSv3)。LS 镜像不支持 NFSv4 客户端或 SAN 客户端协议连接 (FC、FCoE 或 iSCSI)。

7.1 管理负载共享镜像

LS 镜像关系只能通过 Data ONTAP 进行管理。目前，无法使用 System Manager 管理 LS 镜像关系。

LS 镜像关系与 DP 关系的一个不同之处在于，提供了额外的命令来管理 LS 镜像的 snapmirror initialize-ls-set、update-ls-set 和 promote 命令。从同一源卷复制数据的一组 LS 镜像目标卷被称为 LS 镜像集。

创建 LS 镜像集时，必须在适当的聚合中创建各个目标卷，并且创建的目标卷为 DP 类型。在此示例中，我们为名为 vs1 的 SVM 根卷创建了两个分别名为 vs1_ls_a 和 vs1_ls_b 的卷作为 LS 镜像目标卷。

```
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
```

创建了所有 LS 镜像目标卷之后，可以创建各个 LS 类型的 SnapMirror 关系。在此示例中，为目标卷 vs1_ls_a 和 vs1_ls_b 各创建了一个 LS SnapMirror 关系，并设置了一个每小时更新计划。

```
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 镜像关系可以手动进行更新，也可以通过在 -schedule 选项中设置需要的计划来进行更新。对于 LS 镜像关系，这是通过在 LS 镜像集中的任意一个目标卷上设置需要的计划来实现的。Data ONTAP 会自动向该 LS 镜像集中的所有目标卷应用该计划。稍后对 LS 镜像集中任何目标卷的更新计划所做的更改将应用于该 LS 镜像集中所有卷的新计划。因此，在上述示例中，我们仅在创建最后一个关系时使用了 -schedule 选项，该选项会将该计划同时应用于这两个关系。

然后，可以使用 snapmirror initialize-ls-set 命令通过一次性操作为特定 LS 镜像集初始化所有目标卷，如以下示例所示。使用此命令时，请指定源路径（而不是目标路径）来标识 LS 镜像集，因为在 LS 镜像集中，源路径是要初始化的所有关系的公共路径。

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

```
cluster01::> snapmirror show

Source          Destination   Mirror  Relationship  Total          Progress
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   -
```

您可以根据需要使用 `snapmirror update-ls-set` 命令更新 LS 镜像关系，如以下示例所示。使用此命令时，请指定源路径（而不是目标路径）来标识 LS 镜像集，因为在 LS 镜像集中，源路径是要更新的所有关系的公共路径。Data ONTAP 将通过一次操作更新 LS 镜像集的所有目标卷。

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

7.2 访问负载共享镜像卷

默认情况下，将为访问 LS 镜像集中的卷的所有客户端请求授予只读访问权限。读写访问权限通过访问特殊的管理挂载点来授予，这是需要对 LS 镜像集进行读写访问的服务器必须挂载的路径。所有其他客户端将拥有只读访问权限。对源卷进行更改后，必须使用 `snapmirror update-ls-set` 命令或通过计划的更新将更改复制到 LS 镜像集中的其余卷。

卷可以挂载在其他卷内部，也称为嵌套卷。将一个新卷挂载到 LS 镜像集中配置的某个卷的内部后，客户端将无法看到新的挂载点，直到 LS 镜像集完成更新。您可以根据需要使用 `snapmirror update-ls-set` 命令执行此操作，也可以在开始进行 LS 镜像的下一次计划更新时执行此操作。

可向不同的 CIFS 和 NFS 客户端授予对卷的读写访问权限。

要使 CIFS 客户端能够在获得读写访问权限的情况下连接到源卷，请通过向卷路径中添加 `/.admin`，为管理挂载点创建一个 CIFS 共享。在下面的示例中，创建了一个名为 `report_data_rw` 的 CIFS 共享，通过它可以对一个 LS 镜像集中名为 `report_data` 的卷进行读写访问。

通过以下路径，可以使用 CIFS 访问 LS 镜像集的读写管理共享。

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

所有需要读写访问权限的 CIFS 客户端都必须连接到以下路径。

```
\vs1\report_data_rw
```

要从 NFS 客户端以读写访问权限连接到 LS 镜像集的源卷，请挂载 NFS 导出路径，并向卷路径中添加 `/.admin`，如以下示例所示。

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

`nfs_client` 系统上运行的任何进程或应用程序都必须使用 `/client_rw_mountpoint` 路径进行读写访问。

7.3 SVM 命名空间根卷的负载共享镜像

命名空间根卷非常小，其中只包含用作挂载点的目录，即数据卷接合（挂载）到命名空间时的路径。但是，对于 NAS 客户端来说，它们非常重要，因为如果 SVM 根卷不可用，NAS 客户端将无法访问数据。

注： SAN 客户端连接（FC、FCoE 或 iSCSI）不需要使用 SVM 根卷。

可以将新卷挂载到 LS 镜像集中配置的命名空间根卷。挂载卷之后，客户端将无法看到新的挂载点，直到 LS 镜像集完成更新。您可以根据需要使用 `snapmirror update-ls-set` 命令执行此操作，也可以在开始进行 LS 镜像的下一次计划更新时执行此操作。

如前所述，LS 镜像卷是只读卷。当 LS 镜像卷是命名空间根卷时，该卷是只读卷；但是，命名空间中挂载的数据卷可以是读写卷，也可以是只读卷，具体取决于个别卷的特征以及这些卷内文件和文件夹上设置的权限。

最佳实践

在集群中的每个节点上为 NAS SVM 命名空间根卷创建一个 LS 镜像，这样，无论节点是否发生中断或故障转移，命名空间的根卷均可用。

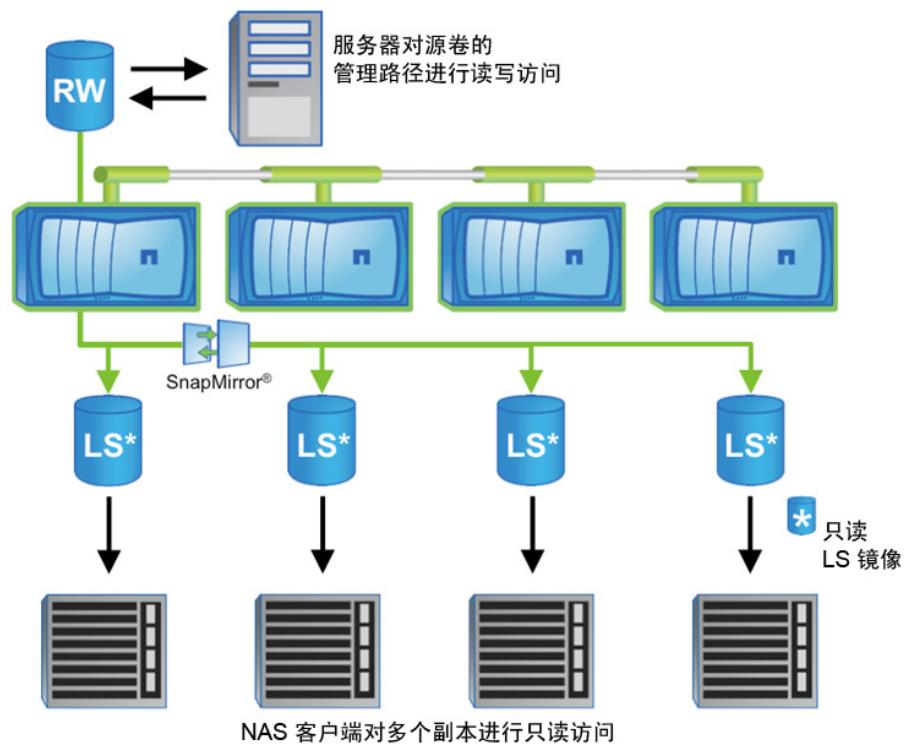
在客户端请求访问已配置 LS 镜像集的卷时，Data ONTAP 会将所有客户端连接仅定向到 LS 镜像目标卷；因此，应在源卷所在的同一节点上创建目标卷，以使命名空间可以为该节点上的数据卷提供直接数据访问路径。

7.4 只读工作负载的负载共享镜像

LS 镜像也可以用于向集群中的其他节点分布数据卷，以提高只读工作负载的性能。如果只有一个或几个客户端可以对数据集进行读写，而大多数客户端只能进行读取，则可以使用该数据卷的 LS 镜像。例如，如果使用少数几个服务器生成大量测试输出数据并将数据写入源卷，并设置许多对测试数据拥有只读访问权限的服务器，将能够以更高性能处理数据并生成输出报告，这是因为只读工作负载已在集群范围内进行分布。

在图 40 所示的配置中，所有卷显示为一个卷，并对所有客户端显示为只读。与命名空间根卷的 LS 镜像可以在集群范围内分布连接一样，可以在每个节点上提供只读 LS 镜像卷，以便所有客户端都可以通过直接数据访问路径连接到只读卷。

图 40) 只读工作负载的 LS 镜像。



7.5 负载共享镜像关系的故障转移

SnapMirror LS 镜像关系的故障转移在执行方面与 DP 关系的故障转移有所不同。在 Data ONTAP 8.1 中，不允许对 LS 镜像目标卷使用 `snapmirror break` 命令。LS 镜像关系的故障转移使用 `snapmirror promote` 命令。此命令会将故障转移目标卷提升为整个 LS 镜像集的源卷，同时会删除初始源卷。在以下示例中，SVM 根卷 `vs1` 会镜像到卷 `vs1_a` 和 `vs1_b`，而卷 `vs1_b` 将被提升以替代 LS 镜像集中的源卷。

