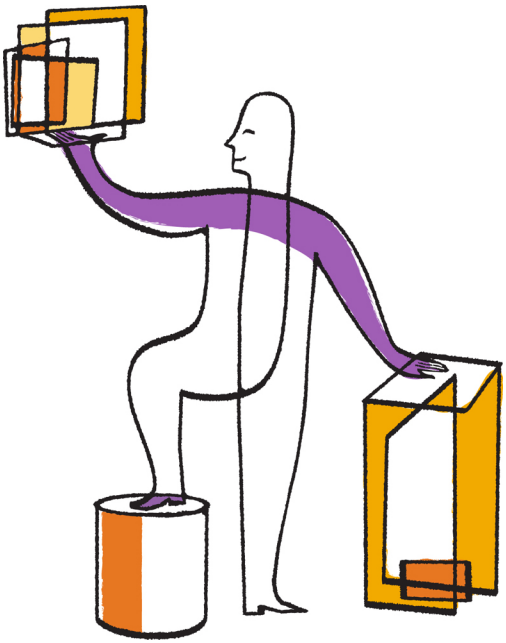
Updated for 8.2.2

Data ONTAP® 8.2

Data Protection Online Backup and Recovery Guide

For 7-Mode



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
Web: www.netapp.com
Feedback: doccomments@netapp.com

Part number: 215-09185_B0
April 2015

Contents

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 provides a variety of features and methods for protection from accidental, malicious, or disaster-induced data loss.

Data protection feature	Description
`aggr copy`	Fast block copy of data stored in aggregates This feature enables you to quickly copy blocks of stored system data from one aggregate to another.
Snapshot copy	Allows 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 enabled, that data can be easily and quickly restored from one of the last several Snapshot copies created. You can also create clones of FlexVol volumes and Data ONTAP LUNs using Snapshot copies. For more details, see the *Data ONTAP Storage Management Guide for 7-Mode*.
SnapRestore (license required)	Fast, space-efficient restoration of large volumes of data backed up to Snapshot copies The SnapRestore feature performs on-request Snapshot recovery from Snapshot copies on an entire volume.

Data protection feature	Description
SnapMirror (license required)	Volume-to-volume and qtree-to-qtree replication This feature enables you to periodically make Snapshot copies of data on one volume or qtree, replicate that data to a partner volume or qtree, usually on another storage system, and archive one or more iterations of that data as Snapshot copies. Replication on the partner volume or qtree ensures quick availability and restoration of data, from the point of the last Snapshot copy, should the storage system containing the original volume or qtree be disabled. If you conduct tape backup and archival operations, you can carry them out on the data already backed to the SnapMirror partner.
SnapVault (license required)	Centralized backup of multiple qtrees on multiple storage systems by using Snapshot technology This feature enables you to back up qtrees on multiple volumes and storage systems to a single SnapVault secondary storage system specialized for quick backup and restore of its sources. You can also install the Open Systems SnapVault agent on Windows, Solaris, Linux, AIX, or HP-UX systems. This agent enables SnapVault to back up and restore data to these systems also. If you conduct tape backup and archival operations, you can carry them out on the data already backed up to the SnapVault secondary storage system.
Tape backup `dump` and `restore` commands	Allow you to back up Snapshot copies to tape The `dump` command takes a Snapshot copy of the volume and then copies that data to tape. Because the Snapshot copy, not the active file system, is backed up to tape, Data ONTAP can continue its normal functions while the tape backup takes place. For more information, see the *Data ONTAP Data Protection Tape Backup and Recovery Guide for 7-Mode*.
`vol copy`	Fast block copy of data from one volume to another The `vol copy` command enables you to quickly block copy stored data from one volume to another.

Data protection feature	Description
SyncMirror (HA pair required)	Continuous mirroring of data to two separate aggregates This feature allows for real-time mirroring of data to matching aggregates physically connected to the same storage system. In case of an unrecoverable disk error on one volume, the storage system automatically switches access to the mirrored volume. This feature requires an HA pair.
`nvfail` option to the `vol options` command	Protection against data corruption by failures of nonvolatile RAM (NVRAM)
Virus scan support	Support for third-party virus-scanning software for files accessed by Common Internet File System (CIFS) clients
MetroCluster	Stretch MetroCluster provides site protection and supports replication up to 500 m. Fabric MetroCluster provides site protection and supports replication up to 100 km by using Fibre Channel (FC) switches. The SyncMirror functionality is enhanced to provide continuous volume mirroring. **Note:** Systems using synchronous SnapMirror might experience performance degradation. Hence, you should use MetroCluster for synchronous replication instead of synchronous SnapMirror.

Related concepts

Data replication using aggr copy on page 26
Data recovery using SnapRestore on page 66
Data protection using SnapVault on page 229
Data replication using volume copy on page 300
Data protection using SnapMirror on page 76
Virus protection for CIFS on page 351

Related references

Data mirroring using SyncMirror on page 315

What online backup and recovery is

Data ONTAP creates online data backups to enable online data recovery. Online backup data is stored on disks, or on array LUNs in the case of third-party storage, rather than on tape. Data stored on disk is available for quick restoring in the event that disaster recovery operations are necessary.

Online backup and recovery solutions include: Snapshot, SnapMirror, SnapRestore, SnapVault, SyncMirror, MetroCluster, the `vol copy` command, and the `ndmpcopy` command.

- The Snapshot feature enables you to schedule weekly, daily, or hourly online backups. Snapshot technology makes online point-in-time copies in the same volume as the original data. It enables users to recover their own deleted or modified files without assistance from a system administrator.
- The SnapMirror feature allows you to schedule regular automatic copies of file system Snapshot copies of a volume or qtree onto another volume or qtree (on the same or a different storage system).
- The SnapRestore feature restores an entire volume to the state recorded in a previously created Snapshot copy with maximum speed and disk space efficiency. The SnapVault feature protects the data in one or more qtrees in a series of Snapshot copies stored on a separate storage system.
- SnapVault maintains an online, asynchronous, permanently read-only replica of the qtree data. SnapVault backup and Snapshot copy creation runs on an automated schedule.

  Note: SnapVault, in addition to providing storage system backup, also provides direct backup to servers running Windows NT, Windows 2000, Solaris, or HP-UX.

- SyncMirror provides continuous real-time mirroring of data between two partner volumes on a shared or partner storage system.
- The MetroCluster feature provides SyncMirror continuous mirroring over extended distances (500 meters to 100 kilometers).
- The `vol copy` command uses a Snapshot copy and copies a volume, manually or by means of a script.
- The `ndmpcopy` command copies any subtree to any location on any storage system. For more information, see the *Data ONTAP Data Protection Tape Backup and Recovery Guide for 7-Mode*.

You can use these online data backup and recovery systems to supplement tape backup and recovery.

Related concepts

Data protection using SnapMirror on page 76

Advantages of online backup and recovery

Online backup and recovery protection offers better speed and ease of use, compared to tape archives.

The main advantages of online backup and recovery are as follows:

- Speedy backups and restores greatly reduce backup time requirements.
- Backups can be made more frequently because they are faster.
- It is easy to recover a particular file, directory, or volume from an online backup.
- Disaster recovery is quicker with online mirroring and restores.
- Data availability is higher because of the high speed of data recovery.
- More data can be backed up in less time.

Disadvantages of online backup and recovery

Online data protection has some disadvantages over tape archives.

The important disadvantages of online backup and recovery compared to tape archives are the following:

- Online data protection is often more expensive to procure.
- Online data protection consumes resources, such as disk space, that can otherwise be used for everyday activities.

Features that enable online backup and recovery of volumes

You can perform online backup and recovery of volumes using any of several Data ONTAP features.

Data ONTAP provides the following features for traditional and FlexVol volumes to help you back up and recover data:

- Snapshot copies
  Makes a read-only image of a file system volume on the same disk.
- SnapRestore
  Restores data to a corrupted volume from a previous Snapshot copy.
- SnapMirror
  Maintains a replica of one volume in another volume and one qtree in another qtree.
- SnapVault
  Keeps copies of volumes on the server, from which individual qtrees are available to the client at any time, for long periods of time.

- vol copy
  Copies data from one volume to another.
- SyncMirror
  Maintains two identical copies of a volume at all times.

Related concepts

Snapshot management on page 28
Data recovery using SnapRestore on page 66
Data protection using SnapVault on page 229
Data replication using volume copy on page 300
Data protection using SnapMirror on page 76

Related references

Data mirroring using SyncMirror on page 315

What the NVFAIL feature is

If NVRAM problems occur that compromise database validity, the NVFAIL feature can warn you and automatically rename the database so that it does not restart automatically. You can then ensure that the database is valid before restarting it.

Data ONTAP provides database protection using the `nvfail` option of the `vol options` command.

Note: You can use this feature only when there are databases on the storage system.

Related concepts

Database protection using NVFAIL on page 346

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.

How to determine a disaster

It is critical that you follow some predefined procedure to confirm whether a disaster really has occurred.

You can use any of the following procedures to determine the status of the supposed disaster site:

- Use the following external interfaces:
  - Ping
  - Remote shell
  - Protection Manager
- Use network management tools to test connectivity to the site.
- Physically inspect the site, if possible.

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

SnapVault is an inter-site Snapshot copy backup and restorability tool on FlexVol volumes. You can locate a SnapVault destination off-site, at any distance from the source system.

Data recoverability

If a data-loss disaster occurs at a source storage site, you can restore data that is backed up to a SnapVault destination storage site. You can restore the data 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 last Snapshot copy that was replicated to the destination system.

Connection requirements

DSL or faster connections are recommended between the source and destination systems. Even connections with very low bandwidth, such as 56 Kbps, are possible.

You should preconfigure routers, switches, and DNS servers to direct users to alternate storage sites if the source system that they first attempt to access is unavailable.

Advantage

SnapVault provides centralized, inexpensive off-site backup.

SnapMirror

SnapMirror is an inter-site Snapshot copy backup, availability, and restorability tool. You can locate a SnapMirror destination off-site, at any distance from the source system.

Data availability

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

Data recoverability

If a data-loss disaster occurs at a source storage site, you can restore data that is backed up to a SnapMirror destination storage site. You can restore the data 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 last Snapshot copy that was replicated to the destination volume.

Connection requirements

DSL or faster connections are recommended between the source and destination systems. Even connections with very low bandwidth, such as 56 Kbps, are possible.

You should preconfigure routers, switches, and DNS servers to direct users to alternate storage sites if the source system that they first attempt to access is unavailable.

Advantage

SnapMirror provides off-site 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 *Data ONTAP High Availability and MetroCluster Configuration Guide for 7-Mode*.

Advantage

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

Related concepts

Data protection using SnapVault on page 229
Data protection using SnapMirror on page 76
What a Snapshot copy is on page 28

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.

For more information about the descriptions of data backup and restore on volumes containing Data ONTAP LUNs, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

Policy-based automated data protection using Protection Manager

Typically, data and resource management is time consuming because it involves manual analysis and management of storage capacity, network bandwidth, schedules, retention policies, and other infrastructure variables. The NetApp Management Console data protection capability simplifies this work by employing configuration policies (that you can assign to multiple storage systems, volumes, or qtrees), convenient wizards, and automated verification of certain aspects of the data protection configuration.

The protection application can perform the following actions:

- Use protection policies to manage primary data, storage, and backup and mirror relationships.
- Manage local and remote backups and mirror copies.
- Provision the secondary storage for backups and mirrored copies based on policies you assign.
- Enable disaster recovery capability if you install the licensed disaster recovery option.

- Automatically validate your backup and disaster recovery configuration with a conformance checker.

For more information, see the *OnCommand Unified Manager Guide to Common Provisioning and Data Protection Workflows for 7-Mode*.

Interoperability between volumes in 32-bit and 64-bit aggregates

Data ONTAP supports interoperability between volumes in 32-bit and 64-bit aggregates for data protection features such as qtree SnapMirror, SnapVault, volume SnapMirror, synchronous and semi-synchronous SnapMirror, and `vol copy` command.

Note: Starting with Data ONTAP 8.1, interoperability between volumes in 32-bit and 64-bit aggregates is supported in a synchronous and semi-synchronous SnapMirror relationship, only if both the source and the destination systems run the same version of Data ONTAP.

You can use the `vol move` command to move volumes between 32-bit aggregates and 64-bit aggregates nondisruptively.

For more information about the `vol move` command, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

Data replication using aggr copy

You can use the `aggr copy` set of commands to perform a block-to-block copy from one aggregate to another aggregate.

The `aggr` family of commands manages aggregates. You can initiate an aggregate copy with the `aggr copy start` command. This enables you to copy all the FlexVol volumes from one aggregate to another aggregate at the same time, either on the same or on a different storage system.

To use `aggr copy` command in Flash Pool aggregates, you must ensure that the following conditions exist:

- The source and the destination storage systems are of Flash Pool aggregate.
- The source and destination storage systems contain the same number of RAID groups, disks of the same size and type, RAID types, and RAID block checksums.
- The RAID groups, disks of the same size and type, RAID types, and RAID block checksums must be added in the same order in the source and destination storage systems.

You can use the `aggr copy start` command to generate aggregate copy operations, which produce screen messages that show the progress of the operations. Each `aggr copy start` command generates two aggregate copy operations, each of which is assigned a number. One operation is for reading data from the source aggregate and the other is for writing data to the destination aggregate. You need the aggregate copy operation number if you want to stop an aggregate copy operation or change the speed of the aggregate copy operation.

Copying one aggregate to another aggregate using the aggr copy command

You can use the `aggr copy start` command to copy one aggregate to another aggregate.

Before you begin

- You must ensure that `rsh` authentication is enabled on the source and destination systems.
- You must ensure that the `/etc/hosts.equiv` entry is available on the source system.

Steps

1. Enter the following command to restrict the destination aggregate:

   `aggr restrict` ***`aggr_name`***

2. Enter the following command on the source or the destination system:

aggr copy start [-p{inet|inet6}] [-S|-s *snapshot_name*] [-C] [*srcfiler*:]*srcpath* [*dstfiler*:]*dstpath*

The following table lists the parameters and their descriptions:

Parameter	Description
`-p`	Selects the IP connection mode. The value for this argument can be inet or inet6.
`-S`	Copies all Snapshot copies from the source aggregate to the destination aggregate.
`-s`	Copies a particular Snapshot copy from the source aggregate to the destination aggregate.
`-C`	Checks if reallocation exists in the source aggregate or the destination aggregate.

Note: If reallocation exists in the source aggregate, then the performance of the system is impacted.

For more information about reallocation, see the *Data ONTAP System Administration Guide for 7-Mode*.

Snapshot management

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.

What a Snapshot copy is

A Snapshot copy is a read-only image of a traditional or FlexVol volume, or an aggregate, that captures the state of the file system at a point in time. Data ONTAP maintains a configurable Snapshot copy schedule that creates and deletes Snapshot copies automatically for each volume. You can also create and delete Snapshot copies manually.

For information about traditional or FlexVol volumes or aggregates, see the *Data ONTAP Storage Management Guide for 7-Mode*.

You can store up to 255 manually initiated Snapshot copies at one time on each volume or up to 254 scheduled Snapshot copies.

You can specify the percentage of disk space that Snapshot copies can occupy. The default setting is 5 percent of the total (both used and unused) space on the disk for a FlexVol volume and 0 percent for aggregates.

Related concepts

Creation of Snapshot copy schedules on page 35
What Snapshot disk consumption is on page 44

How Snapshot copies handle file permissions

Snapshot files carry the same permissions and inode numbers as the original files, keeping the integrity of the security system intact.

Inodes are data structures that hold information (including permissions information) about files on the storage system. Every file in the file system is uniquely identified by its inode.

The inode number for a file in a Snapshot copy is the same as the inode number for the corresponding file in the active file system. As a result, some programs on UNIX clients consider the two files to be the same. For example, if you use an older version of the GNU diff program to compare the two files, it might not find any differences between them. However, newer versions of GNU diff should work. In a few cases, if you try to restore a file from a Snapshot copy, you might see the following error message:

```
cp:.snapshot/xxx and xxx are identical.
```

To ensure that the two files have different inode numbers before copying or comparing them, copy one of the files to a different name.

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 Data ONTAP LUN, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

For information about cloning a FlexVol volume, see the *Data ONTAP Storage Management Guide for 7-Mode*.

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

User access to Snapshot copies

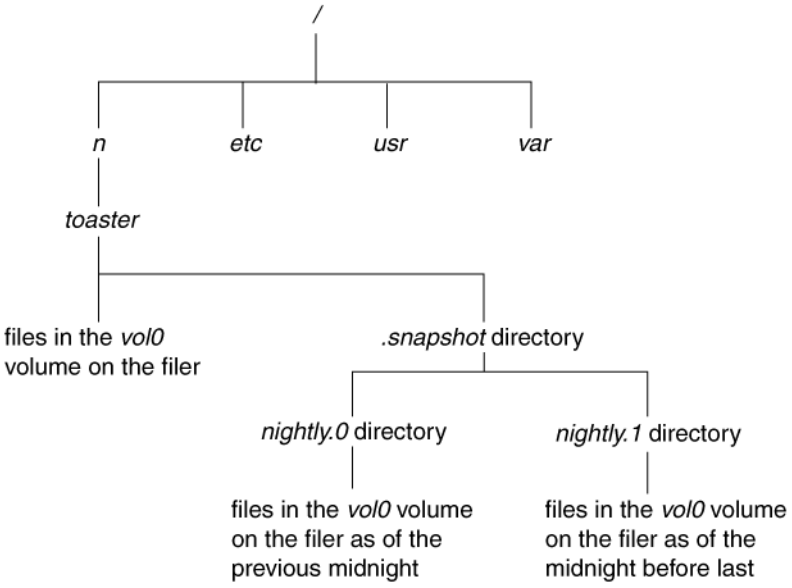
By default, every volume contains a directory named `.snapshot` through which users can access old versions of files in that directory. Users can gain access to Snapshot copies depending on the file-sharing protocol used—NFS or CIFS. Access to Snapshot copies can be turned off.

Snapshot files carry the same read permissions as the original file. A user who has permission to read a file in the volume can read that file in a Snapshot copy. A user without read permission to the volume cannot read that file in a Snapshot copy. Snapshot copies do not have write permissions.

Access to Snapshot copies over NFS

On an NFS client, the user can obtain access to Snapshot copies.

The following illustration shows the directory structure on an NFS client with the *vol0* volume named `toaster` mounted on the `/n/toaster` directory:



In this example, the user can obtain access to Snapshot copies in the `/n/toaster/.snapshot` directory. Notice that the `.snapshot` directory is shown only at the mountpoint, although it actually exists in every directory in the tree. The user, however, can only see the `.snapshot` directory at the mountpoint. That is, the `.snapshot` directory is accessible by name in each directory, but is only seen in the output of the `ls` command at the mountpoint.

For example, at the mountpoint of a file system, a directory listing looks like this:

```
systemA> ls -a
. .. .snapshot dir1 dir2
```

The same command entered in a directory below the mountpoint does not show the `.snapshot` directory; for example:

```
systemA> cd dir1
systemA> ls -a
. .. file1 file2
```

If you enter the `ls` command with the directory name `.snapshot`, you can see a directory for each of the Snapshot copies for the `dir1` directory:

```
systemA> ls .snapshot

hourly.0 hourly.4 nightly.0 nightly.4
hourly.1 hourly.5 nightly.1 nightly.5
hourly.2 hourly.6 nightly.2 weekly.0
hourly.3 hourly.7 nightly.3 weekly.1
```

If the `.snapshot` directory entry appeared in every directory, it would cause many commands to work improperly. For instance, all recursive commands for deleting files would fail because everything below the `.snapshot` directory is read-only. The recursive commands would copy everything in the Snapshot copies as well as files in the active file system. A `find` command would generate a list much longer than expected.

Access to Snapshot copies over CIFS

By default, CIFS users cannot see the `.snapshot` directory. To allow CIFS users to see the `.snapshot` directory, you can set the `cifs.show_snapshot` option to `on`.

To CIFS users, the `.snapshot` directory appears only at the root of a share. For example, if a user's home directory is a share named `bill` that corresponds to the `/vol/vol0/home/bill` directory, only the `/vol/vol0/home/bill/.snapshot` directory is visible. When this user displays the contents of the home directory, the `.snapshot` directory is displayed as `~snapshot` if the operating system supports long file names and as `~SNAPSHT` if the operating system supports only short file names.

Note: The `.snapshot` directory can be viewed in a directory listing or Windows Explorer display if the client operating system is configured to show hidden files.

In each directory within the share, a snapshot directory exists but is not visible to clients. For example, if the client operating system supports long file names, the applications on that operating system can use the Snapshot copy at each level of the share by using `.snapshot`, `~snapshot`, or `~SNAPSHT` as the directory name. The user cannot, however, display the directory name in any listing.

Accessing Snapshot copies from CIFS clients

You can access the data within Snapshot copies from CIFS clients.

Before you begin

Ensure that the `cifs.show_snapshot` option is set to `on`.

Step

1. To access Snapshot copies on Windows NT 4 or other Windows clients (Windows 95 or later), click **Start** > **Run menu**, then enter the following command:

   ***systemname**share*\.snapshot (or ~snapshot or ~SNAPSHT)**

   systemname is the name of the storage system you are using.

   share is the name of the Windows share that you want to access.

   #### Example

   \\systemA\home\.snapshot

Snapshot copies can also be accessed lower in the share by providing a path to a lower directory. Snapshot copies can be accessed through DOS on any system by changing to the `~SNAPSHT` directory.

Restricting access to Snapshot copies

You can restrict client access to Snapshot copies for a particular volume. This restriction could be due to security issues, or to prevent access to virus-infected files. You can enable or disable client access by using the `vol options` command.

Step

1. To specify client access to Snapshot copies within a volume, enter the following command:

   `vol options` *`vol_name`* `nosnapdir {on|off}`

   `vol_name` is the name of the volume for which you want to set client access to Snapshot copies.

   You can set the `nosnapdir` option to either `on` or `off`.

   - `on` - Disables client access to Snapshot copies, and hides the `.snapshot` directory from clients.
   - `off` - Enables client access to Snapshot copies, and makes the `.snapshot` directory visible to clients.

Disabling client access to Snapshot copies

To disable client access to Snapshot copies within the `volA` volume, enter the following command:

`vol options volA nosnapdir on`

How Data ONTAP Snapshot copies work in an iSCSI or FC network

If you take a Snapshot copy of a file system when an application is running, the Snapshot copy might contain inconsistent data. You can take measures (such as quiescing the application) to ensure that the data is consistent before you take the Snapshot copy.

To take a Snapshot copy of these types of applications, you should ensure that the files are closed and cannot be modified. When you quiesce an application or take it offline, the file system caches are committed before the Snapshot copy is taken. The Snapshot copy takes less than one second to complete, after which the application can resume normal operation.

Some applications take a lot of time to quiesce. To avoid a scenario in which the application is unavailable for a long time, some applications have a built-in hot backup mode. This allows a

Snapshot copy or a backup to occur while the application operates in a degraded mode, with limited performance.

Data ONTAP cannot take Snapshot copies of applications that have the ability to work with raw device partitions. You should use specialized modules from a backup software vendor tailored for such applications.

To back up raw partitions, it is best to use hot backup mode for the duration of the backup operation. For more information about backup and recovery of databases using SAN configurations, see the appropriate technical report for the database.

Related information

NetApp Library: www.netapp.com/tech_library

Using Snapshot copies in the SAN environment

You can use Snapshot copies in the SAN environment when the data within a Data ONTAP LUN is in a consistent state.

About this task

Data ONTAP cannot ensure that the data within a LUN is in a consistent state. That is, Data ONTAP does not know whether an application is accessing the data inside the LUN. Therefore, before creating a Snapshot copy, you need to quiesce the application or file system using the LUN. This action flushes the host file system buffers to disk. Quiescing ensures that the Snapshot copy is consistent.

One way to accomplish this is to use batch files and scripts on a host that has administrative access to the system.

Steps

1. Make the data within the LUN consistent with the application by quiescing a database, placing the application in hot backup mode, or taking the application offline.
2. Use the `rsh` or `ssh` command in the script to access the system.
3. Use the `snap` command to create the Snapshot copy on the system (this takes only a few seconds, regardless of volume size or use).
4. Return the application to normal operation.

   Snapshot copy scripts can be scheduled to run at specified intervals. On Windows hosts, you can use the Windows Task Scheduler. On UNIX hosts, you can use `cron` or other utilities. Also, you can use SnapDrive to save the contents of the host file system buffers to disk and to create Snapshot copies. For more information, see the *SnapDrive Installation and Administration Guide*.

Relationship between a LUN and a Snapshot copy

When you take a Snapshot copy of a Data ONTAP LUN, the Snapshot copy is initially backed by data in the LUN. After the Snapshot copy is taken, data written to the LUN is available in the active file system.

After you have a Snapshot copy, you can use it to create a LUN clone for temporary use as a prototype for testing data or scripts in applications or databases. Because the LUN clone is backed by the Snapshot copy, you cannot delete the Snapshot copy until you split the clone from it.

To restore the LUN from a Snapshot copy, you can use SnapRestore. However, the restored LUN does not have any updates to the data since the Snapshot copy was taken.

When you create the LUN, space reservation is enabled by default. This means that enough space is reserved so that write operations to the LUNs are guaranteed. The more space that is reserved, the less free space is available. If free space within the volume is below a certain threshold, Snapshot copies cannot be taken.

Restoring files from Snapshot copies

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

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.

Related concepts

When to use SnapRestore on page 67

Related tasks

Reverting a file to a selected Snapshot copy on page 72

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.

Creation of Snapshot copy schedules

Data ONTAP provides a default Snapshot copy schedule for each FlexVol volume. You can configure the schedule to fit your needs. The schedule creates Snapshot copies automatically, and deletes older Snapshot copies after a specified period.

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.

Related tasks

Changing the Snapshot copy schedule on page 40

Types of user-specified Snapshot copy schedules

You can configure weekly, nightly, or hourly Snapshot copy schedules by using the `snap sched` command.

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

Type	Description
Weekly	Data ONTAP creates these Snapshot copies every Sunday at midnight. Weekly Snapshot copies are named `weekly.`*n*, where *n* is an integer. The most recent weekly Snapshot copy is `weekly.0`, the next most recent weekly Snapshot copy is `weekly.1`, and so on.
Nightly	Data ONTAP creates these Snapshot copies every night at midnight, except when a weekly Snapshot copy is scheduled to occur at the same time. Nightly Snapshot copies are named `nightly.`*n*, where *n* is an integer. The most recent nightly Snapshot copy is `nightly.0`, `nightly.1` is the next most recent nightly Snapshot copy, and so on.

Type	Description
Hourly	Data ONTAP creates these Snapshot copies on the hour or at specified hours, except if a weekly or nightly Snapshot copy is scheduled to occur at the same time. Hourly Snapshot copies are named `hourly.`*n*, where *n* is an integer. The most recent hourly Snapshot copy is `hourly.0`, `hourly.0` is the next most recent hourly Snapshot copy, and so on.

When Data ONTAP creates a weekly, nightly, or hourly Snapshot copy, the value of *n* is adjusted for all the weekly, nightly, or hourly Snapshot copies; the earlier Snapshot copies in the series are renamed. The higher the value of *n*, the older the Snapshot copy.

Snapshot copy schedule conflicts

If SnapMirror or SnapVault is scheduled to perform Snapshot copy management at the same time as a `snap sched` command operation, then the Snapshot copy management operations scheduled using the `snap sched` command might fail with `syslog` messages.

The `syslog` messages are:

```
Skipping creation of hourly snapshot
```

and

```
Snapshot already exists
```

To avoid this condition, you should stagger the Snapshot copy update schedules so that SnapMirror activity does not begin or end at the exact minute a `snap sched` operation attempts to create a Snapshot copy. Additionally, if `snap sched` Snapshot copies conflict with SnapVault activity, you should use the `snapvault snap sched` command to configure equivalent schedules.

Note: If the scheduled creation of a Snapshot copy fails on the first attempt, Data ONTAP attempts up to two more times to create the Snapshot copy. If all three attempts fail, an error is logged.

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:

Type of Snapshot copy schedule	Value of the *period_snap* variable
Weekly	3 days
Nightly	3 days
Hourly	12 hours

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.

Viewing the Snapshot copy schedule using the CLI

You can view the Snapshot copy schedule for a volume by using the `snap sched` command.

Step

1. To view the Snapshot copy schedule for a volume, enter the following command:

   `snap sched [`*`vol_name`*`]`

   Note: If you do not specify a volume name, *`snap sched`* displays the Snapshot copy schedule for each volume on the system.

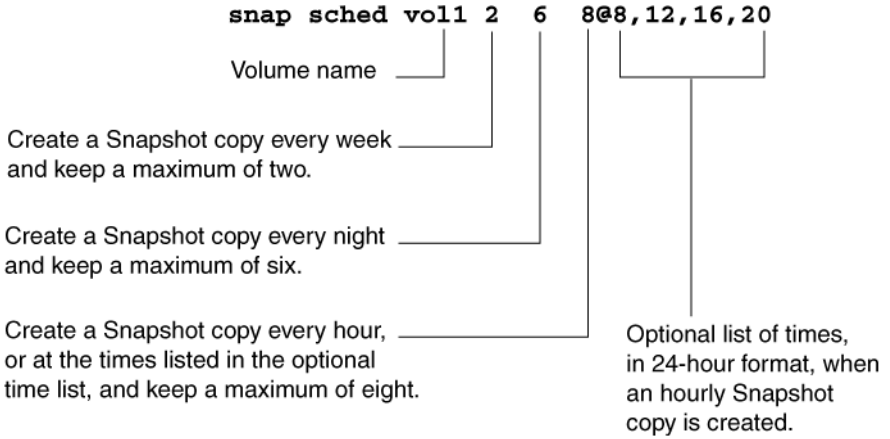
Example

```
systemA> snap sched vol1
Volume vol1: 2 6 8@8,12,16,20
```

What the snap sched command arguments mean

The `snap sched` command arguments allow you to specify the schedule for creating Snapshot copies. You can also specify the number of Snapshot copies to be retained.

The following illustration explains the arguments in a sample `snap sched` command output:



Snapshot copy schedule results: The command shown in the illustration keeps the two most recent weekly Snapshot copies, the six most recent nightly Snapshot copies, and the eight most recent hourly Snapshot copies, created at 8 a.m., 12 p.m., 4 p.m., and 8 p.m. every day. Whenever the Snapshot copy schedule creates a new Snapshot copy of a particular type, it deletes the oldest Snapshot copy and renames the existing Snapshot copies. On the hour, for example, the system deletes `hourly.7`, renames `hourly.0` to `hourly.1`, and so on.

Note: If you omit the `@` argument specifying the hours for the hourly Snapshot copies, Data ONTAP creates a Snapshot copy every hour. Nightly and weekly Snapshot copies are always created at midnight.

Strategies for creating a Snapshot copy schedule

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

Following are some strategies for scheduling and retaining Snapshot copies:

- If users rarely lose files or typically notice lost files right away, you can use the default Snapshot copy schedule.

  This schedule creates no weekly Snapshot copy; it creates a Snapshot copy every night and keeps two; and it creates hourly Snapshot copies at 8 a.m., 12 p.m., 4 p.m., and 8 p.m., and keeps six. Following is the default Snapshot copy schedule command:

  `snap sched *vol_name* 0 2 6@8,12,16,20`

- 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 schedule.

  Following is the recommended Snapshot copy schedule for this situation. It keeps two weekly Snapshot copies, six nightly Snapshot copies, and eight hourly Snapshot copies:

  `snap sched vol1 2 6 8@8,12,16,20`

  On many systems, only 5 to 10 percent of the data changes each week, so the Snapshot copy schedule of six nightly 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 schedules for different volumes on a storage system.

  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:

  `snap sched vol2 0 0 3`

  This schedule does not consume much disk space, and it lets users recover files in recent Snapshot copies as long as they notice their mistake within a couple of hours.

- 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.

Related concepts

What Snapshot disk consumption is on page 44

Changing the Snapshot copy schedule

You can change the Snapshot copy schedule for a specific volume by using the `snap sched` command.

Step

1. To change the Snapshot copy schedule for a specific volume, enter the following command:

   `snap sched` ***`vol_name weekly nightly hourly@n,n,...`***

   `vol_name` is the name of the specific volume for the Snapshot copy.

   `weekly` is the number of weekly Snapshot copies to keep.

   `nightly` is the number of nightly Snapshot copies to keep.

   `hourly` is the number of hourly Snapshot copies to keep.

   `n,n,`... specifies the hours at which to create the hourly Snapshot copies.

   Note: A zero in any of the three schedules (weekly, nightly, hourly) disables Snapshot copies for that interval.

Default snap sched command results

This is the default automatic Snapshot copy schedule:

`snap sched volx 0 2 6 @8,12,16,20`

The following example lists the Snapshot copies created using the default schedule (where January 11 is a Sunday):

```
ls -lu .snapshot
total 64
drwxrwsrwx 1 root 4096 Jan 14 12:00 hourly.0
drwxrwsrwx 1 root 4096 Jan 14 08:00 hourly.1
drwxrwsrwx 1 root 4096 Jan 13 20:00 hourly.2
drwxrwsrwx 1 root 4096 Jan 13 16:00 hourly.3
drwxrwsrwx 1 root 4096 Jan 13 12:00 hourly.4
drwxrwsrwx 1 root 4096 Jan 13 08:00 hourly.5
drwxrwsrwx 1 root 4096 Jan 14 00:00 nightly.0
drwxrwsrwx 1 root 4096 Jan 13 00:00 nightly.1
```

Note: Daily Snapshot copies are created at midnight of each day except Sunday, and weekly Snapshot copies are created at midnight on Sunday. Only one Snapshot copy is created at a time. If a weekly Snapshot copy is being created, for instance, a daily or hourly Snapshot copy is not created even if one is scheduled.

Enabling or disabling automatic Snapshot copies

You can disable the creation of automatic Snapshot copies, without changing the Snapshot copy schedule. Later, you can enable the Snapshot copy schedule.

Steps

1. To view whether the creation of automatic Snapshot copies is disabled, enter the following command:

   vol options *vol_name*

   vol_name is the name of the volume for which you want to view the value of the `nosnap` option.

   `nosnap=off` indicates that the creation of automatic Snapshot copies is enabled. `off` is the default value for the `nosnap` option.

   `nosnap=on` indicates that the creation of automatic Snapshot copies is disabled.

2. To change the value of the `nosnap` option, enter the following command:

   vol options *vol_name* nosnap {on|off}

Creating Snapshot copies manually

You might need to create Snapshot copies outside a specified schedule. You can create Snapshot copies manually by using the `snap create` command.

Step

1. To create a Snapshot copy manually, enter the following command:

   snap create *vol_name snapshot_name*

   vol_name is the name of the volume on which you want to create the Snapshot copy.

   snapshot_name is the name you want to give the Snapshot copy.

   Note: The `snap create` command does not accept a Snapshot copy name containing a slash (/). Therefore, you cannot enter a specific path for the Snapshot copy.

Why you might need to access a particular Snapshot copy

You might need to access an earlier version of a file, because a file was changed, corrupted, or erased. You might notice the issue only after one or more Snapshot copies were created with the

incorrect copy of the file. When looking for the version of the file you need, you should look for it by means of the creation time of the Snapshot copy.

The *version* of a file refers to the last time the file was modified before a Snapshot copy was created. The *access time* of a file refers to the Snapshot copy creation time for a file, regardless of whether any modifications were made to that file.

The best way to find all versions of a particular file preserved in Snapshot copies is to use the `ls` command from an NFS client, or to use the search functionality in Windows.

Finding the Snapshot copy you need from an NFS client

The best way to find all versions of a particular file preserved in Snapshot copies is to use the `ls` command from an NFS client.

Before you begin

Client must be connected to NFS to access the Snapshot copies.

Step

1. To find all the versions of a particular file in the Snapshot copies, enter the following command:

   `ls -l` *`filename`* `.snapshot/*/`*`file_name`*

   A list displays all the versions of the requested file.

Accessing Snapshot copies from an NFS client

```
ls -l myfile.txt .snapshot/*/myfile.txt

-rw-r--r-- 1 smith 0 Jan 14 09:40 myfile.txt
-rw-r--r-- 1 smith 0 Jan 13 18:39 .snapshot/nightly.0/myfile.txt
-rw-r--r-- 1 smith 0 Jan 12 19:17 .snapshot/nightly.1/myfile.txt
```

The version of `myfile.txt` in the active file system was last modified on January 14, but the old versions available in the Snapshot copies were modified on January 13 and January 12. You can use standard UNIX commands to read the earlier versions of `myfile.txt`. However, you cannot modify or delete these older versions because everything in the `.snapshot` directory is read-only.

Determining access times from an NFS client

When Data ONTAP creates a Snapshot copy, the access time of each file in the Snapshot copy is updated to the Snapshot copy creation time. You can use the `ls -lu` command to see the access times.

Step

1. To determine when Snapshot copies were created, from an NFS client, enter the following command:

   `ls -lu` *`filename`* `.snapshot/*/`*`file_name`*

```
ls -lu myfile.txt .snapshot/*/myfile.txt
-rw-r--r-- 1 smith 0 Jan 14 09:40 myfile.txt
-rw-r--r-- 1 smith 0 Jan 14 00:00 .snapshot/nightly.0/myfile.txt
-rw-r--r-- 1 smith 0 Jan 13 00:00 .snapshot/nightly.1/myfile.txt
```

Note: On a UNIX client, the times listed by the `ls -l` command reflect the modification times of the directory at the time of each Snapshot copy, and are not related to the times at which the Snapshot copies are created.

Finding the Snapshot copy you need from a CIFS client

The best way to find all versions of a particular file preserved in Snapshot copies is to use the search functionality in Windows.

Steps

1. To find all the versions of a particular file in the Snapshot copies, click **Start** > **Search** > **For Files or Folders...** in the Windows **Start** menu.

   A search window opens prompting you for a directory and file name.

2. In the search window, enter the file name to search for in the `~snapshot` directory.

CIFS client example

If you map the home share to drive `F:` and want to find all versions of `myfile.txt` in Snapshot copies, you can use the Windows search functionality to search for `myfile.txt` in the `f:\~snapshot` folder.

How to determine access times from a CIFS client

You can determine the access time of a file from a CIFS client by checking the properties of the file.

