Contents

Deciding whether to use this guide ............................................................. 8

Introduction to data protection ................................................................. 9

Methods of protecting data ........................................................................ 9
Monitoring and protecting database validity by using NVFAIL .................. 10
What a data loss disaster is ........................................................................ 10
Tools for protecting against data-loss disasters ....................................... 11
Data protection in a SAN environment .................................................... 12
Types of data protection policies ............................................................. 12

Planning your data protection strategy .................................................... 14

Working with Snapshot copies ................................................................. 14
What a Snapshot copy is .......................................................................... 14
User access to Snapshot copies ............................................................... 14
Backup and recovery tasks you can perform with Snapshot copies .......... 15
Maximum number of Snapshot copies ................................................... 15
Where to find information about Snapshot copies of Infinite Volumes .... 16
Creation of Snapshot copy schedules ..................................................... 16
Deleting Snapshot copies automatically .................................................. 18
Viewing settings for the automatic deletion of Snapshot copies ............... 19
What Snapshot disk consumption is ....................................................... 19
What the Snapshot copy reserve is ........................................................ 21

Working with mirroring technology ......................................................... 22
Components of a mirror relationship ....................................................... 22
Data protection mirror relationships for FlexVol volumes ....................... 23
Data protection mirror relationships for SVMs with FlexVol volumes ..... 23
Where to find information about data protection mirror relationships for
Infinite Volumes .................................................................................... 24
Unexpected destination volume growth during data protection mirror
transfers ................................................................................................. 24
Path name pattern matching .................................................................. 24
Language setting requirement ................................................................ 24
User access to destination volumes ....................................................... 25
Requirements for creating relationships between clusters or SVMs ....... 25
Limitations for data protection mirror relationships ................................ 26

Working with FlexVol volume SnapVault backups .................................. 28
What a SnapVault backup is ................................................................. 28
Which data gets backed up and restored from a SnapVault backup .......... 29
Which data does not get backed up to a SnapVault backup ................. 29
How a SnapVault backup works ............................................................ 30
How SnapVault backups work with data compression ......................... 30
SnapVault backup limitations .............................................................. 31
Data protection for SVM namespace and root information .................... 31
Guidelines for planning Snapshot copy schedule and retention for SnapVault backups ................................................................. 31
Supported data protection deployment configurations .................................................. 32
What a basic backup deployment is ........................................................................... 34
Firewall requirements for intercluster SnapMirror relationships .......................... 34
What source-to-destination-to-tape backup is .......................................................... 34
How a mirror-mirror cascade works ....................................................................... 34
How a SnapMirror-SnapVault cascade works ..................................................... 35
How a SnapVault-SnapMirror cascade works ....................................................... 36
How a mirror-SnapVault fanout works ................................................................... 37
How a multiple-mirrors fanout works ...................................................................... 37

Protecting data using Snapshot copies ................................................................. 39
Managing Snapshot copies ....................................................................................... 39
Commands for managing Snapshot copies ............................................................ 39
Managing Snapshot policies ..................................................................................... 40
How Snapshot policies are associated with volumes ................................................. 40
Commands for managing Snapshot policies and schedules .................................... 40
Strategies for creating a Snapshot copy policy ...................................................... 41
Naming convention for scheduled Snapshot copies ................................................. 42
What prefixes are .................................................................................................... 42
Using prefixes to name automatic Snapshot copies ............................................... 42
Restoring files from the Snapshot copy of a FlexVol volume .................................... 43
Restoring a single file from a Snapshot copy of a FlexVol volume ............................. 43
Restoring part of a file from a Snapshot copy of a FlexVol volume ............................ 44
Restoring the contents of a volume from a Snapshot copy ....................................... 44
Snapshot restoration using Shadow Copy Client tools ............................................. 45
Managing Snapshot copy disk space ....................................................................... 45
Monitoring Snapshot copy disk consumption ....................................................... 45

Managing data protection using SnapMirror policies .......................................... 47
Commands for managing SnapMirror and SnapVault policies ............................... 47
How SnapMirror policies work with clusters and SVMs ......................................... 47
Comparison of what cluster administrators and SVM administrators can manage ......................................................................................... 48
Guidelines for naming Snapmirror and SnapVault policies ...................................... 48
Preserving Snapshot copies for SnapVault relationships after reaching retention limit .................................................................................. 49
Example of creating a tiered backup policy ............................................................ 50

Managing peer relationships for data backup and recovery ................................. 52
Managing cluster relationships ................................................................................. 52
What a cluster peer is ............................................................................................... 52
Commands for managing cluster peer relationships ............................................... 52
Cluster peer security using authentication passphrases ............................................. 54
Connecting one cluster to another cluster in a peer relationship .............................. 54
Modifying a cluster peer relationship ....................................................................... 73
Deleting a cluster peering relationship ................................................................... 74
Managing SVM peer relationships ................................................................. 74
What an SVM peer relationship is .............................................................. 75
States of SVM peer relationships ............................................................... 76
Creating an SVM peer relationship ............................................................ 76
Accepting an SVM peer relationship ........................................................ 79
Rejecting an SVM peer relationship .......................................................... 80
Modifying the peering application on an SVM peer relationship ............. 81
Deleting an SVM peer relationship ............................................................ 82
Suspending an SVM peer relationship ....................................................... 83
Resuming an SVM peer relationship ........................................................ 83
Displaying information about SVM peer relationships ............................ 84

Providing disaster recovery using mirroring technology ......................... 85
Creating a data protection mirror copy for FlexVol volumes .................... 85
Correcting a SnapMirror initialization failure ......................................... 86
Creating a version-flexible SnapMirror relationship ............................... 87
Considerations when using version-flexible SnapMirror relationships ...... 90
Ways to set up version-flexible SnapMirror relationships ...................... 90
Converting a SnapMirror relationship to a version-flexible SnapMirror
    relationship .......................................................................................... 91
Managing mirror relationships ..................................................................... 92
Commands for managing SnapMirror relationships .................................. 92
Using extended queries to operate on many SnapMirror relationships .... 95
What tape seeding is .............................................................................. 95
Performing tape seeding using SMTape .................................................... 95
Scalability limits for SMTape backup and restore sessions ..................... 102
Listing the schedule state of a mirror relationship .................................. 102
Scheduling SnapMirror transfers ............................................................. 102
Changing mirror relationship schedules ............................................... 103
Manually updating data protection mirror copies on destination volumes . 104
Deleting a mirror copy .......................................................................... 104
Considerations when breaking SnapMirror relationships ..................... 105
Reversing the data protection mirror relationship when disaster occurs .... 106
Reversing the version-flexible SnapMirror relationship when disaster
    occurs .................................................................................................. 109
Converting a data protection mirror destination to a writeable volume .... 113
Testing database applications ................................................................. 113

Providing disaster recovery for SVMs .................................................... 115
What a destination SVM is ....................................................................... 115
Deciding whether to replicate SVM network configuration ...................... 116
Configurations replicated in an SVM disaster recovery relationship ........ 116
How volumes and LIFs are placed on the destination SVM .................... 119
Configuring disaster recovery for SVMs .................................................. 120
Managing SVM disaster recovery relationships ....................................... 122
Monitoring the SVM disaster recovery relationship status .................... 122
Providing read-only access from the destination SVM .......................... 124
Protecting data on FlexVol volumes by using SnapVault backups

Creating SnapVault backups on FlexVol volumes
Guidelines for creating SnapVault relationships on FlexVol volumes
SnapVault updates fail if destination aggregate is full
Prepopulated SnapVault secondary scenarios
Creating a SnapVault backup in an empty FlexVol volume
Creating the SnapVault relationship of a mirror-SnapVault cascade
Preserving a Snapshot copy on the primary source volume
Creating a SnapVault backup in a prepopulated FlexVol volume
Creating a destination baseline using a tape backup
Converting a SnapVault relationship to a version flexible SnapMirror relationship
Converting a data protection destination to a SnapVault secondary

Managing backup and restore operations for SnapVault backups

Backing up from a Snapshot copy that is older than the base Snapshot copy
Back up FlexVol volumes that contain the maximum limit of Snapshot copies
Managing the backup of a copied source volume
Guidelines for restoring the active file system
Guidelines for restoring LUNs in SAN environments
How restore operations work from a SnapVault backup
Restoring a volume from a SnapVault backup
Restoring a single file or LUN
Managing a SnapVault-mirror cascade when the SnapVault backup is unavailable
Managing storage efficiency for SnapVault secondaries
Guidelines for managing storage efficiency for SnapVault backups
Enabling storage efficiency on a SnapVault secondary volume
Reenabling storage efficiency on a SnapVault destination volume

Data mirroring using SyncMirror

Mirroring data using the SyncMirror feature
Advantages of using SyncMirror
How mirrored aggregates work
Requirements for using SyncMirror with disks
How SyncMirror works with array LUNs
Requirements for setting up SyncMirror with array LUNs
SyncMirror pool assignment planning for array LUNs
Commands for creating and mirroring an array LUN aggregate
Common errors when setting up SyncMirror pools with array LUNs
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Troubleshooting errors with SyncMirror pool assignment for array LUNs</td>
<td>161</td>
</tr>
<tr>
<td>Requirements when using mirrored aggregates</td>
<td>161</td>
</tr>
<tr>
<td>Commands for managing mirrored aggregates</td>
<td>162</td>
</tr>
<tr>
<td>Creating a mirrored aggregate</td>
<td>162</td>
</tr>
<tr>
<td>Converting an aggregate to a mirrored aggregate</td>
<td>166</td>
</tr>
<tr>
<td>Converting a mirrored aggregate to an aggregate</td>
<td>167</td>
</tr>
<tr>
<td>Re-creating a mirrored aggregate after a plex failure</td>
<td>167</td>
</tr>
<tr>
<td>How disks are assigned to plexes</td>
<td>168</td>
</tr>
<tr>
<td>The states of a plex</td>
<td>168</td>
</tr>
<tr>
<td>Addition of disks or array LUNs to a mirrored aggregate</td>
<td>168</td>
</tr>
<tr>
<td>Rules for adding disks to a mirrored aggregate</td>
<td>169</td>
</tr>
<tr>
<td>Rules for adding array LUNs to a mirrored aggregate</td>
<td>169</td>
</tr>
<tr>
<td>Increasing the size of an aggregate that uses unpartitioned drives</td>
<td>169</td>
</tr>
</tbody>
</table>

**Copyright information** ............................................................................. 172

**Trademark information** ........................................................................... 173

**How to send comments about documentation and receive update notifications** ..................................................................................................................... 174

**Index** ........................................................................................................... 175
Deciding whether to use this guide

This guide describes features and methods available for data protection with clustered Data ONTAP, including Snapshot copies, SnapMirror policies, peer relationships, mirroring technology, SVM disaster recovery, SnapVault backups, and SyncMirror. It includes a conceptual overview, planning guidance, and detailed implementation instructions.

You should use this guide if you want to design and implement a data protection solution under the following circumstances:

• You are using the Data ONTAP command-line interface to implement data protection features.
• You want to provide data protection using online methods, not tape.
• You want to understand the range of Data ONTAP backup and recovery options.
• You want conceptual background for the features and methods.

If you want to create a basic configuration using best practices and you do not want a lot of conceptual background, you should choose among the following documentation:

• Clustered Data ONTAP 8.3 Cluster Peering Express Guide
• Data ONTAP 8.3 SVM Disaster Recovery Express Guide
• Data ONTAP 8.3 SVM Disaster Recovery Preparation Express Guide
• Clustered Data ONTAP 8.3 SVM Root Volume Protection Express Guide
• Clustered Data ONTAP 8.3 Volume Disaster Recovery Preparation Express Guide
• Clustered Data ONTAP 8.3 Volume Disaster Recovery Express Guide
• Clustered Data ONTAP 8.3 Volume Backup Using SnapVault Express Guide
• Clustered Data ONTAP 8.3 Volume Restore Using SnapVault Express Guide

If you require additional configuration or conceptual information, you should choose among the following documentation:

• Synchronous disaster recovery in a MetroCluster configuration
  ◦ Clustered Data ONTAP 8.3 MetroCluster Management and Disaster Recovery Guide
• Data protection by using tape technology
  ◦ Clustered Data ONTAP 8.3 NDMP Configuration Express Guide
  ◦ Clustered Data ONTAP 8.3 Data Protection Tape Backup and Recovery Guide
• Automation of management tasks
  ◦ NetApp Documentation: OnCommand Workflow Automation (current releases)
• Technical Reports (TRs), which include additional information about Data ONTAP technology and interaction with external services.
Introduction to data protection

Data protection means backing up data and being able to recover it. You protect the data by making copies of it so that it is available for restoration even if the original is no longer available.

Businesses need data backup and protection for the following reasons:

• To protect data from accidentally deleted files, application crashes, data corruption, and viruses
• To archive data for future use
• To recover from a disaster

Methods of protecting data

Depending on your data protection and backup needs, Data ONTAP offers a variety of features and methods that enable you to protect data against accidental, malicious, or disaster-induced loss of data.

Snapshot copies

Enable you to manually or automatically create, schedule, and maintain multiple backups (also called Snapshot copies) of data on a volume. Snapshot copies use only a minimal amount of additional volume space, and do not have a performance cost.

If a user accidentally modifies or deletes crucial data on a volume with Snapshot technology enabled, that data can be easily and quickly restored from one of the latest Snapshot copies created. You can also create clones of FlexVol volumes using Snapshot copies.

This method is valid for FlexVol volumes and Infinite Volumes.

SnapRestore (license required)

Enables you to perform fast, space-efficient, on-request Snapshot recovery from Snapshot copies on an entire volume.

This method is valid for FlexVol volumes and Infinite Volumes.

Data protection mirror copies (SnapMirror license required)

Provide asynchronous disaster recovery. Data protection mirror relationships enable you to periodically create Snapshot copies of data on one volume; copy those Snapshot copies to a partner volume (the destination volume), usually on another cluster; and retain those Snapshot copies. The mirror copy on the destination volume ensures quick availability and restoration of data from the time of the latest Snapshot copy, if the data on the source volume is corrupted or lost.

If you conduct tape backup and archival operations, you can perform them on the data that is already backed up on the destination volume.

This method is valid for FlexVol volumes, Infinite Volumes, and SVMs with FlexVol volumes. For SVMs with FlexVol volumes, all of the SVM data and configuration is replicated.

SnapVault backups (SnapVault license required)

Provide storage-efficient and long-term retention of backups. SnapVault relationships enable you to back up selected Snapshot copies of volumes to a destination volume and retain the backups.

If you conduct tape backup and archival operations, you can perform them on the data that is already backed up on the SnapVault secondary volume.
This method is valid only for FlexVol volumes.

**volume copy**

Enables you to perform fast block-copy of data from one volume to another.

This method is valid only for FlexVol volumes.

**nvfail option to the volume modify command**

Provides protection against data corruption by failures of nonvolatile RAM (NVRAM).

This method is valid for FlexVol volumes and Infinite Volumes.

**MetroCluster**

Stretch MetroCluster configurations provide site protection within a campus and support replication up to 500m.

Fabric MetroCluster configurations provide site protection within a metro, and support replication up to 100 km using FC switches. SyncMirror functionality is enhanced to provide continuous volume mirroring.

This method is valid only for FlexVol volumes.

---

**Monitoring and protecting database validity by using NVFAIL**

The `-nvfail` parameter of the `volume modify` command enables Data ONTAP to detect nonvolatile RAM (NVRAM) inconsistencies when the system is booting or after a switchover operation. It also warns you and protects the system against data access and modification until the volume can be manually recovered.

If Data ONTAP detects any problems, database or file system instances stop responding or shut down. Data ONTAP then sends error messages to the console to alert you to check the state of the database or file system. You can enable NVFAIL to warn database administrators of NVRAM inconsistencies among clustered nodes that can compromise database validity.

After a system crash or switchover operation, NFS clients cannot access data from any of the nodes until the NVFAIL state is cleared. CIFS clients are unaffected.

---

**What a data loss disaster is**

A data loss disaster is a situation in which service from one physical site (for example, a building or a corporate campus) on the network is lost for an extended period of time.

The following are examples of disasters:

- Fire
- Earthquake
- Prolonged power outages at a site
- Prolonged loss of connectivity from clients to the storage system at a site

When a disaster occurs, it can affect all the computing infrastructure including storage systems, application servers, networking connectivity, and client connectivity. When you create a disaster plan, you should take your computing infrastructure into consideration.
Tools for protecting against data-loss disasters

Data ONTAP provides tools that enable you to back up or replicate data stored at a primary data storage site to an off-site network location. This ensures that you can restore data if data loss is caused by disaster at a primary data storage site.

SnapVault backups for FlexVol volumes

SnapVault is a Snapshot copy backup and restorability tool on FlexVol volumes. You can locate a SnapVault secondary volume on the same cluster or on a different cluster.

Data recoverability

If a data-loss disaster occurs at a source volume, you can restore data that is backed up to a SnapVault secondary volume. You can restore the data to the source volume after it is running again, or you can restore data to an alternate volume.

Currency of restore data

You can restore data from any Snapshot copy that was replicated to the destination system.

Advantage

A SnapVault backup provides an inexpensive backup solution.

Data protection mirror copy

A data protection mirror copy is a Snapshot copy replication, availability, and restorability tool. You can locate a data protection mirror destination on the same cluster or on a different cluster.

Data availability

If a source site experiences a data-loss disaster, you can quickly make available data at the data protection mirror copy destination site.

Data recoverability

If a data-loss disaster occurs at a source storage site, you can restore data from a data protection mirror copy destination volume. You can restore the data to the source volume after it is running again, or you can restore data to an alternate volume.

Currency of restore data

You can restore data from the last Snapshot copy that was replicated to the destination volume.

Advantage

Data protection mirror copies provide data protection and availability.

MetroCluster configurations for FlexVol volumes

MetroCluster configurations provide inter-site, real-time backup, availability, and restorability. You can locate synchronously mirrored MetroCluster systems at different sites, up to 100 km from one another.

Data availability

If a storage site experiences a data-loss disaster, you can quickly make available data that is mirrored to the partner site.

Data recoverability

You can restore data from a mirrored partner site to the source storage system after it is running again, or you can restore data to a system at an alternate location.

Currency of restore data

You can restore data from the time of the last NVRAM checkpoint.
Connection requirements

Data ONTAP cluster connections supplemented with switches and DSL or faster connections are required.

You should preconfigure routers, switches, and DNS servers to direct users to the MetroCluster partner if the clustered system that they first attempt to access is unavailable.

For more information about MetroCluster configurations, see the *Clustered Data ONTAP 8.3 MetroCluster Installation and Configuration Guide*.

Advantage

MetroCluster configurations provide real-time, off-site protection and availability.

Data protection in a SAN environment

If FlexVol volumes contain logical units of storage (LUNs) created to enable integration into a storage area network (SAN) environment, the procedures to implement data protection might have to be modified. Infinite Volumes do not support SAN environments or LUNs.

Data protection mirror copies and SnapVault backups are achieved by the use of volume-to-volume relationships. Therefore, to protect data in a LUN, you back up the volume that contains the LUN.

Path-related metadata such as Persistent Reservations, are not replicated to a SnapVault backup. When you restore a volume from a SnapVault secondary volume, the LUNs in the SnapVault secondary volume are exported with a different identity from their counterparts in the source volume. Therefore, you must configure new access controls for the restored LUNs.

For more information about the descriptions of data backup and restore on volumes containing Data ONTAP LUNs, see the *Clustered Data ONTAP SAN Administration Guide*.

Types of data protection policies

You can assign Snapshot policies to FlexVol volumes and Infinite Volumes, and SnapMirror policies to data protection mirror relationships and SnapVault relationships.

Snapshot policy

When you assign a Snapshot policy, the policy configures the Snapshot copy creation schedule and retention rules.

You can assign the same Snapshot policy to multiple volumes. For example, you might configure a Snapshot policy to create a Snapshot copy every hour, at the end of every day, and at the end of every week, and then assign that same policy to more than one volume.

You can assign only one Snapshot policy to a volume. You can assign Snapshot policies to FlexVol volumes and Infinite Volumes.

**Note:** You cannot assign a Snapshot policy that contains the `-snapmirror-label` to an Infinite Volume.

SnapMirror policy

The SnapMirror policy specifies the configuration attributes of a relationship.

A SnapMirror policy can be applied to a data protection mirror relationship or a SnapVault relationship. Whether the SnapMirror policy has rules determines if the policy is applied to a SnapVault relationship or applied to a data protection mirror copy. If the policy has rules that define which Snapshot copies are protected, then that policy can be applied to SnapVault relationships only. If the policy does not have rules, then that policy can be applied to data protection mirror copies only.
Note: If no policy is assigned to a relationship, a default policy is assigned. If it is a data protection mirror relationship, the DPDefault policy is assigned. If it is a SnapVault relationship, the XDPDefault policy is assigned.
Planning your data protection strategy

Data ONTAP provides a variety of tools that you can use to build a comprehensive strategy to protect your company’s data.

Storage Virtual Machine (SVM) administrators can plan data protection for FlexVol volumes and Infinite volumes within their assigned SVMs. Cluster administrators can plan data protection for FlexVol volumes and Infinite Volumes within their assigned clusters.

Working with Snapshot copies

Snapshot copies are the first line of defense for data protection. Data ONTAP maintains a configurable Snapshot schedule that creates and deletes Snapshot copies automatically for each FlexVol volume and Infinite Volume. You can also create and delete Snapshot copies, and manage Snapshot schedules based on your requirements.

What a Snapshot copy is

A Snapshot copy is a read-only image of a FlexVol volume or Infinite Volume that captures the state of the file system at a point in time.

For information about FlexVol volumes, see the *Clustered Data ONTAP Physical Storage Management Guide*.

User access to Snapshot copies

A Snapshot copy is a copy of a FlexVol volume that represents the volume's contents at a particular point in time. You can view the contents of the Snapshot copy and use the Snapshot copy to restore data that you lost recently.

A Snapshot copy of a volume is located on the parent volume but has read-only access. It represents the contents of the original volume at a particular point in time. A parent volume and a Snapshot copy of it share disk space for all blocks that have not been modified between the creation of the volume and the time the Snapshot copy is made, thereby making Snapshot copies lightweight.

Similarly, two Snapshot copies share disk space for those blocks that were not modified between the times that the two Snapshot copies were created. You can create a chain of Snapshot copies to represent the state of a volume at a number of points in time. Users can access Snapshot copies online, enabling users to retrieve their own data from past copies, rather than asking a system administrator to restore data from tape. Administrators can restore the contents of a volume from a Snapshot copy.

Each volume has a `.snapshot` directory that is accessible to NFS users by using the `ls` command and to CIFS users by double-clicking the `~snapshot` folder. The contents of the `.snapshot` directory are a set of subdirectories, labeled by type, date, and time, resembling the following:

```
$ ls .snapshot
daily.2006-05-14_0013/        hourly.2006-05-14_1306/
hourly.2006-05-15_1106/       weekly.2006-05-14_0019/
hourly.2006-05-15_1206/
```

Each subdirectory of the `.snapshot` directory includes a list of the parent volume's files and directories. If users accidentally delete or overwrite a file, they can locate it in the most recent Snapshot directory and restore it to their main read-write volume simply by copying it back to the
main directory. The following example shows how an NFS user can locate and retrieve a file named my.txt from the .snapshot directory:

```bash
$ ls my.txt
ls: my.txt: No such file or directory
$ ls .snapshot
daily.2006-05-14_0013/ hourly.2006-05-15_1306/
hourly.2006-05-15_1106/ weekly.2006-05-14_0019/
hourly.2006-05-15_1206/
$ ls .snapshot/hourly.2006-05-15_1506/my.txt
my.txt
$ ls my.txt
my.txt
```

The .snapshot directory is always visible to NFSv2 and NFSv3 clients and available from within the volume, and not visible but still available from any other volume. For NFSv4 clients, the .snapshot directory is not visible, but accessible in all paths of a volume.

**Backup and recovery tasks you can perform with Snapshot copies**

Snapshot copies enable system administrators and end users to perform important tasks in backup and recovery.

Snapshot copies enable system administrators to perform the following tasks:

- Create instantaneous backups
- Create a clone of a FlexVol volume
- Create a clone of a Data ONTAP LUN

For information about cloning a FlexVol volume, see the *Clustered Data ONTAP Logical Storage Management Guide*.

Snapshot copies enable end users to perform the following tasks:

- Recover older versions or sets of files that were accidentally changed or deleted
- Restore their own files without needing a system administrator to restore files from tape

**Maximum number of Snapshot copies**

You should know what the maximum number of Snapshot copies you can accumulate is to minimize the possibility that you do not have Snapshot copies available when you need them.

The maximum number of Snapshot copies follows:

- You can accumulate a maximum of 255 Snapshot copies of a FlexVol volume.
- If the FlexVol volume is in a data protection mirror relationship, the maximum number of Snapshot copies is 254 because one Snapshot copy is reserved for use by the relationship during recovery operations.
- If the FlexVol volume is in a disk to disk backup relationship, the maximum number of Snapshot copies is 251.
- If the Infinite Volume is in a data protection mirror relationship, the maximum number of Snapshot copies is reduced by two because two Snapshot copies are used for the data protection mirror relationship.
Over time, automatically generated hourly, weekly, and monthly Snapshot copies accrue. Having a number of Snapshot copies available gives you a greater degree of accuracy if you have to restore a file.

The number of Snapshot copies can approach the maximum if you do not remove older Snapshot copies. You can configure Data ONTAP to automatically delete older Snapshot copies of volumes as the number of Snapshot copies approaches the maximum.

The following data protection mirror copies affect the maximum number of Snapshot copies available to a volume:

- A FlexVol volume in a data protection mirror relationship
- A FlexVol volume with a load-sharing mirror copy
- An Infinite Volume with one or more namespace mirror constituents
  Each namespace mirror constituent uses two Snapshot copies. By default, a read/write Infinite Volume contains one namespace mirror constituent. If you enable SnapDiff on an Infinite Volume, each additional namespace mirror uses two Snapshot copies.

An Infinite Volume also uses up to four Snapshot copies when technical support runs some commands that require diagnostic privilege. You must keep the number of Snapshot copies far enough below the limit to ensure that technical support can run commands.

Where to find information about Snapshot copies of Infinite Volumes

Information about Snapshot copies of Infinite Volumes is available in the *Clustered Data ONTAP Infinite Volumes Management Guide*.

Creation of Snapshot copy schedules

Data ONTAP provides a default Snapshot copy schedule for each FlexVol volume and Infinite Volume. You can create schedules to fit your needs if the default Snapshot copy schedule is not adequate.

For FlexVol volumes, the default Snapshot copy schedule automatically creates one daily Snapshot copy Monday through Saturday at midnight, an hourly Snapshot copy five minutes past the hour, every hour, and a weekly Snapshot copy. Data ONTAP retains the two most recent nightly Snapshot copies and the six most recent hourly Snapshot copies, and deletes the oldest nightly and hourly Snapshot copies when new Snapshot copies are created.

Types of user-specified Snapshot copy schedules

Data ONTAP contains weekly, daily, and hourly Snapshot copy schedules that you can use to create Snapshot copy policies that retain the number and type of Snapshot copies you want.

The following table describes the available types of Snapshot copy schedules:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly</td>
<td>Data ONTAP creates these Snapshot copies every Sunday at 15 minutes after midnight. Weekly Snapshot copies are named \texttt{weekly.n}, where \texttt{n} is the date in year-month-day format followed by an underscore (_) and the time. For example, a weekly Snapshot copy created on 25 November 2012 is named \texttt{weekly.2012-11-25_0015}.</td>
</tr>
</tbody>
</table>
Type | Description
--- | ---
Daily | Data ONTAP creates these Snapshot copies every night at 10 minutes after midnight. Daily Snapshot copies are named daily.\(n\), where \(n\) is the date in year-month-day format followed by an underscore (\(\_\)) and the time. For example, a daily Snapshot copy created on 4 December 2012 is named daily.2012-12-04_0010.
Hourly | Data ONTAP creates these Snapshot copies five minutes after the hour. Hourly Snapshot copies are named hourly.\(n\), where \(n\) is the date in year-month-day format followed by an underscore (\(\_\)) and the time. For example, an hourly Snapshot copy created on 4 December 2012 at 1:00 (1300) is named hourly.2012-12-04_1305.

Related concepts

*Commands for managing Snapshot policies and schedules* on page 40

Creating a Snapshot copy schedule

If the default Snapshot copy schedule does not meet your needs, you can create a schedule that does.

**Step**

1. Create a Snapshot copy schedule by using the `job schedule cron create` command or the `job schedule interval create` command.

   The command you use depends on how you want to implement the schedule. See the man page for each command to determine the command that meets your needs.

If scheduled Snapshot copy creation fails

Scheduled Snapshot copy creation might fail for various reasons, such as a volume being unavailable. In such cases, Data ONTAP attempts to create a Snapshot copy, when possible, outside the schedule.

If a scheduled Snapshot copy creation fails, Data ONTAP checks the Snapshot copies present in the volume. The checks performed and the actions taken depend on the type of scheduled Snapshot copy creation that failed. The process is described in the following list:

1. When a volume becomes available again for creating a Snapshot copy, Data ONTAP checks whether any Snapshot copies were created during a time period represented by `period_snap`. `period_snap` is a variable representing a time period that depends on the type of Snapshot copy schedule, as shown in the following table:

<table>
<thead>
<tr>
<th>Type of Snapshot copy schedule</th>
<th>Value of the <code>period_snap</code> variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly</td>
<td>3 days</td>
</tr>
<tr>
<td>Nightly</td>
<td>3 days</td>
</tr>
<tr>
<td>Hourly</td>
<td>12 hours</td>
</tr>
</tbody>
</table>

   **Note:** You cannot change the value of `period_snap`.

2. The check in the previous step returns one of the following values:
If the check returns... | Then...
---|---
Yes (One or more Snapshot copies were created in the *period_snap* period) | Data ONTAP performs Step 3.
No (Snapshot copies were not created in the *period_snap* period) | Data ONTAP performs Step 4.

3. Data ONTAP checks whether any scheduled Snapshot copy creations failed after the most recent Snapshot copy. This check returns one of the following values:

| If the check returns... | Then...
---|---
Yes (One or more scheduled Snapshot copy creations were missed) | Data ONTAP creates a Snapshot copy.
No (No scheduled Snapshot copy creation was missed) | Data ONTAP does not create a Snapshot copy.

4. Data ONTAP checks whether any scheduled Snapshot copy creation have failed in the past 25 minutes. This check returns one of the following values:

| If the check returns... | Then...
---|---
Yes (A scheduled Snapshot copy creation was missed in the past 25 minutes) | Data ONTAP creates a Snapshot copy.
No (No scheduled Snapshot copy creation was missed in the past 25 minutes) | Data ONTAP does not create a Snapshot copy.

**Deleting Snapshot copies automatically**

You can define and enable a policy for automatically deleting Snapshot copies and FlexClone LUNs. Automatically deleting Snapshot copies and FlexClone LUNs can help you manage space utilization.

**About this task**

You can automatically delete Snapshot copies from read-write volumes and FlexClone LUNs from read-write parent volumes. You cannot set up automatic deletion of Snapshot copies from Infinite Volumes or from read-only volumes, for example, SnapMirror destination volumes.

**Step**

1. Define and enable a policy for automatically deleting Snapshot copies by using the `volume snapshot autodelete modify` command.

   See the `volume snapshot autodelete modify` man page for information about the parameters that you can use with this command to define a policy that meets your needs.

**Example**

The following command enables the automatic deletion of Snapshot copies and sets the trigger to `snap_reserve` for the vol3 volume, which is part of the vs0.example.com Storage Virtual Machine (SVM):

```
cluster1::> volume snapshot autodelete modify -vserver vs0.example.com 
-volume vol3 -enabled true -trigger snap_reserve
```
Example
The following command enables the automatic deletion of Snapshot copies and of FlexClone LUNs marked for autodeletion for the vol3 volume, which is part of the vs0.example.com Storage Virtual Machine (SVM):

```
cluster1::> volume snapshot autodelete modify -vserver vs0.example.com -volume vol3 -enabled true -trigger volume -commitment try -delete-order oldest_first -destroy-list lun_clone,file_clone
```

Viewing settings for the automatic deletion of Snapshot copies
You can view the settings for the automatic deletion of Snapshot copies to help you when you are deciding if the settings are meeting your needs.

Step
1. View the settings for the automatic deletion of Snapshot copies by using the `volume snapshot autodelete show` command.

See the `volume snapshot autodelete show` command man pages for information about parameters shown by this command.

Example
The following command displays the automatic deletion settings of Snapshot copies for the vol3 volume, which is part of the vs0.example.com Storage Virtual Machine (SVM):

```
cluster1::> volume snapshot autodelete show -vserver vs0 -volume vol3
```

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Volume</th>
<th>Option Name</th>
<th>Option Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs0</td>
<td>vol3</td>
<td>Enabled</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Commitment</td>
<td>try</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trigger</td>
<td>volume</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Target Free Space</td>
<td>20%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delete Order</td>
<td>oldest_first</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defer Delete</td>
<td>user_created</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defer Delete Prefix</td>
<td>(not specified)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Destroy List</td>
<td>none</td>
</tr>
</tbody>
</table>

What Snapshot disk consumption is
Data ONTAP preserves pointers to all the disk blocks currently in use at the time the Snapshot copy is created. When a file is changed, the Snapshot copy still points to the disk blocks where the file existed before it was modified, and changes are written to new disk blocks.

How Snapshot copies consume disk space
Snapshot copies minimize disk consumption by preserving individual blocks rather than whole files. Snapshot copies begin to consume extra space only when files in the active file system are changed or deleted. When this happens, the original file blocks are still preserved as part of one or more Snapshot copies.

In the active file system the changed blocks are rewritten to different locations on the disk or removed as active file blocks entirely. As a result, in addition to the disk space used by blocks in the modified active file system, disk space used by the original blocks is still reserved to reflect the status of the active file system before the change.

The following illustration shows disk space usage for a Snapshot copy:
How changing file content consumes disk space

A given file might be part of a Snapshot copy. The changes to such a file are written to new blocks. Therefore, the blocks within the Snapshot copy and the new (changed or added) blocks both use space within the volume.

Changing the contents of the `myfile.txt` file creates a situation where the new data written to `myfile.txt` cannot be stored in the same disk blocks as the current contents because the Snapshot copy is using those disk blocks to store the old version of `myfile.txt`. Instead, the new data is written to new disk blocks. As the following illustration shows, there are now two separate copies of `myfile.txt` on disk—a new copy in the active file system and an old one in the Snapshot copy:
What the Snapshot copy reserve is

The Snapshot copy reserve sets a specific percent of the disk space for Snapshot copies. For FlexVol volumes, the default Snapshot copy reserve is set to 5 percent of the disk space. By default, the Snapshot copy reserve is 5 percent of the disk space for a FlexVol volume and 0 percent for aggregates.

The active file system cannot consume the Snapshot copy reserve space, but the Snapshot copy reserve, if exhausted, can use space in the active file system.

How Data ONTAP uses deleted active file disk space

When enough disk space is available for Snapshot copies in the Snapshot copy reserve, deleting files in the active file system frees disk space for new files, while the Snapshot copies that reference those files consume only the space in the Snapshot copy reserve.

If Data ONTAP created a Snapshot copy when the disks were full, deleting files from the active file system does not create any free space because everything in the active file system is also referenced by the newly created Snapshot copy. Data ONTAP has to delete the Snapshot copy before it can create any new files.

The following example shows how disk space being freed by deleting files in the active file system ends up in the Snapshot copy:

If Data ONTAP creates a Snapshot copy when the active file system is full and there is still space remaining in the Snapshot reserve, the output from the `df` command—which displays statistics about the amount of disk space on a volume—is as follows:

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>kbytes</th>
<th>used</th>
<th>avail</th>
<th>capacity</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vserver</td>
<td>3000000</td>
<td>300000</td>
<td>0</td>
<td>100%</td>
<td>vs1</td>
</tr>
<tr>
<td>/vol/vol0/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vs1</td>
</tr>
<tr>
<td>vsl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/vol/vol0/.snapshot</td>
<td>1000000</td>
<td>500000</td>
<td>500000</td>
<td>50%</td>
<td>vs1</td>
</tr>
<tr>
<td>vsl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you delete 100,000 KB (0.1 GB) of files, the disk space used by these files is no longer part of the active file system, so the space is reassigned to the Snapshot copies instead.

Data ONTAP reassigns 100,000 KB (0.1 GB) of space from the active file system to the Snapshot reserve. Because there was reserve space for Snapshot copies, deleting files from the active file system freed space for new files. If you enter the `df` command again, the output is as follows:

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>kbytes</th>
<th>used</th>
<th>avail</th>
<th>capacity</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vserver</td>
<td>3002000</td>
<td>2902000</td>
<td>100000</td>
<td>97%</td>
<td>vs1</td>
</tr>
<tr>
<td>/vol/vol0/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>vs1</td>
</tr>
<tr>
<td>--</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/vol/vol0/.snapshot</td>
<td>1002000</td>
<td>6002000</td>
<td>400000</td>
<td>60%</td>
<td>vs1</td>
</tr>
<tr>
<td>--</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example of what happens when Snapshot copies exceed the reserve

Because there is no way to prevent Snapshot copies from consuming disk space greater than the amount reserved for them, it is important to reserve enough disk space for Snapshot copies so that the active file system always has space available to create new files or modify existing ones.

Consider what happens in the following example if all files in the active file system are deleted. Before the deletion, the `node run -node nodename df` output is as follows:
After the deletion, the `node run -node nodename df` command generates the following output:

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>kbytes</th>
<th>used</th>
<th>avail</th>
<th>capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vol/vol0/</td>
<td>3000000</td>
<td>2500000</td>
<td>500000</td>
<td>83%</td>
</tr>
<tr>
<td>/vol/vol0/.snapshot</td>
<td>1000000</td>
<td>3500000</td>
<td>0</td>
<td>350%</td>
</tr>
</tbody>
</table>

The output shows that the entire 3,000,000 KB (3 GB) in the active file system is still being used by Snapshot copies in addition to the 500,000 KB (0.5 GB) that was used by Snapshot copies before the deletion. Therefore, a total of 3,500,000 KB (3.5 GB) is being used by Snapshot copy data, which is 2,500,000 KB (2.5 GB) more than the space reserved for Snapshot copies. This means that 2.5 GB of space that would be available to the active file system is now unavailable to it. The post-deletion output of the `node run -node nodename df` command lists this unavailable space as used even though no files are stored in the active file system.

**Recovery of disk space for file system use**

Whenever Snapshot copies consume more than 100% of the Snapshot reserve, they begin to occupy the active file system space. This process is called Snapshot spill. When the Snapshot copies continue to occupy the active file system space, the system is in danger of becoming full. If the system becomes full due to Snapshot spill, you can create files only after you delete enough Snapshot copies.

If 500,000 KB (0.5 GB) of data is added to the active file system, a `node run -node nodename df` command generates the following output:

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>kbytes</th>
<th>used</th>
<th>avail</th>
<th>capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vol/vol0/</td>
<td>3000000</td>
<td>3000000</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>/vol/vol0/.snapshot</td>
<td>1000000</td>
<td>3500000</td>
<td>0</td>
<td>350%</td>
</tr>
</tbody>
</table>

As soon as Data ONTAP creates a new Snapshot copy, every disk block in the file system is referenced by some Snapshot copy. Therefore, no matter how many files you delete from the active file system, there is still no room to add any more. The only way to recover from this situation is to delete enough Snapshot copies to free more disk space.

**Working with mirroring technology**

Before using mirroring technology, you should understand the components of a mirror relationship, types of mirror copies, where mirror copies are located, path naming and language requirements, and what mirror relationships are not intended to do.

**Components of a mirror relationship**

In its simplest configuration, a mirror relationship is between a source volume and a destination volume and data is replicated to the destination volume using Snapshot copies.

Typically, the source volume is a read-write volume that clients can access and modify. The destination volume is a read-only volume that exports a Snapshot copy to clients for read-only access. The only time the source volume is not a read-write volume is in a cascade configuration
where the source volume is a destination of one mirror relationship and the source of another mirror relationship.

Snapshot copies are used by the source volume to update destination volumes. Snapshot copies are transferred from the source volume to the destination volume using an automated schedule or manually; therefore, mirror copies are updated asynchronously. You use the set of `snapmirror` commands to create and manage mirror relationships.

**Related concepts**

*Supported data protection deployment configurations* on page 32

### Data protection mirror relationships for FlexVol volumes

You can create a mirror relationship to a destination within a cluster to protect your data or, for greater disaster protection, you can create a mirror relationship to a destination in a different cluster in a different location.

A data protection mirror configuration consists of a source volume that can be replicated to one or more destination volumes. Each data protection mirror relationship is independent from the other.

**Note:** The version of Data ONTAP that is running on the destination volume must be the same or a later version than the one running on the source volume.

You can create data protection mirror relationships to destinations on the same aggregate as the source volume, and on the same Storage Virtual Machine (SVM) or on a different SVM. For greater protection, you can create the relationships to destinations on a different aggregate, which enables you to recover from the failure of the source volume's aggregate. Neither of these two configurations protects against a cluster failure.

To protect against a cluster failure, you can create a data protection mirror relationship in which the source volume is on one cluster and the destination volume is on a different cluster. If the cluster on which the source volume resides experiences a disaster, you can direct user clients to the destination volume on the cluster peer until the source volume is available again.

You can also use mirror relationships for limited disaster recovery, off-loading tape backup, data distribution, and making offline copies of production data for research, such as data mining.

### Data protection mirror relationships for SVMs with FlexVol volumes

The cluster administrator can protect the SVM data volumes and configuration by creating a data protection mirror copy of the data volumes and configuration on a destination SVM on the destination cluster.

You can set up only intercluster data protection mirror relationships for SVMs with FlexVol volumes. When the source SVM is unavailable, the cluster administrator of the destination cluster can start the destination SVM to provide data access, with minimum disruption.

When the source SVM is available, the cluster administrator of the source cluster can resynchronize the data and configuration from the destination SVM to serve data from the source SVM.

You cannot create fan-in, fan-out, and cascade chain SVM disaster recovery relationships. You can create only one SVM disaster recovery relationship for a source SVM and use one destination SVM for protecting only one source SVM.

The destination SVM cannot be a source of any other SVM disaster recovery relationship. However, the volumes in the destination SVM can be the source of SnapVault backup or volume-level SnapMirror relationships.

The SnapMirror relationship between the SVMs is also referred to as SVM disaster recovery relationship.
Related concepts

Providing disaster recovery for SVMs on page 115

Where to find information about data protection mirror relationships for Infinite Volumes

Information about creating and managing data protection mirror relationships for Infinite Volumes and recovering Infinite Volumes is available in the Clustered Data ONTAP Infinite Volumes Management Guide.