```
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
```

现在, `promote` 命令中的目标卷会成为 LS 镜像集的源卷, 并且所有 NFS 文件句柄和 CIFS 连接都不会中断。请注意, LS 镜像卷是只读的, 所有客户端访问请求都只会定向到目标卷, 因此, 无法以只读方式连接到该提升操作所删除的源卷, 而只能通过连接访问读写管理共享。NFS 文件句柄以及与读写管理共享的 CIFS 连接将无中断传输到新的源卷。与已提升的卷的只读连接将无中断传输到 LS 镜像集中的其他目标卷。

提升操作还会删除初始源卷, 并且所提升目标卷的特定 SnapMirror 关系也会被删除。

```
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
```

由于提升的卷可能具有与初始源卷不同的卷名, 因此, 可以重命名新的源卷, 以保持 LS 镜像集的源卷名称, 如以下示例所示。

```
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
```

最佳实践

提升操作会删除初始源卷, 因此, 可能需要在当前源卷所在的节点上创建另一个 LS 镜像目标卷。

8 版本灵活复制

从集群模式 Data ONTAP 8.3 操作系统开始，新增了一个版本灵活复制功能，通过该功能，目标控制器的集群模式 Data ONTAP 操作系统主要版本号无需再等于或高于源控制器的主要版本号，从而可以使客户实现无中断复制。

8.1 使用案例

版本灵活复制功能支持双向 SnapMirror 拓扑实现无中断升级，并减少了目标位置所需的二级 Snapshot 副本数量。

8.2 默认策略

从集群模式 Data ONTAP 8.3 操作系统开始，为版本灵活复制功能新增了三个 SnapMirror 策略。它们是：

- **MirrorLatest**。此选项与 7-模式中的 qtree SnapMirror 类似。使用该选项会为活动文件系统创建 Snapshot 副本，并将其从源位置传输至目标位置。
- **MirrorAllSnapshots**。此选项与默认 SnapMirror 类似。使用该选项会将创建的所有源 Snapshot 副本（包括活动文件系统的 SnapMirror Snapshot 副本）从源位置传输至目标位置。
- **MirrorAndVault**。此选项可以在一个卷中实现灾难恢复和备份。使用该选项会按照与存储策略类似的方式为保留设置 SnapMirror 标签及其相应的保留计数。除了传输 Snapshot 副本的匹配标签之外，还会创建和传输活动文件系统的 SnapMirror Snapshot 副本。

表 2) 版本灵活复制策略。

	存储效率模式 (与 QSM 类似)	默认 SnapMirror 兼容模式	灾难恢复与备份
许可证	SnapMirror	SnapMirror	SnapMirror 和 SnapVault
关系类型	XDP	XDP	XDP
SnapMirror 策略 名称	MirrorLatest	MirrorAllSnapshots	MirrorAndVault
SnapMirror 策略 类型	async-mirror	async-mirror	mirror-vault
目标上保留的 Snapshot 副本	SnapMirror Snapshot 副本	相当于源加上 SnapMirror Snapshot 副本	基于保留计数以及 SnapMirror Snapshot 副本

8.3 配置版本灵活复制

以下示例介绍了如何从 CLI 为版本灵活复制功能配置 MirrorAllSnapshots 策略：

```
cluster02::> snapmirror create -source-path svmA:srcvolA -destination-path  
svmB:dstvolB -type XDP -policy MirrorAllSnapshots
```

8.4 将默认 SnapMirror 转换为版本灵活复制

请考虑以下情形：

一家现有客户正在集群模式 Data ONTAP 8.2 操作系统中使用 SnapMirror，希望在集群模式 Data ONTAP 8.3 操作系统中使用版本灵活复制功能，以便通过一个目标卷实现灾难恢复和备份。

此时，可以将源集群和目标集群升级至集群模式 Data ONTAP 8.3 操作系统。目前已创建集群对等关系和 SVM 对等关系。

下面介绍了转换步骤：

1. 删除镜像 (DR) 关系。
2. 中断镜像目标。
3. 使用一个默认版本灵活复制策略，在相同端点间创建 XDP 关系。
4. 在端点间执行重新同步。这样，无需重新进行基线传输，即可转换为版本灵活复制配置。

在主集群上创建卷

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

在远程集群上创建 DP 卷

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

在主集群与远程集群上的卷之间创建 SnapMirror 关系

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

初始化 SnapMirror 关系

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

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

将 SnapMirror 转换为版本灵活复制

SnapMirror Delete

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

SnapMirror Break

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

SnapVault Create

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

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

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

完成以上步骤后，可以相应地调整计划和策略，以便在目标上保留所需的 Snapshot 副本。

注：版本灵活关系无法转换回默认 SnapMirror 关系。如果需要默认 SnapMirror 关系，可以为新的目标设定基线。

9 SnapMirror 和 Data ONTAP 的功能交互

9.1 SnapMirror 和 Snapshot 副本

SnapMirror 在执行复制更新前会创建一个 Snapshot 副本。在源卷上创建一个 SnapMirror Snapshot 副本后，该 Snapshot 副本会与已复制的前一个 SnapMirror Snapshot 副本进行比较。新 SnapMirror Snapshot 副本与前一个副本之间的所有数据（包括卷上这两个副本之间的所有 Snapshot 副本和这些 Snapshot 副本中的所有数据）都将复制到目标卷。完成 SnapMirror 更新后，新的 SnapMirror Snapshot 副本将导出到目标系统。

SnapMirror 将在源卷上保留一个 SnapMirror Snapshot 副本的历史记录，而在目标卷上保留两个。

最佳实践

确认计划的 SnapMirror 更新与其他 Snapshot 副本操作不会同时在源卷上进行。

Data ONTAP 会锁定 SnapMirror 创建的 Snapshot 副本，以防止这些副本被删除；执行计划的更新需要使用这些 Snapshot 副本。如果必须删除 SnapMirror 创建的 Snapshot 副本，则只要两个卷之间的其他通用 Snapshot 副本仍然存在，就可以对这些卷进行重新同步，而无需执行完整基线传输。在此示例中，卷上由 SnapMirror 创建的所有 Snapshot 副本均被删除，此时对该卷执行 SnapMirror 重新同步。

注：系统指定了一个每小时 Snapshot 副本的名称，用作重新同步的基础。