What Snapshot disk consumption is

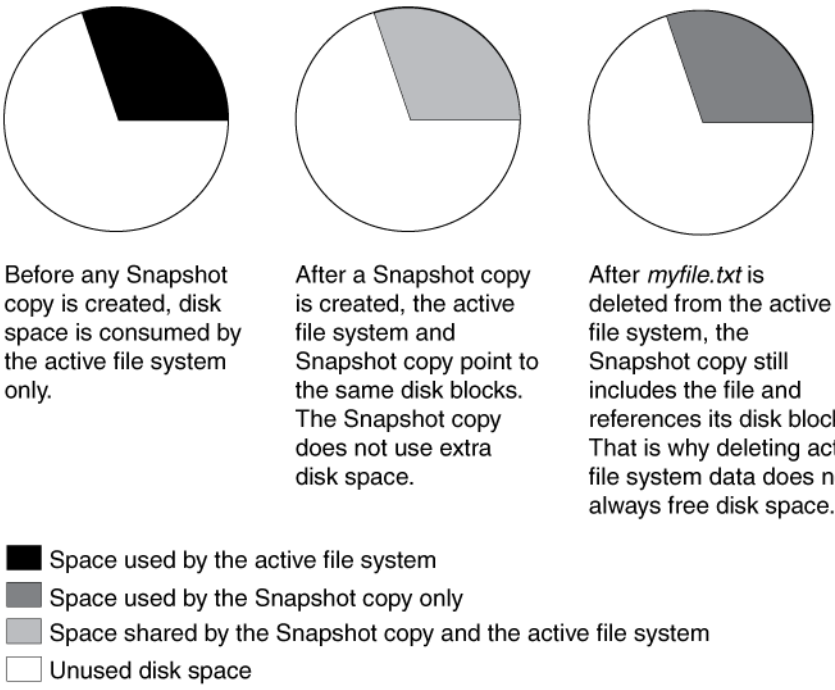
Data ONTAP preserves pointers to all the disk blocks currently in use at the time the Snapshot copy is created. When a file is changed, the Snapshot copy still points to the disk blocks where the file existed before it was modified, and changes are written to new disk blocks.

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.

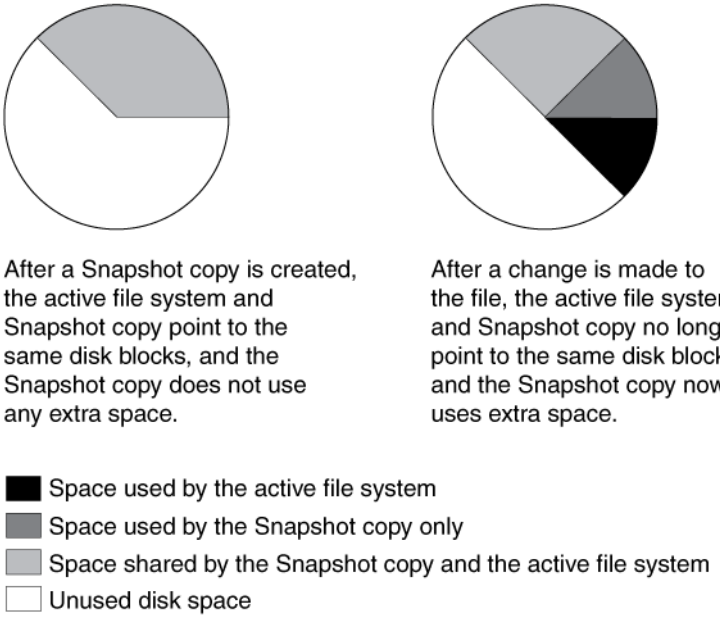
The following illustration shows disk space usage for a Snapshot copy:



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:



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.

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 (`kbytes` 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.

```
systemA> df
Filesystem             kbytes   used     avail    capacity
/vol/vol0              3000000  2000000  1000000   65%
/vol/vol0/.snapshot 1000000  500000   500000    50%
```

Displaying Snapshot copy disk consumption statistics

You can use the `snap list` command to view the disk space utilization by Snapshot copies. This information helps you determine an appropriate Snapshot copy reserve. The `snap list` command also shows whether a Snapshot copy is required for a replication operation, such as SnapMirror.

Step

1. To display the Snapshot copy statistics for a volume, enter the following command:

   snap list *vol_name*

   vol_name is the name of the volume for which you want statistics.

   If you do not specify a volume name in the command, the output contains statistics about each volume in the system.

Example

The following example gives a sample output of the `snap list` command:

```
systemA> snap list vol0

Volume vol0
%/used     %/total    date         name
---------- ---------- ------------ --------
0% ( 0%)   0% ( 0%)   Jan 19 08:01 hourly.0
1% ( 1%)   1% ( 1%)   Jan 19 00:01 nightly.0
2% ( 2%)   2% ( 2%)   Jan 18 20:01 hourly.1
3% ( 2%)   2% ( 2%)   Jan 18 16:01 hourly.2
```

```
3% ( 2%)   3% ( 2%)   Jan 18 12:01 hourly.3
5% ( 3%)   4% ( 3%)   Jan 18 00:01 nightly.1
7% ( 4%)   6% ( 4%)   Jan 17 00:00 nightly.2
8% ( 4%)   7% ( 4%)   Jan 16 00:01 nightly.3
10%( 5%)   9% ( 4%)   Jan 15 00:01 nightly.4
```

How the snap list output is calculated

The `snap list` output is calculated in a number of ways. The output can show both used space and available space.

The **%/used** column shows the space consumed by Snapshot copies as a percentage of disk space being used in the volume. The first number is cumulative for all Snapshot copies listed, and the second number is for the specified Snapshot copy alone.

- The first number is equal to

$$100\% \times \frac{\text{cumulative Snapshot copy space}}{\text{cumulative Snapshot copy space + file system}}$$

- The second number is equal to

$$100\% \times \frac{\text{this Snapshot copy}}{\text{this Snapshot copy + file system space}}$$

The **%/total** column shows space consumed by Snapshot copies as a percentage of total disk space (both space used and space available) in the volume.

- The first number is equal to

$$100\% \times \frac{\text{cumulative Snapshot copy space}}{\text{total disk space in this volume}}$$

  Cumulative Snapshot copy space is the total space used by this Snapshot copy and all other recent Snapshot copies (the ones preceding this Snapshot copy in the `snap list` output).

- The second number is equal to

$$100\% \times \frac{\text{this Snapshot copy}}{\text{total disk space in this volume}}$$

Summary of the `snap list` command output: The **%/used** number is useful for planning the Snapshot copy reserve because it is more likely to remain constant as the file system fills.

The information shows a volume that keeps five nightly Snapshot copies and four hourly Snapshot copies.

The sample output shows that the overhead for Snapshot copies is only 10 percent, so the default Snapshot copy reserve of 20 percent seems to be a waste of disk space. If this pattern of change holds, a reserve of 12 percent to 15 percent provides a safe margin to ensure that deleting files frees disk space when the active file system is full.

The values in parentheses, which show the space used by an individual Snapshot copy, are useful in identifying a particular Snapshot copy to delete when the file system is full. However, deleting a particular Snapshot copy does not necessarily release the amount of disk space indicated because other Snapshot copies might be referring to the same blocks.

Related concepts

What the Snapshot copy reserve is on page 50

How to use cumulative Snapshot copy values

If you want the amount of disk space consumed by all Snapshot copies not to exceed a certain percentage of the used disk space, you can use the cumulative values in the `snap list` command output to determine which Snapshot copies to delete.

For example, if you do not want more than 5 percent of used disk space to be spent by Snapshot copies, you can delete all Snapshot copies listed below `nightly.1` in the `snap list` output; that is, `nightly.2`, `nightly.3`, and `nightly.4`. After deleting the Snapshot copies, `nightly.1` and all the other more recent Snapshot copies consume 5 percent of the used disk space.

Displaying Snapshot copy use and dependencies

The output of the `snap list` command shows whether a Snapshot copy is being actively used by an application or a replication operation.

Step

1. To view the list of Snapshot copies, enter the following command:

   `snap list` *`vol_name`*

Example

A sample output is given in the following example:

```
systemA> snap list vol1

Volume vol1
%/used      %/total  date name
-------- ------   ------------ --------
0% ( 0%) 0% ( 0%) Jan 19 08:01 hourly.0 (busy)
1% ( 1%) 1% ( 1%) Jan 19 00:01 nightly.0(snapmirror)
```

```
2% ( 2%) 2% ( 2%) Jan 18 20:01 hourly.1 (busy,snapmirror)
3% ( 2%) 2% ( 2%) Jan 18 16:01 hourly.2
3% ( 2%) 3% ( 2%) Jan 18 12:01 hourly.3
5% ( 3%) 4% ( 3%) Jan 18 00:01 nightly.1 (backup[9])
7% ( 4%) 6% ( 4%) Jan 17 00:00 nightly.2
8% ( 4%) 7% ( 4%) Jan 16 00:01 nightly.3
10%( 5%) 9% ( 4%) Jan 15 00:01 nightly.4
```

The `snap list` command displays the name of an application next to a Snapshot copy name if the application needs the Snapshot copy currently or at a later time. For example, `backup` is displayed next to the Snapshot copy name to show that the Snapshot copy is the result of a `dump` command transfer that was interrupted but is restartable. The number following `backup` is the backup ID assigned by the `backup status` command. The notation `snapmirror` next to the Snapshot copy name means that SnapMirror is retaining the Snapshot copy to maintain a source-destination relationship.

Note: Ownership information for a busy Snapshot copy is useful for determining whether to stop the activity in progress. For example, if the `snap list` command output displays a locked Snapshot copy that is imposing a resource constraint, you can delete that Snapshot copy and free up space.

snap list performance after a snap restore file operation

If you restore files with the `snap restore` command, and then issue the `snap list` command, the `snap list` command can take up to several minutes to complete.

This condition persists until the Snapshot copy from which you restored the file is purged from the system after reaching the end of its normal Snapshot retention cycle.

Related tasks

Reverting a file to a selected Snapshot copy on page 72

Understanding Snapshot copy reserve

Snapshot copy reserve sets a specific percent of the disk space for storing Snapshot copies. If the Snapshot copies exceed the reserve space, they spill into the active file system and this process is called Snapshot spill.

The Snapshot copy reserve must have sufficient space allocated for the Snapshot copies. If the Snapshot copies exceeds the reserve space, you must delete the existing Snapshot copies from the active file system to recover the space, for the use of the file system. You can also modify the percent of disk space that is allotted to Snapshot copies.

What the Snapshot copy reserve is

The Snapshot copy reserve sets a specific percent of the disk space for Snapshot copies. For traditional volumes, the default Snapshot copy reserve is set to 20 percent of the disk space. For FlexVol volumes, the default Snapshot copy reserve is set to 5 percent of the disk space.

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.

Use of 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.

Example

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:

```
Filesystem           kbytes   used    avail   capacity
/vol/vol0/           3000000  3000000  0       100%
/vol/vol0/.snapshot  1000000  500000   500000  50%
```

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:

```
Filesystem           kbytes    used     avail    capacity
/vol/vol0/           3000000   2900000  100000   97%
/vol/vol0/.snapshot  1000000   600000   400000   60%
```

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 `df` output is as follows:

```
Filesystem          kbytes  used    avail  capacity
/vol/vol0/          3000000 3000000 0       100%
/vol/vol0/.snapshot 1000000 500000  500000   50%
```

After the deletion, the `df` command generates the following output:

```
Filesystem          kbytes  used    avail  capacity
/vol/vol0/          3000000 2500000 500000   83%
/vol/vol0/.snapshot 1000000 3500000 0       350%
```

The entire 3,000,000 KB (3 GB) in the active file system is still being used by Snapshot copies, along with the 500,000 KB (0.5 GB) that was being used by Snapshot copies before, making a total of 3,500,000 KB (3.5 GB) of Snapshot copy data. This is 2,500,000 KB (2.5 GB) more than the space reserved for Snapshot copies; therefore, 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 `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 `df` command generates the following output:

```
Filesystem          kbytes   used     avail capacity
/vol/vol0           3000000  3000000  0     100%
/vol/vol0/.snapshot 1000000  3500000  0     350%
```

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.

Related tasks

Displaying Snapshot copy disk consumption statistics on page 46

Changing the Snapshot copy reserve

You can change the percent of disk space reserved for Snapshot copies by using the `snap reserve` command.

About this task

You must ensure that there is sufficient Snapshot copy reserve space allocated for the Snapshot copies. The space must not be so less so that it consumes active file system space and must not be so large that it wastes space that could be used by the active file system. Be default, for traditional volumes, the Snapshot copy reserve is set to 20 percent of the disk space and for FlexVol volumes, the Snapshot copy reserve is set to 5 percent of the disk space.

Step

1. To change the percent of disk space used for the Snapshot copy reserve, enter the following command:

   `snap reserve` *`vol_name percent`*

   `vol_name` is the name of the volume.

   `percent` is the percent of disk space you want to reserve for Snapshot copies.

   #### Example

   `snap reserve vol1 25`

What file folding means and how it saves disk space

File folding describes the process of checking the data in the most recent Snapshot copy, and if this data is identical to the Snapshot copy currently being created, by referencing the previous Snapshot copy instead of taking up disk space writing the same data in the new Snapshot copy.

File folding saves disk space by sharing unchanged file blocks between the active version of the file and the version of the file in the latest Snapshot copy, if any.

The system must compare block contents when folding a file, so file folding might affect system performance.

If the folding process reaches a maximum limit on memory usage, it is suspended. When memory usage falls below the limit, the processes that were halted are restarted.

Enabling file folding

You can enable or disable file folding by using the `cifs.snapshot_file_folding.enable` option.

About this task

This option is only available for CIFS, and not for NFS.

Step

1. As required, choose one of the actions from the following table:

If you want to turn file folding...	Then enter the following command...
On	**`options cifs.snapshot_file_folding.enable on`**
Off	**`options cifs.snapshot_file_folding.enable off`**

Displaying the rate of change between Snapshot copies

You can use the `snap delta` command to view the rate of change between two Snapshot copies as well as the rate of change between a Snapshot copy and the active file system. This information can help you determine a suitable Snapshot copy schedule and Snapshot copy reserve.

About this task

Data ONTAP displays the rates of change in two tables. The first table displays rates of change between successive Snapshot copies. The second table displays a summary of the rate of change between the oldest Snapshot copy and the active file system. For details, see the na_snap(1) man page.

Step

1. To display data change rates on a volume, enter the following command:

   `snap delta` ***`vol_name`***

   `vol_name` is the name of the volume containing the Snapshot copies.

   Note: You can display change rates for all volumes by omitting the volume name.

Example

The following command lists the rates of change for the vol0 volume:

```
system> snap delta vol0
Volume vol0 working...
From Snapshot To                 KB changed Time       Rate (KB/hour)
------------ ------------------ ---------- ---------- --------------
hourly.0     Active File System 149812     0d 03:43   40223.985
hourly.1     hourly.0           326232     0d 08:00   40779.000
hourly.2     hourly.1           2336 1d    12:00      64.888
hourly.3     hourly.2           1536 0d    04:00      384.000
hourly.4     hourly.3           1420 0d    04:00      355.000
nightly.0    hourly.4           1568 0d    12:00      130.666
hourly.5     nightly.0          1400 0d    04:00      350.000
nightly.1    hourly.5           10800 201d 21:00      2.229
Summary...
From Snapshot To                 KB changed Time       Rate (KB/hour)
------------ ------------------ ---------- ---------- --------------
nightly.1    Active File System 495104     204d       20:43 100.697
```

Displaying rates of change between Snapshot copies

You can display the rate of change between Snapshot copies.

Step

1. To display data change rates between two Snapshot copies, enter the following command:

   snap delta *vol_name snap1 snap2*

   vol_name is the name of the volume containing the Snapshot copies.

   snap1 and *snap2* are the names of the two Snapshot copies.

Example

The following command lists the rate of change between nightly.0 and hourly.1 of the vol0 volume:

```
system> snap delta vol0 nightly.0 hourly.1

Volume vol0 working...
From Snapshot To         KB changed Time       Rate (KB/hour)
------------ ---------- ---------- ---------- --------------
hourly.2     hourly.1   2336       1d 12:00   64.888
hourly.3     hourly.2   1536       0d 04:00   384.000
hourly.4     hourly.3   1420       0d 04:00   355.000
nightly.0    hourly.4   1568       0d 12:00   130.666
```

```
Summary...
From Snapshot To         KB changed Time       Rate (KB/hour)
------------ ---------- ---------- ---------- --------------
nightly.0    hourly.1   6860       2d 08:00   122.500
```

Deleting Snapshot copies automatically

You can define and enable a policy for automatically deleting Snapshot copies by using the `snap autodelete` command. Automatically deleting Snapshot copies can help you manage space utilization.

Step

1. To enable a policy for automatically deleting Snapshot copies, enter the following command:

   `snap autodelete` ***`vol_name`*** **`on`** ***`option value`***

   `vol_name` is the name of the volume.

   The `on` option enables the Snapshot copy autodelete policy.

   Note: You cannot delete Snapshot copies that are locked by LockVault.

   To specify which Snapshot copies to delete, enter the following options and their values:

Option	Value
`commitment`	Specifies whether Snapshot copies locked by data protection utilities (SnapMirror or `ndmpcopy` command) or data backup mechanisms (volume or LUN clones) can be deleted. • `try` Deletes all Snapshot copies that are unlocked. This option deletes Snapshot copies that are not locked by data protection utilities or data backup mechanisms. • `disrupt` Deletes Snapshot copies that are unlocked or locked by data protection utilities. Snapshot copies that are locked by data backup mechanisms cannot be deleted by using the `disrupt` option. This option is used when there are no Snapshot copies for the `try` option to delete. • `destroy` Deletes Snapshot copies that are locked by data protection utilities and data backup mechanisms.

Option	Value
`destroy_list`	Specifies whether Snapshot copies locked by LUN clones, file clones, volume clones, or CIFS share can be deleted. • `lun_clone` Deletes Snapshot copies that are locked by LUN clones. • `file_clone` Deletes Snapshot copies that are locked by file clones. • `vol_clone` Deletes Snapshot copies that are locked by volume clones. • `cifs_share` Deletes Snapshot copies that are locked by CIFS share. • `none` Prevents all Snapshot copies from being deleted. You can specify multiple values (except `none`) as a comma-separated list for the `destroy_list` option. **Note:** You can use this option only if you specify `destroy` for the `commitment` option.

Option	Value
`trigger`	Specifies when to begin automatically deleting Snapshot copies. • `volume` Begins deleting Snapshot copies when the volume reaches its autodelete threshold and no space is available in the Snapshot copy reserve. • `snap_reserve` Begins deleting Snapshot copies when the Snapshot copy reserve reaches the autodelete threshold. • `space_reserve` Begins deleting Snapshot copies when the space reserved in the volume reaches the autodelete threshold and no space is available in the Snapshot copy reserve. The autodelete threshold is some percentage of the volume size, as follows: • If the volume size is less than 20 GB, the autodelete threshold is 85%. • If the volume size is equal to or greater than 20 GB and less than 100 GB, the autodelete threshold is 90%. • If the volume size is equal to or greater than 100 GB and less than 500 GB, the autodelete threshold is 92%. • If the volume size is equal to or greater than 500 GB and less than 1 TB, the autodelete threshold is 95%. • If the volume size is equal to or greater than 1 TB, the autodelete threshold is 98%.

Option	Value
`target_free_space`	This is determined by the "trigger", and it determines when to stop deleting Snapshot copies. If the trigger is `snap_reserve`, then the `target_free_space` value is some percentage of the `snap_reserve` space of the volume. For example, if you specify 20, then snapshot copies are deleted until 20% of the `snap_reserve` is free space. If the trigger is `space_reserve`, then the `target_free_space` value is some percentage of the `space_reserve` space of the volume. `space_reserve` triggers snapshot delete when the space reserved in the volume reaches "threshold" capacity and the volume's snap reserve has been exceeded. "Threshold capacity" is determined by the size of the volume as given above. If the trigger is volume, then the `target_free_space` value is some percentage of the volume. For example, if you specify 20, then Snapshot copies are deleted until 20% of the volume is free space. It is a common practice to specify the `target _free_space value` 2% less than the autodelete threshold to avoid deleting Snapshot copies unnecessarily.
`delete_order`	Specifies which Snapshot copies to delete first: newest or oldest. • `newest_first` Deletes the most recent Snapshot copies first. • `oldest_first` Deletes the oldest Snapshot copies first.
`defer_delete`	Deletes one of the following types of Snapshot copies last: • `scheduled` Snapshot copies that are scheduled automatically. • `user_created` Snapshot copies that are not scheduled automatically. • `prefix` Snapshot copies with the specified prefix string. • `none` Deletes all Snapshot copies immediately.

Option	Value
`prefix`	Deletes Snapshot copies with a specific prefix last. You can specify up to 15 characters (Example: `sv_snap_week`). **Note:** You can use this option only if you specify `prefix` for the `defer_delete` option.

Deleting Snapshot copies automatically without options

You can define and enable a policy for automatically deleting Snapshot copies when there are no options.

Step

1. Enter the following command:

   `snap autodelete` *`vol_name`* `on`

   `on` enables the Snapshot copy autodelete policy.

   Note: Do not specify `on` when using options to specify the snapshot copies to be deleted.

Example

`snap autodelete vol0 on`

Viewing settings for the automatic deletion of Snapshot copies

You can view the settings for the automatic deletion of Snapshot copies by using the `snap autodelete` command.

Step

1. To view the settings for the automatic deletion of Snapshot copies for a given volume, enter the following command:

   `snap autodelete` *`vol_name`* `show`

   The `snap autodelete` settings for Snapshot copies revert to the following defaults:

   - state—off
   - commitment—try
   - trigger—volume
   - target_free_space—20%
   - delete_order—oldest_first

- defer_delete—user_created
- prefix—no prefix specified

Restoring the default settings for the automatic deletion of Snapshot copies

You can restore the default settings for the automatic deletion of Snapshot copies by using the `snap autodelete` command.

Step

1. To restore the default settings for the automatic deletion of Snapshot copies, enter the following command:

   `snap autodelete` *`vol_name`* `reset`

   `vol_name` is the name of the volume.

   The `snap autodelete` settings for Snapshot copies revert to the following defaults:

   - state—off
   - commitment—try
   - trigger—volume
   - target_free_space—20%
   - delete_order—oldest_first
   - defer_delete—user_created
   - prefix—no prefix specified

Disabling a policy for automatic deletion of Snapshot copies

You can disable a policy for automatic deletion of Snapshot copies by using the `snap autodelete` command.

Step

1. To disable a policy for automatic deletion of Snapshot copies, enter the following command:

   `snap autodelete` *`vol_name`* `off`

   `vol_name` is the name of the volume.

   After you disable the policy, Snapshot copies are not automatically deleted when the volume is nearly full.

Displaying space reclaimed from deleted Snapshot copies

You can display the amount of space you can reclaim by deleting one or more Snapshot copies in a volume by using the `snap reclaimable` command. The amount of space displayed is an approximation because writing to the volume, creating Snapshot copies, or deleting Snapshot copies causes the reclaimed amount to change.

Step

1. To display the amount of space you can reclaim by deleting Snapshot copies, enter the following command:

   `snap reclaimable` ***`vol_name snap1`*** **`[`*** ***`snap2`*** **`...]`**

   `vol_name` is the volume which contains the Snapshot copies you might delete.

   `snap1` [*`snap2`* ...] are the names of Snapshot copies you might delete. The names are separated by a space.

   Note: It might take a while for Data ONTAP to display the amount of freed space. You can press Ctrl-C to interrupt the command.

Example

The following command displays the amount of space reclaimed by deleting the hourly.4, hourly.5, and nightly.0 Snapshot copies in the vol1 volume:

```
system> snap reclaimable vol1 hourly.4 hourly.5 nightly.0

Processing (Press Ctrl-C to exit) ...
snap reclaimable: Approximately 240 kbytes would be freed.
```

How to determine which Snapshot copies to delete on the basis of size

You can use the `snap list` command output to determine which Snapshot copies to delete to free the most disk space.

Before trying to conserve space by deleting a large Snapshot file, you should examine the cumulative values in the `snap list` output. If two adjacent Snapshot files show little difference in their cumulative values, most of the data referenced by these Snapshot copies is the same. In this case, deleting only one of the Snapshot copies does not free much disk space.

In many cases, you can use the default Snapshot schedule and the default Snapshot reserve because these settings are appropriate for most environments. When you create a new volume, the new volume inherits the Snapshot schedule from the root volume. After you use the volume for several days, check how much disk space the Snapshot copies are consuming in the volume. If the amount seems high, you can decrease the amount of time that Snapshot copies are kept or increase the Snapshot reserve.

As you use Snapshot copies, you should continue to watch the statistics change over time. The statistics help you gain a better understanding of how Snapshot copies use disk space.

Attention: As a general rule, you should avoid deleting Snapshot copies that are not the product of the `snap sched` command (for example, Snapshot copies generated by SnapMirror or SnapVault commands). Deleting these Snapshot copies could halt the SnapMirror or SnapVault processes. An exception would be Snapshot copies left over from old SnapMirror relationships that you no longer want to maintain.

Related tasks

Displaying Snapshot copy disk consumption statistics on page 46

Deleting a Snapshot copy manually

You can use the `snap delete` command to delete a Snapshot copy before the preset interval to free disk space or because it is a manual Snapshot copy that is no longer needed but is not going to be automatically deleted.

Step

1. To delete a Snapshot copy manually from a specific volume, enter the following command:

   `snap delete` *`vol_name snapshot_name`*

   `vol_name` is the name of the volume that contains the Snapshot copy to delete.

   `snapshot_name` is the specific Snapshot copy to delete.

   Note: To delete all Snapshot copies on a volume, use the `-a` parameter:

   `snap delete -a` *`vol_name`*

Manual deletion of a busy or locked Snapshot copy

You can use the `snap delete` command to view ownership information of busy Snapshot copies. Before you can delete a busy Snapshot copy, you need to release the Snapshot copy from the application that is using it.

This information is useful for determining why a particular Snapshot copy is busy, and whether to stop the activity in progress. For example, if the `snap delete` command output displays a locked Snapshot copy that is imposing a resource constraint, you can delete that Snapshot copy and free up space.

If a Snapshot copy is locked, the `snap delete` operation fails until you execute a `snapmirror release` or `snapvault release` command to unlock the Snapshot copy. Snapshot copies are locked because SnapMirror or SnapVault is maintaining these copies for the next update.

Attention: Deleting a locked Snapshot copy would prevent SnapMirror or SnapVault from correctly replicating a file or volume as specified in the schedule you set up.

How to delete a locked SnapMirror Snapshot copy

The following example shows how to delete a SnapMirror Snapshot copy that is locked because SnapMirror requires it for an update:

```
systemA> snap delete vol0 oldsnap
Can't delete oldsnap: snapshot is in use by snapmirror.
Use 'snapmirror destinations -s' to find out why.
systemA> snapmirror destinations -s vol0
Path Destination
/vol/vol0 systemB:vol0
systemA> snapmirror release vol0 systemB:vol0
systemA> snap delete vol0 oldsnap
```

How to delete a locked SnapVault Snapshot copy

The following example shows how to delete a SnapVault Snapshot copy that is locked because SnapVault requires it for an update:

```
systemA> snap delete vol0 oldsnap
Can't delete oldsnap: snapshot is in use by snapvault.
Use 'snapvault status -l' to find out why.
systemA> snapvault status -l
SnapVault client is ON.
Source: systemA:/vol/vol0/qt3
Destination systemB:/vol/sv_vol/qt3...
systemA> snapvault release /vol/vol0/qt3
systemB:/vol/sv_vol/qt3
systemA> snap delete vol0 oldsnap
```

Related tasks

Releasing SnapVault relationships on page 282
Releasing partners from a SnapMirror relationship on page 177

Renaming Snapshot copies

You might want to rename a Snapshot copy generated by the `snap sched` command if it contains data that you want to save. The `snap sched` command overwrites and deletes regularly scheduled Snapshot copies. You can use the `snap rename` command to save a Snapshot copy.

About this task

When renaming a Snapshot copy, you should use a name that does not begin with one of the following standard prefixes: `weekly`, `nightly`, or `hourly`. Otherwise, Data ONTAP deletes the renamed Snapshot copy as per the schedule.

Step

1. To rename a Snapshot copy, enter the following command:

   snap rename *vol_name* *from_name* *to_name*

   vol_name is the name of the volume that contains the Snapshot copy to rename.

   from_name is the current name of the Snapshot copy to rename.

   to_name is the new name you want to give to the Snapshot copy.

   Example

   ```
   snap rename vol0 hourly.2 MyDataSave
   ```

Volume move and snap commands

During the volume move cutover phase, the SnapMirror source volume is not accessible. Therefore, the `snap` commands do not work.

The following are the `snap` commands that do not work during the volume move cutover phase:

- `snap autodelete`
- `snap create`
- `snap delete`
- `snap delta`
- `snap list`
- `snap reclaimable`
- `snap rename`

- `snap reserve`
- `snap restore`
- `snap sched`
- `snap status`

For more information about volume move, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

Data recovery using SnapRestore

SnapRestore uses Snapshot technology and enables you to recover data from any one of the Snapshot copies stored on the file system in case of a disaster.

What SnapRestore is

You can use the SnapRestore feature to recover data that is no longer available or if you are testing a volume or file and want to restore that volume or file to pre-test conditions.

Note: SnapRestore is a licensed feature. You must purchase and install the license key before you can use it.

What SnapRestore does

SnapRestore enables you to quickly revert a local volume or file to the state it was in when a particular Snapshot copy was taken. In most cases, reverting a file or volume is much faster than restoring files from tape or copying files from a Snapshot copy to the active file system.

How SnapRestore works: After you select a Snapshot copy for reversion, the Data ONTAP reverts the specified file or the volume to the data and timestamps that it contained when the selected Snapshot copy was taken. Data that was written after the selected Snapshot copy was taken is lost.

Note: If the volume you select to revert is a root volume, the system reboots.

What SnapRestore reverts: SnapRestore reverts only the file contents. It does not revert attributes of a volume. For example, the Snapshot copy schedule, volume option settings, RAID group size, and maximum number of files per volume remain unchanged after the reversion.

When to use SnapRestore: You use SnapRestore to recover from data corruption. If a primary system application corrupts data files in a volume, you can revert the volume or specified files in the volume to a Snapshot copy taken before the data corruption.

Why use SnapRestore rather than copying from a Snapshot copy: SnapRestore performs Snapshot copy restoration more quickly, using less disk space, than an administrator can achieve by manually copying volumes, qtrees, directories, or large files to be restored from the Snapshot copy system to the active file system. A large volume directory restore can be carried out in a few seconds using the SnapRestore feature.

SnapRestore can restore large volumes or files even if space limitations would prevent restoring by copying from a Snapshot copy.

When to use SnapRestore

You can use SnapRestore to recover from data corruption. If a primary system application corrupts data files in a volume, you can revert the volume or specified files in the volume to a Snapshot copy taken before the data corruption.

You must take into account certain considerations, prerequisites, and general cautions before deciding whether to use SnapRestore to revert a file or volume.

Related tasks

Reverting a file to a selected Snapshot copy on page 72
Reverting a volume to a selected Snapshot copy on page 70
Obtaining correct incremental backups after reversion on page 75

Considerations before using SnapRestore

You must take into account certain considerations before deciding whether to use SnapRestore to revert a file or volume.

- If the volume that you need to restore is a root volume, it is easier to copy the files from a Snapshot copy or restore the files from tape than to use SnapRestore.
  This enables you to avoid rebooting. However, if you need to restore only a corrupted file on a root volume, a reboot is not necessary.
- If you revert the entire root volume, the system reboots with configuration files that were in effect when the Snapshot copy was taken.
- If the amount of data to be recovered is large, SnapRestore is the preferred method, because it takes a long time to copy large amounts of data from a Snapshot copy or to restore from tape.
- If a file to be recovered needs more space than the amount of free space in the active file system, you cannot restore the file by copying from the Snapshot copy to the active file system.
  For example, if a 10-GB file is corrupted and only 5 GB of free space exists in the active file system, you cannot copy the file from a Snapshot copy to recover the file. However, SnapRestore can quickly recover the file in these conditions. You do not have to spend time making the additional space available in the active file system.
- If you revert a FlexVol volume using SnapRestore, the size of the volume remains unchanged after the reversion.
  However, the volume that is being reverted assumes the size of the volume in the Snapshot copy if the following conditions are met:
  - The volume that is being reverted is contained in a 64-bit aggregate.
  - The size of the volume that is being reverted is greater than 16 TB.

- The Snapshot copy that the volume is being reverted to was contained in a 32-bit aggregate.

Attention: SnapRestore does not allow reversion to a Snapshot copy that was created prior to Data ONTAP 6.5. Though you can revert to a Snapshot copy from a previous Data ONTAP release later than Data ONTAP 6.5, this can cause problems because of potential version incompatibilities and can prevent the system from booting completely.

Prerequisites for using SnapRestore

You must meet certain prerequisites before using SnapRestore.

- SnapRestore must be licensed on your storage system.
- There must be at least one Snapshot copy on the system that you can select to revert.
- The volume to be reverted must be online.
- The volume to be reverted must not be in use for data replication.

General cautions for using SnapRestore

You must be aware of certain cautions before you start using SnapRestore.

- SnapRestore overwrites data permanently and might disrupt a SnapMirror relationship.
- SnapRestore overwrites all data in the file or volume.
  After you use SnapRestore to revert to a selected Snapshot copy, you cannot undo the reversion.
- If you revert to a Snapshot copy created before a SnapMirror Snapshot copy, Data ONTAP can no longer perform an incremental update of the data using the `snapmirror update` command.
  However, if there is any common Snapshot copy (SnapMirror Snapshot copy or other Snapshot copy) between the SnapMirror source and SnapMirror destination, then you can use the `snapmirror resync` command to resynchronize the SnapMirror relationship.
  If there is no common Snapshot copy between the SnapMirror source and SnapMirror destination, you should reinitialize the SnapMirror relationship.
- If you revert to a selected Snapshot copy after a SnapMirror relationship is broken, Data ONTAP deletes all the Snapshot copies that were taken after the selected Snapshot copy from the destination.
  If one of the deleted Snapshot copies is the SnapMirror created Snapshot copy, then the information about the SnapMirror relationship is lost. If the information lost is from the destination volume, then you must edit the `snapmirror.conf` file and add an entry about the SnapMirror relationship to ensure that an instance of this information is available. You can use the `snapmirror status` command to view the SnapMirror relationship information.
- Snapshot copy deletions are irrevocable.
  If you delete a Snapshot copy, you cannot recover the Snapshot copy by using SnapRestore.

- After you revert a volume to a selected Snapshot copy, you lose all the Snapshot copies that were taken after the selected Snapshot copy.
- Between the time you enter the `snap restore` command and the time when reversion is complete, Data ONTAP stops deleting and creating Snapshot copies.
- If you are reverting a file from a Snapshot copy, you can delete other Snapshot copies, except for the Snapshot copy you are reverting from.
- If you use the SnapRestore feature to restore a FlexClone from its base Snapshot copy, the space optimization relationship between the FlexClone and its parent is lost.
  For more information, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Caution about reverting the root volume

Because the `/etc` directory of the root volume contains configuration information about the system, reverting the root volume might change the configuration.

In addition, restoring the root volume restores the options for the entire system to the settings that were in effect when the Snapshot copy was taken. Reverting a root volume requires rebooting the system.

Preserving configuration files

To preserve the data, you must store all configuration file data in a volume other than a root volume.

Step

1. Store all data that needs to be reverted in a volume other than the root volume.

   This ensures that you never need to revert the root volume.

Reverting a root volume before using SnapRestore

If the data you want to revert resides in the root volume, you should back up the `/etc` directory to another volume or another system before using SnapRestore. After you revert the root volume, you can then restore the `/etc` directory and reboot.

Step

1. If you back up the `/etc` directory to another volume, use the following command to make the system reboot with that volume as the root volume:

   `vol options` *`volume`* `root`

   In this way, when the system reboots during a revert, it can use the correct settings in the `/etc` directory.

Installing the SnapRestore license

You must purchase and install the license key before you can use SnapRestore.

Step

1. On the server, enter the following command:

   `license add` ***`xxxxxxx`***

   xxxxxxx is the license key you purchased.

   This setting persists across reboots.

Reverting a volume to a selected Snapshot copy

In certain situations, you need to revert a volume to a selected Snapshot copy using SnapRestore. You should notify the users of the volume before you revert that volume.

Before you begin

Ensure that you notify the users of the volume that you are going to revert a volume, and that the current data in the volume will be replaced by the selected Snapshot copy.

Note: NFS users should unmount the files and directories in the volume before the reversion. If they do not unmount the files and directories, they might get a "`stale file handle`" error message after the volume reversion.

About this task

When you revert a volume using SnapRestore, the `maxdirsize` option for the volume is also reverted as per the Snapshot copy used for restoration. You can view the value of the `maxdirsize` option for a volume by using the `vol options` command.

Steps

1. As required, choose one of the actions from the following table:

If...	Then...
You know the name of the Snapshot copy for each volume you want to revert	Go to Step 5.

If...	Then...
You want to choose a Snapshot copy from the list of Snapshot copies available for reversion	Go to Step 2.

2. Enter the following command:

`snap restore [-f] -t vol` *`vol_name`*

`-t vol` specifies the volume name to revert.

`vol_name` is the name of the volume to be reverted. Enter the name only, not the complete path. You can enter only one volume name.

Use the `-f` option to avoid warning messages and prompts to confirm your decision to revert the volume. For more information, see the na_snap(1) man page.

3. Press `y` to confirm that you want to revert the volume.

Data ONTAP displays a list of Snapshot copies.

4. Enter the name of the Snapshot copy for reverting the volume, then go to Step 8.

Data ONTAP displays the name of the volume to be reverted and the name of the Snapshot copy to be used for the reversion.

5. Enter the following command:

`snap restore [-f] -t vol -s` *`snapshot_name vol_name`*

`-t vol` specifies the volume name to revert.

`-s` *`snapshot_name`* specifies the name of the Snapshot copy from which to revert the data. You can enter only one Snapshot copy name.

6. Press `y` to confirm that you want to revert the volume.

Data ONTAP displays the name of the volume and the name of the Snapshot copy for the reversion and, if you have not used the `-f` option, prompts you to decide whether to proceed with the reversion.

Note: To cancel volume reversion, press Ctrl-C at any time before you enter `y` in Step 8.

7. As required, choose one of the actions from the following table:

If...	Then...
You want to continue with the reversion	Press `y`. **Result**: The system reverts the volume from the selected Snapshot copy. If you are reverting the root volume, the system reboots.

If...	Then...
You do not want to proceed with the reversion	Press `n` or `Ctrl-C`. **Result**: The volume is not reverted and you are returned to a prompt.

Example

```
system> snap restore -t vol -s nightly.0 /vol/vol1
system> WARNING! This will restore a volume from a snapshot into
the active file system. If the volume already exists in the active
file system, it will be overwritten with the contents from the
snapshot.
Are you sure you want to do this? y
You have selected file /vol/vol1, snapshot nightly.0
Proceed with restore? y
```

Result: Data ONTAP restores the volume called `vol1` at `/vol/vol1`.

After a volume is reverted with SnapRestore, all user-visible information (data and attributes) for that volume in the active file system is identical to that contained in the Snapshot copy.

Reverting a file to a selected Snapshot copy

Using `snap restore` to revert a single file to a selected Snapshot copy is practical when the file is so large that you cannot copy the previous file version from the Snapshot copy to the active file system.