Unexpected destination volume growth during data protection mirror transfers

During a data protection mirror transfer, the destination volume grows as a normal part of the transfer. You do not need to be concerned if you observe volume growth during this time.

At the start of a data protection mirror transfer, the destination volume grows in size if the source volume has grown. This occurs irrespective of any automatic growth setting on the destination volume. You cannot prevent Data ONTAP from growing or limiting its growth.

Path name pattern matching

You can use pattern matching when you use snapmirror commands to have the command work on selected mirroring relationships.

The snapmirror commands use fully qualified path names in the following format: vserver:volume. You can abbreviate the path name by not entering the Storage Virtual Machine (SVM) name. If you do this, the snapmirror command assumes the local SVM context of the user.

Assuming that the SVM is called “vserver1” and the volume is called “vol1”, the fully qualified path name is vserver1:vol1.

You can use the asterisk (*) in paths as a wildcard to select matching, fully qualified path names. The following table provides examples of using the wildcard to select a range of volumes.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Matches all paths.</td>
</tr>
<tr>
<td>vs*</td>
<td>Matches all SVMs and volumes with SVM names beginning with vs.</td>
</tr>
<tr>
<td>*:<em>src</em></td>
<td>Matches all SVMs with volume names containing the src text.</td>
</tr>
<tr>
<td>*:<em>vol</em></td>
<td>Matches all SVMs with volume names beginning with vol.</td>
</tr>
</tbody>
</table>

![Snapmirror show output example](image)

Language setting requirement

The source and destination FlexVol volumes or Infinite Volumes of a mirror relationship must have the same language setting; otherwise, NFS or CIFS clients might not be able to access data. For FlexVol volumes, it is not a problem if the source and destination volumes are located on the same Storage Virtual Machine (SVM) because the language is set on the SVM. For FlexVol volumes and Infinite Volumes with mirror relationships between volumes on two different SVMs, the language setting on the SVMs must be the same.
User access to destination volumes

Users have read-only access to the active file system on the destination FlexVol volume or Infinite Volume. The active file system on the destination volume is an exported Snapshot copy of the active file system from the source volume.

For information about user access to destination Infinite Volumes, see the Clustered Data ONTAP Infinite Volumes Management Guide.

When clients can access the active file system on the destination FlexVol volume

The active file system on a destination volume is available to clients after the system transfers Snapshot copies of the source volume to the destination volume. When the active file system is available differs between FlexVol volumes in mirror relationships and SnapVault relationships.

For a FlexVol volume in a mirror relationship, the storage system automatically directs clients to use the active file system in the latest Snapshot copy on the destination FlexVol volume. For FlexVol volumes that are secondary volumes of a SnapVault relationship, the active file system on the secondary volume is available after the baseline transfer. Attributes of the file system, such as the number of files or the amount of space consumed, are refreshed after the Snapshot copy for the volume is transferred.

Requirements for creating relationships between clusters or SVMs

Before you create a SnapMirror or SnapVault relationship between volumes in different clusters or Storage Virtual Machines (SVMs), you should ensure that the relationship follows the supported configurations.

Relationships between volumes in different clusters

Before you can create a relationship between volumes in different clusters, there must be a cluster peer relationship established between the two clusters and an SVM peer relationship established between the two SVMs.

SnapMirror relationships between clusters running different versions of Data ONTAP

The Data ONTAP versions supported depend on the relationship type and policy defined for that SnapMirror relationship.

For SnapMirror relationships using type DP and policy async-mirror, a relationship can be built from an earlier Data ONTAP release to a one that is no more than two releases later. The following relationships are permitted:

<table>
<thead>
<tr>
<th>If the source volume is located on a system running Data ONTAP...</th>
<th>The destination volume can reside on a system running one of the following releases...</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.0.x</td>
<td>8.1.x</td>
</tr>
<tr>
<td>8.0.x</td>
<td>yes</td>
</tr>
<tr>
<td>8.1.x</td>
<td>no</td>
</tr>
<tr>
<td>8.2.x</td>
<td>no</td>
</tr>
<tr>
<td>8.3.x</td>
<td>no</td>
</tr>
</tbody>
</table>

SnapMirror relationships using type XDP and policy mirror-vault, also known as version-flexible SnapMirror, are available with Data ONTAP 8.3. Such a relationship can be built only from source and destination volumes running an 8.3.x release.
Note: The version-flexible SnapMirror feature is not available before Data ONTAP 8.3; therefore, you cannot have a version-flexible SnapMirror relationship with a volume earlier than Data ONTAP 8.3.

The `snapmirror show` command displays mixed-version relationships in addition to same-version relationships.

For more information about SnapMirror relationships between clusters running different versions of Data ONTAP, see the *Clustered Data ONTAP Upgrade and Revert/Downgrade Guide*.

**SnapVault relationships between clusters running different versions of Data ONTAP**

For SnapVault relationships, the version of Data ONTAP that is running on the primary and secondary volumes must be Data ONTAP 8.2 or later. Starting with Data ONTAP 8.2, a SnapVault relationship can be built from an earlier Data ONTAP release to a one that is no more than two releases later. Also starting with 8.2, a SnapVault relationship can be built from a later release to a one that is no more than two releases earlier, providing that the earlier release is 8.2 or later.

The following SnapVault relationships are permitted using type `XDP` and policy `async-mirror`:

<table>
<thead>
<tr>
<th>If the primary volume is located on a system running Data ONTAP…</th>
<th>The secondary volume can reside on a system running one of the following releases…</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1.x</td>
<td>8.2.x</td>
</tr>
<tr>
<td>8.2.x</td>
<td>no</td>
</tr>
<tr>
<td>8.3.x</td>
<td>no</td>
</tr>
</tbody>
</table>

**Relationships between volumes in different SVMs**

Before you can create a relationship between volumes in different SVMs, there must be a peer relationship established between the two SVMs. You can only establish an SVM peer relationship between SVMs with unique names. You should use unique, fully qualified domain names for each SVM.

You can create a peer relationship between two SVMs with FlexVol volumes or between two SVMs with Infinite Volume, but you cannot create a peer relationship between an SVM with FlexVol volume and an SVM with Infinite Volume.

**Limitations for data protection mirror relationships**

When working with data-protection mirror relationships, you should be aware of the limitations.

The following limitations apply to data-protection mirror relationships:

- Snapshot copies cannot be deleted on destination volumes.
- An empty junction path on a destination FlexVol volume is not accessible from CIFS clients.
- A volume can have a maximum of 255 Snapshot copies.

**Supported number of destination volumes in fanout SnapMirror relationships**

When you are planning the number and types of SnapMirror relationships for a single source volume, you should remember that the source volume supports a certain number of destination volumes.

The number of destination volumes you can fan out depends on the type of SnapMirror relationship that you want to fan out from a single source volume:
For load-sharing mirror relationships, you can fan out a maximum of one destination volume on a node for a single source volume. The maximum number of nodes within a cluster depends on the platform model and licensed protocols. For details about cluster size limits, see the Hardware Universe at hwu.netapp.com.

For data protection mirror relationships, you can fan out a maximum of eight destination volumes from a single source volume.

A single source volume can have both one load-sharing destination volume on a node and eight data protection destination volumes.

**Snapshot copies cannot be deleted automatically on destination volumes**

You cannot automatically delete old Snapshot copies on destination FlexVol volumes or Infinite Volumes of mirror relationships because the destination volume is a read-only version of the source volume and should contain the same data as the source.

This is not true of Snapshot copies on destination FlexVol volumes of SnapVault relationships. You can delete old Snapshot copies on SnapVault secondary volumes.

**Note:** Using the `snap autodelete` command to automatically delete Snapshot copies from a destination volume to remove older Snapshot copies will fail.

**Empty junction path on a destination volume is not accessible from CIFS clients**

If internally mounted FlexVol volumes form a namespace and you have a mirror relationship, CIFS clients on a destination volume that attempt to view mirrored volumes not at the highest level of the namespace are denied access.

This occurs when you create a namespace using more than one volume, in which one volume is the source volume of a mirror relationship and the other volumes are members of the namespace. For example, assume that you have two volumes: vol x, which has a junction path `/x`, and vol y, which has a junction path `/x/y`. When a SnapMirror transfer occurs, a directory under vol x is created for vol y on the destination volume. From an NFS client, you can see that the directory is empty, but from a CIFS client, you get the following message:

```
access is denied.
```

**Maximum number of Snapshot copies for volumes that are mirrored**

The maximum number of Snapshot copies that a FlexVol volume in a mirror relationship can contain is 251. The maximum number of Snapshot copies that an Infinite Volume in a data protection mirror relationship can contain is 250.

For FlexVol volumes, whenever an update to a data protection mirror copy or set of load-sharing mirror copies occurs, Data ONTAP creates one new Snapshot copy. For Infinite Volumes, whenever an update to a data protection mirror copy occurs, Data ONTAP creates one new Snapshot copy. You should consider this as you manage the number of Snapshot copies on the source volume. You must keep the number of Snapshot copies far enough below the limit that updates to the mirror copy do not exceed the limit.

You should allow a maximum of 251 Snapshot copies for a FlexVol volume in a SnapVault relationship. By using a maximum of 251 Snapshot copies, you provide yourself a buffer in which you can manage your Snapshot copies before reaching the hard 255 Snapshot copy limit.

**Related concepts**

`Maximum number of Snapshot copies` on page 15
Working with FlexVol volume SnapVault backups

Before using SnapVault technology, you should understand how SnapVault backups work, where SnapVault volumes are located, and what SnapVault relationships are not intended to do.

What a SnapVault backup is

A SnapVault backup is a collection of Snapshot copies on a FlexVol volume that you can restore data from if the primary data is not usable. Snapshot copies are created based on a Snapshot policy. The SnapVault backup backs up Snapshot copies based on its schedule and SnapVault policy rules.

A SnapVault backup is a disk-to-disk backup solution that you can also use to offload tape backups. In the event of data loss or corruption on a system, backed-up data can be restored from the SnapVault secondary volume with less downtime and uncertainty than is associated with conventional tape backup and restore operations.

The following terms are used to describe SnapVault backups:

**baseline transfer**
An initial complete backup of a primary storage volume to a corresponding volume on the secondary system.

**secondary volume**
A volume to which data is backed up from a primary volume. Such a volume can be a secondary or tertiary (and onward) destination in a cascade or fanout backup configuration. The SnapVault secondary system maintains Snapshot copies for long-term storage and possible restore operations.

**incremental transfer**
A follow-up backup to the secondary system that contains only the changes to the primary data since the last transfer action.

**SnapMirror label**
An attribute that identifies Snapshot copies for the purpose of selection and retention in SnapVault backups. Each SnapVault policy configures the rules for selecting Snapshot copies on the primary volume and transferring the Snapshot copies that match a given SnapMirror label.

**Snapshot copy**
The backup images on the source volume that are created manually or automatically as scheduled by an assigned policy. Baseline Snapshot copies contain a copy of the entire source data being protected; subsequent Snapshot copies contain differential copies of the source data. Snapshot copies can be stored on the source volume or on a different destination volume in a different Storage Virtual Machine (SVM) or cluster.

Snapshot copies capture the state of volume data on each source system. For SnapVault and mirror relationships, this data is transferred to destination volumes.

**primary volume**
A volume that contains data that is to be backed up. In cascade or fanout backup deployments, the primary volume is the volume that is backed up to a SnapVault backup, regardless of where in the chain the SnapVault source is. In a cascade chain configuration in which A has a mirror relationship to B and B has a SnapVault relationship to C, B serves as the source for the SnapVault backup even though it is a secondary destination in the chain.

**SnapVault relationship**
A backup relationship, configured as a SnapVault relationship, between a primary volume and a secondary volume.

**Which data gets backed up and restored from a SnapVault backup**

You create SnapVault relationships to back up and restore volumes. You can select the Snapshot copies that the SnapVault relationship uses to backup and restore volumes.

The SnapVault operation backs up a specified volume on the primary system to the associated volume on the SnapVault secondary system. If necessary, data is restored from the SnapVault secondary volume back to the associated primary volume or to a different volume.

The Snapshot policy assigned to the source volume specifies when Snapshot copies are performed. The SnapVault policy assigned to the SnapVault relationship specifies which of the source volume Snapshot copies are replicated to the SnapVault backup.

If the destination volume is a FlexClone volume, the volume retains two more Snapshot copies than the number you configure in the policy. This occurs because the volume retains the FlexClone Snapshot copy and an exported Snapshot copy. For example, if your policy specifies to retain three Snapshot copies, five Snapshot copies are retained (three specified Snapshot copies, one FlexClone Snapshot copy, and one exported Snapshot copy).

In SAN environments, LUN identifiers are preserved on the SnapVault secondary volume.

The secondary system uses slightly more disk space and directories than the source system.

**Related concepts**

- [Which data does not get backed up to a SnapVault backup](#) on page 29
- [Guidelines for restoring the active file system](#) on page 143

**Which data does not get backed up to a SnapVault backup**

If you back up an entire Storage Virtual Machine (SVM) to a SnapVault backup by establishing a SnapVault relationship for each volume in the SVM, namespace and root information is not backed up. To protect namespace and root information for an SVM, you must manually create the namespace and root on the SnapVault secondary volume. When backing up LUNs to a SnapVault secondary volume, not all LUN information is replicated.

In SAN environments, the following LUN attributes are not replicated to the secondary volume:

- **Path**
  The LUN in the SnapVault secondary volume can be in a different SVM or volume from the source LUN. Path-related metadata, such as persistent reservations, are not replicated to the SnapVault primary volume.

- **Serial number**

- **Device ID**

- **UUID**

- **Mapped status**

- **Read Only state**
  The Read Only state is always set to **true** on the destination LUN.

- **NVFAIL attribute**
  The NVFAIL attribute is always set to **false** on the destination LUN.

You can set persistent reservations for LUNs on the SnapVault secondary volume.
How a SnapVault backup works

Backing up volumes to a SnapVault backup involves starting the baseline transfers, making scheduled incremental transfers, and restoring data upon request.

Baseline transfers

In general, baseline transfers work as follows:

A baseline transfer occurs when you initialize the SnapVault relationship. When you do this, Data ONTAP creates a new Snapshot copy. Data ONTAP transfers the Snapshot copy from the primary volume to the secondary volume. This Snapshot copy is the baseline of the volume at the time of the transfer and is a complete transfer, not an incremental transfer. As a result, none of the other Snapshot copies on the primary volume are transferred as part of the initial SnapVault transfer, regardless of whether they match rules specified in the SnapVault policy.

Incremental transfers

The source system creates incremental Snapshot copies of the source volume as specified by the Snapshot policy that is assigned to the primary volume. Each Snapshot copy for a specific volume contains a label that is used to identify it.

The SnapVault secondary system selects and retrieves specifically labeled incremental Snapshot copies, according to the rules that are configured for the SnapVault policy that is assigned to the SnapVault relationship. The Snapshot label is retained to identify the backup Snapshot copies.

Snapshot copies are retained in the SnapVault backup for as long as is needed to meet your data protection requirements. The SnapVault relationship does not configure a retention schedule, but the SnapVault policy does specify number of Snapshot copies to retain.

SnapVault backup updates

At the end of each Snapshot copy transfer session, which can include transferring multiple Snapshot copies, the most recent incremental Snapshot copy in the SnapVault backup is used to establish a new common base between the primary and secondary volumes and is exported as the active file system.

Data restore

If data needs to be restored to the primary volume or to a new volume, the SnapVault secondary transfers the specified data from the SnapVault backup.

How SnapVault backups work with data compression

SnapVault relationships preserve storage efficiency when replicating data from the source to the SnapVault secondary volume except when additional data compression is enabled.

If additional compression is enabled on the SnapVault secondary volume, storage efficiency is affected as follows:

- Storage efficiency is not preserved for data transfers between the primary and secondary volumes.
- You do not have the option of returning to replicating data while preserving storage efficiency.

Related tasks

Managing storage efficiency for SnapVault secondaries on page 151
SnapVault backup limitations

When planning SnapVault relationships, you must keep in mind what is supported and what is not supported.

The following limitations apply to SnapVault backups:

- 32-bit aggregates are not supported.
  Clustered Data ONTAP systems do not support the SnapVault backup feature for volumes in 32-bit aggregates.
- A SnapVault secondary volume cannot be the secondary volume for multiple primary volumes. A volume can be the secondary for one SnapVault relationship only. However, that same volume can be the source for other relationships.
- SnapVault backups are not supported on Infinite Volumes.

Data protection for SVM namespace and root information

Backups to secondary volumes in SnapVault relationships between FlexVol volumes replicate only volume data, not the Storage Virtual Machine (SVM) namespace or root information.

SnapVault relationships replicate only volume data. If you want to back up an entire SVM to a SnapVault secondary volume, you must first create SnapVault relationships for every volume in the SVM.

To provide data protection of the SVM namespace information, you must manually create the namespace on the SnapVault secondary immediately after the first data transfer is completed for all of the volumes in the SVM and while the source SVM volumes are still active. When subsequent changes are made to the namespace on the source SVM, you must manually update the namespace on the destination SVM.

You cannot create the namespace for an SVM on a SnapVault secondary volume if only a subset of the SVM volumes are in a SnapVault relationship, or if only a subset of the SVM volumes have completed the first data transfer.

Guidelines for planning Snapshot copy schedule and retention for SnapVault backups

It is important to plan the Snapshot copy transfer schedule and retention for your SnapVault backups.

When planning SnapVault relationships, consider the following guidelines:

- Before you create a SnapVault policy, you should create a table to plan which Snapshot copies you want replicated to the SnapVault secondary volume and how many of each you want to keep. For example:
  - Hourly (periodically throughout the day)
    Does the data change often enough throughout the day to make it worthwhile to replicate a Snapshot copy every hour, every two hours, or every four hours?
  - Nightly
    Do you want to replicate a Snapshot copy every night or just workday nights?
  - Weekly
    How many weekly Snapshot copies is it useful to keep in the SnapVault secondary volume?
- The primary volume should have an assigned Snapshot policy that creates Snapshot copies at the intervals you need, and labels each Snapshot copy with the appropriate `snapmirror-label` attribute name.
The SnapVault policy assigned to the SnapVault relationship should select the Snapshot copies you want from the primary volume, identified by the `snapmirror-label` attribute name, and specify how many Snapshot copies of each name you want to keep on the SnapVault secondary volume.

### Sample transfer schedule and retention

<table>
<thead>
<tr>
<th><code>snapmirror-label</code> attribute value</th>
<th>Source volume: Snapshot copy schedule</th>
<th>Primary volume: Snapshot copies retained</th>
<th>SnapVault secondary volume: Snapshot copies retained</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>weekly</strong></td>
<td>Every Saturday at 19:00</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>nightly</strong></td>
<td>Every Monday through Friday at 19:00</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td><strong>hourly</strong></td>
<td>Every hour from 07:00 through 18:00</td>
<td>11</td>
<td>120</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>n/a</td>
<td>25</td>
<td>188</td>
</tr>
</tbody>
</table>

### Supported data protection deployment configurations

A simple data protection deployment consists of a FlexVol volume or Infinite Volume in a single mirror relationship or a FlexVol volume in a SnapVault relationship. More complex deployment configurations that provide additional data protection consist of a cascade chain of relationships between FlexVol volumes or a set of fanout relationships for a FlexVol volume or Infinite Volume.

Although a single volume-to-volume relationship does provide data protection, your data protection needs might require the additional protection that is provided by more complex cascade and fanout configurations.

An example of a cascade chain is an A to B to C configuration. In this example, A is the source that is replicated to B as a data protection mirror, and B is the primary that is backed up to C as a SnapVault backup. Cascade chains can be more complex than A to B to C, but the more relationships that are involved in the chain, the greater the number of temporary locks on volumes while replication or update operations are in progress.

An example of a fanout is an A to B and A to C backup or mirror replication configuration. In this example, A is the primary source that is replicated to both B (either in a mirror or SnapVault relationship) and C.

**Note:** Only one SnapVault relationship is supported in a cascade chain configuration, but many SnapVault relationships are supported in a fanout configuration; multiple mirror relationships are supported.

**Attention:** The longer you configure a chain of relationships or the more you add fanout destinations, the greater the risk of Snapshot copies being locked on the source. Depending on the update schedule, the worst case is when one Snapshot copy is locked for each cascade or fanout destination.

The types of supported deployment configurations are as follows:

- Basic data protection configuration (for FlexVol volumes and Infinite Volumes)
A FlexVol volume or Infinite Volume is in a single relationship with another volume as the source or the destination of mirror replication operations, or a FlexVol volume is in a single relationship with another volume as the primary or the secondary of SnapVault operations.

- **Cascade** (one-to-one-to-one relationship)
  The three types of cascade chain relationships that you can configure are as follows:
  
  ◦ **Mirror-mirror cascade** (for FlexVol volumes only)
    A chain of at least two mirror relationships in which a volume is the source for replication operations to a secondary volume, and the secondary volume is the source for replication operations to a tertiary volume. This configuration might be described as follows: A mirror to B mirror to C.

  ◦ **Mirror-SnapVault cascade** (for FlexVol volumes only)
    A chain of a mirror relationship followed by a SnapVault relationship in which a volume is the source for replication operations to a secondary volume, and the secondary volume is the primary for SnapVault operations to a tertiary volume. This configuration might be described as follows: A mirror to B SnapVault backup to C.

  ◦ **SnapVault-mirror cascade** (for FlexVol volumes only)
    A chain of a SnapVault relationship followed by a mirror relationship in which a volume is the primary for SnapVault operations to a secondary volume, and the secondary volume is the source for replication operations to a tertiary volume. This configuration might be described as follows: A SnapVault backup to B mirror to C.

  A load-sharing mirror source volume or destination volume cannot be a part of any cascade relationship. See the *Clustered Data ONTAP Logical Storage Management Guide* for information about load-sharing mirror relationships.

- **Fanout** (one-to-many relationship)
  In a fanout relationship structure, the source is replicated to multiple destinations, which can be mirror or SnapVault destinations. Only one SnapVault relationship is allowed in a fanout.

  ◦ **Mirror-SnapVault fanout** (for FlexVol volumes only)
    A volume is the source for replication operations to a secondary volume and also the source for SnapVault operations to a different secondary volume. This configuration might be described as follows: A mirror to B and A also SnapVault backup to C.

  ◦ **Multiple-mirrors fanout** (for FlexVol volumes and Infinite Volumes)
    A volume is the source for replication operations to a destination volume and also the source for replication operations to another, different destination volume. This configuration might be described as follows: A mirror to B and A also mirror to C.

**Related concepts**

- *Components of a mirror relationship* on page 22
- *What source-to-destination-to-tape backup is* on page 34
- *How a mirror-mirror cascade works* on page 34
- *How a SnapMirror-SnapVault cascade works* on page 35
- *How a SnapVault-SnapMirror cascade works* on page 36
- *How a mirror-SnapVault fanout works* on page 37
- *How a multiple-mirrors fanout works* on page 37
What a basic backup deployment is

A basic data protection deployment consists of two volumes, either FlexVol volumes or Infinite Volumes, in a one-to-one, source-to-destination relationship. This deployment backs up data to one location, which provides a minimal level of data protection.

In a data protection configuration, source volumes are the data objects that need to be replicated. Typically, users can access and write to source volumes.

Destination volumes are data objects to which the source volumes are replicated. Destination volumes are read-only. Destination FlexVol volumes are usually placed on a different Storage Virtual Machine (SVM) from the source SVM. Destination Infinite Volumes must be placed on a different SVM from the source SVM. Destination volumes can be accessed by users in case the source becomes unavailable. The administrator can use SnapMirror commands to make the replicated data at the destination accessible and writable.

The following illustration depicts a basic data protection deployment:

Firewall requirements for intercluster SnapMirror relationships

SnapMirror relationships that have source volumes on one cluster and destination volumes on another cluster require certain ports on the intercluster network.

SnapMirror relationships use port 11104 and port 11105 on the intercluster network to replicate data from their source volumes to their destination volumes. Clustered Data ONTAP uses port 11104 to manage intercluster communication sessions and uses port 11105 to transfer data.

What source-to-destination-to-tape backup is

A common variation of the basic data protection backup deployment adds a tape backup of a destination FlexVol volume. By backing up to tape from the destination volume, you do not subject the heavily accessed source volume to the performance degradation and complexity of a direct tape backup.

The following illustration depicts a data protection chain deployment with a tape backup:

How a mirror-mirror cascade works

A mirror-mirror cascade deployment is supported on FlexVol volumes and consists of a chain of mirror relationships in which a volume is replicated to a secondary volume and the secondary is
replicated to a tertiary volume. This deployment adds one or more additional backup destinations without degrading performance on the source volume.

By replicating source A (as shown in the following illustration) to two different volumes (B and C) in a series of mirror relationships in a cascade chain, you create an additional backup. The base for the B-to-C relationship is always locked on A to ensure that the backup data in B and C always stay synchronized with the source data in A.

If the base Snapshot copy for the B-to-C relationship is deleted from A, the next update operation from A to B fails and an error message is generated that instructs you to force an update from B to C. The forced update establishes a new base Snapshot copy and releases the lock, which enables subsequent updates from A to B to finish successfully.

If the volume on B becomes unavailable, you can synchronize the relationship between C and A to continue protection of A without performing a new baseline transfer. After the resynchronize operation finishes, A is in a direct mirror relationship with C, bypassing B. Before you perform a resynchronization operation in a cascade, you should be aware that a resynchronization operation deletes Snapshot copies and might cause a relationship in the cascade to lose its common Snapshot copy. If this happens, the relationship will require a new baseline.

The following illustration depicts a mirror-mirror cascade chain:

![Mirror-mirror cascade chain illustration]

### How a SnapMirror-SnapVault cascade works

A SnapMirror-SnapVault cascade deployment, which is supported on FlexVol volumes, consists of a chain of relationships in which a volume is replicated to a destination volume, and then the destination volume becomes the primary for a SnapVault backup on a tertiary volume. This deployment adds a SnapVault backup, which fulfills more strict protection requirements.

In a typical SnapMirror-SnapVault cascade, only the exported Snapshot copies from the SnapMirror destination are transferred to the SnapVault secondary when the SnapVault update occurs. These exported Snapshot copies are created by Data ONTAP and have a “snapmirror” prefix and a “sm_created” SnapMirror label.

If the default SnapVault policy is used, the SnapVault destination will accumulate up to 251 “sm_created” Snapshot copies. After this limit is reached, when a newer “sm_created” Snapshot copy is transferred, the oldest one is rotated out. This retention and rotation behavior can be managed by adding a rule for the “sm_created” SnapMirror label to the SnapVault policy.

For example, if a rule is added with a `snapmirror-label` of “sm_created” and with a `keep` value of 40, then only 40 “sm_created” Snapshot copies are retained on the SnapVault destination. If the `preserve` value for this rule is set to `true`, then no rotation will occur and “sm_created” Snapshot copy transfers will halt when the SnapVault destination reaches a count of 40 “sm_created” Snapshot copies. If the `preserve` value for this rule is set to `false`, then “sm_created” Snapshot copy transfers will occur after 40 Snapshot copies with the oldest copy rotating out for the newest copy.
Note: A cascade chain can contain multiple SnapMirror relationships but only one SnapVault relationship. The SnapVault relationship can occur anywhere in the chain, depending on your data protection requirements.

As with other cascade configurations, a source or destination volume can become unavailable and you might consider temporarily breaking that relationship to bypass the issue and resynchronizing the relationship after fixing the issue. Before you perform a resynchronization operation in a cascade, you should be aware that a resynchronization operation deletes Snapshot copies and might cause a relationship in the cascade to lose its common Snapshot copy. If this happens, the relationship will require a new baseline.

The following illustration depicts a SnapMirror-SnapVault cascade chain:

![Cascade Chain Diagram]

Related references

Creating the SnapVault relationship of a mirror-SnapVault cascade on page 133

How a SnapVault-SnapMirror cascade works

A SnapVault-SnapMirror cascade consists of a chain of relationships in which a volume has a SnapVault backup on a secondary volume, and then that secondary volume data is replicated to a tertiary volume. In effect, this deployment provides two SnapVault backups.

A SnapVault-SnapMirror cascade deployment is only supported on FlexVol volumes. The first leg of the cascade consists of a SnapVault backup. A cascade chain in which the first leg is a SnapVault relationship behaves in the same manner as does a single leg SnapVault relationship. The updates to the SnapVault backup include the Snapshot copies that are selected in conformance with the SnapVault policy assigned to the relationship. In a typical SnapVault-SnapMirror cascade, all Snapshot copies up to the latest one are replicated from the SnapVault backup to the SnapMirror destination.

As with other cascade configurations, a source or destination volume can become unavailable and you might consider temporarily breaking that relationship to bypass the issue and resynchronizing the relationship after fixing the issue. Before you perform a resynchronization operation in a cascade, you should be aware that a resynchronization operation deletes Snapshot copies and might cause a relationship in the cascade to lose its common Snapshot copy. If this happens, the relationship will require a new baseline.

The following illustration depicts a SnapVault-SnapMirror cascade chain:
How a mirror-SnapVault fanout works

A mirror-SnapVault fanout deployment is supported on FlexVol volumes and consists of a source volume that has a direct mirror relationship to a secondary volume and also a direct SnapVault relationship to a different secondary volume.

Note: A fanout deployment might not provide as much data protection as a cascade chain.

The following illustration depicts a mirror and SnapVault fanout:

How a multiple-mirrors fanout works

A multiple-mirrors fanout deployment is supported on FlexVol volumes and Infinite Volumes, and consists of a source volume that has a direct mirror relationship to multiple secondary volumes.

The volume on A (as shown in the following illustration) always contains the base Snapshot copies for both B and C. Because updates to B or C automatically include the base Snapshot copy of the other relationship, B and C always have a common Snapshot copy.

Note: A fanout deployment might not provide as much data protection as a cascade chain.
The following illustration depicts a mirror and mirror fanout:

Storage system A

Storage system B

Storage system C

vol 1

vol 1

vol 1
Protecting data using Snapshot copies

You can use Snapshot copies to restore data that is lost because of accidental deletion to FlexVol volumes and Infinite Volumes.

Data ONTAP maintains a configurable Snapshot schedule that creates and deletes Snapshot copies automatically for each volume. You can also create and delete Snapshot copies, and manage Snapshot schedules based on your requirements.

If you lose data due to a disaster, you use data protection mirror copies to restore data.

Managing Snapshot copies

You can create multiple schedules that create and delete Snapshot copies, as desired.

Commands for managing Snapshot copies

Cluster administrators can use the `volume snapshot` commands to create and manage all Snapshot copies. Storage Virtual Machine (SVM) administrators can use the same commands to create and manage Snapshot copies within SVMs.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about Snapshot copies</td>
<td><code>volume snapshot show</code></td>
</tr>
</tbody>
</table>
| Display information about Snapshot copies created before, after, or on a certain date | `volume snapshot show` with the `-create-time` parameter  
For example, you can view Snapshot copies created fewer than 5 days ago by using the following command:  
`volume snapshot show -create-time >5d` |
| Create a Snapshot copy of a volume | `volume snapshot create`  
If you are using Infinite Volumes, you must ensure that the Infinite Volume is in an online state. You cannot create a Snapshot copy if the Infinite Volume is in a mixed state due to an offline constituent. |
| Modify the attributes of a Snapshot copy | `volume snapshot modify`  
If you are using Infinite Volumes, you cannot change the comment or name associated with a Snapshot copy of an Infinite Volume. |
| Rename a Snapshot copy of a FlexVol volume | `volume snapshot rename`  
You cannot rename a Snapshot copy that is created as a reference copy during execution of the `volume copy` or `volume move` commands.  
If you are using Infinite Volumes, you cannot rename a Snapshot copy of an Infinite Volume. |
If you want to... | Use this command...
---|---
Delete a Snapshot copy | `volume snapshot delete`
  - If you are using Infinite Volumes, the Infinite Volume must be online. You cannot delete a Snapshot copy of an Infinite Volume when the Infinite Volume is in a mixed state without assistance from technical support.
  - If you are using SnapMirror, base Snapshot copies must exist and at least one common Snapshot copy must exist between the source and destination volume to use the `snapmirror resync` command.

**Related information**

*Clustered Data ONTAP 8.3 Commands: Manual Page Reference*

**Managing Snapshot policies**

Snapshot policies automatically manage Snapshot copy schedules and retention on FlexVol volumes or Infinite Volumes. You must be a cluster administrator or Storage Virtual Machine (SVM) administrator to perform most of the Snapshot policy commands.

**How Snapshot policies are associated with volumes**

Unless you specify a Snapshot policy when you create a FlexVol volume or an Infinite Volume, a volume inherits the Snapshot policy associated with its containing Storage Virtual Machine (SVM).

When you create the SVM, you can specify a Snapshot policy. If you do not specify a Snapshot policy when you create the SVM, a default Snapshot policy is associated with the SVM. The default Snapshot policy for an SVM with FlexVol volume is named `default`, and the default Snapshot policy for an SVM with Infinite Volume is named `default-1weekly`.

**Note:** When you upgrade the SVM with Infinite Volume from Data ONTAP 8.1.x, the default Snapshot policy changes from `default` to `default-1weekly`.

When you create a volume, you can specify a Snapshot policy. If you do not specify a Snapshot policy when you create a volume, the volume inherits the Snapshot policy associated with its containing SVM.

**Note:** A Snapshot policy is not associated with each constituent in an Infinite Volume, and you cannot associate a Snapshot policy with constituents. A Snapshot policy is only associated with an Infinite Volume.

**Commands for managing Snapshot policies and schedules**

Cluster administrators can use the `volume snapshot policy` commands to create and manage all Snapshot copy policies. Storage Virtual Machine (SVM) administrators can use the same commands to create and manage Snapshot policies within SVMs.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about Snapshot copy policies</td>
<td><code>volume snapshot policy show</code></td>
</tr>
<tr>
<td>Create a new Snapshot copy policy</td>
<td><code>volume snapshot policy create</code></td>
</tr>
</tbody>
</table>
If you want to... | Use this command...
---|---
Create a schedule that can be used in Snapshot copy policies | job schedule cron create
Add a schedule to an existing Snapshot copy policy | volume snapshot policy add-schedule
A Snapshot policy can have up to five schedules. If you are using Infinite Volumes, scheduled Snapshot copies cannot occur more often than hourly. Cluster administrator only
Remove a schedule from a Snapshot copy policy Cluster administrator only | volume snapshot policy remove-schedule
Modify the maximum number of Snapshot copies for a Snapshot copy policy schedule Cluster administrator only | volume snapshot policy modify-schedule
Modify the description of a Snapshot copy policy | volume snapshot policy modify
Dissociate a Snapshot copy policy from a volume Cluster administrator only | volume modify
Delete a Snapshot copy policy Cluster administrator only | volume snapshot policy delete

See the man page for each command for more information.

**Strategies for creating a Snapshot copy policy**

You should create a Snapshot copy policy that meets the needs of your organization and users.

Following are some strategies for using policies and schedules to schedule and retain Snapshot copies:

- If users rarely lose files or typically notice lost files right away, you can use the default Snapshot copy policy.
  This policy uses the weekly schedule to create two weekly Snapshot copies; the daily schedule to create a Snapshot copy every day and keeps two; and the hourly schedule to create hourly Snapshot copies and keeps six.

- If users commonly lose files or do not typically notice lost files right away, you should delete the Snapshot copies less often than you would if you used the default policy.
  Following is the recommended Snapshot copy policy for this situation. It uses the weekly schedule to keep two weekly Snapshot copies, the daily schedule to keep six daily Snapshot copies, and the hourly schedule to keep eight hourly Snapshot copies:

```
snapshot policy create -vserver vs1.example.com -policy keep-more-snapshot -enabled true -schedule1 weekly -count1 2 -prefix1 weekly -schedule2 daily -count2 6 -prefix2 daily -schedule3 hourly -count3 8 -prefix3 hourly
```

On many systems, only 5 to 10 percent of the data changes each week, so the Snapshot copy schedule of six daily and two weekly Snapshot copies consumes 10 to 20 percent of disk space. Considering the benefits of Snapshot copies, it is worthwhile to reserve this amount of disk space for Snapshot copies.
• You can create different Snapshot copy policies for different volumes on a Vserver. On a very active volume, you should schedule Snapshot copies every hour and keep them for just a few hours, or turn off Snapshot copies. For example, the following schedule creates a Snapshot copy every hour and keeps the last three:

```
snapshot policy create -vserver vs1.example.com -policy hourly-keep-3 -enabled true -schedule1 hourly -count1 3 -prefix1 hourly
```

• When you create a new volume, the new volume inherits the Snapshot copy schedule from the root volume. After you use the volume for a while, you should check how much disk space the Snapshot copies consume and how often users need to recover lost files, and then adjust the schedule as necessary.

### Naming convention for scheduled Snapshot copies

The scheduled Snapshot copy name is composed of an optional prefix or the schedule name specified in the Snapshot policy, and the timestamp. Snapshot names cannot be longer than 255 characters.

If prefix is specified, the Snapshot name is made up of prefix and the timestamp.

If you do not specify the prefix, by default, the schedule name is prepended with the timestamp to form a Snapshot name.

### What prefixes are

A prefix is an optional string that you can specify to be used in creating automatic Snapshot copies. Using prefixes in Snapshot names provides more flexibility than using schedule names in naming automatic Snapshot copies.

Prefix names must be unique within a policy. The length of the prefix cannot exceed the maximum allowable length for Snapshot names; that is, Snapshot names cannot be longer than 255 characters. Prefix names must follow the character encoding rules used by Snapshot names.

If a prefix is specified in the Snapshot schedule, then the schedule name is not used to name Snapshot copies. If the prefix is not specified for a Snapshot schedule within a Snapshot policy, then the schedule name is used.

### Using prefixes to name automatic Snapshot copies

You can use prefixes to provide flexibility to the naming convention for scheduled Snapshot copies. It removes the dependency on using the schedule names in naming scheduled Snapshot copies.

### About this task

• A schedule cannot have more than one prefix.

• Prefixes within a policy should be unique.

### Step

1. You can specify prefixes when you create a Snapshot policy or when you add a schedule to the Snapshot policy.

### Example

The following command creates a Snapshot policy “test”, which contains the schedule named “5min” having the temp prefix:

```
cluster1::> volume snapshot policy create -policy test -enabled true -schedule1 5min -count1 2 -prefix1 temp
```
Example
The following command adds the “6min” schedule with the “test” prefix to the default policy:

```
cluster1::> volume snapshot policy add-schedule -policy default
           -schedule 6min -count 4 -prefix test
```

**Restoring files from the Snapshot copy of a FlexVol volume**

You might have to restore a file from the Snapshot copy of a FlexVol volume if the file was accidentally erased or corrupted. You can use the SnapRestore feature to automatically restore files from the Snapshot copy of a FlexVol volume.

**Steps**

1. If the original file still exists and you do not want it overwritten by the file in a Snapshot copy, then use your UNIX or Windows client to rename the original file or move it to a different directory.

2. Locate the Snapshot copy that contains the version of the file that you want to restore.

3. Copy the file from the `.snapshot` directory to the directory in which the file originally existed.

**Restoring a single file from a Snapshot copy of a FlexVol volume**

You can restore a single file to the required version from a Snapshot copy of a FlexVol volume.

**Before you begin**

- The volume to which you want to restore the file should be online and writeable.
- The volume to which you want to restore the file should have enough space for the restore operation to be completed successfully.

**About this task**

The restored file can replace an existing file with the same name in the active file system or become a new file if there is data in the existing file that you want to retain. You can also restore LUNs, but you cannot restore a single file from a Snapshot copy of an Infinite Volume.

If you are restoring an existing LUN, a LUN clone is created and is backed up in the form of a Snapshot copy. During the restore operation, you can read to and write from the LUN.

**Step**

1. To restore a single file or LUN, use the `volume snapshot restore-file` command.

   The restore operation might take a long time, depending on the size of the file or LUN that you are restoring.

   If you want to display the number of in-progress single file restore operations, use the `volume snapshot restore-file-info` command.
Restoring part of a file from a Snapshot copy of a FlexVol volume

You can restore a range of data from a file in a Snapshot copy to an existing file in the active file system. Partial file restores can only be used to restore specific pieces of a LUN, and NFS or CIFS container files.

Before you begin

- You must understand the metadata of the host LUN or container file so that you know which bytes belong to the object that you want to restore.
- The volume where the LUN or the container file is to be restored must be online and writeable.

About this task

Write operations are not allowed on the object that you are restoring; otherwise, it might result in inconsistent data.

Steps

1. Restore part of a file by using the `volume snapshot partial-restore-file` command. To get the settings for partial file restore on a cluster, use the `volume snapshot partial-restore-file-list-info` command.
2. After the restore is complete, purge operating system or application buffers so that the stale data is cleaned.

Restoring the contents of a volume from a Snapshot copy

You can restore the contents of a FlexVol volume or Infinite Volume from a Snapshot copy to quickly recover lost or damaged data.

Before you begin

- You must have the advanced privilege level or higher to run the command.
- If you are working with a Snapshot copy of an Infinite Volume, the Snapshot copy must be valid and the Infinite Volume must be online.
- You must not have any I/O traffic running on the volume.

Steps

1. If the volume is an Infinite Volume, use the `volume unmount` command to unmount it.
2. Use the `volume snapshot restore` command to restore the contents of a volume from a Snapshot copy.

Example

The following command restores data to a volume named `src_os` from a Snapshot copy named `src_os_snap_3` on the Storage Virtual Machine (SVM) named vs0:

```
vsl:~*> volume snapshot restore -vserver vs0.example.com -volume src_os -snapshot src_os_snap_3
```

3. If the volume is an Infinite Volume, use the `volume mount` command to mount it.
4. If the volume has SnapMirror relationships, manually replicate all mirror copies of the volume immediately after you restore from a Snapshot copy. Not doing so can result in unusable mirror copies that must be deleted and re-created.

Snapshot restoration using Shadow Copy Client tools

You can access and restore Data ONTAP Snapshot files using the Windows Shadow Copy Client. The Shadow Copy Client provides a Previous Versions tab in the Properties menu from which you can view and restore Data ONTAP Snapshot images.

The Shadow Copy Client software for Windows 2003 is called the Previous Versions Client. Downloads available from Microsoft allow you to use Shadow Copy client tools on most older versions of Windows. For more information about Shadow Copy Client or Previous Versions Client software, consult the Microsoft documentation.

Managing Snapshot copy disk space

The data referenced by a Snapshot copy cannot be accidentally deleted because of the Snapshot feature's design.

Monitoring Snapshot copy disk consumption

You can monitor Snapshot copy disk consumption using the `df` command, which displays the amount of free space on a disk.

About this task

For an Infinite Volume, the `df` command displays information about all of the data constituents, not about the Infinite Volume as a whole.