```
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.
```

9.2 SnapMirror 和 qtree

qtree 是一类特殊的目录，通过它可以为 NAS 应用文件系统配额。集群模式 Data ONTAP 操作系统允许创建 qtree，并且 qtree 可以位于使用 SnapMirror 复制的卷中。但是，SnapMirror 不允许复制单个 qtree 或进行 qtree 级别的复制。所有 SnapMirror 复制只在卷级别进行。

9.3 SnapMirror 和 FlexClone

NetApp FlexClone 卷是 FlexVol 卷的一个可写时间点克隆。FlexClone 卷与父卷共享数据块，仅存储新数据或对克隆所做的更改。也可以将 FlexClone 卷与其父卷分离，从而创建独立的新卷。

您可以使用 FlexClone 卷作为源卷来创建 SnapMirror 关系；但是，不可以将 FlexClone 卷作为 SnapMirror 目标卷。从集群模式 Data ONTAP 8.3 操作系统开始，可以使用 FlexClone 卷作为源卷和（或）目标卷来创建 SnapMirror 关系。

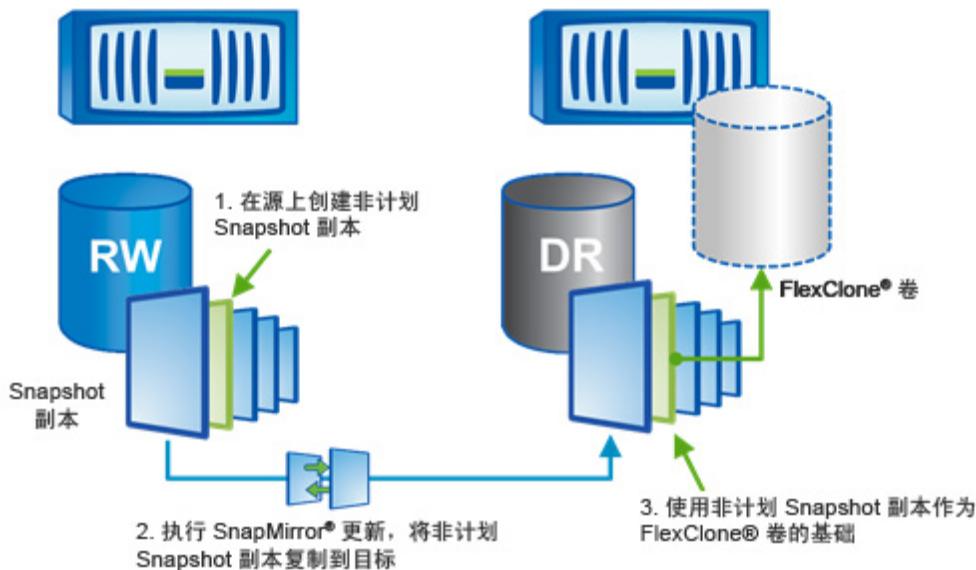
借助 FlexClone 技术，还可以在不中断 SnapMirror 复制过程或生产操作的情况下从只读 SnapMirror 目标创建可写卷。图 41 说明了在 SnapMirror 目标上创建 FlexClone 卷的过程。

最佳实践

SnapMirror 可将 Snapshot 副本的历史记录从源卷复制到目标卷。如果从源卷中删除了某个 Snapshot 副本，则下次 SnapMirror 更新时，也会从目标卷中删除该 Snapshot 副本。如果无法从目标卷中删除该 Snapshot 副本，例如，如果该 Snapshot 副本因作为 FlexClone 卷的基本 Snapshot 副本而被锁定，则 SnapMirror 更新将失败。要使 SnapMirror 更新继续进行，唯一的方法是删除 FlexClone 卷，或将其拆分并删除 Snapshot 副本的依赖关系。

在 SnapMirror 目标上创建 FlexClone 卷时，为了避免此问题，请在源系统上创建 FlexClone 卷所需的基本 Snapshot 副本，然后将该 Snapshot 副本复制到目标系统并使用该 Snapshot 副本作为 FlexClone 卷的基础，如图 41 中所示。以这种方式使用专为 FlexClone 卷创建的 Snapshot 副本，可防止 SnapMirror 更新因为自动创建的 Snapshot 副本从源系统中删除而失败。

图 41) 在 SnapMirror 目标上创建 FlexClone 卷。



9.4 SnapMirror 和无限卷

SnapMirror 可以像用于 FlexVol 卷一样用于无限卷。

但是有一些主要区别：

- 对于命名空间 (NS) 成分卷，仅限在集群内进行镜像。从集群模式 Data ONTAP 8.2 操作系统开始，创建无限卷时会自动创建 NS 镜像。如果使用 SnapDiff，则会自动为每个节点创建一个 NS 镜像，如果不使用 SnapDiff，则只会在无限卷上创建一个 NS 镜像。
- 仅支持使用集群间 SnapMirror 来镜像整个无限卷。

创建无限卷的过程与创建 FlexVol 卷有所不同。除此之外，SnapMirror 关系设置与 FlexVol 卷是相同的。本节将完整介绍无限卷的 SnapMirror 生命周期操作。

创建无限卷

在源上创建 SVM 和卷。确保将 SVM 的 `-is-repository` 设置为 `true`，以指定由该 SVM 存放无限卷。

```
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
```

在目标上执行相同操作。这些 SVM 只能存放一个无限卷。

```
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
```

在源和目标 SVM 之间创建 SVM 对等关系。

SnapMirror Create

从目标集群创建 SnapMirror 关系。

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

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

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

SnapMirror Initialize

初始化关系，以启动从源到目标的基线数据传输。

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

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

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

SnapMirror Update

执行更新，使目标卷成为源卷的最新镜像。这将使源卷创建一个新的 Snapshot 副本，该副本将传输至目标卷。

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

SnapMirror Break

此命令将中断源无限卷与目标无限卷之间的 SnapMirror 关系，并使目标无限卷成为读写卷。

```
cluster02::> snapmirror break -destination-path DVS1:DIV1
cluster02::> snapmirror show
Source           Destination   Mirror  Relationship  Total          Progress
Path            Type    Path        State   Status       Progress  Healthy  Updated
-----  -----  -----  -----  -----  -----  -----  -----
VS1:IV1  DP    DVS1_:DIV1
                           Broken-off
                           Idle
                           -   -   -
```

SnapMirror Resync

此命令用于在目标镜像中断的位置恢复镜像。目标卷变为 DP 镜像，并且该镜像可以手动更新，也可以按计划更新。

```
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

删除 SnapMirror 关系将删除源无限卷与目标无限卷之间的 SnapMirror 关系。但不会销毁无限卷。

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

SnapMirror Release

此命令将从源 SVM 中删除 DP 关系信息。将解除锁定并清理与关系相关的 Snapshot 副本。

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

有关集群模式 Data ONTAP 8.2 操作系统中无限卷的详细信息，请参见 TR-4178。

9.5 SnapMirror 和 NetApp 存储效率

SnapMirror 可在复制的卷中保持存储效率优势。如果源卷已删除重复数据，目标卷也将是已删除重复数据的状态。SnapMirror 在传输过程中不会增加已经过重复数据删除的数据。如果源卷已压缩数据，目标卷也将是已压缩数据的状态。复制已压缩的卷时，并不会解压缩源卷来读取要传输的数据；数据将以压缩状态复制到目标卷。

无法在源卷与目标卷之间启用不同配置的存储效率功能。例如，如果 SnapMirror 源卷上未启用数据压缩或重复数据删除，将无法单独对 SnapMirror 目标卷执行数据压缩或重复数据删除。

SnapMirror 在执行更新传输前会创建一个 Snapshot 副本。该 Snapshot 副本中的所有块都将锁定，并且无法对其进行重复数据删除。因此，如果需要最大限度地利用重复数据删除来节省空间，请在执行 SnapMirror 更新前运行重复数据删除过程。

9.6 SnapMirror 和卷移动

使用卷移动功能可以在集群中的节点之间无中断地移动卷。DP 镜像源卷或目标卷可使用 `volume move` 命令来移动。执行卷移动时，不必在源或目标上重新配置或修改 SnapMirror 关系。如果要移动集群间 SnapMirror 关系中的卷，则将该卷移动到的节点必须具有集群间 LIF 并连接到集群间网络，才能在日后执行 SnapMirror 更新。

卷移动对 SnapMirror 关系的影响取决于所移动的是源卷还是目标卷。如果 SnapMirror 传输当前正在进行，并且所移动的是 SnapMirror 源卷，则 SnapMirror 传输和卷移动传输可同时运行。但是，如果发生卷移动转换（集群模式 Data ONTAP 操作系统会将 I/O 重定向到新卷），活动的 SnapMirror 传输会暂时中断，然后自动从源卷的新位置开始继续进行。

注：在集群模式 Data ONTAP 8.1 操作系统中，对于 SnapMirror 目标卷，不能同时进行 SnapMirror 传输和卷移动传输。如果正在对 SnapMirror 目标卷进行 SnapMirror 传输，则无法为其启动卷移动。如果正在使用卷移动功能迁移 SnapMirror 目标卷，则无法执行 SnapMirror 更新传输。但是，从集群模式 Data ONTAP 8.2 操作系统开始，对于 SnapMirror 目标卷，可以同时运行 SnapMirror 传输和卷移动传输，但发生卷移动转换时除外，此时它们是互斥的（此转换会短暂持续几秒）。