Before you begin

Ensure that you notify the network users before reverting a file so that they know that the current data in the file will be replaced by that of the selected Snapshot copy.

Note: NFS users who try to access a reverted file without first reopening it might get a `stale file handle` error message after the volume reversion.

About this task

When you use `snap restore` for file reversion, note the following information:

- You cannot use SnapRestore to revert a single file if that file is a directory or a symbolic link, or contains NT streams.
- If you restore single files with the `snap restore` command, and then issue the `snap list` command, the `snap list command` might take up to several minutes to complete. You can minimize the amount of time required to complete by using the `snap list -n` command. For more details, see the man pages.

- You can use the Single File Snap Restore (SFSR) to restore the NFSv4 security descriptors (ACLs) and NT ACLs of the source file in the Snapshot copy to the destination target.

Steps

1. As required, choose one of the actions from the following table:

If...	Then...
You know the name of the Snapshot copy for the file you want to revert	Go to Step 5.
You want to choose a Snapshot copy from the list of Snapshot copies available for reversion	Go to Step 2.

2. Enter the following command:

   `snap restore [-f] -t file -r` *`restore_as_new_path path_and_file_name`*

   `-t file` specifies that you are entering the name of a file to revert.

   `-r` *`restore_as_new_path`* restores the file to a location different from (but in the same volume as) the location in the Snapshot copy. For example, if you specify `/vol/vol0/vol3/myfile` as the argument to `-r`, SnapRestore reverts the file called `myfile` to the location `/vol/vol0/vol3` instead of to the path in vol3 indicated by *`path_and_file_name`*.

   `path_and_file_name` is the complete path to the name of the file to be reverted. You can enter only one path name.

   You can restore a file only to the volume where it was originally located. The directory structure to which a file is to be restored must be the same as that specified in the path. If this directory structure does not exist, you must create it before restoring the file.

   Note: Use the `-f` option to avoid warning messages and prompts to confirm your decision to revert the volume. For more information, see the na_snap(1) man page.

   Data ONTAP displays a warning message and prompts you to confirm your decision to revert the file.

3. Press `y` to confirm that you want to revert the file.

   Data ONTAP displays a list of Snapshot copies.

4. Enter the name of the Snapshot copy for reverting the file, then go to Step 8.

   Data displays the name of the file to revert and the name of the Snapshot copy to be used for the reversion.

5. Enter the following command:

snap restore [-f] -t file -s *snapshot_name* -r *restore_as_path* *path_and_file_name*

`-t` file specifies that you are entering the name of a file to revert.

`-s` *`snapshot_name`* specifies the name of the Snapshot copy from which to revert the data.

`-r` *`restore_as_path`* restores the file to a location different from the location in the Snapshot copy. For example, if you specify `/vol/vol0/vol3/myfile` as the argument to `-r`, SnapRestore reverts the file called myfile to the location `/vol/vol0/vol3` instead of to the file structure indicated by the path in *`path_and_file_name`*.

`path_and_file_name` is the complete path to the name of the file to be reverted. You can enter only one path name.

You can restore a file only to the volume where it was located originally. The directory structure to which a file is to be restored must be the same as specified in the path. If this directory structure does not exist, you must create it before restoring the file.

Unless you enter `-r` and a path name, only the file at the end of the *`path_and_file_name`* is reverted. You can enter only one path name.

Note: Use the `-f` option to avoid warning messages and prompts that confirm your decision to revert the file. For more information, see the na_snap(1) man page.

6. Press `y` to confirm that you want to revert the file.

Data ONTAP displays the name of the file and the name of the Snapshot copy for the reversion and, if you have not used the `-f` option, prompts you to decide whether to proceed with the reversion.

7. As required, choose one of the actions from the following table:

If...	Then...
You want to continue with the reversion	Press `y`. **Result**: The system reverts the file from the selected Snapshot copy.
You do not want to continue with the reversion and want to choose another Snapshot copy from the list of Snapshot copies available for reversion	Press `n` or `Ctrl-C`. **Result**: The file is not reverted and you are returned to a prompt.

Example

```
system> snap restore -t file /vol/vol1/users/jim/myfile -s nightly.0
system> WARNING! This will restore a file from a snapshot into the active
file system. If the file already exists in the active file system, it will
be overwritten with the contents from the snapshot. Are you sure you want to
do this? y
You have selected file /vol/vol1/users/jim/myfile, snapshot nightly.0
Proceed with restore? y
```

Result: Data ONTAP restores the file called `myfile` to the existing volume and directory structure `/vol/vol1/users/jim`.

```
system>snap restore -t file -s nightly.0 -r /vol/vol2/archive/eng/
myfile /vol/vol2/users/jim/myfile

system>WARNING! This will restore a file from a snapshot into the active
file system. If the file already exists in the active file system, it will
be overwritten with the contents from the snapshot.
Are you sure you want to do this? y
You have selected file /vol/vol1/users/jim/myfile, snapshot nightly.0
Proceed with restore? y
```

Result: Data ONTAP restores the file called `myfile` to a new location at `/vol/vol2/archive/eng`.

After a file has been reverted with SnapRestore, check whether all user-visible information (data and file attributes) for that file in the active file system is identical to that contained in the Snapshot copy.

Obtaining correct incremental backups after reversion

All files in a reverted volume have timestamps that are the same as those when the Snapshot copy was created. After a revert operation, incremental backup and restore operations on the file or volume cannot rely on the timestamps to determine what data needs to be backed up or restored.

Steps

1. Perform a base-level backup of the volume after you restore it.
2. When restoring data from tape, use only the backups that were created after the volume was restored.

Data protection using SnapMirror

SnapMirror is a feature of Data ONTAP that enables you to replicate data. SnapMirror enables you to replicate data from specified source volumes or qtrees to specified destination volumes or qtrees, respectively. You need a separate license to use SnapMirror.

You can use SnapMirror to replicate data within the same storage system or with different storage systems.

After the data is replicated to the destination storage system, you can access the data on the destination to perform the following actions:

- You can provide users immediate access to mirrored data in case the source goes down.
- You can restore the data to the source to recover from disaster, data corruption (qtrees only), or user error.
- You can archive the data to tape.
- You can balance resource loads.
- You can back up or distribute the data to remote sites.

You can configure SnapMirror to operate in one of the following modes:

- Asynchronous mode: SnapMirror replicates Snapshot copies to the destination at specified, regular intervals.
- Synchronous mode: SnapMirror replicates data to the destination as soon as the data is written to the source volume.
- Semi-synchronous mode: SnapMirror replication at the destination volume lags behind the source volume by 10 seconds. This mode is useful for balancing the need for synchronous mirroring with the performance benefit of asynchronous mirroring.

SnapMirror can be used with traditional volumes and FlexVol volumes.

Note: SnapMirror is supported for V-Series systems also. Information about SnapMirror in this chapter applies to both FAS systems and V-Series systems, unless specified otherwise.

Related concepts

What synchronous SnapMirror is on page 78
SnapMirror over Fibre Channel on page 207

How SnapMirror works

SnapMirror replicates data from a source volume or qtree to a partner destination volume or qtree, respectively, by using Snapshot copies. Before using SnapMirror to copy data, you need to establish a relationship between the source and the destination.

You can specify a SnapMirror source and destination relationship between volumes or qtrees by using one of the following options:

- The `/etc/snapmirror.conf` file
- The `snapmirror.access` option
- The `/etc/snapmirror.allow` file

The SnapMirror feature performs the following operations:

1. Creates a Snapshot copy of the data on the source volume.
2. Copies it to the destination, which can be a read-only volume or qtree.
3. Updates the destination to reflect incremental changes on the source, as per the schedule you specify.

The result of this process is an online, read-only volume or qtree that contains the same data as the source at the time of the most recent update.

Each of the following replication methods consists of a pair of operations, one operation each at the source storage system and the destination storage system:

- Volume SnapMirror replication
- Qtree SnapMirror replication
- SnapVault replication

If a storage system is the source for one replication and the destination for another replication, it uses two replication operations. Similarly, if a storage system is the source as well as the destination for the same replication, it uses two replication operations.

Applications of SnapMirror

SnapMirror is used to replicate data. Its qualities make SnapMirror useful in several scenarios, including disaster recovery, data backup, and data restoration.

You can copy or use the data stored on a SnapMirror destination. The additional advantages of SnapMirror make it useful in data retrieval situations such as those described in the following table:

Situation	How to use SnapMirror
Disaster recovery: You want to provide immediate access to data after a disaster has made a qtree, volume, or system unavailable.	You can make the destination writable so clients can use the same data that was on the source volume the last time data was copied.
Disaster recovery testing: You want to test the recovery of data and restoration of services in the event of a disaster.	You can use FlexClone technology on the SnapMirror destination, and test for disaster recovery, without stopping or pausing other replication operations.
Data restoration: You want to restore lost data on a qtree or volume source from its mirrored qtree or volume SnapMirror partner.	You can temporarily reverse the roles for the source and destination qtrees or volumes and copy the mirrored information back to its source.
Application testing: You want to use an application on a database, but you want to test it on a copy of the database in case the application damages the data.	You can make a copy of the database to be used in the application testing to ensure that the data on the source cannot be lost.
Load balancing: A large number of users need read-only access to a qtree or volume.	You can copy the data in a qtree or volume to multiple volumes or systems to distribute the load.
Off-loading tape backups: You need to reserve all processing and networking resources on a system for serving NFS and CIFS requests.	After copying data on the source system, you can back up the data in the destination to tape. This means that the source system does not have to allocate resources for performing backups.
Access to remote data: Users who need read access to a volume are distributed over a large geographical area.	You can copy the source volume to other systems that are geographically closer to the users. Users accessing a local system can read the data using less resource time than if they connected to a distant system.

What synchronous SnapMirror is

In the synchronous mode, SnapMirror enables you to replicate data to the destination as soon as it is written to the source volume.

Synchronous SnapMirror is a feature of SnapMirror. You can use synchronous SnapMirror to replicate data between systems situated at remote sites, using either an IP or a Fibre Channel connection.

You can use synchronous SnapMirror only with volumes, not with qtrees. The source and destination volumes must be of the same type: traditional volumes or FlexVol volumes.

Synchronous SnapMirror modes

There are two modes available for synchronous SnapMirror replication: `sync` and `semi-sync`. The `semi-sync` mode helps in achieving a balance between the benefits of synchronous and asynchronous replication.

Note: You cannot set up a synchronous or semi-synchronous SnapMirror relationship between the two nodes of an HA pair.

You can specify either of the following two modes, when defining a SnapMirror relationship in the `snapmirror.conf` file:

- `sync`: The source system acknowledges a client write operation only after both the source and destination systems have completed the write operation. The `sync` option provides a recovery point objective of 0 seconds.
- `semi-sync`: The source system acknowledges the client write operation immediately after the source receives the data. The destination system is synchronized with the source at intervals of approximately 10 seconds. The `semi-sync` option provides a recovery point objective of about 10 seconds. This means that if the source becomes unavailable, you might lose up to 10 seconds worth of data changes. The `semi-sync` mode provides a performance advantage over the `sync` mode.

Note: If neither of these two modes is specified, then the SnapMirror relationship is set as asynchronous.

To enable the `sync` or `semi-sync` mode for a volume SnapMirror relationship, you need to specify the mode in the `snapmirror.conf` file entry for the relationship, as given in the following line:

***src_system*:*src_path* *dst_system*:*dst_path* - {sync|semi-sync}**

src_system is the name of the SnapMirror source system.

dst_system is the name of the SnapMirror destination system.

src_path is the path of the SnapMirror source volume.

dst_path is the path of the SnapMirror destination volume.

Example of an entry for a SnapMirror relationship with semi-sync mode

```
systemA:volA systemB:volB - semi-sync
```

Related references

Syntax for snapmirror.conf file entries on page 131

How SnapMirror replicates data synchronously

Before Data ONTAP saves data to the disk, it collects written data in NVRAM. Then, at a point in time called a consistency point, it sends the data to disk.

As data is added or changed, the data is not directly written to the disk. Changes or additions to data are temporarily stored in the NVRAM. Then, at a consistency point, SnapMirror writes the data to the disks on the destination system. When the synchronous SnapMirror feature is enabled, the source system forwards data to the destination system as it is written in NVRAM. Then, at the consistency point, the source system sends its data to disk and tells the destination system to also send its data to disk. Finally, the source system waits for the destination system to acknowledge that it sent data to disk before continuing with the next write.

How synchronous SnapMirror handles network issues

If SnapMirror encounters any network issues that restrict the operation of synchronous replication, SnapMirror goes into the asynchronous mode.

The source and destination systems communicate with each other continuously. If a network failure disrupts the communication, SnapMirror initiates the following process:

1. SnapMirror sets the replication to the asynchronous mode.
2. In the asynchronous mode, the source system tries to communicate with the destination system once a minute.
3. When the source system reestablishes communication with the destination system, the source system asynchronously replicates data to the destination.
4. SnapMirror gradually transitions the replication relationship to the synchronous mode.

If the latest common Snapshot copy is deleted from the source, SnapMirror does not transition back from asynchronous to synchronous mode. The SnapMirror relationship should be broken and resynchronized, by using the `snapmirror break` and `snapmirror resync` commands. However, you can avoid this situation and let the relationship go to synchronous mode automatically, by setting the `replication.volume.use_auto_resync` option to `on`. The default value of this option is `off`.

Related references

SnapMirror options on page 99

Guidelines for growing an aggregate with a synchronous SnapMirror destination volume

When increasing the size of an aggregate that contains a synchronous SnapMirror destination volume, you need to follow several guidelines.

- Add a minimum of four disks.
- Ensure that any new RAID group created by the addition of new disks has at least four data disks.
- Ensure that the RAID group size is 16 or fewer disks.

Enabling SnapMirror by entering license keys

Before any SnapMirror replication process can begin, you must add the SnapMirror license on the system and enable SnapMirror.

Step

1. To add the SnapMirror license key, enter the following command:

   `license add` ***`xxxxxxx`***

   xxxxxxx is the license key you purchased.

Related concepts

What synchronous SnapMirror is on page 78

Turning SnapMirror on

Before using SnapMirror you need to enable the SnapMirror license on both the source and the destination systems.

Steps

1. To turn SnapMirror on, enter the following command on both the source system and destination system:

   `options snapmirror.enable on`

   Alternatively, you can use the `snapmirror on` command to turn SnapMirror on.

   Note: This setting persists across reboots.

2. Depending on whether you use the `snapmirror.access` option or the `/etc/snapmirror.allow` file to specify allowed destinations, choose one of the actions from the following table:

If you choose...	Then...
`snapmirror.access` option	On the source, enter the following command as a single line: **`options snapmirror.access host=[dest_system1,dest_ system2,...]`** The default value for the `snapmirror.access` option is `legacy`, which lets the `/etc/snapmirror.allow` file define the access permissions. This option persists across reboots.
`/etc/snapmirror.allow` file	Add the names of the destination systems, each on a separate line, in the `/etc/snapmirror.allow` file.

Considerations for the use of SnapMirror

When planning to use SnapMirror for replication, you need to be aware of the prerequisites, restrictions, points of caution, and recommended actions.

You also need to understand issues related to deduplication, adjusting the TCP window size for SnapMirror, possible conflicts between SnapMirror and Snapshot copy schedules, and destination accessibility when using CIFS with SnapMirror.

Prerequisites for SnapMirror

You need to fulfill a set of prerequisites before you can use SnapMirror.

- You must purchase and enable the SnapMirror license. If the SnapMirror source and destination are on different systems, you must enable the SnapMirror license on each system.
- For SnapMirror volume replication, you must create a restricted volume to be used as the destination volume.

  SnapMirror does not automatically create a volume. For information about how to create volumes, see the section on organizing data using volumes and qtrees in the *Data ONTAP Storage Management Guide for 7-Mode*.
- For SnapMirror volume replication, the destination system must use a version of Data ONTAP that is the same as or later than that of the SnapMirror source system.

  Note: Depending on the Data ONTAP version of the destination, the source can run any version from the previous two Data ONTAP release families. For example, if the destination is Data ONTAP 8.2, the source can be either version in Data ONTAP 8.1 or Data ONTAP 8.0 release families.

  If you configure volume SnapMirror to support replication for the purpose of disaster recovery, both the source and destination system must use the same version of Data ONTAP. The following table describes how SnapMirror volume replication is dependent on Data ONTAP versions of the source and destination systems:

Volume SnapMirror Source	Volume SnapMirror Destination	Replication (Yes or No)
Data ONTAP 7.3	Data ONTAP 8.1	Yes
Data ONTAP 8.0	Data ONTAP 8.1	Yes
Data ONTAP 8.1	Data ONTAP 7.2	No
Data ONTAP 8.1	Data ONTAP 7.3	No
Data ONTAP 8.1	Data ONTAP 8.0	No
Data ONTAP 8.1	Data ONTAP 8.1	Yes

Note: If you upgrade your system to a later version of Data ONTAP, you must upgrade the SnapMirror destination before upgrading the SnapMirror source.

- For SnapMirror qtree replication, you must not create a qtree to be used as a destination qtree; the `snapmirror initialize` command creates the destination qtree automatically.
- For SnapMirror qtree replication, the destination system must be using Data ONTAP 6.2 or later.
- The name and IP address of the source system must be in the `/etc/hosts` file of the destination system or must be resolvable through the DNS or by using the `yp` command.

Related tasks

Enabling SnapMirror by entering license keys on page 81

Volume SnapMirror interoperability matrix

Volume SnapMirror supports interoperability between volumes in 32-bit and 64-bit aggregates for different Data ONTAP releases. For systems in a volume SnapMirror relationship, the format of a volume can be different from that of its containing aggregate. When the SnapMirror relationship is initialized or updated, the format of the destination volume changes to match the format of the source volume.

The following interoperability matrix shows the different combinations of volume types for which a volume SnapMirror relationship is supported across Data ONTAP releases:

		Destination volume type						
		Data ONTAP 7.3.x	Data ONTAP 8.0.x		Data ONTAP 8.1.x		Data ONTAP 8.2.x	
	Aggregate type	32-bit	32-bit	64-bit	32-bit	64-bit	32-bit	64-bit

Source volume type	**Data ONTAP 7.3.x**	32-bit	Yes	Yes[1]	No	Yes[1]	Yes[1]	No[1,4]	No[1,4]
	Data ONTAP 8.0.x	32-bit	No	Yes	No	Yes[1]	Yes[1]	Yes[1]	Yes[1]
		64-bit	No	No	Yes	Yes[1,2,3]	Yes[1]	Yes[1,3]	Yes[1]
	Data ONTAP 8.1.x	32-bit	No	No	No	Yes	Yes	Yes[1]	Yes[1]
		64-bit	No	No	No	Yes[2,3]	Yes	Yes[1,2,3]	Yes[1]
	Data ONTAP 8.2.x	32-bit	No	No	No	No	No	Yes	Yes
		64-bit	No	No	No	No	No	Yes[2,3]	Yes

[1] After a volume SnapMirror relationship is broken, resynchronization to the original source volume does not work until the Data ONTAP version of the source system is upgraded to match the version of the destination system.

[2] Do not grow the source volume beyond 16 TB until the 64-bit expansion process is complete on the aggregate and the destination volume grows beyond 16 TB. The 64-bit expansion process expands a 32-bit aggregate and all its containing volumes to 64 bits, when new disks are added, such that the total size of the aggregate expands beyond 16 TB.

[3] A 64-bit source volume that has data compression enabled cannot be mirrored to a 32-bit aggregate.

[4] Volume SnapMirror is supported only from a Data ONTAP 7.3.7 source to a Data ONTAP 8.2 destination.

You can use interoperability support in situations such as the following:

- Migrating data between volumes in 32-bit and 64-bit aggregates
  For example, you can migrate data from a 32-bit source volume in a 32-bit aggregate to a 64-bit destination volume in a 64-bit aggregate, with the source system running Data ONTAP 8.0 and the destination system running Data ONTAP 8.1.
- Expanding 32-bit aggregates to 64-bit
  For example, if the source and the destination systems in a volume SnapMirror relationship contain 32-bit volumes in 32-bit aggregates, you can expand aggregates and volumes on both the systems to 64-bits.

For more information about interoperability between 32-bit and 64-bit volumes, see the technical report *SnapMirror Async Overview and Best Practices Guide*.

Related concepts

Interoperability between volumes in 32-bit and 64-bit aggregates on page 25

Related information

SnapMirror Async Overview and Best Practices Guide: media.netapp.com/documents/tr-3446.pdf

Restrictions on using SnapMirror

When planning the configuration of SnapMirror, you need to consider the relevant restrictions.

- The source volume must be online.
  For information about how to put a volume online, see the *Data ONTAP Storage Management Guide for 7-Mode*.
- For SnapMirror volume replication, the capacity of the destination volume must be greater than or equal to the capacity of the source volume.
  For information about how to add disks to a volume, see the *Data ONTAP Storage Management Guide for 7-Mode*.
- For SnapMirror volume replication, if you create a large file after the last volume SnapMirror update, the file is deleted before the next update.
  The time taken to delete the file depends on its size. Before all the blocks are freed, if you attempt volume SnapMirror update the blocks that are not freed are also transferred. This is because those blocks are captured in the volume SnapMirror created Snapshot copy.
- To support SnapMirror qtree replication, the destination volume must contain 5 percent more free space than the source qtree consumes.
- For SnapMirror qtree replication, the deleted file does not involve transferring of deleted block data to the destination.
- The SnapMirror destination volume cannot be the root volume of a storage system.
  The SnapMirror source volume, however, can be the root volume.
- A destination qtree can be on the root volume, but the `/etc` qtree cannot be a destination qtree.
- A destination qtree name should not:
  - Contain “/etc”, “*”, “-”, “.”
  - Contain the character combination “->”
    This restriction applies to source qtrees as well.
  - Contain the tab character
  - Be longer than 64 characters
  - Be specified as “`/vol/vol_name/`” (with no qtree name)
  - Be specified as “`vol_name/qtree_name`” (without `/vol/`)

    Note: When creating or specifying a qtree, you can use the space character in the qtree name. However, if you do so, you need to enclose the qtree name in double quotes. You can also use a

double quote within a qtree name. The following example shows how to specify the qtree name, `vol/vol1/x y"z`, with both a space character and a double quote.

```
"systemA:/vol/vol1/x y""z"
```

- There must be a functional network to transfer data between two different storage systems.
- Each storage system model supports a specified number of `snapmirror update` command or `vol copy` command operations at a time.
- When you convert an aggregate that contains a volume SnapMirror destination into a Flash Pool, the destination volumes will not be cached.

  Note: If you create a Flash Pool with the same name as the destination volume of a SnapMirror relation, the subsequent SnapMirror updates fail.

Related concepts

Initialization of a SnapMirror destination on page 116
Initialization of a SnapMirror destination from tape on page 117

Related tasks

Verifying the size of each volume on page 303
Initializing a destination for non-qtree data on page 119
Initializing a SnapMirror destination on page 117

Related references

Maximum number of concurrent replication operations on page 123

Points of caution while using SnapMirror

While using SnapMirror, you need to exercise adequate caution on several points, including the use of Snapshot copies, the use of SnapMirror commands, and the use of SnapMirror destination volumes.

- Do not delete Snapshot copies that SnapMirror creates in the source volume before copying the data to the destination. The most recent SnapMirror Snapshot copy is referred to as the newest common Snapshot copy (NCS). Incremental changes to the destination depend on the NCS. If SnapMirror cannot find the required Snapshot copy on the source, it cannot perform incremental changes to the destination.
- Do not use the `snapmirror release` or `snapmirror break` command on the destination volume or qtree unless you no longer need to copy incremental changes from the source. The destination must be actively functioning as a destination to receive incremental updates.
- Do not restrict or take the destination volume offline while SnapMirror is configured to transfer. Taking the destination offline prevents SnapMirror from performing updates to the destination.

Note:

- If Flash Cache is used in SnapMirror source and destination, during an application failover at the disaster recovery site, then there will be latency during the read operations on the destination as compared to the source, even though the destination system has the same Flash configuration as the source.
- If Flash Cache is deployed with synchronous SnapMirror, then there might be latency during the read operations on the destination storage system.

Symmetrical disk geometry

When replicating data with FlexVol volumes, disk geometry is not an issue, unlike replicating data with traditional volumes.

With FlexVol volumes, the symmetry of disk geometry between the source and the destination volumes is less important. It is not necessary that the source and destination volumes have the same number of disks or have disks with identical sizes.

A qtree SnapMirror or SnapVault transfer might fail when the newer version of Data ONTAP on the source system has more number of subdirectories than the supported limit of subdirectories on the destination system running an older version of Data ONTAP.

Related references

Recommended actions while using SnapMirror on page 87

Recommended actions while using SnapMirror

While using SnapMirror, you can increase the efficiency of data copying by performing certain actions. This includes the staggering of Snapshot copy schedules and SnapMirror update schedules.

- To optimize performance, stagger your Snapshot copy update schedules so that SnapMirror activity does not begin or end at the exact minute a `snap sched` command operation attempts to create a Snapshot copy.

  If the SnapMirror feature is scheduled to perform Snapshot copy management at the same time as a `snap sched` activity, then the Snapshot copy management operations scheduled using the `snap sched` command might fail with syslog messages: `"Skipping creation of hourly snapshot"` and `"Snapshot already exists."`
- For optimum SnapMirror volume replication performance, ensure that the SnapMirror source volume and destination volume contain disks of the same size, organized in the same RAID configuration.
  - If the SnapMirror source and destination are FlexVol volumes, the RAID configurations do not make a difference.

- If the SnapMirror source and destination are qtrees, volume size and configuration do not make any difference.

Related references

Firewall usage with SnapMirror on page 106

Deduplication with volume SnapMirror

Starting with Data ONTAP 8.1, two copies of the deduplication metadata are maintained for a FlexVol volume. One copy of the metadata resides in the volume while the other copy resides in the aggregate.

When replicating data using volume SnapMirror, the deduplication metadata for the volume is replicated along with the volume. The data in the volume is usable both on the source and the destination.

Note: When configuring volume SnapMirror and deduplication, you should ensure that deduplication and volume SnapMirror operations do not run at the same time. You should start volume SnapMirror transfer of a deduplicated volume after the deduplication operation is complete. This prevents any impact to the replication performance while deduplication is in progress and sending of undeduplicated data and additional temporary deduplication metadata files over the network.

To achieve maximum space savings on the destination volume, you must scan the entire file system to re-create the deduplication metadata for the destination volume. Use the `sis start -s` command to do so.

Note: The destination volume is accessible for read-write operations when the deduplication scan is in progress.

If you use the `sis start` command without the `-s` option, the potential for space savings on the destination volume is reduced because only the new data written to the volume is scanned for deduplication.

For more information about deduplication, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Related references

Considerations before using synchronous SnapMirror on page 90

Data compression with qtree SnapMirror

Because qtree SnapMirror operates at the logical level, when data compression is enabled on the source system, the data is uncompressed in memory before being replicated to the destination system.

If data compression is enabled on the secondary system, then all transfers are compressed on the secondary system.

When data compression is enabled on the source volume, no bandwidth savings are achieved over the network because the data is uncompressed on the source volume before it is sent for replication. If inline compression is enabled on the destination volume, the data is compressed inline at the destination before it is written to the disk. If inline compression is not enabled on the destination volume, you must manually compress the data after the qtree SnapMirror transfer is completed to achieve storage space savings in the destination volume.

Note: Inline compression does not guarantee compression of all the data that is being transferred using qtree SnapMirror. The space savings at the destination and the source systems are the same if inline compression is enabled on the source system.

For more information about inline compression and how data compression works with qtree SnapMirror, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Possible conflicts between SnapMirror operation and Snapshot copy schedule

Some of the operations of SnapMirror might conflict with the actions of a Snapshot copy management schedule. Certain steps enable you to avoid these conflicts.

If the SnapMirror feature is scheduled to perform Snapshot copy management at the same time as a `snap sched` activity, then the Snapshot copy management operations scheduled using the `snap sched` command might fail, generating syslog messages such as: `Skipping creation of hourly snapshot` and `Snapshot already exists`.

To avoid this situation, you should stagger the Snapshot copy update schedules so that SnapMirror activity does not begin, or end at the exact minute that a `snap sched` command operation attempts to create a Snapshot copy.

Destination accessibility when using CIFS with SnapMirror

Before copying a directory on a SnapMirror volume that supports CIFS clients, you should ensure that the directories are in the Unicode format. This ensures that the read-only directory copied on the destination is in the Unicode format. This also enables requests through CIFS to access the directory and its files on the destination, and prevents `Access denied` errors.

You can ensure that both source volume and destination volume directories are in the Unicode format by using one of the following methods:

Method 1

On the system console for the source volume, enter these two commands.

- `vol options` *`vol_name`* `convert_ucode on`

  Use this command to convert any existing directories in a volume to the Unicode format.

- `vol options` *`vol_name`* `create_ucode on`

  Use this command to ensure that any new directories created in a volume are in the Unicode format.

Method 2

Alternatively, ensure that all directories on the source volume that will be accessed by CIFS clients are accessed by a CIFS client before initial replication to a destination. Such access on a writable source volume automatically converts that directory to the Unicode format.

Considerations before using synchronous SnapMirror

You need to consider certain issues when planning to use SnapMirror for synchronous replication.

- One source volume cannot have synchronous SnapMirror relationships to multiple destinations volumes.
- You cannot create a synchronous SnapMirror relationship between FlexVol volumes within the same system or within the same HA pair.
- You must ensure that the source and destination have the same version of Data ONTAP installed.
- You cannot use synchronous or semi-synchronous SnapMirror to replicate volumes that use deduplication.
- A volume with a synchronous or a semi-synchronous SnapMirror relationship should not be placed in the same aggregate as a volume using deduplication.

For more information about deduplication, see the *Data ONTAP Storage Management Guide for 7-Mode*.

For synchronous SnapMirror, the disks underlying the source and destination volumes should be of the same type. Also, you must ensure that the speed of the disks on the destination system is equal to, or faster than, the speed of the disks on the source system. You can have other types and speeds of disks attached to the source or destination system but the combined load of different disk types and speeds might negatively impact the performance of synchronous SnapMirror. If you face such performance issues, you might need to reduce the load on the system to resolve the issue.

The following table indicates the support for bidirectional synchronous and semi-synchronous SnapMirror replication with the two different types of volumes:

Type of volume	Data ONTAP version
Traditional volumes	7.2 or later
FlexVol volumes	7.2.2 or later

You can choose the data to be synchronously replicated. For example, you can synchronously replicate database data and asynchronously replicate home directories. If the home directories contain important log data, you can use the synchronous SnapMirror option to replicate the log data. For more details, see the na_snapmirror.conf(5) man page.

The source and destination systems should be adequately configured for the replication traffic. Synchronous SnapMirror is supported for traditional volumes only for configurations in which the source and destination systems are of the same type and have the same disk geometry.

Note: There is no disk geometry restriction for FlexVol volumes.

The type of system and the disk configuration on the destination system affects the performance of the source system. Therefore, the destination system should have the bandwidth required for the increased traffic and for message logging. The NVLOG files are stored in the parent aggregate of the volume being replicated.

The network transport should be optimized for SnapMirror replication. You must use a dedicated, high-bandwidth, low-latency network between the source and destination systems. Synchronous SnapMirror can support traffic over Fibre Channel and IP transports. SnapMirror also allows multipathing, enabling you to either balance the load between two paths or to reserve the second path for failover. For optimizing performance, you can use the best route available between the source and destination systems, and you can restrict the route to the traffic between the two systems.

You should keep well below the maximum number of Snapshot copies. Synchronous SnapMirror needs three Snapshot copies to get into synchronization. Therefore, you should limit the combined total of Snapshot copies retained on any one volume to 252 or fewer.

Related concepts

What the snapmirror.conf file does on page 129
SnapMirror over multiple paths on page 144

Related references

Syntax for snapmirror.conf file entries on page 131

Disk type requirements for synchronous and semi-synchronous SnapMirror with systems that use array LUNs

The same general guidelines about disk types with synchronous and semi-synchronous SnapMirror apply to both types of Data ONTAP systems (FAS systems and V-Series systems). There are some additional guidelines for systems that use array LUNs.

Note: V-Series (“V”) systems and new FAS platforms released in Data ONTAP 8.2.1 and later can use array LUNs if the proper license is installed.

The following additional information about disk types is specific to Data ONTAP systems that use array LUNs:

- The storage array that is presenting storage to a Data ONTAP system must be using only one type of disk (FC or SATA), and the source or destination must use the same type of disks as the storage array.
- To replicate between a V-Series system and a FAS system, the controllers must be the same platform model, and the disks on the FAS system must be the same type as the disks on the storage array that provides storage for the V-Series system.

Estimating aggregate size for synchronous SnapMirror destination volumes

For synchronous replication of volumes using SnapMirror, the aggregates that contain destination volumes should have enough free space to store the NVLOG data.

Before you begin

You must have ascertained the model name of the system that is the synchronous SnapMirror source.

Steps

1. Determine the size of the source system NVRAM.

   For information, see the *Hardware Universe*.

2. Multiply the NVRAM size by 20 to determine the estimated free space size.

   #### Example

   2 GB times 20 is 40 GB; therefore, 40 GB of free space is required on the aggregate containing the destination volume.

Deployment of SnapMirror

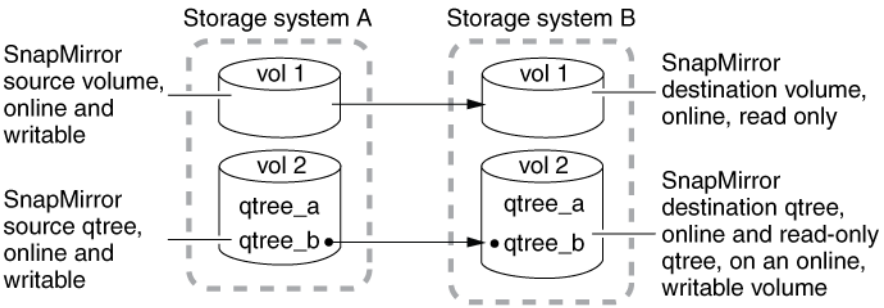
A basic deployment of SnapMirror consists of source volumes and qtrees, and destination volumes and qtrees.

Source volumes or qtrees: In a SnapMirror configuration, source volumes and qtrees are the data objects that need to be replicated. Normally, users of storage can access and write to source volumes and qtrees.

Destination volumes or qtrees: In a SnapMirror configuration, destination volumes and qtrees are data objects to which the source volumes and qtrees are replicated. The destination volumes and qtrees are read-only, and usually placed on a separate system than the source. The destination volumes and qtrees 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.

Note: Destination volumes have to be writable when using qtree SnapMirror for replication.

The following illustration depicts a basic SnapMirror deployment:



Related concepts

SnapMirror deployment variations on page 95

Supported SnapMirror configurations

You can use SnapMirror to replicate both traditional and FlexVol volumes. However, there are certain configurations that are not supported for replication.

Volume SnapMirror

Volume SnapMirror only supports replication between the same type of volumes. The source and destination volumes must be of the following types: traditional volumes or FlexVol volumes.

Qtree SnapMirror

Qtree SnapMirror supports replication between different volume types. The source and destination volumes can be any of the following types:

- Traditional volumes
- FlexVol volumes (7-Mode, 32-bit)
- FlexVol volumes (7-Mode, 64-bit)

Note: Starting with Data ONTAP 7.3.2, qtree SnapMirror does not support the use of source snapshot copies created by releases prior to Data ONTAP 6.5.

Comparison between volume SnapMirror and qtree SnapMirror

You can configure SnapMirror replication for either entire volumes or individual qtrees on a volume. You should consider the differences between the two options.

The following table describes the characteristics of SnapMirror replication:

Volume SnapMirror	Qtree SnapMirror
Synchronous or asynchronous replication is supported for volumes.	Only asynchronous replication is supported for qtrees.
Destination volume is read-only.	Destination qtree is read-only. However, the volume on which the qtree is located must be online and writable.
Source and destination volumes must both be either traditional volumes or FlexVol volumes.	Source and destination qtrees can be on any type of volumes, traditional volumes or FlexVol volumes.
Replicates Snapshot copies of a source volume and all its qtrees, to the destination volume	Replicates only the contents of an individual qtree to a destination
You need to set a destination volume to restricted, read-only status, before setting it up for replication.	The destination volume for qtree replication is writable, and must not be read-only.
Replication of a volume on the destination takes up the space allocated to the source volume, irrespective of how much of the volume is used for storing data.	If you need to mirror only the data stored on an individual qtree, then SnapMirror replication of that individual qtree uses slightly more disk space and directories on the destination qtree than the source qtree.
Replication can be set up to a destination volume from only one source volume. This implies that one destination volume cannot be used for replicating multiple source volumes.	Replication can be set up for a maximum of 255 qtrees on any one volume.

Volume SnapMirror	Qtree SnapMirror
Block-for-block replication It transfers the file system verbatim. Therefore, older releases of Data ONTAP cannot understand file system transfers from a later release of Data ONTAP.	Logical replication All the files and directories in the source file system are created in the destination file system. Therefore, you can replicate data between a storage system running an older version of Data ONTAP and a storage system running a newer version. **Note:** If the source file system contains a file type that cannot be represented on the destination file system, the replication will fail. For example, Data ONTAP 7.0 supports files up to 16 TB in size, whereas earlier versions of Data ONTAP support files up to 4 TB. If the source system is running Data ONTAP 7.0, the qtree you want to replicate contains a file greater than 4 TB, and the destination system is running an earlier version of Data ONTAP, the replication will fail.

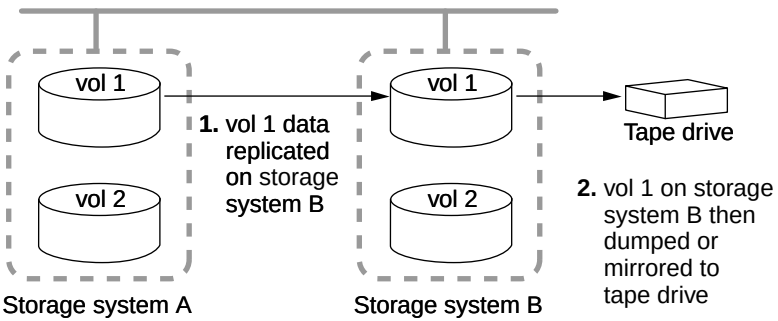
Related references

Prerequisites for SnapMirror on page 82

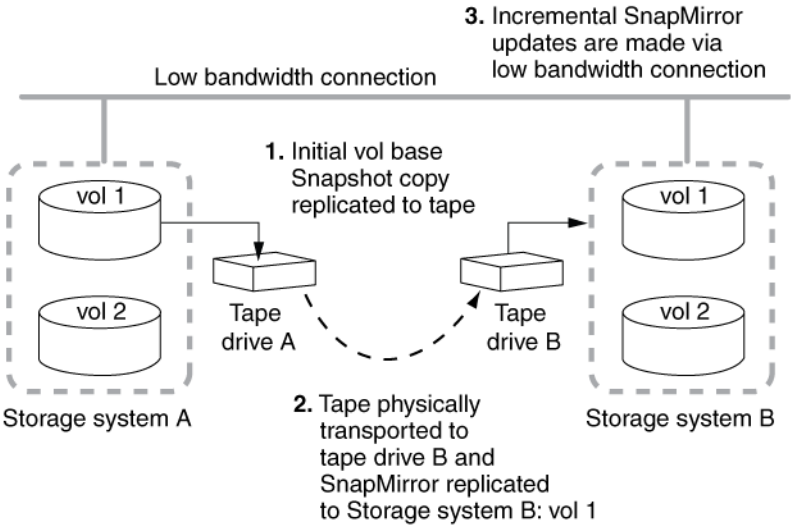
SnapMirror deployment variations

There are several variations possible while deploying SnapMirror. These variations allow you to customize the solution to suit your requirements.