Step

1. To display information about Snapshot copy disk consumption, use the `df` command.

Example

The `df` command treats Snapshot copies as a partition different from the active file system. The following example shows a volume with these characteristics:

- The total volume capacity (`bytes` column) is 4,000,000 KB (4 GB): 3,000,000 KB (75 percent) for the active file system, and 1,000,000 KB (25 percent) for Snapshot copies.

- The active file system is using 2,000,000 KB of its 3,000,000 KB capacity (66 percent, rounded to 65 percent in the `capacity` column), leaving 1,000,000 KB (34 percent) available.

- Snapshot copies are using 500,000 KB of their 1,000,000 KB capacity (50 percent in the `capacity` column), leaving 500,000 KB (50 percent of the space allotted for Snapshot copies, not 50 percent of disk space) available.

Note: It is important to understand that the `/vol/vol0/.snapshot` line counts data that exists only in a Snapshot copy. The Snapshot copy calculation does not include Snapshot copy data that is shared with the active file system.
<table>
<thead>
<tr>
<th>Filesystem</th>
<th>kbytes</th>
<th>used</th>
<th>avail</th>
<th>capacity</th>
<th>Mounted on</th>
<th>Vaserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>/vol/vol0/</td>
<td>3000000</td>
<td>200000</td>
<td>1000000</td>
<td>65%</td>
<td>vs1</td>
<td>vs1</td>
</tr>
<tr>
<td>/vol/vol0/.snapshot</td>
<td>1000000</td>
<td>500000</td>
<td>500000</td>
<td>50%</td>
<td>---</td>
<td>vol</td>
</tr>
</tbody>
</table>
Managing data protection using SnapMirror policies

To manage a data protection mirror, vault, or mirror and vault relationship, you must assign a policy to the relationship. You use the policy to maximize the efficiency of the transfers to the backup secondaries and manage the update operations for SnapVault backups.

SVM disaster recovery relationships support only SnapMirror policies.

FlexVol volumes support data protection mirror, vault, and mirror and vault relationships and policies. Infinite Volumes support only data protection mirror relationships and policies.

Commands for managing SnapMirror and SnapVault policies

Cluster administrators can use the snapmirror policy commands to create and manage all data protection mirror and SnapVault policies. Storage Virtual Machine (SVM) administrators can use the same commands to create and manage all data protection mirror and SnapVault policies within SVMs.

- All policy-management commands (except for the snapmirror policy show command) must be run on the SVM that contains the destination volume.

- Commands for SnapVault policies are supported only by FlexVol volumes.

### If you want to... | Use this command...
---|---
Add a new rule to a SnapVault policy | snapmirror policy add-rule create
Create a new SnapMirror or SnapVault policy | snapmirror policy create
Delete a SnapMirror or SnapVault policy | snapmirror policy delete
Modify a SnapMirror or SnapVault policy | snapmirror policy modify
Add network compression to a SnapMirror or SnapVault policy | snapmirror policy create with the -is-network-compression-enabled true parameter or snapmirror policy modify with the -is-network-compression-enabled true parameter
Modify an existing rule in a SnapVault policy | snapmirror policy modify-rule command
Remove a rule in a SnapVault policy | snapmirror policy remove-rule
Display a list of SnapMirror and SnapVault policies | snapmirror policy show

Related information

*Clustered Data ONTAP 8.3 Commands: Manual Page Reference*

How SnapMirror policies work with clusters and SVMs

A SnapMirror policy in which the vserver parameter contains the cluster name is a cluster-wide policy. You can assign cluster-wide SnapMirror policies to relationships in a cluster. Cluster-wide policies can be configured only by a cluster administrator.
A SnapMirror policy in which the `vserver` parameter contains a Storage Virtual Machine (SVM) name is an SVM-wide policy. You can assign SVM-wide SnapMirror policies to relationships within the SVM in which the policy was created. SVM policies can be configured by either a cluster administrator or an SVM administrator.

**Comparison of what cluster administrators and SVM administrators can manage**

Cluster administrators and Storage Virtual Machine (SVM) administrators have different privileges for creating, managing, and assigning policies to mirror and SnapVault relationships.

Cluster administrators can do the following:

- Create and manage any policy in a cluster or SVM
  - For cluster-wide policies, the `vserver` parameter contains the cluster name. For SVM-wide policies, the `vserver` parameter contains the SVM name.
- View, modify, or delete policies in a cluster or SVM
- Assign a cluster-wide or SVM-wide policy to a relationship

SVM administrators can do the following:

- Create and manage policies within an SVM
  - Policies created by an SVM administrator are automatically configured with the SVM name in the `vserver` parameter.
- View cluster-wide policies and SVM-wide policies created in a specified SVM
  - Although SVM administrators can view cluster-wide policies, they cannot modify or delete them. SVM administrators can view only the SVM-wide policies that were created within the SVM on which the `snapmirror policy show` command is executed.
- Assign a cluster-wide or SVM-wide policy to a relationship

SVM administrators cannot access SVM-wide policies of another SVM.

**Guidelines for naming Snapmirror and SnapVault policies**

Before you create a SnapMirror or SnapVault policy, you should ensure that the policy name is unique.

Cluster-wide policy names must be unique within the cluster and must not conflict with any Storage Virtual Machine (SVM)-wide policy names.

SVM-wide policy names must be unique within the SVM in which the policy is created. However, an SVM policy name can be the same as a policy name created in a different SVM, as long as the name does not conflict with any cluster-wide policy name.
Preserving Snapshot copies for SnapVault relationships after reaching retention limit

After the Snapshot copy retention limit defined by a SnapMirror policy for a SnapVault relationship is reached, the oldest Snapshot copy is automatically deleted to create space before transferring a new Snapshot copy. You can configure or modify the policy rule to preserve all Snapshot copies.

About this task

You can configure or modify the policy rule to preserve all Snapshot copies when you create the SnapMirror policy rule, or you can modify a previously created SnapMirror policy. Configuring or modifying the policy rule to preserve Snapshot copies causes incremental updates to the SnapVault secondary to fail after Snapshot copies reach the retention count.

Steps

1. Configure or modify the policy rule to preserve Snapshot copies by using the `snapmirror policy add-rule` command or the `snapmirror policy modify-rule` command with the `-preserve` parameter.

   Example
   
   The following example configures the XDPDefault policy rule to preserve the 40 sm_created Snapshot copies that you want to retain.

   ```bash
   cluster1::> snapmirror policy add-rule -vserver vs1 -policy XDPDefault -snapmirror-label sm_created -keep 40 -preserve true
   ```

   Example
   
   The following example modifies the XDPDefault policy rule to preserve the 40 sm_created Snapshot copies that you want to retain.

   ```bash
   cluster1::> snapmirror policy modify-rule -vserver vs1 -policy XDPDefault -snapmirror-label sm_created -preserve true
   ```

2. Optionally, check the policy rules to ensure that you enabled the `-preserve` parameter by using the `snapmirror policy show` command with the `-instance` parameter:

   Example
   
   ```bash
   cluster1::> snapmirror policy show -instance
   Vserver: vs1
   SnapMirror Policy Name: XDPDefault
   Policy Owner: cluster-admin
   Tries Limit: 8
   Transfer Priority: normal
   Ignore accesstime Enabled: false
   Transfer Restartability: always
   Comment: Default policy for XDP relationship with daily and weekly rules.
   Total Number of Rules: 3
   Total Keep: 139
   Rules: Snapmirror-label Keep Preserve Warn
   ```
Example of creating a tiered backup policy

Data ONTAP uses the `snapmirror-label` attribute to identify Snapshot copies between primary and secondary FlexVol volumes in a SnapVault relationship. When you configure rules in a SnapVault policy, you enter the `snapmirror-label` name that you want to use to identify the Snapshot copies to which the rule applies.

In a tiered backup strategy, a SnapVault policy might have several rules, each one identifying a different set of Snapshot copies. In this example, you have a volume to which you have assigned a Snapshot policy that specifies the following schedule:

- **An hourly Snapshot copy**
  Every two hours, a Snapshot copy is created and is assigned the attribute `-snapmirror-label hourly`.

- **A daily Snapshot copy**
  Every day at 5:00 p.m., a Snapshot copy is created and is assigned the attribute `-snapmirror-label daily`.

- **A weekly Snapshot copy**
  Every Friday at 6:00 p.m., a Snapshot copy is created and is assigned the attribute `-snapmirror-label weekly`.

In addition, the volume is part of an Oracle database. Using the online management tool for Host Services Agent for Oracle, you set up a schedule that creates a Snapshot copy every day at 5:00 p.m. These Snapshot copies are assigned the attribute `-snapmirror-label Oracle-consistent`.

To set up tiered, disk-to-disk data protection for this volume, in which only the Snapshot copies labeled `daily`, `weekly`, and `Oracle-consistent` are replicated to the SnapVault backup, you do the following:

1. **Create a separate rule for each of the three types of Snapshot copies that you want replicated to the SnapVault secondary volume.**
   You should have three rules. Each rule must specify the retention count. For this example, you configure a retention count of 20 for the daily Snapshot copies, 24 for the weekly Snapshot copies, and 100 for the Oracle-consistent Snapshot copies.

2. **Create a new “TieredOracle” SnapVault policy by using the `snapmirror policy create` command, and add the rules you created in Step 1.**

3. **Assign the new SnapVault policy to the SnapVault relationship that exists between the primary and secondary volumes.**

The new SnapVault policy configuration is as follows:

```
<table>
<thead>
<tr>
<th>Vserver</th>
<th>Policy Name</th>
<th>Number Of Rules</th>
<th>Transfer Tries</th>
<th>Priority</th>
<th>Restart</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1</td>
<td>TieredOracle</td>
<td>3</td>
<td>8</td>
<td>normal</td>
<td>default</td>
<td>Example of a tiered backup policy</td>
</tr>
</tbody>
</table>

SnapMirror-label: daily Keep: 20
weekly 24
Oracle-consistent 100
```

**Related concepts**

*Commands for managing SnapMirror and SnapVault policies* on page 47
Related references

*Commands for managing SnapMirror and SnapVault policies* on page 47
Managing peer relationships for data backup and recovery (cluster administrators only)

Establishing peer relationships between two clusters or two Storage Virtual Machines (SVMs) enables you to back up and recover the data on the clusters or SVMs.

Managing cluster peer relationships

You can create data protection mirroring relationships from one cluster to another and you can manage the jobs on a remote cluster from another cluster if you have cluster peer relationships.

What a cluster peer is

The cluster peer feature allows two clusters to coordinate and share resources between them.

Commands for managing cluster peer relationships

There are specific Data ONTAP commands for managing cluster peer relationships.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an authenticated cluster peer relationship</td>
<td><code>cluster peer create</code> This creates an authenticated cluster peer relationship by default. If you want to create an unauthenticated cluster peer relationship, you use this command with the <code>-no-authentication</code> parameter and unauthenticated cluster peer relationships must also be allowed by the cluster peer policy.</td>
</tr>
<tr>
<td>Create an authenticated cluster peer relationship with an extended authentication offer</td>
<td><code>cluster peer create</code> with the <code>-offer-expiration</code> parameter. This is useful if the second cluster in the relationship cannot be authenticated in the default time of one hour.</td>
</tr>
<tr>
<td>Create an unauthenticated cluster peer relationship</td>
<td><code>cluster peer create</code> with the <code>-no-authentication</code> parameter. Unauthenticated cluster peer relationships must also be allowed by the cluster peer policy.</td>
</tr>
<tr>
<td>Create an authenticated cluster peer relationship using a specific IPspace</td>
<td><code>cluster peer create</code> with the <code>ipspace</code> parameter.</td>
</tr>
<tr>
<td>Delete a cluster peer relationship</td>
<td><code>cluster peer delete</code></td>
</tr>
<tr>
<td>Reverting a cluster peer relationship's IPspace designation back to the Default IPspace</td>
<td><code>cluster peer modify</code> with the <code>ipspace</code> parameter set to <code>Default</code>.</td>
</tr>
<tr>
<td>Modify a cluster peer relationship</td>
<td><code>cluster peer modify</code></td>
</tr>
<tr>
<td>If you want to...</td>
<td>Use this command...</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>Change an IP space for a cluster in a cluster peer relationship</td>
<td><code>cluster peer modify</code> with the <code>ipspace</code> parameter.</td>
</tr>
<tr>
<td></td>
<td>This is useful if you have an existing cluster peer relationship and you want to use intercluster LIFs in a designated IP space.</td>
</tr>
<tr>
<td>Initiate an intercluster connectivity test</td>
<td><code>cluster peer ping</code></td>
</tr>
<tr>
<td>Display information about the cluster peer relationship</td>
<td><code>cluster peer show</code></td>
</tr>
<tr>
<td>Display TCP connection information for a cluster peer</td>
<td><code>cluster peer connection show</code></td>
</tr>
<tr>
<td>Display health information of the nodes in a cluster peer relationship from the local cluster perspective</td>
<td><code>cluster peer health show</code></td>
</tr>
<tr>
<td>Display information about outstanding authentication offers to peer clusters</td>
<td><code>cluster peer offer show</code></td>
</tr>
<tr>
<td>Disable an existing cluster peer relationship</td>
<td><code>cluster peer modify</code> with the <code>-auth-status-admin</code> parameter set to <code>revoked</code>.</td>
</tr>
<tr>
<td>Update a cluster peer relationship to use authentication with a different passphrase</td>
<td><code>cluster peer modify</code> with the <code>-auth-status-admin</code> parameter set to <code>use-authentication</code>.</td>
</tr>
<tr>
<td></td>
<td>You can use this command to perform one of the following tasks:</td>
</tr>
<tr>
<td></td>
<td>• Add authentication to an existing unauthenticated cluster peer relationship</td>
</tr>
<tr>
<td></td>
<td>• Add authentication to a revoked cluster peer relationship</td>
</tr>
<tr>
<td></td>
<td>• Change a passphrase for an authenticated cluster peer relationship</td>
</tr>
<tr>
<td>Reestablish a disabled cluster peer relationship with no-authentication</td>
<td><code>cluster peer modify</code> with the <code>-auth-status-admin</code> parameter set to <code>no-authentication</code>.</td>
</tr>
<tr>
<td>Modify an outstanding authentication offer to a peer cluster</td>
<td><code>cluster peer offer modify</code> with the <code>-offer-expiration</code> parameter.</td>
</tr>
<tr>
<td></td>
<td>You can change when the authentication offer expires if you determine that there is not enough time to authenticate the cluster peer relationship before the offer expires.</td>
</tr>
<tr>
<td>Cancel an outstanding authentication offer to a peer cluster</td>
<td><code>cluster peer offer cancel</code></td>
</tr>
<tr>
<td>Display whether unauthenticated cluster peer relationships can exist and what the minimum passphrase character length is</td>
<td><code>cluster peer policy show</code></td>
</tr>
<tr>
<td>If you want to...</td>
<td>Use this command...</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Modify whether unauthenticated cluster peer relationships can exist and change the minimum passphrase character length</td>
<td><code>cluster peer policy modify</code></td>
</tr>
</tbody>
</table>

Related information

*Clustered Data ONTAP 8.3 Commands: Manual Page Reference*

**Cluster peer security using authentication passphrases**

When creating a cluster peer relationship, a passphrase is used by the administrators of the two clusters to authenticate the relationship. This ensures that the cluster to which you send data is the cluster to which you intend to send data.

A part of the cluster peer creation process is to use a passphrase to authenticate the cluster peers to each other. The passphrase is used when creating the relationship from the first cluster to the second and, again, when creating the relationship from the second cluster to the first. The passphrase is not exchanged on the network by Data ONTAP, but each cluster in the cluster peer relationship recognizes the passphrase when Data ONTAP creates the cluster peer relationship.

When you create the cluster peer relationship from the first cluster to the second, the first cluster waits for the administrator of the second cluster to create the cluster peer relationship. The administrator of the second cluster must create the cluster peer relationship before the waiting period expires, one hour by default, but can be shortened. If the cluster peer relationship is not created from the second cluster to the first before the waiting period expires, the cluster peer relationship is not created and the administrators must start again.

**Related tasks**

*Creating the cluster peer relationship* on page 71

**Related references**

*Commands for managing cluster peer relationships* on page 52

**Connecting one cluster to another cluster in a peer relationship**

You connect clusters together in a cluster peer relationship to share information and to provide access to operations on the peer cluster.

**About this task**

Connecting clusters together requires network ports, network interfaces configured with the intercluster role, and creating the cluster peer relationship.

**Steps**

1. What cluster peer intercluster connectivity is on page 55
2. Supported cluster peer network topologies on page 55
3. Comparison of designated and undesignated intercluster connectivity on page 58
4. Prerequisites for cluster peering on page 59
5. Considerations when sharing data ports on page 60
6. Considerations when using dedicated ports on page 60
7. Configuring intercluster LIFs to share data ports on page 61
8. Configuring intercluster LIFs to use dedicated intercluster ports on page 63
9. Configuring intercluster LIFs to use intercluster ports in their own networks on page 66
10. Creating the cluster peer relationship on page 71

What cluster peer intercluster connectivity is

You should know about the interfaces and ports that you put together to create a cluster peer intercluster connection and how they are used. Knowing this information might reduce the amount of time you use to create the cluster peer intercluster connectivity.

Cluster peer intercluster connectivity consists of intercluster logical interfaces (LIFs) that are assigned to network ports. The intercluster connection on which replication occurs between two different clusters is defined when the intercluster LIFs are created. Replication between two clusters can occur on the intercluster connection only; this is true regardless of whether the intercluster connectivity is on the same subnet as a data network in the same cluster.

The IP addresses assigned to intercluster LIFs can reside in the same subnet as data LIFs or in a different subnet. When an intercluster LIF is created, it uses routes that belong to the System SVM that the intercluster LIF is in.

Related concepts

Considerations when sharing data ports on page 60
Considerations when using dedicated ports on page 60

Supported cluster peer network topologies

To provide data protection, all of the intercluster LIFs of one cluster must be able to communicate with all of the intercluster LIFs of the cluster peer using pair-wise full-mesh connectivity. You need to understand how this connectivity works for different cluster topologies.

Pair-wise full-mesh connectivity applies only to the two clusters in the peer relationship. All of the intercluster LIFs of an IPspace in one cluster must be able to communicate with all of the intercluster LIFs of an IPspace in the other cluster.

Using the concept of pair-wise full-mesh connectivity helps you to build more complex cluster peer topologies. Understanding how this connectivity works for two cluster, cluster cascade, and cluster fan-out or fan-in topologies will help you to create viable intercluster networks without adding intercluster networks that are unnecessary.

Intercluster networking between two clusters

Creating an intercluster network between two clusters is the basic cluster peer configuration. For example, you want to create an intercluster network between two clusters, Cluster A and Cluster B. Cluster A has two intercluster LIFs, A1 and A2, in its Default IPspace and Cluster B has two intercluster LIFs, B1 and B2, in its Default IPspace. The LIFs are connected as follows:

- A1 communicates with B1 and
- A1 communicates with B2 and
- A2 communicates with B1 and
- A2 communicates with B2
**Intercluster networking in a cluster cascade**

When you connect three clusters in a cascade, all of the intercluster LIFs of the primary cluster must be able to communicate with all of the intercluster LIFs of the secondary cluster. Likewise, all of the intercluster LIFs of the secondary cluster must be able to communicate with all of the intercluster LIFs of the tertiary cluster. You do not need to create an intercluster network between the primary cluster and the tertiary cluster if you do not want to connect the two clusters in a cluster peer relationship.

For example, you want to create an intercluster network between Cluster A and Cluster B and an intercluster network between Cluster B and Cluster C. Cluster A has two intercluster LIFs, A1 and A2, in its Default IPspace, Cluster B has two intercluster LIFs, B1 and B2, in its Default IPspace, and Cluster C has two intercluster LIFs, C1 and C2, in its Default IPspace. The intercluster LIFs between Cluster A and Cluster B are connected as follows:

- A1 communicates with B1 and
- A1 communicates with B2 and
- A2 communicates with B1 and
- A2 communicates with B2 and

The intercluster LIFs between Cluster B and Cluster C are connected as follows:

- B1 communicates with C1 and
- B1 communicates with C2 and
- B2 communicates with C1 and
- B2 communicates with C2

![Diagram](image_url)

You might have a cluster cascade configured in which you want the tertiary cluster to connect to the primary cluster if something happens to the secondary cluster. An example is if you have a disaster recovery relationship between the primary cluster and the secondary cluster, and a backup relationship between the secondary cluster and the tertiary cluster, and you want the tertiary cluster to communicate with the primary cluster if something happens to the secondary cluster. If this configuration is what you want, then the intercluster LIFs of the tertiary cluster must be able to communicate with all of the intercluster LIFs of the primary cluster. Therefore, in addition to the connections previously mentioned, you would also have the following intercluster LIF connections between Cluster C and Cluster A:

- A1 communicates with C1 and
- A1 communicates with C2 and
- A2 communicates with C1 and
- A2 communicates with C2
Intercluster networking in a cluster fan-out or fan-in

When you connect clusters in a fan-out or fan-in configuration, the intercluster LIFs of each cluster that connects to the primary cluster must be able to communicate with all of the intercluster LIFs of the primary cluster. You do not need to connect intercluster LIFs between the remote clusters if the remote clusters do not need to communicate with each other.

For example, you want to create an intercluster network between Cluster A and Cluster B and an intercluster network between Cluster A and Cluster C. Cluster A has two intercluster LIFs, A1 and A2, in its Default IPspace, Cluster B has two intercluster LIFs, B1 and B2, in its Default IPspace, and Cluster C has two intercluster LIFs, C1 and C2, in its Default IPspace. The intercluster LIFs between Cluster A and Cluster B are connected as follows:

• A1 communicates with B1 and
• A1 communicates with B2 and
• A2 communicates with B1 and
• A2 communicates with B2

The intercluster LIFs between Cluster A and Cluster C are connected as follows:

• A1 communicates with C1 and
• A1 communicates with C2 and
• A2 communicates with C1 and
• A2 communicates with C2

Cluster B is not connected to Cluster C.

If you do want a cluster peer relationship between two remote clusters in addition to the fan-in or fan-out configuration, then use the concept of pair-wise full-mesh connectivity to create an intercluster network between them.
Comparison of designated and undesignated intercluster connectivity

Unlike undesignated intercluster connectivity between clusters in their respective default IPspaces, designated intercluster connectivity uses a non-default IPspace to contain the interactions that a cluster has with its peer. Understanding how connectivity differs helps you decide if you want to use designated intercluster connectivity.

Undesignated intercluster connectivity operates within the default IPspaces of the clusters. Because of the pair-wise full-mesh connectivity requirement within an IPspace, all of the intercluster LIFs of the two clusters in the cluster peer relationship must be able to connect to each other. This connectivity requirement does not allow separation of connectivity that storage service providers might need. You would have to introduce hardware, such as a router, into the network to separate intercluster connectivity between clusters.

Designated intercluster connectivity operates within a specified non-default IPspace on at least one of the clusters in the peer relationship. The requirement of pair-wise full-mesh connectivity still exists, but the connectivity is within the IPspace that each cluster defines for the peering relationship and not across IPspaces. This keeps the intercluster connectivity isolated from different IPspaces that a cluster might have with other peers and also reduces the scope of the full-mesh connectivity requirement. In this way, a storage service provider can control the intercluster connectivity separation that might be needed.

As an example, you want Cluster A to have a peering relationship with two other clusters, Cluster B and Cluster C. Additionally, you want the connectivity between Cluster A and Cluster B to be separate from the connectivity between Cluster A and Cluster C. To do this, you can create two IPspaces on Cluster A. The first IPspace, called ipspaceAB, will contain the intercluster LIFs of Cluster A that you want to use to communicate with Cluster B. Likewise, the second IPspace, called ipspaceAC, will contain the intercluster LIFs of Cluster A that you want to use to communicate with Cluster C.

The intercluster LIFs on Cluster B and Cluster C are contained in the default IPspaces of their respective clusters. It is not a requirement that these intercluster LIFs be in the default IPspace. These intercluster LIFs can be in the default IPspace or in their own designated IPspaces.

For the peering relationship between Cluster A, Cluster B, and Cluster C, Cluster A would use only those LIFs in IPspace ipspaceAB to communicate with Cluster B, and only those LIFs in IPspace ipspaceAC to communicate with Cluster C. There is no requirement that intercluster LIFs in Cluster B communicate with any other intercluster LIFs on Cluster A other than those in IPspace ipspaceAB, and no requirement that intercluster LIFs in Cluster C communicate with any other intercluster LIFs on Cluster A other than those in IPspace ipspaceAC.
Prerequisites for cluster peering

Before you set up cluster peering, you should confirm that the IPspace, connectivity, port, IP address, subnet, firewall, and cluster-naming requirements are met.

Connectivity requirements

The subnet used in each cluster for intercluster communication must meet the following requirements:

• The subnet must belong to the broadcast domain that contains the ports used for intercluster communication.

• IP addresses used for intercluster LIFs do not need to be in the same subnet, but having them in the same subnet is a simpler configuration.

• You must have considered whether the subnet will be dedicated to intercluster communication or shared with data communication.

• The subnet must have enough IP addresses available to allocate to one intercluster LIF per node. For example, in a six-node cluster, the subnet used for intercluster communication must have six available IP addresses.

The intercluster network must be configured so that cluster peers have pair-wise full-mesh connectivity within the applicable IPspace, which means that each pair of clusters in a cluster peer relationship has connectivity among all of their intercluster LIFs.

A cluster's intercluster LIFs must use the same IP address version: all IPv4 addresses or all IPv6 addresses. Similarly, all of the intercluster LIFs of the peered clusters must use the same IP addressing version.

Port requirements

The ports that will be used for intercluster communication must meet the following requirements:

• The broadcast domain that is used for intercluster communication must include at least two ports per node so that intercluster communication can fail over from one port to another.

• All of the ports must be cabled.

• All of the ports must be in a healthy state.

• The MTU settings of the ports must be consistent.

• You must have considered whether the ports used for intercluster communication will be shared with data communication.

Firewall requirements

Firewalls and the intercluster firewall policy must allow the following:

• ICMP service

• TCP to the IP addresses of all of the intercluster LIFs over all of the following ports: 10000, 11104, and 11105

• HTTPS

Although HTTPS is not required when you set up cluster peering, HTTPS is required later if you use OnCommand System Manager to configure data protection. However, if you use the
command-line interface to configure data protection, HTTPS is not required to configure cluster peering or data protection.

The default intercluster firewall policy allows access through the HTTPS protocol and from all IP addresses (0.0.0.0/0), but the policy can be altered or replaced.

**Cluster requirements**

Clusters must meet the following requirements:

- Each cluster must have a unique name. You cannot create a cluster peering relationship with any cluster that has the same name or is in a peer relationship with a cluster of the same name.
- The time on the clusters in a cluster peering relationship must be synchronized within 300 seconds (5 minutes). Cluster peers can be in different time zones.

**Considerations when sharing data ports**

When determining whether sharing a data port for intercluster replication is the correct intercluster network solution, you should consider configurations and requirements such as LAN type, available WAN bandwidth, replication interval, change rate, and number of ports.

Consider the following aspects of your network to determine whether sharing data ports is the best intercluster connectivity solution:

- For a high-speed network, such as a 10-Gigabit Ethernet (10-GbE) network, a sufficient amount of local LAN bandwidth might be available to perform replication on the same 10-GbE ports that are used for data access. In many cases, the available WAN bandwidth is far less than 10 GbE LAN bandwidth.
- All nodes in the cluster might have to replicate data and share the available WAN bandwidth, making data port sharing more acceptable.
- Sharing ports for data and replication eliminates the extra port counts required to dedicate ports for replication.
- The maximum transmission unit (MTU) size of the replication network will be the same size as that used on the data network.
- Consider the data change rate and replication interval and whether the amount of data that must be replicated on each interval requires enough bandwidth that it might cause contention with data protocols if sharing data ports.
- When data ports for intercluster replication are shared, the intercluster LIFs can be migrated to any other intercluster-capable port on the same node to control the specific data port that is used for replication.

**Considerations when using dedicated ports**

When determining whether using a dedicated port for intercluster replication is the correct intercluster network solution, you should consider configurations and requirements such as LAN type, available WAN bandwidth, replication interval, change rate, and number of ports.

Consider the following aspects of your network to determine whether using a dedicated port is the best intercluster network solution:

- If the amount of available WAN bandwidth is similar to that of the LAN ports and the replication interval is such that replication occurs while regular client activity exists, then you should dedicate Ethernet ports for intercluster replication to avoid contention between replication and the data protocols.
• If the network utilization generated by the data protocols (CIFS, NFS, and iSCSI) is such that the network utilization is above 50 percent, then you should dedicate ports for replication to allow for nondegraded performance if a node failover occurs.

• When physical 10 GbE ports are used for data and replication, you can create VLAN ports for replication and dedicate the logical ports for intercluster replication. The bandwidth of the port is shared between all VLANs and the base port.

• Consider the data change rate and replication interval and whether the amount of data that must be replicated on each interval requires enough bandwidth that it might cause contention with data protocols if sharing data ports.

Configuring intercluster LIFs to share data ports

Configuring intercluster LIFs to share data ports enables you to use existing data ports to create intercluster networks for cluster peer relationships. Sharing data ports reduces the number of ports you might need for intercluster networking.

About this task

Creating intercluster LIFs that share data ports involves assigning LIFs to existing data ports. In this procedure, a two-node cluster exists in which each node has two data ports, e0c and e0d, and these data ports are in the default IPspace. These are the two data ports that are shared for intercluster replication. You must configure intercluster LIFs on the peer cluster before you can create cluster peer relationships. In your own environment, you replace the ports, networks, IP addresses, subnet masks, and subnets with those specific to your environment.

Steps

1. List the ports in the cluster by using the `network port show` command:

   Example

   ```
   cluster01::> network port show
   Node   Port      IPspace      Broadcast Domain Link   MTU    Admin/Oper
   ------ --------- ------------ ---------------- ----- ------- ------------
   cluster01-01
   e0a       Cluster      Cluster          up       1500  auto/1000
   e0b       Cluster      Cluster          up       1500  auto/1000
   e0c       Default      Default          up       1500  auto/1000
   e0d       Default      Default          up       1500  auto/1000
   cluster01-02
   e0a       Cluster      Cluster          up       1500  auto/1000
   e0b       Cluster      Cluster          up       1500  auto/1000
   e0c       Default      Default          up       1500  auto/1000
   e0d       Default      Default          up       1500  auto/1000
   ```

2. Create an intercluster LIF on the admin SVM cluster01 by using the `network interface create` command.

   Example

   This example uses the LIF naming convention `adminSVMname_icl` for the intercluster LIF:

   ```
   cluster01::> network interface create -vserver cluster01 -lif cluster01_icl01 -role intercluster
   -home-node cluster01-01 -home-port e0c -address 192.168.1.201 -netmask 255.255.255.0
   ```

   ```
   cluster01::> network interface create -vserver cluster01 -lif cluster01_icl02 -role intercluster
   -home-node cluster01-02 -home-port e0c -address 192.168.1.202 -netmask 255.255.255.0
   ```
3. Verify that the intercluster LIFs were created properly by using the `network interface show` command with the `-role intercluster` parameter:

**Example**

```
cluster01::> network interface show -role intercluster
```

<table>
<thead>
<tr>
<th>Logical Vserver</th>
<th>Status</th>
<th>Network Address/Mask</th>
<th>Current Node</th>
<th>Current Is Home</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster01_icl01</td>
<td>up/up</td>
<td>192.168.1.201/24</td>
<td>cluster01-01</td>
<td>e0c</td>
</tr>
<tr>
<td>cluster01_icl02</td>
<td>up/up</td>
<td>192.168.1.202/24</td>
<td>cluster01-02</td>
<td>e0c</td>
</tr>
</tbody>
</table>

4. Verify that the intercluster LIFs are configured to be redundant by using the `network interface show` command with the `-role intercluster` and `-failover` parameters.

**Example**

The LIFs in this example are assigned the e0c port on each node. If the e0c port fails, the LIF can fail over to the e0d port.

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

<table>
<thead>
<tr>
<th>Logical Vserver</th>
<th>Home:Port</th>
<th>Failover Policy</th>
<th>Failover Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster01_icl01</td>
<td>cluster01-01:e0c</td>
<td>local-only</td>
<td>192.168.1.201/24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster01_icl02</td>
<td>cluster01-02:e0c</td>
<td>local-only</td>
<td>192.168.1.201/24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Display the routes in the cluster by using the `network route show` command to determine whether intercluster routes are available or you must create them.

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

**Example**

In this example, no intercluster routes are available:

```
cluster01::> network route show
```

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0/0</td>
<td>192.168.0.1</td>
<td>20</td>
</tr>
<tr>
<td>0.0.0.0/0</td>
<td>192.168.0.1</td>
<td>10</td>
</tr>
</tbody>
</table>

6. If communication between intercluster LIFs in different clusters requires routing, create an intercluster route by using the `network route create` command.

The gateway of the new route should be on the same subnet as the intercluster LIF.

**Example**

In this example, 192.168.1.1 is the gateway address for the 192.168.1.0/24 network. If the destination is specified as 0.0.0.0/0, then it becomes a default route for the intercluster network.

```
cluster01::> network route create -vserver cluster01
```

7. Verify that you created the routes correctly by using the `network route show` command.
8. Repeat these steps on the cluster to which you want to connect.

Configuring intercluster LIFs to use dedicated intercluster ports

Configuring intercluster LIFs to use dedicated data ports allows greater bandwidth than using shared data ports on your intercluster networks for cluster peer relationships.

About this task

Creating intercluster LIFs that use dedicated ports involves creating a failover group for the dedicated ports and assigning LIFs to those ports. In this procedure, a two-node cluster exists in which each node has two data ports that you have added, e0e and e0f. These ports are ones you will dedicate for intercluster replication and currently are in the default IP space. These ports will be grouped together as targets for the intercluster LIFs you are configuring. You must configure intercluster LIFs on the peer cluster before you can create cluster peer relationships. In your own environment, you would replace the ports, networks, IP addresses, subnet masks, and subnets with those specific to your environment.

Steps

1. List the ports in the cluster by using `network port show` command.

Example

```
cluster01::> network port show
Speed (Mbps)
Node   Port      IPspace      Broadcast Domain Link   MTU    Admin/Oper
------ --------- ------------ ---------------- ----- ------- ------------
cluster01-01
e0a       Cluster      Cluster          up       1500  auto/1000
e0b       Cluster      Cluster          up       1500  auto/1000
e0c       Default      Default          up       1500  auto/1000
e0d       Default      Default          up       1500  auto/1000
e0e       Default      Default          up       1500  auto/1000
e0f       Default      Default          up       1500  auto/1000
cluster01-02
e0a       Cluster      Cluster          up       1500  auto/1000
e0b       Cluster      Cluster          up       1500  auto/1000
e0c       Default      Default          up       1500  auto/1000
e0d       Default      Default          up       1500  auto/1000
e0e       Default      Default          up       1500  auto/1000
e0f       Default      Default          up       1500  auto/1000
```

2. Determine whether any of the LIFs are using ports that are dedicated for replication by using the `network interface show` command.

Example

Ports e0e and e0f do not appear in the following output; therefore, they do not have any LIFs located on them:

```
cluster01::> network interface show -fields home-port,curr-port
vserver lif home-port curr-port
-------- ------------------------- ---------
Cluster cluster01-01_clus1   e0a   e0a
Cluster cluster01-01_clus2   e0b   e0b
Cluster cluster01-02_clus1   e0a   e0a
Cluster cluster01-02_clus2   e0b   e0b
```
3. If a LIF is using a port that you want dedicated to intercluster connectivity, migrate the LIF to a different port.

   a. Migrate the LIF to another port by using the `network interface migrate` command.

   **Example**
   
   The following example assumes that the data LIF named `cluster01_data01` uses port `e0e` and you want only an intercluster LIF to use that port:
   
   ```
   cluster01::> network interface migrate -vserver cluster01 -lif cluster01_data01 -dest-node cluster01-01 -dest-port e0d
   ```

   b. You might need to modify the migrated LIF home port to reflect the new port where the LIF should reside by using the `network interface modify` command:

   **Example**
   
   ```
   cluster01::> network interface modify -vserver cluster01 -lif cluster01_data01 -home-node cluster01-01 -home-port e0d
   ```

4. Group the ports that you will use for the intercluster LIFs by using the `network interface failover-groups create` command.

   **Example**
   
   ```
   cluster01::> network interface failover-groups create -vserver cluster01 -failover-group intercluster01 -targets cluster01-01:e0e,cluster01-01:e0f,cluster01-02:e0e,cluster01-02:e0f
   ```

5. Display the failover-group that you created by using the `network interface failover-groups show` command.

   **Example**
   
   ```
   cluster01::> network interface failover-groups show
   Failover Vserver Group Targets
   ---------------- ---------------- --------------------------------------------
   Cluster Cluster cluster01-01:e0a, cluster01-01:e0b, cluster01-02:e0a, cluster01-02:e0b
   cluster01 Default cluster01-01:e0c, cluster01-01:e0d, cluster01-02:e0c, cluster01-02:e0d, cluster01-01:e0e, cluster01-01:e0f, cluster01-02:e0e, cluster01-02:e0f
   intercluster01 cluster01-01:e0e, cluster01-01:e0f, cluster01-02:e0e, cluster01-02:e0f
   ```

6. Create an intercluster LIF on the admin SVM `cluster01` by using the `network interface create` command.

   **Example**
   
   This example uses the LIF naming convention `adminSVMname_icl#` for the intercluster LIF:
7. Verify that the intercluster LIFs were created properly by using the `network interface show` command.

**Example**

```
cluster01::> network interface show
Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
Cluster
cluster01-01_clus_1    up/up    192.168.0.xxx/24   cluster01-01  e0a     true
cluster01-01_clus_2    up/up    192.168.0.xxx/24   cluster01-01  e0b     true
cluster01-02_clus_1    up/up    192.168.0.xxx/24   cluster01-02  e0a     true
cluster01-02_clus_2    up/up    192.168.0.xxx/24   cluster01-02  e0b     true
cluster01               up/up    192.168.0.xxx/24   cluster01-01  e0c     true
cluster01_icl01        up/up    192.168.1.201/24   cluster01-01  e0e     true
cluster01_icl02        up/up    192.168.1.202/24   cluster01-02  e0e     true
cluster01-01_mgmt1     up/up    192.168.0.xxx/24   cluster01-01  e0c     true
cluster01-02_mgmt1     up/up    192.168.0.xxx/24   cluster01-02  e0c     true
```

8. Verify that the intercluster LIFs are configured for redundancy by using the `network interface show` command with the `role intercluster` and `-failover` parameters.

**Example**

```
The LIFs in this example are assigned the e0e home port on each node. If the e0e port fails, the
LIF can fail over to the e0f port.
```

```
cluster01::> network interface show -role intercluster -failover
Logical     Home             Failover        Failover
Vserver     Interface   Node:Port             Policy          Group
----------- ------------- --------------------- --------------- --------
cluster01-01
cluster01-01_icl01  cluster01-01:e0e     local-only      intercluster01
Failover Targets:  cluster01-01:e0f, cluster01-01:e0f
cluster01-01_icl02  cluster01-02:e0e     local-only      intercluster01
Failover Targets:  cluster01-02:e0e, cluster01-02:e0f
```

9. Display the routes in the cluster by using the `network route show` command to determine whether intercluster routes are available or you must create them.

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

**Example**

```
In this example, no intercluster routes are available:
```

```
cluster01::> network route show
Vserver   Destination     Gateway         Metric
--------- --------------- --------------- ----
Cluster   0.0.0.0/0       192.168.0.1     20
cluster01 0.0.0.0/0       192.168.0.1     10
```
10. If communication between intercluster LIFs in different clusters requires routing, create an intercluster route by using the `network route create` command.

   The gateway of the new route should be on the same subnet as the intercluster LIF.

   **Example**

   In this example, 192.168.1.1 is the gateway address for the 192.168.1.0/24 network. If the destination is specified as 0.0.0.0/0, then it becomes a default route for the intercluster network.

   ```bash
   cluster01::> network route create -vserver cluster01 -destination 0.0.0.0/0 -gateway 192.168.1.1 -metric 40
   ```

11. Verify that you created the routes correctly by using the `network route show` command.

   **Example**

   ```bash
   cluster01::> network route show
   Vserver    Destination     Gateway         Metric
   --------- --------------- --------------- -----
   Cluster   0.0.0.0/0       192.168.0.1     20
   cluster01 0.0.0.0/0       192.168.0.1     10
   0.0.0.0/0   192.168.1.1     40
   ```

12. Repeat these steps to configure intercluster networking in the peer cluster.

13. Verify that the ports have access to the proper subnets, VLANs, and so on.

   Dedicating ports for replication in one cluster does not require dedicating ports in all clusters; one cluster might use dedicated ports, while the other cluster shares data ports for intercluster replication.

**Configuring intercluster LIFs to use intercluster ports in their own networks**

You might need to direct intercluster traffic over a designated network. For example, you might want to connect to different clusters that are not reachable in the default IPspace. You can do this by moving ports to their own IPspaces and configuring intercluster LIFs.

**About this task**

In this procedure, a two-node cluster exists in which each node has two ports that you want to use for cluster peer relationships: e0e and e0f. These ports are ones you move from the default IPspace to their own IPspace. In the examples, these ports are configured with intercluster LIFs only, but you could configure ports to share data LIFs as well. In your own environment, you would replace the ports, networks, IP addresses, subnet masks, and subnets with those specific to your environment.

**Steps**

1. List the ports in the cluster by using `network port show` command:

   **Example**

   ```bash
   cluster01::> network port show
   Node  Port  IPspace  Broadcast Domain Link  MTU  Admin/Oper
   ------ -------- -------- ---------------- ----- ------------
   cluster01-01 e0a  Cluster  Cluster  up  1500  auto/1000
   e0b  Cluster  Cluster  up  1500  auto/1000
   e0c  Default  Default  up  1500  auto/1000
   e0d  Default  Default  up  1500  auto/1000
   e0e  Default  Default  up  1500  auto/1000
   e0f  Default  Default  up  1500  auto/1000
   cluster01-02 e0a  Cluster  Cluster  up  1500  auto/1000
   ```

   ```bash
   ```
2. Create a nondefault IPspace on the cluster on which you want to segregate the intercluster network by using the `network ipspace create` command:

**Example**

```bash
cluster01::> network ipspace create -ipspace ipspace-IC1
```

3. Determine whether any of the LIFs are using ports that are dedicated for replication by using the `network interface show` command.

**Example**

Ports e0e and e0f do not appear in the following output; therefore, they do not have any LIFs located on them:

```
cluster01::> network interface show -fields home-port,curr-port
vserver lif         home-port curr-port
------------------- -------------------
Cluster cluster01-01_clus1 e0a       e0a
Cluster cluster01-01_clus2 e0b       e0b
Cluster cluster01-02_clus1 e0a       e0a
Cluster cluster01-02_clus2 e0b       e0b
cluster01
    cluster_mgmt      e0c       e0c
cluster01
    cluster01-01_mgmt1 e0c       e0c
cluster01
    cluster01-02_mgmt1 e0c       e0c
```

4. If a LIF is using a port that you want dedicated to intercluster connectivity, migrate the LIF to a different port.

a. Migrate the LIF to another port by using the `network interface migrate` command.