有关卷移动的详细信息，请参见 TR-3975。

9.7 使用 SnapMirror 进行磁盘架故障保护

如果您决定要使用 SnapMirror 保护磁盘架以免发生故障，需要注意以下两点：

- 不能将卷镜像到同一 HA 对中。
- 故障转移不会自动进行。

您可以将卷镜像到同一集群上不同 HA 对中的不同节点。镜像到不同节点将确保另一个卷始终位于不同磁盘架中。如果您尝试将卷镜像到同一节点上的不同磁盘架，则必须是不同的聚合，但即便如此，仍然存在聚合可能在任意磁盘架中具有磁盘的风险。即使您尝试以其他方式进行设置（将聚合保留在各自的机架上），情况也可能因驱动器发生故障和使用备用磁盘而发生变化。这样可避免发生单点故障，并防止出现磁盘架故障。此处需要注意的是，故障转移不会自动进行。您必须手动中断 SnapMirror 关系、卸载客户端、重新在目标卷上挂载客户端，并更改 NFS 导出策略。

9.8 SnapMirror 和卷自动调整

目标卷必须大于或等于源卷。如果目标卷小于源卷，SnapMirror 更新将会失败。

最佳实践

保持源卷和目标卷的大小相同；但目标卷可略大于源卷。`-filesys-size-fixed` 选项确保 SnapMirror 卷的文件系统大小保持不变，这样即使目标卷大于源卷，也可以反转 SnapMirror 关系。

如果卷自动调整功能自动增加了源卷大小，或者您手动增加了源卷大小，则必须相应增加目标卷大小，以便与源卷大小相匹配。集群模式 Data ONTAP 8.1 操作系统不会自动增加目标卷的大小。请使用 CLI 或 System Manager 来增大目标卷；下次 SnapMirror 更新会自动将文件系统大小值复制到目标卷，以便与源卷的值相匹配。

如果自动调整功能增加了源卷大小，请将目标卷大小调整为至少与源卷的最大自动调整值相同，以避免需要手动调整目标卷大小。要避免需要额外容量来保证更大的目标卷，可以在目标上禁用空间保证。但是请注意，必须对目标系统的容量进行适当管理，以便在目标系统上为生成数据的操作留有空间。

从 Data ONTAP 8.2 开始，当自动调整功能增加了 SnapMirror 关系中源卷的大小时，目标卷的大小也将自动增加。这一点仅适用于 FlexVol 卷，而不适用于无限卷。

9.9 SnapMirror 和网络数据管理协议

您可以从 SnapMirror 源卷或目标卷执行网络数据管理协议 (NDMP) 备份。从 SnapMirror 目标卷执行 NDMP 备份比从源卷执行更具优势，其中包括：

- SnapMirror 传输可以快速进行，并且对源系统的影响小于 NDMP 备份。在备份的第一阶段，使用 NetApp Snapshot 副本并从主系统执行 SnapMirror 复制可显著缩短或消除备份时间。然后，从二级系统执行向磁带的 NDMP 备份。
- 出于性能或容量原因，很有可能会使用卷移动功能来移动 SnapMirror 源卷。将卷移动到不同节点后，必须重新配置 NDMP 备份作业才能从新位置备份卷。如果从 SnapMirror 目标卷执行备份，这些卷不太可能需要移动；因此，也就不太可能需要重新配置 NDMP 备份作业。

9.10 SnapMirror 和 Data ONTAP 的版本依赖关系

在运行 7-模式和集群模式 Data ONTAP 操作系统的系统之间，无法进行 DP 或 DR 复制。

在集群模式 Data ONTAP 8.1 操作系统中，SnapMirror 新增了几项复制功能，例如块级复制、支持 NetApp 存储效率、集群间复制以及中断、反转、重新同步关系。Data ONTAP 8.1 的 SnapMirror 实施不兼容 Data ONTAP 8.0 的 SnapMirror 实施。无法在运行集群模式 Data ONTAP 8.0 和 8.1 操作系统的系统之间进行复制。有关将运行集群模式 Data ONTAP 8.0 操作系统的系统升级到 8.1 的信息，请参见 [NetApp 支持站点上的《NetApp Data ONTAP 8.1 集群模式升级和还原/降级指南》](#)。

集群模式 Data ONTAP 8.2 操作系统推出了 SnapVault，可支持 SnapMirror 级联以及 SnapMirror 到磁带（仅用于传播），此外，它还可以实现多租户功能，可以让 SVM 管理员对复制进行管理。集群模式 Data ONTAP 8.2 操作系统还可以提高可扩展能力（并发传输的数量），并加快数据传输的速度。

在集群模式 Data ONTAP 8.2 操作系统中实施的 SnapMirror 与在集群模式 Data ONTAP 8.1 操作系统中实施的 SnapMirror 不兼容。可以在运行集群模式 Data ONTAP 8.1 和 8.2 的系统之间进行复制。从集群模式 Data ONTAP 8.1 操作系统升级到 8.2 后，现有的 SnapMirror 关系仍会保持在集群范围内。除非在将源节点和目标节点都升级到集群模式 Data ONTAP 8.2 操作系统后建立 SVM 对等关系，同时删除并重新创建 SnapMirror 关系，否则提高可扩展性不会给 SnapMirror 关系带来什么益处。

从集群模式 Data ONTAP 8.2.1 操作系统开始，在从 8.1 升级到 8.2.1 时，系统将自动完成以上过程。换句话说，在将目标节点升级到 8.2.1 并将源节点升级到 8.2 或更高版本后，如果源卷和目标卷位于同一个 SVM 中，或者在托管源卷和目标卷的两个 SVM 之间建立了 SVM 对等关系，则 SnapMirror 关系就会进行自动转换，从而因可扩展性提高而获益。有关将运行集群模式 Data ONTAP 8.1 操作系统的系统升级到 8.2 的信息，请参见 [NetApp 支持站点上的《NetApp Data ONTAP 8.2 集群模式升级和还原/降级指南》](#)。

从集群模式 Data ONTAP 8.1 操作系统开始，支持在不同版本之间进行以下复制：

- 在次要版本之间，允许在任意方向上进行复制。集群模式 Data ONTAP 8.1.x 操作系统是一个次要版本；因此可从 8.1.x 复制到 8.1.y 或从 8.1.y 复制到 8.1.x。
- 在主要版本之间，只允许从旧版复制到新版。集群模式 Data ONTAP 8.1 操作系统是一个主要版本；因此可以从 8.1 复制到更高的主要版本（如 8.2）。但是，不允许从较高版本（如 8.2）复制到较低的主要版本（如 8.1）。

注：集群模式 Data ONTAP 8.0 操作系统版本系列不在此列。

9.11 SnapMirror 和网络压缩

随着数据量不断增长，而网络带宽成本日益提高，客户必须能够花更少的钱办更多的事。随着要保护的数据量增长，需要更多的网络带宽来维护回复时间目标 (RPO) 或复制窗口。否则，复制时间会随着通过网络发送到灾难恢复站点的数据的量的增加而增加。另一种方法是如果您不希望或不能增加网络带宽，可以降低复制频率，从而降低 RPO 值，进而降低丢失较大数据的可能。

SnapMirror 本机网络压缩功能可以降低通过网络复制数据的量。如下所述，它还能为您提供更多的灵活性和选择空间。

保持同等 RPO 水平

问题。您的数据复制需求增加。您需要更多的带宽以保持相同的 RPO 水平。

解决方案。通过使用网络压缩，无需购买额外网络带宽即可保持相同的 RPO。

无需购买额外带宽即可提高 RPO

问题。您正在使用您所有的网络带宽。然而，客户希望降低数据丢失的风险 – 换言之，就是提高其 RPO。

解决方案。通过使用网络压缩，无需购买更多的网络带宽您便可以改善您的 RPO。

将网络带宽用于其他用途

问题。您的复制占用了您所有的带宽。您希望无需购买额外带宽，即可将网络带宽另作他用，如客户端访问或应用程序等。

解决方案。通过使用网络压缩，可以在不降低 RPO 的情况下降低 SnapMirror 占用的带宽，从而腾出网络带宽用于其他用途。

加快初始传输速度

问题。初始 SnapMirror 传输会较大，因此在带宽约束下会耗费较长时间才可完成。

解决方案。通过使用网络压缩，可以加速初始 SnapMirror 传输。

什么是 SnapMirror 网络压缩？

SnapMirror 网络压缩在 SnapMirror 传输网络上启用数据压缩。这是 SnapMirror 软件内置的一项本机功能。SnapMirror 网络压缩与卷数据压缩不同。SnapMirror 网络压缩不是压缩空闲数据。下图显示非常高水平的 SnapMirror 网络压缩流量。