Source to destination to tape variation: A common variation to the basic SnapMirror backup deployment adds a tape backup of the destination volume. By running a tape backup off the SnapMirror destination volume (as shown in the following illustration), you do not subject the heavily-accessed source volume to the performance degradation and complexity of a direct tape backup.

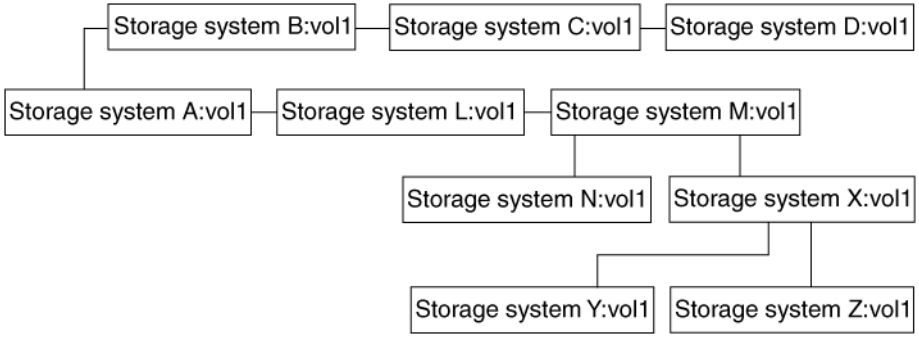


Source to tape to destination variation: A SnapMirror deployment that supports SnapMirror replication over low-bandwidth connections accommodates an initial mirroring between a source and destination volume using physically-transported tape (as shown in the following illustration). After the large base Snapshot copy has been replicated, smaller, incremental Snapshot copy updates can be carried out over a low-bandwidth connection.



Cascading destinations variation: A variation on the basic SnapMirror deployment and function involves a writable source volume replicated to multiple read-only destinations. The function of this deployment is to make a uniform set of data available on a read-only basis to users from various locations throughout a network and to allow for updating that data uniformly at regular intervals.

Note: The cascade deployment (as shown in the following illustration) is supported for volume SnapMirror only. It is not supported for qtree SnapMirror.



In a qtree SnapMirror relationship, the resync operation from the destination qtree to the source qtree fails if the source qtree is busy updating another destination qtree. For example, you have qtree SnapMirror relationships from qtree A to qtree B and from qtree A to qtree C. If you try to perform

SnapMirror resync from qtree B to qtree A when qtree A is busy updating qtree C, the resync operation fails.

Related concepts

Data replication from one destination to another in a series (cascading) on page 107

Migration from traditional volumes to FlexVol volumes

You can use only qtree SnapMirror to migrate data from traditional volumes to FlexVol volumes, if you use SnapMirror for this type of migration. You cannot use volume SnapMirror for this purpose because it cannot replicate to a different type of volume.

Related concepts

Comparison between volume SnapMirror and qtree SnapMirror on page 93

SnapMirror commands

You can use SnapMirror commands to perform different SnapMirror operations for a volume, qtree, or system.

The following table lists the commands for using SnapMirror, along with their corresponding operations:

Command	Operation
`snapmirror on`	Enable SnapMirror on the system, in order to enable the system to work as both a source and a destination for SnapMirror transfers. **Note:** Alternatively, you can use the `options snapmirror.enable on` command. **Attention:** After using the `snapmirror off` command, you should wait for at least 60 seconds before using the `snapmirror on` command. This ensures that all subsequent SnapMirror transfers work properly.

`vol create`
and
`vol restrict`
	Use these commands together to create a restricted, read-only volume, which is required as a destination for volume SnapMirror replication.
	Attention: You should not use the `vol restrict` command for a qtree SnapMirror destination volume.

`snapmirror initialize`	Start the initial, complete SnapMirror (baseline) transfer from a source volume or qtree to a destination.
`snapmirror status`	View the status of SnapMirror data transfers.

Command	Operation
`snapmirror update`	Perform a manual update of the SnapMirror destination.
`snapmirror quiesce`	Stabilize the contents of a destination before a Snapshot copy is taken, by allowing active SnapMirror transfers to finish, and temporarily preventing new transfers. This action ensures a manual Snapshot copy of a stable database.
`snapmirror resume`	Resume normal data transfer to a destination after it has been quiesced.
`snapmirror abort`	Stop an active SnapMirror transfer.
`snapmirror break`	Break the SnapMirror relationship between the source and destination, and convert the destination to a writable volume or qtree.
`snapmirror resync`	Reestablish the SnapMirror relationship between the source and a former destination volume or qtree. Use this command after the `snapmirror break` command to resynchronize the contents of the source and destination volumes or qtrees, without repeating the initial transfer.
`snapmirror release`	Release SnapMirror Snapshot copies on former source volumes or qtrees so that the Snapshot copies can be deleted.
`snapmirror off`	Turn off SnapMirror functionality for a specified system. **Note:** Alternatively, you can use the `options snapmirror.enable off` command.
`snapmirror destinations`	Set up a cascading series of SnapMirror destinations. Use this command to make a uniform set of data available on a read-only basis to users from various locations throughout a network.

For more information about SnapMirror commands, see the `na_snapmirror` man page.

Related concepts

Initialization of a SnapMirror destination on page 116
Initialization of a SnapMirror destination from tape on page 117
Manual update of a SnapMirror destination on page 141
Scheduled updates for volumes or qtrees on page 138
Conversion of a destination to a writable volume or qtree on page 168
What the quiesce command does on page 175
What the snapmirror resync command does on page 186
How the snapmirror resync command helps minimize data loss on page 190

Related tasks

Enabling SnapMirror by entering license keys on page 81
Turning SnapMirror on on page 81
Initializing a SnapMirror destination on page 117
Initializing a destination for non-qtree data on page 119
Aborting a SnapMirror transfer on page 176
Turning off SnapMirror updates on page 140
Checking SnapMirror data transfer status on page 151
Performing a manual SnapMirror update on page 141
Changing scheduled updates for one volume or qtree on page 139
Turning off scheduled updates for one volume or qtree on page 140
Converting a SnapMirror destination to a writable volume or qtree on page 169
Releasing partners from a SnapMirror relationship on page 177
Stabilizing destinations before a Snapshot copy on page 174
Resuming transfers after quiescing a destination on page 176
Resynchronizing a SnapMirror relationship on page 187
Initializing a SnapMirror destination by using tape on page 115

Related references

Methods for specifying destination systems on the SnapMirror source on page 127
What SnapMirror status check shows on page 152
Information messages in the SnapMirror status check on page 154
Quota restrictions on page 168
After using the snapmirror break command on page 169

SnapMirror options

You can use the SnapMirror options to specify different SnapMirror options for a system.

You can view the values for the different SnapMirror options by using the `options` command.

Viewing SnapMirror options

```
system_A> options snapmirror.enable
snapmirror.enable            on
```

You can use the `options snapmirror` command to view the values for the SnapMirror options.

Similarly, you can use the `options replication` command to view the values for the replication options.

The following table lists the SnapMirror options that you can use, along with their corresponding functions:

Option	Function	Default value
`snapmirror.enable {on|off}`	Specifies whether SnapMirror is enabled for the system Alternatively, you can use the `snapmirror on` and `snapmirror off` commands. **Attention:** After changing the `snapmirror.enable` option to `off`, you should wait for at leaonst 60 seconds before changing the option back to `on`. This ensures that all subsequent SnapMirror transfers work.	`off`
`snapmirror.access host=`*`list`* *`list`* is a comma-separated list of the host names of allowed systems.	Specifies the SnapMirror destinations that are allowed to copy from the system You can also use the `/etc/snapmirror.allow` file to specify the allowed destinations. However, using the `snapmirror.access` option is the preferred method. When the option is set to `legacy`, access is controlled by the `/etc/snapmirror.allow` file. **Note:** If both the `snapmirror.access` option and the `/etc/snapmirror.allow` file are used, the `snapmirror.access` option takes precedence. This can affect the initialization of SnapMirror relationships.	`legacy`

Option	Function	Default value
`snapmirror.log.enable {on|off}`	Specifies whether SnapMirror activity is logged in the `/etc/log/snapmirror.x` files. When this option is set to `on`, all the SnapMirror and SnapVault logging activity is logged in the `/vol/vol0/etc/log/snapmirror` files. When this option is set to `off`, the logging activity is logged in the respective vFiler root volume.	`on`
`replication.volume.use_auto_resync {on|off}`	Specifies automatic resynchronization for synchronous SnapMirror relationships	`off`
`replication.volume.reserved_transfers` *n*	Specifies the number of reserved transfers for SnapMirror volumes *n* is a variable number, and depends on the system model.	0
`replication.logical.reserved_transfers` *n*	Specifies the number of reserved transfers for SnapMirror and SnapMirror qtrees *n* is a variable number, and depends on the system model.	0
`snapmirror.volume.local_nwk_bypass.enable`	Specifies whether local SnapMirror transfer optimization is enabled **Note:** If this option is set to `on`, the `replication.throttle.enable` option and the `snapmirror throttle` option settings do not have an impact on SnapMirror transfers within the same system.	`on`

For more information about SnapMirror options, see the `na_options` man page.

Related concepts

How synchronous SnapMirror handles network issues on page 80

Related tasks

Turning SnapMirror logging on on page 180

Related references

Methods for specifying destination systems on the SnapMirror source on page 127

SnapMirror files

SnapMirror uses configuration files, log files, and other files.

The following table lists the files used by SnapMirror, along with their corresponding functions:

File	Function
`/etc/snapmirror.conf`	Enables you to specify SnapMirror source and destination relationships, along with the following settings: • SnapMirror update schedules for a relationship • Type of relationship: single path, multipath, or failover • Other options for a given SnapMirror relationship
`/etc/snapmirror.allow`	Enables you to specify the SnapMirror destinations that are allowed to copy from the system **Note:** You can also use the `options snapmirror.access` command to specify the allowed destinations. However, if both the `options snapmirror.access` command and the `/etc/snapmirror.allow` file are used, `options snapmirror.access` takes precedence. This can affect the initialization of SnapMirror relationships.
`/etc/log/snapmirror.x`	Records the SnapMirror data transfer history **Note:** There might be one or more SnapMirror log files. The latest logs are stored in the file named `snapmirror`. The older logs are named `snapmirror.0` and `snapmirror.1`.
`/etc/hosts`	Uses entries in this file to resolve host names

For more information about SnapMirror files, see the following man pages:

- `na_snapmirror.conf`
- `na_snapmirror.allow`
- `na_hosts`

Related concepts

What the snapmirror.conf file does on page 129

SnapMirror data transfer logs on page 179
Scheduled updates for volumes or qtrees on page 138
Data replication from one destination to another in a series (cascading) on page 107

Related tasks

Turning SnapMirror logging on on page 180
Changing scheduled updates for one volume or qtree on page 139
Turning off scheduled updates for one volume or qtree on page 140
Listing SnapMirror destinations for a volume in a cascading series on page 110
Restructuring a cascade on page 111

Related references

Methods for specifying destination systems on the SnapMirror source on page 127
Format of SnapMirror log files on page 181

SnapMirror support for IPv6

SnapMirror supports the use of IPv6 addresses to specify source and destination systems. However, there are some differences between the specification of IPv6 and IPv4 addresses.

The SnapMirror commands that are affected by the support for IPv6 addresses are given in the following list:

- `snapmirror initialize`
- `snapmirror update`
- `snapmirror resync`

Note: Before using IPv6 functionality for a system, ensure that the `ip.v6.enable` option is set to `on`.

When using an IPv6 address to specify a SnapMirror source or destination system, you need to enclose the IPv6 address within square brackets. The usage is shown in the following examples:

Use of IPv6 address with the snapmirror initialize command

```
systemB> snapmirror initialize -S
[fd20:8b1e:b255:4166:2a0:98ff:fe07:23f3]:src_vol dst_vol
```

Use of IPv6 address in a snapmirror.conf file entry

```
[fd20:8b1e:b255:4166:2a0:98ff:fe07:23f3]:src_vol dst_system:dst_vol -
15 2 * *
```

Use of IPv6 addresses in a snapmirror.conf file entry for a multipath SnapMirror relationship

```
relation_1=multi(fd20:8b1e:b255:4166:2a0:98ff:fe07:23f3,dst_system)
(2001:0:0:0:0:ffd3:0:57ab,dst_system)

relation_1:src_vol dst_system:dst_vol - * * * *
```

Use of IPv6 address in the snapmirror.allow file

```
fd20:8b1e:b255:4166:2a0:98ff:fe07:23f3
```

Use of IPv6 address with the snapmirror.access option

When using an IPv6 address with the `snapmirror.access` option, it is optional to enclose the access specification within double quotes. The usage is shown in the following example.

```
systemB> options snapmirror.acess
"host=fd20:8b1e:b255:4166:2a0:98ff:fe07:23f3"
```

Related references

Syntax for snapmirror.conf file entries on page 131

Setting up a basic SnapMirror operation

Before initiating SnapMirror operations, you must enable the appropriate licenses on the source and destination systems. Also, you need to specify the destination systems that can access the source system for updates.

Before you begin

If your source volumes contain directories that are accessed by CIFS clients, you should have ensured that those directories are in the Unicode format before replicating the volume using SnapMirror.

Also, you should have ensured that you have appropriate SnapMirror licenses for both the source and destination systems.

Steps

1. For both the source and the destination system consoles, enter the following command to enable the SnapMirror license on the source and destination systems:

   license add *snapmirror_license_key*

2. On the source system console, use the `options snapmirror.access` command to specify the host names of systems that are allowed to copy data directly from the source system.

   Example

   `options snapmirror.access host=d_systemA`

3. On the destination system, create or edit the `/etc/snapmirror.conf` file to specify the volumes and qtrees to be copied and the schedule (*`minute hour day_of_month day_of_week`* or `sync`) on which the destination is updated.

   Example

   The following entry specifies Snapshot copy replication from `vol0` of `s_systemA` to `vol1` of `d_systemA` at a maximum of 2,000 kilobytes per second 15 minutes past every hour, Monday through Friday:

   `s_systemA:vol0 d_systemA:vol1 kbs=2000,restart=always 15 * * 1,2,3,4,5`

   To synchronously mirror `vol0` to `vol1`, you must use the following entry:

   `s_systemA:vol0 d_systemA:vol1 - sync`

   For more information about schedule entries in the `/etc/snapmirror.conf` file of the destination system, see the na_snapmirror.conf(5) man page.

4. On both the source and destination system consoles, use the `snapmirror on` command to enable SnapMirror on the source and destination systems.

5. Prepare the destination system appropriately, depending on whether you are setting up SnapMirror volume or qtree replication.

If you are setting up a...	Then...
Volume SnapMirror relationship	On the destination system console, use the `vol create` command to create a destination volume, then use the `vol restrict` command to mark the volume as restricted.
Qtree SnapMirror relationship	Ensure that the volume on the destination system where you want to replicate a qtree with SnapMirror is online and not restricted. Do not manually create a destination qtree.

6. On the destination system console, use the `snapmirror initialize` command to create an initial complete (baseline) copy of the source on the destination and start the mirroring process.

   For SnapMirror volume replication:

   Example

   Invoking the following command transfers a complete copy of the source volume (`vol0` on `systemA`) to the destination volume (`vol2` on `systemB`):

snapmirror initialize -S systemA:vol0 systemB:vol2
The destination volume must be configured as restricted and read-only.

For SnapMirror qtree replication:

Example

The following command creates a destination qtree (`qtree4` on `vol1` on `systemB`) and transfers a complete copy of the source qtree (`qtree4` on `vol1` on `systemA`) to that destination qtree:

snapmirror initialize -S systemA:/vol/vol1/qtree4 systemB:/vol/vol1/qtree4
The volume in which the destination qtree is created must be online and writable.

After using the `snapmirror initialize` command, the scheduled Snapshot copy replication that you specified in Step 3 automatically updates the destination volume or qtree at the specified times.

After you finish

If the SnapMirror source volume or qtree becomes unavailable, you can use the `snapmirror break` command to make the destination volume or qtree writable. This enables you to provide continued access to data for the clients who are no longer able to access the unavailable source.

Related concepts

What the snapmirror.conf file does on page 129
Initialization of a SnapMirror destination on page 116
Conversion of a destination to a writable volume or qtree on page 168

Related tasks

Enabling SnapMirror by entering license keys on page 81
Turning SnapMirror on on page 81

Related references

Destination accessibility when using CIFS with SnapMirror on page 89
Methods for specifying destination systems on the SnapMirror source on page 127

Firewall usage with SnapMirror

SnapMirror uses the typical socket/bind/listen/accept sequence on a TCP socket.

SnapMirror source binds on port 10566. The destination storage system contacts the SnapMirror source storage system at port 10566 using any of the available ports assigned by the system. The firewall must allow requests to this port of the SnapMirror source storage system.

Synchronous SnapMirror requires additional TCP ports to be open. The source storage system listens on TCP ports 10566 and 10569. The destination storage system listens on TCP ports 10565, 10567,

and 10568. Therefore, you should ensure that the firewall allows a range of TCP ports from 10565 to 10569.

Data replication from one destination to another in a series (cascading)

You can replicate data from a SnapMirror destination to another system using SnapMirror. Therefore, a system that is a destination for one SnapMirror relationship can act as the source for another SnapMirror relationship. This is useful when you need to copy data from one site to many sites.

Instead of replicating data from a single source to each of the destinations, you can replicate data from one destination to another destination, in a series. This is referred to as cascading.

Note: You can replicate data from a destination volume in the same way you replicate from a writable source volume.

Related concepts

Initialization of a SnapMirror destination on page 116

Related tasks

Setting up a basic SnapMirror operation on page 104
Enabling SnapMirror by entering license keys on page 81

Related references

Methods for specifying destination systems on the SnapMirror source on page 127
Maximum number of concurrent replication operations on page 123

Supported cascade configurations for SnapMirror

Only certain types of SnapMirror configurations support cascading.

The supported cascading configurations are listed in the following table. Any other configuration, such as extending the cascade beyond the number of cascades shown in the table, is not supported. This limitation does not apply to the strictly asynchronous volume SnapMirror cascading configuration, which can propagate to more than three systems.

The following table lists the two-hop cascade configurations that are supported for SnapMirror replication:

System A to system B	System B to system C
Synchronous SnapMirror	Asynchronous volume SnapMirror
Asynchronous volume SnapMirror	Asynchronous volume SnapMirror
Asynchronous volume SnapMirror	Qtree SnapMirror
Qtree SnapMirror	Asynchronous volume SnapMirror

Example

The first line states that system A has a synchronous SnapMirror relationship with system B, and that system B has an asynchronous volume SnapMirror relationship with system C.

When the first hop in the cascade is synchronous SnapMirror, the synchronous replication can be to one destination system only. Subsequent SnapMirror replications cascading from that destination system must be asynchronous and can be to multiple destination systems.

Supported three-hop cascade configurations for SnapMirror

Only certain combinations of SnapMirror replication types are supported for three-hop cascade configurations.

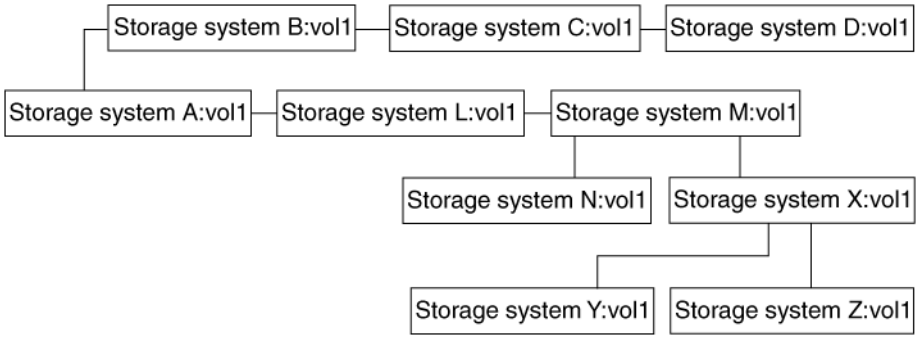
Cascades of three SnapMirror relationships whose first relationship is a synchronous SnapMirror relationship are supported on Data ONTAP 7.1.2 and 7.2.1 releases, and later releases.

The following table lists the three-hop cascade configurations that are supported for SnapMirror replication:

System A to system B	System B to system C	System C to system D
Synchronous SnapMirror	Asynchronous volume SnapMirror	Asynchronous volume SnapMirror
Synchronous SnapMirror	Asynchronous volume SnapMirror	Qtree SnapMirror
Asynchronous volume SnapMirror	Asynchronous volume SnapMirror	Asynchronous volume SnapMirror
Asynchronous volume SnapMirror	Asynchronous volume SnapMirror	Qtree SnapMirror
Asynchronous volume SnapMirror	Qtree SnapMirror	Asynchronous volume SnapMirror
Qtree SnapMirror	Asynchronous volume SnapMirror	Asynchronous volume SnapMirror

Sample cascade setup

The following diagram depicts the series of cascading volume destinations:



To support the configurations depicted in the preceding diagram, the entries in the `/etc/snapmirror.conf` file in each of the systems is the cascade should be similar to the following entries:

```
systemA:vol1 systemB:vol1 - 15 * * 1,2,3,4,5
systemA:vol1 systemL:vol1 - 15 * * 1,2,3,4,5
systemB:vol1 systemC:vol1 - 25 * * 1,2,3,4,5
systemC:vol1 systemD:vol1 - 35 * * 1,2,3,4,5
systemL:vol1 systemM:vol1 - 25 * * 1,2,3,4,5
systemM:vol1 systemX:vol1 - 35 * * 1,2,3,4,5
systemM:vol1:systemN:vol1 - 35 * * 1,2,3,4,5
systemX:vol1 systemY:vol1 - 45 * * 1,2,3,4,5
systemX:vol1 systemZ:vol1 - 45 * * 1,2,3,4,5
```

Note: When specifying the destination update schedule in the `snapmirror.conf` file, you should stagger the update times instead of starting multiple destination updates at the same time. If SnapMirror does not have enough resources to perform all scheduled destination updates, it postpones some updates. As a result, SnapMirror might need to perform subsequent updates at times that are different from those you specify in the `snapmirror.conf` file.

How SnapMirror handles Snapshot copies for cascading destinations

SnapMirror retains on the original source volume the Snapshot copies needed for transfers to destinations further down the line. Snapshot copies that are still needed by a destination are labeled

`snapmirror` in the output of the `snap list` command. SnapMirror deletes the Snapshot copies it no longer needs.

If you remove a destination from the cascade, you can use the `snapmirror release` command from the immediate source to tell SnapMirror to delete the Snapshot copies associated with that destination.

Listing SnapMirror destinations for a volume in a cascading series

You can use the `snapmirror destinations` command to display the destinations for a volume in a cascading series.

About this task

The `snapmirror destinations` command also displays entries related to `vol clone` command and `dump` command operations (if any) for SnapMirror source or destination volumes.

Step

1. From the system with the volume serving as the source, enter the following command:

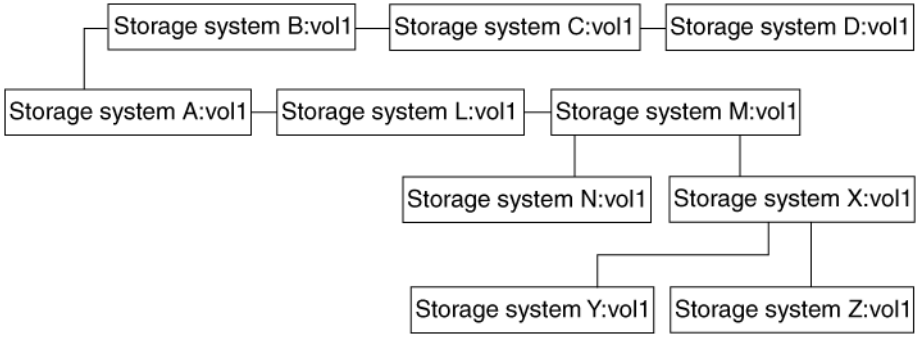
   `snapmirror destinations [-s] [`*`volume_name`*`]`

   The `-s` option generates a list of the names of the Snapshot copies retained for each destination.

   `volume_name` is the name of the source volume for which you want to see the destinations.

Listing SnapMirror destinations for a volume in a cascading series

Suppose that you have used the `snapmirror destinations` command for a cascade configuration depicted in the following figure:



The output for the `snapmirror destinations` command for such a cascade configuration is similar to the following:

```
systemA> snapmirror destinations vol1
```

```
Path        Destination
/vol/vol1   systemB:vol1->systemC:vol1->systemD:vol1
/vol/vol1   systemL:vol1->systemM:vol1->systemX:vol1->systemY:vol1
/vol/vol1   systemL:vol1->systemM:vol1->systemX:vol1->systemZ:vol1
/vol/vol1   systemL:vol1->systemM:vol1->systemN:vol1
```

Note: If you do not specify a volume name in the command, the output includes information about each destination volume on the system.

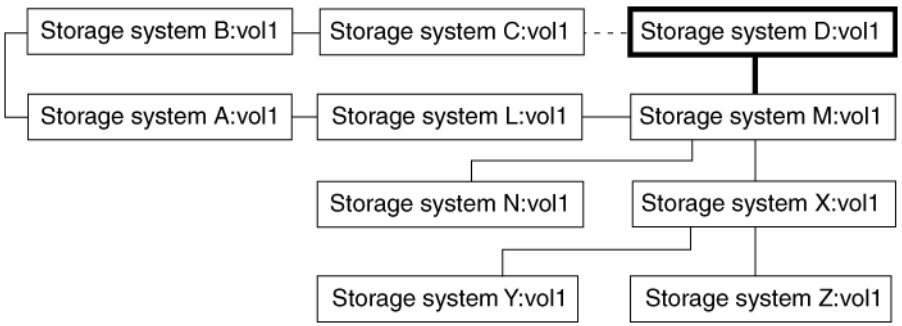
Restructuring a cascade

You might want to restructure a cascade to balance the load on your systems; to use a system or volume for a different purpose; or to perform upgrades, maintenance, or repairs.

About this task

For example, in the following cascade structure, you might want to make `systemD:vol1` a destination of `systemM:vol1` instead of a destination of `systemC:vol1`.

The following diagram illustrates restructuring the relationship of the destinations in a cascade:



Steps

1. On the destination system, change the `/etc/snapmirror.conf` file to indicate the new source for the destination.

   Example

   ```
   systemM:vol1 systemD:vol1 - 35 * * 1,2,3,4,5
   ```

2. As required, choose one of the actions from the following table.

If the newest Snapshot copy on the destination...	Then...
Exists on the source	Use the following command to update the destination from the new source. **snapmirror update -S *source_volume dest_system*:*dest_volume*** For example: **snapmirror update -S systemM:vol1 systemD:vol1**
Does not exist on the source	Perform one of the following tasks. • Update the new source from the original source using the `snapmirror update` command. Wait for the destination to update. • Make the destination writable using the `snapmirror break` command. Then resynchronize the destination with the new source using the `snapmirror resync` command.

3. Release the former source using the following command:

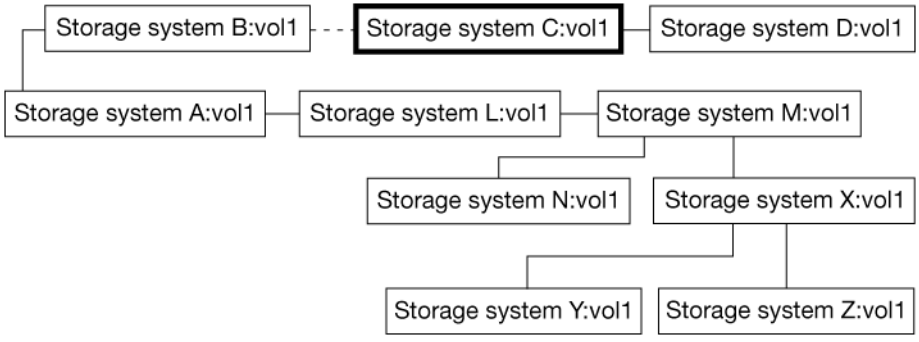
   snapmirror release *source_volume* [[*dest_system*:]*dest_volume*]

   Example

   ```
   systemC> snapmirror release systemC:vol1 systemD:vol1
   ```

Disconnecting a destination from a cascading series

The following diagram depicts the change in the SnapMirror cascade configuration:



For the configuration depicted in the preceding diagram, suppose that from `systemB` you enter the following command:

snapmirror release vol1 systemC:vol1

These results follow.

- `systemA:vol1` continues to be the source for the destination `systemB:vol1`.
- `systemC:vol1` no longer copies from `systemB:vol1`. SnapMirror retains Snapshot copies for `systemC` and below.
- If `systemC` requests an update from `systemB`, the destination is reestablished if it is still not writable and the base Snapshot copy still exists on the source.
- `systemD:vol1` still copies `systemC:vol1`.
- All the destinations that depend on `systemL:vol1` continue functioning as before.

You can check that the destination was released by running the `snapmirror destinations` command on `systemA`, as follows.

```
systemA> snapmirror destinations -s systemA:vol1

Volume Snapshot                      Destination
vol1   systemB(0015269532)_vol1.37  systemB:vol1
vol1   systemL(0015269532)_vol1.42  systemL:vol1->systemM:vol1->systemXvol1-
>systemY:vol1
vol1   systemL(0015269532)_vol1.42  systemL:vol1->systemM:vol1->systemXvol1-
>systemZ:vol1
vol1   systemL(0015269532)_vol1.42  systemL:vol1->systemM:vol1->systemN:vol1
```

Note: If you want to permanently release a destination, you should delete the entry in the `/etc/snapmirror.conf` file. Alternatively, you can comment out the entry by preceding it with a pound sign (#). Otherwise, SnapMirror attempts to update the destination.

Data replication using tape

You can replicate data from the SnapMirror source to the destination storage system using tape. You can use the `smtape backup` command to backup data from the source and the `smtape restore` command to restore data to the destination.

In this scenario, you want to establish a SnapMirror relationship between a source system and a destination system over a low-bandwidth connection. Incremental Snapshot mirroring from the source to the destination over the low bandwidth connection is feasible, but the initial base Snapshot mirroring is not.

In such a case, it might be faster to first transfer the initial base Snapshot image from source to destination using tape, and then set up incremental SnapMirror updates to the destination system through the low-bandwidth connection.

This scenario uses the following configuration:

- A low-bandwidth connection between the source and destination systems
- A local tape drive attached to the source system

- A local tape drive attached to the destination system

  Note: To prevent extended tape-to-storage system transfer time, you should ensure that the destination system disks be the same size and in the same RAID configuration as the source system disks.

You must follow this sequence of activities to set up this arrangement.

1. On the source system, use the `smtape backup` command to copy all volume Snapshot copies, including the base Snapshot copy, to tape, and use the `smtape continue` command to continue the copy if more than one backup tape is necessary.
2. Physically transport the backup tapes from the source system to the destination system.
3. On the destination system, use the `vol create` and `vol restrict` commands to set up a SnapMirror target volume.
4. Use the `smtape restore` command to copy the initial SnapMirror tape to the destination system and, if necessary, use the `smtape continue` command to continue the copy if it is stored on more than one backup tape.
5. Either use the `snapmirror update` command to manually mirror an incremental update from the source to the destination system over the low-bandwidth connection, or edit the `snapmirror.conf` file to set up an incremental update schedule from the source to destination system.
6. After completing manual or scheduled incremental update over a connection, you can use the `snapmirror release` command to eliminate the source-to-tape relationship and associated Snapshot copy.

For more information about `smtape` commands, see the *Data ONTAP Data Protection Tape Backup and Recovery Guide for 7-Mode*.

Copying source to intermediate tape

For an initial, baseline transfer for a SnapMirror relationship, you might want to do the transfer using a tape. To do this, you first need to copy the base data to tape.

Steps

1. At the source system, start the data transfer to tape by using the `smtape backup` command.
2. If you are prompted for another tape, add another tape to the drive, and continue transfer of data to tape by using the `smtape continue` command.
3. Repeat the previous step until the volume is completely copied to tape.

Related concepts

Initialization of a SnapMirror destination on page 116
Initialization of a SnapMirror destination from tape on page 117

Related tasks

Initializing a SnapMirror destination by using tape on page 115

Initializing a SnapMirror destination by using tape

For an initial, baseline transfer for a SnapMirror relationship, you might want to perform the transfer using a tape. After copying the base data to the tape, you need to copy the data from the tape to the intended destination, and initialize the destination.

Steps

1. Create a volume on the SnapMirror destination system.

   For more information about how to create a volume, see the *Data ONTAP Storage Management Guide for 7-Mode*.

2. Put the volume in the restricted state. For more information about how to restrict a volume, see the *Data ONTAP Storage Management Guide for 7-Mode*.

3. Load the tape (made with `smtape backup`) into the destination system's local tape device.

4. Start the initialization by using the `smtape restore` command on the destination system.

5. If the system prompts you for another tape, add the next tape to the drive and continue the initialization by using the `smtape continue` command.

6. Repeat the previous step until the volume is completely copied from the tape.

7. Optional: You can update the data online manually with the following command:

   **`snapmirror update [-k` *`n`*`] -S` *`source_system`*`:`*`source_volume`*
   `[`*`dest_system`*`:]`*`dest_volume`

   `-k` *`n`* sets the maximum transfer speed to *`n`* kilobytes per second. This option has the same effect as the `kbs` argument in the `/etc/snapmirror.conf` file.

   `-S` *`source_system`*`:`*`source_volume`* specifies the source system and volume for the migration. *`source_volume`* is the volume you want to copy.

   `dest_system` is the name of the destination system.

   `dest_volume` is the destination volume.

   Note: Alternatively, you can update the baseline transfer automatically with the schedule you set in the `/etc/snapmirror.conf` file.

Related concepts

Manual update of a SnapMirror destination on page 141
What the snapmirror.conf file does on page 129
Initialization of a SnapMirror destination on page 116
Initialization of a SnapMirror destination from tape on page 117

Related tasks

Editing the snapmirror.conf file on page 130

Initialization of a SnapMirror destination

You must use the `snapmirror initialize` command to perform a complete (baseline) transfer of information whenever you start a SnapMirror source-destination relationship for the first time. This process is known as initializing a destination. You need to consider a number of issues when initializing a destination.

Quotas for SnapMirror destination qtrees

Qtree quotas apply to qtrees in a SnapMirror relationship.

If a destination qtree is limited to 100 GB, transfers from a source qtree greater than 100 GB will fail unless the source qtree drops to less than 100 GB or the quota for the destination qtree is increased.

Note: User quotas also apply to SnapMirror destination qtrees. However, a SnapMirror qtree update does not fail if the user exceeds the quotas.

Guidelines for creating a qtree SnapMirror relationship

When creating a qtree SnapMirror relationship, you should follow certain guidelines.

- You must establish a qtree SnapMirror relationship between volumes that have the same `vol lang` settings.
- After establishing a qtree SnapMirror relationship, you must not change the language assigned to the destination volume.
- You must avoid white space (space or tab characters) in names of source and destination qtrees.
- You must not rename volumes or qtrees after establishing a qtree SnapMirror relationship. If you rename volumes or qtrees, you must update the SnapVault or SnapMirror configuration.

Initialization of a SnapMirror destination from tape

You can initialize a SnapMirror destination volume from tape using the `smtape backup` command on the source volume and the `smtape restore` command on the destination volume.

The `smtape backup` and `smtape restore` functions are valid for volumes, but not for qtrees in a SnapMirror relationship.

For more information about `smtape` commands, see the *Data ONTAP Data Protection Tape Backup and Recovery Guide for 7-Mode*.

Related tasks

Initializing a SnapMirror destination by using tape on page 115

Initializing a SnapMirror destination

You can initialize a SnapMirror destination by using the `snapmirror initialize` command.

Before you begin

If your source volumes contain directories that are accessed by CIFS clients, you should ensure that those directories are in the Unicode format before carrying out the initial SnapMirror replication of that volume.

Steps

1. If the destination is a volume, is online, and has not been initialized before, enter the following command from the destination system:

   `vol restrict *dest_volume*`

   `dest_volume` is the destination volume.

   Note: Do not use the `vol restrict` command on a qtree. If you are initializing a qtree, go to Step 2.

2. From the destination system, enter the following command:

   `snapmirror initialize [*options*] [*dest_system*:] {*dest_volume*|*qtree_path*}`

   `options` can be one or more of the following:

   - `-k` *`n`* sets the maximum transfer speed to *`n`* kilobytes per second. This option has the same effect as the `kbs` argument in the `/etc/snapmirror.conf` file.
   - `-S` [*`source_system`*:]{*`source_volume`* | *`source_qtree_path`*} specifies the source system and volume or qtree to copy.
     `source_volume` is the volume you want to copy.

Note: The source specified must match an entry for *source_volume* in the `/etc/snapmirror.conf` file, if one exists. If an entry exists but does not match, the operation displays an error message and terminates. If there is no entry for the specified source, the command runs.

source_qtree_path is the path to the qtree you want to copy. If the `-S` option is not set, the source must be specified in the `/etc/snapmirror.conf` file. If it is not specified, the operation displays an error message and terminates.

Note: The *source_qtree_path* can be a qtree in a SnapMirror destination volume.

- `-c` *snapshot_name* creates a Snapshot copy (with the name *snapshot_name*) of a qtree on the destination after the next update (so that it does not compete with any ongoing updates). SnapMirror does not lock or delete this Snapshot copy.

  Note: *snapshot_name* cannot be `minutely.`*x*, `hourly.`*x*, `nightly.`*x*, or `weekly.`*x*, because these names are reserved for scheduled Snapshot copies.

  Note: This option is valid only for a qtree.

- `-s` *snapshot_name* specifies an existing source qtree Snapshot copy to be transferred. This prevents the normal action of the source creating a Snapshot copy to transfer. SnapMirror does not lock or delete this Snapshot copy.

  Note: This option is valid only for a qtree SnapMirror replication.

dest_system is the name of the destination system. The destination can reside on the same system as the source or on another system.

dest_volume or *qtree_path* specifies the destination volume or qtree. If it is associated with a local source specified in the `/etc/snapmirror.conf` file, SnapMirror uses that source. If the destination volume or qtree specified is not in a scheduled relationship, then the `-S` option must be used to provide a source.