**Example**

The following example assumes that the data LIF named `cluster01_data01` uses port e0e and you want only an intercluster LIF to use that port.

```bash
cluster01::> network interface migrate -vserver cluster01
-lif cluster01_data01 -dest-node cluster01-01 -dest-port e0d
```

b. You might need to modify the migrated LIF home port to reflect the new port where the LIF should reside by using the `network interface modify` command.

**Example**

```bash
cluster01::> network interface modify -vserver cluster01
-lif cluster01_data01 -home-node cluster01-01 -home-port e0d
```

5. Remove ports e0e and e0f from the default broadcast domain by using the `network port broadcast-domain remove-ports` command.

Ports must be removed from the broadcast domain before being added to another broadcast domain because a port cannot be in more than one broadcast domain at one time.
Example

```
cluster01::> network port broadcast-domain remove-ports -broadcast-domain Default -ports cluster01-01:e0e,cluster01-01:e0f,cluster01-02:e0e,cluster01-02:e0f
```

6. Verify that the ports are unassigned by using the network port show command.

Ports that are not assigned to a broadcast domain display - in the Broadcast Domain column.

Example

```
cluster01::> network port show
```

```

<table>
<thead>
<tr>
<th>Node</th>
<th>Port</th>
<th>IPspace</th>
<th>Broadcast Domain</th>
<th>Link</th>
<th>MTU</th>
<th>Admin/Oper</th>
</tr>
</thead>
<tbody>
<tr>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td>------------------</td>
<td>------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0a</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0b</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0c</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0d</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0e</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0f</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0g</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td>------------------</td>
<td>------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0a</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0b</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0c</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0d</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0e</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0f</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>1000</td>
<td>e0g</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td>------------------</td>
<td>------</td>
<td>-------</td>
<td>------------</td>
</tr>
</tbody>
</table>
```

7. Create the broadcast domain in the “ipspace-IC1” IPspace for the ports you want to dedicate to intercluster operations by using the network port broadcast-domain create command.

Part of the process of creating the broadcast domain for the ports is assigning the unassigned ports to the broadcast domain.

Example

This example creates the “ipspace-IC1-bd” broadcast domain in the “ipspace-IC1” IPspace:

```
cluster01::> network port broadcast-domain create -ipspace ipspace-IC1 -broadcast-domain ipspace-IC1-bd -mtu 1500 -ports cluster01-01:e0e,cluster01-01:e0f, cluster01-02:e0e,cluster01-02:e0f
```
8. Optional: Group the ports that you want to use for the intercluster LIFs by using the `network interface failover-groups create` command.

   If the intercluster connectivity requirements allow for the intercluster LIFs to use any port in the broadcast domain, then you do not need to create a separate failover group.

Example

```bash
cluster01::> network interface failover-groups create -vserver cluster01 -failover-group intercluster01 -targets cluster01-01:e0e,cluster01-01:e0f, cluster01-02:e0e,cluster01-02:e0f
```

9. Verify that the broadcast domain was created and the ports were assigned by using the `network port broadcast-domain show` command:

Example

```bash
cluster01::> network port broadcast-domain show

IPspace Broadcast                         Update
Name    Domain Name    MTU  Port List                     Status
Details  ---------------- ----------------- ----------------
---------- ----------- ------  -----------------------------
          ------- ----------- ------  -----------------------------
Cluster Cluster       9000
          cluster01-01:e0a        complete
          cluster01-01:e0b        complete
          cluster01-02:e0a        complete
          cluster01-02:e0b        complete
          cluster01-01:e0c        complete
          cluster01-01:e0d        complete
          cluster01-01:e0f        complete
          cluster01-01:e0g        complete
          cluster01-02:e0c        complete
          cluster01-02:e0d        complete
          cluster01-02:e0e        complete
          cluster01-02:e0f        complete
          cluster01-02:e0g        complete
Default Default       1500
          cluster01-01:e0c        complete
          cluster01-01:e0d        complete
          cluster01-01:e0f        complete
          cluster01-01:e0g        complete
          cluster01-02:e0c        complete
          cluster01-02:e0d        complete
          cluster01-02:e0e        complete
          cluster01-02:e0f        complete
          cluster01-02:e0g        complete
ipspace-IC1 ipspace-IC1-bd         1500
          cluster01-01:e0e        complete
          cluster01-01:e0f        complete
          cluster01-02:e0e        complete
          cluster01-02:e0f        complete
```

10. Display the failover-group that you created by using the `network interface failover-groups show` command:

Example

```bash
cluster01::> network interface failover-groups show

Vserver          Group            Targets
---------------- ----------------- --------------------------------------------
Cluster Cluster   Cluster           cluster01-01:e0a, cluster01-01:e0b,
          cluster01-02:e0a, cluster01-02:e0b
Default Default  cluster01-01:e0c, cluster01-01:e0d,
          cluster01-01:e0f, cluster01-01:e0g,
          cluster01-02:e0c, cluster01-02:e0d,
          cluster01-02:e0e, cluster01-02:e0f
ipspace-IC1 ipspace-IC1-bd       cluster01-01:e0e, cluster01-01:e0f,
          cluster01-02:e0e, cluster01-02:e0f
```
11. Create an intercluster LIF on the system SVM ipspace-IC1 by using the network interface create command.

**Example**

This example uses the LIF naming convention adminSVMname_icl# for the intercluster LIF:

```bash
cluster01::> network interface create -vserver ipspace-IC1 -lif cluster01_icl01 -role intercluster -home-node cluster01-01 -home-port e0e -address 192.168.1.201 -netmask 255.255.255.0 -failover-group intercluster01
cluster01::> network interface create -vserver ipspace-IC1 -lif cluster01_icl02 -role intercluster -home-node cluster01-02 -home-port e0e -address 192.168.1.202 -netmask 255.255.255.0 -failover-group intercluster01
```

12. Verify that the intercluster LIFs were created properly by using the network interface show command:

**Example**

```bash
cluster01::> network interface show
Logical    Status     Network            Current       Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
Cluster
cluster01-01_clus_1 up/up    192.168.0.xxx/24   cluster01-01  e0a     true
cluster01-01_clus_2 up/up    192.168.0.xxx/24   cluster01-01  e0b     true
cluster01-02_clus_1 up/up    192.168.0.xxx/24   cluster01-01  e0a     true
cluster01-02_clus_2 up/up    192.168.0.xxx/24   cluster01-01  e0b     true
cluster_mgmt up/up    192.168.0.xxx/24   cluster01-01  e0c     true
cluster01_icl01 up/up    192.168.1.201/24   cluster01-01  e0e     true
cluster01_icl02 up/up    192.168.1.202/24   cluster01-01  e0f     true
cluster01-01_mgmt1 up/up    192.168.0.xxx/24   cluster01-01  e0c     true
cluster01-02_mgmt1 up/up    192.168.0.xxx/24   cluster01-02  e0c     true
```

13. Verify that the intercluster LIFs are configured for redundancy by using the network interface show command with the -role intercluster and -failover parameters.

**Example**

The LIFs in this example are assigned the e0e home port on each node. If the e0e port fails, the LIF can fail over to the e0f port.

```bash
cluster01::> network interface show -role intercluster -failover
Logical    Home                  Failover        Failover
Vserver  Interface       Node:Port             Policy          Group
-------- --------------- --------------------- --------------- --------
cluster01-01 cluster01-01_icl01 cluster01-01:e0e   local-only      intercluster01
Failover Targets:  cluster01-01:e0e,
cluster01-01:e0f
cluster01-01_icl02 cluster01-02:e0e   local-only      intercluster01
Failover Targets:  cluster01-02:e0e,
cluster01-02:e0f
```

14. Display the routes in the cluster by using the network route show command to determine whether intercluster routes are available or you must create them.

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

**Example**

In this example, no intercluster routes are available:
15. If communication between intercluster LIFs in different clusters requires routing, create an intercluster route by using the `network route create` command.

   The gateway of the new route should be on the same subnet as the intercluster LIF.

   **Example**

   In this example, 192.168.1.1 is the gateway address for the 192.168.1.0/24 network. If the destination is specified as 0.0.0.0/0, then it becomes a default route for the intercluster network.

   ```bash
   cluster01::> network route create -vserver cluster01 -destination 0.0.0.0/0 -gateway 192.168.1.1 -metric 40
   ```

16. Verify that you created the routes correctly by using the `network route show` command:

   **Example**

   ```bash
   cluster01::> network route show
   Vserver  Destination     Gateway         Metric
            --------------- --------------- -----
   Cluster  0.0.0.0/0       192.168.0.1     20
   cluster01 0.0.0.0/0       192.168.0.1     10
   0.0.0.0/0       192.168.1.1     40
   ```

17. Repeat these steps to configure intercluster networking in the peer cluster.

   The peer cluster can have its intercluster LIFs in its Default IPspace, or any other IPspace, as long as there is connectivity between the intercluster LIFs of the two clusters.

18. Verify that the ports have access to the proper subnets, VLANs, and so on.

   Dedicating ports for replication in one cluster does not require dedicating ports in all clusters; one cluster might use dedicated ports, while the other cluster shares data ports for intercluster replication.

**Creating the cluster peer relationship**

You create the cluster peer relationship using a set of intercluster logical interfaces to make information about one cluster available to the other cluster for use in cluster peering applications.

**Before you begin**

- Intercluster LIFs should be created in the IPspaces of both clusters you want to peer.
- You should ensure that the intercluster LIFs of the clusters can route to each other.
- If there are different administrators for each cluster, the passphrase used to authenticate the cluster peer relationship should be agreed upon.

**About this task**

If you created intercluster LIFs in a nondefault IPspace, you need to designate the IPspace when you create the cluster peer.
Steps

1. Create the cluster peer relationship on each cluster by using the `cluster peer create` command.

   The passphrase that you use is not displayed as you type it.

   If you created a nondefault IPspace to designate intercluster connectivity, you use the `ipspace` parameter to select that IPspace.

Example

In the following example, cluster01 is peered with a remote cluster named cluster02. Cluster01 is a two-node cluster that has one intercluster LIF per node. The IP addresses of the intercluster LIFs created in cluster01 are 192.168.2.201 and 192.168.2.202. Similarly, cluster02 is a two-node cluster that has one intercluster LIF per node. The IP addresses of the intercluster LIFs created in cluster02 are 192.168.2.203 and 192.168.2.204. These IP addresses are used to create the cluster peer relationship.

```
cluster01::> cluster peer create -peer-addrs 192.168.2.203,192.168.2.204
Please type the passphrase:
Please type the passphrase again:
```

```
cluster02::> cluster peer create -peer-addrs 192.168.2.201,192.168.2.202
Please type the passphrase:
Please type the passphrase again:
```

If DNS is configured to resolve host names for the intercluster IP addresses, you can use host names in the `-peer-addrs` option. It is not likely that intercluster IP addresses frequently change; however, using host names allows intercluster IP addresses to change without having to modify the cluster peer relationship.

Example

In the following example, an IPspace called IP01A was created on cluster01 for intercluster connectivity. The IP addresses used in the previous example are used in this example to create the cluster peer relationship.

```
cluster01::> cluster peer create -peer-addrs 192.168.2.203,192.168.2.204 -ipspace IP01A
Please type the passphrase:
Please type the passphrase again:
```

```
cluster02::> cluster peer create -peer-addrs 192.168.2.201,192.168.2.202
Please type the passphrase:
Please type the passphrase again:
```

2. Display the cluster peer relationship by using the `cluster peer show` command with the `-instance` parameter.

   Displaying the cluster peer relationship verifies that the relationship was established successfully.
3. Preview the health of the nodes in the peer cluster by using the `cluster peer health show` command.

Previewing the health checks the connectivity and status of the nodes on the peer cluster.

### Related tasks

- **Configuring intercluster LIFs to share data ports** on page 61
- **Configuring intercluster LIFs to use dedicated intercluster ports** on page 63

### Modifying a cluster peer relationship

You can modify a cluster peer relationship if the name of the cluster you connected to, the logical interface you used, or the IP address you used when creating the cluster peer relationship changes. For example, the IP address of the cluster you used when creating the relationship changed.

#### Step

1. To change the configuration of a cluster peer relationship, use the `cluster peer modify` command.

   The following example changes the IP address of the cluster peer configuration of a cluster named `cluster_b` to 172.19.7.3:

   ```bash
   node::> cluster peer modify -cluster cluster_b -stable-addrs 172.19.7.3
   ```
Deleting a cluster peering relationship

You can delete a cluster peering relationship if the relationship is no longer needed. You must delete the cluster peering relationship from each of the clusters in the relationship.

Before you begin

All Storage Virtual Machine (SVM) peer relationships between the two cluster peers must have been deleted.

About this task

This procedure assumes that you are the administrator of only one of the clusters in the cluster peering relationship.

Steps

1. Delete the cluster peering relationship from the cluster of which you are the administrator by using the `cluster peer delete` command.

   Example
   
   The following example deletes the cluster peering relationship with the cluster2 cluster from the cluster1 cluster:
   
   ```
   cluster1::> cluster peer delete -cluster cluster2
   ```

2. Ask the administrator of the other cluster to delete the cluster peering relationship from the other cluster by using the `cluster peer delete` command.

   Example
   
   The following example deletes the cluster peering relationship with the cluster1 cluster from the cluster2 cluster:
   
   ```
   cluster2::> cluster peer delete -cluster cluster1
   ```

Related tasks

*Deleting an SVM peer relationship* on page 82

Managing SVM peer relationships

A cluster administrator can create and manage SVM peer relationships between two Storage Virtual Machines (SVMs, formerly known as Vservers) either existing within a cluster (intracluster) or in peered clusters (intercluster) to provide an infrastructure for peering applications, such as SnapMirror.

Peered clusters and peered SVMs can be managed either by the same cluster administrator or different cluster administrators.

The cluster administrator can perform the following SVM peer management tasks:

- Creating SVM peer relationships
- Accepting SVM peer relationships
- Rejecting SVM peer relationships
- Suspending SVM peer relationships
- Resuming SVM peer relationships
- Modifying SVM peering applications on the SVM peer relationships
- Deleting SVM peer relationships
- Viewing SVM peer relationships
- Setting up SnapMirror relationships between volumes of the peered SVMs

**Note:** You cannot set up a load-sharing SnapMirror relationship between volumes of intercluster SVM peers.

An SVM administrator can perform only the following SVM peer management tasks:

- Viewing SVM peer relationships to identify the peered SVMs
- Setting up SnapMirror relationships, such as a data protection relationship (DP), SnapVault relationship (XDP), and transition relationship (TDP), between volumes of the peered SVMs

**Note:** The Data ONTAP command-line interface (CLI) continues to use the term *Vserver* in the output, and *vserver* as a command or parameter name has not changed.

### What an SVM peer relationship is

An SVM peer relationship is an authorization infrastructure that enables a cluster administrator to set up peering applications such as SnapMirror relationships between SVMs either existing within a cluster (intracluster) or in the peered clusters (intercluster). Only a cluster administrator can set up SVM peer relationships.

The following illustration shows the intercluster and intracluster SVM peer relationships:

![SVM Peer Relationships](image)

The SVM peer infrastructure enables you to set up a backup and recovery mechanism between SVMs. You can set up a mirroring relationship at the volume level between peered SVMs. If a volume in the SVM becomes unavailable, the cluster administrator or SVM administrator can configure the respective mirrored volume of the peered SVM to serve data.

One SVM can be peered with multiple SVMs within a cluster or across clusters.

You can set up only SnapMirror data protection (DP) and SnapVault (XDP) relationships by using the SVM peer infrastructure.
States of SVM peer relationships

SVM peer relationships can be in different states depending on the operation performed on the SVM peer relationship. You must be aware of the states of the SVM peer relationship to perform other operations such as SnapMirror data transfer between peered SVMs.

The following table lists the different states of an SVM peer relationship and helps you understand when an SVM peer relationship is in a particular state:

<table>
<thead>
<tr>
<th>SVM peer relationship is in...</th>
<th>When...</th>
</tr>
</thead>
<tbody>
<tr>
<td>initializing state on the local cluster</td>
<td>• The local cluster is communicating with the peer cluster for initializing the SVM peer relationship</td>
</tr>
<tr>
<td>initiated state on the local cluster pending state on the peered cluster</td>
<td>• An intercluster SVM peer relationship is requested from the local cluster</td>
</tr>
<tr>
<td>peered state on the local and peered clusters</td>
<td>• An intercluster SVM peer relationship is accepted from the peered cluster</td>
</tr>
<tr>
<td></td>
<td>• An intracluster SVM peer relationship is established</td>
</tr>
<tr>
<td></td>
<td>• An intercluster or intracluster SVM peer relationship is resumed</td>
</tr>
<tr>
<td>rejected state on the local cluster</td>
<td>• An intercluster SVM peer relationship is rejected from the peered cluster</td>
</tr>
<tr>
<td>suspended state on the local and peered clusters</td>
<td>• An intercluster or intracluster SVM peer relationship is suspended from the local or peered cluster</td>
</tr>
</tbody>
</table>

Creating an SVM peer relationship

A cluster administrator can create a Storage Virtual Machine (SVM) peer relationship to provide an authorization infrastructure for running SVM peering applications between two SVMs by using the `vserver peer create` command. You can create an SVM peer relationship between two SVMs either in a single cluster (intracluster) or in peered clusters (intercluster).

Before you begin

- If you want to create an intercluster SVM peer relationship, you must have ensured that both the clusters are peered with each other.
- The names of the SVMs in the peered clusters must be unique across the clusters to be peered and any other clusters with which either of the clusters are individually peered.
  If the SVMs do not have unique names, you must rename one of the SVMs by using the `vserver rename` command.
  For example, consider two clusters, cluster A and cluster B, that are peered with cluster C. Clusters A and cluster B must not have SVMs with identical names even though cluster A and cluster B are not peered. You must rename one of the SVMs, if there are SVMs with identical names.
- The admin state of the SVMs to be peered must not be in initializing or deleting state.
• If any previously attempted SVM peer relationship between the same SVMs is in the rejected state, you must have deleted these SVM peer relationships.

About this task
• Peered clusters can be managed by a single cluster administrator or different cluster administrators.
• You can specify the applications that will communicate over the peer relationship when you create an SVM peer relationship.
  If you do not specify the application for the peer relationship, such as snapmirror, an SVM administrator cannot perform operations related to the applications between the peered SVMs.
• For SVMs with FlexVol volumes, you can create intercluster and intracluster SVM peer relationships.
• For SVMs with Infinite Volume, you can create only intercluster SVM peer relationships.
• You cannot create an SVM peer relationship between SVMs with FlexVol volumes and SVMs with Infinite Volume.
• You can create multiple SVM peer relationships simultaneously either by using different SSH sessions or by using a script.
  Note: It is best to create not more than five SVM peer relationships simultaneously to avoid any performance degradation.

Choices
• Creating an intercluster SVM peer relationship on page 77
• Creating an intracluster SVM peer relationship on page 78

Creating an intercluster SVM peer relationship
You can create intercluster SVM peer relationships between two clusters to provide the infrastructure for use cases such as intercluster volume SnapMirror configurations. You can create intercluster SVM peer relationships between two clusters to provide the infrastructure for use cases such as intercluster volume SnapMirror configurations and SVM disaster recovery.

Before you begin
The two clusters must already be peered.

Steps
1. Use the vserver peer create command to create an SVM peer relationship.

Example
The following command creates an intercluster SVM peer relationship between vs1.example0.com (on cluster1) and vs3.example0.com (on cluster2):

```
cluster1::> vserver peer create -vserver vs1.example0.com -peer-vserver vs3.example0.com -applications snapmirror -peer-cluster cluster2
Info: [Job 43] 'vserver peer create' job queued
```

The intercluster SVM peer relationship is in initiated state.

2. Use the vserver peer show-all command to view the status and other details of the SVM peer relationship.
Example

```
cluster1::> vserver peer show-all

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Peer</th>
<th>Peer Cluster</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1.example0.com</td>
<td>vs3.example0.com</td>
<td>Cluster2</td>
<td>snapmirror</td>
</tr>
</tbody>
</table>
```

For more information about this command, see the man pages.

After you finish

You must inform the cluster administrator of the peered cluster about the SVM peer request for the authentication to be completed.

The SVM peer relationship is not established until the cluster administrator of the peered cluster accepts the SVM peer request.

Related tasks

- [Accepting an SVM peer relationship](#) on page 79

Creating an intracluster SVM peer relationship

You can create SVM peer relationships between SVMs within a cluster for operations such as backup of SVM data within a cluster.

About this task

You cannot create intracluster SVM peer relationships for SVMs with Infinite Volumes.

Steps

1. Use the `vserver peer create` command to create an SVM peer relationship.

Example

The following command creates an intracluster SVM peer relationship between the SVMs vs4.example1.com and vs0.example1.com, both residing on cluster2:

```
cluster2::> vserver peer create -vserver vs4.example1.com -peer-vserver vs0.example1.com -applications snapmirror

Info: 'vserver peer create' command is successful.
```

An intracluster SVM peer relationship is created and is in peered state. Authentication is not required because the cluster is managed by a single cluster administrator.

2. Use the `vserver peer show-all` command to view the status and other details of the SVM peer relationship.

Example

```
cluster2::> vserver peer show-all

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Peer</th>
<th>Peer Cluster</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs4.example1.com</td>
<td>vs0.example1.com</td>
<td>cluster2</td>
<td>snapmirror</td>
</tr>
<tr>
<td>vs0.example1.com</td>
<td>vs4.example1.com</td>
<td>cluster2</td>
<td>snapmirror</td>
</tr>
</tbody>
</table>
```
Accepting an SVM peer relationship

When a cluster administrator creates an intercluster SVM peer relationship, the cluster administrator of the remote cluster can accept the SVM peer request to establish the peer relationship between the SVMs by using the \texttt{vserver peer accept} command.

About this task

Peered clusters can be managed by a single administrator or different cluster administrators. If a single cluster administrator is managing the peered clusters, the cluster administrator has to accept the SVM peer request on the peered cluster. If different administrators are managing the peered clusters, the cluster administrator who initiates the SVM peer request has to notify the cluster administrator of the peered cluster about the incoming SVM peer request through any channel such as email.

Steps

1. Use the \texttt{vserver peer show} command to view the SVM peer requests.

   Example

   The following example shows how to view the SVM peer requests on cluster2:

   \begin{verbatim}
   cluster2::> vserver peer show
   Vserver          Peer         Peer
   -----------     -----------     -----------
   vs3.example0.com vs1.example0.com pending
   \end{verbatim}

2. Use the \texttt{vserver peer accept} command to accept the SVM peer request and establish the SVM peer relationship.

   Example

   The following example shows how to accept an incoming SVM peer request to establish an SVM peer relationship between vs1.example0.com and vs3.example0.com on cluster1 and cluster2 respectively:

   \begin{verbatim}
   cluster2::> vserver peer accept -vserver vs3.example0.com -peer-vserver vs1.example0.com
   Info: [Job 46] 'vserver peer accept' job queued
   \end{verbatim}

   The SVM peer relationship is established and state is peered.

3. Use the \texttt{vserver peer show} command on either of the peered clusters to view the state of the SVM peer relationship.

   Example

   The following example shows how to view to state of the SVM peer relationships:

   \begin{verbatim}
   cluster2::> vserver peer show
   Vserver          Peer         Peer
   -----------     -----------     -----------
   vs3.example0.com vs1.example0.com peered
   \end{verbatim}
For more information about these commands, see the man pages.

**Result**

A cluster or SVM administrator can establish peering applications such as SnapMirror between the peered SVMs.

**Rejecting an SVM peer relationship**

When a cluster administrator creates an intercluster SVM peer relationship, the cluster administrator of the peered cluster can reject the SVM peer request to prevent peer relationship between the SVM by using the `vserver peer reject` command.

**About this task**

If the SVM peer request is initiated with an unauthorized SVM, then the cluster administrator of the peered cluster can reject the relationship. Other peering operations cannot be performed on the rejected peering relationship.

**Steps**

1. Use the `vserver peer show` command to view the SVM peer requests on the peered cluster.

   **Example**

   The following example shows how to view the SVM peer requests on cluster2:

   ```bash
   cluster2::> vserver peer show
   Peer               Peer
   Vserver          Vserver            State
   -----------      -----------        ------------
   vs5.example0.com vs1.example0.com    pending
   ```

2. Use the `vserver peer reject` command to reject the SVM peer request.

   **Example**

   The following example illustrates how to reject an incoming SVM peer request between vs1.example0.com and vs5.example0.com on cluster1 and cluster2 respectively:

   ```bash
   cluster2::> vserver peer reject -vserver vs5.example0.com -peer-vserver vs1.example0.com
   Info: [Job 48] 'vserver peer reject' job queued
   ```

   The SVM peer relationship is in rejected state.

3. Use the `vserver peer show` command on the cluster from which the SVM peer request was created to view the state of the SVM peer relationship.

   **Example**

   The following example shows how to view to state of the SVM peer relationships:

   ```bash
   cluster1::> vserver peer show
   Peer               Peer
   Vserver          Vserver            State
   -----------      -----------        ------------
   vs1.example0.com vs5.example0.com   rejected
   ```
4. Use the \texttt{vserver peer delete} command to delete the rejected SVM peer requests because when you create the SVM relationship between the same SVM again, it fails.

**Example**

The following example shows how to delete the rejected SVM peer requests:

```
cluster1::> vserver peer delete -vserver vs1.example0.com -peer-vserver vs5.example0.com
Info: 'vserver peer delete' command is successful.
```

For more information about these commands, see the man pages.

### Modifying the peering application on an SVM peer relationship

A cluster administrator can modify an SVM peering application running on the SVM peer relationship by using the \texttt{vserver peer modify} command. The SVM peering relationship can have SnapMirror, FileCopy, or no application.

**About this task**

The SVM peer relationship must have peering application as \texttt{snapmirror} for all SnapMirror operations between the peered SVMs or \texttt{file-copy} for all the FileCopy related operations between the peered SVMs.

**Steps**

1. Use the \texttt{vserver peer modify} command to modify the application on the SVM peer relationship.

**Example**

The following command modifies the application on the SVM peer relationship:

```
cluster2::>vserver peer modify -vserver vs4.example.com -peer-vserver vs0.example.com -applications snapmirror
Warning: The following applications were enabled between Vserver "vs4.example.com" and peer Vserver "vs0.example.com": file-copy, snapmirror. The following applications will be removed: file-copy. Any operations related to the removed application in the context of this Vserver peer relationship will be disabled.
Do you want to continue? {y|n}: y
Info: 'vserver peer modify' command is successful.
```

2. Use the \texttt{vserver peer show-all} to view the applications running on the SVM peer relationship.

**Example**

The following command displays the applications running on the SVM peer relationship:

```
cluster2::> vserver peer show-all
Vserver            Vserver           State      Peer Cluster   Applications
-----------        -----------       -------    ------------  ------------
vs4.example1.com   vs0.example1.com  peered     cluster2        snapmirror
```
Deleting an SVM peer relationship

A cluster administrator can delete the Storage Virtual Machine (SVM) peer relationship by using the `vserver peer delete` command when the relationship between two SVMs is no longer required.

**Before you begin**

The SnapMirror relationship defined on the SVM peer relationship must be deleted.

**About this task**

If one of the peered clusters is running clustered Data ONTAP 8.2 or 8.2.1, then you must delete the SVM peer relationship from both the peered clusters.

You can delete multiple SVM peer relationships simultaneously either by using different SSH sessions or by using a script.

**Note:** It is best to delete not more than five SVM peer relationships simultaneously to avoid any performance degradation.

**Steps**

1. Use the `vserver peer delete` command on one of the peered clusters to delete the SVM peer relationship.

   **Example**

   The following command deletes the SVM peer relationship from both the clusters:

   ```
   cluster1::> vserver peer delete -vserver vs1.example0.com -peer-vserver vs3.example0.com
   Info: [Job 47] 'vserver peer delete' job queued
   ```

2. If the `vserver peer delete` command fails due to unavailability of one of the peered clusters, choose one of the following actions:

   - Establish the network connectivity between the two clusters and use the `vserver peer delete` command to delete the SVM peer relationship (recommended).
   - Use the `vserver peer delete` command with the `-force` option on both the local and peered clusters to delete the SVM peer relationship if the cluster peer relationship is not reestablished.

3. Use the `vserver peer show` command on both the clusters to verify that the deleted SVM peer relationship is not displayed.

   **Example**

   ```
   cluster1::> vserver peer show
   
   Peer               Peer
   Vserver         Vserver            State
   -----------       -----------        ------------
   vs1.example0.com  vs3.example0.com    peered
   ```

4. If any SVM peer relationship is in the `deleted` state, delete that SVM peer relationship again by using the `vserver peer delete` command.
Suspending an SVM peer relationship

A cluster administrator can suspend an established SVM peer relationship whenever needed by using the `vserver peer suspend` command. For example, during the maintenance period, you might want to suspend the SVM peer relationship.

About this task

When you suspend the SVM peer relationship, any SnapMirror data transfer that was initiated before suspending an SVM peer relationship is not affected and the operation is completed. Any data transfer that was scheduled to run during suspension period will not get initiated.

Steps

1. Use the `vserver peer suspend` command on either of the peered cluster to suspend an active SVM peer relationship.

Example

The following example shows how to suspend an SVM peer relationship:

```
cluster2::> vserver peer suspend -vserver vs4.example1.com -peer-vserver vs0.example1.com
Info: [Job 50] 'vserver peer suspend' job queued
```

The SVM peer relationship is in suspended state.

2. Use the `vserver peer show` command to verify the status of the SVM peer relationship.

Example

The following example shows how to verify the status of the SVM peer relationship:

```
cluster2::> vserver peer show

       Peer Vserver          Peer
---------- -------        ----------
vs4.example1.com vs0.example1.com suspended
```

For more information about these commands, see the man pages.

Resuming an SVM peer relationship

A cluster administrator can resume a suspended SVM peer relationship by using the `vserver peer resume` command. For example, after the maintenance is complete, you can resume the suspended SVM peering relationship.

About this task

Any SnapMirror data transfer that was scheduled to run during the suspension period will not get initiated when you resume the SVM peer relationship. You must manually initiate the data transfer.
Steps

1. Use the `vserver peer resume` command to resume a suspended SVM peer relationship from either of the peered clusters.

   **Example**

   The following example shows how to resume a suspended SVM peer relationship:

   ```
   cluster1::> vserver peer resume -vserver vs4.example1.com -peer-vserver vs0.example1.com
   Info: [Job 76] 'vserver peer resume' job queued
   ```

   The SVM peer relationship is in peered state.

2. Use the `vserver peer show` command to verify the status of the SVM peer relationship.

   **Example**

   The following example shows how to verify the status of the SVM peer relationship:

   ```
   cluster1::> vserver peer show
   Peer              Peer
   Vserver           Vserver           State
   ----------------- ----------------- ---------
   vs4.example1.com  vs0.example1.com   peered
   ```

   For more information about these commands, see the man pages.

Displaying information about SVM peer relationships

Peer Storage Virtual Machines (SVMs) are fully functional SVMs which could be either local or remote. Cluster administrators and SVM administrators can view the peers of the SVM to set up peering applications such as SnapMirror between volumes of the peer SVMs by using the `vserver peer show` command.

**About this task**

You can also view the status of the SVM peer relationships.

**Step**

1. Use the `vserver peer show` command to view the peered SVMs and the state of the SVM peer relationship.

   **Example**

   The following example shows how to view the information about peered SVMs:

   ```
   vs1.example.com::> vserver peer show
   Vserver              Peer              Peer
   -------------------  ------------------- ---------
   vs1.example0.com    vs5.example0.com   peered
   vs1.example0.com    vs3.example0.com   peered
   ```

   For more information about this command, see the man pages.
Providing disaster recovery using mirroring technology

Stored data is susceptible to disaster, either through hardware failure or environmental catastrophe. You can use mirroring technology to create an identical second set of data to replace the primary set of data, should something happen to the primary set of data.

Creating a data protection mirror copy for FlexVol volumes

You can protect data by replicating it to data protection mirror copies. You can use data protection mirror copies to recover data when a disaster occurs.

Before you begin

• You must have installed a SnapMirror license on both the source and destination cluster.
• You must have created the cluster and Storage Virtual Machine (SVM) peering relationship. To learn about creating cluster and SVM peering, see the Clustered Data ONTAP System Administration Guide for Cluster Administrators.

About this task

You can create data protection mirror copies in a cluster using FlexVol volumes only.

Steps

1. Create a destination volume on the destination SVM that will become the data protection mirror copy by using the `volume create` command.

   Example

   The following command creates a data protection mirror volume named `dept_eng_dr_mirror1` on SVM `vs1.example.com`. The destination volume is located on an aggregate named `aggr3`. The destination volume is also on SVM `vs1.example.com`.

   ```bash
   vs1::> vol create -volume dept_eng_dr_mirror1 -aggregate aggr3 -size 20MB -type DP
   ```

   If you are creating a data protection mirror copy on an SVM peer, the destination volume is created on the SVM peer:

   ```bash
   vs2::> volume create -volume dept_eng_dr_mirror1 -aggregate aggr3 -size 20MB -type DP
   ```

2. Create a data protection mirror relationship by using the `snapmirror create` command.

   Example

   The following command creates a data protection relation with the destination volume named `dept_eng_dp_mirror2` of the source volume named `dept_eng`. The SVM is named `vs1`.

   ```bash
   vs1::> snapmirror create -destination-path vs1.example.com:dept_eng_dp_mirror2 -source-path vs1.example.com:dept_eng -type DP -schedule 5min
   ```
If you are creating the data protection mirror relationship with the destination volume on an SVM peer, you create the data protection mirror relationship from the SVM that contains the destination volume. For example, if the destination volume were on the SVM peer named vs2, the command to create the data protection mirror relationship is as follows:

```
vs2::> snapmirror create -destination-path vs2.example.com:dept_eng_dp_mirror2 -source-path vs1.example.com:dept_eng -type DP -schedule 5min
```

Data ONTAP creates the data protection mirror relationship, but the relationship is left in an uninitialized state.

3. Initialize the data protection mirror copy by using the `snapmirror initialize` command.

**Example**

The following command initializes a data protection mirror copy named dept_eng_dp_mirror2 of a source volume named dept_eng. The source volume and the data protection mirror copy are on the same SVM named vs1.example.com.

```
vs1::> snapmirror initialize -destination-path vs1.example.com:dept_eng_dp_mirror2
```

If you are initializing the data protection mirror relationship with the destination volume on an SVM peer, you must initialize the data protection mirror relationship from the SVM that contains the destination volume. For example, if the destination volume of the single SVM example were on an SVM peer named vs2.example.com, the command to create the data protection mirror relationship is as follows:

```
vs2::> snapmirror initialize -destination-path vs2.example.com:dept_eng_dp_mirror2
```

**Correcting a SnapMirror initialization failure**

A SnapMirror initialization can fail with the error message `Volume volume_name is restricted` if a previous initialization attempt failed. The initialization fails because the destination volume was restricted in the first failed attempt.

**About this task**

You correct a SnapMirror initialization failure by changing the state of the destination volume from `restricted` to `online` and then running another initialization attempt.

**Steps**

1. Change the state of the destination volume by using the `volume modify` command with the `-state` parameter.

   **Example**

   ```
   vs2::> volume modify -vserver vs2.example.com -volume vol3_dst -state online
   ```

2. Initialize the SnapMirror relationship by using the `snapmirror initialize` command.
Creating a version-flexible SnapMirror relationship

You can protect data by replicating selected Snapshot copies to a version-flexible SnapMirror destination volume on another Storage Virtual Machine (SVM) or cluster.

Before you begin

- You must have cluster administrator privileges to perform this task for a cluster, and SVM administrator privileges to perform this task for an SVM.

- If the primary and secondary volumes are in different SVMs, the SVMs must be in a peer relationship.
  - If the primary and secondary volumes are in different clusters, the clusters must be in a peer relationship.
  - For information about creating peer relationships, see the Clustered Data ONTAP System Administration Guide for Cluster Administrators.

- A version-flexible SnapMirror policy must exist.
  - You must either create a version-flexible SnapMirror policy or use an existing one (named MirrorAllSnapShots, MirrorLatest, or MirrorAndVault).
  - Version-flexible SnapMirror supports two policy types, async-mirror and mirror-vault. The policy type async-mirror does not support rules with user-defined Snapshot labels. It only allows two predefined labels “sm_created” and “all_source_snapshots”. The policy rules support only two combinations of these labels, either just “sm_created” or both “sm_created” and “all_source_snapshots”. The policy type mirror-vault always has a rule with the label “sm_created”. In addition, users can add rules with any user-defined labels.

- The Snapshot policy assigned to the primary volume must include the snapmirror-label attribute.
  - You set the snapmirror-label attribute for the set of Snapshot copies that you want backed up to the version-flexible SnapMirror secondary volume. Other Snapshot copies on the primary volume are ignored by the version-flexible SnapMirror relationship.

- Your work environment must be able to accommodate the time it might take to transfer a baseline Snapshot copy with a large amount of data.

Steps

1. On the destination SVM, create a SnapMirror destination volume with a volume type DP.
   - For information about creating a FlexVol volume, see the Clustered Data ONTAP System Administration Guide for Cluster Administrators.

2. On the source SVM, create a Snapshot copy policy that contains the schedule of when Snapshot copies with snapmirror-label attributes occur.
   - You can use the volume snapshot policy create command with the snapmirror-label parameter.

   Example

   The following command creates a Snapshot copy policy called “keep-more-snapshot”:
The name specified in the `snapmirror-label` attribute for the new Snapshot policy must match the `snapmirror-label` attribute that is specified in the version-flexible SnapMirror policy. This ensures that all subsequent Snapshot copies created on the primary volume have labels that are recognized by the version-flexible SnapMirror policy.

The default Snapshot copy policy has two `snapmirror-label` attributes associated with it, daily and weekly.

3. Create a version-flexible SnapMirror policy by using the `snapmirror policy create` command.

Example

The following command creates a version-flexible SnapMirror policy called “vserverB-DR-policy” that will be used for SnapMirror style disaster recovery in which only the SnapMirror created Snapshot copy is transferred:

```
vserverB::> snapmirror policy create -vserver vserverB -policy vserverB-DR-policy -policy-type async-mirror -comment "DR policy"
```

Example

The following command creates a version-flexible SnapMirror policy called “vserverB-asyncDR-policy” that will be used for SnapMirror style disaster recovery in which all source Snapshot copies are transferred:

```
vserverB::> snapmirror policy create -vserver vserverB -policy vserverB-asyncDR-policy -policy-type async-mirror -comment "Async DR policy"
```

Example

The following command creates a version-flexible SnapMirror policy called “vserverB-SM-SV-policy” that will be used for both SnapMirror and SnapVault relationships in the same volume:

```
vserverB::> snapmirror policy create -vserver vserverB -policy vserverB-SM-SV-policy -policy-type mirror-vault -comment "SnapMirror and SnapVault combo policy"
```

4. Add the `snapmirror-label` attribute to the version-flexible SnapMirror policy you created by using the `snapmirror policy add-rule` command.

Example

No added rule is required because the SnapMirror policy transfers on the SnapMirror created Snapshot copies and keeps one Snapshot copy.
The following command adds a rule to the vserverB-asyncDR-policy to transfer Snapshot copies with the “sm_created” and “all_source_snapshots” snapmirror-label attribute and to keep 1 Snapshot copy of each:

```
vserverB::> snapmirror policy add-rule -vserver vserverB -policy vserverB-asyncDR-policy -snapmirror-label all_source_snapshots -keep 1
```

The following command adds rules to the vserverB-SM-SV-policy to transfer Snapshot copies with the “sm_created”, “daily”, and “weekly” snapmirror-label attributes and to keep 1 “sm_created” Snapshot copy, 20 “daily” Snapshot copies, and 26 “weekly” Snapshot copies:

```
vserverB::> snapmirror policy add-rule -vserver vserverB -policy vserverB-SM-SV-policy -snapmirror-label sm_created -keep 1
vserverB::> snapmirror policy add-rule -vserver vserverB -policy vserverB-DR-policy -snapmirror-label daily -keep 20
vserverB::> snapmirror policy add-rule -vserver vserverB -policy vserverB-DR-policy -snapmirror-label weekly -keep 26
```

5. On the destination SVM, create a version-flexible SnapMirror relationship and assign a version-flexible SnapMirror policy by using the `snapmirror create` command with the type XDP parameter and the policy parameter.

In the path specification, a single name is interpreted as a volume name in the SVM from which the command is executed. To specify a volume in a different SVM or in a different cluster, you must specify the full path name.

```
vserverB::> snapmirror create -source-path vserverA:srcvolA -destination-path vserverB:dstvolB -type XDP -policy vserverB-DR-policy -schedule daily
```

If you are creating version-flexible SnapMirror relationships using one of the other version-flexible SnapMirror policies, the command syntax is the same but you replace the policy name with the policy that you want to use.

6. On the destination SVM, initialize the version-flexible SnapMirror relationship by using the `snapmirror initialize` command to start a baseline transfer.

The command creates a new Snapshot copy that is transferred to the destination volume and used as a baseline for subsequent incremental Snapshot copies. The command does not use any Snapshot copies that currently exist on the source volume.

**Note:** Creating a baseline for a large amount of data might take hours.

```
vserverB::> snapmirror initialize -source-path vserverA:srcvolA -destination-path vserverB:dstvolB -type XDP -policy vserverB-DR-policy
```

The following command begins the relationship initialization by creating and transferring a baseline Snapshot copy to the destination volume “dstvolB” on SVM “vserverB”:
Related concepts

Managing Snapshot policies on page 40

Considerations when using version-flexible SnapMirror relationships

You should understand the conditions in which your use of Version-flexible SnapMirror relationships will best meet your needs.

The following considerations should be understood before creating version-flexible SnapMirror relationships:

• The schedule frequency for replicating volumes must be more than 60 minutes.

• You should not use version-flexible SnapMirror relationships for volumes that contain millions of files because performance can be affected.

Ways to set up version-flexible SnapMirror relationships

You can set up version-flexible SnapMirror relationships to optimize the number of Snapshot copies that you need for your application.

You can set up version-flexible SnapMirror relationships to retain the Snapshot copies that you want to retain by creating policies and rules that transfer and keep combinations of SnapMirror-created and user-created Snapshot copies. Because of the flexibility that version-flexible SnapMirror relationships gives you, you can set up a relationship to transfer and retain Snapshot copies according to your application. The following are some examples of useful applications that demonstrate this flexibility:

• You can create a relationship that transfers only SnapMirror created Snapshot copies during initialization and updates, and retains only the last two Snapshot copies.
  This behavior is useful if you want SnapMirror relationships that are similar to how qtree SnapMirror works in earlier versions of Data ONTAP. The policy for such an application has no rules or retention set, but is configured to replicate SnapMirror created Snapshot copies.