图 42) SnapMirror 网络压缩功能图。



源系统上，需要发送到目标系统的数据块传递到压缩引擎，压缩引擎将数据块进行压缩。根据存储系统上可用的 CPU 数量，源系统上的压缩引擎会创建若干线程。这些压缩线程有助于以并行方式压缩数据。然后通过网络发送经压缩的块。

在目标系统上，经压缩的块通过网络接收，然后进行解压缩。目标压缩引擎也会创建若干线程以并行方式解压数据。经解压缩的数据重新排序并保存到相应的卷上的磁盘中。

换言之，启用 SnapMirror 网络压缩后，系统会另外执行两个步骤：在通过网络发送数据之前在源系统上进行压缩以及在将数据写入磁盘之前在目标系统上进行解压缩。

前提条件

源系统和目标系统上的集群模式 Data ONTAP 8.3 操作系统都支持 SnapMirror 网络压缩。

支持 Data ONTAP 8.3 操作系统的所有平台也都支持 SnapMirror 网络压缩。

启用和禁用网络压缩

SnapMirror 网络压缩可以通过 SnapMirror 策略中的 `-is-network-compression-enabled` 选项启用或禁用。但不能为活动传输启用此类压缩。要为现有传输启用压缩，必须先中止传输，将 SnapMirror 策略中的 `-is-network-compression-enabled` 选项设置为 `true`，然后再恢复传输。

报告压缩比例

SnapMirror 网络压缩比例会在 snapmirror show -instance 输出中进行报告。

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

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

只有处于传输状态时，才会显示压缩比例

10 SnapMirror 规模估算建议

10.1 并发复制操作

支持的可同时执行的 SnapMirror 操作数量是有限的。每个节点都有此限制，并且此限制因 Data ONTAP 的平台和版本而异。有关每个节点允许的并发 SnapMirror 操作数量的信息，请参见 [NetApp 支持站点上的《NetApp Data ONTAP 集群模式数据保护管理指南》](#)（请选择相应的 Data ONTAP 版本）。

最佳实践

了解集群模式 Data ONTAP 操作系统中哪些操作属于 SnapMirror 操作也非常重要。定期按计划对 SnapMirror DP 或 LS 关系进行的 SnapMirror 更新就属于 SnapMirror 操作。但是，卷移动和卷复制操作也使用 SnapMirror 机制将数据从一个聚合移至另一聚合。因此，在计划并发操作时，最好考虑到环境中卷移动和卷复制操作的频率。

集群模式 Data ONTAP 操作系统可以将 NetApp 集群扩展为两个节点以上，从而提高了可扩展性。集群中的每个节点均提供 CPU 和内存资源，以用于复制该节点所拥有的卷。

最佳实践

为了优化复制，请将复制的卷分布在集群中的不同节点上，不要将需要复制的所有卷放在一个节点上。这种最佳做法使集群中的所有节点可以分担复制活动。

10.2 建议的复制时间间隔

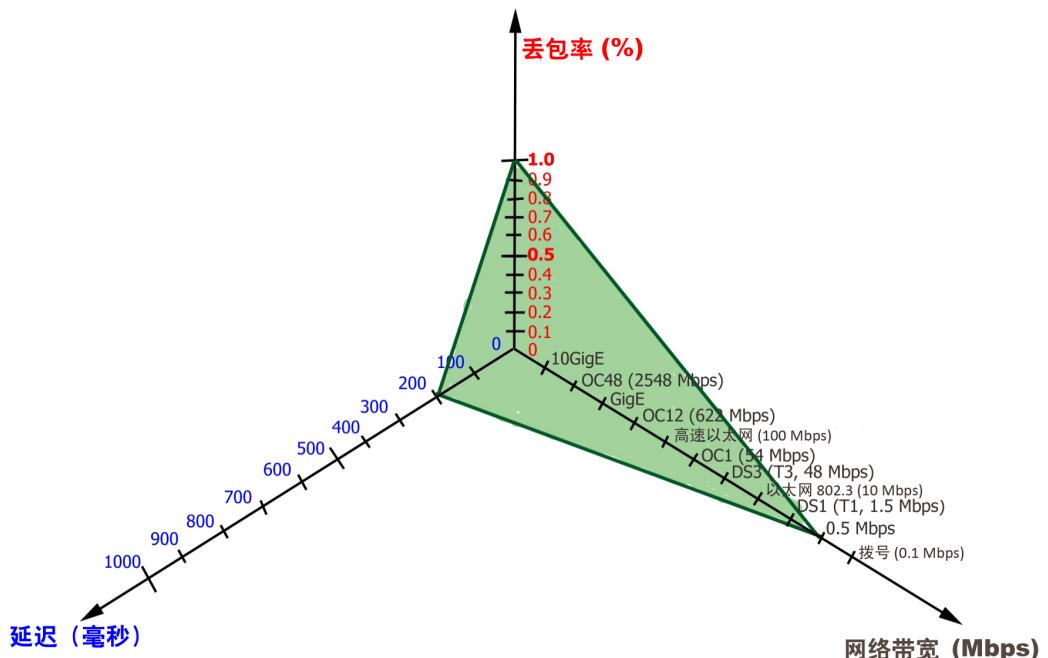
SnapMirror 更新需要在源节点与目标节点之间建立通信会话、创建和删除 Snapshot 副本，并确定要将哪些数据块发送到目标节点。因此，虽然 Data ONTAP 计划程序支持创建每分钟运行的计划，但是 NetApp 不建议每分钟执行一次 SnapMirror 更新操作。不过，根据您的环境，可以单次执行 SnapMirror 更新操作。有关适当的 SnapMirror 和系统规模估算准则，请参见 [System Performance Modeler](#)。

10.3 网络规模估算要求

要支持所需的复制时间间隔，需要一个具有适当带宽可用于传输系统数据载入速率的网络。对于可支持集群间复制的网络特性，存在一定限制。

集群间复制的网络规模估算要求

图 43) 实现最佳性能需要考虑的因素: 丢包率 (%)、延迟 (毫秒) 和网络带宽 (Mbps)。



必须根据数据变化率和更新时间间隔恰当地估算集群间网络的规模，以符合解决方案的恢复点目标 (RPO) 和各个节点的性能特性。可以在最小带宽为 0.5 Mbps、最大往返网络延迟为 200 毫秒往返时间 (Round-Trip Time, RTT)、丢包率为 1% 的网络 (图中绿色三角覆盖的部分) 中执行集群间 SnapMirror。

最佳实践

所有用于集群间复制的路径应具有同等的性能特性，这一点非常重要。在配置多路径时，如果节点在慢速路径和快速路径上各安排一个集群间 LIF，将对性能产生不利影响，因为这样数据将同时跨慢速和快速路径进行多路传输。

集群内复制的网络规模估算要求

所有集群内传输，包括集群内 SnapMirror DP 镜像、LS 镜像以及卷移动与卷复制操作，都使用同一集群内的节点间的私人集群互连。无法配置集群互连带宽。

11 故障排除技巧

11.1 集群对等关系的故障排除

1. 运行 `cluster peer show` 命令验证集群对等关系的可用性；此命令将显示现有已配置的所有集群对等关系。

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

2. 将 `-instance` 选项添加到命令中，查看有关集群对等关系的更多详细信息；加入 `-cluster <cluster_name>` 可查看特定集群的结果。`-instance` 选项将显示用于集群间通信的远程地址。

```
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. 运行 `cluster peer ping` 命令查看有关每个集群间地址之间的连接情况的信息，包括 RTT 响应时间。对于多个已配置的集群对等关系，使用 `-cluster <cluster_name>` 选项为一个特定对等关系执行 Ping 操作。`cluster peer ping` 命令将显示集群间接口间的 Ping 结果。如前面所述，在通过本地集群与远程集群之间的多个路径执行集群间 SnapMirror 镜像时，每个路径都必须具有相同的性能特性。在此示例中，Ping 响应时间 (RTT) 相对等同于 Ping 到目标集群显示为 `cluster02` 的节点的时间。