The `snapmirror initialize` command creates the destination qtree, but you must specify the destination qtree name at the end of the path as though it already existed.

Note: If the destination qtree exists before the command runs, the command fails.

Example

Using the following command, SnapMirror transfers a complete copy of the source volume (`vol0` on `systemA`) to the destination volume (`vol2` on `systemB`):

```
systemB> snapmirror initialize -S systemA:vol0 systemB:vol2
```

Example

Using the following command, SnapMirror transfers a complete copy of the qtree source (`qtree4` on `vol1` on `systemA`) to the destination qtree (`qtree4bak` on `vol1` on `systemB`):

```
systemB> snapmirror initialize -S systemA:/vol/vol1/qtree4
systemB:/vol/vol1/qtree4bak
```

Related references

Destination accessibility when using CIFS with SnapMirror on page 89

Space guarantee for a volume SnapMirror destination

You can reserve space for a FlexVol volume in an aggregate by specifying the appropriate option for the volume. You can use this feature to ensure a space guarantee for a volume SnapMirror destination.

You can view the settings for a given volume by using the `vol options` command.

The default setting for a volume is: `guarantee=volume`. This indicates that space for the volume is reserved in the aggregate. You can change this setting by using the following command:

`vol options *vol_name* guarantee {none | file | volume}`

If you change the space guarantee type for the destination volume to **`file`**, the space guarantee is disabled until you break the SnapMirror relationship. The space guarantee for a SnapMirror destination volume with `guarantee=volume` is based on the size of the destination volume that is displayed in the output of the `vol size` command.

For more information about volume space guarantees, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Note: In a volume SnapMirror relationship, you can have `guarantee=volume` for both the source and the destination volumes.

Initializing a destination for non-qtree data

Non-qtree data is any data on a system that is not contained in its qtrees. Non-qtree data can include configuration and logging directories on the system (for example, `/etc` or `/logs`) that are not normally visible to system clients. You can use SnapMirror to replicate non-qtree data.

Step

1. To use SnapMirror to replicate non-qtree data from a source to a destination, enter the following command on the destination system:

   `snapmirror initialize -S *source_system*:/vol/*source_volume*/-*dest_system*:/vol/*dest_volume*/*qtree_name*`

   The dash (-) character indicates all non-qtree data in the specified volume.

Note: The `snapmirror initialize` command creates a destination qtree with the name specified, and then transfers the non-qtree data in the volume. The qtree must not exist before using the command, otherwise the command fails.

Example

Using the following command, SnapMirror transfers to the destination qtree (`non_qtree_data_in_vol3` on `vol4` on `systemB`) a complete copy of all the data in `vol3` (of `systemA`) that is not a part of a qtree.

```
systemB> snapmirror initialize -S systemA:/vol/vol3/- systemB:/vol/
vol4/non_qtree_data_in_vol3
```

Note: The non-qtree data can only be a SnapMirror source, never a destination. Although you can copy data from the non-qtree data to another qtree, you cannot perform a vice versa operation.

Initializing a destination for non-qtree data

If you run the `snapmirror quiesce` or the `snapmirror break` command on the destination volume (`/vol/vol4/non_qtree_data_in_vol3`), you can resynchronize the destination volume to the source volume.

```
systemB> snapmirror resync -S /vol/vol3/- /vol/vol4/
non_qtree_data_in_vol3
```

Note: You cannot resynchronize the SnapMirror relationship in the opposite direction.

How the snapmirror initialize command copies volumes

When the `snapmirror initialize` command copies a volume, it creates a Snapshot copy of all the data on the source and transfers it to the destination. The destination is a volume that you have already created and marked restricted. After SnapMirror finishes transferring the data, it brings the destination online in a read-only state. This version of the destination is the baseline for the first incremental update.

While the initial data transfer is taking place, the destination is marked `invalid` in the output of a `vol status` command. The volume becomes valid and goes online after the initial transfer is complete.

Note: Any attempt to bring this volume online manually will only succeed after the initial transfer is complete.

How the snapmirror initialize command copies qtrees

To use SnapMirror to copy a qtree, you do not create a destination qtree because the `snapmirror initialize` command creates it. The volume where you want the destination qtree to be must be online. After the destination qtree is initialized, it is no longer writable. However, the rest of the volume where that qtree resides is still writable.

The destination Snapshot copy created by qtree initialization is marked `busy` in the output of the `snap list` command until the next transfer is complete.

What happens after SnapMirror makes the initial copy to the destination

After you initialize a SnapMirror volume replication, the files and Snapshot copies in the source volume are available on the destination. After you initialize a SnapMirror qtree replication, the files on the source qtree are available on its destination qtree.

You can export the destination for NFS mounting or add a share corresponding to the destination for CIFS sharing.

How to check the initialization of a volume

To check that a destination volume has been initialized, you can use the `snapmirror status` command.

If you specify no options or arguments, the `snapmirror status` command displays the status of the volumes in the system, as shown in the following example. You also can use the `vol status` or the `qtree` command to check whether the volume or qtree is a SnapMirror destination.

```
systemA> snapmirror status
Snapmirror is on.
Source          Destination       State          Lag       Status
systemA:vol0    systemA:vol0bak   Snapmirrored   00:56:58  Idle
systemA:vol1    systemB:vol6      Source         23:69:26  Transferring (126 MB
done)
```

Related tasks

Checking SnapMirror data transfer status on page 151

Checking the initialization of a qtree

You can check the creation and initialization of a destination qtree by using the `qtree` command.

Step

1. Enter the following command:

   `qtree`

Note: If you specify no options or arguments, the `qtree` command displays the status of all the qtrees in the system.

Example

```
systemA> qtree

Volume        Tree  Style  Oplocks   Status
--------      ----- -----  -------   ------
vol0                unix   enabled   normal
qtree24             unix   enabled   normal
systemB_vol0        unix   disabled  normal
systemB_vol0  qt1   mixed  enabled   snapmirrored
systemB_vol0  qt2   unix   disabled  normal
systemB_vol0  qt3   ntfs   enabled   snapmirrored
```

How the snapmirror initialize command matches source and destination volume size

When you use the `snapmirror initialize` command to initialize a volume replication, SnapMirror sets the `vol options fs_size_fixed` option to `on`. This option forces the file system on the destination volume to remain the same size as the file system on the source volume.

What you can do if an initial SnapMirror transfer fails

If an initial SnapMirror transfer fails, you can resume the transfer by re-entering the `snapmirror initialize` command, under certain conditions.

If the following conditions are met, you can resume the initialization of a SnapMirror relationship:

- The value for restart mode in the `/etc/snapmirror.conf` file is set to `always` or is set to the default, and the next scheduled update has not begun.
- The output of the `snapmirror status` command displays that the process has a restart checkpoint.
- The Snapshot copy used for the initial SnapMirror transfer still exists.
- The disk geometry has not changed.

Note: If these conditions are not satisfied, you cannot resume the initial transfer. You need to start the initial SnapMirror transfer again.

SnapMirror does not automatically retry to initialize a destination.

Maximum number of concurrent replication operations

The limit on the number of concurrent replication operations depends on your system model. The system resources are shared between SnapMirror and SnapVault replication operations. Therefore, the limit for a particular type of replication is reduced if there are any other types of replication operations being performed concurrently.

A SnapMirror replication or a SnapVault backup consists of two replication operations: one operation on the source system and the other on the destination. Therefore, if a system is both a source and a destination, it uses two replication operations. For example, *systemA* is the source as well as the destination system in a SnapMirror or SnapVault relationship and the total number of replication operations available for that relationship is 100. Then, the total number of available replication operations, when the transfer is active, is 98.

Each value listed in the following two tables indicates the maximum number of concurrent replication operations allowed for a specific model, when using a single type of replication. The values are exclusive, and not cumulative. For example, if you are using the maximum number of synchronous SnapMirror replication operations for a system as a source, you cannot use any more replication operations of any type for the system.

The NearStore personality license is enabled by default in Data ONTAP 8.1. However, to enable the NearStore option, you must set the `licensed_feature.nearstore_option.enable` option to **on**.

The following table lists the maximum number of concurrent replication operations without the NearStore option enabled:

Model	Volume SnapMirror		Synchronous SnapMirror		Qtree SnapMirror		SnapVault		Open Systems SnapVault
	SRC	DST	SRC	DST	SRC	DST	SRC	DST	DST
FAS2220	50	50	16	16	64	64	64	64	16
FAS2240	50	50	16	16	64	64	64	64	16
CBvM100	N/A	N/A	N/A	N/A	N/A	N/A	16	16	N/A
FDvM100	8	8	8	8	16	16	16	16	N/A
3140	50	50	16	16	64	64	64	64	16
3160	50	50	16	16	64	64	64	64	16
3170	50	50	16	16	64	64	64	64	16
3210	50	50	16	16	64	64	64	64	16
3220	50	50	16	16	64	64	64	64	16

Model	Volume SnapMirror		Synchronous SnapMirror		Qtree SnapMirror		SnapVault		Open Systems SnapVault
	SRC	DST	SRC	DST	SRC	DST	SRC	DST	DST
3240	50	50	16	16	64	64	64	64	16
3250	50	50	16	16	64	64	64	64	16
3270	50	50	16	16	64	64	64	64	16
6040A	100	100	24	24	96	96	96	96	24
6080	150	150	32	32	128	128	128	128	32
6210	150	150	32	32	128	128	128	128	32
6220	150	150	32	32	128	128	128	128	32
6240	150	150	32	32	128	128	128	128	32
6250	150	150	32	32	128	128	128	128	32
6280	150	150	32	32	128	128	128	128	32
6290	150	150	32	32	128	128	128	128	32
FAS8020	150	150	32	32	128	128	128	128	32
FAS8040	150	150	32	32	128	128	128	128	32
FAS8060	150	150	32	32	128	128	128	128	32
FAS8080	150	150	32	32	128	128	128	128	32
FAS2554	50	50	16	16	64	64	64	64	16
FAS2552	50	50	16	16	64	64	64	64	16
FAS2520	50	50	16	16	64	64	64	64	16

The following table lists the maximum number of concurrent replication operations with the NearStore option enabled:

Model	**Volume SnapMirror**		**Synchronous SnapMirror**		**Qtree SnapMirror (single path)**		**Qtree SnapMirror (multipath)**		**SnapVault**		**Open Systems SnapVault**
	SRC	**DST**	**SRC**	**DST**	**SRC**	**DST**	**SRC**	**DST**	**SRC**	**DST**	**DST**
FAS2220	50	100	16	16	120	120	60	60	120	120	64
FAS2240	50	100	16	16	120	120	60	60	120	120	64
3140	50	100	16	16	160	160	80	80	160	160	64
3160	50	100	16	16	320	320	160	160	320	320	128
3170	50	100	16	16	384	384	192	192	384	384	128
3210	50	100	16	16	120	120	60	60	120	120	64
3220	50	100	16	16	320	320	160	160	320	320	128
3240	50	100	16	16	320	320	160	160	320	320	128
3250	50	100	16	16	320	320	160	160	320	320	128
3270	50	100	16	16	320	320	160	160	320	320	128
6040A	100	200	24	24	384	384	192	192	384	384	96
6080	150	300	32	32	512	512	256	256	512	512	128
6210	150	300	32	32	512	512	256	256	512	512	128
6220	150	300	32	32	512	512	256	256	512	512	128
6240	150	300	32	32	512	512	256	256	512	512	128
6250	150	300	32	32	512	512	256	256	512	512	128
6280	150	300	32	32	512	512	256	256	512	512	128
6290	150	300	32	32	512	512	256	256	512	512	128
FAS8020	150	300	32	32	512	512	256	256	512	512	128
FAS8040	150	300	32	32	512	512	256	256	512	512	128
FAS8060	150	300	32	32	512	512	256	256	512	512	128
FAS8080	150	300	32	32	512	512	512	512	512	512	128
FAS2554	50	100	16	16	384	384	192	192	384	384	128
FAS2552	50	100	16	16	384	384	192	192	384	384	128

Model	Volume SnapMirror		Synchronous SnapMirror		Qtree SnapMirror (single path)		Qtree SnapMirror (multipath)		SnapVault		Open Systems SnapVault
	SRC	DST	SRC	DST	SRC	DST	SRC	DST	SRC	DST	DST
FAS2520	50	100	16	16	384	384	192	192	384	384	128

SRC—Source; DST—Destination

Note: The NearStore personality license is not supported on the following systems:

- CBvM100
- FDvM100

The following factors affect the maximum number of concurrent replication operations that a system can achieve:

- Heavy use of system resources, such as CPU, memory, disk bandwidth, or network bandwidth, might reduce the resources available for SnapMirror or SnapVault operations.
- The use of TCP window sizes higher than the default value, for SnapMirror relationships, might restrict the system from achieving the maximum limit specified.
- A system with the NearStore personality enabled is optimized as a destination for SnapMirror and SnapVault replication operations.
  Therefore, a system with the NearStore personality can handle more replication operations than a system without the NearStore personality.

Related concepts

Limit on the number of concurrent SnapVault targets on page 240
Use of SnapMirror with S Family storage systems on page 205

Maximum number of concurrent replication operations in an HA pair

If a failover occurs, the available system cannot process more than the maximum number of concurrent replication operations specified for that system. These operations can be those that were scheduled for the surviving system, the failed-over system, or both.

If a failover occurs during data backup, all of the concurrent transfers happening at the time of the failure on the failed node are aborted, and are rescheduled by the partner node. Conversely, if a giveback occurs during data backup, all of the concurrent transfers happening at the time of the giveback on behalf of the partner node are aborted, and are rescheduled by the partner node.

If more than the maximum number of SnapMirror volume or qtree replications are scheduled to run concurrently, each additional transfer generates an error message stating that resource limits have been reached. Each transfer beyond the maximum is re-attempted once per minute until it succeeds, SnapMirror is turned off, or the update is terminated.

Methods for specifying destination systems on the SnapMirror source

There are two methods of specifying destination systems on the source systems.

The two methods to specify destination systems on the source are:

- By using the `snapmirror.access` option
- By using the `snapmirror.allow` file

Specifying SnapMirror destinations using the snapmirror.access option

You can specify the destination systems that are allowed access to the source system by using the `snapmirror.access` option. This option specifies which SnapMirror destination system can initiate transfers, and which network interfaces they can use. This is the preferred method for controlling SnapMirror access on SnapMirror source system.

Step

1. To specify the SnapMirror destinations that are allowed access to the SnapMirror source using the `snapmirror.access` option, on the source system, enter the following command:

   `options snapmirror.access` ***`access_specification`***

   The syntax is the same for SNMP, Telnet, and rsh, and is described in the na_protocolaccess(8) man page. For more information about the `options` command, see the na_options(1) man page.

   Note: This option setting persists across reboots.

Example

If you want a destination (`systemB`) to have access to the source (`systemA`), for a SnapMirror relationship to copy data from `systemA` to `systemB`, enter the following at the prompt on `systemA`.

```
systemA> options snapmirror.access host=systemA,systemB
```

Specifying SnapMirror destinations using the snapmirror.allow file

You can create a `snapmirror.allow` file in the `/etc/` directory on the source system. You can create entries in the `snapmirror.allow` file to specify the destination systems that are allowed to copy data directly from the source system. If the `snapmirror.access` option is set to `legacy` (the default setting), the `snapmirror.allow` file defines the access permissions.

Steps

1. As required, choose one of the actions from the following table.

If the `snapmirror.allow` file...	Then...
Does not exist in the `/etc/` directory on the root volume of the source system.	Create the `snapmirror.allow` file in the `/etc/`directory on the root volume of the source system by using a text editor.
Exists in the `/etc/` directory on the root volume of the source system.	Go to the next step.

2. Specify the SnapMirror destinations that can access the SnapMirror source using the `snapmirror.allow` file. Add the name of each allowed system on a separate line in the `snapmirror.allow` file.

   Note: You do not need to add the name of the local system.

3. Save edits to the file.

Example

If you want SnapMirror to copy data locally on `systemA` and to other systems named `systemB` and `systemC`, add the following entries in the `/etc/snapmirror.allow` file on `systemA`.

```
systemB
systemC
```

Note: Entries in the `snapmirror.allow` file are case-sensitive. You can use the `hostname` command on the destination systems to find the correct entries for the `snapmirror.allow` file.

Resolving host names to their IP addresses

By default, SnapMirror checks host names in the `/etc/snapmirror.allow` file against the host name sent from the destination system. Alternatively, you can set SnapMirror to resolve the host

names in the `/etc/snapmirror.allow` file to their IP addresses, and compare them with the IP address of the destination system.

Before you begin

The `/etc/snapmirror.allow` file entry must map to the IP address of the originating network interface on the destination system. For example, if the request comes from the IP address of a Gigabit Ethernet interface `e10` named `systemA-e10`, then the `/etc/snapmirror.allow` file must contain `systemA-e10` or `systemA-e10.acme.com` so that the name resolves to the correct IP address.

About this task

The `snapmirror.checkip.enable` option controls how the host names are checked. When the option is `off`, which is the default, the entries in the `/etc/snapmirror.allow` file must match the host name of the destination system reported by the `hostname` command. When the option is `on`, the source system resolves the names in the `snapmirror.allow` file to IP addresses and then checks for a match with the IP address of the requesting destination system. In this mode, literal IP addresses (for example, `123.45.67.89`) and fully qualified names (for example, `systemA.acme.com`) can be valid entries in the `/etc/snapmirror.allow` file.

Note: A local SnapMirror relationship, between two volumes on the same system, does not require an entry in the `/etc/snapmirror.allow` file.

Step

1. To configure SnapMirror to resolve host names to their IP addresses, enter the following command on the source system:

   `options snapmirror.checkip.enable on`

What the snapmirror.conf file does

The snapmirror.conf file is used to specify the details related to the copying of data, by using SnapMirror, from the source to the destination. This file resides on the destination storage system.

The `/etc/snapmirror.conf` file defines:

- The relationship between the source and the destination.
- The schedule used by the destination to copy data.
- The arguments that control SnapMirror when copying data.

Distribution of the snapmirror.conf file

You can create a single `snapmirror.conf` file for your site and copy it to all the storage systems that use SnapMirror. The `snapmirror.conf` file can contain entries pertaining to other storage systems.

For example, the `/etc/snapmirror.conf` file on systemB can contain an entry for copying a volume from system C to system D. When systemB reads the `/etc/snapmirror.conf` file, it ignores the entries for other storage systems.

Note: Each time the file is read, a warning message is displayed on the storage system console for each line that is ignored.

Limit on entries in the snapmirror.conf file

The limit on the number of entries for each system in the `/etc/snapmirror.conf` file is 1,024. Entries beyond this limit are ignored and the system console displays a warning message.

If you have an HA pair, the limit on the number of entries is shared between the two systems in the HA pair. For example, in an HA pair, if one system uses 480 entries, the other system has 544 entries available for its use.

Note: This limitation is different from the maximum number of concurrent replications you can have on a system.

Related references

Maximum number of concurrent replication operations on page 123

Editing the snapmirror.conf file

You should edit the `snapmirror.conf` file only when there are no active SnapMirror transfers for the source and destination relationships that you want to change. If SnapMirror is enabled, changes take effect within two minutes. If SnapMirror is not enabled, changes take effect immediately after you enable SnapMirror.

Before you begin

Check whether the `snapmirror.conf` file exists in the `/etc/` directory on the root volume of the destination system. If the file does not exist, create the `snapmirror.conf` file by using a text editor.

About this task

If you change the host name in a `snapmirror.conf` file entry when there are more than one active SnapMirror transfers for that specific source and destination relationship, one or more of the transfers might be aborted.

Steps

1. For each destination volume or qtree on this system, type an entry specifying the source, destination, characteristics, and schedule of the data transfer on one line using the following syntax.

   source_system*:{*source_volume* | */vol/volume_name/qtree_name*} *dest_system*:{*dest_volume* | */vol/volume_name/qtree_name*} *arguments schedule

   Note: When using a qtree as a source, you should not create the destination qtree manually. SnapMirror automatically creates the destination qtree for you, using the name you specify. However, you must specify the name and path of the destination qtree, either in the `/etc/snapmirror.conf` file or in the `snapmirror initialize` command.

   Note: You can only specify up to 254 destination qtrees for a specific volume in the `/etc/snapmirror.conf` file.

   If you want to replicate all non-qtree data within a volume to a destination qtree, use the following syntax.

   source_system*:*/vol/source_volume/-* *dest_system*:*/vol/dest_volume/qtree_name

   The hyphen (-) character indicates all non-qtree data in the specified volume.

   Note: The data in `/vol/`*`source_volume`*`/-` qtree can only be a SnapMirror source, not a destination.

2. If you want to add comments to the `/etc/snapmirror.conf` file, precede the comment with a pound (#) sign.

   #### Example

   ```
   # Replicating from systemA
   ```

3. Save edits to the file.

Related references

Syntax for snapmirror.conf file entries on page 131

Syntax for snapmirror.conf file entries

You need to use the correct syntax for each entry in the `snapmirror.conf` file, to specify the source, the destination, the options, and the schedule for a SnapMirror relationship.

The syntax for entries in the `snapmirror.conf` file is as follows.

src_system*:/vol/*src_vol*[*/src_qtree*] *dest_system*:/vol/*dest_vol*[*/dest_qtree*] *arguments schedule

The parameters in the entries are as follows:

Parameter	Description
src_system	The name of the system from which you are copying data. SnapMirror uses the `/etc/hosts` file or the database used by DNS and NIS for name resolution. When SnapMirror searches for the source system name, it should find the IP address for the source system on the network over which you want the transfer to occur. Example: Suppose you created a private network for connecting source and destination systems, and you name the interface on the source as `systemA-e0`. You must enter the interface name `systemA-e0` in the *src_system* field.
src_vol/*src_qtree*	The name of the volume or qtree that you are copying. Use only the volume name for volumes. Use the full path name for qtrees. Example: If the name of the volume is `vol1`, enter `vol1` in the *src_vol/src_qtree* field. If the name of the qtree is `qtree3`, and it is contained in the volume `vol3`, enter the full path, `/vol/vol3/qtree3`, in the *src_vol/src_qtree* field.
dest_system	The host name of the system to which the data is copied. The name you use must be the exact host name of the destination system. **Note:** The *dest_system* field is case-sensitive. You can use the `hostname` command on the destination system to determine what you can enter in this field. Example: If the name of the destination system is `systemA`, enter `systemA` in the *dest_system* field.
-	Using a hyphen (`-`) specifies the use of default parameters.
dest_vol [/*dest_qtree*]	The name of the destination volume or qtree to which you are copying data. Use only the volume name for volumes. Use the full path name for qtrees. Example: If the name of the volume is `vol1`, enter `vol1` in the *dest_vol*[/*dest_qtree*] field. If the name of the qtree is `qtree4`, and it is in `vol2`, enter the full path, `/vol/vol2/qtree4`, in the *dest_vol*[/*dest_qtree*] field.

Parameter	Description
`kbs=`*`kbs`*	Maximum transfer speed, in kilobytes per second, that Data ONTAP can use to transfer data. The `kbs` and `restart` arguments are expressed as a comma-separated list of *`name=value`* pairs, with no spaces. For example: **`kbs=2000,restart=always`**
`restart={ never | always }`	SnapMirror uses the restart mode to continue an incremental transfer from a checkpoint, if it is interrupted. The options are: • `never`—Restarts the transfer from the beginning and never from the point of interruption. This mode is useful if you must have the latest data on the destination. • `always`—Restarts the transfer, if possible, from the point of interruption. This mode is useful for copying large volumes. If no option is specified for the restart mode, then the system assigns it the `default` option, which is set to `always`. The transfer always restarts from the point of interruption unless it conflicts with a scheduled transfer, which is configured in the `snapmirror.conf file`. In case of a conflict, the transfer restarts from the beginning.

Parameter	Description
`cksum=` { `none` | `crc32c` | `crc32c_header_only` }	Selects the checksum algorithm that is used to check the data replicated by SnapMirror. `none` specifies that SnapMirror does not perform a checksum for the data. This option assumes that the network used by SnapMirror is reliable and delivers the data packets from the source to the destination without errors. `crc32c` specifies that the data being transferred by SnapMirror is checked by an application-level 32-bit cyclic redundancy check. This check ensures that any transmission errors not corrected at the network level are detected by SnapMirror. However, using this option adds to the processing load of both the source and destination systems. `crc32c_header_only` specifies that only the headers of the SnapMirror data packets are checked by the cyclic redundancy check. This check enhances the reliability of volume SnapMirror, with negligible performance impact. This option can be used only with asynchronous volume SnapMirror. **Note:** If the source of the SnapMirror relation is specified through the snapmirror.conf file, the default value of this parameter will be `none`. If the source is specified through the command line interface, the default value will be `crc32`.
`visibility_interval={`*x*`s` | *x*`m` | *x*`h}`	Determines the amount of time before an automatic Snapshot copy is created on the source volume that is being replicated using synchronous or semi-synchronous SnapMirror. When replicating synchronously or semi-synchronously, using SnapMirror, changes to the source volume do not show immediately on the destination volume, even though the changes have been replicated. The changes are shown only after the source system takes an automatic Snapshot copy of the source volume. This event occurs every three minutes by default. You can change the interval for automatic Snapshot copies, but performance can degrade if you set smaller intervals because Snapshot copies are taken more often. The smallest interval you can set is 30 seconds. **Note:** The `s`, `m`, and `h` suffixes specify seconds, minutes, and hours, respectively.

Parameter	Description
`wsize=`*`size`*	Determines the TCP window size used by the SnapMirror relationship. **Note:** The TCP window size is specified in bytes.
`connection_mode={inet6 | inet | unspec}`	Specifies whether the relationship uses an IPv6 or IPv4 connection. • `unspec` is the default value of the `connection_mode` option. An IPv6 connection is attempted first. If an IPv6 connection is not established, then an IPv4 connection is attempted. • `inet6` specifies the use of an IPv6 connection. • `inet` specifies the use of an IPv4 connection.

`schedule` specifies the time and frequency of the SnapMirror updates on the destination. Specifying the schedule is mandatory.

`schedule` consists of four space-separated fields in the following order:

`minute hour dayofmonth dayofweek`

- *`minute`* can be a value from 0 to 59.
- *`hour`* can be a value from 0 to 23.
- *`dayofmonth`* can be a value from 1 to 31.
- *`dayofweek`* can be a value from 0 (Sunday) to 6 (Saturday).

Multiple values, separated by commas, can be entered for any field.

All possible values for a field can be applied with an asterisk (*). If you specify an asterisk in each field of the schedule, SnapMirror updates the destination every minute.

A single dash (-) in any field means “never” and prevents this schedule entry from executing. (This option is useful if you want the server to appear in the `/etc/snapmirror.conf` file so that `snapmirror update` can find it, but you do not want the SnapMirror scheduler to run automatically.)

A range of values for any field can be indicated with a low value and a high value separated by a dash. For example, you can indicate that you want an update every hour from 8:00 a.m. to 5:00 p.m. by entering this value in the hour field:

A range of values followed by a slash and a number indicates the frequency of the update. For example, you can indicate that you want an update every five minutes by entering this value in the minute field:

```
0-59/5
```

Using `sync` or `semi-sync` instead of the four space-separated fields specifies synchronous or semi-synchronous replication, respectively. For more information, see the na_snapmirror.conf(5) man page.

Example of snapmirror.conf file schedule entries

Suppose you create a private network between `systemA` and `systemB`. In the `/etc/hosts` file on `systemA`, you specify the host name for the interface as `systemA-e0`, and you ensure that `systemA-e0` is also in the `/etc/hosts` file on `systemB`, the destination system. You want to copy `vol0` of `systemA` to `vol1` of `systemB` over the private network every Monday, Wednesday, and Friday at 11 p.m. You also want to use the default for the arguments field.

To copy the data over the private network every Monday, Wednesday, and Friday at 11 p.m., you would enter the following in the `/etc/snapmirror.conf` file.

```
systemA-e0:vol0 systemB:vol1 - 0 23 * 1,3,5
```

The following table indicates what the entry in each field in the example means.

Entry	Description
`systemA-e0`	Name of the source system, along with the interface to be used
`vol0`	Source volume
`systemB`	Name of the destination system
`vol1`	Destination volume
`-`	Use default values for arguments
`0`	Starts a SnapMirror update at the 0th minute of the hour(s), as specified in the next (hour) field
`23`	Starts a SnapMirror update at 11p.m.
`*`	Starts a SnapMirror update on all applicable days of the month, as specified in the next field
`1,3,5`	Starts a SnapMirror update on Monday, Wednesday, and Friday

Example of a snapmirror.conf file entry with an IPv6 address

The following `snapmirror.conf` file entry shows the use of an IPv6 address to specify the source system for a given SnapMirror relationship. Note that the IPv6 address is enclosed within square brackets.

```
[fd20:8b1e:b255:4166:2a0:98ff:fe07:23f3]:src_vol dst_system:dst_vol -
15 2 * *
```

Example of a snapmirror.conf file entry with space character and double quotes in qtree name

The following `snapmirror.conf` file entry shows how qtree names with space character and double quotes must be specified. This entry indicates that data from the qtree `vol/vol1/x y"z` on `systemA` is replicated to the qtree `vol/vol2/x y"z` on `systemB`.

```
"systemA:/vol/vol1/x y""z" "systemB:/vol/vol2/x y""z" - * * * *
```

Note: Qtree names with space character or double quotes might not work with previous versions of Data ONTAP. Therefore, when reverting to a previous version of Data ONTAP, you must check for compatibility.

Example setting maximum update speed

The following line in an `/etc/snapmirror.conf` file sets the speed to 2000 kilobytes per second.

```
systemA:vol0 systemA:vol1 kbs=2000 15 * * 1,2,3,4,5
```

Note: The specified transfer speed might not be achievable because transfer speed is limited by factors such as network bandwidth.

Example specifying always restart

The following line in an `/etc/snapmirror.conf` file sets the *`restart`* value to `always`.

```
systemA:vol0 systemA:vol1 kbs=2000,restart=always 15 * * 1,2,3,4,5
```

Example specifying default values for maximum speed and restart

If you set the value of only one argument (`kbs` or `restart`), the other argument uses the default value. If you want to use the default argument for both values, enter a dash (-).

The following line in an `/etc/snapmirror.conf` file sets both arguments to the default value.

```
systemA:vol0 systemA:vol1 - 15 * * 1,2,3,4,5
```

Example specifying 15-minute interval updates during specific hours

If you want to schedule an update every afternoon at 1:00, 1:15, 1:30, 1:45, 5:00, 5:15, 5:30, 5:45, 7:00, 7:15, 7:30, and 7:45, you should enter the following in the schedule field.

```
systemA:vol0 systemA:vol1 - 0,15,30,45 13,17,19 * *
```

Note: An update is started when the current time matches a value in all four fields. Be careful that a value in the day of the month field does not exclude a value in the day of the week field. For example, the following schedule updates the destination every afternoon at 1:00, 1:30, 3:00, 3:30, 5:00, and 5:30, but only when the first day of the month falls on a Monday.

```
systemA:vol0 systemA:vol1 - 0,30 13,15,17 1 1
```

Note: The schedule represents the goal of the SnapMirror feature. Factors that might prevent SnapMirror from updating every minute include resource limitations or network connectivity. If an update is in progress when another is scheduled to occur, SnapMirror will start another transfer as soon as the first transfer is complete. However, if more than one update gets queued while a transfer is in progress, SnapMirror only performs the update queued last. The other updates become obsolete due to the later update.

Related concepts

Synchronous SnapMirror modes on page 79
SnapMirror support for IPv6 on page 103

Related tasks

Adjusting the TCP window size for a SnapMirror relationship on page 158

Scheduled updates for volumes or qtrees

You can edit the destination's `snapmirror.conf` file to change or turn off scheduled updates for a particular volume or qtree if you decide that there is no need to update the destination.

You might want to change the time or frequency of scheduled updates if the pattern of use or the configuration of the systems has changed. Or, if you want to use the volume for a different purpose, you can change the destination to a writable volume.

Note: Editing entries in the destination's `snapmirror.conf` file to turn off scheduled updates does not change the destination to a writable volume. If you want to change the destination to a writable volume or qtree, you use the `snapmirror break` command to turn the destination into a

writable volume or qtree and the `snapmirror release` command to allow SnapMirror to delete the Snapshot copies it no longer needs on the source.

You can edit the destination's `snapmirror.conf` file to turn off or change scheduled updates at any time, even when data transfer is underway. The destination remains the same as before the transfer. The Snapshot copy taken in the source for the data transfer remains, but it can be deleted and replaced by a new Snapshot copy the next time the destination is updated.

Related concepts

Conversion of a destination to a writable volume or qtree on page 168

Related tasks

Converting a SnapMirror destination to a writable volume or qtree on page 169
Releasing partners from a SnapMirror relationship on page 177

Changing scheduled updates for one volume or qtree

You can change scheduled updates for one volume or qtree by editing the `snapmirror.conf` file on the destination.

Step

1. In the destination's `snapmirror.conf` file, edit the destination volume or the schedule information to specify the configuration you want.

Example

The following example shows the original update schedule:

```
systemA:vol0 systemA:vol1 - 0 23 * 1,3,5
systemA:vol1 systemB:vol6 - 0 23 * 1,3,5
```

The following example shows the changed update schedule

```
systemA:vol0 systemA:vol2 - 0 23 * 1,3,5
systemA:vol1 systemB:vol6 - 0 23 * 2,4,6
```

Turning off SnapMirror updates

You can use the `snapmirror off` command to turn off updates, both scheduled and manual, for the entire system at any time, even when copying is underway. Any active transfer is aborted when you turn off SnapMirror for the system. The destination remains unchanged after you turn off updates.

About this task

If you turn off SnapMirror for the system, any active SMTape operations, such as `smtape backup` and `smtape restore`, are aborted.

This process affects all SnapMirror transfers for the system, whether the system is the source or the destination of the SnapMirror relationship.

Steps

1. Enter the following command on both the source system and destination system to disable SnapMirror:

   `options snapmirror.enable off`

   Alternatively, you can use the `snapmirror off` command to turn off SnapMirror.

   If SnapMirror is currently transferring data from one volume or qtree to another, the transfer aborts immediately. The destination remains the same as it was before the transfer. The Snapshot copy created in the source volume for the data transfer remains.

   SnapMirror stops monitoring the `/etc/snapmirror.conf` file for changes.

   Entering the `snapmirror off` command on the destination system alone does not affect SnapMirror on the source system. Other systems can continue to copy data from the source system.

   Note: Both the `snapmirror off` command and the `snapmirror.enable off` option are persistent across reboots.

2. If the `snapmirror on` command is in the `/etc/rc` file, remove the command (to keep the current setting after reboot).

   Otherwise, the setting in the `/etc/rc` file overrides the command you entered.

Turning off scheduled updates for one volume or qtree

You can turn off scheduled updates for one volume or qtree, by editing the `snapmirror.conf` file.

Step

1. Either delete the entry in the `snapmirror.conf` file or change the entry by:

   - Commenting out the entry by preceding it with a pound sign (#).

Example

```
systemA:vol0 systemA:vol1 - 0 23 * 1,3,5
systemA:vol1 systemB:vol6 - 0 23 * 1,3,5
#systemB:vol1 systemC:vol2 - 0 23 * 1,3,5
```

- Putting a dash (-) in one of the schedule fields (minute/hour/dayofmonth/dayofweek).

  Note: Deleting or commenting out a destination or putting a dash in one of the schedule fields of a destination in the `/etc/snapmirror.conf` file does not prevent you from performing manual updates to that destination.

Example

```
systemA:vol0 systemA:vol1 - 0 23 * 1,3,5
systemA:vol1 systemB:vol6 - 0 23 * 1,3,5
systemB:vol1 systemC:vol2 - - 23 * 1,3,5
```

Manual update of a SnapMirror destination

SnapMirror automatically updates the destination according to the update schedule specified in the `snapmirror.conf` file. You can also initiate updates manually by using the `snapmirror update` command.

You might need a manual update to prevent data loss due to an upcoming power outage, scheduled maintenance, or data migration. You can also include the `snapmirror update` command in an external script if you want to drive updates using that script.

Performing a manual SnapMirror update

You can perform an unscheduled SnapMirror incremental update, independent of the schedule in the `/etc/snapmirror.conf` file.

Step

1. From the destination system, enter the following command:

   snapmirror update [*options*] [*dest_system*:] {*dest_volume* | /vol/*dest_volume/qtree_path*}

   options can be one or more of the following:

   - `-k` *n* sets the maximum transfer speed to *n* kilobytes per second. This option has the same effect as the `kbs` argument in the `/etc/snapmirror.conf` file.

- `-s` *snapshot_name* specifies an existing (qtree only) source Snapshot copy to be transferred, rather than a Snapshot copy taken by the source. SnapMirror does not lock or delete this Snapshot copy.

  Note: *snapshot_name* cannot be `minutely.`*x*, `hourly.`*x*, `nightly.`*x*, `weekly.`*x*, `snapshot_for_backup.`*x* or `snapshot_for_volcopy.`*x*. You must rename such Snapshot copies on the source and then copy them.

- `-c` *snapshot_name* creates a Snapshot copy named *snapshot_name* of a qtree on the destination after the next update (so that it does not compete with any ongoing updates). SnapMirror does not lock or delete this Snapshot copy.

  Note: *snapshot_name* cannot be `minutely.`*x*, `hourly.`*x*, `nightly.`*x*, or `weekly.`*x*, because these names are reserved for scheduled Snapshot copies.

- `-S` [*source_system*:]*source_volume* | *qtree_path* specifies the source system and volume for the update. *source_volume* is the volume you want to copy.
  The source specified by the `-S` option must match an entry for *source_volume* in the `/etc/snapmirror.conf` file. If an entry exists but does not match, the operation displays an error message and terminates. If there is no entry for the specified source volume, the command runs.

  Note: If the `-S` option is not set, the source must be specified in the `/etc/snapmirror.conf` file. If it is not specified, the operation displays an error message and terminates.

dest_system specifies the name of the destination system.

dest_volume specifies the destination volume. If it is a scheduled destination of a local source volume, as specified in the `/etc/snapmirror.conf` file, that source volume is considered to be the source. If the destination volume specified is not in a scheduled relationship, then the `-S` option must be used to provide a source.

Example

Using the following command, SnapMirror updates the destination (vol2 on systemB) from the source specified in the `/etc/snapmirror.conf` file.

```
systemB> snapmirror update systemB:vol2
```

Example

Using the following command, SnapMirror updates the qtree destination on `systemB:/vol/vol2/usersbak` from the source qtree on `systemA:/vol/vol1/users`.

```
systemB> snapmirror update -S systemA:/vol/vol1/users systemB:/vol/
vol2/usersbak
```

Creating extra backup Snapshot copies for SnapMirror qtrees

You might want to create an extra backup Snapshot copy on the source and destination in a qtree SnapMirror relationship. You can create a manual Snapshot copy using the `snap create` command. Then, you can use the `-c` and `-s` options of the `snapmirror update` command together to establish an extra backup Snapshot copy on both sides of a SnapMirror qtree relationship.