• You can create a relationship that transfers all Snapshot copies on the source volume, including SnapMirror created Snapshot copies.
  This behavior is useful because it retains the same number of Snapshot copies on the destination volume as on the source volume. During the initialization, all Snapshot copies are transferred. The Snapshot copies retained on the destination match the Snapshot copies on the source. For updates, transferred Snapshot copies that are deleted on the source are also deleted on the destination and new Snapshot copies created on the source are also transferred to the destination. The policy for such an application allows a specific value for rules and retention that would replicate and retain the same Snapshot copies as the source volume. This includes configuring the policy to replicate SnapMirror created Snapshot copies.

• You can create a relationship that transfers a specified set of Snapshot copies in addition to the SnapMirror created Snapshot copies.
  This behavior is useful for retaining application consistent Snapshot copies on the destination volume in addition to SnapMirror created Snapshot copies. Also, you have the flexibility to retain more Snapshot copies on the destination volume than on the source volume. The policy for such an application allows rules and retention that would replicate the application-consistent Snapshot copies and retain more Snapshot copies than on the source volume. This includes configuring the policy to replicate SnapMirror created Snapshot copies.
Converting a SnapMirror relationship to a version-flexible SnapMirror relationship

You can convert SnapMirror relationships that you created in a previous Data ONTAP release to version-flexible SnapMirror relationships because you can better control the Snapshot copies you replicate and retain for better resource utilization.

Steps

1. On the destination volume, break the SnapMirror relationship by using the `snapmirror break` command.

   **Example**
   ```
   cluster2::> snapmirror break -destination-path vserverB:dstvolB
   ```

2. On the destination volume, delete the SnapMirror relationship by using the `snapmirror delete` command.

   **Example**
   ```
   cluster2::> snapmirror delete -destination-path vserverB:dstvolB
   ```

3. Create a policy by using the `snapmirror policy create` command.

   Alternatively, you can use an existing policy that is a policy type of `async-mirror`.

   **Example**
   The following command creates a version-flexible SnapMirror policy called “vserverB-DR-policy” that will be used for SnapMirror style disaster recovery in which only the SnapMirror created Snapshot copy is transferred:
   ```
   cluster2::> snapmirror policy create -vserver vserverB -policy vserverB-DR-policy -policy-type mirror-vault -comment "DR policy"
   ```

4. Add a rule to the “vserverB-DR-policy” policy by using the `snapmirror policy add-rule` command.

   Adding a rule that matches the all_source_snapshots SnapMirror label will cause the version-flexible SnapMirror relationship to transfer all of the Snapshot copies to the destination volume, closely matching the behavior of the previous SnapMirror relationship.

   **Example**
   The following command adds a rule to the “vserverB-DR-policy” policy to retain all of the Snapshot copies on the source volume:
   ```
   cluster2::> snapmirror policy add-rule -vserver vserverB -policy vserverB-DR-policy -snapmirror-label all_source_snapshots
   ```

5. On the destination volume, create the version-flexible SnapMirror relationship by using the `snapmirror create` command with the `-type XDP` parameter and the `-policy` parameter.
The version-flexible SnapMirror relationship must use the same source volume and destination volume as the previous SnapMirror relationship.

**Example**

```
cluster2::> snapmirror create -source-path vserverA:srcvolA
-destination-path vserverB:dstvolB -type XDP -policy
vserverB-DR-policy -schedule daily
```

6. On the destination volume, resynchronize the source volume and destination volume of the version-flexible SnapMirror relationship by using the `snapmirror resync` command.

```
cluster2::> snapmirror resync -destination-path
vserverB:dstvolB
```

### Managing mirror relationships

You manage mirror relationships to optimize the performance of those relationships.

### Commands for managing SnapMirror relationships

Data ONTAP includes many commands for managing SnapMirror relationships of FlexVol volumes and Infinite Volumes.

You must have installed a SnapMirror license before you can manage SnapMirror relationships.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abort an active transfer</td>
<td><code>snapmirror abort</code></td>
</tr>
<tr>
<td></td>
<td>You can use the <code>snapmirror show</code> command to determine</td>
</tr>
<tr>
<td></td>
<td>the status of the abort operation.</td>
</tr>
<tr>
<td></td>
<td>If you are using Infinite Volumes, management tasks</td>
</tr>
<tr>
<td></td>
<td>must be performed on the Infinite Volume and not its</td>
</tr>
<tr>
<td></td>
<td>individual constituents.</td>
</tr>
<tr>
<td>Make a data protection</td>
<td><code>snapmirror break</code></td>
</tr>
<tr>
<td>mirror copy destination</td>
<td>This command must be used from the destination SVM.</td>
</tr>
<tr>
<td>writeable</td>
<td>You must not have I/O traffic running on the volume</td>
</tr>
<tr>
<td></td>
<td>when you use this command.</td>
</tr>
<tr>
<td>Create a new data</td>
<td><code>snapmirror create</code></td>
</tr>
<tr>
<td>protection mirror</td>
<td>This command must be used from the destination SVM.</td>
</tr>
<tr>
<td>relationship</td>
<td>If you are using Infinite Volumes, you can create data</td>
</tr>
<tr>
<td></td>
<td>protection mirror relationships between clusters only,</td>
</tr>
<tr>
<td></td>
<td>not within a cluster.</td>
</tr>
<tr>
<td>Delete a data protection</td>
<td><code>snapmirror delete</code></td>
</tr>
<tr>
<td>mirror relationship</td>
<td>This command must be used from the destination SVM.</td>
</tr>
<tr>
<td></td>
<td>If you are using Infinite Volumes, management tasks</td>
</tr>
<tr>
<td></td>
<td>must be performed on the Infinite Volume and not its</td>
</tr>
<tr>
<td></td>
<td>individual constituents.</td>
</tr>
<tr>
<td>If you want to...</td>
<td>Use this command...</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Start a baseline transfer</td>
<td><code>snapmirror initialize</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>destination SVM.</td>
</tr>
<tr>
<td>Display a list of data protection mirror relationships whose source endpoints are in the current SVM</td>
<td><code>snapmirror list-destinations</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>source SVM.</td>
</tr>
<tr>
<td>Modify a data protection mirror relationship</td>
<td><code>snapmirror modify</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>destination SVM.</td>
</tr>
<tr>
<td>Display a list of data protection and load-sharing mirror relationships or display the state of a scheduled transfer for a SnapMirror relationship</td>
<td><code>snapmirror show</code></td>
</tr>
<tr>
<td></td>
<td>The information that this command</td>
</tr>
<tr>
<td></td>
<td>shows is updated periodically;</td>
</tr>
<tr>
<td></td>
<td>therefore, any changes to a</td>
</tr>
<tr>
<td></td>
<td>relationship might not be displayed</td>
</tr>
<tr>
<td></td>
<td>immediately.</td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>destination SVM.</td>
</tr>
<tr>
<td>Display a list of completed SnapMirror operations</td>
<td><code>snapmirror show-history</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>destination SVM or cluster.</td>
</tr>
<tr>
<td></td>
<td>This task does not support</td>
</tr>
<tr>
<td></td>
<td>relationships with a “Pre 8.2”</td>
</tr>
<tr>
<td></td>
<td>relationship capability, for example,</td>
</tr>
<tr>
<td></td>
<td>load-sharing mirror relationships.</td>
</tr>
<tr>
<td>Disable future transfers for a mirror relationship</td>
<td><code>snapmirror quiesce</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>destination SVM.</td>
</tr>
<tr>
<td>Enable future transfers for a mirror relationship</td>
<td><code>snapmirror resume</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>destination SVM.</td>
</tr>
<tr>
<td>Start a resynchronize operation</td>
<td><code>snapmirror resync</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the</td>
</tr>
<tr>
<td></td>
<td>destination SVM.</td>
</tr>
<tr>
<td></td>
<td>You must not have I/O traffic running</td>
</tr>
<tr>
<td></td>
<td>on the volume when you use this command.</td>
</tr>
<tr>
<td></td>
<td>Quotas are turned off on the volume</td>
</tr>
<tr>
<td></td>
<td>you resynchronize. After resynchronizing, if you had quotas on the volume before the resynchronization you must reactivate quotas on the volume.</td>
</tr>
<tr>
<td></td>
<td><strong>Attention:</strong> A resynchronize operation can cause data loss on the destination volume because the command can remove the exported Snapshot copy on the destination volume.</td>
</tr>
<tr>
<td>If you want to...</td>
<td>Use this command...</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Add an owner to prevent premature deletion of a user-created Snapshot copy for a SnapMirror-to-SnapVault cascade configuration</td>
<td><code>snapmirror snapshot-owner create</code> A typical use case is to preserve an application-consistent Snapshot copy. This task is not supported for Infinite Volumes.</td>
</tr>
<tr>
<td>Delete an owner used to preserve a user-created Snapshot copy for a SnapMirror-to-SnapVault cascade configuration</td>
<td><code>snapmirror snapshot-owner delete</code> This task is not supported for Infinite Volumes.</td>
</tr>
<tr>
<td>Show all the Snapshot copies with owners that were added using the <code>snapmirror snapshot-owner create</code> command</td>
<td><code>snapmirror snapshot-owner show</code> This task is not supported for Infinite Volumes.</td>
</tr>
<tr>
<td>Start an incremental transfer</td>
<td><code>snapmirror update</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the destination cluster.</td>
</tr>
<tr>
<td></td>
<td>If you are using Infinite Volumes, aggregate requirements must be met before performing the incremental transfer. Management tasks must be performed on the Infinite Volume and not its individual constituents.</td>
</tr>
<tr>
<td></td>
<td>You can disregard error messages that result from updating a SnapMirror relationship from a Snapshot copy that exists on the destination volume. Any such messages are for support use.</td>
</tr>
<tr>
<td>Create a new policy for a data protection mirror relationship</td>
<td><code>snapmirror policy create</code></td>
</tr>
<tr>
<td>Delete a policy of a data protection mirror relationship</td>
<td><code>snapmirror policy delete</code></td>
</tr>
<tr>
<td>Add a new rule to a SnapVault relationship</td>
<td><code>snapmirror policy add-rule</code></td>
</tr>
<tr>
<td>Modify an existing rule in the policy of a SnapVault relationship</td>
<td><code>snapmirror policy modify-rule</code></td>
</tr>
<tr>
<td>Modify a policy of a data protection mirror relationship</td>
<td><code>snapmirror policy modify</code></td>
</tr>
<tr>
<td>Remove a rule from the policy of a data protection mirror relationship</td>
<td><code>snapmirror policy remove-rule</code></td>
</tr>
<tr>
<td>Show the policy of a data protection mirror relationship</td>
<td><code>snapmirror policy show</code></td>
</tr>
<tr>
<td>Copy data to a volume</td>
<td><code>snapmirror restore</code></td>
</tr>
<tr>
<td></td>
<td>Quotas are turned off on the volume you restore. After the restore, you must activate quotas on the volume before the restore.</td>
</tr>
<tr>
<td></td>
<td>This task is not supported for Infinite Volumes.</td>
</tr>
<tr>
<td>Remove the SnapMirror relationship information from the source SVM</td>
<td><code>snapmirror release</code></td>
</tr>
<tr>
<td></td>
<td>This command must be used from the source SVM.</td>
</tr>
</tbody>
</table>
Using extended queries to operate on many SnapMirror relationships

You can use extended queries to perform SnapMirror operations on many SnapMirror relationships at one time. For example, you might have many uninitialized SnapMirror relationships that you want to initialize using one command.

About this task

You can apply extended queries to the following SnapMirror operations:

- Initializing many Uninitialized SnapMirror relationships
- Resuming many Quiesced SnapMirror relationships
- Resynchronizing many Broken SnapMirror relationships
- Updating many Idle SnapMirror relationships
- Aborting many currently Transferring SnapMirror relationships

Step

1. You perform a SnapMirror operation on many SnapMirror relationships by using the following syntax: `snapmirror command {-state state } *`

Example

The following command initializes only SnapMirror relationships that are in an Uninitialized state:

```
vs1::> snapmirror initialize {-state Uninitialized} *
```

What tape seeding is

Tape seeding is an SMTape functionality that helps you initialize a destination FlexVol volume in a data protection mirror relationship.

Tape seeding enables you to establish a data protection mirror relationship between a source system and a destination system over a low-bandwidth connection.

Incremental mirroring of Snapshot copies from the source to the destination is feasible over a low bandwidth connection. However, an initial mirroring of the base Snapshot copy takes a long time over a low-bandwidth connection. In such cases, you can perform an SMTape backup of the source volume to a tape and use the tape to transfer the initial base Snapshot copy to the destination. You can then set up incremental SnapMirror updates to the destination system using the low-bandwidth connection.

Performing tape seeding using SMTape

Using SMTape, cluster administrators can perform tape seeding to initialize a destination FlexVol volume in a data protection mirror relationship. The time taken to initialize this destination volume
over a low bandwidth connection using SMTape is faster when compared to using the `snapmirror initialize` command.

**Before you begin**

- The tape must be connected to the same node on which the volume is located.
- All nodes in the cluster must be running Data ONTAP 8.2 or later.
- Source and destination volumes must be located on storage systems running clustered Data ONTAP.

**About this task**

If you back up 32-bit volumes, then you can restore these volumes only in the Data ONTAP 8.2 release family.

**Steps**

1. Determine which Snapshot copy you want to use for tape seeding by using the `volume snapshot show` command.

   **Example**

   The following example lists the Snapshot copies:

   ```
   clus1::> vol snapshot show -vserver vs1 -volume voll
   (volume snapshot show)
   
   Vserver  Volume  Snapshot               State       Size   Total% Used%
    -------- ------ ---------- -------------------------------- ----  ------
   vs1       voll     hourly.2013-01-25_0005   valid     224KB     0%    0%
   daily.2013-01-25_0010   valid     92KB      0%    0%
   hourly.2013-01-25_0105  valid     228KB     0%    0%
   hourly.2013-01-25_0205  valid     236KB     0%    0%
   hourly.2013-01-25_0305  valid     244KB     0%    0%
   hourly.2013-01-25_0405  valid     244KB     0%    0%
   hourly.2013-01-25_0505  valid     244KB     0%    0%
   7 entries were displayed.
   
   2. If you do not have an existing Snapshot copy, manually create a Snapshot copy of the source volume by using the `volume snapshot create` command.

   You must specify Storage Virtual Machine (SVM, formerly known as Vserver) name, source volume name, and Snapshot copy name.

   **Attention:** You must not delete this Snapshot copy until tape seeding is over.

   **Example**

   The following example shows how to create a Snapshot copy `mysnap` of the source volume `src1` on the SVM `vs1`. You can view the details of the Snapshot copy `mysnap` by using the `volume snapshot show` command:

   ```
   clus1::> volume snapshot create -vserver vs1 -volume src1 -snapshot mysnap
   clus1::> volume snapshot show -vserver vs1 -volume src1 -snapshot mysnap
   
   Vserver: vs1
   Volume: src1
   Snapshot: mysnap
   Creation Time: Thu Aug 09 12:03:46 2012
   Snapshot Busy: false
   ```
3. Move and position the tape correctly by using the `storage tape position` command.

**Example**

The following example moves and positions the no-rewind tape device, st01, and shows the status of the rewind operation:

```
clus1::> storage tape position -node clus1-01 -name nrst01 -operation rewind
```

Note: Rewind operation in progress. Use the "storage tape show -status" command to view the status of the operation.

```
clus1::> storage tape show -status -device-name-nr nrst01 -node clus1-01
```

<table>
<thead>
<tr>
<th>Node</th>
<th>Alias</th>
<th>Device Status</th>
<th>FileNo</th>
<th>BlockNo</th>
<th>Resid</th>
</tr>
</thead>
<tbody>
<tr>
<td>clus-01</td>
<td>st01</td>
<td>read-write-enabled</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

4. Use the `smtape backup` command to copy all the volume Snapshot copies, including the base Snapshot copy, to tape.

**Example**

The following example backs up the Snapshot copy mysnap to the tape device st0l:

```
clus1::> system smtape backup -vserver vs1 -volume src1 -backup-snapshot mysnap -tape /clus1-01/nrst01
```

Session 35 created successfully

5. Use the `smtape status show` command to view the progress of the baseline transfer.

**Example**

The following example shows the progress and status of the SMTape backup operation triggered in the previous step:

```
clus1::> system smtape status show -session 35 -instance
```

Session Identifier: 35
Node Name: clus1-01
Operation Type: backup
Session Status: ACTIVE
Path Name: /vs1/src1
Device Name: /clus1-01/nrst01
Bytes Transferred: 0B
6. Depending on the status of the SMTape backup operation, perform one of the following actions:

<table>
<thead>
<tr>
<th>If the Status shows...</th>
<th>Then...</th>
</tr>
</thead>
</table>
| COMPLETED              | Go to step 7.  
The baseline transfer is complete. |
| WAITING                | a. Load and position the new tape by using the `storage tape position` command.  
b. Continue the SMTape backup operation by using the `smtape continue` command. |
| FAILED                 | The SMTape backup operation may fail due to a number of reasons; such as loss of network connectivity, failure to access the named snapshot, unable to find the snapshot and so on. You must resolve the failure and restart the operation. |
| ACTIVE                 | The system displays the following message if the SMTape backup is in active state:  
   The SMTape backup is in progress. |

7. Physically transport the tapes to the destination node.

8. Optional: View the data backed up on a tape by using the `smtape showheader` command.

9. Create a destination volume of type DP and appropriate size (same or larger than the source volume size) on the destination cluster that will become the data protection mirror by using the `volume create` command and restrict the volume.

**Example**

The following example creates the data protection mirror volume dst1 on the Storage Virtual Machine (SVM, formerly known as Vsphere) vs1. The destination volume is located on the aggregate aggr5 and the destination volume dst1 is in the restricted state:

```
clus1::> volume create -vserver vs1 -volume dst1 -aggregate aggr5 -size 400m -type DP -state restricted
[Job 83] Job succeeded: Successful
```

10. Move and position the tape correctly by using the `storage tape position` command.

**Example**

The following example moves and positions the no-rewind tape device, st01, at the destination volume and shows the status of the rewind operation:
clus1::> storage tape position -node clus1-01 -name nrst01 -operation rewind
Note: Rewind operation in progress. Use the "storage tape show -status" command to view the status of the operation.

clus1::> storage tape show -status -device-name-nr nrst01 -node clus1-01
Device ID: fc215-21:5.126L1
Description: IBM LTO 4 ULTRIUM
Device Type: tape drive
    WWNN: 5:00a:098200:01dc69
    WWPN: 5:10a:098200:01dc69
Serial Number: bdf31432387ba0980a026c
    Errors: -

<table>
<thead>
<tr>
<th>Node</th>
<th>Alias</th>
<th>Device Status</th>
<th>FileNo</th>
<th>BlockNo</th>
<th>Resid</th>
</tr>
</thead>
<tbody>
<tr>
<td>clus-01</td>
<td>st01</td>
<td>read-write-enabled</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

11. Use the smtape restore command to restore all the volume Snapshot copies, including the base Snapshot copy, from tape to the destination volume.

Example

The following example restores all the data from tape to the destination volume dst1 on the Storage Virtual Machine (SVM, formerly known as Vserver) vs1:

clus1::> system smtape restore -vserver vs1 -volume dst1 -tape /clus1-01/nrst0l
    Session 36 created successfully

12. Use the smtape status show command to view the progress of the baseline transfer.

Example

The following example shows the progress and status of the SMTape restore operation triggered in the previous step:

clus1::> system smtape status show -session 36 -instance

13. Depending on the status of the SMTape restore operation, perform one of the following actions:
<table>
<thead>
<tr>
<th>If the Status shows...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPLETED</strong></td>
<td>Go to step 14.</td>
</tr>
<tr>
<td></td>
<td>The baseline transfer is complete.</td>
</tr>
<tr>
<td><strong>WAITING</strong></td>
<td>a. Load and position the new tape by using the <code>storage tape position</code> command.</td>
</tr>
<tr>
<td></td>
<td>b. Continue the SMTape restore operation by using the <code>smtape continue</code> command.</td>
</tr>
</tbody>
</table>

14. Use the `smtape break` command to break the volume and tape relationship.

**Note:** This command is also available for SVM administrators.

**Example**

The following example breaks the SMTape relationship between the tape and the volume dst1:

```
clus1::> system smtape break -vserver vs1 -volume dst1
Operation succeeded: snapmirror break for destination vs1:dst1
```

The destination volume is now writeable and a mirror relationship can be reestablished.

15. Establish the SnapMirror or SnapVault relationship by using the `snapmirror resync` command.

A SnapMirror relationship creates a data protection copy of the source volume; a SnapVault relationship creates a backup copy. The `-type` parameter determines the type of relationship established. The value of the `-type` parameter for a SnapMirror relationship is **DP** and the value for a SnapVault relationship is **XDP**.

**Example**

The following example reestablishes a SnapMirror relationship between the destination volume dst1 and the source volume src1:

```
clus1::> snapmirror resync -destination-path vs1:dst1 -source-path vs1:src1 -type DP
[Job 85] Job is queued: initiate snapmirror resync to destination "vs1:dst1".
[Job 85] Job succeeded: SnapMirror Resync Transfer Queued
```

16. Use the `snapmirror show` command to view the progress of the data protection mirror relationship reestablished between the destination volume and source volume in the previous step.

**Example**

The following example shows the data protection mirror relationship established between the source volume vs1 and destination volume dst1. The data protection mirror relationship type established is **DP**:

```
clus1::> snapmirror show -destination-path vs1:dst1

Source Path: vs1:src1
Destination Path: vs1:dst1
Relationship Type: DP
SnapMirror Schedule: -
```
Providing disaster recovery using mirroring technology

When the relationship status shows **idle**, the data protection mirror relationship is established and tape seeding is complete.

**Example**

```bash
clus1::> snapmirror show -destination-path vs1:dst1
```

Source Path: vs1:src1
Destination Path: vs1:dst1
Relationship Type: DP
SnapMirror Schedule: -
Tries Limit: -
Throttle (KB/sec): unlimited
Mirror State: Snapmirrored
Relationship Status: Idle
Transfer Snapshot: -
Snapshot Progress: -
Total Progress: -
Snapshot Checkpoint: -
Newest Snapshot: snapmirror.58621f01-e214-11e1-833d-123478563412_2147484708.2012-08-09_120444
Newest Snapshot Timestamp: 08/09 12:04:44
Exported Snapshot: snapmirror.58621f01-e214-11e1-833d-123478563412_2147484708.2012-08-09_120444
Exported Snapshot Timestamp: 08/09 12:04:44
Healthy: true
Constituent Relationship: false
Relationship ID: 6485d262-e21a-11e1-833d-123478563412
Transfer Type: resync
Transfer Error: -
Current Throttle: 103079214
Current Transfer Priority: normal
Last Transfer Type: -
Last Transfer Error: -
Last Transfer Size: -
Last Transfer Duration: -
Last Transfer From: -
Progress Last Updated: 08/09 12:04:45
Relationship Capability: 8.2 and above
Lag Time: -
Policy: DPDefault
Related information

Documentation on the NetApp Support Site: mysupport.netapp.com

Scalability limits for SMTape backup and restore sessions

While performing SMTape backup and restore operations through NDMP or CLI (tape seeding), you must be aware of the maximum number of SMTape backup and restore sessions that can be performed simultaneously on storage systems with different system memory capacities. This maximum number depends on the system memory of a storage system.

**Note:** SMTape backup and restore sessions scalability limits are different from NDMP session limits and dump session limits.

<table>
<thead>
<tr>
<th>System memory of the storage system</th>
<th>Total number of SMTape backup and restore sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 16 GB</td>
<td>6</td>
</tr>
<tr>
<td>Greater than or equal to 16 GB but less than 24 GB</td>
<td>16</td>
</tr>
<tr>
<td>Greater than or equal to 24 GB</td>
<td>32</td>
</tr>
</tbody>
</table>

You can obtain the system memory of your storage system by using the `sysconfig -a` command (available through the nodeshell). For more information about using this command, see the man pages.

**Listing the schedule state of a mirror relationship**

You might want to see what state a scheduled transfer for a mirror relationship is in to ensure that jobs are running as they should.

**About this task**

The state of a scheduled job might be dormant. The dormant state means that the job is waiting for the scheduled start time to begin the transfer. There is nothing wrong with the job and you do not need to do anything.

**Step**

1. To see the state of scheduled jobs, use the `snapmirror show` command.

**Scheduling SnapMirror transfers**

If you want scheduled SnapMirror transfers, you can add a schedule to a mirror relationship after you initially create the relationship.

**About this task**

Unless you create and implement a schedule for SnapMirror transfers, you are limited to manually updating destination FlexVol volumes or Infinite Volumes with mirror relationships. The following are characteristics of adding a SnapMirror transfer schedule:

- When you add a schedule for a data protection mirror copy of Infinite Volumes, do not schedule updates for less than one-hour intervals.
  - If you schedule updates for less than one-hour intervals, Data ONTAP tries but cannot meet the schedule for Infinite Volumes, and the data protection mirror relationship is displayed as unhealthy.
• Scheduled SnapMirror transfers (or even manual updates) can disrupt Snapshot copy schedules when a transfer lasts longer than the full retention period of the schedule.

Steps
1. Create the schedule you want to implement by using the `job schedule cron create` command.

   **Note:** You cannot use the `job schedule interval create` command to schedule SnapMirror transfers.

2. Apply the schedule to the mirror relationship by using the `--schedule` option of the `snapmirror modify` command.

   See the `snapmirror modify` command man page for more information about the command.

Changing mirror relationship schedules

You can change a schedule that updates mirror relationships for FlexVol volumes and Infinite Volumes if the schedule impacts other backups or updates.

Before you begin

• You must have created the cluster and Storage Virtual Machine (SVM) peering relationship.

   To know about creating cluster and SVM peering, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.

About this task

Changing a schedule affects load-sharing mirror copies differently than it does for data-protection mirror copies. If you change a schedule to a load-sharing mirror relationship, Data ONTAP makes the change to the relationships of all the load-sharing mirror copies in the group. Data ONTAP determines the load-sharing mirror group by the SVM and source volume specified by the command. See the *Clustered Data ONTAP Logical Storage Management Guide* for more information about load-sharing mirror copies.

Steps
1. Create the new schedule by using the `job schedule cron create` command.

   Creating a schedule is described in the cron job creation section of the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*. See the man page for details about the `job schedule cron create` command.

2. Change the schedule for a mirror relationship by using the `snapmirror modify --schedule` command.

   This command must be used from the destination SVM.

Example

The following command changes the update schedule used by a data protection mirror relationship for destination volume named “dept_eng_ls1” to a schedule named “dept_eng_mirror_sched”:

```
vs2::> snapmirror modify --source-path vs1:dept_eng
   --destination-path vs2:dept_eng_ls1 --schedule dept_eng_mirror_sched
```
Manually updating data protection mirror copies on destination volumes

You can schedule updates to data protection mirror copies for destination FlexVol volumes or Infinite Volumes, or you can manually update a data protection mirror copy to transfer Snapshot copies between the source and destination volume. However, for Infinite Volumes, you cannot choose which Snapshot copies to transfer.

Before you begin

- A base Snapshot copy must exist on the source volume and the destination volume.
- The destination volume must be the same size or bigger than the source volume.

About this task

You can have cluster administrator or Storage Virtual Machine (SVM) administrator privileges to perform this task.

When you update a destination volume, all of the Snapshot copies from the source volume are transferred to the destination volume. In addition, any Snapshot copies deleted from the source volume are deleted from the destination volume during the update. Similarly, any new Snapshot copies on the source volume are transferred to the destination volume.

Step

1. On the destination cluster, manually update a destination volume by using the `snapmirror update` command.

Example

The following command updates the data protection mirror relationship for a destination volume named repo_vol_dest on an SVM named vs0_dest:

```
vs2::> snapmirror update -destination-path vs0_dest:repo_vol_dest
```

Deleting a mirror copy

You can delete a mirror relationship and the destination FlexVol volume or Infinite Volume if you no longer want the mirror copy.

About this task

When you delete a mirror copy, you must delete the mirror relationship and the destination volume. Deleting the mirror relationship does not delete SnapMirror created Snapshot copies on either the source or destination volumes. Deleting the mirror relationship attempts to delete Snapshot copy owners for the SnapMirror created Snapshot copies on both source and destination volumes.

When you delete a load-sharing mirror copy from a set of load-sharing mirror copies, the destination volume of the deleted load-sharing mirror relationship cannot be used again as a destination volume of a load-sharing relationship if it contains any data or Snapshot copies.

Steps

1. Optional: On the source Storage Virtual Machine (SVM), use the `snapmirror list-destination` command to view the list of destination volumes for that source volume.
Example

```bash
vs1::> snapmirror list-destinations
```

<table>
<thead>
<tr>
<th>Source Path</th>
<th>Type</th>
<th>Destination Path</th>
<th>Status</th>
<th>Transfer Progress</th>
<th>Last Updated</th>
<th>Relationship Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1:src_ui</td>
<td>DP</td>
<td>vs2:vsrsrc_ui_ls_mir2</td>
<td>Idle</td>
<td>-</td>
<td>-</td>
<td>3672728c-</td>
</tr>
<tr>
<td>ad06-11e2-981e-123478563412</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Use the `snapmirror delete` command to delete a mirror relationship.

Example

The following command deletes a mirror relationship between a destination volume named `src_ui_ls_mir2` and a source volume named `src_ui`:

```bash
vs2::> snapmirror delete -source-path vs1:src_ui -destination-path vs2:src_ui_ls_mir2
```

The command deletes the mirror relationship, but does not delete the destination volume. In the case of a load-sharing mirror copy, the destination volume will be in the restricted state. If you want to use it as a destination volume of a data protection relationship, you must wait at least 10 minutes. This is the amount of time required to refresh internal caches and place the volume back online.

3. Use the `snapmirror release` command from the source SVM to remove the configuration information and Data ONTAP created Snapshot copies on the source volume.

Example

The following command removes the DP or XDP relationship from the source SVM named vs1:

```bash
vs1::> snapmirror release -source-path vs1:src_ui -destination-path vs2:src_ui_ls_mir2
```

This command removes the DP or XDP relationship information from the source SVM and does not delete any volumes. This command deletes the base Snapshot copies for the destination volume named `src_ui_ls_mir2` from the source volume named `src_ui`.

4. Optional: Use the `volume delete` command to delete the destination volume.

Delete the destination volume if you no longer need the volume.

Considerations when breaking SnapMirror relationships

You might break a SnapMirror relationship when a disaster impacts the source volume so that you can temporarily serve data from the destination volume. There are some considerations you should understand before breaking a SnapMirror relationship so that you can avoid issues.

- When you break a SnapMirror relationship, the common Snapshot copy between the source and destination volumes is not protected on the source volume and can be deleted.
  This is evident when the SnapMirror relationship is part of a cascade and is the expected behavior. If, for example, you have a cascade from volume A to volume B and volume B to volume C, and you break the relationship between volume A and volume B. The relationship between volume B and volume C still exists. When a replication update from volume B to volume C occurs, volume B will lose the common Snapshot copy it has with volume A.
  You can avoid this issue by creating your own Snapshot copy on the source that will not get deleted automatically, and then replicating it to the destination volume before breaking the relationship.
The destination volume of a SnapMirror relationship that has the NVFAIL parameter disabled will have the NVFAIL parameter enabled after you break the relationship.

**Reversing the data protection mirror relationship when disaster occurs**

When disaster disables the source FlexVol volume of a data protection mirror relationship, you can use the destination FlexVol volume to serve data while you repair or replace the source, update the source, and reestablish the original configuration of the systems.

**About this task**

The following procedure describes a data protection mirror relationship that has the source volume on one Storage Virtual Machine (SVM) and the destination volume on another SVM. The source and the destination clusters and source and destination SVMs are in peer relationships. The original source (the one disabled by the disaster) is vs1:volA and the original destination is vs2:volB. All data from the last scheduled SnapMirror Snapshot copy before the source was disabled and all the data written to vs2:volB after it was made writeable is preserved. Any data written to vs1:volA between the last SnapMirror Snapshot copy and the time that vs1:volA was disabled is not preserved.

For information about retrieving data from Infinite Volumes during disaster recovery, see the *Clustered Data ONTAP Infinite Volumes* Management Guide.

**Steps**

1. Temporarily make the original source volume a read-only destination volume and reverse the data protection mirror relationship to continue to serve data.

   If the source vs1:volA is recoverable and its data is intact, complete the following steps:

   a. After the source volume (in this case, vs1:volA) is disabled, use the `snapmirror break` command on the destination volume, vs2:volB, to make the destination volume, vs2:volB, writeable.

   ```
   Example
   vs2::> snapmirror break vs2:volB
   ```

   b. Redirect the clients of the source volume vs1:volA to the new source volume vs2:volB. The former clients of vs1:volA access and write to vs2:volB.

   c. On the new destination volume, vs1:volA, use the `snapmirror create` command to create the mirror relationship, but with vs2:volB as the new source and vs1:volA as the new destination.

   ```
   Example
   vs1::> snapmirror create vs2:volB vs1:volA -type DP
   ```

   d. If there are LUNs on the original source volume, vs1:volA, remove the mapping by using the `lun unmap` command.

   e. On the new destination volume, vs1:volA, use the `snapmirror resync` command to resynchronize vs1:volA with vs2:volB.
Example

```
vs1::> snapmirror resync vs1:volA
```

f. On the original destination volume, `vs2:volB`, use the `snapmirror delete` command to remove the data protection mirror relationship between the source `vs1:volA` and the destination `vs2:volB`.

Example

```
vs2::> snapmirror delete vs2:volB
```

g. On the original source volume, `vs1:volA`, use the `snapmirror release` command to remove relationship information from the source.

Example

```
vs1::> snapmirror release vs2:volB
```

h. If there were LUNs, map the LUNs on the new source `vs2:volB` by using the `lun map` command.

If the source `vs1:volA` is unrecoverable, complete the following steps:

a. After the source volume (in this case, `vs1:volA`) is disabled, use the `snapmirror break` command on the destination volume, `vs2:volB`, to make the destination volume, `vs2:volB`, writeable.

Example

```
vs2::> snapmirror break vs2:volB
```

b. Redirect the clients of the source volume `vs1:volA` to the new source volume `vs2:volB`.

The former clients of `vs1:volA` access and write to `vs2:volB`.

c. On the destination volume, `vs2:volB`, use the `snapmirror delete` command to remove the data protection mirror relationship between the source `vs1:volA` and the destination `vs2:volB`.

Example

```
vs2::> snapmirror delete vs2:volB
```

d. On the source SVM, `vs1`, use the `snapmirror release` command to remove relationship information from the source.

Even though the source volume is unrecoverable, the data protection mirror relationship still exists and must be removed.

Example

```
vs1::> snapmirror release vs2:volB
```

e. Delete the old volume `volA` and use the `volume create` command to create a new data protection destination volume called `vs1:volA`. 
Note: Remember to use the -type DP parameter when creating the destination volume.

Example

vs1::> volume delete -volume vs1:volA

vs1::> volume create -volume volA -aggr aggr1 -type DP -vserver vs1

f. On the new destination volume, vs1:volA, use the snapmirror create command to create the data protection mirror relationship with vs2:volB as the new source volume and vs1:volA as the new destination volume.

Example

vs1::> snapmirror create vs2:volB vs1:volA -type DP

g. On the new destination volume, vs1:volA, use the snapmirror initialize command to create the baseline on the data protection mirror copy. This command also makes vs1:volA a read-only destination.

Example

vs1::> snapmirror initialize vs1:volA

h. If there were LUNs, map the LUNs on the new source vs2:volB by using the lun map command.

You can keep this configuration or, after resolving the problem with the original source volume, you can complete the following steps to reestablish the original data protection mirror relationship.

2. On the new destination volume, vs1:volA, update the new destination volume vs1:volA to transfer the latest data from the new source volume vs2:volB by using the snapmirror update command.

Example

vs1::> snapmirror update vs1:volA

3. On the new destination volume, vs1:volA, use the snapmirror break command to make vs1:volA writeable.

Example

vs1::> snapmirror break -source-path vs2:volB -destination-path vs1:volA

4. On the original destination volume, vs2:volB, use the snapmirror create command to re-create the original data protection mirror relationship with vs1:volA as the source and vs2:volB as the destination.
Example

```
vs2::> snapmirror create vs1:volA vs2:volB -type DP
```

5. If there are LUNs on the source `vs2:volB`, remove the mapping by using the `lun unmap` command.

6. On the original destination volume, `vs2:volB`, use the `snapmirror resync` command to resynchronize the original source and original destination volumes.

Example

```
vs2::> snapmirror resync vs2:volB
```

7. On the new destination volume, `vs1:volA`, use the `snapmirror delete` command to remove the data protection mirror relationship between the new source `vs2:volB` and the new destination `vs1:volA`.

Example

```
vs1::> snapmirror delete vs1:volA
```

8. On the new source volume, `vs2:volB`, use the `snapmirror release` command to remove the data protection mirror relationship between the new source `vs2:volB` and the new destination `vs1:volA`.

Example