```
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
```

11.2 SVM 对等关系故障排除

下面是常见问题列表以及对这些问题进行故障排除的方法：

1. 在集群间环境中执行 SVM 对等操作失败：
 - a. 检查两个集群是否位于同一环境中。
 - b. 检查对等集群是否可以访问。
 - c. 检查两个集群是否都在运行 SN 且已启用 SVM 对等功能。
 - d. 检查对等 SVM 名称是否未与 SVM 对等表中使用对等 SVM 名称的另一个集群关联。
 - e. 检查 `mgwd.log` 和控制台日志中是否有错误消息。
2. 在集群内/集群间环境中执行 SVM 对等操作失败：
 - a. 检查两个集群是否都在运行 SN 且已启用 SVM 对等功能。
 - b. 检查本地 SVM 名称与对等 SVM 名称是否不同。
 - c. 检查 `mgwd.log` 和控制台日志中是否有错误消息。
3. 运行 `vserver peer show` 命令验证 SVM 对等关系；此命令将显示已配置的所有现有 SVM 对等关系。

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

4. 通过 vserver peer show-all 命令检查是否有任何通知。

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

11.3 了解 SnapMirror 关系的状态

“Healthy”（运行状况良好）列指示了 SnapMirror 关系的状态。此列会显示在 CLI 上的 snapmirror show 命令输出中以及 Cluster Element Manager Web 界面上，同时还会与“Healthy”（运行状况良好）列一样显示在 OnCommand System Manager 的 SnapMirror 关系状态中。

在此示例中，snapmirror show 命令显示“Healthy”（运行状况良好）列。

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

“Healthy”（运行状况良好）列显示 SnapMirror 关系的运行状况。它还指示是否保持了 RPO，而无需通过判断上次更新的时期来分析关系的运行状况。例如，对于计划定期更新的 SnapMirror 关系，如果上次更新在试图开始下次更新之前完成，则“Healthy”（运行状况良好）列将显示 true，如此示例呈现的输出中的第一个关系中所示。

如果某个计划更新在下一次计划更新开始时还在进行，该关系的“Healthy”（运行状况良好）列将显示 false。此外，如果之前的计划更新或手动更新失败，该关系的“Healthy”（运行状况良好）列也将显示 false。

如果正在进行传输，则“Healthy”（运行状况良好）列将显示 -，而“Total Progress”（总进度）列将显示当前运行的传输的进度。

当关系处于未初始化状态时，“Healthy”（运行状况良好）列也将显示 -，如第三个关系中所示。如果因为使用了 snapmirror break 命令而使关系处于中断状态，也会显示 -。

对于源系统上的关系，“Healthy”（运行状况良好）列会显示 -。要查看有关 SnapMirror 关系运行状况的权威信息，请从目标卷查看该关系。

如果目标卷脱机或无法访问，则“Mirror State”（镜像状态）列也会显示 -。

11.4 SnapMirror 关系故障排除

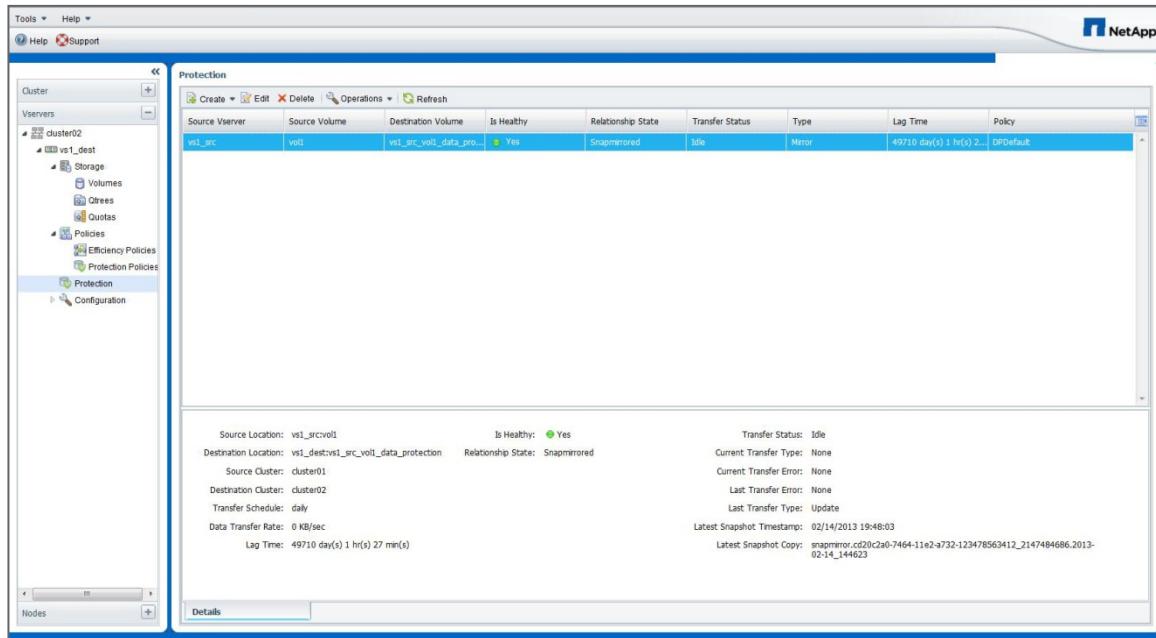
要确定上次对某个关系完成 SnapMirror 传输的时间，可以查看集群模式 Data ONTAP 8.2 操作系统上实例信息中的“Exported Snapshot Timestamp”（导出的 Snapshot 的时间戳）部分。

```
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
```

注： 上次 Snapshot 的时间戳信息也会显示在 System Manager 界面的底部，如图 44 中所示。

图 44) 传输的时间戳信息。



有关 SnapMirror 关系的故障排除问题，请查看事件日志中有关关系的信息。在 `event log show` 命令中使用 `-messagename` 选项，可以在事件日志中筛选与 SnapMirror 相关的消息，如以下示例所示。指定 `mgmt.snapmir*` 消息名称可以筛选输出内容并仅查找与 SnapMirror 相关的消息。

```
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.
```

要查找有关特定卷的错误消息，可通过在 `-event` 选项中指定卷的名称（用星号括起）来进一步筛选消息列表，如以下示例所示。