About this task

These Snapshot copies can serve as the newest common Snapshot copy in case a base Snapshot copy is accidentally deleted. You can also use them to resynchronize a SnapMirror qtree relationship to an earlier resynchronization point.

Steps

1. Enter the following command:

   snap create *vol_name snapshot_name*

   vol_name is the name of the volume whose Snapshot copy you want to create.

   snapshot_name is the name of the Snapshot copy.

   #### Example

   ```
   systemB> snap create vol2 my_snap
   ```

   This creates a Snapshot copy, my_snap, of the volume vol2.

2. Enter the following command:

   snapmirror update -S *source* -s *src_snap* -c *dest_snap destination*

   source is the name of the source.

   src_snap is the name of the Snapshot copy on the source.

   dest_snap is the name of the Snapshot copy on the destination.

   destination is the name of the destination.

   #### Example

   ```
   systemA> snapmirror update -S systemB:/vol/vol2/qtree1 -s my_snap -c
   my_dest_snap vol/vol4/qtreeSafe
   ```

What happens after SnapMirror makes incremental updates to the destination

The destination reflects the changes on the source after SnapMirror completes the transfer. If the SnapMirror transfer is incomplete or interrupted, the changes on the destination are not visible till the transfer is complete. After SnapMirror completes the destination update, you can see the changes when you open the file.

Note: SnapMirror automatically deletes old Snapshot copies that are no longer necessary for updating data.

SnapMirror over multiple paths

You might want more than one physical path for a SnapMirror relationship. SnapMirror supports up to two paths for a particular SnapMirror relationship.

When using multiple paths, you need to set up the configuration in one of the following ways:

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

The paths can be Ethernet, Fibre Channel, or a combination of Ethernet and Fibre Channel. The two paths can be used in one of these two modes:

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

  Note: The failover mode using only one pair of connections is not supported with SnapMirror network compression.

Setting up a multipath SnapMirror relationship

You can use multiple paths between the source and destination systems for baseline initialization.

About this task

You can set up SnapMirror to use multiple paths at the outset. You can also convert a single path SnapMirror relationship to use multiple paths.

Steps

1. Ensure that you have two valid paths using the `ping` command from the source system to each of the IP addresses on the destination system.
2. On the source system console, use the `options snapmirror.access` command to specify the host names of systems that are allowed to copy data directly from the source system.

   #### Example

   `options snapmirror.access host=d_systemA`
3. Edit the `snapmirror.conf` file on the destination system to add an entry that defines the mode of the connection and what the two connections are. The format of the entry is as follows:

   `name=`*`mode`*`(`*`src_system-e0`*`,`*`dst_system-e0`*`)(`*`src_system-e1`*`,`*`dst_system-e1`*`)`

   where *`mode`* is either `multi` or `failover`. See the na_snapmirror.conf(5) man page for details.
4. Edit the `/etc/snapmirror.conf` file on the destination system to specify the volumes and qtrees to be copied and the schedule (*`minute`*, *`hour`*, *`day_of_month`*, *`day_of_week`*, `sync`, or `semi-sync`) on which the destination is updated. Use the connection name specified in previous step as the source system.

Related references

Methods for specifying destination systems on the SnapMirror source on page 127

Converting a single-path SnapMirror relationship to multipath

You can set up SnapMirror to use multiple paths at the outset. You can also convert a single-path SnapMirror relationship to use multiple paths.

Steps

1. Ensure that you have two valid paths using the `ping` command from the source system to each of the IP addresses on the destination system.
2. Edit the `snapmirror.conf` file on the destination system to add a connection line that defines the mode of the connection and what the two connections are. The format of the line is as follows:

   `name=`*`mode`*`(`*`src_system-e0`*`,`*`dst_system-e0`*`)(`*`src_system-e1`*`,`*`dst_system-e1`*`)`

   `mode` is either `multi` or `failover`. See the na_snapmirror.conf(5) man page for details.
3. In the same `snapmirror.conf` file, edit the schedule entry to reflect the new connection name as the source system.

   Note: Multiple paths are supported by SnapMirror running asynchronously and synchronously. The following are examples of implementing multiple paths using synchronous SnapMirror.

Example

You want to synchronously replicate volume `vol1` on a system called `NYC` to volume `vol1` on a system called `Newark`. You require two physical paths between source and destination systems for each synchronously-mirrored volume. You have two network interface cards on each system. You named the two interfaces on the `NYC` system `NYC-pri` and `NYC-sec`, and the two on the `Newark` system `Newark-pri` and `Newark-sec`. To implement multiple paths in the `failover` mode, you edit the `snapmirror.conf` file on `Newark` to include the following two lines.

```
NYC-Newark=failover(NYC-pri,Newark-pri)(NYC-sec,Newark-sec)
NYC-Newark:vol1 Newark:vol1 - sync
```

Example

If `NYC-pri` and `Newark-pri` are Fibre Channel NIC adapters and you want to replicate data using both connections, you follow the procedure to configure Fibre Channel NIC adapters for SnapMirror. Then, you edit the `snapmirror.conf` file on `Newark` to include the following two lines to implement multiple paths in `multi` mode.

```
NYC-Newark=multi(NYC-pri,Newark-pri)(NYC-sec,Newark-sec)
NYC-Newark:vol1 Newark:vol1 - sync
```

Related tasks

Configuring SnapMirror over Fibre Channel on page 212

SnapMirror network compression

SnapMirror network compression compresses the data stream on the source system, transfers the compressed data stream over the network, and uncompresses the data stream on the destination system before writing it to disk.

SnapMirror network compression also increases resource utilization on both the SnapMirror source and destination systems. Therefore, you need to evaluate the resource usage and benefits before deploying compression. For example, compression might not be useful for a high-bandwidth, low-latency connection. But it can be useful for connections that have relatively low bandwidth, such as WAN connections.

SnapMirror network compression is supported only for asynchronous volume SnapMirror.

Enabling SnapMirror network compression

You can enable network compression for a volume SnapMirror relationship by specifying the compression option in the `snapmirror.conf` file.

Before you begin

- You must have ensured that the compression feature is supported only for asynchronous volume SnapMirror.
- You must have ensured that the SnapMirror destination system should be using a Data ONTAP release that supports SnapMirror network compression. Data ONTAP 7.3.2 and later releases of the 7.x release family, and Data ONTAP 8.0.1 and later releases of the 8.x release family support SnapMirror network compression.

About this task

There are certain considerations for enabling and disabling SnapMirror network compression.

- To enable compression, you must specify a name for the connection between a SnapMirror source and destination system pair.
- You can enable or disable compression for both initial and incremental SnapMirror transfers.
- You need to enable the compression feature on a per-relationship basis.
- You cannot enable or disable compression for an active SnapMirror transfer.
  However, the transfers that get activated after you edit the `snapmirror.conf` file use the updated compression configuration. If you want the change in the option to be applied immediately, enter the following command:

  `snapmirror on`

Steps

1. Open the `snapmirror.conf` file.
2. Specify the SnapMirror relationship by using the following command:

   ***`connection_name`*`=mode(`*`src_system-e0,dst_system-e0`*`)(`*`src_system-e1,dst_system-e1`*`)`**

   ***`connection_name`*`:`*`src_vol dst_system`*`:`*`dst_vol - * * * *`**

   `connection_name` is the name of this specific connection between a SnapMirror source and destination system pair.

   `mode` is the mode of the connection. It can either be `multi` or `failover`.

   `src_system-e0` and *`src_system-e1`* are the names of the SnapMirror source systems.

dst_system-e0 and *dst_system-e1* are the names of the SnapMirror destination systems.

src_vol is the path of the SnapMirror source volume.

dst_vol is the path of the SnapMirror destination volume.

3. Enable compression by adding the `compression=enable` option to the SnapMirror relationship specification.

   The syntax of the entries in the `snapmirror.conf` file required for enabling compression is given in the following lines:

   ***connection_name*=mode(*src_system-e0*,*dst_system-e0*)(*src_system-e1*,*dst_system-e1*)**

   ***connection_name*:*src_vol* *dst_system*:*dst_vol* compression=enable * * * *

   Example

   ```
   conxn_1=multi(src_system,dst_system)
   conxn_1:src_vol dst_system:dst_vol compression=enable * * * *
   ```

   Example

   The following example shows the SnapMirror relationship status, SnapMirror log files on both the source and the destination systems, and the compression ratio:

   SnapMirror relationship status

   ```
   f3070-202-xx*> snapmirror status -l
   Snapmirror is on.

   Source:                 f3070-202-xx:src
   Destination:            f3070-202-yy:dst
   Status:                 Transferring
   Progress:               1687772 KB
   Compression Ratio:      2.6 : 1
   State:                  Source
   Lag:                    -
   Mirror Timestamp:       -
   Base Snapshot:          -
   Current Transfer Type:  -
   Current Transfer Error: -
   Contents:               -
   Last Transfer Type:     -
   Last Transfer Size:     -
   Last Transfer Duration: -
   Last Transfer From:     -
   ```

   SnapMirror log file on source system

```
log Wed May 19 07:04:17 GMT FILER_REBOOTED
sys Wed May 19 07:10:31 GMT SnapMirror_on (registry)
src Wed May 19 08:55:56 GMT f3070-202-xx:src f3070-202-yy:dst Request
(10.**.***.**)
src Wed May 19 08:55:56 GMT f3070-202-xx:src f3070-202-yy:dst Abort
(connection to destination dropped)
src Wed May 19 09:08:54 GMT f3070-202-xx:src f3070-202-yy:dst Request
(10.**.***.**)
slk Wed May 19 09:08:55 GMT state.softlock.src.00000022.002.f3070-202-
yy:dst Softlock_add (Transfer)
src Wed May 19 09:08:55 GMT f3070-202-xx:src f3070-202-yy:dst Start
src Wed May 19 09:10:16 GMT f3070-202-xx:src f3070-202-yy:dst End
(1687772 KB, Compression 2.6 : 1)
```

SnapMirror log file on destination system

```
sys Mon May 17 03:58:06 GMT SnapMirror_off (shutdown)
log Mon May 17 04:04:06 GMT FILER_REBOOTED
slk Mon May 17 10:33:32 GMT state.softlock.newvol.
00000012.002.snapmirror_tape_f8568c4a-6331-11df-80d6-00a0980c07a7
Softlock_add (Transfer)
src Mon May 17 10:33:32 GMT f3070-202-40:newvol f3070-202-yy:/etc/pqr
Request (Store)
src Mon May 17 10:33:32 GMT f3070-202-40:newvol f3070-202-yy:/etc/pqr
Start
src Mon May 17 10:33:32 GMT f3070-202-40:newvol f3070-202-yy:/etc/pqr
End (344 KB)
sys Wed May 19 06:45:11 GMT SnapMirror_off (shutdown)
log Wed May 19 06:48:07 GMT FILER_REBOOTED
sys Wed May 19 08:53:44 GMT SnapMirror_on (registry)
dst Wed May 19 08:59:40 GMT conn:src f3070-202-yy:dst Request
(Initialize)
dst Wed May 19 08:59:42 GMT conn:src f3070-202-yy:dst Abort
(replication transfer failed to complete)
dst Wed May 19 09:12:38 GMT conn:src f3070-202-yy:dst Request
(Initialize)
dst Wed May 19 09:12:40 GMT conn:src f3070-202-yy:dst Start
dst Wed May 19 09:14:01 GMT conn:src f3070-202-yy:dst End (1687772
KB, Compression 2.6 : 1)
```

The data shown in the above log files is pre-compression data.

Note: To disable compression, remove the `compression=enable` option from the SnapMirror relationship entry in the `snapmirror.conf` file.

Compression for a multipath SnapMirror relationship

The syntax of the entries in the `snapmirror.conf` file required for enabling compression for a multipath SnapMirror relationship is given in the following lines:

***connection_name*=multi(*src_ip1*,*dst_ip1*)(*src_ip2*,*dst_ip2*)**

***connection_name*:*src_vol* *dst_system*:*dst_vol* compression=enable * * * *

src_ip1 and *src_ip2* are the IP addresses of the SnapMirror source system.

dst_ip1 and *dst_ip2* are the IP addresses of the SnapMirror destination system.

```
conxn_2=multi(10.72.146.74,10.79.106.40)(10.72.147.74,10.79.107.40)
conxn_2:src_vol dst_system:dst_vol compression=enable * * * *
```

Viewing SnapMirror network compression ratio

For a SnapMirror relationship that uses the compression feature, you can view the compression ratio for an active transfer.

Before you begin

Ensure that the compression feature is enabled. To enable the compression feature, you need to add the appropriate option to the SnapMirror relationship details, in the `snapmirror.conf` file.

About this task

- The compression ratio is displayed only for a SnapMirror transfer that is in progress.
- The compression ratio is also logged in the SnapMirror log file.

Step

1. To view the compression ratio for an active SnapMirror transfer, enter the following command on the SnapMirror destination:

   snapmirror status -l *dst_vol*

   dst_vol is the destination volume.

Viewing SnapMirror network compression ratio

```
dst_system> snapmirror status -l dst_vol
Snapmirror is on.

Source:                 src_system:src_vol
Destination:            dst_system:dst_vol
Status:                 Transferring
Progress:               24 KB
Compression Ratio:      4.5 : 1
State:                  -
Lag:                    -
Mirror Timestamp:       -
Base Snapshot:          -
Current Transfer Type:  Initialize
Current Transfer Error: -
Contents:               -
```

```
Last Transfer Type:     Initialize
Last Transfer Size:     132 MB
Last Transfer Duration: 00:00:27
Last Transfer From:     src_system:src_vol
```

Related tasks

Enabling SnapMirror network compression on page 147

Checking SnapMirror data transfer status

You need to check the data transfer status, by using the `snapmirror status` command, to determine the status of all existing SnapMirror relationships on the system.

Step

1. Enter the following command:

   `snapmirror status [`*`options`*`] [[`*`system`*`:] [`*`path`*`] ...]`

   `options` can be one of the following.

   - `-l` displays the long format of the output, which contains more detailed information.
   - `-q` displays which volumes or qtrees are quiesced or quiescing.
   - `-t` displays which volumes or qtrees are active.

   `system` is the name of the source system.

   `path` is the name of the source volume or the path to and name of the source qtree.

   Note: When you use the `-t` option, the output displays the active relationships. A relationship is considered active if the source or destination is involved in one of the following:

   - Data transfer to or from the network.
   - Reading or writing to a tape device.
   - Waiting for a tape change.
   - Performing local on-disk processing or cleanup.

Result

If no arguments or options are given, SnapMirror displays a message that indicates whether a transfer is in progress, how much of the data transfer has been completed, the state of the destination, and the amount of time since the last Snapshot copy was created and transferred successfully.

Related tasks

Stabilizing destinations before a Snapshot copy on page 174

What SnapMirror status check shows

The SnapMirror status check shows information about SnapMirror transfers for volumes or qtrees.

If you check the status and you enabled SnapMirror, messages similar to the following are displayed:

```
systemA> snapmirror status

Snapmirror is on.
Source                  Destination            State         Lag       Status
systemA:vol0            systemA:vol1           Snapmirrored  02:25:11  Transferring (60 MB
done)
systemB:/vol/vol1/qt3   systemB:/vol/vol3/qt3  Quiesced      00:01:15  Idle
```

You see a status report for any SnapMirror source that contains the base Snapshot copy, and for any destination in a current SnapMirror relationship or listed in the `/etc/snapmirror.conf` file. Destinations that were broken through the `snapmirror break` command but still contain the base Snapshot copy are listed.

If you check the status of data transfer and you did not enable SnapMirror, the following message is displayed:

```
systemA> snapmirror status

Snapmirror is off.
```

Note: The status of SnapMirror relationships, if any, are still displayed, as shown in the preceding example.

Example

With no options, the information displayed by the `snapmirror status` command looks similar to the following:

```
systemB> snapmirror status

Snapmirror is on.
Source                  Destination            State         Lag       Status
systemA:vol0            systemB:vol2           Broken-off    29:09:58  Idle
systemC:vol0            systemB:vol3           Snapmirrored  00:09:53  Idle with restart
checkpoint (23 MB done)
systemC:vol4            systemB:vol5           Snapmirrored  00:04:58  Transferring (36
MB done)
systemA:/vol/vol1/qt5   systemB:/vol/vol4/qt5  Quiesced      00:05:12  Idle
systemC:/vol/vol2/qt1   systemB:/vol/vol1/qt2  Snapmirrored  00:02:33  Quiescing
```

Example

With the `-l` option, the configuration described in the previous example looks similar to the following:

```
systemB> snapmirror status -l

Snapmirror is on.
Source:                  systemA:vol0
Destination:             systemB:vol2
Status:                  Syncing
Progress:                60 KB
State:                   Source
Lag:                     00:01:17
Mirror Timestamp:        Sat Jul 15 00:50:02 GMT 2000
Base Snapshot:           tpubs-f720(0016791363)_vol2.1249
Current Transfer Type:   -
Current Transfer Error:  -
Contents:                -
Last Transfer Type:      Update
Last Transfer Size:      1052 KB
Last Transfer Duration:  00:00:02
Last Transfer From:      systemA:vol0

Source:                  systemC:vol0
Destination:             systemB:vol3
Status:                  Idle with restart checkpoint
Progress                 23552 KB done
State:                   Snapmirrored
Lag:                     00:09:53
Mirror Timestamp:        Sun Jul 16 05:50:07 GMT 2000
Base Snapshot:           system2(0016778780)_vol3.985
Current Transfer Type:   -
Current Transfer Error:  Abort by user
Contents:                Replica
Last Transfer Type:      Update
Last Transfer Size:      432000 KB
Last Transfer Duration:  00:01:23
Last Transfer From:      systemC:vol0

Source:                  systemC:vol4
Destination:             systemB:vol5
Status:                  Transferring
Progress                 36864 KB done
State:                   Snapmirrored
Lag:                     00:04:58
Mirror Timestamp:        Sun Jul 16 05:55:02 GMT 2000
Base Snapshot:           systemB(0016778780)_vol5.57843
Current Transfer Type:   Scheduled
Current Transfer Error:  -
Contents:                Replica
Last Transfer Type:      Scheduled
Last Transfer Size:      345000 KB
Last Transfer Duration:  00:03:23
Last Transfer From:      systemB:vol4

Source:                  systemC:/vol/vol1/qt5
Destination:             systemB:/vol/vol4/qt5
Status:                  Idle
Progress                 -
State:                   Quiesced
Lag:                     0:05:12
Mirror Timestamp:        Sun Jul 16 05:56:12 GMT 2000
Base Snapshot:           systemB(0016778780)_vol_vol4_qt5.54
Current Transfer Type:   -
Current Transfer Error:  -
Contents:                Replica
Last Transfer Type:      Scheduled
Last Transfer Size:      45000 KB
```

```
Last Transfer Duration: 0:00:12
Last Transfer From:     systemC:/vol/vol1/qt5

Source:                 systemC:/vol/vol2/qt1
Destination:            systemB:/vol/vol4/qt2
Status:                 Quiescing
Progress                -
State:                  Snapmirrored
Lag:                    0:02:33
Mirror Timestamp:       Sun Jul 16 05:58:20 GMT 2000
Base Snapshot:          systemB(0016778780)_vol_vol4_qt2.122
Current Transfer Type:  -
Current Transfer Error: -
Contents:               Transitioning
Last Transfer Type:     Scheduled
Last Transfer Size:     80 KB
Last Transfer Duration: 0:00:08
Last Transfer From:     systemC:/vol/vol2/qt1
```

Example

With the `-q` option, the output looks similar to the following:

```
systemC> snapmirror status -q

Snapmirror is on.
vol3 is quiesced
vol2 has quiesced/quiescing qtrees:
         /vol/vol2/qt1 is Quiescing
         /vol/vol2/qt2 is Quiesced
```

Related references

Information messages in the SnapMirror status check on page 154

Information messages in the SnapMirror status check

The `snapmirror status` command displays information messages.

The SnapMirror status entries on the source category are as follows:

Source entry	Description
system:vol	The source system and source volume
system:qtree_path	The source system and qtree path
-	Either the SnapMirror destination is an imported volume without an entry in the `/etc/snapmirror.conf` file, or a Data ONTAP upgrade is in progress
system:tape_device	The source tape device; transfer from this tape device is still in progress

Source entry	Description
base snapshot	The name of the base Snapshot copy from which a completed transfer was made, if the source is a tape device and there is no entry in the `/etc/snapmirror.conf` file for the destination

The SnapMirror status entries on the destination category are as follows:

Destination entry	Description
system:vol	The destination system and volume
system:qtree_path	The destination system and qtree path
system:tape_device	The destination tape device; transfer to this tape device is in progress
tape_destination	Displayed after a transfer to tape is finished. The `snapmirror destinations` command also displays this information

The SnapMirror status entries on the state category are as follows:

State entry	Description
Uninitialized	The destination is listed in the `/etc/snapmirror.conf` file, but the volume or qtree has not been initialized or the destination is being initialized.
Snapmirrored	The volume or qtree is in a SnapMirror relationship.
Broken-off	The destination was in a SnapMirror relationship, but a `snapmirror break` command made the volume or qtree writable. This state is reported as long as the base Snapshot copy is still present in the volume. If the Snapshot copy is deleted, the state is listed as "uninitialized" if the destination is in the `/etc/snapmirror.conf` file or is no longer listed if it is not. A successful `snapmirror resync` command restores the `snapmirrored` status.
Quiesced	SnapMirror is in a consistent internal state and no SnapMirror activity is occurring. In this state, you can create Snapshot copies knowing that all destinations are consistent. The `snapmirror quiesce` command brings the destination into this state. The `snapmirror resume` command restarts all SnapMirror activities.
Unknown	The destination volume or the volume that contains the destination qtree is in an unknown state. It might be offline or restricted.
Source	When the `snapmirror status` command is run on the source system and the destination is on another system, the state of the destination is unknown, so the source status is reported.

The SnapMirror status entries on the lag category are as follows:

Lag entry	Description
hh:mm:ss	Indicates the difference between the current time and the timestamp of the Snapshot copy last successfully transferred to the destination.
-	The destination is not initialized.

The SnapMirror status entries on the status category are as follows:

Status entry	Description
Idle	No data is being transferred.
Idle with restart checkpoint (*n* *X*B done)	No data is being transferred. The last transfer attempt was aborted, but the transfer saved a restart checkpoint and thus can be restarted at the next attempt. Transfer sizes are reported in the following units: • KB up to 10,240 KB • MB up to 10,240 MB • GB up to 10,240 GB • TB
Transferring	Transfer has been initiated but has not yet started, or is just finishing.
Transferring (*nX*B done)	Data transfer is in progress. Transfer sizes are reported in the following units: • KB up to 10,240 KB • MB up to 10,240 MB • GB up to 10,240 GB • TB
Pending	The destination was not updated because of a transfer failure; the transfer will be retried automatically.

Status entry	Description
Pending with restart checkpoint (*nX*B done)	The destination was not updated because of a transfer failure. The transfer will be retried automatically from the restart checkpoint. Transfer sizes are reported in the following units: • KB up to 10,240 KB • MB up to 10,240 MB • GB up to 10,240 GB • TB
Aborting	A transfer is being aborted and cleaned up.
Quiescing	The specified volume or qtree is waiting for all existing transfers to complete. The destination is being brought into a stable state.
Resyncing	The specified volume or qtree is being matched with data in the common Snapshot copy.
Waiting	SnapMirror is waiting for a new tape to be put in the tape device.

Additional SnapMirror status entries are as follows:

Additional `-l` option entries	Description
Progress	Displays the amount of data (in KB) transferred by the current transfer. Displays the restart check point if the status is Idle or Pending.
Mirror Timestamp	The timestamp of the last Snapshot copy successfully transferred from the source to the destination. **Note:** A resynchronization might change the base Snapshot copy to a Snapshot copy with an older timestamp.
Base Snapshot copy	The name of the base Snapshot copy for the destination. For volumes in a SnapMirror relationship, this field is the same on the source side and the destination side. For qtrees in a SnapMirror relationship, the destination side lists the name of the exported Snapshot copy for that qtree on the destination. **Note:** A resynchronization might change the name of the base Snapshot copy.
Current Transfer Type	Indicates the kind of transfer now in progress: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the destination side.

Additional `-l` option entries	Description
Current Transfer Error	Displays an error message if the latest transfer attempt failed.
Contents	Indicates whether the contents of the destination volume or qtree in the active file system are up-to-date replicas or in transition. The field applies only to the destination side. • Under SnapMirror volume replication, the contents are always a replica. • Under SnapMirror qtree replication, the contents are usually a replica, but sometimes are transitioning.
Last Transfer Type	Indicates the kind of transfer previously performed: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the destination side.
Last Transfer Size	Displays the amount of data (in KB) transferred in the last successful transfer.
Last Transfer Duration	Displays the elapsed time for the last successful transfer to complete. If the transfer failed and restarted, this includes time waiting to restart the transfer. If a transfer aborted and was retried from the beginning, it includes only the time required for the final successful attempt.
Last Transfer From	This field applies only to the destination side and shows the name of the source system and volume or qtree. This field is useful if you have changed the source in the `/etc/snapmirror.conf` file but the data is actually from the old source.

Related concepts

Data replication from one destination to another in a series (cascading) on page 107

Related tasks

Listing SnapMirror destinations for a volume in a cascading series on page 110

Related references

What SnapMirror status check shows on page 152

Adjusting the TCP window size for a SnapMirror relationship

The TCP window size for SnapMirror might have an impact on SnapMirror performance. You can change the default value of the TCP window size to suit the network configuration. You can specify

the window size for a particular SnapMirror relationship by modifying the corresponding entry in the `snapmirror.conf` file.

Before you begin

- Ascertain the round-trip time between the source and the destination for a SnapMirror relationship.
- Determine the bandwidth available for the SnapMirror relationship.
- The default TCP window size for a SnapMirror relationship is 1,994,752 bytes.
- Adjustment of the TCP window size is applicable only for asynchronous SnapMirror relationships.
- For qtree SnapMirror relationships, TCP window sizes higher than the default value are not supported.

Note: You should only adjust the TCP window size for a SnapMirror relationship if there are throughput issues related to bandwidth utilization.

About this task

The TCP window size specifies the amount of data that a source can send through a connection before it requires an acknowledgement from the destination for the data received. A larger TCP window size can increase SnapMirror throughput in certain scenarios. You can change the TCP window size to optimize SnapMirror transfers for the network in use. Therefore, you can change the TCP window size to optimize SnapMirror transfers.

Note: When using higher TCP window sizes than the default, the system might not be able to achieve the maximum concurrent replication operations specified for the system. This is due to increased resource utilization by the higher TCP window sizes.

The maximum TCP window size that you can specify for a SnapMirror relationship depends on the connection type, as given in the following table.

Connection type	Default TCP window size	Maximum TCP window size
Single path	1,994,752 bytes	7,340,032 bytes (7 MB)
Multipath	1,994,752 bytes	14,680,064 bytes (14 MB)

Note: To limit the network bandwidth used by a particular SnapMirror relationship, use the `kbs` parameter for the relationship entry in the `snapmirror.conf` file.

Steps

1. Calculate a TCP window size that works well for a particular SnapMirror relationship by using the following formula:

window size = (round-trip time) × (available bandwidth)

Example

If the average round trip delay is 130 milliseconds and the available bandwidth is 200 Mbps, the equation is:

window size = (((0.13 sec) × (200,000,000 bps)) / 8) bytes = 3,250,000 bytes

Therefore, you should set the TCP window size for the SnapMirror relationship to 3,250,000 bytes.

Example

Similarly, you can calculate the optimal TCP window size for different round-trip time and bandwidth values. The following table provides a few examples:

Round-trip time	Available bandwidth	TCP window size to maximize throughput
120 ms	400 Mbps	6,000,000 bytes
100 ms	1000 Mbps	12,500,000 bytes (can be used only for a multi-path SnapMirror relationship)
50 ms	155 Mbps	968,750 bytes

2. Specify the required TCP window size by adding the following option to the SnapMirror relationship entry in the `snapmirror.conf` file:

   `wsize=`*`value`*

   `value` is the required TCP window size (in bytes), as calculated in the preceding step.

Example

The following entry specifies a TCP window size of 3,250,000 bytes for the SnapMirror relationship:

```
src_system:src_vol dst_system:dst_vol wsize=3250000 * * * *
```

Example

The following entries specify a TCP window size of 3,250,000 bytes for the multipath SnapMirror relationship:

```
conxn_2=multi(10.72.146.74,10.79.106.40)(10.72.147.74,10.79.107.40)
conxn_2:src_vol dst_system:dst_vol wsize=3250000 * * * *
```

Related references

Maximum number of concurrent replication operations on page 123

Syntax for snapmirror.conf file entries on page 131

Setting a maximum transfer rate for all transfers

Setting a maximum transfer rate for all transfers enables you to limit the total bandwidth used by all transfers at any time. You can set a maximum rate for transfers coming into a system, and a maximum rate for transfers going out of a system.

About this task

You can configure a maximum transfer rate for a system and set maximum transfer rates for each transfer using the `/etc/snapmirror.conf` file. When both the rates are configured, the system-level maximum transfer rate is applied only if the combined bandwidth of transfers goes above the system-level maximum rate. This setting applies to all SnapMirror and SnapVault transfers.

Note: If the SnapMirror relationships exist within the same system, you must set the `snapmirror.volume.local_nwk_bypass.enable` option to `off` to enable throttling of the SnapMirror transfers.

Steps

1. Enable the ability to set system-level maximum transfer rates by using the following command:

   `options replication.throttle.enable on`

2. You can specify the maximum transfer rate used by transfers for a system, as a source or destination. Choose one of the actions from the following table:

If you want to specify the maximum rate for...	Then enter the following command...
Outgoing SnapMirror transfers (applied at the SnapMirror source).	**`options replication.throttle.outgoing.max_kbs`** ***`value`***
Incoming SnapMirror transfers (applied at the SnapMirror destination).	**`options replication.throttle.incoming.max_kbs`** ***`value`***

   `value` is the maximum transfer rate in kilobytes per second. Valid transfer rate values are `1` to `125000`. The default value is unlimited.

   For more information about the `replication.throttle.enable` option, the `replication.throttle.incoming.max_kbs` option, and the `replication.throttle.outgoing.max_kbs` option, see the options(1) man page.

Changing the maximum transfer rate for a single SnapMirror transfer

You can specify the maximum transfer rate for scheduled SnapMirror transfers using the `kbs` option in the `snapmirror.conf` file. However, you can change the maximum transfer rate for the current transfer by using the `snapmirror throttle` command. The change applies to the current transfer only, and the next scheduled transfer uses the maximum transfer rate that is specified in the `snapmirror.conf` file.

About this task

If you change the maximum transfer rate while the current transfer is active, the new maximum transfer rate takes effect within two minutes.

Note: If the SnapMirror relationships exist within the same system, you must set the `snapmirror.volume.local_nwk_bypass.enable` option to `off` to enable throttling of the SnapMirror transfers.

Step

1. To change the maximum transfer rate and apply it to the current transfer, enter the following command on either the source or destination system.

   `snapmirror throttle` ***`n`*** **`[`*** ***`system`*** **`:]`*** ***`path`***

   `n` is the new maximum transfer rate in kilobytes per second. A value of zero (`0`) disables throttling.

   `system` is the destination system. Use this variable if you are executing the command on the source system.

   `path` is the destination path. The path can be the `/`*`volume_name`* or `/vol/`*`volume_name/qtree_name`*.

About moving SnapMirror sources

Whether you are moving a volume SnapMirror source or qtree SnapMirror source to new systems or newer drives, as long as there is a Snapshot copy in common on the source and destination, the transition goes smoothly.

Volume SnapMirror transfers all of the Snapshot copies as part of the SnapMirror replication process.

For qtree SnapMirror, the source and destination have only one Snapshot copy in common. Different qtree SnapMirror destinations have no common Snapshot copy, unless the Snapshot copy is specifically replicated.

In a production environment, you should perform the process of moving SnapMirror relationships from one volume or system to another only in a maintenance or out-of-service window. You should also ensure that new data is not added to the original source during the move.

Moving volume SnapMirror sources

You can move a volume SnapMirror source volume to another source volume.

Before you begin

Ensure that the Data ONTAP release on the destination system is from a release family that is the same as, or later than, the Data ONTAP release on the new source system.

About this task

The following terms are used in the task description for moving volume SnapMirror sources.

- oldsource—The original system on which the source resides.
- newsource—The system to which you are moving the source.
- destination—The system to which the source is replicated.
- oldsourcevol—The original source volume.
- newsourcevol—The new source volume to which you are moving.
- destinationvol—The volume to which the source is replicated.

Steps

1. Copy the original source to the new source using the following command:

   ```
   newsource> snapmirror initialize -S oldsource:oldsourcevol
   newsource:newsourcevol
   ```

   Note: This might take some time to finish.

2. You should make oldsource read-only before continuing.
3. Create a manual Snapshot copy on the oldsource system by using the following command:

   ```
   oldsource> snap create oldsourcevol common_Snapshot
   ```

4. Update newsource and destination based on oldsource using the following commands.

   ```
   newsource> snapmirror update -S oldsource:oldsourcevol
   newsource:newsourcevol
   ```

   ```
   destination> snapmirror update -S oldsource:oldsourcevol
   destination:destinationvol
   ```

Note: The common_Snapshot Snapshot copy is on all volumes because all Snapshot copies are mirrored using volume SnapMirror.

5. Quiesce and break the SnapMirror relationship between oldsource and destination, and oldsource and newsource by using the following commands.

```
destination> snapmirror quiesce destinationvol
destination> snapmirror break destinationvol
newsource> snapmirror quiesce newsourcevol
newsource> snapmirror break newsourcevol
```

6. Using an editor, update the `/etc/snapmirror.conf` file on the destination for the new relationship by replacing the oldsource information with newsource information.

   Before edit

```
oldsource:oldsourcevol destination:destinationvol restart=always 0 * * *
*
```

   After edit

```
newsource:newsourcevol destination:destinationvol restart=always 0 * * *
*
```

7. Establish the new SnapMirror relationship by using the following command:

```
destination> snapmirror resync -S newsource:newsourcevol
destination:destinationvol
```

   Note: The SnapMirror relationship discards any Snapshot copies older than common_Snapshot: namely, the ones used for the last SnapMirror update. This is expected and no data is lost if you ensure that no new data was added to the original source volume during the move.

   The new SnapMirror relationship automatically picks the newest Snapshot copy in common to mirror. This is the common_Snapshot Snapshot copy.

8. Verify that the SnapMirror relationship is resynchronizing by using the following command:

```
destination> snapmirror status
```

Related references

Considerations for the use of SnapMirror on page 82
Prerequisites for SnapMirror on page 82

Moving qtree SnapMirror sources

With qtree SnapMirror you must create a Snapshot copy on the source and force its propagation to the destination and new source. This behavior is unlike volume SnapMirror, in which all of the Snapshot copies from the source are replicated to the destination.

About this task

The process of moving the qtree SnapMirror source involves creating a Snapshot copy on the original source and then replicating the Snapshot copy on the destinations, both the new source and the existing destination. After this is done, the Snapshot copy is common on all volumes, allowing for the SnapMirror relationship to be broken from the original source and established between the new source and the existing destination.

The following terms are used in the task description for moving qtree SnapMirror sources:

- oldsource—The original system on which the source resides.
- newsource—The system to which you are moving the source.
- destination—The system to which the source is replicated.
- oldsourcevol—The original source volume.
- newsourcevol—The new source volume to which you are moving.
- destinationvol—The volume to which the source is replicated.

Steps

1. Copy the original source to the new source by using the following command:

   `newsource> snapmirror initialize -S oldsource:/vol/oldsourcevol/qtree newsource:/vol/newsourcevol/qtree`

   Note: This might take some time to finish.

2. Create a manual Snapshot copy on the `oldsource` system by using the following command:

   `oldsource> snap create oldsourcevol common_Snapshot`

3. Update the destinations by using the following commands:

   `newsource> snapmirror update -c common_Snapshot -s common_Snapshot -S oldsource:/vol/oldsourcevol/qtree newsource:/vol/newsourcevol/qtree`

   `destination> snapmirror update -c common_Snapshot -s common_Snapshot -S oldsource:/vol/oldsourcevol/qtree destination:/vol/destinationvol/qtree`

   The `-s` option of the `snapmirror update` command synchronizes newsource with oldsource and destination with oldsource based on common_Snapshot. The `-c` option of the `snapmirror update` command creates the common_Snapshot Snapshot copy on the destination systems.

4. Quiesce and break the SnapMirror relationship between oldsource and destination, and oldsource and newsource, using the following commands:

   ```
   destination> snapmirror quiesce /vol/destinationvol/qtree
   destination> snapmirror break /vol/destinationvol/qtree
   newsource> snapmirror quiesce /vol/volnewsourcevol/qtree
   newsource> snapmirror break /vol/volnewsourcevol/qtree
   ```

5. Using an editor, update the `/etc/snapmirror.conf` file on the destination for the new relationship by replacing the oldsource information with newsource information.

   Before edit

   ```
   oldsource:/vol/oldsourcevol/qtree destination:/vol/destinationvol/qtree
   restart=always 0 * * * *
   ```

   After edit

   ```
   newsource:/vol/newsourcevol/qtree destination:/vol/destinationvol/qtree
   restart=always 0 * * * *
   ```

6. Establish the new SnapMirror relationship using the following command on the destination system:

   ```
   snapmirror resync -S newsource:/vol/newsourcevol/qtree destination:/vol/
   destinationvol/qtree
   ```

   Note: SnapMirror discards any Snapshot copies older than the common Snapshot copy, namely, the ones used for the latest SnapMirror update. This is expected and no data is lost if you ensure that no new data was added to the original source volume during the move.

   The new SnapMirror relationship automatically picks the newest common Snapshot copy for replication. This is the common Snapshot copy.

7. Verify that the SnapMirror relationship is resynchronizing by using the following command:

   ```
   destination> snapmirror status
   ```

Related references

Considerations for the use of SnapMirror on page 82
Prerequisites for SnapMirror on page 82

Methods to migrate data between volumes

You can migrate data between volumes by using the `snapmirror migrate` command or by performing the volume move operation to move data nondisruptively.

For information about migrating data between volumes by performing the volume move operation, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

Migrating data between volumes by using SnapMirror

SnapMirror can migrate data between volumes and redirect NFS clients to the new volume without rebooting the system or remounting to volume on NFS clients.

Before you begin

If you are migrating data within the same storage system and the source volume contains LUN(s), then you must have unmapped the source LUN(s) by using the `lun unmap` command.

About this task

The migration must be run on two volumes which are currently the source volume and destination volume in a SnapMirror relationship. When you start the migration process, SnapMirror does the following:

- Performs a SnapMirror incremental transfer to the destination volume.
- Stops NFS and CIFS services on the entire system with the source volume.
- Migrates NFS file handles to the destination volume.
- Makes the source volume restricted.
- Makes the destination volume read-write.