```
vs2::> snapmirror release vs1:volA
```

9. Redirect the clients from `vs2:volB` back to their original source volume `vs1:volA`.

10. If there were LUNs, map them back to the original source `vs1:volA`.

Reversing the version-flexible SnapMirror relationship when disaster occurs

When disaster disables the source FlexVol volume of a version-flexible SnapMirror relationship, you can use the destination FlexVol volume to serve data while you repair or replace the source, update the source, and reestablish the original configuration of the systems.

About this task

The following procedure describes the following version flexible SnapMirror relationship:

- The original source volume and Storage Virtual Machine (SVM) is `vs1:volA`. This is the volume disabled by the disaster.
- The original destination volume and SVM is `vs2:volB`. This is the volume that will serve data while you address the issues with the source.
- The source and destination clusters and source and destination SVMs are in peer relationships. Peer relationships are required for you to have any kind of SnapMirror replication between SVMs or clusters.

All data from the last scheduled version-flexible SnapMirror Snapshot copy before the source was hit by the disaster and all the data written to the destination, `vs2:volB`, after it is made writeable during
the procedure, is preserved. Any data written to vs1:volA between the last SnapMirror Snapshot copy and the time that vs1:volA was stricken is not preserved.

**Steps**

1. Temporarily create a new SnapMirror relationship that uses the destination volume from the disaster-stricken SnapMirror relationship as the new source volume.

   This allows you the time to either reuse the stricken source volume, if it is recoverable, or create a new volume that can become the source volume, if it is not recoverable.

   If the source vs1:volA is recoverable and its data is intact, complete the following steps:

   a. Make the vs2:volB destination volume writable by using the `snapmirror break` command.

   **Example**

   ```bash
   vs2::> snapmirror break vs2:volB
   ```

   b. Redirect the clients from the original vs1:volA source volume to the vs2:volB volume.

   This gives the former clients of the vs1:volA volume access and write capability to the vs2:volB volume. The vs2:volB volume will become the temporary source volume of a new SnapMirror relationship while you repair the original vs1:volA source volume.

   c. On the vs1:volA volume, use the `snapmirror create` command to create the mirror relationship with vs2:volB as the source and vs1:volA as the destination.

   This SnapMirror relationship is the reverse of the original SnapMirror relationship and will replicate new data to the vs1:volA volume.

   **Example**

   This example uses the “MirrorLatest” policy, which retains only the latest SnapMirror created Snapshot copy.

   ```bash
   vs1::> snapmirror create -source-path vs2:volB
       -destination-path vs1:volA -type XDP -policy MirrorLatest
   ```

   d. If there are LUNs on the original source volume, vs1:volA, remove the mapping by using the `lun unmap` command.

   e. On the vs1:volA destination volume, use the `snapmirror resync` command to resynchronize vs1:volA with vs2:volB.

   **Example**

   ```bash
   vs1::> snapmirror resync vs1:volA
   ```

   f. From the SVM that contains the vs2:volB volume, remove the original SnapMirror relationship between volumes by using the `snapmirror delete` command.

   **Example**

   ```bash
   vs2::> snapmirror delete vs2:volB
   ```

   g. From the SVM that contains the vs1:volA volume, remove the original SnapMirror relationship information and Snapshot copies by using the `snapmirror release` command.
Example

vs1::> snapmirror release vs2:volB

If there were LUNs, map the LUNs on the new source vs2:volB by using the `lun map` command.

If the source vs1:volA is unrecoverable, complete the following steps:

a. After the source volume (in this case, vs1:volA) is disabled, use the `snapmirror break` command on the destination volume, vs2:volB, to make the destination volume, vs2:volB, writeable.

Example

vs2::> snapmirror break vs2:volB

b. Redirect the clients of the source volume vs1:volA to the new source volume vs2:volB. The former clients of vs1:volA access and write to vs2:volB.

c. Delete the old volA volume by using the `volume delete` command.

Example

vs1::> volume delete -volume vs1:volA

d. Create a new SnapMirror destination volume called vs1:volA by using the `volume create` command with the `type DP` parameter.

   **Note:** Remember to use the `type DP` parameter when creating the destination volume.

Example

vs1::> volume create -volume volA -aggr aggr1 -type DP -vserver vs1

e. On the new destination volume, vs1:volA, use the `snapmirror create` command to create the data protection mirror relationship with vs2:volB as the new source volume and vs1:volA as the new destination volume.

Example

vs1::> snapmirror create -source-path vs2:volB -destination-path vs1:volA -type XDP -policy MirrorLatest

f. On the new destination volume, vs1:volA, use the `snapmirror initialize` command to create the baseline on the data protection mirror copy. This command also makes vs1:volA a read-only destination.

Example

vs1::> snapmirror initialize vs1:volA
g. On the destination volume, vs2:volB, use the `snapmirror delete` command to remove the data protection mirror relationship between the source vs1:volA and the destination vs2:volB.

   **Example**
   ```
   vs2::> snapmirror delete vs2:volB
   ```

h. Remove relationship information from the vs1 SVM by using the `snapmirror release` command.

   Even though the source volume is unrecoverable, the data protection mirror relationship still exists and must be removed.

   **Example**
   ```
   vs1::> snapmirror release vs2:volB
   ```

i. If there were LUNs, map the LUNs on the new source vs2:volB by using the `lun map` command.

   You can keep this configuration or, after resolving the problem with the original source volume, you can complete the following steps to reestablish the original data protection mirror relationship.

2. On the new destination volume, vs1:volA, update the new destination volume vs1:volA to transfer the latest data from the new source volume vs2:volB by using the `snapmirror update` command.

   **Example**
   ```
   vs1::> snapmirror update vs1:volA
   ```

3. On the new destination volume, vs1:volA, use the `snapmirror break` command to make vs1:volA writeable.

   **Example**
   ```
   vs1::> snapmirror break -source-path vs2:volB -destination-path vs1:volA
   ```

4. On the original destination volume, vs2:volB, use the `snapmirror create` command to re-create the original data protection mirror relationship with vs1:volA as the source and vs2:volB as the destination.

   **Example**
   ```
   vs2::> snapmirror create -source-path vs1:volA -destination-path vs2:volB -type XDP -policy MirrorLatest
   ```

5. If there are LUNs on the source vs2:volB, remove the mapping by using the `lun unmap` command.

6. On the original destination volume, vs2:volB, use the `snapmirror resync` command to resynchronize the original source and original destination volumes.
Example

\texttt{vs2::> snapmirror resync vs2:volB}

7. On the new destination volume, \texttt{vs1:volA}, use the \texttt{snapmirror delete} command to remove the data protection mirror relationship between the new source \texttt{vs2:volB} and the new destination \texttt{vs1:volA}.

Example

\texttt{vs1::> snapmirror delete vs1:volA}

8. On the new source volume, \texttt{vs2:volB}, use the \texttt{snapmirror release} command to remove the data protection mirror relationship between the new source \texttt{vs2:volB} and the new destination \texttt{vs1:volA}.

Example

\texttt{vs2::> snapmirror release vs1:volA}

9. Redirect the clients from \texttt{vs2:volB} back to their original source volume \texttt{vs1:volA}.

10. If there were LUNs, map them back to the original source \texttt{vs1:volA}.

Related information

\textit{Clustered Data ONTAP 8.3 Infinite Volumes Management Guide}

Converting a data protection mirror destination to a writeable volume

You can convert the destination volume of a data protection mirror relationship to a writable volume if you want to use that volume to serve data. For example, you might do this if you want to migrate a volume.

Steps

1. On the destination Storage Virtual Machine (SVM), make the destination volume writable by using the \texttt{snapmirror break} command.

2. Remove the data protection mirror relationship that the destination volume has with the source volume by using the \texttt{snapmirror delete} command.

3. On the source SVM, remove the configuration information and Data ONTAP created Snapshot copies by using the \texttt{snapmirror release} command.

Testing database applications

You can use data protection mirror relationships to create a copy of database data on which to test software applications that run on a database, to avoid the possibility of changing or corrupting the database.

Before you begin

The volume that contains the database must be in a data protection mirror relationship.
Steps

1. On the destination Storage Virtual Machine (SVM), make the destination volume writeable by using the `snapmirror break` command.

   **Example**

   The following example breaks a SnapMirror relationship that has the `vs2:Test_vol` volume as its destination volume.

   ```bash
   vs2::>> snapmirror break -destination-path vs2:Test_vol
   ```

2. Run the application on the data in the former destination volume (`vs2:Test_vol`).

3. Check the data in the former destination volume (`vs2:Test_vol`).

4. If testing results in alterations to the data that you do not want, use the `snapmirror resync` command to reestablish the mirror relationship.

5. Repeat Steps 2, 3, and 4, until you are satisfied with the testing.

6. Optional: After completing the test, you can restore the SnapMirror relationship by resynchronizing the source and the destination volumes.
Providing disaster recovery for SVMs

Data stored in a Storage Virtual Machine (SVM) is susceptible to disaster, either through hardware failure or environmental catastrophe on the cluster. You can protect the SVM data volumes and configuration by creating a mirror copy of the data volumes and configuration details on a destination SVM on the destination cluster.

You can create a SnapMirror relationship (also referred as SVM disaster recovery relationship) from the source SVM to the destination SVM to provide asynchronous disaster recovery. When you create a SnapMirror relationship, you can choose to replicate all or a subset of configuration (excluding network and protocol configuration) along with the data volumes, depending on the cluster setup.

When the source SVM is unavailable, the cluster administrator of the destination cluster can start the destination SVM to provide data access, with minimum disruption.

When the source SVM is available, the cluster administrator of the source cluster can resynchronize the data and configuration from the destination SVM to serve data from the source SVM.

As the cluster administrator, you must configure the source and destination SVMs for disaster recovery and understand the tasks to manage the SVM disaster recovery relationships.

You cannot set up an SVM disaster recovery relationship for SVMs with Infinite Volumes.

You can use `snapmirror` commands to configure SVM disaster recovery and manage the SVM disaster recovery relationships. After the configuration is complete, the SnapMirror operations are executed only at the SVM level.

What a destination SVM is

A destination Storage Virtual Machine (SVM) is an SVM with the `dp-destination` subtype created for protecting a source SVM that serves data. A destination SVM contains replicated data and configuration of the source SVM.

When you create a destination SVM, it does not contain a root volume and is in the stopped state. The destination SVM remains in the stopped state until you activate it for serving data in the event of a disaster. The destination SVM can be started to provide read-only access to the clients if the option `identity-preserve` option is set to `false`.

When the destination SVM subtype is `dp-destination`, you can perform SnapMirror operations as well as delete the destination SVM when required. You cannot modify any volumes or configurations that are replicated on the destination SVM. You cannot create new volumes or FlexClone volumes on the destination SVM.

You can resize the volumes on the destination SVM when you want to over-provision them. The volumes on the destination SVM are in DP state and therefore provide read-only access to the NFS clients.

When you activate the destination SVM, it becomes writable and the subtype changes from `dp-destination` to `default`. As a result, all the volumes have read/write access.

The destination SVM can contain LS mirror volumes that are created for the root volume only.
Deciding whether to replicate SVM network configuration

Depending on the data and network configuration of the source SVM, you can choose to replicate all or a subset of SVM configuration (excluding the network and protocol configuration), when setting up the destination SVM for disaster recovery.

**Choices**

- Replicate data and all of the SVM configuration in the following scenarios:
  - The destination SVM must have the same NAS configuration as the source SVM.
  - The source SVM is not configured for SAN data access.
  - The destination SVM is not required to provide read-only access.

If the source and destination clusters are in the same network subnet, the destination SVM will use the same IP addresses as the source SVM so that clients can access data by using the same IP addresses and network configuration when the source SVM is not available.

If the source and destination clusters are in different network subnets, on the source SVM, you must create destination LIFs belonging to the subnet of the destination SVM.

LIFs that belong to the subnet of the destination SVM will not be reachable until the destination SVM is started. After the destination SVM is started, the LIFs that are replicated and that belong to the subnet of the source SVM will not be reachable.

- Replicate data and a subset of the SVM configuration in the following scenarios:
  - The destination SVM must have a different NAS configuration from the source SVM.
  - The source SVM is configured only for SAN data access.
  - The destination SVM is required to provide read-only access.

  **Note:** Network, protocol, and name services configurations are not replicated.

You can set up the destination SVM to provide read-only access for NFS clients and SAN hosts by configuring the network and protocols on the destination SVM.

Configurations replicated in an SVM disaster recovery relationship

When you set up the SVM disaster recovery relationship, the value you select for the `--identity-preserve` option of the `snapmirror create` command determines the configurations that are replicated in the destination SVM.

If you set the `--identity-preserve` option to `true`, all the configuration details except the SAN configuration are replicated. If you set the option to `false`, only a subset of the configuration details that are not associated with the network configuration are replicated.

The following table lists the configuration details that are replicated when the `--identity-preserve` option is set to `true` or `false`:
<table>
<thead>
<tr>
<th>Configuration</th>
<th>Replicated if the <code>-identity-preserve</code> is set to true</th>
<th>Replicated if the <code>-identity-preserve</code> is set to false</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFS</td>
<td>CIFS server</td>
<td>Yes</td>
</tr>
<tr>
<td>CIFS policy</td>
<td>Local groups and local user</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Privilege</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Shadow copy</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>BranchCache</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Server options</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Server security</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Home directory, share</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Symlink</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Fpolicy policy, Fsecurity policy, and Fsecurity NTFS</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Name mapping and group mapping</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Audit information</td>
<td>Yes</td>
</tr>
<tr>
<td>NFS</td>
<td>Export policies</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Export policy rules</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>NFS server and Kerberos configuration</td>
<td>Yes</td>
</tr>
<tr>
<td>Network</td>
<td>NAS LIFs</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>SAN LIFs</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Firewall policies</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Routes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Broadcast domain</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Subnet</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>IPspace</td>
<td>No</td>
</tr>
<tr>
<td>RBAC</td>
<td>Security certificates</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Login user, public key, role, and role configuration</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>SSL</td>
<td>Yes</td>
</tr>
<tr>
<td>Configuration</td>
<td>Replicated if the <code>-identity-preserve</code> is set to true</td>
<td>Replicated if the <code>-identity-preserve</code> is set to false</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Name services</td>
<td>DNS and DNS hosts</td>
<td>Yes</td>
</tr>
<tr>
<td>UNIX user and UNIX group</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Kerberos realm and Kerberos keyblocks</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LDAP and LDAP client</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Netgroup</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NIS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Web and web access</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Volume</td>
<td>Object</td>
<td>Yes</td>
</tr>
<tr>
<td>Snapshot copies, Snapshot policy, and autodelete policy</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Efficiency policy</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Quota policy and quota policy rule</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Recovery queue</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Root volume</td>
<td>Namespace</td>
<td>Yes</td>
</tr>
<tr>
<td>User data</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Qtrees</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Quotas</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>File-level QoS</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Attributes: state of the root volume, space guarantee, size, export policies, autosize, and total number of files</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Storage QoS</td>
<td>QoS policy group</td>
<td>Yes</td>
</tr>
<tr>
<td>Fibre Channel (FC)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>iSCSI</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LUNs</td>
<td>Object</td>
<td>Yes</td>
</tr>
<tr>
<td>igroups</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>portsets</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SNMP</td>
<td>v3 users</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note**: Cluster-level objects such as aggregates are not replicated.
How volumes and LIFs are placed on the destination SVM

You must understand how aggregates are selected to place the volumes and ports to place the LIFs on the destination SVM in case you want to move the volumes or migrate the LIFs on the destination SVM.

Criteria for placing volumes on the destination SVM

When replicating volumes from the source SVM to the destination SVM, aggregates are selected based on the following criteria:

1. Volumes are always placed on non-root aggregates.

2. If a volume on the source SVM is located on a Flash Pool aggregate, then the volume is placed on a Flash Pool aggregate on the destination SVM, if such an aggregate exists and has enough free space.

3. Non-root aggregates are selected based on the available free space and the number of volumes already hosted on the aggregate.
   Aggregates with more free space and less number of volumes are given priority. The aggregate with the highest priority is selected.

4. If the -space-guarantee option of the volume that is replicated is set to volume, only aggregates with free space greater than the volume size are considered.

5. The volume size grows automatically on the destination SVM during replication, based on the source volume size.
   If you want to pre-reserve the size on the destination SVM, you must resize the volume. The volume size does not shrink automatically on the destination SVM based on the source SVM.

If you want to move a volume from one aggregate to another, you can use the volume move command on the destination SVM.

Criteria for placing LIFs on the destination SVM

When replicating network configuration LIFs from the source SVM to the destination SVM, ports are selected based on the following criteria:

1. Ports belonging to the same IPspace as that of the destination SVM are tested for connectivity to the subnet of the LIF that is replicated.

2. The connectivity of such ports is verified by sending ARP probes (RFC5227).
   Ports that receive a response are considered as candidates for placing LIFs.
   If no port receives any response and if a subnet object exists that is configured same as the subnet of the LIF to be placed, then all ports belonging to that subnet are considered as candidates for the port list.
   If no port receives any response and there is no subnet object for all the LIF subnets, then all ports belonging to the IPspace are considered as candidates for the port list.

3. From the candidate list of ports, a single port is selected based on the following criteria:
   - If the LIF is placed on a VLAN port on the source SVM and a port with the same VLAN tag in the candidate port list exists, then that port is chosen for placing the LIF.
   - If the LIF was placed on an interface group and an interface group with same name exists in the candidate port list, then that interface group is selected for placing the LIF.
   - If the LIF was placed on a physical port on the source SVM, then a physical port with the same speed as that of the source SVM is preferred from the candidate port list.
• If same-speed ports are not available in the candidate port list, then the LIFs are distributed across candidate ports based on a weighted random algorithm. Ports with higher speed have more preference, and therefore LIFs are placed on the higher speed ports.

If you want to move the LIFs from one port to another, you can use the `network interface migrate` or `network interface modify -home-node -home-port` command.

Configuring disaster recovery for SVMs

The SVM disaster recovery workflow includes creating and preparing the destination SVM, activating the destination SVM in the event of a disaster, reactivating and reconfiguring the source SVM for disaster recovery.

**Before you begin**

The source and destination clusters must be peered.

*Creating the cluster peer relationship* on page 71

**About this task**

This task provides a high-level overview of the tasks involved in setting up disaster recovery for an SVM.

Express Guides are available that provide detailed information about configuring SVM disaster recovery.

**Steps**

1. Create and prepare the destination SVM to protect the data and configuration of an SVM.

   *Data ONTAP 8.3 SVM Disaster Recovery Preparation Express Guide*

   The following illustration shows the source and destination SVMs in the preparation phase:

   After the preparation is complete, the destination SVM is in the stopped state.

   The data flows from the source SVM to the destination SVM, and the flow is unidirectional. SnapMirror transfers occur based on the SnapMirror policy schedule. You can also perform SnapMirror updates whenever required.
If the source cluster reboots, then the source SVM is operationally stopped and is locked for any management operations to avoid data corruption in case data is accessed inadvertently from both the source and destination SVMs.

**Note:** You must not associate the destination SVM with any other source SVM.

2. Activate the destination SVM either to provide data access if the source SVM is unavailable or if you want to test the SVM disaster recovery setup.

   *Data ONTAP 8.3 SVM Disaster Recovery Express Guide*

   The following illustration shows the impact of a disaster on the source SVM and shows that the destination SVM is activated:

   ![Diagram showing SVM activation](image)

   In this phase, the SnapMirror relationship is in the **Broken-off** state.

3. Depending on whether the source SVM exists, either reactivate the existing source SVM or create and activate a new source SVM.

   - Create and activate a new source SVM.
     
     When the source cluster and SVM are completely destroyed, the cluster administrator of the source cluster creates the cluster and source SVM. The cluster administrator resynchronizes the data and configuration from the destination SVM, activates the source SVM, and protects the new source SVM.

   - Reactivate the existing source SVM.
     
     If the source SVM is available, the cluster administrator resynchronizes the data and configuration from the destination SVM to the existing source SVM, activates the source SVM, and protects the source SVM.

   *Data ONTAP 8.3 SVM Disaster Recovery Express Guide*

   The following illustration shows the configuration and data flow during the reactivation phase of the source SVM:
The data flows from the destination SVM to the source SVM.

Managing SVM disaster recovery relationships

After creating and configuring SVM disaster recovery relationships, you can monitor the relationship status to ensure that the transfers are occurring per the schedules and manage the SnapMirror relationships between the source and destination SVMs.

You can use `snapmirror` commands to manage the SVM disaster recovery relationships. You can specify the source and the destination SVMs as either paths or SVM names. If you want to specify the source and destination SVMs as paths, then the SVM name must be followed by a colon.

You can use the wildcard “*:*” to perform an operation on all the SVM disaster recovery relationships in a cluster.

Monitoring the SVM disaster recovery relationship status

You can monitor the status of the SnapMirror relationship between the source and the destination SVMs to ensure that the updates are occurring based on the schedule.

About this task

SNMP is not supported for monitoring SnapMirror relationships between the source and destination SVMs.

Steps

1. View the details of the SnapMirror relationship status by using the `snapmirror show -instance` command.

Example

```
destination_cluster::> snapmirror show -instance
    Source Path: vs1:
    Destination Path: dvs1:
    Relationship Type: DP
    Relationship Group Type: vserver
    SnapMirror Schedule: -
    SnapMirror Policy Type: async-mirror
    SnapMirror Policy: DFDefault
    Mirror State: Snapmirrored
    Relationship Status: Idle
    ...
    Snapshot Checkpoint: -
    Newest Snapshot: vserverdr.4ebf1aa-
```
2. If the Mirror State is **Uninitialized**, initiate the SnapMirror transfers again.

The SnapMirror initialization or resynchronization operation might have failed because data replication failed at the volume level, configuration replication failed, or other such failures.

   - Complete the initialize operation by running the `snapmirror initialize` command again.
   - Complete the resynchronization operation by running the `snapmirror resync` command again.

3. If the Healthy field is **false** and the Last Transfer Type is **break**, run the `snapmirror break` command again for the SVM disaster recovery relationship.

   Some volume-level SnapMirror relationships might have failed to break during the last SnapMirror break operation on the SVM disaster recovery relationship.

4. Optional: After a cluster reboot, ensure that the source SVM is unlocked for any management operations.

   The source SVM is locked during the cluster reboot.

   a. Verify that the source SVM is locked by using the `vserver show -fields config-lock` command.

   **Example**

   ```
   source_cluster::> vserver show -fields config-lock
   vserver config-lock
   --------  -----------
   vs1        true
   ```

   b. If the source SVM is locked, then unlock it forcibly from the advanced privilege by using the `vserver unlock` command.

   **Example**

   ```
   source_cluster::> set advanced
   Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel.
   Do you want to continue? {y|n}: y
   source_cluster::*> vserver unlock -vserver vs1 -force
   source_cluster::*> vserver show -fields config-lock
   vserver config-lock
   --------  -----------
   vs1        false
   ```
Providing read-only access from the destination SVM

For remote data access or data distribution, you can provide read-only access to the NFS clients and SAN hosts from the destination SVM, if you chose to set the `identity-preserve` option to `false`. You can also verify that the disaster recovery setup by verifying the data access from the destination SVM.

**Before you begin**
- The baseline transfer from the source SVM to the destination SVM must be completed.
- The network and protocols must be configured on the destination SVM.

**About this task**
- You must perform this task from the destination cluster.
- The volumes in the destination SVM are in DP state and therefore can provide read-only access to the clients.
- You cannot provide read-only access if you chose to set the `identity-preserve` option to `true`.

**Steps**
1. Start the destination SVM by using the `vserver start` command.
   
   **Example**
   
   destination_cluster::> vserver start -vserver dvs1

2. Verify that the destination SVM is in the running state and the subtype is `dp-destination` by using the `vserver show` command.
   
   **Example**
   
   destination_cluster::> vserver show

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Type</th>
<th>Subtype</th>
<th>Admin State</th>
<th>Operational Root State</th>
<th>Volume</th>
<th>Aggregate</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvs1</td>
<td>data</td>
<td>dp-destination</td>
<td>running</td>
<td>running</td>
<td>voll</td>
<td>aggr1</td>
</tr>
</tbody>
</table>

**After you finish**

The clients must mount or scan as per the protocol requirements for accessing the data from the destination SVM.

**Updating SVM disaster recovery relationships**

You can either schedule updates or manually perform an update to transfer Snapshot copies between the source and destination SVMs. During the update, all the changes to data and configuration performed on the source SVM are replicated on the destination SVM.

**About this task**

The SnapMirror schedule is applicable to all the volumes and configuration of the source SVM.
**Choices**

- Schedule an update from the source cluster:

  1. Create the schedule you want to implement by using the `job schedule cron create` command.

  2. Apply the schedule to the SVM disaster recovery relationship by using the `-schedule` option of the `snapmirror modify` command.

**Example**

```
source_cluster::> job schedule cron create -name weekly -dayofweek "Sunday" -hour 0 -minute 15

source_cluster::> snapmirror modify -destination-vserver dvs1 -schedule weekly
```

- Perform a manual update from the destination cluster by using the `snapmirror update` command.

**Example**

```
destination_cluster::> snapmirror update -destination-path dvs1:
```

**Modifying the SVM disaster recovery relationship attributes**

After creating the SnapMirror relationship between the source and destination SVMs, you can modify attributes such as the schedule, policy, and throttle.

**Steps**

1. Modify the attributes by using the `snapmirror modify` command.

   The throttle value that you specify is applicable to each of the volumes.

   **Example**

   ```
   destination_cluster::> snapmirror modify -destination-vserver dvs1 -schedule daily -policy policy1 -throttle unlimited
   ```

2. Verify the modified attributes by using the `snapmirror show` command.

   **Example**

   ```
   destination_cluster::> snapmirror show -fields schedule,policy,throttle
   source-path destination-path schedule policy throttle
   --------------- --------------- ------- --------- ---------
   vs1:           dvs1:          daily policy1 unlimited
   ```
Releasing SVM disaster recovery relationships

If the SVM disaster recovery relationship is no longer required, you must release the SnapMirror relationship between the source and destination SVMs. By doing so, resources such as system-created Snapshot copies and configuration replication streams are deleted from the source SVM.

About this task

You must perform this task from the source cluster.

After the SVM disaster recovery relationship is released, you might not be able to re-create the SnapMirror relationship between these source and destination SVMs if there are no common Snapshot copies.

Steps

1. If the SVM disaster recovery relationship is in the Snapmirrored state, break the relationship by using the snapmirror break command.
2. Delete the SVM disaster recovery relationship by using the snapmirror delete command.
3. Release the SVM disaster recovery relationship by using the snapmirror release command. You can use the -relationship-info-only option to preserve the Snapshot copies.

Example

```
source_cluster::> snapmirror release -source-vserver vs1 -destination-vserver dvs1
```

Commands for managing SVM disaster recovery relationships

You can use snapmirror commands to manage SVM disaster recovery relationships. You can specify the source and the destination SVMs as either paths or SVM names. If you want to specify the source and destination SVMs as paths, then the SVM name must be followed by a colon.

You can use the wildcard “*:” to perform an operation on all the SVM disaster recovery relationships in a cluster.

**Important:** You cannot perform volume-level SnapMirror operations when a disaster recovery relationship is configured for the SVM.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abort an active SnapMirror transfer</td>
<td><code>snapmirror abort</code>... You must run this command on the destination cluster.</td>
</tr>
<tr>
<td>Quiesce scheduled SnapMirror transfers</td>
<td><code>snapmirror quiesce</code></td>
</tr>
<tr>
<td>Enable scheduled SnapMirror transfers</td>
<td><code>snapmirror resume</code></td>
</tr>
<tr>
<td>Provide read/write data access from the destination SVM</td>
<td><code>snapmirror break</code></td>
</tr>
</tbody>
</table>
If you want to... | Use this command...  
---|---
Delete the SVM disaster recovery relationship | snapmirror delete

If you delete and re-create the SnapMirror relationship when it is in the Snapmirrored state, the subsequent updates fail. You must resynchronize the relationship before updating it.

Perform an incremental SnapMirror transfer | snapmirror update

Start a resynchronization operation | snapmirror resync

Unlock and manage the source SVM after a cluster reboot | vserver unlock -force

---

Deleting the destination SVM

If the SVM disaster recovery relationship between the source and destination SVMs is no longer required, then you must delete the destination SVM because the same destination SVM must not be part of any other SVM disaster recovery relationship.

Steps

1. Break the SVM disaster recovery relationship between the source and destination SVMs by using the snapmirror break command.

2. Delete the SVM disaster recovery relationship by using the snapmirror delete command.

3. Release the SVM disaster recovery relationship by using the snapmirror release command.

4. Delete the SVM peer relationship by using the vserver peer delete command.

5. If the SVM contains LUNs, unmap the LUNs, take them offline, and delete them.

   *Clustered Data ONTAP 8.3 SAN Administration Guide*

6. Unmount all the volumes, take them offline, and delete them including the SVM root volume.

   *Clustered Data ONTAP 8.3 Logical Storage Management Guide*

7. Stop the destination SVM by using the vserver stop command.

8. Delete the destination SVM by using the vserver delete command.
Protecting data on FlexVol volumes by using SnapVault backups

You can create a SnapVault relationship between FlexVol volumes and assign a SnapVault policy to it to create a SnapVault backup. A SnapVault backup contains a set of read-only backup copies, located on a secondary volume.

**Note:** SnapVault relationships are supported on clusters running Data ONTAP 8.2 or later. SnapVault relationships are not supported on Infinite Volumes.

A SnapVault backup differs from a set of Snapshot copies or a set of mirror copies on a destination volume. In a SnapVault backup, the data in the secondary volume is periodically updated to keep the data in the secondary volume up to date with changes made in the primary data.

Creating SnapVault backups on FlexVol volumes

You configure a SnapVault relationship and assign a SnapVault policy to the relationship to establish a SnapVault backup.

**About this task**

The commands you use to create SnapVault backups are the same commands you use to create data protection mirrors. You must have a SnapVault license before you can create and manage SnapVault relationships.

**Related concepts**

- *Supported data protection deployment configurations* on page 32

**Related references**

- *Commands for managing SnapMirror relationships* on page 92

**Related information**

*Clustered Data ONTAP 8.3 System Administration Guide*

Guidelines for creating SnapVault relationships on FlexVol volumes

You must follow certain guidelines when creating SnapVault relationships.

**General guidelines for creating a SnapVault relationship**

The following guidelines apply to all SnapVault relationships:

- A volume can be in multiple relationships, either as the secondary or the primary.
- A volume can be the primary for multiple secondaries and also the secondary for another primary.
- A volume can be the secondary for only one SnapVault relationship.
- You cannot configure SnapVault relationships from multiple primary volumes to a single SnapVault secondary volume.

For example, if you want to back up an entire Storage Virtual Machine (SVM) to a SnapVault backup, then you must create a separate secondary volume for each volume in the SVM, and create a separate SnapVault relationship for each primary volume.
• You can configure SnapVault relationships to be used simultaneously with data protection mirror relationships.

• Primary or secondary volumes cannot be 32-bit volumes.

• The primary of a SnapVault backup should not be a FlexClone volume. The relationship will work, but the efficiency provided by FlexClone volumes is not preserved.

• Primary and secondary volumes must have the same `vol lang` settings.

• After you establish a SnapVault relationship, you cannot change the language assigned to the secondary volume.

• A SnapVault relationship can be only one leg of a cascade chain.

• After you establish a SnapVault relationship, you can rename primary or secondary volumes. If you rename a primary volume, it can take a few minutes for the relationship to recover from the name change.

Guidelines for creating a SnapVault relationship to a prepopulated secondary

Typically, you create a prepopulated secondary volume when you copy a primary volume to a secondary volume using tape. This process is known as `tape seeding`.

If the SnapVault secondary volume already contains data, you can create a SnapVault relationship by using the `snapmirror resync` command with the `-type XDP` option.

Before creating a SnapVault relationship to a prepopulated secondary, you must use the following guidelines:

• The primary and secondary volumes must have a common Snapshot copy.

• Snapshot copies on the secondary volume that are newer than the common Snapshot copy are deleted.
  
  When a SnapVault relationship is created, all Snapshot copies on the secondary volume that are more recent than the common Snapshot copy and that are not present on the primary volume are deleted. Newer Snapshot copies on the primary volume that match the configured SnapVault policy are transferred to the secondary volume according to the SnapVault policy.

  You can use the `-preserve` option to keep any Snapshot copies that are more recent than the common Snapshot copy on the SnapVault secondary volume and that are not present on the primary volume.

  When you use the `-preserve` option, data on the secondary volume is logically made the same as the common Snapshot copy. All newer Snapshot copies on the primary volume that match the SnapVault policy are transferred to the secondary volume.

  This option is useful when the latest common Snapshot copy is deleted from the primary volume but another, older common Snapshot copy between the primary and secondary volumes still exists.

SnapVault updates fail if destination aggregate is full

If the aggregate that contains the secondary volume of the SnapVault backup is out of space, SnapVault updates fail, even if the secondary volume has space.

Ensure that there is free space in the aggregate and the volume for transfers to succeed.

Prepopulated SnapVault secondary scenarios

There are several ways in which a secondary FlexVol volume for a SnapVault relationship might be prepopulated with data.

The following are some scenarios in which a SnapVault secondary might be populated before a SnapVault relationship is created:
• You used tape backups to provide a baseline transfer to a secondary volume.
   
   **Note:** Disk seeding to establish a baseline is not supported for SnapVault backups.

• A SnapVault primary volume in a cascade becomes unavailable.
   You have a data protection mirror relationship between a source and a destination volume (a mirror relationship from A to B) and a SnapVault relationship between the secondary destination volume and a tertiary destination volume (a SnapVault relationship from B to C). The backup cascade chain is A mirror to B and B SnapVault backup to C. If the volume on B becomes unavailable, you can configure a SnapVault relationship directly from A to C. The cascade chain is now A SnapVault backup to C, where C was prepopulated with data.

• You created a SnapVault relationship between two flexible clones.
   You create a SnapVault relationship between two flexible clones for which their respective parent volumes are already in a SnapVault relationship.

• You extended the SnapVault backup protection beyond 251 Snapshot copies.
   To extend the SnapVault backup protection beyond the volume limit of 251 Snapshot copies, you can clone the secondary volume. The original SnapVault secondary volume is the parent volume for the new flexible clone.

• You restored data from a SnapVault secondary to a new primary volume.
   You have a SnapVault relationship from A to B. A becomes inaccessible, so the SnapVault secondary volume (B) is used for a baseline restore operation to a new SnapVault secondary volume (C).
   After the restore operation finishes, you establish a new SnapVault relationship from the new secondary volume (C), which now becomes the primary volume, and the original SnapVault secondary volume (in other words, C to B). The disk to disk backup relationship is now C to B, where B was prepopulated with data.

• You deleted the base Snapshot copy from the primary volume.
   You deleted the base Snapshot copy from the primary volume that was used for a SnapVault transfer, but another, older Snapshot copy exists that is common between the primary and secondary volumes.

**Creating a SnapVault backup in an empty FlexVol volume**

You can protect data that has long-term storage requirements on a FlexVol volume by replicating selected Snapshot copies to a SnapVault backup on another Storage Virtual Machine (SVM) or cluster.

**Before you begin**

• You must have cluster administrator privileges to perform this task for a cluster, and SVM administrator privileges to perform this task for an SVM.

• If the primary and secondary volumes are in different SVMs, the SVMs must be in a peer relationship.
  If the primary and secondary volumes are in different clusters, the clusters must be in a peer relationship.
  For information about creating peer relationships, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.

• A SnapVault policy must exist.
  You must either create one or accept the default SnapVault policy (named **XDPDefault**) that is automatically assigned.
  Only Snapshot copies with the labels configured in the SnapVault policy rules are replicated in SnapVault operations.
• The Snapshot policy assigned to the primary volume must include the `snapmirror-label` attribute.
  You can create a new Snapshot policy by using the `volume snapshot policy create` command, or you can modify an existing policy by using the `volume snapshot policy modify` command to set the `snapmirror-label` attribute for the set of Snapshot copies that you want backed up to the SnapVault secondary volume. Other Snapshot copies on the primary volume are ignored by the SnapVault relationship.

• Your work environment must be able to accommodate the time it might take to transfer a baseline Snapshot copy with a large amount of data.

Steps

1. On the destination SVM, create a SnapVault secondary volume with a volume type DP.
   For information about creating a FlexVol volume, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.

2. Create a schedule that Data ONTAP uses to update the SnapVault relationship by using the `job schedule cron create` command.
   For more information, see *Scheduling SnapMirror transfers* on page 102.

   Example

   The following command creates a schedule that runs on the weekend at 3 a.m.:

   ```bash
   vserverB::> job schedule cron create -name weekendcron -dayofweek "Saturday, Sunday" -hour 3 -minute 0
   ```

3. On the source SVM, create a Snapshot copy policy that contains the schedule of when Snapshot copies with `snapmirror-label` attributes occur by using the `volume snapshot policy create` command with the `snapmirror-label` parameter, or use the default Snapshot copy policy called default.

   Example

   The following command creates a Snapshot copy policy called “keep-more-snapshot”:

   ```bash
   vserverB::> snapshot policy create -vserver vs1 -policy keep-more-snapshot -enabled true -schedule1 weekly -count1 2 -prefix1 weekly -snapmirror-label1 weekly -schedule2 daily -count2 6 -prefix2 daily -snapmirror-label2 daily -schedule3 hourly -count3 8 -prefix3 hourly -snapmirror-label3 hourly
   ```

   The name specified in the `snapmirror-label` attribute for the new Snapshot policy must match the `snapmirror-label` attribute that is specified in the SnapVault policy. This ensures that all subsequent Snapshot copies created on the primary volume have labels that are recognized by the SnapVault policy.

   The default Snapshot copy policy has two snapmirror-label attributes associated with it, daily and weekly.

4. Create a SnapVault policy by using the `snapmirror policy create` command, or use the default SnapVault policy called `XDPDefault`.

   Example

   The following command creates a SnapVault policy called “vserverB-vault-policy”:
5. Add the snapmirror-label attribute to the SnapVault policy you created by using the `snapmirror policy add-rule` command.

If you used the XDPDefault SnapMirror policy, you do not need to perform this step. The XDPDefault SnapVault policy uses the daily and weekly snapmirror-label attributes specified by the default Snapshot copy policy.

Example

The following command adds a rule to the vserverB-vault-policy to transfer Snapshot copies with the “weekly” snapmirror-label attribute and to keep 40 Snapshot copies:

```bash
vserverB::> snapmirror policy add-rule -vserver vserverB -policy vserverB-vault-policy -snapmirror-label weekly -keep 40
```

6. On the destination SVM, create a SnapVault relationship and assign a SnapVault policy by using the `snapmirror create` command with the `type XDP` parameter and the `policy` parameter.

In the path specification, a single name is interpreted as a volume name in the SVM from which the command is executed. To specify a volume in a different SVM or in a different cluster, you must specify the full path name.

Example

The following command creates a SnapVault relationship between the primary volume “srcvolA” on SVM “vserverA” and the empty secondary volume “dstvolB” on SVM “vserverB”. It assigns the SnapVault policy named “vserverB-vault-policy” and uses the “weekendcron” schedule:

```bash
vserverB::> snapmirror create -source-path vserverA:srcvolA -destination-path vserverB:dstvolB -type XDP -policy vserverB-vault-policy -schedule weekendcron
```

7. On the destination SVM, initialize the SnapVault relationship by using the `snapmirror initialize` command to start a baseline transfer.

The command creates a new Snapshot copy that is transferred to the secondary volume and used as a baseline for subsequent incremental Snapshot copies. The command does not use any Snapshot copies that currently exist on the primary volume.

**Note:** Creating a baseline for a large amount of data might take a while.

Example

The following command begins the relationship initialization by creating and transferring a baseline Snapshot copy to the destination volume “dstvolB” on SVM “vserverB”:

```bash
vserverB::> snapmirror initialize -destination-path vserverB:dstvolB
```

Related concepts

- Guidelines for creating SnapVault relationships on FlexVol volumes on page 128

Related references

- Commands for managing SnapMirror and SnapVault policies on page 47
Creating the SnapVault relationship of a mirror-SnapVault cascade

The SnapVault relationship of a mirror-SnapVault cascade requires a different configuration from a SnapVault relationship that is not a part of a mirror-SnapVault cascade.

Before you begin

- You must have cluster administrator privileges to perform this task for a cluster, and Storage Virtual Machine (SVM) administrator privileges to perform this task for an SVM.
- If the primary and secondary volumes are in different SVMs, the SVMs must be in a peer relationship.
- If the primary and secondary volumes are in different clusters, the clusters must be in a peer relationship.

For information about creating peer relationships, see the Clustered Data ONTAP System Administration Guide for Cluster Administrators.

About this task

The Snapshot copies that are exported to the mirror destination are ones that are created by Data ONTAP. These Snapshot copies are called “sm_created” Snapshot copies. Only these Snapshot copies are replicated from the mirror to the SnapVault backup. If the default SnapVault policy is used, the SnapVault secondary accumulates up to 251 “sm_created” Snapshot copies. The next Snapshot copy transferred after this limit is reached will be added and the oldest “sm_created” Snapshot copy will be rotated out. You can manage this retention and rotation behavior by adding a rule specifying the “sm_created” SnapMirror label to the default SnapVault policy.

Steps

1. On the destination SVM, create a SnapVault secondary volume with a volume type DP.

   For information about creating a FlexVol volume, see the Clustered Data ONTAP System Administration Guide for Cluster Administrators.

2. Create a SnapVault policy by using the `snapmirror policy create` command, or use the default SnapVault policy called XDPDefault.

   Example

   This procedure uses the XDPDefault policy in the examples.

3. Add the sm_created snapmirror-label to the SnapVault policy by using the `snapmirror policy add-rule` command.

   Only the sm_created rule is observed. Any other rules associated with the SnapVault policy, such as the daily or weekly rule, are disregarded.

   Example

   The following command adds a rule to the XDPDefault policy to retain 40 sm_created Snapshot copies on the SnapVault secondary:

   ```bash
   vserverB::> snapmirror policy add-rule -vserver vserverC -policy XDPDefault -snapmirror-label sm_created -keep 40
   ```

4. On the destination SVM, create a SnapVault relationship and assign a SnapVault policy by using the `snapmirror create` command with the type XDP parameter and the policy parameter.
Example
The following command creates a SnapVault relationship between the primary volume srcvolB on SVM vserverB and the empty secondary volume dstvolC on SVM vserverC. It assigns the SnapVault policy named XDPDefault:

```bash
vserverC::> snapmirror create -source-path vserverB:srcvolB
            -destination-path vserverC:dstvolC -type XDP -policy XDPDefault
```

5. On the destination SVM, initialize the SnapVault relationship by using the `snapmirror initialize` command to start a baseline transfer.

   **Note:** Creating a baseline for a large amount of data can take many hours.

Example
The following command begins the relationship initialization by creating and transferring a baseline Snapshot copy to the secondary volume dstvolC on SVM vserverC:

```bash
vserverC::> snapmirror initialize -destination-path vserverC:dstvolC
```

Related tasks
* How a SnapMirror-SnapVault cascade works on page 35

Preserving a Snapshot copy on the primary source volume
In a mirror-SnapVault cascade, you must preserve a Snapshot copy on the primary source volume until it transfers to the secondary volume of the SnapVault backup. For example, you want to ensure that application-consistent Snapshot copies are backed up.

**Before you begin**
You must have created the mirror-SnapVault cascade.

**Steps**
1. Ensure that the Snapshot copy you want to preserve has a snapmirror-label by using the `volume snapshot show` command.

2. If the Snapshot copy does not have a snapmirror-label associated with it, add one by using the `volume snapshot modify` command.

Example
The following command adds a snapmirror-label called “exp1” to the Snapshot copy called “snapappa”:

```bash
clust1::> volume snapshot modify -volume vol1 -snapshot snapappa
            -snapmirror-label expl
```

3. Preserve the Snapshot copy on the source volume by using the `snapmirror snapshot-owner create` command to add an owner name to the Snapshot copy.

Example
The following command adds ApplicationA as the owner name to the snap1 Snapshot copy in the testvol volume on the vs1 Storage Virtual Machine (SVM):
4. Update the destination volume of the data protection mirror relationship by using the `snapmirror update` command.
   Alternatively, you can wait for the scheduled update of the data protection mirror relationship to occur.

5. Update the secondary volume of the SnapVault relationship to transfer the specific Snapshot copy from the SnapMirror destination volume to the SnapVault secondary volume by using the `snapmirror update` command with the `-source-snapshot` parameter.

6. Remove the owner name from the primary source volume by using the `snapmirror snapshot-owner delete` command.

Example
The following command removes ApplicationA as the owner name to the snap1 Snapshot copy in the testvol volume on the vs1 SVM:

```
clust1::> snapmirror snapshot-owner delete -vserver vs1 -volume voll
          -snapshot snapappa -owner ApplicationA
```

Creating a SnapVault backup in a prepopulated FlexVol volume

You can protect data that has long-term storage requirements on a FlexVol Volume by replicating selected Snapshot copies to a SnapVault backup on another Storage Virtual Machine (SVM) or cluster. The SnapVault secondary volume might contain data that already exists from a previous data protection mirror or SnapVault relationship or has been loaded from a tape backup.

Before you begin
- You must have cluster administrator privileges to perform this task for a cluster, and you must have SVM administrator privileges to perform this task for an SVM.
- If the primary and secondary volumes are in different SVMs, the SVMs must be in a peer relationship.
  - If the primary and secondary volumes are in different clusters, the clusters must be in a peer relationship.
  - For information about creating peer relationships, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.
- The secondary volume must be prepopulated with data.
- A SnapVault policy must exist.
  - You must either create one or accept the default SnapVault policy (named *XDPDefault*) that is automatically assigned.
  - The SnapVault policy configuration includes the `snapmirror-label` attribute that is used to select Snapshot copies on the primary volume and match Snapshot copies between the primary and secondary volumes. Only Snapshot copies with the labels configured in the SnapVault policy rules are replicated in SnapVault operations.
- The Snapshot policy assigned to the primary volume must include the `snapmirror-label` attribute.
  - The name specified in the `snapmirror-label` attribute for the new Snapshot policy must match the `snapmirror-label` attribute that is specified in the SnapVault policy. This ensures that all subsequent Snapshot copies created on the primary volume have labels that are recognized by the SnapVault policy.
You can create a new Snapshot policy by using the `volume snapshot policy add-schedule`, or you can modify an existing Snapshot policy by using the `volume snapshot policy modify-schedule` command to set the `snapmirror-label` attribute for the set of Snapshot copies that you want replicated to the SnapVault secondary volume. Other Snapshot copies on the primary volume are ignored by the SnapVault relationship.

- Your work environment must be able to accommodate the time it might take to transfer a baseline Snapshot copy with a large amount of data.

**Step**

1. On the destination SVM, establish the relationship by using the `snapmirror resync` command and the `-type XDP` parameter.

   If the most recent common Snapshot copy between the primary and the secondary is deleted from the primary but there exists another, older common Snapshot copy, you can also use the `-preserve` option. This option performs a logical local rollback to make the data in the primary and the secondary the same, and then it replicates all newer Snapshot copies from the source that match the SnapVault policy.

**Example**

The following command creates a SnapVault relationship between the primary volume `srcvolA` on SVM `vserverA` and the prepopulated secondary volume `dstvolB` on SVM `vserverB`:

```
  vserverB::> snapmirror resync -source-path vserverA:srcvolA -destination-path vserverB:dstvolB -type XDP
```

**Related concepts**

- Guidelines for creating SnapVault relationships on FlexVol volumes on page 128
- Prepopulated SnapVault secondary scenarios on page 129

**Related tasks**

- Creating a destination baseline using a tape backup on page 136

**Related references**

- Commands for managing SnapMirror and SnapVault policies on page 47

### Creating a destination baseline using a tape backup

You can perform a baseline transfer from local tape copies to a SnapVault secondary volume to manage your bandwidth or timing constraints over a network.

**Before you begin**

- You must have cluster administrator privileges to perform this task for a cluster.
- You must have Storage Virtual Machine (SVM) administrator privileges to perform this task for an SVM.
- The destination volume must not contain data.

**About this task**

This operation physically copies data from tape to one or more secondary volumes. When the operation finishes, the secondary volume contains all the Snapshot copies that existed on the primary volume at the time the tape copy was created.
Steps

1. Create a copy of the primary volume on the tape by using the `system smtape backup` command.
   
   For information about backing up and restoring from tape, see *Performing tape seeding using SMTape* on page 95.

2. Restore the data to the empty secondary volume from the tape copy.
   
   For information about backing up and restoring from tape, see *Performing tape seeding using SMTape* on page 95.

3. Initialize the SnapVault relationship by using the `snapmirror resync` command with the `-type XDP` parameter on the secondary volume, and enable incremental updates.

Converting a SnapVault relationship to a version flexible SnapMirror relationship

You can convert a SnapVault relationship to a version flexible SnapMirror relationship to better control the Snapshot copies you replicate and retain.

About this task

Converting from a SnapVault relationship to a version-flexible SnapMirror relationship requires only a change of policy with its associated rules and retention. What Data ONTAP does with existing Snapshot copies on the destination volume depends on how you set up the policy.

- If you set up the policy to replicate and retain only the latest SnapMirror created Snapshot copies, or if the policy applies to SnapVault and SnapMirror replication on the same volume, Data ONTAP does nothing to the current backup Snapshot copies on the destination volume. You must delete the Snapshot copies from the destination if you do not want them.

- If you set up the policy as a SnapMirror style of replication in which the policy indicates a symmetric Snapshot copy selection and retention, Data ONTAP deletes the backup Snapshot copies on the destination volume. Only the Snapshot copies that are present on the source volume are kept on the destination volume.

Steps

1. Create a policy that the version flexible SnapMirror relationship will use by using the `snapmirror policy create` command with the `-type async-mirror or -type mirror-vault` parameter.

   **Example**

   The following command creates a version flexible SnapMirror policy called “vserverB-DR-policy” that will be used for SnapMirror style disaster recovery in which only the SnapMirror created Snapshot copy is transferred:

   ```shell
   cluster2::> snapmirror policy create -vserver vserverB -policy vserverB-DR-policy 
   -policy-type async-mirror -comment "DR policy"
   ```

2. Apply the policy to the existing SnapVault relationship by using the `snapmirror modify` command with the `-policy` parameter.
3. Update the relationship, which is now a version flexible SnapMirror relationship, by using the `snapmirror update` command.

Example

vs2::> snapmirror update -destination-path vserverB:dstvolB

Converting a data protection destination to a SnapVault secondary

You convert a data protection destination volume to a SnapVault secondary volume after a tape seeding operation or after you lose a SnapVault secondary volume in a backup to disaster protection mirror cascade.

Before you begin

- You must have cluster administrator privileges to perform this task for a cluster.
- You must have Storage Virtual Machine (SVM) administrator privileges to perform this task for an SVM.

About this task

In the case of tape seeding, after you transfer the data from the tape to the volume, the volume is a data protection destination volume.

In the case of a SnapVault secondary volume to disaster protection volume cascade, if the SnapVault secondary volume is lost, you can resume SnapVault protection by creating a direct relationship between the SnapVault primary volume and the disaster protection destination volume. You must make the disaster protection destination volume a SnapVault secondary volume to do this.

Steps

1. Break the data protection mirror relationship by using the `snapmirror break` command.
   The relationship is broken and the disaster protection volume becomes a read-write volume.

2. Delete the existing data protection mirror relationship, if one exists, by using the `snapmirror delete` command.

3. Remove the relationship information from the source SVM by using the `snapmirror release` command.
   This also deletes the Data ONTAP created Snapshot copies from the source volume.

4. Create a SnapVault relationship between the primary volume and the read-write volume by using the `snapmirror create` command with the `-type XDP` parameter.

5. Convert the destination volume from a read-write volume to a SnapVault volume and establish the SnapVault relationship by using the `snapmirror resync` command.
Managing backup and restore operations for SnapVault backups

You configure SnapVault relationships on FlexVol volumes to establish SnapVault backups. You manage SnapVault relationships to optimize the performance of the relationships.

Backing up from a Snapshot copy that is older than the base Snapshot copy

You might want to replicate a special, manually initiated Snapshot copy to the SnapVault backup. The Snapshot copy is one that is not in the sequence scheduled by the SnapVault policy assigned to the SnapVault relationship.

Before you begin

You must have cluster administrator privileges to perform this task for a cluster. You must have Storage Virtual Machine (SVM) administrator privileges to perform this task for an SVM.

Step

1. Begin the backup transfer of the older Snapshot copy by using the `snapmirror update` command.

Example

The following command starts an out-of-order transfer of Snapshot copy SC3 from the source volume `srcvolA` on SVM `vserverA` and the secondary volume `dstvolB` on SVM `vserverB`:

```
vserverA::> snapmirror update -source-path vserverA:srcvolA -destination-path vserverB:dstvolB -snapshot SC3
```

Result

After the backup finishes, the transferred Snapshot copy becomes the base.

How an out-of-order Snapshot copy transfer works

The transfer of a Snapshot copy that does not conform to the usual sequence scheduled by a SnapVault policy is an out-of-order Snapshot copy transfer.

In SnapVault relationships, Snapshot copies are selected and transferred from the primary volume to the secondary volume, according to the configured SnapVault policy. Only Snapshot copies that are newer than the common Snapshot copy between the primary and secondary volume are transferred. However, you can use the `snapmirror update` command to initiate the transfer of a Snapshot copy that was not originally selected and transferred.

When you initiate an out-of-order transfer, an older Snapshot copy is used to establish the base. To avoid subsequent transfers of Snapshot copies that already exist on the SnapVault secondary volume, the list of Snapshot copies that are selected for transfer in this update cycle are reconciled against the Snapshot copies that are already present on the secondary volume. Snapshot copies that are already present on the secondary volume are discarded from the transfer list.

Example of a new base that is established from an out-of-order Snapshot copy transfer

In this example, the SnapVault policy has a schedule in which only the even-numbered Snapshot copies on the primary volume are transferred to the secondary volume. Before the out-of-order transfer begins, the primary volume contains Snapshot copies 2 through 6; the
secondary volume contains only the even-numbered Snapshot copies (noted as “SC” in the figures). Snapshot copy 4 is the common Snapshot copy that is used to establish the base, as shown in the following figure:

![Snapshot copy diagram](image)

After Snapshot copy 3 is transferred to the secondary volume, out of order, it becomes the new common Snapshot copy that is used to establish the base, as shown in the following figure:

![Snapshot copy diagram](image)

**Note:** Although Snapshot copy 3 is now the base, the exported Snapshot copy is still Snapshot copy 4.

When Snapshot copies are selected for subsequent updates according to the SnapVault policy, the policy selects Snapshot copy 4 and Snapshot copy 6 for transfer to the secondary volume. When the transfer list is reconciled, Snapshot copy 4 is removed from the transfer list because it already exists on the secondary volume. Only Snapshot copy 6 is transferred, which becomes the new common Snapshot copy that is used to establish the base, as shown in the following figure:
Example of SnapVault transfer behavior with an out-of-order Snapshot copy transfer

In this example, Data ONTAP created two SnapVault-labeled Snapshot copies, you made a user-created Snapshot copy, and then more SnapVault-labeled Snapshot copies were created. The order of Snapshot copies made would appear as shown in the following figure:

You perform an out-of-order Snapshot transfer using the user-created Snapshot copy, which establishes that Snapshot copy as the new base Snapshot copy, as shown in the following figure:
When the next SnapVault scheduled transfer occurs, only the SnapVault labeled Snapshot copies made after the user-created Snapshot copy are transferred. This occurs because the Snapshot copies created between the previous base Snapshot copy and the current base Snapshot copy are not transferred.

**Backing up FlexVol volumes that contain the maximum limit of Snapshot copies**

To work around the limit of 251 Snapshot copies per volume, you can create a new destination volume clone, then establish a SnapVault relationship with the new clone.

**Before you begin**

You must have cluster administrator privileges to perform this task for a cluster. You must have Storage Virtual Machine (SVM) administrator privileges to perform this task for an SVM.

**About this task**

Creating a new SnapVault relationship to a new volume clone enables you to continue SnapVault protection with minimum disruption on the clone volume and without starting a new baseline transfer. Because the source clone and the volume clone share the latest common Snapshot copy, subsequent updates are performed as usual, according to the policy assigned to the SnapVault relationship.
Steps

1. Quiesce the SnapVault relationship between the primary volume and the secondary volume by using the `snapmirror quiesce` command.
   This step prevents updates from starting until after the task is complete.

2. Verify that there are no active transfers on the relationship by using the `snapmirror show` command.
   The Relationships field should be Idle.

3. Create a volume clone based on the most recent common Snapshot copy between the SnapVault primary volume and the SnapVault secondary volume by using the `volume clone create` command with the `-type DP` parameter.

4. Establish the SnapVault relationship between the primary volume and the newly created secondary volume clone by using the `snapmirror resync` command and the `-type XDP` parameter.

5. Delete the SnapVault relationship between the primary volume and the original SnapVault secondary volume by using the `snapmirror delete` command.

Managing the backup of a copied source volume

If you use the `volume copy` command to copy the primary volume of a SnapVault relationship to a different volume, Data ONTAP does not copy SnapMirror labels for Snapshot copies, and you lose the capability to back up from the primary volume copy.

About this task

You must add the SnapMirror labels back before you can back up the volume copy.

Step

1. Add the SnapMirror labels to the copied volume by using the `volume snapshot modify` command or by using the `snapmirror update -s` command.

Guidelines for restoring the active file system

The restore operation from a SnapVault backup copies a single, specified Snapshot copy from a SnapVault secondary volume to a specified volume. Restoring a volume from a SnapVault secondary volume changes the view of the active file system but preserves all earlier Snapshot copies in the SnapVault backup.

Before restoring a volume, you must shut down any application that accesses data in a volume to which a restore is writing data. Therefore, you must dismount the file system, shut down any database, and deactivate and quiesce the Logical Volume Manager (LVM) if you are using an LVM.

The restore operation is disruptive. When the restore operation finishes, the cluster administrator or Storage Virtual Machine (SVM) administrator must remount the volume and restart all applications that use the volume.

The restore destination volume must not be the destination of another mirror or the secondary of another SnapVault relationship.

You can restore to the following volumes:

- Original source volume
  You can restore from a SnapVault secondary volume back to the original SnapVault primary volume.
- New, empty secondary volume
You can restore from a SnapVault secondary volume to a new, empty secondary volume. You must first create the volume as a data protection (DP) volume.

- New secondary that already contains data
  You can restore from a SnapVault secondary volume to a volume that is prepopulated with data. The volume must have a Snapshot copy in common with the restore primary volume and must not be a DP volume.

**Guidelines for restoring LUNs in SAN environments**

The restore operation from a SnapVault backup copies a single, specified LUN from a SnapVault secondary volume to a specified volume. Restoring a LUN from a SnapVault secondary volume changes the view of the active system on the volume to which data is being restored, preserving all earlier Snapshot copies.

The following guidelines apply only to SAN environments:

- You can restore a single file or single LUN from a SnapVault secondary volume by using the NetApp OnCommand management software online management tools.
- When LUNs are restored to existing LUNs, new access controls do not need to be configured. You must configure new access controls for the restored LUNs only when restoring LUNs as newly created LUNs on the volume.
- If LUNs on the SnapVault secondary volume are online and mapped before the restore operation begins, they remain so for the duration of the restore operation and after the operation finishes.
- The host system can discover the LUNs and issue non-media access commands for the LUNs, such as inquiries or commands to set persistent reservations, while the restore operation is in progress.
- You cannot create new LUNs in a volume during a restore operation with the `lun create` command.
- Restore operations from tape and from a SnapVault backup are identical.
- You cannot restore a single LUN from a SnapVault secondary volume that is located on a system that is running in 7-Mode.

For more information about backing up and restoring data in a SAN environment, see the *Clustered Data ONTAP SAN Administration Guide*.

**How restore operations work from a SnapVault backup**

A restore operation from a SnapVault backup consists of a series of actions performed on a temporary restore relationship and on the secondary volume.

During a restore operation, the following actions occur:

1. A new temporary relationship is created from the restore source (which is the original SnapVault relationship secondary volume) to the restore destination.
   The temporary relationship is a restore type (RST). The `snapmirror show` command displays the RST type while the restore operation is in progress.
   The restore destination might be the original SnapVault primary or might be a new SnapVault secondary.

2. During the restore process, the restore destination volume is changed to read-only.

3. When the restore operation finishes, the temporary relationship is removed and the restore destination volume is changed to read-write.
Restoring a volume from a SnapVault backup

If the data on a volume becomes unavailable, you can restore the volume to a specific time by copying a Snapshot copy in the SnapVault backup. You can restore data to the same primary volume or to a new location. This is a disruptive operation.

Before you begin

- You must have cluster administrator privileges to perform this task for a cluster.
- You must have Storage Virtual Machine (SVM) administrator privileges to perform this task for an SVM.
- CIFS traffic must not be running on the SnapVault primary volume when a restore operation is running.

About this task

This task describes how to restore a whole volume from a SnapVault backup. To restore a single file or LUN, you can restore the whole volume to a different, non-primary volume, and then select the file or LUN, or you can use the NetApp OnCommand management software online management tools.

Steps

1. If the volume to which you are restoring has compression enabled and the secondary volume from which you are restoring does not have compression enabled, disable compression.
   
   You disable compression to retain storage efficiency during the restore.

2. Restore a volume by using the `snapmirror restore` command.

   Example

   ```
   vs1::> snapmirror restore -destination-path vs1:vol1
   -source-path vs2:vol1_dp_mirror2 -source-snapshot snap3
   Warning: All data newer than Snapshot copy snap6 on volume vs1:vol1
   will be deleted.
   Do you want to continue? {y|n}: y
   [Job 34] Job is queued: snapmirror restore from source vs2:vol1_dp_mirror2 for the snapshot snap3.
   ```

   For more information about the `snapmirror restore` command, see the man pages.

3. If the volume had quotas before the restore operation, activate the quotas on the restored volume by using the `volume quota modify` command with the `-state` parameter.

   Quotas are not turned on when you restore a volume.

4. Remount the restored volume and restart all applications that use the volume.

5. If you previously disabled compression, reenable compression on the volume.
Restoring a single file or LUN

You can restore a single file or LUN or a set of files or LUNs from a Snapshot copy in a SnapVault secondary volume to the active file system of a primary volume. You can restart a failed or aborted single file or LUN restore operation by reissuing the `snapmirror restore` command.

Before you begin

- The primary volume must be a read/write volume.
- The primary volume must not be the destination of a data protection mirror relationship. However, the primary volume can be the source volume of a data protection mirror relationship.
- The secondary and primary volumes must not be the source or destination of a load-sharing mirror relationship. However, the secondary volume can be the source volume of a data protection mirror relationship and the secondary volume in other single file or LUN restore operations.
- Source path of each file or LUN that is copied from the Snapshot copy must be available. Each file or LUN on the secondary volume is copied to the same path in the active file system of the primary volume unless a different destination path is specified.

About this task

Multiple concurrent single file or LUN restore operations cannot be performed on the primary volume (to which file or LUNs are restored) if the volume is already in a single file or LUN restore operation.

An incremental restore occurs if the SnapVault secondary volume and the primary volume have common Snapshot copies and the file or LUN in the Snapshot copy on the primary volume is a different version from the file or LUN that is being restored. Otherwise, a baseline restore occurs.

During a baseline restore, one of the following occurs:

- If the restored file or LUN does not exist on the primary volume, then a new file or LUN is created on the primary volume.
  The data from the file or LUN on the SnapVault secondary volume is copied to the new file or LUN on the primary volume.
- If the restored file or LUN exists on the primary volume, then the data from the file or LUN in the SnapVault secondary volume replaces the data on the existing file or LUN in the primary volume.

Steps

1. Use the `snapmirror restore` command with the `-source-snapshot` and `-file-list` parameters to restore a single file or LUN or a set of files or LUNs from a Snapshot copy in the SnapVault secondary volume to the primary volume.

Example

The following command restores the files `file1` and `file2` from the Snapshot copy `snap1` in the SnapVault secondary volume `secondary1` to the same location of the active file system on the primary volume `primary1`:

```
```
Example
The following command restores the files file1 and file2 from the Snapshot copy snap1 in the SnapVault secondary volume secondary1 to a different location on the primary volume primary1:

The destination file path begins with the @ symbol followed by the path of the file from the root of the active file system of the primary volume. In this example, file1 is restored to @/dir1/file1.new and file2 is restored to @/dir2.new/file2 on primary1.

Example
The following command restores the files file1 and file3 from the Snapshot copy snap1 in the SnapVault secondary volume secondary1 to different locations on the primary volume primary1, and restores file2 from snap1 to the same location in the active file system on primary1:

The file file1 is restored to @/dir1/file1.new and file3 is restored to @/dir3.new/file3

2. Optional: Use the snapmirror show command with the -file-restore-file-list parameter to display the files that you are restoring to keep an account of the files you are restoring.

The file list that is displayed uses the UTF-8 Unicode format.

3. If the single file or LUN restore operation fails or is aborted, rerun the snapmirror restore command on the primary volume.

How single file or LUN restore works
If a file or LUN is accidentally deleted, modified, or corrupted, you can restore such a file or LUN from a Snapshot copy to a specific point in time. You can copy the file or LUN to a new location.

In a single file or LUN restore operation, a set of files or LUNs or a single file or LUN from a single Snapshot copy is copied from one volume to another volume. The volume from which the file or LUN is restored need not be a SnapVault secondary volume, and the volume to which the file or LUN is restored is not restricted to the volume from which the files or LUNs were initially backed up. You can restore the file or LUN from a volume that is not a SnapVault secondary volume, and you can restore to a volume that is not the volume from which the files or LUNs were originally backed up.

In single file or LUN restore operation, the volumes need not have a common Snapshot copy. If a common Snapshot copy exists, an incremental restore is performed for those files or LUNs that exist in the common Snapshot copy.
**Note:** You cannot change the compression capability when the single file restore operation is in progress.

**Which data does not get restored during a single file or LUN restore**

You need to be aware of the type of files that cannot be restored in a single file or LUN restore operation. Only regular files and LUNs and their associated streams can be restored in this operation. However, just a stream of a file or LUN by itself cannot be restored.

Certain types of files cannot be restored during a single file or LUN restore operation such as the following:

- Symbolic links
- Junctions
- Directories
  Only files within a directory can be restored.
- UNIX domain sockets
- Special files in UNIX systems such as a device, block, and character

**Cleaning up a failed single file or LUN restore operation**

If you do not want to restart a failed or aborted single file or LUN restore operation, you can clean up the partially restored files or LUNs on the primary volume.

**About this task**

In the following scenarios, LUNs with client I/O restriction are not removed and therefore you must manually remove the LUNs:

- When the `-clean-up-failure` parameter is used to clean up the partially restored files or LUNs.
  This parameter removes files with client I/O restriction.
- When a single file or LUN restore operation is aborted by using the `-hard` parameter.
  This parameter deletes the files being restored.

**Choices**

- If you want to remove a partially restored file on the primary volume, use the `snapmirror restore` with the `-clean-up-failure` parameter.
- If you want to remove a partially restored LUN on the primary volume, manually delete the LUN that has a client I/O restriction.
  1. Use the `lun show` command with the `-restore-inaccessible` parameter to view the LUN that has a client I/O restriction.
  2. Use the `lun delete` command with the `-force-fenced` parameter to delete the LUN that has a client I/O restriction.
Managing a SnapVault-mirror cascade when the SnapVault backup is unavailable

You can manipulate relationships in a SnapVault-mirror cascade to maintain data backup relationships if the secondary of the SnapVault relationship becomes unavailable.

Before you begin
You must have a SnapVault-mirror cascade already configured.

About this task
The destination of the SnapVault relationship is the middle of the SnapVault-mirror cascade. If it becomes unavailable, you might have the following issues:

• You cannot update the SnapVault backup.
• You cannot update the mirror copy of the SnapVault secondary.

To manage this issue, you can temporarily remove the SnapVault secondary volume from the cascade and establish a SnapVault relationship to the mirror copy of the SnapVault secondary volume. When the unavailable secondary volume becomes available, you can reestablish the original cascade configuration.

In the following steps, the primary volume of the cascade is called “A”, the secondary volume of the SnapVault relationship is called “B”, and the destination volume of the data protection mirror relationship is called “C”.

Steps
1. Identify the current exported Snapshot copy on C by using the volume snapshot show command with the -fields busy parameter.
   The busy field is set to true for the exported Snapshot copy.

   Example
   
   `volume snapshot show C -fields busy`

2. Break the data protection mirror relationship by using the snapmirror break command on C.

   Example
   
   `snapmirror break C`

3. Create a dummy snapmirror-label on the exported Snapshot copy you previously identified by using the volume snapshot modify command with the -snapmirror-label parameter.
   If a snapmirror-label already exists for the exported Snapshot copy, you do not need to perform this step.

   Example
   
   `volume snapshot modify -volume C -snapshot name -snapmirror-label expl`

4. Create a Snapshot owner on the exported Snapshot copy of C by using the snapmirror snapshot-owner create command.
   This prevents clustered Data ONTAP from deleting the Snapshot copy.
Example
snapmirror snapshot-owner create -volume C -snapshot exported -owner admin1

5. Delete the data protection mirror relationship between B and C by using the `snapmirror delete` command.

Example
snapmirror delete C

6. Create the SnapVault relationship between A and C by using the `snapmirror resync` command and the `-type XDP` parameter.

Example
snapmirror resync -source-path A -destination-path C -type XDP

You can maintain this SnapVault relationship until you recover the original SnapVault secondary volume. At that time, you can reestablish the original cascade relationship by using the steps that follow this step.

7. Delete the data protection mirror relationship between A and B by using the `snapmirror delete` command.

8. Perform a disaster recovery resynchronization from C to B by using the `snapmirror resync` command.

Example
snapmirror resync -source-path C -destination-path B

This step copies from C to B, all of the Snapshot copies made after B became unavailable.

9. Identify the current exported Snapshot copy on B by using the `volume snapshot show` command with the `-fields busy` parameter.

Example
volume snapshot show B -fields busy

The `busy` field is set to `true` for the exported Snapshot copy.

10. Break the data protection mirror relationship by using the `snapmirror break` command on B.

Example
snapmirror break B

11. Create a dummy `snapmirror-label` on the exported Snapshot copy you previously identified by using the `volume snapshot modify` command with the `-snapmirror-label` parameter.

Example
volume snapshot modify -volume B -snapshot name -snapmirror-label exp2

If a `snapmirror-label` already exists for the exported Snapshot copy, you do not need to perform this step.

12. Create a Snapshot owner on the exported Snapshot copy of B by using the `snapmirror snapshot-owner create` command.

This prevents clustered Data ONTAP from deleting the Snapshot copy.
Example
snapmirror snapshot-owner create -volume B -snapshot exported -owner admin

13. Delete the data protection mirror relationship between C and B by using the snapmirror delete command.

14. Perform a SnapVault resynchronization from A to B by using the snapmirror resync command and the -type XDP parameter.

Example
snapmirror resync -source-path A -destination-path B -type XDP

New Snapshot copies that meet the Snapshot policy of the SnapVault relationship are transferred from A to B.

15. Delete the data protection mirror relationship between A and C by using the snapmirror delete command.

16. Perform a disaster recovery resynchronization from B to C by using the snapmirror resync command.

This step copies from B to C, all of the Snapshot copies made after reestablishing the A to B relationship without deleting any Snapshot copies on C.

Example
snapmirror resync -source-path B -destination-path C

17. Remove the Snapshot copy owner from volumes B and C by using the snapmirror snapshot-owner delete command.

Example
snapmirror snapshot-owner delete -volume B -snapshot exported_snap

18. Remove SnapMirror labels that you created from volumes B and C by using the snapshot modify command.

Example
snapshot modify -volume B -snapshot exported_snap -snapmirror-label text

Example
snapshot modify -volume C -snapshot exported_snap -snapmirror-label text

Managing storage efficiency for SnapVault secondaries

SnapVault relationships preserve storage efficiency when backing up data from the primary volume to the secondary volume, with one exception: if post-process and optionally inline compression are enabled on the secondary volume, storage efficiency is not preserved for data transfers between the primary and secondary volumes.

Guidelines for managing storage efficiency for SnapVault backups

If both the primary and secondary volumes in a SnapVault relationship have storage efficiency enabled, then data transfers to the SnapVault secondary volume preserve storage efficiency. If the
primary volume does not have storage efficiency enabled, you might want to enable storage efficiency only on the secondary volume.

Because SnapVault secondary volumes typically contain a large amount of data, storage efficiency on SnapVault secondary volumes can be very important.

If storage efficiency is enabled on the primary volumes

If the primary volume in a SnapVault relationship is enabled for storage efficiency, all data backup operations preserve the storage efficiency.

If storage efficiency is enabled only on the secondary volume

If the primary volume in a SnapVault relationship does not have storage efficiency enabled, you might want to enable storage efficiency for the secondary volume because it is likely to contain a large amount of data over time.

You can use the `volume efficiency` command to start a scan on the volume if there is already data on the volume from transfers. If this is a new relationship with no transfers, then there is no need to run the scan manually.

Changes to the volume’s efficiency schedule do not take effect for a SnapVault secondary volume. Instead, when storage efficiency is enabled, the SnapVault relationship manages the schedule. When a data transfer begins, the storage efficiency process automatically pauses until the transfer is finished, and then automatically begins again after the data transfer is complete. Because data transfers to a SnapVault secondary volume might include more than one Snapshot copy, the storage efficiency process is paused for the entire duration of the update operation. After the transfer is finished and the post-transfer storage efficiency process is complete, the last Snapshot copy created in the secondary volume is replaced by a new, storage-efficient Snapshot copy.

If the last Snapshot copy that is created in the secondary volume is locked before it can be replaced by a new, storage-efficient Snapshot copy, then a new, storage-efficient Snapshot copy is still created, but the locked Snapshot copy is not deleted. That Snapshot copy is deleted later during the storage-efficient cleanup process after a subsequent update to the SnapVault secondary volume and after the lock is released. A Snapshot copy in a SnapVault secondary volume might be locked because the volume is the source in another relationship, such as a data protection mirror relationship.

If the secondary volume has additional compression enabled, storage efficiency is not preserved

Storage efficiency on all data transfers in SnapVault relationships is not preserved when the secondary volume has additional compression enabled. Because of the loss of storage efficiency, a warning message is displayed when you enable compression on a SnapVault secondary volume. After you enable compression on the secondary volume, data transfers are no longer storage efficient. You can reenable storage efficiency on data transfers after you disable compression on the secondary volume.

Related tasks

*Enabling storage efficiency on a SnapVault secondary volume* on page 152

Enabling storage efficiency on a SnapVault secondary volume

If the primary volume does not have storage efficiency enabled, you can enable storage efficiency on a SnapVault secondary volume by enabling storage efficiency on the volume.

Before you begin

You must have cluster administrator privileges to perform this task for a cluster. You must have Vserver administrator privileges to perform this task for an SVM.
About this task

For information about increasing storage efficiency using deduplication and compression, see the *Clustered Data ONTAP Logical Storage Management Guide.*

Steps

1. Use the `volume efficiency` command with the `-on` parameter to enable storage efficiency.

2. If the volume already has data which you want to make storage efficient, use the `volume efficiency` command with the `-start` and `-scan-old-data` parameters to start a scan of the volume.

Related concepts

*Guidelines for managing storage efficiency for SnapVault backups* on page 151

Reenabling storage efficiency on a SnapVault destination volume

To reestablish storage-efficient data transfers after you disable compression for a SnapVault destination volume, you must manually enable storage efficiency on the volume. After you disable data compression, all data transfers continue to be physical transfers (storage efficiency is not preserved) until you manually reenable storage efficiency.

Before you begin

- You must have cluster administrator privileges to perform this task for a cluster.
- You must have Storage Virtual Machine (SVM) administrator privileges to perform this task for an SVM.
- Data compression on the SnapVault destination volume must have been enabled and then subsequently disabled.

Step

1. On the destination cluster of the SnapVault relationship, use the `snapmirror update` command with the `-enable-storage-efficiency` parameter.

   The command enables storage efficiency and begins processing a data transfer. The operation first enters the “preparing” state, during which the system might perform compression and uncompression operations to achieve symmetry between the source and destination volumes. When symmetry is achieved, a storage-efficient Snapshot copy is transferred to the SnapVault destination.

   **Note:** Because the processing time for this transfer might take a longer time than usual, transfer progress is displayed as the percent complete instead of the number of bytes.
Data mirroring using SyncMirror

You can use SyncMirror to mirror aggregates, and thus provide increased data resiliency. SyncMirror removes single points of failure in connecting to disks or array LUNs.

Mirroring data using the SyncMirror feature

The SyncMirror feature is an optional feature of Data ONTAP that gives you the capability for real-time mirroring of data within a single aggregate.

SyncMirror provides for synchronous mirroring of data, implemented at the RAID level. You can use SyncMirror to create aggregates that consist of two copies of the same WAFL file system. The two copies, known as plexes, are simultaneously updated. Therefore, the copies are always identical. The two plexes are within a single aggregate.

The following provides information about the activities of SyncMirror:

- SyncMirror can be used to mirror aggregates.
- SyncMirror cannot be used to mirror FlexVol volumes. However, FlexVol volumes can be mirrored as part of an aggregate.

Related information

Clustered Data ONTAP 8.3 Logical Storage Management Guide
NetApp Documentation: Product Library A-Z

Advantages of using SyncMirror

A SyncMirror aggregate has two plexes. This setup provides a high level of data availability because the two plexes are physically separated.

For a system using disks, the two plexes are on different shelves connected to the system with separate cables and adapters. Each plex has its own collection of spare disks. For a system using array LUNs, the plexes are on separate sets of array LUNs, either on one storage array or on separate storage arrays.

Note: You cannot set up SyncMirror with disks in one plex and array LUNs in the other plex.

Physical separation of the plexes protects against data loss if one of the shelves or the storage array becomes unavailable. The unaffected plex continues to serve data while you fix the cause of the failure. Once fixed, the two plexes can be resynchronized.

Another advantage of mirrored plexes is faster rebuild time.

In contrast, if an aggregate using SnapMirror for replication becomes unavailable, you can use one of the following options to access the data on the SnapMirror destination (secondary):

- The SnapMirror destination cannot automatically take over the file serving functions.
  However, you can manually set the SnapMirror destination to allow read/write access to the data.

- You can restore the data from the SnapMirror destination to the primary (source).

An aggregate that is mirrored using SyncMirror requires twice as much storage as an unmirrored aggregate. Each of the two plexes requires an independent set of disks or array LUNs. For example, you require 2,880 GB of disk space to mirror a 1,440-GB aggregate—1,440 GB for each plex of the mirrored aggregate.
How mirrored aggregates work

Mirrored aggregates have two plexes (copies of their data), which use the SyncMirror functionality to duplicate the data to provide redundancy.

When a mirrored aggregate is created (or when a second plex is added to an existing unmirrored aggregate), Data ONTAP copies the data in the original plex (plex0) to the new plex (plex1). The plexes are physically separated (each plex has its own RAID groups and its own pool), and the plexes are updated simultaneously. This provides added protection against data loss if more disks fail than the RAID level of the aggregate protects against or there is a loss of connectivity, because the unaffected plex continues to serve data while you fix the cause of the failure. After the plex that had a problem is fixed, the two plexes resynchronize and reestablish the mirror relationship.

Note: The time for the two plexes to resynchronize can vary and depends on many variables such as aggregate size, system load, how much data has changed, and so on.

The disks and array LUNs on the system are divided into two pools: pool0 and pool1. Plex0 gets its storage from pool0 and plex1 gets its storage from pool1.

The following diagram shows an aggregate composed of disks with SyncMirror enabled and implemented. A second plex has been created for the aggregate, plex1. The data in plex1 is a copy of the data in plex0, and the RAID groups are also identical. The 32 spare disks are allocated to pool0 or pool1, 16 disks for each pool.

The following diagram shows an aggregate composed of array LUNs with SyncMirror enabled and implemented. A second plex has been created for the aggregate, plex1. Plex1 is a copy of plex0, and the RAID groups are also identical.

The following diagram shows an aggregate composed of array LUNs with SyncMirror enabled and implemented. A second plex has been created for the aggregate, plex1. Plex1 is a copy of plex0, and the RAID groups are also identical.
Requirements for using SyncMirror with disks

If you want to mirror aggregates, you need nodes that support the SyncMirror feature and an appropriate configuration of disk shelves.

The following are requirements for using SyncMirror:

• The nodes should support the SyncMirror feature.
• You must connect disk shelves in a configuration that supports mirrored aggregates.

Related information

*Clustered Data ONTAP 8.3 MetroCluster Management and Disaster Recovery Guide*

*NetApp Hardware Universe*

How SyncMirror works with array LUNs

For array LUN aggregates, SyncMirror creates two physically separated copies of the aggregate, just as it does for disks.

These copies of the aggregate, called *plexes*, are simultaneously updated; therefore, the two copies of the data are always identical. Data continues to be served if one copy becomes unavailable.

For array LUNs, the physical separation of the plexes protects against data loss if the following occurs:

• An array LUN fails.
  For example, a LUN failure can occur because of a double disk failure on the storage array.

• A storage array becomes unavailable.

• In a MetroCluster configuration, an entire site fails.
  The entire site might fail because of a disaster or prolonged power failure. If this situation occurs, the surviving site can take over the functions of the disaster site as a result of a switchover. Data is accessed on the plex of the surviving site.

Each of the two plexes must be on a separate set of array LUNs. In a MetroCluster configuration with Data ONTAP systems that use array LUNs, each plex must be on a separate set of LUNs on different storage arrays. For MetroCluster configurations with Data ONTAP systems that use both array LUNs and disks, the plexes for disks are separate from the plexes for array LUNs.
When SyncMirror is used in a setup other than a MetroCluster configuration, each of the plexes can be on the same storage array or on different storage arrays.

Plexes can be considered local or remote in the context of the storage array that is connected to the Data ONTAP system on which the aggregate is configured. For example, in MetroCluster configurations, the plex at the local site is the local plex while the one at the remote site is the remote plex.

The following illustration shows the relationships of plexes and pools to an aggregate. One plex is associated with pool 0 and one plex is associated with pool 1. The local pool is pool 0 while the remote pool is pool 1. The remote plex is the mirror of the aggregate.

### Requirements for setting up SyncMirror with array LUNs

To set up SyncMirror with array LUNs, you must fulfill standard requirements for any SyncMirror deployment, plus a number of requirements that are unique to setting up SyncMirror with array LUNs. There are a few additional requirements specific to setting up SyncMirror for a MetroCluster configuration with array LUNs.

### Storage type considerations for mirroring

When planning for mirroring of aggregates for systems that can use both array LUNs and disks, consider the following:

- You can mirror data only between the same types of storage. You cannot mirror an aggregate between a native disk shelf on a Data ONTAP system and a storage array.
- If your Data ONTAP system has disk shelves, you can mirror an aggregate with disks between two different disk shelves. The rules for setting up mirroring with disks are the same for FAS systems and V-Series systems.
- While setting up SyncMirror with array LUNs, you must follow the appropriate requirements because they are different from setting up SyncMirror with disks.

### Number and size of array LUNs needed

SyncMirror requires twice as many array LUNs as you ordinarily would need for storage so you can create mirrored aggregates. The same number and size of array LUNs must be available in each set of array LUNs that you are going to use for the two plexes of the aggregate.

For example, assume that you have a 40-GB aggregate that is composed of four 10-GB LUNs, and you want to mirror it. You must have four 10-GB LUNs available in the local location and four 10-GB LUNs in the remote location to be able to mirror the aggregate.

If the LUNs are not the same size, the following occurs:

- If a LUN in the remote location is larger than the LUN in the local location, the mirror is created.
However, space is wasted and cannot be reused. For example, if the array LUN in pool0 is 10 GB and the array LUN in pool0 is 20 GB, the mirror will be 10 GB (the pool0 LUN size.) The remaining 10 GB of space in the pool0 LUN is wasted and cannot be reused.

- If the local LUN is larger than the remote LUN, Data ONTAP does not allow creation of the mirror.
  For example, if the pool0 (local) array LUN is 20 GB and the pool0 array LUN is 10 GB, mirroring fails.

**Number of storage arrays needed for SyncMirror mirroring**

For a MetroCluster configuration that uses array LUNs, you must use two separate storage arrays for the mirror. For SyncMirror that is not used in a MetroCluster configuration with array LUNs, you can use one or two storage arrays.

If you are using two storage arrays for mirroring, the requirements are as follows:

- Both storage arrays must be from the same vendor and from the same model family.
- The firmware version running on both the storage arrays must be the same.
- You must have two sets of LUNs—one set for the aggregate on the local storage array and another set of LUNs at the remote storage array for the mirror of the aggregate (the other plex of the aggregate).

If you are using only one storage array for mirroring, the requirements are as follows:

- The two sets of LUNs must be physically separated on the storage array.
- Each LUN must be from a different disk group (RAID group).

**Disk ownership assignment**

You must assign all array LUNs that are used for the plexes of the aggregate to the same Data ONTAP system. This system *owns* the aggregate.

**Checksum consistency requirement**

All array LUNs in both sets of LUNs that are used for the plexes of the aggregate must be the same checksum type.

**SyncMirror pool assignment**

You want the data mirrored exactly on the two storage arrays so that if one plex becomes unavailable, all data can continue to be served. How you assign array LUNs to SyncMirror pools determines how the array LUNs are distributed between the two storage arrays in the MetroCluster configuration.

For array LUNs, you must explicitly assign each array LUN to the local pool or the remote pool. To group the LUNs correctly, you must plan ahead so that you know which array LUNs are located on which storage array. Data ONTAP cannot determine this for you.

**Specific requirements for a MetroCluster configuration that uses array LUNs**

A MetroCluster configuration can preserve data only if volumes are mirrored. Unmirrored aggregates are not supported.

In addition, you must configure each plex to be on a separate set of array LUNs on different storage arrays.

For more information about a MetroCluster configuration that uses array LUNs, see the *Clustered Data ONTAP 8.3 MetroCluster Installation and Configuration Guide.*
Verification of pathing

Before you create your aggregate and mirror it, you must ensure that there are two paths to an array LUN for a given Data ONTAP system.

For more information about checking paths to array LUNs, see the FlexArray Virtualization Installation Requirements and Reference Guide.

Related concepts

- SyncMirror pool assignment planning for array LUNs on page 159
- Troubleshooting errors with SyncMirror pool assignment for array LUNs on page 161

SyncMirror pool assignment planning for array LUNs

To set up SyncMirror with array LUNs, you must provide Data ONTAP information about which array LUNs are local and which array LUNs are remote.

For native disks, Data ONTAP automatically assigns a disk to the local pool or remote pool, as appropriate, or you can assign a disk to a pool. However, Data ONTAP cannot detect whether an array LUN is located on the local storage array (the local pool) or on the remote storage array (the remote pool). You must explicitly provide this information to Data ONTAP.

You want the data mirrored exactly the same on the two storage arrays so that if one plex becomes unavailable, all data can continue to be served. Your goal is to group the LUNs belonging to the storage arrays into two SyncMirror pools. One is the local pool and the other is the remote pool. Then, when you later create a mirrored aggregate, the LUNs for the same plex are derived from the same pool.

To group the LUNs, you must identify the appropriate SyncMirror pool for each array LUN you are using to create the two plexes of the aggregate. To specify the correct pool for each array LUN, you must know which array LUNs are located on which storage array. Data ONTAP cannot determine this for you.

You must ensure that each LUN group has the same number of LUNs and that the LUNs in each group are the same size.

If you are using one storage array (for example, in a setup other than a MetroCluster configuration), you must ensure that each LUN is from a different disk group (RAID group) on the storage array.

<table>
<thead>
<tr>
<th>Physical location of storage (assuming two storage arrays)</th>
<th>Pool to which the array LUNs need to be assigned</th>
<th>Command setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array LUN is on the storage array that is connected to the Data ONTAP system (the local storage array). The aggregate is created on this Data ONTAP system.</td>
<td>Local pool (pool0)</td>
<td>storage disk assign -pool 0</td>
</tr>
<tr>
<td>Array LUN is on the storage array whose LUNs are to be used to mirror the array LUNs in the aggregate. (This is the remote storage array.)</td>
<td>Remote pool (pool1)</td>
<td>storage disk assign -pool 1</td>
</tr>
</tbody>
</table>

Note: You use the -pool parameter of the storage disk assign command to specify the SyncMirror pool assignment. For more information about the command, see the man pages.
Commands for creating and mirroring an array LUN aggregate

You can either use a single command to create an array LUN aggregate and mirror it or use different commands to create the aggregate first and then mirror it.

**Important:** In a MetroCluster configuration, you must create a mirrored aggregate in a single step.

The commands for creating and mirroring array LUN aggregates are as follows:

<table>
<thead>
<tr>
<th>To...</th>
<th>The command to use is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an aggregate and mirror it in the same step</td>
<td><code>storage aggregate create</code>&lt;br&gt;You can use the <code>-mirror</code> parameter and the <code>-diskcount</code> with this command to distribute the array LUNs into two plexes according to their count.&lt;br&gt;If you want to specify the array LUNs to include in the plexes, you can use the <code>-disklist</code> and the <code>-mirror-disklist</code> parameters.</td>
</tr>
<tr>
<td>Mirror an existing aggregate</td>
<td><code>storage aggregate mirror</code>&lt;br&gt;<em>Note:</em> You cannot create an unmirrored aggregate for a MetroCluster configuration.</td>
</tr>
</tbody>
</table>

For more information about these commands, see the man pages.

Common errors when setting up SyncMirror pools with array LUNs

Your SyncMirror setup for array LUNs will not be successful if your local and remote pool assignments do not match the actual location of the array LUNs.

The following table shows the result of common errors in array LUN SyncMirror pool assignment:

<table>
<thead>
<tr>
<th>Error</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>You assign some array LUNs from the local storage array to the remote pool, or you assign some array LUNs from the remote storage array to the local pool.</td>
<td>You cannot create the mirror for the aggregate. The mirror creation process does not allow mixed pools within a plex.</td>
</tr>
<tr>
<td>You reverse the pool settings for each set of array LUNs. That is, you assign all the LUNs on the local storage array that you want to use for mirroring the aggregate to the remote pool and assign the set of LUNs on the remote storage array to the local pool.</td>
<td>Data ONTAP allows you to create the mirrored aggregate. If one storage array becomes unavailable, the wrong side of the plex is reported as unavailable. The data is still on the storage array that is available.</td>
</tr>
<tr>
<td>You plan to use two storage arrays for SyncMirror but you mistakenly create a mirrored aggregate with both pools from the same storage array.</td>
<td>Data is lost if the storage array fails.</td>
</tr>
</tbody>
</table>

Related concepts

- *Requirements for setting up SyncMirror with array LUNs* on page 157
- *Troubleshooting errors with SyncMirror pool assignment for array LUNs* on page 161
Troubleshooting errors with SyncMirror pool assignment for array LUNs

To troubleshoot SyncMirror pool assignment problems with array LUNs, you need to look at the back-end setup and Data ONTAP configuration. You need to determine whether the pool assignment in Data ONTAP matches the actual location of the LUNs.

If the plexes are on two different storage arrays, you need to know which of the two storage arrays a specific array LUN is located on. You must know how the storage array is cabled to the switch to be able to determine which array LUNs are from the local storage array and which array LUNs are from the remote storage array.

You can use a combination of the following methods to obtain information about where the LUNs are located with reference to the plexes:

- Look at switch zoning
- Look at the output of the following Data ONTAP command: `storage disk show -pool`

You can also correct the array LUN pool assignment errors as required.

Requirements when using mirrored aggregates

If you want to use mirrored aggregates to retain two copies of data, you can either create a new aggregate with two mirrored plexes, or, if you have an aggregate already, add a plex to the existing aggregate (a mirrored aggregate can have only two plexes).

The rules for the selection of disks or array LUNs for use as mirrored aggregates are as follows:

- Disks or array LUNs selected for each plex must be in different pools.
- The same number of disks or array LUNs must be in both plexes.
- Disks are selected first on the basis of equivalent bytes per sector (bps), and then on the basis of the size of the disk.
- If there is no equivalent-sized disk, Data ONTAP uses a larger-capacity disk and limits the size to make it identically sized.
- Data ONTAP names the plexes of the mirrored aggregate.

When creating an aggregate, Data ONTAP selects disks from the pool that has the most available disks. You can override this selection policy by specifying the disks to use.

**Note:** This is true only for clusters that are not MetroCluster clusters.

Related information

*Clustered Data ONTAP 8.3 MetroCluster Management and Disaster Recovery Guide*
Commands for managing mirrored aggregates

There are specific Data ONTAP commands for managing mirrored aggregates.

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>Use this command...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a mirrored aggregate</td>
<td><code>storage aggregate create</code> with the</td>
</tr>
<tr>
<td></td>
<td><code>-mirror</code> parameter</td>
</tr>
<tr>
<td></td>
<td>You cannot use the <code>-disklist</code> or <code>-mirror-</code></td>
</tr>
<tr>
<td></td>
<td><code>disklist</code> parameters when using the <code>-mirror</code></td>
</tr>
<tr>
<td></td>
<td>parameter.</td>
</tr>
<tr>
<td>Convert an aggregate to a mirrored aggregate</td>
<td><code>storage aggregate mirror</code></td>
</tr>
<tr>
<td>Display status about a plex</td>
<td><code>storage aggregate plex show</code></td>
</tr>
<tr>
<td>Add disks to an aggregate</td>
<td><code>storage aggregate add-disks</code></td>
</tr>
<tr>
<td>Show resync status information for each plex</td>
<td><code>storage aggregate show-resync-status</code></td>
</tr>
<tr>
<td>Correct differences between plexes</td>
<td><code>storage aggregate verify</code> with the</td>
</tr>
<tr>
<td></td>
<td><code>-action start</code> and <code>-plex-to-fix</code></td>
</tr>
<tr>
<td></td>
<td>parameters</td>
</tr>
<tr>
<td>Display the status of a plex comparison</td>
<td><code>storage aggregate verify</code> with the</td>
</tr>
<tr>
<td></td>
<td><code>-action status</code> parameter</td>
</tr>
<tr>
<td>Start comparing plexes of a mirrored aggregate</td>
<td><code>storage aggregate verify</code> with the</td>
</tr>
<tr>
<td></td>
<td><code>-action start</code> parameter</td>
</tr>
<tr>
<td></td>
<td>If any discrepancies are found, they appear</td>
</tr>
<tr>
<td></td>
<td>in the EMS log with the following signature:</td>
</tr>
<tr>
<td></td>
<td><code>raid.mirror.verify.mismatch</code>. If no such</td>
</tr>
<tr>
<td></td>
<td>entries exist, no discrepancies were found.</td>
</tr>
<tr>
<td>Stop comparing plexes</td>
<td><code>storage aggregate verify</code> with the</td>
</tr>
<tr>
<td></td>
<td><code>-action stop</code> parameter</td>
</tr>
<tr>
<td>Resume comparing plexes</td>
<td><code>storage aggregate verify</code> with the</td>
</tr>
<tr>
<td></td>
<td><code>-action resume</code> parameter</td>
</tr>
<tr>
<td>Temporarily suspend comparing plexes</td>
<td><code>storage aggregate verify</code> with the</td>
</tr>
<tr>
<td></td>
<td><code>-action suspend</code> parameter</td>
</tr>
<tr>
<td>Bring a plex online</td>
<td><code>storage aggregate plex online</code></td>
</tr>
<tr>
<td>Take a plex offline</td>
<td><code>storage aggregate plex offline</code></td>
</tr>
<tr>
<td>Remove one of the plexes</td>
<td><code>storage aggregate plex delete</code></td>
</tr>
</tbody>
</table>

Creating a mirrored aggregate

You can protect data on a new aggregate by creating it as a mirrored aggregate.

About this task

When you create an aggregate, you can specify for it to use SyncMirror. This ensures that the aggregate is a mirrored one from the start.
### Steps

1. Show all of the available disks by using the `storage disk show` command with the `-fields` parameter and the `disk`, `pool`, and `container-type` fields.

   **Example**

   ```
   cluster1:/> storage disk show -fields disk,pool,container-type
   disk     container-type pool
   -------- -------------- -----  
   1.0.0    aggregate      Pool0
   1.0.1    aggregate      Pool0
   1.0.2    aggregate      Pool0
   1.0.3    aggregate      Pool0
   1.0.4    aggregate      Pool0
   1.0.5    spare          Pool0
   1.0.6    spare          Pool0
   1.0.7    aggregate      Pool0
   1.0.8    spare          Pool0
   1.0.9    aggregate      Pool0
   1.0.10   aggregate      Pool0
   1.0.11   aggregate      Pool0
   1.0.12   aggregate      Pool0
   1.0.13   spare          Pool0
   1.0.14   spare          Pool0
   1.0.15   spare          Pool0
   1.0.16   spare          Pool0
   1.0.17   spare          Pool0
   1.0.18   spare          Pool0
   1.0.19   spare          Pool0
   1.0.20   spare          Pool0
   1.0.21   spare          Pool0
   1.0.22   spare          Pool0
   1.0.23   spare          Pool0
   1.1.0    spare          Pool0
   1.1.1    spare          Pool0
   1.1.2    spare          Pool0
   1.1.3    spare          Pool0
   1.1.4    spare          Pool0
   1.1.5    spare          Pool0
   1.1.6    spare          Pool0
   1.1.7    spare          Pool0
   1.1.8    spare          Pool0
   1.1.9    spare          Pool0
   1.1.10   spare          Pool0
   1.1.11   spare          Pool0
   1.1.12   spare          Pool0
   1.1.13   spare          Pool0
   1.1.14   spare          Pool0
   1.1.15   spare          Pool0
   1.1.16   spare          Pool0
   1.1.17   spare          Pool0
   1.1.18   spare          Pool0
   1.1.19   spare          Pool0
   1.1.20   spare          Pool0
   1.1.21   spare          Pool0
   1.1.22   spare          Pool0
   1.1.23   spare          Pool0
   48 entries were displayed.
   ```

2. Separate disk shelves on different loops configured as pool 0 and pool 1 by using the `storage assign` command.
Example