```
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.
```

所有 SnapMirror 事件均记录在目标卷所在节点上的 `SnapMirror_audit.log` 和 `SnapMirror_error.log` 文件中。此节点可能不是发出命令的节点。可以通过运行 `snapmirror show -fields destination-volume-node` 命令来确定运行此操作的节点。使用 OnCommand System Manager 3.0 可以查看 SnapMirror 日志文件。

您也可以使用 System Manager 从其余的事件日志中单独查看 SnapMirror 日志：“Cluster”（集群）>“Diagnostics”（诊断）>“Logs”（日志）>“SnapMirror Log”（SnapMirror 日志）。从“Select node”（选择节点）下拉列表中，选择您要查看的卷所属的节点，如图 45 中所示。

图 45) SnapMirror 日志。

目前无法在 System Manager 3.0 中查看 SnapMirror 日志。正在开发中。

12 通过配置和故障转移实现灾难恢复

通过配置和故障转移实现灾难恢复实际上就是对集群内 SnapMirror DP 镜像的灾难恢复过程进行了概述。该过程将分两个部分来介绍。第一部分介绍在需要进行故障转移之前必须完成的步骤，以便为目标做好故障转移的准备。对于灾难恢复场景，应完成这些步骤来准备灾难恢复站点。第二部分介绍执行故障转移的必要步骤。

每个环境都有自己的独特特征；每个环境对灾难恢复计划都有影响。根据所部署灾难恢复解决方案的类型，每个组织的灾难恢复情况可能截然不同。要取得成功，需要正确规划、记录并实际演练灾难恢复场景。

12.1 环境的故障转移的要求和假设前提

要获得成功的灾难恢复体验，需要考虑一些常规要求和假设前提。以下所列各项并不详尽。根据每个环境的配置，还有许多可变因素需要考虑。

- 系统管理员可以访问用于管理灾难恢复站点和执行故障转移的工作站或服务器桌面会话。
- 系统管理员具有访问系统所需的所有相应凭据、帐户、密码等。
- 从执行操作的任何位置均可连接到灾难恢复网络。
- 灾难恢复站点已经有一些基础架构服务器，并且这些服务器可访问。这些系统可为管理员提供在环境中工作并执行恢复计划所需的基本服务。
 - 灾难恢复站点的 Active Directory[®] 服务，用于提供身份验证
 - 灾难恢复站点的 DNS 服务，用于提供名称解析
 - 灾难恢复站点的许可证服务器，可以为所有需要许可服务的应用程序提供许可服务

注：请确保灾难恢复站点可提供执行必要 Active Directory FSMO 角色的服务器，这一点非常重要。有关将角色转移到未受故障影响的 Active Directory 服务器或从故障服务器获取这些角色的信息，请参见 [Microsoft 知识库 255504](http://support.microsoft.com/kb/255504): <http://support.microsoft.com/kb/255504>。

- 灾难恢复站点的时间已经与主站点的同一个源或某个与主站点同步的源进行了同步。
- 已使用 SnapMirror 将所有必需的 NetApp 卷复制到灾难恢复站点。
- 已对 SnapMirror 操作实施监控，并按照设计的 RPO 对其进行了更新。
- 灾难恢复 NetApp 控制器具有所需的容量。该容量必须能够为灾难恢复环境中计划进行的日常操作提供支持。
- 灾难恢复站点的所有应用程序服务器均已配置适当的连接，能够连接到灾难恢复存储阵列。
- 有方法能够将发生故障的主网络与灾难恢复站点隔离开。这一点非常必要，因为如果导致灾难的事件本身是临时性的或间歇性的（如长时间断电），则在主站点系统重新启动时，服务可能会与灾难恢复站点此时正在运行的已恢复操作发生冲突。
- 可以通过相应计划使用户和应用程序能够访问灾难恢复站点上的数据和服务，例如，更新 DNS 以使向主站点 SVM 发出的主目录挂载请求定向到灾难恢复站点的 SVM。

12.2 灾难恢复配置的最佳实践

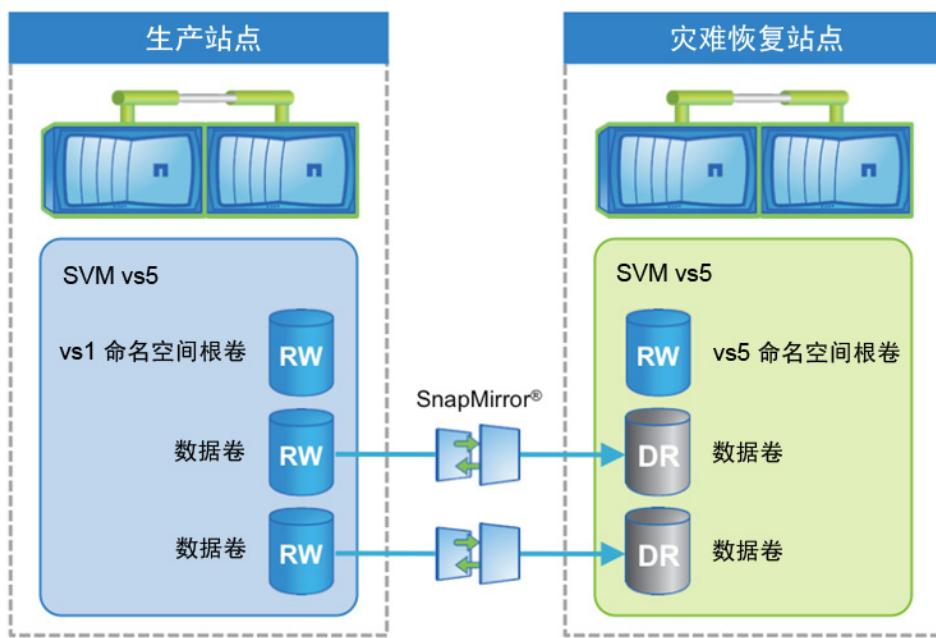
最佳实践
<ul style="list-style-type: none">应该将源站点上某个 SVM 中的卷复制到目标站点上的某个 SVM 中。对于 NAS 客户端，SVM 是 NAS 命名空间的根；而在 SAN 环境中，则是单个存储目标。如果将一个 SVM 中的某些 NAS 卷复制到目标站点的不同 SVM，则无法将所有这些卷恢复到同一个命名空间中。包含 LUN 的卷也是如此；如果将这些卷复制到目标站点的不同 SVM，则不会将所有这些 LUN 置于同一个 SAN 目标下。目标 SVM 应与源 SVM 属于同一个 Active Directory、LDAP 或 NIS 域。这一点是必需的，只有这样，在将 NAS 卷恢复到无法对 NAS 文件中存储的访问控制列表 (ACL) 进行身份验证的 SVM 中后，这些 ACL 才不会损坏。更改并修正文件级 ACL 以便从其他域访问的过程可能极为困难，并且非常耗时。此外，可以使用相同凭据对 SAN 客户端中运行的工具（如 NetApp SnapDrive® for Windows®）进行身份验证也非常重要。由于目标 SVM 与源 SVM 不同，而 NetApp 又建议将其加入同一个 Active Director 域，因此目标 SVM 必须使用不同的 SVM 名称加入该域。通常的做法是，设置一个与源系统名称不同的灾难恢复系统。在灾难恢复故障转移情形中，通常会通过更改 DNS 名称解析或使用 DNS 别名来将客户端重定向到已恢复的存储系统的名称，以便仍然可使用相同 UNC 路径名访问 CIFS 共享，并且 NFS 客户端也能访问预期的路径。使用与源卷名称相同的目标卷名称并不是必需的，但是如果挂载卷的接合路径也与该卷同名，则这样做更便于管理向目标位置挂载目标卷的过程。在为 SVM 构建目标 NAS 命名空间时，其路径和目录结构应与源 SVM 完全一致。许多 SAN 客户端无法访问位于完全只读的容器（如 SnapMirror 目标卷）中的 LUN。一般情况下，在执行 SnapMirror 中断操作后，应将 LUN 映射到 igrup 并由 SAN 客户端挂载。按照下一节所述，提前配置目标 SVM。这样做可以大幅提升存储系统灾难恢复过程的速度，进而可能缩减该过程，使其只需执行少量 SnapMirror 中断操作并更新部分 DNS 别名。在源站点上创建新卷后，必须创建 SnapMirror 关系来复制这些卷。创建并复制这些卷之后，应在灾难恢复站点配置与这些卷有关的设置，以便一旦发生灾难事件，它们可以准备就绪。

12.3 准备故障转移的目标系统

从集群模式 Data ONTAP 8.1 操作系统开始，灾难恢复过程的许多部分都可以在发生灾难恢复事件之前提前进行准备。例如，将卷挂载到命名空间、创建 CIFS 共享、分配 NFS 导出策略等任务，均可提前完成。不能使用 SnapMirror 复制目标 SVM 上可能存在的独立配置信息，例如 SVM 域成员资格、CIFS 配置、NFS 策略、Snapshot 策略计划或 NetApp 存储效率策略。

图 46 说明了灾难恢复的卷布局。

图 46) 灾难恢复的卷布局。



复制卷之后，完成以下步骤以准备用于故障转移的目标系统，如图 46 中所示。

NAS 和 SAN 环境

1. 配置目标 SVM 在适当 Active Directory、LDAP 或 NIS 域中的成员资格。
2. 确定目标 SVM 与源 SVM 属于同一个域，这样 NetApp SnapDrive for Windows 等工具的身份验证才不会中断，并且可以根据 SnapMirror 复制的文件级 ACL 对相同用户进行身份验证。
3. 在目标集群中创建任何所需的非默认 Snapshot 副本策略。
注：NetApp 建议在目标集群中配置与源集群中的 Snapshot 副本策略采用相同计划的 Snapshot 副本策略。必须在故障转移后将 Snapshot 副本策略应用到 DP 卷。
4. 在目标 SVM 中创建 NetApp 存储效率策略。
注：如果为源 SVM 中的卷分配了 NetApp 存储效率策略，则必须在目标 SVM 中创建一个策略，以便在故障转移后在灾难恢复站点计划重复数据删除过程。必须在故障转移后将 NetApp 存储效率策略应用于 DP 卷。

NAS 环境

1. 确认已将源 SVM 中所有必需的卷复制到目标 SVM。卷可以挂载在子文件夹中，也可以挂载在命名空间中的其他卷内。为满足此条件，请务必确保在目标上正确重建命名空间所需的所有卷均已进行了复制。
2. 验证目标 SVM 根卷上的安全模式和权限。必须正确设置目标 SVM 命名空间的根卷上的安全模式和权限，否则故障转移后可能无法访问 NAS 命名空间。
3. 将目标 NAS 卷挂载到目标 SVM 命名空间。

SnapMirror 不会复制 SVM 命名空间的接合路径信息。NAS 卷没有接合路径，因此除非在故障转移前事先已挂载它们或在故障转移后挂载它们，否则在 SnapMirror 中断后将无法访问这些卷。

挂载这些卷时，请使用与源卷在源 SVM 中的挂载位置相同的接合路径将它们挂载到命名空间中。这样做非常重要，目的是使恢复的命名空间中的路径与主站点中存在的路径相同。如果路径不同，客户端的挂载点、链接、快捷方式和别名可能无法找到正确的路径。

注：不能将卷挂载（嵌套）在仍处于 DP 状态的其他卷内。运行 `snapmirror break` 命令后，必须挂载在复制的卷中嵌套了挂载点的所有卷，并且必须创建所有 CIFS 共享。