SnapMirror does not transfer IP addresses, license keys, or quota information. You must remount on the NFS clients unless one of the following is true:

- The IP address of the source system is transferred to the destination system independently after the migration.
- The source and destination volumes reside on the same system, in which case, the IP address to access either volume is the same.

SnapMirror does not migrate CIFS clients. You must reestablish CIFS client sessions after migrating data to the destination volume.

Step

1. Enter the following command:

   snapmirror migrate [*src_system*:]*src_vol* [*dst_system*:]*dst_vol*

   src_system is the source system.

   src_vol is the source volume.

   dst_system is the destination system.

   dst_vol is the destination volume.

Conversion of a destination to a writable volume or qtree

You can use the `snapmirror break` command to convert a SnapMirror destination, with read-only status, to a writable volume or qtree.

You might want to convert a destination to a writable volume or qtree to perform one of the following tasks:

- Data migration—Moving data from one volume or qtree (original source) to another volume or qtree (present destination) and make the data on the destination accessible and writable.
- Disaster recovery—If your source becomes unavailable, and you want your present destination to substitute as the users' retrieval and input source.
- Application testing—You want to make your current destination volume or qtree writable to test a new application on a mirrored replication of your current data rather than risk corruption of original data on the source volume or qtree.

Converting the destination to a writable volume or qtree enables you to use data on the destination, especially when the original source is unavailable.

Quota restrictions

Quotas are always disabled on a SnapMirror volume destination, regardless of whether quotas are enabled on the source volume. If you try to enable quotas on a volume destination, SnapMirror displays an error message. Quotas are not disabled on SnapMirror destination qtrees.

If the source volume or qtree and the destination reside on different storage systems, and you want the same quota restrictions to be applied after you make the destination writable, the destination system must have an `/etc/quotas` file that includes all the entries from the `/etc/quotas` file used by the source system.

- If you use SnapMirror replication for data migration, you can copy the `/etc/quotas` entries from the source system to the `/etc/quotas` file of the destination system before you use the `snapmirror break` command to make the destination writable.
- If you use SnapMirror replication for backup and potential disaster recovery, you must keep a copy on the destination system of all `/etc/quotas` entries used by the source system at all times. That way, you can apply the quota entries to the destination volume or qtree if the source system becomes unavailable.

Converting a SnapMirror destination to a writable volume or qtree

You can convert a SnapMirror destination to a writable volume or qtree.

Steps

1. On the destination system, use the `snapmirror break` command to make the destination volume or qtree writable.
   - To make a destination volume writable, enter the following command on the destination system.

     `snapmirror break` *`volume_name`*
   - To make a destination qtree writable, enter the following commands on the destination system.

     `snapmirror quiesce /vol/`*`volume_name/qtree_name`*

     `snapmirror break /vol/`*`volume_name/qtree_name`*
2. If you want to enable quotas on the former destination volume, carry out the following steps:
   a. Edit the `/etc/quotas` file on the former destination system so that, after the conversion, the former destination includes the same quota restrictions as the source volume.

      If the original source volume uses per-volume quotas, replace the original source volume name with the former destination name in the quota entries.

   b. Enter the following command to enable quotas on the former destination:

      `quota on` *`volume_name`*
3. Consider the following optional measures:
   - If you want to stop a SnapMirror source from trying to update a broken-off destination, you can delete or comment out the entry in the `/etc/snapmirror.conf` file. Otherwise, SnapMirror continues to try to update the destination.
   - You might also want to use the `options fs_size_fixed off` command to turn off the option that restricts the size of the file system on a destination volume.

     Note: If you set `options fs_size_fixed off`, the ability of the destination and source volumes to resync is not guaranteed.

After using the snapmirror break command

After using the `snapmirror break` command to temporarily break a SnapMirror relationship between a source and destination, you can use other SnapMirror commands to either make the break permanent, or restore or redefine the SnapMirror relationship.

- Use the `snapmirror release` command to make the break permanent.

- Use the `snapmirror resync` command to restore or redefine the SnapMirror relationship.

Related concepts

What the snapmirror resync command does on page 186
How the snapmirror resync command helps minimize data loss on page 190

Related tasks

Releasing partners from a SnapMirror relationship on page 177
Resynchronizing a SnapMirror relationship on page 187

Resizing a SnapMirror source and destination volume pair for a FlexVol volume

You can increase the size of the source volume for a FlexVol volume. After you increase the size of a source volume and perform a SnapMirror transfer, the size of the destination volume is automatically increased to the same size as that of the source volume, provided the destination aggregate has sufficient space to contain the resized volume.

Steps

1. To increase the size of the SnapMirror source volume, enter the following command:

   `vol size` *`vol_name size`*

   `vol_name` is the name of the SnapMirror source volume.

   `size` is the required size of the SnapMirror source volume.

   Note: SnapMirror updates the size of the destination volume to match the source in the next SnapMirror transfer.

2. On the destination system, enter the following command to check the size of the destination volume and the file system:

   `vol status` *`vol_name`* `-b`

   `vol_name` is the name of the SnapMirror destination volume.

   The size of the destination volume must have increased to the same size as that of the source volume.

Resizing a SnapMirror source and destination volume pair for traditional volumes

You can increase the size of the source volume for traditional volumes. If the destination volume is not large enough to contain the resized source volume, you have to manually increase the size of the destination volume to the same size as that of the resized source volume.

About this task

Before increasing the size of a source volume, you must compare the size of the source and destination volumes for a specific volume SnapMirror relationship. If the destination volume is not large enough to contain the larger source volume, you must manually resize the destination volume.

Note: In an active volume SnapMirror relationship, the size of the source and destination file systems is identical. However, the size of the destination volume can be larger than or equal to the size of the source volume. The size of the file systems must be identical to allow the source to be restored from the destination.

Steps

1. On the source system, enter the following command to check whether the `fs_size_fixed` option is `off`:

   `vol status` ***`vol_name`*** **`-v`**

   `vol_name` is the name of the source volume for SnapMirror.

   #### Example

```
SRC_A> vol status sm_src -v
        Volume State    Status            Options
        sm_src online   raid4, trad       nosnap=off, nosnapdir=off, minra=off,
                        parity uninit'd!  no_atime_update=off, raidtype=raid4, raidsize=8,
                        32-bit            nvfail=off, ignore_inconsistent=off,
                                          snapmirrored=off, resyncsnaptime=60,
                                          create_ucode=off, convert_ucode=off,
                                          maxdirsize=41861, schedsnapname=ordinal,
                                          fs_size_fixed=off, svo_enable=off,
                                          svo_checksum=off, svo_allow_rman=off,
                                          svo_reject_errors=off, no_i2p=off,
                                          fractional_reserve=100, extent=off,
                                          read_realloc=off, snapshot_clone_dependency=off,
                                          dlog_hole_reserve=off, ha_policy=cfo,
                                          nbu_archival_snap=off
                    Volume UUID: 73b3baae-653d-11e2-9c13-123478563412
               Containing aggregate: <N/A>

                Plex /sm_src/plex0: online, normal, active
                    RAID group /sm_src/plex0/rg0: normal, block checksums

          Snapshot autodelete settings for sm_src:
                                state=off
                                target_free_space=0%
                                delete_order=oldest_first
```

```
                          defer_delete=none
                          prefix=(not specified)
                          destroy_list=none
     Hybrid Cache:
            Eligibility=None
```

Note: If the `fs_size_fixed` option is set to `off`, the size of the destination volume's file system might be different from that of the source volume. To restore the source volume from the SnapMirror destination, the size of the source and destination file systems should be identical. When the status of a volume SnapMirror relationship is `Idle`, and the size of the destination volume is increased, then you can change the `fs_size_fixed` option to `on`.

2. Depending on the value of the `fs_size_fixed` option, choose one of the actions from the following table:

If the `fs_size_fixed` option is set to...	Then...
`off`	Go to the next step.
`on`	On the source system, enter the following command: **`vol options vol_name fs_size_fixed off`** *`vol_name`* is the name of the source volume for SnapMirror. **Example** `SRC_A> vol options sm_src fs_size_fixed off`

3. On the source system, enter the following command to find the size of the source volume and the size of the file system:

   `vol status vol_name -b`

   `vol_name` is the name of the SnapMirror source volume.

   Example

```
SRC_A> vol status sm_src -b
Volume              Block Size (bytes)  Vol Size (blocks)   FS Size (blocks)
------              ------------------  ------------------  ----------------
sm_src              4096                256000              256000
```

4. On the destination system, enter the following command to find the size of the destination volume and the size of file system:

   `vol status vol_name -b`

   `vol_name` is the name of the SnapMirror destination volume.

Example

```
DST_B> vol status sm_dst -b
Volume              Block Size (bytes)  Vol Size (blocks)   FS Size (blocks)
------              ------------------  -----------------   ----------------
sm_dst              4096                512000              256000
```

5. Compare the size of the volume and the file system, for the source and destination pair of a SnapMirror relationship.

   Note: If you want to increase the size of the source volume beyond the present size of the destination volume, you must manually increase the size of the destination volume.

   Example

   In the example shown in the previous steps, the destination volume size is greater than the source volume size.

6. Depending on the present size of the SnapMirror destination volume, choose one of the actions from the following table:

If the size of the destination volume...	Then...
Is enough to contain the intended increase in the size of the source volume	Go to Step 8.
Needs to be increased to contain the intended increase in the size of the source volume	Go to the next step.

7. To increase the size of the SnapMirror destination volume, enter the following command:

   `vol add` *`vol_name disks`*

   `vol_name` is the name of the SnapMirror source volume.

   `disks` is the number of disks that you want to add to the traditional volume.

   You can use the `vol status` command to confirm the increase in the source volume size.

   Note: You can increase the destination volume size to be equal to or greater than the intended increased size of the source volume.

8. Perform Step 7 for the source volume to increase the size of the SnapMirror source volume.

9. On the source system, enter the following command:

   `vol options` *`vol_name`* `fs_size_fixed on`

   `vol_name` is the name of the source volume for SnapMirror.

This option ensures that the size of the SnapMirror source and destination file systems are identical.

Note: SnapMirror updates the size of the destination file system to match the source in the next SnapMirror transfer for the volume pair.

Converting asynchronous SnapMirror replication to synchronous

You can change an asynchronous volume SnapMirror relationship to replicate data synchronously by editing the `snapmirror.conf` file on the destination system.

About this task

Synchronous replication is not supported for qtree SnapMirror relationships. Therefore, a qtree SnapMirror relationship cannot be converted to a synchronous SnapMirror relationship.

Step

1. To convert an asynchronous SnapMirror relationship to a synchronous SnapMirror relationship, on the administration host, edit the `snapmirror.conf` file on the destination system to change the schedule to `sync`.

Stabilizing destinations before a Snapshot copy

You might need to temporarily stop transfers to a destination, by using the `snapmirror quiesce` command. For example, if you want to create a Snapshot copy of a SnapMirror destination volume or qtree that contains a database, you need to ensure that its contents are stable during the Snapshot copy.

Step

1. Enter the following command on the system on which you want to block transfers:

   snapmirror quiesce {*dest_volume* | /vol/*volume_name*/*qtree_name*}

   dest_volume is the name of the destination volume.

   qtree_name is the name of a qtree in *volume_name*.

Example

```
systemA> snapmirror quiesce vol1
```

```
snapmirror quiesce: in progress.
snapmirror quiesce: vol1: successfully quiesced
```

SnapMirror stops any further data transfers to vol1.

Example

```
systemA> snapmirror quiesce vol2

snapmirror quiesce: in progress.
This can be a long-running operation. Use Control-C to interrupt.
.....................................
snapmirror quiesce: vol2: successfully quiesced
```

SnapMirror waits for a transfer to finish and stops any further data transfers to vol2.

Example

```
systemA> snapmirror quiesce /vol/vol1/qtree1
```

SnapMirror stops data transfers to qtree1 in vol1.

If you use the `snapmirror break` command on a destination that is quiesced, the quiesce condition is automatically cleared when the destination becomes writable.

Note: If you decide to abort a SnapMirror quiesce operation, press Ctrl-C or enter the `snapmirror resume` command at any time.

A SnapMirror destination volume might have been deleted after the volume was quiesced. If you want to create a SnapMirror destination volume with the same name as the deleted volume, first use the `snapmirror release` command. This step would ensure that the SnapMirror relationship is set up properly.

What the quiesce command does

The `snapmirror quiesce` command waits for all volume and qtree SnapMirror transfers to complete, and blocks any further updates. If a qtree is not in a stable state (is in transition), the `snapmirror quiesce` command forces it into a stable state.

You can quiesce only volumes and qtrees that are online and that are SnapMirror destinations. You cannot quiesce a restricted or offline volume or a qtree in a restricted or offline volume.

The `snapmirror quiesce` command stops a volume or qtree from acting as a SnapMirror destination, but does not prevent it from acting as a SnapMirror source.

Note: The quiesced state persists across reboots.

Resuming transfers after quiescing a destination

You can use the `snapmirror resume` command to restore the capability for data transfer to a volume or qtree you have quiesced.

Step

1. Enter the following command for the system on which you want to resume transfers:

   `snapmirror resume {`*`dest_volume`*` | /vol/`*`vol_name/qtree_name`*`}`

   `dest_volume` is the name of the destination volume.

   `qtree_name` is the name of a qtree in *`vol_name`*.

Example

```
systemA> snapmirror resume vol2
snapmirror resume: vol2: Successfully resumed
```

SnapMirror resumes normal data transfer capability for vol2.

Aborting a SnapMirror transfer

You can use the `snapmirror abort` command to abort a volume or qtree replication operation before the transfer is complete. You can abort a scheduled update, a manual update, or an initial SnapMirror transfer.

About this task

You should consider the following issues before aborting a SnapMirror transfer:

- If you abort a copy operation, data transfer stops and SnapMirror is put in a restartable mode.
- If you use the `-h` (hard abort) option with the `snapmirror abort` command, you cannot restart the transfer.

Step

1. From either the source or the destination system, enter the following command:

   `snapmirror abort [-h] {[`*`dest_system`*`:]`*`dest_volume`*` | [`*`dest_system`*`:]`*`/vol/volume_name/qtree_name`*` ...}`

   `-h` specifies a hard abort; the transfer cannot be restarted. SnapMirror stops the transfer and clears the restartable transfer log. This option applies only to the SnapMirror destination.

   `dest_system` is the name of the destination system.

dest_volume is the destination volume.

/vol/ *volume_name/qtree_name* is the path name of a destination qtree.

Note: If no destination system is specified, the local host's name is used for the system name. You can enter more than one destination volume.

You can obtain the destination system and volume from the `snapmirror status` output.

Note: If no destination volume or qtree is specified, the command returns an error message; it does not abort all transfers. To abort all transfers, use the `snapmirror off` command.

If you enter an invalid SnapMirror destination (one that is not displayed in the output of the `snapmirror status` command), the command fails and displays an error message.

Example

```
systemA> snapmirror abort vol1 systemB:vol2 systemC:/vol3/qtree3

snapmirror abort: Aborting transfer to vol1 systemB:vol2 systemC:/
vol3/qtree3
```

SnapMirror aborts the transfer to `vol1` on `systemA`, where the command was entered, and aborts the transfer to `vol2` on `systemB` and the transfer to `qtree3` in `vol3` on `systemC`.

Releasing partners from a SnapMirror relationship

To permanently end a SnapMirror relationship between a source and destination pair of volumes or qtrees, you need to use the `snapmirror release` command on the source and the `snapmirror break` command on the destination.

About this task

Releasing a source from a destination volume or qtree allows the source to delete its base Snapshot copy for the SnapMirror relationship. After breaking the relationship, you need to take additional steps to scrub the destination. Unless these extra steps are performed, the Snapshot copies associated with the broken relationship remain stored on the destination system, and a `snapmirror status` command continues to list the former destination object as a current destination object.

Steps

1. On the source system, enter the following command:

   snapmirror release {*source_volume* | *qtree_path*}[*dest_system*:] {*dest_volume* | *qtree_path*}

   source_volume or *qtree_path* is the name of the source volume or path to the qtree that you want to release from the destination.

dest_system is the name of the system where the destination is located.

dest_volume or *qtree_path* is the name of the volume or path to the qtree that is the destination.

If you do not enter the name of the destination system, SnapMirror uses the name of the system on which you entered the command.

Example

For a SnapMirror volume relationship:

```
systemA> snapmirror release vol0 systemB:vol2
```

Example

For a SnapMirror qtree relationship:

```
systemA> snapmirror release vol/vol1/qtree2 systemB:/vol/vol2/qtree5
```

SnapMirror frees all resources on the source system that had been dedicated to the SnapMirror relationship.

2. On the destination system, enter the following command to break the SnapMirror relationship between the source and destination objects.

   snapmirror break {*vol_name* | *qtree_path*}

   vol_name is the name of the volume that you want to release from the relationship.

   qtree_path is the path of the qtree that you want to release from the relationship.

3. On the destination system, use the `snapmirror status -l` command to determine which Snapshot copy basename is associated with the SnapMirror relationship that you just broke.

   - For a broken SnapMirror volume relationship:

     snapmirror status -l *dest_vol*

   - For a broken SnapMirror qtree relationship:

     snapmirror status -l /vol/*dest_vol*/*dest_qtree*

   In the detailed output that is displayed, note the Snapshot copy basename associated with the SnapMirror relationship that you just broke.

4. On the destination system, use the following command to delete the Snapshot copy set that you displayed in the previous step.

   snap delete *dest_vol snapshot_basename*

5. Through the Admin host client, edit the `/etc/snapmirror.conf` file on the destination system. Locate and delete the entry that specifies the SnapMirror relationship you want to end.

Related concepts

What the snapmirror.conf file does on page 129
Conversion of a destination to a writable volume or qtree on page 168

Related tasks

Editing the snapmirror.conf file on page 130
Converting a SnapMirror destination to a writable volume or qtree on page 169

Related references

Syntax for snapmirror.conf file entries on page 131

SnapMirror data transfer logs

You can use the `options snapmirror.log.enable` command to check SnapMirror data transfer logs. You can find out whether transfers are occurring as planned, how long the transfers take, and how well the system setup works. You find this information in the SnapMirror log file.

The SnapMirror log file provides the following information:

- The start time and the end time of the SnapMirror logging process.
- The start time, end time, and size of each transfer.
- Any abnormal termination and restart of a transfer.
- Other SnapMirror-related activities.

You can use the raw information provided to do the following:

- Calculate the average transfer size.
- Calculate the average transfer time.
- Look at the number of successful transfers and the failure rate.
- Tune the schedule.
- Create a notifier for aborted transfers.
- Monitor performance on a per-volume level.
- Be assured that things are working as planned.

Checking for SnapMirror logging

SnapMirror logging is on, by default. However, you can find out whether SnapMirror logging is on by using the `snapmirror.log.enable` option.

Step

1. Enter the following command on the system for which you want the information:

   `options snapmirror.log.enable`

   SnapMirror reports whether logging is enabled.

Example

```
systemA> options snapmirror.log.enable
snapmirror.log.enable      on
```

Turning SnapMirror logging on

You can turn on SnapMirror logging by setting the `snapmirror.log.enable` option to `on`.

About this task

SnapMirror keeps the current log on the root volume of the system as `/etc/log/snapmirror.0`. A new log file is generated every week as `/etc/log/snapmirror.0`. Older log files are renamed `/etc/log/snapmirror.[1-5]` and the oldest log file is deleted. You can read the log files using a text editor.

Step

1. Enter the following command on the system for which you want the log:

   `options snapmirror.log.enable on`

   Note: This setting is persistent across reboots.

Result

SnapMirror enables the logging of transfers for the system.

Location of SnapMirror logs

Starting with Data ONTAP 8.2.2, an option to disable or enable all the SnapMirror logging, including non-default vFilers into vfiler0 is provided. In releases prior to Data ONTAP 8.2.2, the SnapMirror

logs of each non-default vFiler were stored in the respective non-default vFiler root volume, separate from the vfiler0 and other non-default vFiler's logs.

The option `snapmirror.vfiler0.logging.enable` is ON by default in Data ONTAP 8.2.2. All SnapMirror logging activity in the vfiler0 and non-default vFiler context is now saved into a single SnapMirror log file`/vol/vol0/etc/log/snapmirror`.

If you prefer the earlier behavior, you can turn the `snapmirror.vfiler0.logging.enable` option OFF.

Format of SnapMirror log files

Understanding the format of SnapMirror log files can help you better handle issues related to SnapMirror transfers.

The log file is in the following format:
`type timestamp source_system:source_path dest_system:dest_path event_info`

`type` can be one of the following: `src`, `dst`, `log`, `cmd`. *`type`* specifies whether the record is for the source side (`src`) or destination side (`dst`) of the transfer. Certain events apply to only one side. The type `log` indicates a record about the logging system itself, for example, `Start_Logging` and `End_Logging`. The type `cmd` indicates a record of user commands, for example, `Release_command` and `Resync_command`.

`timestamp` is expressed in `ctime` format, for example:
`Fri Jul 27 20:41:09 GMT.`

`event_info` includes the following event names:
`Request (IP address | transfer type) Start Restart (@ num KB) End (num KB done) Abort (error_msg) Defer (reason) Rollback_start Rollback_end Rollback_failed Start_Logging End_Logging Wait_tape New_tape Snapmirror_on Snapmirror_off Quiesce_start Quiesce_end Quiesce_failed Resume_command Break_command Release_command Abort_command Resync_command Migrate_command`

The `Request` event on the source side includes the IP address of the system that made the transfer request; the `Request` event on the destination side includes the type of transfer. At the end of each successful transfer, the `End` event also reports the total size of the transfer in KB. Error messages are included with the `Abort` and `Defer` events.

Example

The following is an example of a log file from the source side:

```
log Fri Jul 27 20:00:01 GMT - - Start_Logging
cmd Fri Jul 27 20:00:20 GMT - - Snapmirror_on
src Fri Jul 27 20:41:09 GMT system1:vol1 system2:vol1 Request (10.56.17.133)
src Fri Jul 27 20:41:32 GMT system1:vol1 system2:vol1 Abort (Destination not allowed)
src Fri Jul 27 20:45:31 GMT system1:vol0 system1:vol1 Request (10.56.17.132)
src Fri Jul 27 20:45:35 GMT system1:vol0 system1:vol1 Start
src Fri Jul 27 20:51:40 GMT system1:vol0 system1:vol1 End (26200 KB)
src Fri Jul 27 22:41:09 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Request
(10.56.17.133)
src Fri Jul 27 22:41:12 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Start
src Fri Jul 27 22:41:13 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Abort (Non-
```

```
unicode directory found in source qtree.)
src Fri Jul 27 22:45:53 GMT system1:/vol/vol1/qtb system2:/vol/vol1/qsmb Request
(10.56.17.133)
src Fri Jul 27 22:45:56 GMT system1:/vol/vol1/qtb system2:/vol/vol1/qsmb Start
src Fri Jul 27 22:45:59 GMT system1:/vol/vol1/qtb system2:/vol/vol1/qsmb End (3800 KB)
cmd Fri Jul 27 22:50:29 GMT system1:/vol/vol1/qtb system2:/vol/vol1/qsmb Release_command
```

Example

The following is an example of a log file from the destination side:

```
dst Fri Jul 27 22:50:18 GMT system1:vol0 system1:vol1 Request (Initialization)
dst Fri Jul 27 22:50:20 GMT system1:vol0 system1:vol1 Abort (Destination is not
restricted)
dst Fri Jul 27 22:57:17 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Request
(Initialize)
dst Fri Jul 27 22:57:24 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Start
dst Fri Jul 27 22:57:36 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB End (55670 KB)
dst Fri Jul 27 23:10:03 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Request
(Scheduled)
dst Fri Jul 27 23:10:07 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Start
dst Fri Jul 27 23:10:18 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB End (12900 KB)
cmd Sat Jul 28 00:05:29 GMT - system2:/vol/vol1/qtB Quiesce_start
cmd Sat Jul 28 00:05:29 GMT - system2:/vol/vol1/qtB Quiesce_end
cmd Sat Jul 28 00:05:40 GMT - system2:/vol/vol1/qtB Break_command
cmd Sat Jul 28 00:41:05 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Resync_command
log Sat Jul 28 00:41:10 GMT - - End_Logging
```

Example

The following is an example of a log file from a retrieve (from tape) request:

```
dst Fri Jun 22 03:07:34 GMT filer_1:rst0l filer_1:bigtwo Request (retrieve)
dst Fri Jun 22 03:07:34 GMT filer_1:rst0l filer_1:bigtwo Start
dst Fri Jun 22 05:03:45 GMT filer_1:rst0l filer_1:bigtwo Wait_tape
dst Fri Jun 22 15:16:44 GMT filer_1:rst0l filer_1:bigtwo New_tape
dst Fri Jun 22 17:13:24 GMT filer_1:rst0l filer_1:bigtwo Wait_tape
dst Fri Jun 22 17:56:43 GMT filer_1:rst0l filer_1:bigtwo New_tape
dst Fri Jun 22 18:10:37 GMT filer_1:rst0l filer_1:bigtwo End (98602256 KB)
```

Turning SnapMirror logging off

You can turn off the SnapMirror log process by setting the `snapmirror.log.enable` option to `off`.

Step

1. Enter the following command on the system for which you want to disable SnapMirror logging:

   `options snapmirror.log.enable off`

Listing SnapMirror Snapshot copies

You can use the `snap list` command to list all Snapshot copies, including the SnapMirror-specific Snapshot copies that are stored on the system.

Step

1. In the console of either your source or destination system, enter the following command:

   `snap list` ***`vol_name`***

Result

A list of all Snapshot copies stored on your system is displayed. SnapMirror Snapshot copies are distinguished from system Snapshot copies by a more elaborate naming convention and the label "snapmirror" in parentheses.

Naming conventions for Snapshot copies used by SnapMirror

When you run the `snap list` command, you can distinguish SnapMirror Snapshot copies from the regular system Snapshot copies by their naming conventions.

For volume replication, SnapMirror creates a Snapshot copy of the entire source volume that is copied to the destination volume.

A SnapMirror volume Snapshot copy name is in the following format:

`dest_system(sysid)_name.number`

- *`dest_system`* is the host name of the destination system.
- *`sysid`* is the destination system ID number.
- *`name`* is the name of the destination volume.
- *`number`* is the number of successful transfers for the Snapshot copy, starting at 1. Data ONTAP increments this number for each transfer.

Note: In the output of the `snap list` command, SnapMirror Snapshot copies are followed by the SnapMirror name in parentheses.

Volume example

```
systemA(0016791363)_vol0.9 (snapmirror)
```

For qtree replication, SnapMirror creates Snapshot copies of one or more source qtrees on the source volume that are copied to a qtree on the destination volume.

A qtree SnapMirror Snapshot copy name is in the following format:

dest_system(sysid)_name-src|dst.number

- *dest_system* is the host name of the destination system.
- *sysid* is the destination system ID number.
- *name* is the name of the destination volume or qtree path.
- *src|dst* is the source or destination name.
- *number* is an arbitrary start point number for the Snapshot copy. Data ONTAP increments this number for each transfer.

Qtree example

```
systemA(0016789302)_vol1_qtree3-dst.15 (snapmirror)
```

Attention: You should not delete manually-created Snapshot copies marked `snapmirror` in the output of the `snap list` command. Otherwise, later SnapMirror updates might fail.

Use of the snap list command to display SnapMirror updates on the destination volume

The `snap list` command displays information for each Snapshot copy on a storage system. Along with the name of the Snapshot copy, it displays when the Snapshot copy was created and the size of the Snapshot copy.

Example

The following example describes SnapMirror Snapshot copies that are created on a source volume and copied to a destination volume. In this example, data is copied from vol1 of systemA (the source) to vol2 of systemB (the destination).

To create a baseline version of a destination volume, systemA creates a Snapshot copy named systemB(0016782130)_vol2.1 on systemA. All Snapshot copies in vol1 of systemA, including systemB(0016782130)_vol2.1, are transferred to vol2 of systemB. When replicating a qtree, SnapMirror transfers only the qtree's data in the Snapshot copy for the qtree.

If the administrator runs the `snap list` command on the destination systemB after the systemB(0016782130)_vol2.1 Snapshot copy is transferred from systemA to systemB, a listing similar to the following example is generated.

```
systemB> snap list vol2

working.....

%/used   %/total   date          name
-------- --------  ------------  --------
0% ( 0%) 0% ( 0%)  Nov 17 10:50  systemB(0016782130)_vol2.1 (snapmirror)
1% ( 0%) 0% ( 0%)  Nov 17 10:00  hourly.0
1% ( 0%) 0% ( 0%)  Nov 17 00:00  nightly.0
1% ( 0%) 0% ( 0%)  Nov 15 16:00  hourly.1
1% ( 0%) 1% ( 0%)  Nov 15 15:00  hourly.2
2% ( 0%) 1% ( 0%)  Nov 15 14:00  hourly.3
2% ( 0%) 1% ( 0%)  Nov 15 13:00  hourly.4
2% ( 0%) 1% ( 0%)  Nov 15 12:00  hourly.5
```

When it is time to update the destination, another Snapshot copy is created on systemA.

The `snap list` command on systemA generates the following display after the systemB(0016782130)_vol2.2 Snapshot copy is created on systemA.

```
systemA> snap list vol1

working....

%/used   %/total   date          name
-------- --------  ------------  --------
0% ( 0%) 0% ( 0%)  Nov 17 10:52  systemB(0016782130)_vol2.2 (snapmirror)
0% ( 0%) 0% ( 0%)  Nov 17 10:51  systemB(0016782130)_vol2.1 (snapmirror)
1% ( 0%) 0% ( 0%)  Nov 17 10:00  hourly.0
1% ( 0%) 0% ( 0%)  Nov 17 00:00  nightly.0
1% ( 0%) 0% ( 0%)  Nov 15 16:00  hourly.1
1% ( 0%) 1% ( 0%)  Nov 15 15:00  hourly.2
```

After the systemB(0016782130)_vol2.2 Snapshot copy is transferred from systemA to systemB, both Snapshot copies exist on systemB. On systemA, however, systemB(0016782130)_vol2.1 is no longer needed and is deleted; only systemB(0016782130)_vol2.2 is retained to be used for the next transfer.

You can see a list of each SnapMirror Snapshot copy on the server, and the qtrees it contains, and the client sources of those qtrees and their timestamps by using the `snap list -q` command.

You can use the `snap list -o` command to display the names, timestamps, and sources (if they are copies) of the qtrees in a specified volume or at a path name.

What SnapMirror restarts and retries are

In SnapMirror, a retry is an automatic attempt to start the transfer process after an interruption, whether or not any data was successfully transferred. A restart is the resumption of a previous transfer process from a restart checkpoint.

SnapMirror sets a restart checkpoint every 5 minutes during a transfer. SnapMirror restarts the previous transfer where it left off, if the following conditions are met:

- A restart checkpoint exists.
- All Snapshot copies being transferred still exist.
- The value for the restart mode in the `snapmirror.conf` file is set to `always` or is not set, and the next scheduled update has not arrived.

If the conditions are not met, SnapMirror creates a new Snapshot copy and starts a new transfer.

If a scheduled transfer fails (for example, due to network failure), SnapMirror automatically retries the transfer the next minute. If a transfer fails due to an error that renders it unfit for a retry (for example, if a user aborts the transfer), or if the source denied the transfer for any reason, the transfer is not retried the next minute. In such cases, an update is always attempted according to the schedule specified in the `snapmirror.conf` file.

Note: If a manual update fails, the update is not tried automatically and the user is informed. The user needs to reissue the command if an update is required.

After a reboot, SnapMirror does not automatically retry a transfer that was interrupted; however, the next scheduled or manual transfer restarts it at that restart checkpoint, if the checkpoint is still valid.

An initial transfer can be restarted but will not be retried automatically. To restart an initial transfer, enter the `snapmirror initialize` command again. Scheduled incremental updates automatically retry the transfer.

What the snapmirror resync command does

You can use the `snapmirror resync` command to reestablish the connection between the source and the destination. This command is applied after the SnapMirror relationship between the source and destination is broken.

You can apply the `snapmirror resync` command to either the original SnapMirror destination or the original source:

- When applied to the original destination, the `snapmirror resync` command puts a volume or qtree back into a SnapMirror relationship and resynchronizes its contents with the source without repeating the initial transfer.

- When applied to the source volume, the `snapmirror resync` command turns the source volume into a copy of the original destination volume. In this way, the roles of source and destination are reversed.

  Note: Resynchronization is not possible if SnapMirror cannot find a common Snapshot copy on the source and destination to use as the basis for resynchronization. SnapMirror generates a "`No common snapshot to use as the base for resynchronization`" error message and terminates the command. You must reinitialize the destination to establish the SnapMirror relationship.

When you run the `snapmirror resync` command on the source, a reverse relationship from the destination to the source is established. This resynchronizes all the updated content from the destination to the source. However, you cannot resynchronize using the Snapshot copies taken on the destination volume when the destination qtree was in a mirrored state.

Resynchronizing a SnapMirror relationship

You can use the `snapmirror resync` command to restore or redefine a SnapMirror source or destination relationship that was broken with the `snapmirror break` command.

About this task

You might want to resynchronize a source and a destination volume or qtree under the following circumstances:

- When you change the current source to a different volume or qtree.
- When you make a destination volume writable for application testing and then want to make it a SnapMirror destination again.
- When you need to recover from a disaster that disabled the source.
- When you want to reverse the functions of the source and the destination.

Note: When you perform resynchronization for the destination system, the contents on the destination are overwritten by the contents on the source.

Steps

1. From the destination system, enter the following command:

   snapmirror resync [*options*] [*dest_system*:]{*dest_volume* | /vol/*qtree_path*}

   options can be any of the following:

   - `-n` does not execute the resynchronization, but displays what is done if the `snapmirror resync` command is run. You can use this option to find whether you have a Snapshot copy on the source and the destination that can be used as the newest common Snapshot copy (base Snapshot copy) so that you can resynchronize a specific SnapMirror relationship.

- `-f` forces the operation to proceed without prompting you for confirmation.
- `-k` *n* sets the maximum transfer speed to *n* kilobytes per second. This option has the same effect as the `kbs` argument in the `/etc/snapmirror.conf` file.
- `-S` [*source_system*:]{*source_volume* | *qtree_path*} specifies the system and volume or qtree you want to use as the source for resynchronization.
  The source specified by the `-S` option must match a source entry in the `/etc/snapmirror.conf` file. If entries exist but the source does not match, the operation displays an error message and terminates. If there is no entry for the specified source, the command runs.

  Note: If the `-S` option is not set, the source must be specified in the `/etc/snapmirror.conf` file. If it is not specified, the operation displays an error message and terminates.

- `-c` *snapshot_name* creates a Snapshot copy (with the name *snapshot_name*) of a qtree on the destination after the resynchronization transfer completes (so that it does not compete with any ongoing updates).
  SnapMirror does not lock or delete this Snapshot copy.

  Note:

  - *snapshot_name* cannot be `minutely.`*x*, `hourly.`*x*, `nightly.`*x*, or `weekly.`*x*, because these names are reserved for scheduled Snapshot copies.
  - This option is valid only for a qtree SnapMirror replication.

- `-s` *snapshot_name* specifies an existing Snapshot copy of a source qtree to be transferred. This prevents the normal action of the source creating a Snapshot copy to transfer. SnapMirror does not lock or delete this Snapshot copy.

  Note: This option is valid only for a qtree SnapMirror replication.

dest_system is the name of the destination system.

dest_volume or /*vol*/*qtree_path* is the destination volume or qtree. If it is a scheduled destination as specified in the `/etc/snapmirror.conf` file, that source volume or qtree is considered to be the source. If the destination volume or qtree specified is not in a scheduled relationship, then the `-S` option must be used to provide a source.

SnapMirror identifies the newest common Snapshot copy, which is to be used as the base for resynchronization, and generates a list of Snapshot copies on the destination volume that meet the following criteria:

- The Snapshot copies are newer than the base Snapshot copy and are deleted.
- The Snapshot copies are older than the base Snapshot copy and have already been deleted from the source.

Note: For qtree resynchronization, only the common Snapshot copy is displayed.

SnapMirror then prompts you to choose whether to continue.

2. As required, choose one of the actions from the following table:

If...	Then...
You want to: • Reestablish the SnapMirror pair • Delete the listed Snapshot copies on the destination volume (if you are resynchronizing volumes)	Type `y` at the prompt. Result: SnapMirror: • Deletes the listed Snapshot copies on the destination volume (if you are resynchronizing volumes). • Makes the destination read-only. • Initiates an update of the destination.
You do not want to lose the data in a Snapshot copy that was created after the common Snapshot copy on the destination, but you want to resynchronize the two volumes or qtrees after the data is saved	• Type `n` at the prompt. • Manually copy the data you want to save to the source or other volume. • Return to Step 1 to rerun the `snapmirror resync` command.
You do not want to reestablish the SnapMirror relationship	Type `n` at the prompt. Result: SnapMirror terminates the command.

Example

SnapMirror resynchronization for volumes

```
systemB> snapmirror resync systemB:vol2

The resync base snapshot will be vol2(0001234567)_d.4
These newer snapshots will be deleted from the destination:
hourly.0
hourly.1
These older snapshots have already been deleted from the source and
will be deleted from the destination:
vol2(0001234567)_d.3
Are you sure you want to resync the volume?
```

Example

SnapMirror resynchronization for qtrees

```
systemB> snapmirror resync -S systemA:/vol/vol2/qtreeBob systemB:/vol/
vol3/qtreeBak

The resync base snapshot will be vol2(0001234567)_d.4
Data could be lost as a result of this operation.
Are you sure you want to resync the volume?
```

How the snapmirror resync command helps minimize data loss

The `snapmirror resync` command enables you to reestablish a broken SnapMirror relationship without a lengthy baseline transfer.

This command offers the choice of either source or destination to serve as the source in the restarted SnapMirror relationship. It finds the newest common Snapshot copy (NCS) shared by the two volumes or qtrees, and removes all newer information on the system on which the command is run.

Note: The `snapmirror resync` command requires that the two volumes or qtrees have at least one Snapshot copy in common. You can resynchronize a volume or qtree to any other volume or qtree as long as both have at least one Snapshot copy in common.

Resynchronization causes the loss of all data written to the destination after the base Snapshot copy was made. The `snapmirror resync` command informs you what data might be lost during the resynchronization and requests permission to proceed. If you want to save the data on the destination, you can stop the resynchronization, manually copy the desired data elsewhere, and reissue the `snapmirror resync` command to the destination.

Resynchronization of FlexVol volumes

If there are any changes on the destination system that you need to replicate to the source, you can use qtree SnapMirror to resynchronize data.

You can resynchronize the data at the source with any changes made at the destination by using the `snapmirror resync` command. When resynchronizing data, the Data ONTAP version restrictions apply in case of volume SnapMirror.