```bash
cluster1::> storage disk assign -disk 1.1.* -pool 1
```

```bash
cluster1::> storage disk> show -fields disk,pool,container-type
```

```
disk     container-type pool
-------- -------------- -----  
1.0.0    aggregate      Pool0  
1.0.1    aggregate      Pool0  
1.0.2    aggregate      Pool0  
1.0.3    aggregate      Pool0  
1.0.4    aggregate      Pool0  
1.0.5    spare          Pool0  
1.0.6    spare          Pool0  
1.0.7    aggregate      Pool0  
1.0.8    spare          Pool0  
1.0.9    aggregate      Pool0  
1.0.10   aggregate      Pool0  
1.0.11   aggregate      Pool0  
1.0.12   aggregate      Pool0  
1.0.13   spare          Pool0  
1.0.14   spare          Pool0  
1.0.15   spare          Pool0  
1.0.16   spare          Pool0  
1.0.17   spare          Pool0  
1.0.18   spare          Pool0  
1.0.19   spare          Pool0  
1.0.20   spare          Pool0  
1.0.21   spare          Pool0  
1.0.22   spare          Pool0  
1.0.23   spare          Pool0  
1.1.0    spare          Pool1  
1.1.1    spare          Pool1  
1.1.2    spare          Pool1  
1.1.3    spare          Pool1  
1.1.4    spare          Pool1  
1.1.5    spare          Pool1  
1.1.6    spare          Pool1  
1.1.7    spare          Pool1  
1.1.8    spare          Pool1  
1.1.9    spare          Pool1  
1.1.10   spare          Pool1  
1.1.11   spare          Pool1  
1.1.12   spare          Pool1  
1.1.13   spare          Pool1  
1.1.14   spare          Pool1  
1.1.15   spare          Pool1  
1.1.16   spare          Pool1  
1.1.17   spare          Pool1  
1.1.18   spare          Pool1  
1.1.19   spare          Pool1  
1.1.20   spare          Pool1  
1.1.21   spare          Pool1  
1.1.22   spare          Pool1  
1.1.23   spare          Pool1  
```

48 entries were displayed.

3. Create the mirrored aggregate by using the `storage aggregate create` command with the `-mirror` parameter.

Example

```bash
cluster1::> storage aggregate create aggr4 -mirror -diskcount 10
```
4. Confirm that the mirrored aggregate was created by using the `storage aggregate show` command.

**Example**

```
cluster1::> storage aggregate show aggr4

Aggregate: aggr4
  Checksum Style: block
  Number Of Disks: 6
  Mirror: true
  Node: node1
  Disks for First Plex: 1.0.9, 1.0.10, 1.0.11,
  1.0.12, 1.0.13
  Disks for Mirrored Plex: 1.1.12, 1.1.13, 1.1.14,
  1.1.15, 1.1.16
  Partitions for First Plex: -
  partitions for Mirrored Plex: -
  Free Space Reallocation: on
  HA Policy: sfo
  Ignore Inconsistent: off
  Space Reserved for Snapshot Copies: 5%
  Aggregate Nearly Full Threshold Percent: 95%
  Aggregate Full Threshold Percent: 98%
  Block Checksum Protection: on
  RAID Lost Write: on
  Zoned Checksum Protection: -
  Enable Thorough Scrub: off
  Hybrid Enabled: false
  Available Size: 696.0GB
  Checksum Enabled: true
  Checksum Status: active
  Cluster: cluster1
  Home Cluster ID: 74515f83-f398-11e2-bca8-123456789123
  DR Home ID: -
  DR Home Name: -
  Has Mroot Volume: false
  Has Partner Node Mroot Volume: false
  Home ID: 2014941400
  Home Name: node1
  Total Hybrid Cache Size: 0B
  Hybrid: false
  Inconsistent: false
  Is Aggregate Home: true
  Max RAID Size: 16
  Flash Pool SSD Tier Maximum RAID Group Size: -
  Owner ID: 2014941400
  Owner Name: node1
  Used Percentage: 0%
  Plexes: /aggr4/plex0,
  /aggr4/plex1
  RAID Groups: /aggr4/plex0/rg0 (block)
  /aggr4/plex1/rg0 (block)
  RAID Lost Write State: on
  RAID Status: raid_dp, mirrored, normal
  RAID Type: raid_dp
  SyncMirror Resync Snapshot Frequency in Minutes: 60
  Is Root: false
  Space Used by Metadata for Volume Efficiency: 0B
  Size: 698.0GB
  State: online
  Aggregate Type: aggr
  Maximum Write Alloc Blocks: 0
  Used Size: 2.02GB
  Uses Shared Disks: false
  UUID String: f9c49c6f-1821-4570-9d3f-b0178b180407
```

Number Of Volumes: 2
Is Flash Pool Caching: -
Is Eligible for the Balancer: false
State of the Aggregate Being Balanced: ineligible
Converting an aggregate to a mirrored aggregate

You can convert an existing aggregate to a mirrored aggregate to protect the data on the aggregate. You convert the aggregate to a mirrored aggregate by adding a plex to the aggregate.

Before you begin

The aggregate you are converting must have only one plex. This is because you add a plex to create a mirrored aggregate, and mirrored aggregates cannot have more than two plexes.

About this task

You can convert an aggregate to a mirrored aggregate in two ways:

- Manually determine which disks or array LUNs are available, and specify which ones to use. This method is best suited for when you know the disks or array LUNs to be used for the plex addition and want to specify those disks or array LUNs manually.

- Allow Data ONTAP to automatically use the disks or array LUNs that are available. This is the easiest method of adding a plex to an aggregate. If the aggregate that you want to mirror uses disks or array LUNs of different capacities, Data ONTAP can select disks or array LUNs that match the smallest capacity from a different pool. If there are not enough disks or array LUNs of that capacity in the pool, Data ONTAP selects higher-capacity disks or array LUNs and downsizes them.

Step

1. Add a plex to the aggregate:

<table>
<thead>
<tr>
<th>If you are using the...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual method</td>
<td>a. Use the <code>storage disk show</code> command to list the disks from which to choose.</td>
</tr>
<tr>
<td></td>
<td>b. Choose the correct number and size of disks or array LUNs in the list.</td>
</tr>
<tr>
<td></td>
<td>The disks or array LUNs should be from a different pool than the pool that is already being used by the aggregate.</td>
</tr>
<tr>
<td></td>
<td>c. Use the <code>storage aggregate mirror</code> command with the <code>-mirror-disklist</code> parameter.</td>
</tr>
<tr>
<td></td>
<td>This step adds a plex to the aggregate, making it a mirrored aggregate.</td>
</tr>
</tbody>
</table>

| Automatic method       | Use the `storage aggregate mirror` command. |

Example

The following command manually adds a plex to the aggregate `aggrD` with disks 7.1, 7.2, 7.3, 7.4, and 7.5 that you selected, which makes `aggrD` a mirrored aggregate:

```
storage aggregate mirror -aggregate aggrD -mirror-disklist 7.1, 7.2, 7.3, 7.4, 7.5
```

Example

The following command automatically adds a plex to the aggregate `aggrE` using disks that Data ONTAP selected, which makes `aggrE` a mirrored aggregate:

```
storage aggregate mirror -aggregate aggrE
```
Converting a mirrored aggregate to an aggregate

You convert a mirrored aggregate to an (unmirrored) aggregate by removing a plex. You might do this if you want to stop mirroring the aggregate, or if there is a problem with the plex.

Steps

1. Take the selected plex offline by using the `storage aggregate plex offline` command.
2. Destroy the plex you took offline by using the `storage aggregate plex delete` command.

Result

After destroying the plex, Data ONTAP converts the disks or array LUNs used by the plex into hot spares.

Re-creating a mirrored aggregate after a plex failure

In case of a failure that causes a plex to fail, you can remove the plex from the mirrored aggregate, fix the problem, and then re-create it. You can also re-create the mirrored aggregate using a different set of disks or array LUNs, if the problem cannot be fixed.

Steps

1. Destroy the failed plex by using the `storage aggregate plex delete` command.

   Example

   The following command destroys plex0 from the mirrored aggregate:

   ```
   cluster1::> storage aggregate plex delete -aggregate aggr1 -plex plex0
   ```

2. Convert the aggregate to a mirrored aggregate by using the `storage aggregate mirror` command.

   Example

   The following command selects the disks and adds a plex to create mirrored aggregate aggr1:

   ```
   cluster1::> storage aggregate mirror -aggregate aggr1
   ```

Related tasks

- Converting an aggregate to a mirrored aggregate on page 166

Related information

- Clustered Data ONTAP 8.3.2 man page: storage disk show - Display a list of disk drives and array LUNs
- Clustered Data ONTAP 8.3.2 man page: storage aggregate mirror - Mirror an existing aggregate
- Clustered Data ONTAP 8.3 Physical Storage Management Guide
How disks are assigned to plexes

You need to understand how Data ONTAP assigns disks to plexes to configure your disk shelves and host adapters.

When a mirrored aggregate is created, Data ONTAP uses spare disks from two disk pools: pool0 and pool1.

When assigning a disk to a pool, Data ONTAP determines the shelf for the disk and ensures that the disks in pool0 are from different shelves than the disks in pool1. Disk pools must be physically separate to ensure high availability of the mirrored aggregate.

Disks from pool0 are used to create plex0, while disks from pool1 are used to create plex1.

Plexes local to the host node in an HA pair must be connected to the disk pool named pool0. pool0 consists of the storage attached to host adapters in slots 3 through 7.

Note: Pool rules for MetroCluster configurations that use switches are different.

Related information

*NetApp Hardware Universe*

The states of a plex

A plex can either be in an online state or in an offline state. In the online state, the plex is available for read or write access and the contents of the plex are current. In an offline state, the plex is not accessible for read or write.

An online plex can be in the following states.

- Active—The plex is available for use.
- Adding disks or array LUNs—Data ONTAP is adding disks or array LUNs to the RAID group or groups of the plex.
- Empty—The plex is part of an aggregate that is being created and Data ONTAP needs to zero out one or more of the disks or array LUNs targeted to the aggregate before adding the disks to the plex.
- Failed—One or more of the RAID groups in the plex failed.
- Inactive—The plex is not available for use.
- Normal—All RAID groups in the plex are functional.
- Out-of-date—The plex contents are out of date and the other plex of the aggregate has failed.
- Resyncing—The plex contents are being resynchronized with the contents of the other plex of the aggregate.

Addition of disks or array LUNs to a mirrored aggregate

You can add disks or array LUNs to a mirrored aggregate by using one of the following methods.

- Allow Data ONTAP to select the disks or array LUNs.
- Select the disks or array LUNs manually.
• Preview the disks or array LUNs Data ONTAP has selected. You can use the same selection or modify the selection.

Rules for adding disks to a mirrored aggregate

You need to follow certain rules regarding the distribution and size of disks when adding disks to a mirrored aggregate.

• The number of disks must be even, and the disks must be equally divided between the two plexes.
• The disks for each plex must come from different disk pools.
• The disks that you add must have equivalent bytes per sector (bps) sizes.

When adding new disks to a RAID group, the utilization of the new disks depends on the RAID level used. If the storage capacity of the new disks is more than the disks already in the RAID group, the larger-capacity disks might be downsized to suit the RAID group.

• RAID-DP: Larger-capacity disks are downsized to size of parity disks.
• RAID-4: Larger-capacity disks can replace the parity disks.

Rules for adding array LUNs to a mirrored aggregate

When you add array LUNs to a mirrored aggregate, you need to ensure that the number and size of the array LUNs in the two plexes remain identical.

You must consider the following when adding array LUNs to a mirrored aggregate:

• You must add an even number of array LUNs to the mirrored aggregate.
• You must distribute the array LUNs equally between the two plexes.
• The array LUNs for each plex must belong to a different LUN group.
  Do not mix LUNs from the two LUN groups in the same plex.
• All array LUNs in the mirrored aggregate must be the same checksum type.

Related concepts

- Requirements for setting up SyncMirror with array LUNs on page 157
- Common errors when setting up SyncMirror pools with array LUNs on page 160

Increasing the size of an aggregate that uses unpartitioned drives

You can add disks or array LUNs to an aggregate so that it can provide more storage to its associated volumes.

Before you begin

• You must understand the requirement to add disks or array LUNs owned by the same system and pool
• For aggregates composed of disks, you must understand the following:
  ◦ Benefits of keeping your RAID groups homogeneous for disk size and speed
  ◦ Which types of disks can be used together
  ◦ Checksum rules when disks of more than one checksum type are in use
  ◦ How to ensure that the correct disks are added to the aggregate (the disk addition operation cannot be undone)
How to add disks to aggregates from heterogeneous storage

Minimum number of disks to add for best performance

Number of hot spares you need to provide for protection against disk failures

Requirement to add storage to both plexes of a mirrored aggregate at the same time to ensure that the plexes are the same size and contain the same disk types

If you are adding cache to a Flash Pool aggregate, the cache limit for your system model and how much cache you are adding towards the limit

About this task

This procedure should not be used for aggregates composed of root or data partitions.

Following these best practices when you add storage to an aggregate optimizes aggregate performance:

• Add a complete RAID group at one time.
  The new RAID group does not have to be exactly the same size as the existing RAID groups, but it should not be less than one half the size of the existing RAID groups.

• If any small RAID groups exist already, you can bring them up to the size of the other RAID groups, as long as you add at least as many data drives as are already in the RAID group.

• Avoid adding a small number of drives to an existing RAID group.
  Doing so results in the added disks being the target for a disproportionate percentage of new data, causing the new disks to become a performance bottleneck.

Steps

1. Verify that appropriate spare disks or array LUNs are available for you to add:

   storage aggregate show-spare-disks -original-owner node_name

   For disks, make sure that enough of the spares listed are of the correct type, size, speed, and checksum type for the target RAID group in the aggregate to which you are adding the disks.

2. Add the disks or array LUNs:

   storage aggregate add-disks -aggregate aggr_name [-raidgroup new] disks

   If you are adding disks with a different checksum than the aggregate, as when creating a Flash Pool aggregate, or if you are adding disks to a mixed checksum aggregate, you must use the -checksumstyle parameter.

   If you are adding disks to a Flash Pool aggregate, you must use the -disktype parameter to specify the disk type.

   You can use the -disksize parameter to specify a size of the disks to add. Only disks with approximately the specified size are selected for addition to the aggregate.

   If you specify the -raidgroup parameter with a value of new, the storage is added to a new RAID group; this is generally the best way to add storage to an aggregate. You can also specify an existing RAID group with the -raidgroup parameter. If you are adding SSDs to the SSD cache of a Flash Pool aggregate, you do not need to specify the RAID group name; the SSD RAID group is selected by default based on the type of the disks you are adding.

   disks specifies the disks to be added in one of the following ways:

   • -diskcount, usually further qualified by disk type, disk size, or checksum type
   • -disklist disk1 [disk2...]
If possible, you should use the `diskcount` option rather than selecting specific disks. Doing so allows Data ONTAP to optimize the disk selection for your configuration.

If you are adding disks to a mirrored aggregate and you are specifying disk names, you must also use the `-mirror-disklist` parameter.
Copyright information

Copyright © 1994–2016 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).
Trademark information

NetApp, the NetApp logo, 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, Snap Creator, SnapCenter, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, StorageGRID, Tech OnTap, Unbound Cloud, and WAFL and other names are trademarks or registered trademarks of NetApp, Inc., in the United States, and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web at http://www.netapp.com/us/legal/netapptmlist.aspx.
How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email to doccomments@netapp.com. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277
Index

A
about
   - snapshot copy reserve 21
active file disk space
   - how Data ONTAP uses deleted 21
active file systems
   - access on destination volumes 25
   - guidelines for restoring from SnapVault backups 143
   - when available on destination volumes 25
advantages
   - of using SyncMirror 154
aggregates
   - adding unpartitioned drives or array LUNs to 169
   - advantages of using SyncMirror 154
   - commands for managing mirrored 162
   - converting mirrored aggregates to 167
   - converting unmirrored to mirrored 166
   - creating mirrored 162
   - increasing the size of with unpartitioned drives 169
   - mirrored, explained 155
   - mirrored, rules for adding array LUNs 169
   - re-creating mirrored, after a plex failure 167
aggregates, mirrored
   - requirements when using 161
applications
   - testing database 113
array LUNs
   - adding to aggregates 169
authentication
   - cluster peer relationships 54
automatic Snapshot copies
   - using prefixes to name 42
automatically deleting
   - Snapshot copies 18

B
backup and restore sessions, SMTape
   - scalability limits for 102
backup policies
   - example of creating tiered 50
backup tasks
   - Snapshot copy 15
backups
   - guidelines for managing storage efficiency for SnapVault 151
   - guidelines for restoring active file system from SnapVault 143
   - how restore operations work from SnapVault 144
   - introduction to creating SnapVault, on FlexVol volumes 128
   - managing copied source volume, in SnapVault relationships 143
   - SnapVault, how they work 30
   - source-to-destination-to-tape, defined 34
backups, SnapVault
   - requirements for using data protection reference
guide to understand, plan, and implement 8
baseline transfers
   - defined 28
benefits
   - of using SyncMirror 154

C
cascade configurations
   - networking requirements for cluster peering 59
cascades
   - considerations when breaking SnapMirror relationships in 105
   - creating SnapVault backup in mirror-SnapVault 133
   - managing SnapVault-mirror, when SnapVault backup is unavailable 149
   - SnapMirror-SnapVault, how they work 35
   - SnapVault-SnapMirror, how they work 36
CIFS clients
   - destination volume not accessible 27
CIFS users
   - accessing Snapshot copies 14
   - cleaning up
     - a failed single file or LUN restore 148
   - client access
     - to active file systems on destination volumes 25
   - cluster administrators
     - mirror and SnapVault relationship management capabilities 48
cluster peer relationships
   - authentication of 54
   - commands for managing 52
   - requirements for 59
cluster peers
   - creating relationships between 71
   - definition of intercluster networking 55
   - deleting the relationship 74
   - supported network topologies 55
clusters
   - how SnapMirror policies work with 47
   - naming requirements for cluster peering 59
   - requirements for creating SnapMirror or SnapVault relationships between 25
commands
   - for managing cluster peer relationships 52
   - for managing mirror and SnapVault policies 47
   - for managing mirrored aggregates 162
   - for managing SnapMirror relationships 92
   - for managing Snapshot copy policies and schedules 40
   - for monitoring Snapshot copy disk consumption 45
   - for scheduling when SnapMirror transfers occur 102
   - job show 102
   - managing SVM disaster recovery relationships 126
   - snapmirror break 113
   - snapmirror break, for testing database applications 113
   - snapmirror delete 113
   - snapmirror delete, when deleting mirror relationships 104
volume delete, when deleting mirror relationships 104
volume snapshot autodelete show 19
volume snapshot partial-restore-file 44
volume snapshot partial-restore-file-list-info 44
volume snapshot restore 44
volume snapshot restore-file 43
volume snapshot restore-file-info 43
comments
how to send feedback about documentation 174
configuration details
deciding to replicate for disaster recovery 116
configurations
cascade, for cluster peering 59
deployment, how SnapMirror-SnapVault cascades work 35
fan-out, for cluster peering 59
source-to-destination-to-tape, defined 34
supported data protection deployment 32
connectivity
comparison of designated and undesignated intercluster 58
supported cluster peer network topologies 55
conversion
SnapVault relationships to version-flexible SnapMirror relationships 137
SnapVault to version-flexible SnapMirror 91
copied source volumes
managing SnapVault backup of 143
copies, Snapshot
requirements for using data protection reference guide to understand, plan, and implement 8
creating
intercluster SVM peer relationships 77
intracluster SVM peer relationships 78
SVM peer relationships 76
cross cluster relationships
deleting 74

D
data compression
how SnapVault backups work with 30
data loss
tools to protect against 11
data loss disaster 10
data ports
configuring intercluster LIFs to share 61
considerations when sharing intercluster and 60
data protection
creating mirrored aggregates for 162
in SAN environments 12
levels of provided by mirror relationships 23
overview of methods for 9
requirements for using reference guides to understand, plan, and implement 8
data protection mirror copies
updating manually 104
data protection mirror relationships
SVMs 23
data protection mirror transfers
unexpected destination volume growth during 24
data protection mirrors
active file systems for 25
creating 85
FlexVol volumes, support of 23
making a destination volume writable 113
reestablishing the relationship after disaster 106
reversing the relationship after disaster 106
using to test database applications 113
data protection policies
types of 12
data protection strategy
planning 14
data-protection mirror relationships
limitations for 26
database applications
testing 113
databases
introduction to using NVFAIL to monitor and protect validity of 10
dedicated ports
considerations when using for intercluster replication 60
deleting
Snapshot copies automatically 18
SVM peer relationships 82
SVMs 127
deployment configurations
basic, defined 34
how mirror-mirror cascades work 34
how SnapMirror-SnapVault cascades work 35
how SnapVault-SnapMirror cascades work 36
mirror-SnapVault fanout, defined 37
multiple-mirrors fanout, how they work 37
source-to-destination-to-tape, defined 34
supported data protection 32
destination SVM
data access
providing read-only from the destination SVM 124
providing read-only data access from 124
destination SVMs
deleting 127
how volumes and LIFs are placed on 119
understanding 115
destination volume growth, unexpected
during data protection mirror transfers 24
destination volumes
access to active file systems on 25
CIFS clients cannot access 27
components of a mirror relationship 22
converting to SnapVault secondary 138
reenabling storage efficiency on SnapVault 153
supported number of SnapMirror relationship fanout 26
disaster recovery
deciding to replicate all or some configuration details 116
mirror relationships for SVMs 23
providing for SVMs 115
reestablishing the original data protection mirror relationship 106
reestablishing the original version flexible SnapMirror relationship 109
reversing the data protection mirror relationship 106
information
how to send feedback about improving documentation 174
initialization failures
correcting SnapMirror 86
intercluster connectivity
comparison of designated and undesignated 58
intercluster LIFs
configuring to share data ports 61
configuring to use dedicated intercluster ports 63
configuring to use dedicated ports in their own networks 66
considerations when sharing with data ports 60
intercluster networking
definition of cluster peer 55
intercluster networks
configuring intercluster LIFs for 61, 63
configuring LIFs in 66
considerations when sharing data and intercluster ports 60
firewall requirements for SnapMirror relationships 34
supported topologies 55
intercluster ports
configuring intercluster LIFs to use dedicated 63
considerations when using dedicated 60
IP addresses
requirements for cluster peering 59
IPspaces
requirements for cluster peering 59
using for cluster peer connectivity 66
L
language settings
requirement between SVMs 24
LIFs
configuring to use dedicated intercluster ports 63
configuring to use dedicated ports in their own intercluster networks 66
how they are placed on destination SVMs 119
LIFs, intercluster
configuring to share data ports 61
limitations
data-protection mirror relationship 26
SnapVault backup 31
limits, scalability
SMTape backup and restore session 102
load-sharing mirrors
modifying relationship schedules 103
low bandwidth connection
initializing destination volume 95
LUN restore
how it works 147
LUNs
cleaning up a failed single LUN restore 148
guidelines for restoring in SAN environments 144
protecting data 12
restoring a single LUN 146
LUNs (array)
array LUN aggregates
creating and mirroring 160
commands for 160
creating and mirroring aggregates 160
how it works 156
planning pools 159
requirements 157
rules for adding to mirrored aggregate 169
troubleshooting 160, 161
with SyncMirror 156, 157, 159–161
M
managing
commands for SVM disaster recovery relationships 126
methods
overview of data protection 9
methods, data protection
requirements for using reference guides to understand, plan, and implement 8
MetroCluster configurations
array LUNs
    SyncMirror requirements 157
mirror copies
    manually updating data protection, on volumes 104
mirror policies
    commands for managing 47
description of 12
mirror relationships
    components of 22
differences between cluster administrator and SVM administrator policy management privileges 48
limitations for data-protection 26
monitoring the SVM status 122
providing disaster recovery for SVMs 115
SVMs 23
mirror transfers, data protection
unexpected destination volume growth during 24
mirror-mirror cascades
how they work 34
mirror-SnapVault cascades
creating SnapVault backup for 133
preserving Snapshot copies on primary source volume of 134
mirrored aggregates
commands for managing 162
converting to aggregates 167
converting unmirrored aggregates to 166
creating 162
explained 155
re-creating after a plex failure 167
requirements when using 161
rules for adding array LUNs 169
what you can do with 154
mirroring technology
requirements for using data protection reference guide to understand, plan, and implement 8
mirrors
creating data protection 85
deleting 104
deleting Snapshot copies automatically 27
listing the state of a scheduled transfer 102
making a destination volume writable 113
modifying load-sharing relationship schedules 103
reestablishing the relationship after disaster 106
reestablishing the version flexible SnapMirror relationship after disaster 109
reversing the relationship after disaster 106
reversing the version flexible SnapMirror relationship after disaster 109
scheduling when SnapMirror transfers occur 102
using to test database applications 113
modifying
    SnapMirror relationships
    modifying the attributes 125
    SVM disaster recovery relationship attributes 125
monitoring
    SnapMirror relationship status 122
multiple-mirrors fanout deployments
    how they work 37

N
namespaces
    data protection for SVM 31
naming guidelines
    SnapMirror and SnapVault policy 48
naming guidelines for SnapMirror 48
NAS
    SVM configuration details replicated for disaster recovery 116
network
    full-mesh connectivity described 59
    requirements for cluster peering 59
network compression
    SnapMirror and SnapVault 47
NFS users
    accessing Snapshot copies 14
NVFAIL
    description of 10

O
older-than-base Snapshot copies
    backing up from 139
out-of-order Snapshot copies
    how they work 139

P
pair-wise full-mesh connectivity
    supported cluster peer network topologies 55
passphrases
    cluster peer authentication using 54
path names
    abbreviating 24
    pattern matching of 24
    wildcard use for 24
pattern matching
    path names 24
peer relationships
    authentication of cluster 54
    commands for managing cluster 52
    creating between SVMs 76
    creating cluster 71
    creating intracluster SVM 78
    deleting cluster 74
    deleting SVM 82
    for SVMs, described 75
    intercluster, creating on SVMs 77
    managing SVM 74
    modifying SVM 81
    requirements for clusters 59
    requirements for using data protection reference guide to understand, plan, and implement 8
    states of SVMs in 76
plexes
    how disks are assigned to 168
    mirrored aggregate, explained 155
    re-creating mirrored aggregate after failed 167
    removing from a mirrored aggregate to convert to aggregate 167
policies
    commands for managing mirror and SnapVault 47
    commands for managing Snapshot copy 40
    differences between cluster administrator and SVM administrator management privileges 48
    example of creating tiered backup 50
    flexibility of version-flexible SnapMirror 90
    introduction to managing data protection using SnapMirror 47
    introduction to managing Snapshot 40
    naming guidelines for SnapVault 48
    SnapMirror, how they work with clusters and SVMs 47
    Snapshot copy schedule and retention planning guidelines for SnapVault backups 31
    types of data protection 12
port usage
    intercluster SnapMirror relationship firewall requirements 34
ports
    configuring intercluster LIFs to use dedicated intercluster 63
    considerations when sharing data and intercluster roles on 60
    considerations when using dedicated intercluster 60
    requirements for cluster peering 59
ports, data
    configuring intercluster LIFs to share 61
ports, dedicated
    configuring intercluster LIFs to use in their own networks 66
prefixes
    using to name automatic Snapshot copies 42
primary volumes
    defined 28
protect data
    creating mirrored aggregates to 162
protection deployments
    See deployment configurations
protection policies
    types of data 12
protection, data
requirements for using reference guides to understand, plan, and implement 8

R

read-only access
providing from the destination SVM 124

recovery tasks
Snapshot copy 15
reference guides
requirements for using data protection guide 8
relationships
authentication of cluster peer 54
commands for managing cluster peer 52
commands for managing SnapMirror 92
components of mirror 22
considerations when breaking SnapMirror 105
creating cluster peer 71
creating intercluster SVM peer 77
creating intracluster SVM peer 78
creating SnapVault backup, in a prepopulated FlexVol volume 135
creating SnapVault backup, in empty FlexVol volumes 130
creating SVM peer 76
creating version-flexible SnapMirror 87
deleting cluster peer 74
deleting SVM peer 82
guidelines for creating SnapVault, on FlexVol volumes 128
peering SVMs, described 75
prepopulated SnapVault secondary scenarios 129
requirements for creating SnapMirror or SnapVault, between clusters or SVMs 25
SnapVault, creating baseline from tape 136
transition (TDP) 9
relationships, peer
requirements for using data protection reference guide to understand, plan, and implement 8
relationships, SnapMirror
replicating and retaining 91
releasing
SVM disaster recovery relationship 126
requirements
cluster naming when peering 59
firewall for cluster peering 59
IP addresses for cluster peering 59
IPspaces for cluster peering 59
network for cluster peering 59
ports for cluster peering 59
subnets for cluster peering 59
SVM language setting 24
reserve
eexample of what happens when exceeding Snapshot copy 21
restarting
a failed single file or LUN restore operation 146
restore
cleaning up a failed file or LUN 148
how single file or LUN restore works 147
restore operations
guidelines for LUNs in SAN environments 144
guidelines for restoring active file system from SnapVault backups 143
how they work from a SnapVault backup 144
restoring volumes from SnapVault backups 145
restoring Snapshot copies
Shadow Copy Client tools 45
restricted volume errors
correcting 86
root information
data protection for SVM 31

S

SAN
SVM configuration details replicated for disaster recovery 116
SAN (storage area network)
data protection of volumes containing LUNs 12
SAN environments
guidelines for restoring LUNs in 144
which LUN data is backed up to SnapVault backups 29
which LUN data is not backed up to SnapVault backups 29
scalability limits
SMTape backup and restore session 102
schedules
commands for managing Snapshot copy 40
creating a Snapshot copy 17
for default Snapshot copies 16
frequency consideration for version-flexible SnapMirror relationships 90
guidelines for planning Snapshot copy 31
strategies for creating Snapshot copy policies 41
secondary volumes
defined 28
enabling storage efficiency on SnapVault 152
sessions, SMTape backup and restore scalability limits for 102
single file or LUN restore
types of files not restored 148
single file restore
cleaning up a failed 148
how it works 147
restarting a failed 146
single files
restoring 146
single LUN restore
cleaning up a failed 148
how it works 147
restarting a failed 146
SMTape
performing tape seeding 95
SMTape
what tape seeding is 95
SMTape backup and restore sessions
calability limits for 102
SnapMirror
considerations when breaking relationships 105
considerations when using version-flexible 90
correcting an initialization failure 86
creating a data protection 85
creating version-flexible relationships 87
deleting a mirror relationship 104
firewall requirements for intercluster relationships 34
listing the state of a scheduled transfer 102
modifying relationship schedules 103
network compression for 47
reestablishing the version flexible relationship after disaster 109
reversing the version flexible relationship after disaster 109
scheduling when transfers occur 102
source and destination port requirements on intercluster relationships 34
using extended queries 95
SnapMirror commands for managing mirror and SnapVault policies 47
SnapMirror labels defined 28
SnapMirror policies how they work with clusters and SVMs 47
introduction to managing data protection using 47
requirements for using data protection reference guide to understand, plan, and implement 8
retention flexibility of version-flexible SnapMirror relationships 90
SnapMirror policy altering to preserve Snapshot copies after reaching retention limit 49
SnapMirror relationships commands for managing 92
considerations when breaking 105
converting SnapVault relationships to version-flexible 137
converting to version-flexible SnapMirror 91
managing for SVM disaster recovery 122
monitoring the status of 122
releasing SVM 126
requirements for creating, between clusters or SVMs 25
supported number of fanout volumes 26
SVMs 23
SnapMirror-SnapVault cascades how they work 35
Snapshot copies backing up FlexVol volumes with over 251 142
backing up from older than base 139
backup and recovery tasks you can perform with 15
commands for managing 39
commands for managing policies and schedules 40
creating a schedule 17
default schedule 16
defined 28
defining 14
deleting automatically 18
example of what happens when the reserve is exceeded 21
how out-of-order transfers work 139
limit on source volume 27
maximum number 15
monitoring disk consumption of 45
preserving after reaching retention limit 49
preserving on primary source volume in mirror-SnapVault cascades 134
protecting data, about 39
requirements for using data protection reference guide to understand, plan, and implement 8
restoring a single file from 43
restoring part of a file from 44
restoring volume contents 44
retention flexibility using version-flexible SnapMirror relationships 90
types of user-specified schedules 16
user access to 14
using prefixes to name automatic 42
using version-flexible SnapMirror for better resource utilization 91
viewing automatic deletion settings for 19
where to find information about Infinite Volumes and 16
SnapVault backing up FlexVol volumes with over 255 Snapshot copies 142
backing up from older-than-base Snapshot copy 139
converting data protection destination to secondary 138
creating a baseline from tape 136
creating backup, in a prepopulated FlexVol volume 135
differences between cluster administrator and SVM administrator policy management privileges 48
network compression for 47
which data is backed up and restored from FlexVol volume 29
which data is not backed up to 29
SnapVault backups creating in empty FlexVol volumes 130
creating in mirror-SnapVault cascades 133
data protection for SVM namespace and root information 31
guidelines for managing storage efficiency for 151
guidelines for planning Snapshot copy schedule and retention for 31
guidelines for restoring active file system from 143
guidelines for restoring LUNs in SAN environments 144
how restore operations work from 144
how they work 30
how they work with data compression 30
introduction to creating on FlexVol volumes 128
introduction to protecting FlexVol volume date using 128
limitations for FlexVol volume backup 31
managing SnapVault-mirror cascade when unavailable 149
preserving Snapshot copies on primary source volume in mirror-SnapVault cascades 134
requirements for using data protection reference guide to understand, plan, and implement 8
example of creating 50

tools
  for data-loss protection 11
topologies
  supported cluster peer network 55
transfers
  scheduling SnapMirror 102
transfers, data protection mirror
  unexpected destination volume growth during 24
Twitter
  how to receive automatic notification of
documentation changes 174

U
understanding
  destination SVMs 115
unexpected destination volume growth
during data protection mirror transfers 24
unmirrored aggregates
  converting to mirrored aggregates 166
  See also aggregates
updating
  SVM disaster recovery relationship 124

V
volume growth, unexpected destination
  during data protection mirror transfers 24
volume is restricted errors
  correcting 86
volume snapshot commands
  for managing Snapshot copies 39
volume snapshot policy commands
  for managing Snapshot copy policies and schedules 40
volumes
  commands for managing SnapMirror relationships of
  FlexVol and Infinite 92
  components of a mirror relationship 22
  converting data protection destination to SnapVault secondary 138
  creating SnapVault backup in prepopulated FlexVol 135
  creating SnapVault backups in empty FlexVol 130
  creating version-flexible SnapMirror relationships to
  FlexVol destination 87
  enabling storage efficiency on SnapVault secondary 152
  firewall requirements for intercluster SnapMirror
  relationships 34
  guidelines for creating SnapVault relationships on
  FlexVol 128
  how Snapshot policies are associated with 40
  how they are placed on destination SVMs 119
  introduction to creating SnapVault backups on
  FlexVol 128
  manually updating data protection mirror copies on
  104
  maximum number of Snapshot copies for 15
  reenabling storage efficiency on SnapVault
  destination 153
  requirements for creating SnapMirror or SnapVault
  relationships between, in different clusters or SVMs 25
  restoring from SnapVault backup 145
  SnapVault backup limitations for FlexVol 31
  which data gets backed up and restored from FlexVol
  29
Vservers
  See SVMs

W
wildcards
  path names using 24
workflow
  SVM disaster recovery 120
  SVM DR 120