4. 使用与源 SVM 上相同的共享名在目标 SVM 上创建 CIFS 共享。客户端可以访问 CIFS 共享；但是在卷完成故障转移之前，所有数据均为只读。
5. 将相应的 ACL 应用到目标 Storage Virtual Machine 上的 CIFS 共享。
6. 为目标 SVM 创建适当的 NFS 导出策略。
7. 将 NFS 导出策略分配给目标卷。客户端可以访问 NFS 共享；但是在卷完成故障转移之前，所有数据均为只读。

SAN 环境

1. 如果目标 SVM 使用端口集，则可以在故障转移前根据需要对这些端口集进行配置。
2. 在目标 SVM 上配置 igroup。

通常情况下，会有不同的应用程序服务器连接到灾难恢复站点上恢复的存储。可以将这些服务器中的启动程序预先配置到目标 SVM 的适当 igroup 中。

由于某些主机不支持连接到只读容器（SnapMirror 目标卷）中的 LUN，因此通常会在故障转移后将 LUN 映射到 igroup。

SnapMirror 工具包（适用于集群模式 Data ONTAP 8.2 操作系统）

此工具（SnapMirror 工具包）的主要用途是，改善在集群模式 Data ONTAP 8.2 操作系统中设置和运行 SnapMirror（以及 SnapVault）的用户体验。参与集群模式 Data ONTAP 8.2 操作系统前期验证计划的质量保证人员、可用性评测团队以及客户提供的反馈表明，集群模式 Data ONTAP 8.2 操作系统中的 SnapMirror 在设置和管理方面比 7-模式 SnapMirror 更加复杂。我们将在未来的集群模式 Data ONTAP 操作系统版本中改进 SnapMirror 的可用性，但是，使用这些脚本的效果立竿见影。它们具有轻型、可移植的特点，与使用集群模式 Data ONTAP 操作系统命令行界面（CLI）相比，用户体验更为简单。此外，客户还能够以这些脚本为基础构建自己内部的自动化工具。

您可以从 SE 实用程序工具箱下载 SnapMirror 工具包（SMTK）：

<http://support.netapp.com/NOW/download/tools/smtk>

12.4 执行故障转移

由于灾难恢复所需的大部分必要配置工作已在故障转移前执行完毕，因此灾难恢复期间进行故障转移所需的实际步骤已大大减少。这些步骤如下所示。

NAS 环境

1. 执行 SnapMirror 中断操作，对每个卷进行故障转移。在集群模式 Data ONTAP 操作系统中，可以使用通配符通过一个命令对多个卷执行 SnapMirror 操作。以下示例将对名为 vs5 的目标 SVM 中的所有卷执行故障转移；可以在此命令中使用部分卷名仅对部分卷执行该操作。

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

2. 如果已将卷挂载到命名空间，并且已创建并应用了 CIFS 共享和 NFS 导出策略，则客户端将具有对 NAS 数据的读写访问权限。
3. 将客户端重定向到已恢复的存储。

通常的做法是，设置一个与源系统名称不同的灾难恢复系统。在灾难恢复故障转移场景中，通常通过更改 DNS 名称解析或使用 DNS 别名来将客户端重定向到已恢复的存储系统的名称。这样就可以使用同一 UNC 路径名访问 CIFS 共享，并且 NFS 客户端也可访问预期的路径。或者，也可以从 Active Directory 中删除发生故障的源存储系统，然后删除恢复存储系统，再使用与源系统相同的名称将恢复存储系统重新添加到 Active Directory。但是，在大范围的 Active Directory 环境中传播此项更改可能需要花费一些时间。

SAN 环境

1. 执行 SnapMirror 中断操作，对每个卷进行故障转移。在集群模式 Data ONTAP 8.1 操作系统中，可以使用通配符通过一个命令对多个卷执行 SnapMirror 操作。以下示例将对名为 vs5 的目标 SVM 中的所有卷执行故障转移；可以在此命令中使用部分卷名仅对部分卷执行该操作。

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

2. 通过将卷中的 LUN 映射到适当的 igrup，使这些 LUN 可在灾难恢复站点供 SAN 客户端使用。
3. 在 SAN 客户端上，执行存储重新扫描操作以发现已连接的 LUN。

12.5 故障转移后的卷配置

由于无法将 Snapshot 副本策略和 NetApp 存储效率策略分配给处于 DP 状态的卷，因此必须在故障转移后分配这些策略。

1. 如果使用 Data ONTAP Snapshot 副本计划，请向恢复的卷分配 Snapshot 副本策略。在 SAN 环境中，通常会在客户端上计划 Snapshot 副本。
2. 如果正在使用 NetApp 存储效率技术，请向已恢复的卷分配存储效率策略。

13 SnapMirror 的过渡

目前仍有一些 7-模式客户使用 QSM 和 VSM（同步、半同步和异步）。如何帮助这些客户过渡到集群模式 Data ONTAP 操作系统？TR-4052：《集群模式 Data ONTAP 过渡指南》将介绍方面的内容。

参考资料

本技术报告使用了以下参考资料：

- [NetApp 支持站点上的《Data ONTAP 8.1 集群模式数据保护指南》](#)
- [NetApp 支持站点上的《Data ONTAP 8.1 集群模式升级和还原/降级指南》](#)
- [NetApp 支持站点上的《Data ONTAP 8.2 集群模式数据保护指南》](#)
- [NetApp 支持站点上的《Data ONTAP 8.2 集群模式升级和还原/降级指南》](#)
- [NetApp 支持站点上的《Data ONTAP 8.3 集群模式数据保护指南》](#)
- [NetApp 支持站点上的《Data ONTAP 8.3 集群模式升级和还原/降级指南》](#)
- [TR-3975：《集群模式 Data ONTAP 8.2 中的 DataMotion for Volumes 概述》](#)
- [TR-4178：《无限卷部署和实施指南》](#)
- [TR-4183：《集群模式 Data ONTAP 的 SnapVault 最佳实践指南》](#)
- [TR-4052：《集群模式 Data ONTAP 过渡指南》](#)

版本历史

版本	日期	文档版本历史
3.0 版	2015 年 4 月	Vincent Goveas
2.2 版	2013 年 11 月	Amit Prakash Sawant
2.1 版	2013 年 7 月	Amit Prakash Sawant
2.0 版	2013 年 4 月	Amit Prakash Sawant
1.0 版	2012 年 2 月	Larry Touchette

要验证您的特定环境是否支持本文档所述的确切产品和功能版本，请参见 NetApp 支持站点上的[互操作性表工具 \(IMT\)](#)。NetApp IMT 中定义的产品组件和版本可用于构建 NetApp 所支持的配置。具体的配置结果取决于每个客户如何依照所发布规格进行安装。

版权信息

版权所有 © 1994 - 2015 NetApp, Inc. 保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NETAPP 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NETAPP 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：美国政府使用、复制或公开本文档受 DFARS 252.277-7103（1988 年 10 月）和 FAR 52-227-19（1987 年 6 月）中“技术数据和计算机软件权利”条款第 (c)(1)(ii) 条规定的限制条件的约束。

商标信息

NetApp、NetApp 标识、Go Further, Faster、AltaVault、ASUP、AutoSupport、Campaign Express、Cloud ONTAP、Clustered Data ONTAP、Customer Fitness、Data ONTAP、DataMotion、Fitness、Flash Accel、Flash Cache、Flash Pool、FlashRay、FlexArray、FlexCache、FlexClone、FlexPod、FlexScale、FlexShare、FlexVol、FPolicy、GetSuccessful、LockVault、Manage ONTAP、Mars、MetroCluster、MultiStore、NetApp Insight、OnCommand、ONTAP、ONTAPI、RAID DP、RAID-TEC、SANtricity、SecureShare、Simplicity、Simulate ONTAP、SnapCenter、Snap Creator、SnapCopy、SnapDrive、SnapIntegrator、SnapLock、SnapManager、SnapMirror、SnapMover、SnapProtect、SnapRestore、Snapshot、SnapValidator、SnapVault、StorageGRID、Tech OnTap、Unbound Cloud、WAFL 以及其他名称是 NetApp Inc. 在美国和/或其他国家或地区的商标或注册商标。所有其他品牌或产品均为其各自所有者的商标或注册商标，应予同样对待。有关最新的 NetApp 商标列表，请参见 <http://www.netapp.com/cn/legal/netapptmlist.aspx>。TR-4015-0415-zhCN