Related concepts

Comparison between volume SnapMirror and qtree SnapMirror on page 93

Testing database applications: A special use of snapmirror resync

Testing software applications that run on a database can sometimes change or corrupt the database. To ensure that you do not lose data while testing such applications, you can copy the data to another volume for testing purposes, break the SnapMirror relationship and return the destination volume to

writable state, and run the test application on it. Upon completion of the test, you can resynchronize the source and the destination volume.

Before you begin

Ensure that SnapMirror is enabled on the source and destination systems.

About this task

In the following procedure, you can use a combination of the `snapmirror break` and `snapmirror resync` commands to perform the following tasks:

- Make a destination volume writable for testing.
- Restore the newly writable volume to its original state if further testing is required.

Steps

1. Create or choose a volume or qtree to be used as a destination for the volume or qtree containing the database. (This example uses a volume called Test_vol.)
2. On the destination system, enter the following command to make the destination writable.

   `snapmirror break Test_vol`

   For a qtree, the path must be specified as shown in the following example. You must ensure that the qtree is quiesced before breaking the relationship.

   ```
   dst> snapmirror quiesce /vol/dst_vol/Testqtree
   dst> snapmirror break /vol/dst_vol/Testqtree
   ```

3. Run the application on the data in the former destination (Test_vol).
4. Check the data in the former destination (Test_vol).
5. As required, choose one of the actions from the following table.

If...	Then...
The data has been altered in some way that is not useful and you want to import a fresh copy of the data for further testing.	From the destination system, enter the following command: **`snapmirror resync Test_vol`** **Note:** For a qtree, the path must be specified as shown in the following example: `src> snapmirror resync dst_system:/vol/dst_vol/ Testqtree /vol/src_vol/Testqtree` SnapMirror makes the former destination volume into a SnapMirror destination again and updates the destination with the latest data.

If...	Then...
The data has not been altered adversely, or you wish to stop testing.	The task is completed.

6. Repeat Steps 3, 4, and 5, until you are satisfied with the testing.

Retrieving data for disaster recovery: A special use of snapmirror resync

When disaster disables the source of a SnapMirror relationship, you can use the `snapmirror resync` command as part of a strategy to update the repaired source and reestablish the original configuration of the systems.

About this task

In the following example, the original source (the one disabled by the disaster) is `systemA:vol/volA` and the original destination is `systemB:/vol/volB`. You use a combination of `snapmirror break` and `snapmirror resync` or `snapmirror initialize` commands to perform the following tasks:

- Temporarily make `systemB:volB` the source and `systemA:volA` the destination to restore mirrored data back to `systemA:volA` and to update `systemA:volA`.
- Restore `systemA:/vol/volA` and `systemB:volB` to their original roles as SnapMirror source and SnapMirror destination volume.

In this example, all data from the last scheduled SnapMirror Snapshot copy before the source was disabled and all the data written to `systemB:vol/volB` after it was made writable is preserved. Any data written to `systemA:vol/volA` between the last SnapMirror Snapshot copy and the time that `systemA:vol/volA` was disabled is not preserved.

Steps

1. After the source volume (in this case, `systemA:volA`) is disabled, use the `snapmirror break` command to make the destination volume, `systemB:volB`, writable.

   `snapmirror break systemB:volB`

2. Redirect the clients of source `systemA` to source `systemB`.

   The former clients of `systemA` are now accessing and writing to `systemB`.

3. Temporarily make the original source volume a read-only destination volume.

   - If `systemA:volA` is recoverable, and its data is intact, then use the `snapmirror resync` command on `systemA` to resynchronize `systemA` with `systemB`.

     `snapmirror resync -S systemB:VolB systemA:volA`

- If systemA:volA is unrecoverable, make a new volA on systemA, and from systemA, initialize systemA:volA from systemB.

  snapmirror initialize -S systemB:volB systemA:volA

  This command also makes systemA:volA a read-only destination.

Note: These commands need to be performed on the original source system.

4. Redirect the clients from systemB to systemA.

   The clients cannot access or write to systemA:volA, but they are no longer writing new data to systemB:volB.

5. Update systemA:volA from systemB to transfer the latest data from systemB.

   Example

   Perform the following step from systemA:

   snapmirror update -S systemB:volB systemA:volA

6. Use the snapmirror break command to make systemA:volA writable. On systemA, enter the following command:

   snapmirror break volA

7. On systemB, use the snapmirror resync command to make systemB, the original destination, the destination again.

   snapmirror resync volB

Operation of SnapMirror with other features and products

You can use SnapMirror in conjunction with other features and products to support replication requirements between FlexClone volumes, management requirements in Protection Manager, and secondary volume protection requirements associated with SnapVault.

Comparison between SnapMirror and the vol copy command

You can use SnapMirror or the vol copy command to copy volumes from the source to the destination. There are some similarities between them. However, there are significant differences too.

The following points list the similarities between SnapMirror and the vol copy command:

- Both enable you to copy Snapshot copies from a source to a destination volume.
- The source and destination volumes should both be either traditional volumes or FlexVol volumes.

Note: You can use qtree SnapMirror to replicate data between traditional volumes and FlexVol volumes.

- The volumes should be of the same type.

These commands differ in several important ways, as listed in the following table:

SnapMirror	The `vol copy` command
Supports scheduled, incremental updates of Snapshot copies, replicated from the source to the destination volumes	Does not support incremental updates
Supports qtree-level replication between the source and destination systems	Supports only volume replication and not qtree-level replication
Requires an appropriate license	Does not require an additional license

Comparison between qtree SnapMirror and SnapVault

There are several similarities between qtree SnapMirror and SnapVault. However, there are differences in how they are used.

The following table compares qtree SnapMirror with SnapVault:

Qtree SnapMirror	SnapVault
More suitable for providing immediate failover capability.	More suitable where data availability is less critical, and immediate failover is not required.
Uses the same functionality and licensing on the source and destination systems.	Uses SnapVault source system and SnapVault destination system, which provide different functionality.
Transfers can be scheduled at a maximum rate of once every minute.	Transfers can be scheduled at a maximum rate of once every hour.
Every qtree within a source volume uses one Snapshot copy each on the source system.	Only one Snapshot copy is used.
Snapshot copies are deleted by qtree SnapMirror when not required for replication.	Snapshot copies are retained and deleted on a specified schedule.
Relationships can be reversed. This allows the source to be resynchronized with changes made at the destination.	Relationships cannot be reversed. It provides the capability to transfer data from the destination to the source only to restore data. The direction of replication cannot be reversed.

Qtree SnapMirror	SnapVault
Can be used to replicate data between systems running Data ONTAP only.	Can be used to back up both NetApp systems and open systems. However, the destination system should be a NetApp system.

Transfer of LUN clones using qtree SnapMirror

In versions earlier than Data ONTAP 7.3, SnapMirror considers each LUN clone as a new LUN. Therefore, during the initial transfer of the LUN clone, all data from the LUN clone and the original Data ONTAP LUN is transferred to the secondary system.

For descriptions of data backup and restore on volumes containing Data ONTAP LUNs, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

The transfer of LUN clones using SnapMirror works the same way as the transfer of LUN clones using SnapVault in the non-optimized mode.

Note: Qtree SnapMirror transfers LUN clones in the non-optimized mode only. Qtree SnapMirror does not have the option of optimized transfers.

The use of `snapmirror resync` for restoring data to a source qtree with LUN clones is not supported. When you replicate qtrees with LUN clones, each LUN clone within the qtree is stored as a LUN within the destination qtree. Therefore, when you recover data from such a destination qtree, the original LUN clones are restored as complete LUNs.

Attention: If you attempt to recover data from the destination to a source qtree with LUN clones, using a `snapmirror resync` operation, the system displays the following error message:

```
cannot resync as qtree has one or more lun clones
Aborting resync.
```

To recover data for a qtree with LUN clones, you can replicate the destination qtree to a new qtree.

Attention: For a qtree with LUN clones, ensure that the volume has enough free space to store the LUN clones as complete LUNs before you initiate data recovery using qtree SnapMirror.

Related concepts

About LUN clones and SnapVault on page 267

Managing SnapMirror operations through the NetApp Management Console data protection capability

You can use the NetApp Management Console data protection capability graphical user interface to perform some of the management tasks in a SnapMirror environment.

You can perform the following tasks by using the NetApp Management Console data protection capability.

- Creating and managing asynchronous SnapMirror relationships.
- Creating and managing policies for replication and failover.
- Reporting on relationships and lag times.
- Configuring alerts about replication state changes.
- Scheduling replica updates.
- Visualizing relationships.
- Simplifying data services recovery after a failure.

For more information, see the *OnCommand Unified Manager Guide to Common Provisioning and Data Protection Workflows for 7-Mode*.

Managing SnapMirror operations through the OnCommand System Manager

You can use the OnCommand System Manager to perform different SnapMirror operations, such as creating, deleting, and managing SnapMirror relationships.

You can perform the following tasks as part of managing SnapMirror relationships:

- Initializing SnapMirror destinations
- Updating SnapMirror relationships
- Quiescing SnapMirror relationships
- Resuming quiesced SnapMirror relationships
- Breaking SnapMirror relationships
- Resynchronizing and reverse resynchronizing SnapMirror relationships
- Aborting SnapMirror data transfers

For more information, see the *OnCommand System Manager Help For Use With Data ONTAP 7-Mode*.

Use of SnapMirror with SnapDrive

If you are using SnapDrive software, you can use SnapMirror to replicate your data.

For more information, see the *SnapDrive Installation and Administration Guide* for your version of SnapDrive.

SnapDrive supports the use of volume SnapMirror in the asynchronous mode. For volume SnapMirror in the synchronous or semi-synchronous mode, you need to use SnapDrive 5.0.

SnapDrive does not support qtree SnapMirror replication.

SnapMirror and MultiStore

If you are using MultiStore software, you can use SnapMirror to replicate data in the volumes of a vFiler unit, for disaster recovery. You can also use SnapMirror to migrate a vFiler unit from one system to another.

When performing online migration of a vFiler unit, any SnapMirror operations that start a transfer do not work for volumes owned by the vFiler unit. Therefore, any manual or scheduled SnapMirror update operations for the vFiler unit will not work. The following SnapMirror commands are not allowed for use during an online migration of a vFiler unit:

- `snapmirror initialize`
- `snapmirror update`
- `snapmirror resync`

Note: Synchronous SnapMirror is not supported in a nondefault vFiler unit.

For more information about vFiler units, see the *Data ONTAP MultiStore Management Guide for 7-Mode*.

Note:

In a volume SnapMirror relationship, if you rename or move the destination volume or its vFiler unit and perform a snapmirror update operation, then the SnapMirror status entries and softlocks are locked and are not deleted from the source system. To delete these old SnapMirror status entries and softlocks, you must run the `snapmirror release` command on the source system.

Moving Qtree SnapMirror configurations across vFiler units

You can move all the configurations and relationships of the volume from one vFiler unit to another vFiler unit.

Before you begin

You must remove all the SnapMirror related entries and schedules of the volume from the `snapmirror.conf` file of the vFiler unit.

Step

1. Enter the following command on the vFiler unit source:

   vfiler move *source_vfiler destination_vfiler* [-f] [-i *ip_address*] [-i *ip_address*]...] [path [path...]]

   The following qtree SnapMirror configurations can move from one vFiler unit to another vFiler unit:

   - Softlocks
   - Checkpoint
   - Status

Checking qtree SnapMirror transfers in vFiler units

The qtree SnapMirror status displays all the qtree SnapMirror relationships and configurations of a volume in a vFiler unit. You can use the `snapmirror status` command to check the status of the active transfers in the default vFiler unit and the nondefault vFiler unit.

About this task

In the default vFiler unit, the `snapmirror status` command displays all the relationships (active and idle) configured in the default vFiler unit and only the active transfer of all other vFiler units in the volume. In the nondefault vFiler unit, the `snapmirror status` command displays all the relationships (active and idle) of its own volume.

Step

1. Enter the following command to check the status of active transfers:

   vfiler run * snapmirror status

How FlexClone volumes impact SnapMirror

You can create FlexClone volumes from SnapMirror source or destination volumes. However, you should understand the behavior of the resulting FlexClone volume before creating it.

FlexClone volumes create a nearly instantaneous replica of a volume within the same aggregate. For information about FlexClone volumes, see the *Data ONTAP Storage Management Guide for 7-Mode*.

The following two sections list the key differences between cloning volumes that use qtree SnapMirror and cloning volumes that use volume SnapMirror.

Volume SnapMirror and FlexClone

- When a clone is created on a volume SnapMirror destination system, Data ONTAP locks the Snapshot copy that the clone is based on. To protect the clone, Data ONTAP does not allow you to delete this Snapshot copy. Data ONTAP also puts a soft lock on the corresponding Snapshot copy on the SnapMirror source system.
- Although Data ONTAP will not delete this Snapshot copy that is the source of a clone, you can manually delete this Snapshot copy on the SnapMirror source volume. If you delete the Snapshot copy on the source volume, the next SnapMirror update will fail because it attempts to delete the corresponding Snapshot on the destination volume. All SnapMirror updates to the destination volume continue to fail until the clone is destroyed or split.

  Attention: Delete Snapshot copies carefully when SnapMirror and FlexClone are involved.

- Always create a clone from the most recent Snapshot copy in the SnapMirror destination, because that copy is guaranteed to exist in the source volume.

  If a FlexClone volume is created from a Snapshot copy in the destination volume that is not the most recent copy, and that Snapshot copy no longer exists on the source volume, all the SnapMirror updates to the destination volume will fail until the clone is destroyed or split. This happens because SnapMirror update attempts to delete the snapshot copy on the destination system, which is locked due to the creation of a FlexClone volume.

Qtree SnapMirror and FlexClone

- Qtree SnapMirror does not maintain the same Snapshot copies of the volume on the source and destination systems. Because of this characteristic, a FlexClone volume created from a Snapshot copy on the qtree SnapMirror destination does not lock that Snapshot copy on the source volume.
- Accordingly, deleting that Snapshot copy on the source volume has no impact on the replication or the destination volume. Therefore, the advantage of qtree SnapMirror is that a FlexClone volume can live for a long time on the SnapMirror destination system without space implications on the source system.
- If a Snapshot copy is not specified when creating a FlexClone volume on the qtree SnapMirror destination volume, the `vol clone` command creates a new Snapshot copy on that volume.

- If a FlexClone volume is created using the qtree SnapMirror baseline Snapshot copy, the qtree in the FlexClone volume will be writable.
- If a FlexClone volume is created on the qtree SnapMirror destination volume without specifying a backing Snapshot copy for the clone creation, a separate SnapMirror relationship appears in the `snapmirror status` command output.

Setting up a SnapMirror relationship between two FlexClone volumes

The SnapMirror relationship between two FlexClone volumes that have the common base snapshot helps you to achieve a SnapMirror relationship without transferring the common snapshot data again to the destination system. Both the FlexClone volumes should be created from the SnapMirror source and the corresponding SnapMirror destination volumes with a common base snapshot.

About this task

Setting up a SnapMirror relationship between the two FlexClone volumes does not consume any extra space on the parent aggregate for shared Snapshot copies. In this way, you save disk space and network resources. If you delete any of the inherited Snapshot copies from the parent system or transfer new data from the SnapMirror source clone to the SnapMirror destination clone, additional disk space is consumed.

Step

1. Establish the SnapMirror relationship between the two FlexClone volumes by entering the following command on the destination FlexClone volume:

   `snapmirror resync -S *source*:*sourcevol* *destination*:*destinationvol*`

   Note: The `snapmirror resync` command cannot establish the SnapMirror relationship if the background scanners are still processing the parent destination volume.

Example for setting up a SnapMirror relationship between two FlexClone volumes

You have two systems SystemA and SystemB. The SystemA system has a volume volA and the SystemB system has a volume volB. The volA and volB volumes are in the SnapMirror relationship where volA is the source volume and volB is the destination volume. A FlexClone volume cloneA is created from volA and a FlexClone volume cloneB is created from volB with a common base snapshot.

Establish the SnapMirror relationship between two FlexClone volumes by entering the following command on SystemB:

```
snapmirror resync -S SystemA:cloneA cloneB
```

If you are updating the `/etc/snapmirror.conf` file on SystemB, add the following entry to the file:

```
SystemA:cloneA SystemB:cloneB - - - - -
```

After updating the `/etc/snapmirror.conf` file, the following command establishes the SnapMirror relationship between the two FlexClone volumes:

`snapmirror resync cloneB`

Note: The SnapMirror relationship where cloneB is the SnapMirror source and cloneA is the SnapMirror destination is also supported. The cloneB FlexClone volume is created from the parent destination volume and the cloneA FlexClone volume is created from the parent source volume.

Guidelines for creating a clone of a qtree SnapMirror destination volume

When using the `vol clone create` command to create a clone of a qtree SnapMirror destination volume, you need to consider the status of any qtree SnapMirror transfers for that volume.

You can check the status of SnapMirror transfers by using the `snapmirror status -l` command.

The following table describes the different scenarios possible, when creating a clone of a qtree SnapMirror destination volume:

If the status of qtree SnapMirror transfers for a specific volume is...	**Then, for the creation of a clone of the qtree SnapMirror destination...**
Idle	The clone will be writable. **Note:** It is recommended that you create a clone of a qtree SnapMirror destination, when the status is `idle`.
Active	You can specify a Snapshot copy that is older than any of the Snapshot copies being used for transfers. In this case, the clone will be writable.
	If you do not specify a Snapshot copy for the creation, the clone will be read-only. For a read-only clone created when a SnapMirror transfer is active for the volume, the SnapMirror relationship is active for the clone. Therefore, for the clone, you need to quiesce the SnapMirror transfer and break the SnapMirror relationship. This would make the clone writable. **Note:** The original SnapMirror transfer or relationship need not be modified. You only need to change the status of the clone to make it writable.

Note: If you use the `snapmirror quiesce` or `snapmirror break` command on a writable clone, then the system displays an error message and proceeds without effect.

For more information about creating volume clones, see the *Data ONTAP Storage Management Guide for 7-Mode*.

How SnapMirror works with the dump command

You can use the `dump` command to back up data from a SnapMirror destination volume. The `dump` command picks the most recent Snapshot copy and copies that to tape.

Note: The use of the `dump` command to copy data from a synchronous SnapMirror destination volume is not supported.

You can back up any Snapshot copy displayed by the `snap list` command on the destination. You can also create a Snapshot copy on the source volume, copy the Snapshot copy to the destination, and use the `dump` command to back up this Snapshot copy from the destination to tape.

Effect of the dump command on the SnapMirror destination update schedule

Running the `dump` command on a SnapMirror destination affects SnapMirror operations on that destination in several ways.

- Scheduled incremental SnapMirror updates of a destination volume can occur concurrently with a `dump` command operation to tape; however, if a scheduled SnapMirror update to the destination volume involves the deletion of a Snapshot copy that the `dump` command operation is currently writing to tape, the SnapMirror update will be delayed until the `dump` command operation is complete.

  Note: SnapMirror updates of a destination qtree are not affected by `dump` command operations under any circumstances.

- The operation of `snapmirror break`, `snapmirror resync`, and `snapmirror migrate` commands cannot be carried out concurrently with the operation of the `dump` command.

Protection of SnapVault secondaries using volume SnapMirror

Volume SnapMirror protects SnapVault secondaries by creating SnapMirror relationships to migrate data from the volumes on the SnapVault secondary system to volumes on a remote (tertiary) system running Data ONTAP. SnapMirror provides an exact replica of the SnapVault secondary data on the tertiary system.

The advantage of protecting SnapVault secondaries using volume SnapMirror is that soft lock support enables you to continue SnapVault relationships between the original SnapVault primary system and the tertiary system, without initial baseline transfers.

Note: To deploy this solution, ensure that you have the appropriate licenses for SnapMirror and SnapVault. You cannot execute SnapVault commands on the new secondary system (which is also a SnapMirror destination) if you do not license `sv_ontap_sec` on this tertiary system.

For example, if your SnapVault secondary system becomes unusable because of a disaster, you can manually redirect your next SnapVault transfers to the tertiary system instead of the unusable

secondary system. Your tertiary system becomes the new SnapVault secondary system, and your SnapVault transfers continue, using the latest Snapshot copy common to both the primary and the tertiary systems.

Migrating SnapVault data using volume SnapMirror

You can migrate a volume that contains SnapVault destination qtrees from one secondary system to a tertiary system without having to perform a baseline transfer.

Before you begin

Ensure that you have Open Systems SnapVault baselines. For example, in the following procedure, consider a baseline of the `bno:C:\500MB` directory was backed up to `system-old:/vol/old_vol/bno_C_500MB`.

Steps

1. Using SnapMirror, replicate the volume from the present secondary system to a volume on the new secondary system.

   #### Example

   To replicate the `old_vol` volume from the `system-old` secondary system to the `new_vol` volume on the `system-new` secondary system, complete the following steps on the new secondary system (`system-new`).

   a. Create the `new_vol` volume.

      ```
      system-new> vol create new_vol 3
      ```

   b. Mark the `new_vol` volume restricted.

      ```
      system-new> vol restrict new_vol
      ```

   c. Transfer the old_vol volume to the `new_vol` volume.

      ```
      system-new> snapmirror initialize -S system-old:old_vol new_vol
      ```

2. Quiesce and break the SnapMirror relationship between the old secondary system and the new secondary system.

   #### Example

   To quiesce and break the SnapMirror relationship between `system-old` and `system-new`, complete the following steps on `system-new`.

   a. **`snapmirror quiesce new_vol`**

   b. **`snapmirror break new_vol`**

3. Check the SnapMirror status and SnapVault status on the new secondary system. SnapMirror status should be `Broken-off`. SnapVault status should be `Snapvaulted` to the new volume on the new secondary system.

   Example

   Perform the following steps from `system-new`.

   a. `snapmirror status`

   ```
   Source             Destination        State
   system-old:old_vol   system-new:new_vol   Broken-off
   ```

   b. `snapvault status`

   ```
   Source         Destination                          State
   bno:C:\500MB   system-new:/vol/new_vol/bno_C_500MB   Snapvaulted
   ```

4. Confirm that SnapVault configuration information is not present on the new secondary system by using the `snapvault status -c` command.

   Example

   Perform the following step from `system-new`.

   `snapvault status -c`

   ```
   Snapvault is ON.
   ```

5. Enable access to the new SnapVault secondary system from the SnapVault primary system using the `options snapvault.access` command.

   Example

   Perform the following step from `system-new`.

   `options snapvault.access host=system-old`

   Note: When using SnapVault, access needs to be specified on both the primary and secondary systems.

6. Add SnapVault configuration information to the registry on the new secondary system using the `snapvault start` command.

   Note: This does not start a new baseline, it updates the registry.

Example

Perform the following step from `system-new`.

`snapvault start -S bno:C:\500MB system-new:/vol/new_vol/bno_C_500MB`

```
SnapVault configuration for the qtree has been set.
Qtree /vol/new_vol/bno_C_500MB is already a replica.
```

7. Confirm that SnapVault configuration information is present on the new secondary system using the `snapvault status -c` command.

Example

Perform the following step from `system-new`.

`snapvault status -c`

```
Snapvault is ON.
/vol/new_vol/bno_C_500MB source=bno:C:\500MB
```

8. Test the new SnapVault relationship by manually updating `system-new`.

   If you are using the CLI to manage your environment, continue to the next step; otherwise, you have completed the task.

Example

Perform the following step from `system-new`.

`snapvault update system-new:/vol/new_vol/bno_C_500MB`

```
Transfer started.
Monitor progress with ‘snapvault status’ or the snapmirror log.
```

9. Re-create any schedules used on the old secondary system to the new secondary system and ensure access permissions are in place.

Use of SnapMirror with S Family storage systems

You can use SnapMirror S Family Edition to replicate data between S Family storage systems.

S Family storage systems only support the following versions of Data ONTAP:

- Data ONTAP 7.2.1 S4
- Data ONTAP 7.2.1 S8
- Data ONTAP 7.2.1 S9

SnapMirror S Family Edition works only for asynchronous replication. When using S Family storage systems as source and destination, you can replicate data in the following ways:

- One to another
- One to many
- Many to one

Note: When using SnapMirror S Family Edition for replicating volumes or qtrees, the maximum number of concurrent replication operations is four.

You can use the `snapmirror resync` command to restore data to an S Family storage system.

SnapMirror S Family Edition also supports replication from an S Family storage system to a FAS series storage system.

Note: In this replication relationship, the FAS series storage system is the destination. Therefore, the FAS series storage system should be using Data ONTAP 7.2.1 or later.

SnapMirror S Family Edition does not support replication from a FAS series storage system to an S Family storage system.

Related references

Maximum number of concurrent replication operations on page 123

SnapMirror and ACLs

SnapMirror and SnapVault support the replication of NFS version 4 access control lists (ACLs).

When replicating ACLs, the destination might or might not understand NFS version 4 ACLs. The following points provide information on the methods in which the ACLs are replicated:

- If the destination can understand NFS version 4 ACLs, the ACL is replicated as is from the source to the destination.
  The access rules applied for the data on the destination is identical to the access rules on the source.
- If the destination can understand NFS version 4 ACLs with a large number of ACEs, the ACL is replicated as is from the source to the destination.
  The access rules applied for the data on the destination is identical to the access rules on the source.
- If the destination cannot understand NFS version 4 ACLs, a new ACL, which the destination can understand is created.
  The access rules in this new ACL are equivalent or stricter than the original ACL. This is done to avoid any security issues.
- If the destination cannot understand NFS version 4 ACLs with a large number of ACEs, a new ACL, which the destination can understand is created.

The access rules in this new ACL are equivalent or stricter than the original ACL. This is done to avoid any security issues.

Note: This alternative also implies that a user might not be given access to a set of data on the destination, although the user has access to the same data on the source.

Volume move and replication

During the cutover phase of volume move operation, you cannot initiate any SnapMirror operation on the volume that is being moved. Also, when the `snapmirror initialize`, `snapmirror update`, and `snapmirror resync` operations are in progress, volume move does not enter into the cutover phase.

The volume move operation might fail, if you perform `snapmirror off`, `snapmirror abort`, `snapmirror break`, `snapmirror quiesce`, or `snapmirror release` operation when the volume move is in progress.

For more information about volume move, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

SnapMirror over Fibre Channel

SnapMirror over Fibre Channel enables you to use the SnapMirror feature over Fibre Channel in a SAN environment.

SnapMirror over Fibre Channel includes all the features that are available with SnapMirror over Ethernet. The operational concepts and the command interfaces are identical for both. However, there are a few differences between them.

Hardware requirements for SnapMirror over FC

You must install a Fibre Channel (FC) adapter on the system for using SnapMirror over FC.

You can install any of the following adapters on the SnapMirror source and destination systems:

Adapter name	Number of FC ports	Port speed
X1124	2	4 Gb
X1142A-R6	2	8 Gb
X1142A-R6-C	2	8 Gb

Note: The adapters that are mentioned in the preceding table are the only ones that support Fibre Channel Virtual Interface (FCVI) Architectural Mapping. FCVI functionality is not supported by other FC adapters or FC ports in the systems.

For more information about the hardware supported with different V-Series models, see the *Hardware Universe*.

When using SnapMirror over FC, these systems run in the switch fabric topology logged in to the FC switch as an F-Port. Each port discovers other FCVI-capable ports by querying the switch name server and then logs in to these ports, if required. Each port is identified by its unique worldwide name (WWN).

The adapter operates as a standard network interface. Each port must be configured with its own IP address and network mask, just as an Ethernet adapter is configured. For an HA pair, the administrator must also configure the partner IP address.

Related tasks

Configuring SnapMirror over Fibre Channel on page 212

Supported Fibre Channel switches

SnapMirror over Fibre Channel works with Fibre Channel switches from the following vendors.

- Brocade
- Cisco

You must have a homogeneous Fibre Channel SAN environment. To comply with SnapMirror over Fibre Channel certification, the switches in the SnapMirror data path must be from a single vendor.

SnapMirror over Fibre Channel topology

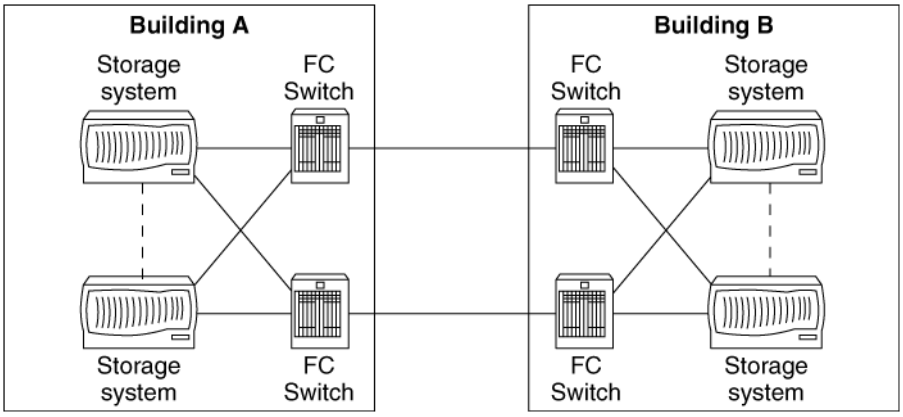
SnapMirror over Fibre Channel is typically used in SAN-based data centers where Fibre Channel infrastructure may already be in use.

SnapMirror traffic between source and destination storage systems travels through the Fibre Channel network. The adapters installed in the storage systems translate the SnapMirror IP packets to and from Fibre Channel frames.

Topology choices

SnapMirror over Fibre Channel requires at least one Fibre Channel switch in the data path.

More complex topologies can involve multiple paths and switches between source and destination storage systems, as shown in the following illustration:



The storage systems in each building are arranged in an HA pair. An HA pair removes single points of failure and provides a more practical, fault-tolerant setup.

Multipath support

SnapMirror over Fibre Channel supports multipath configurations, just as SnapMirror over Ethernet. The configurations are identical.

Private subnet considerations

The storage system needs to direct Fibre Channel Virtual Interface (FCVI) traffic to the Fibre Channel NIC adapter. Therefore, you should configure the IP address of a Fibre Channel NIC port within the range of the private subnet, one that is not globally routable.

The private subnets include:

- 10/8
- 172.16/12
- 192.168/16

Multiple SAN islands support

You might partition your SAN into multiple SAN islands. SnapMirror over Fibre Channel currently supports the Cisco VSAN feature.

Note: The Brocade Multiprotocol Router feature is not supported in the SnapMirror over Fibre Channel data path.

Extended fabric considerations

A typical SnapMirror over Fibre Channel setup involves an extended fabric configuration to mirror data on the primary storage system to a remote site.

The remote site might be quite far from the primary site. To achieve proper operation and expected throughput of the SnapMirror data transfer, you need to follow these guidelines.

- Verify that proper equipment is used in the data path, including:
  - Long-distance, small form-factor pluggable (SFP) optical transceiver modules
  - Long-distance fiber optic cables and repeaters
- Verify that the Fibre Channel switch is extended-fabric capable and has the appropriate license for the capability.
- Configure the inter-switch links (ISLs) to be extended-fabric ports.

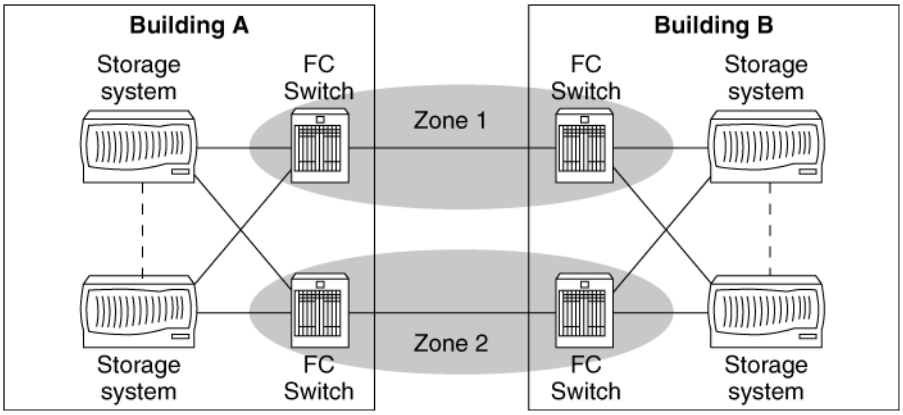
For information about how to check the extended license and configure a switch port to be in extended-fabric mode, see the documentation for the specific switch.

SnapMirror traffic zones

SnapMirror over Fibre Channel requires traffic to occur in a dedicated zone.

In a typical SAN, SnapMirror traffic shares the same physical network with other Fibre Channel traffic. Zoning enables you to partition the servers and storage into different groups, and to enforce access privileges to these groups.

SnapMirror over Fibre Channel supports both WWN zoning and physical port number zoning. When you have a multipath setup, you should use a separated zone for each data path, as shown in the following figure:



Note: For information about detailed zoning configuration procedures, see the documentation for the specific switch.

Related tasks

Configuring SnapMirror over Fibre Channel on page 212

Requirements for deploying SnapMirror over Fibre Channel

You need to ensure that the minimum requirements are met for deploying SnapMirror over Fibre Channel.

- You must use supported switches and compatible firmware.

  Note: To view the supported switches and firmware versions, see *Requirements for SnapMirror over Fibre Channel transport (Asynchronous, Synchronous, and Semi-synchronous modes)* on the NetApp Support Site.

- There must be at least one switch between the source and destination.
- The SAN configuration must be homogenous with respect to the switch vendors.
- The SAN must be enabled for in-order frame delivery (IOD) if there is more than one path between the source and destination systems.
- The path between the source and destination must use dedicated switches or the Storage Area Network (SAN) must be configured such that SnapMirror traffic occurs in a dedicated zone. For Cisco switches, you can use the dedicated VSAN capability.

Related information

NetApp Support Site: support.netapp.com/NOW/knowledge/docs/switches/sm_fc_switch_support

Functionality supported by SnapMirror over Fibre Channel

SnapMirror over Fibre Channel supports both Port-based and Exchange-based SAN routing policies. SnapMirror over Fibre Channel also supports distance extension using dark fibre or DWDM equipment.

The following table explains the fabric configuration options:

If the quantity of ISLs is...	Out-of-order delivery setting on FC NIC on storage controller must be...	Fabric load balancing policy must be...
Greater than 1	`On`	`Exchange`
Equal to 1	`Off`	`Port`

The following functionality is not supported:

- Distance extension solutions, such as Fibre Channel over IP, unless explicitly supported on the Requirements for SnapMirror over Fibre Channel transport (Asynchronous, Synchronous, and Semi-synchronous modes) on the NetApp Support Site.

Related tasks

Configuring SnapMirror over Fibre Channel on page 212

Related references

Troubleshooting issues related to SnapMirror over Fibre Channel on page 222

Configuring SnapMirror over Fibre Channel

To use SnapMirror over Fibre Channel, you must install the Fibre Channel NIC adapters, and set up the SnapMirror relationships. You can also set up multiple SnapMirror traffic zones on supported switches.

Before you begin

You must ensure that the switches and firmware are supported. To view the supported switches and firmware versions, see *Requirements for SnapMirror over Fibre Channel transport (Asynchronous, Synchronous, and Semi-synchronous modes)* on the NetApp Support Site.

About this task

For updated information about the hardware supported with different system models, see the appropriate hardware and service guide and the NetApp Support Site for installation instructions.

Steps

1. Install the Fibre Channel NIC adapters in the source and destination systems.

   See the appropriate hardware and service guide and the NetApp Support Site for installation instructions.

2. Connect the systems to Fibre Channel switches.

   For more information, see the hardware and service guide for your system.

3. Use the `sysconfig` command to identify the Fibre Channel NIC adapters on each system.

   #### Example

   In the following example, the `sysconfig` command shows that the system has a Fibre Channel NIC adapter installed in slot 4:

   ```
   system_1> sysconfig
   Release 7.3: Wed Mar 31 02:47:49 PST 2008
   System ID: 0033587346 (system_1); partner ID 0033586737
   (system_2)
   System Serial Number: 1022105 (system_1)
   System Rev: B0
   Backplane Part Number: 0
   Backplane Rev: B0
   ```

```
Backplane Serial Number: 1022105
slot 0: System Board
Processors:1
Memory Size:3072 MB
CIOB Revision ID:5
slot 0:FastEnet-10/100 Ethernet Controller
e0 MAC Address: 00:a0:98:00:f5:39 (auto-100x-fd-up)
slot 4: FCVI Host Adapter 4a
slot 4: FCVI Host Adapter 4b
slot 6: Gigabit Ethernet Controller IV
e6 MAC Address 00:02:b3:aa:19:d6: (auto-100x-fd-up)
slot 7: FC Host Adapter 7
28 Disks 1904.0GB
slot 8: FC Host Adapter 8
28 Disks: 1904.0GB
slot 10: VI Cluster Adapter
slot 11: NVRAM
Memory Size: 128 MB
```

4. Enter the `sysconfig -v` command to display port connection details, including host port ID acquired from the login to the switch, as well as the ID of the switch port to which it connects.

   Example

   The following command displays port connections for the adapter in slot 4:

```
system_1> sysconfig -v 4
slot 4: FCVI Host Adapter 4a (Dual Channel, QLogic 2312
(2352) rev. 2, 64
-bit,F-port, <UP>
Firmware rev: 3.1.18
Host Loop Id:0xffHost Port Id: 0xa01200
Cacheline size:8FC Packet Size: 2048
SRAM parity:yesExternal GBIC: No
Link Data Rate:2 Gbit
Switch Port brcd_sw_1: 2
slot 4: FCVI Host Adapter 4a (Dual Channel, QLogic 2312
(2352) rev. 2, 64
-bit,F-port, <UP>
Firmware rev: 3.1.18
Host Loop Id:0xffHost Port Id: 0xa01200
Cacheline size:8FC Packet Size: 2048
SRAM parity:yesExternal GBIC: No
Link Data Rate:2 Gbit
Switch Port brcd_sw_2: 2
```

5. Determine the zoning configuration.

   See the documentation for your switch configuration information. You must ensure that the Fibre Channel switches are in fabric mode and support the Simple Name Service (SNS) protocol with support for symbolic names.

Steps 6 through 9 show you how to create new zones and a multipath setup between a source system and a destination system with a Brocade switch. Each system has one X1024 Fibre Channel NIC card installed. The port WWNs are listed in the following table:

Source system	Destination system
port a: 20:00:00:e0:8b:0a:aa:6d	port a: 20:00:00:e0:8b:14:70:af
port b: 20:01:00:e0:8b:2a:aa:6d	port b: 20:01:00:e0:8b:34:70:af

The primary path is formed by `port a` on the source and destination, and the secondary path is formed by `port b` on the source and destination. SnapMirror traffic needs a dedicated zone, and there should be separate zones for the two paths.

Steps 10 through 13 show you how to create new zones and a multipath setup with a Cisco switch. The zoning ideas are the same when using Cisco or Brocade switches. However, the command semantics are different. You must use the `config` command to enter the switch configuration mode and set up the zoning configuration as shown in Steps 14 through 17.

6. Use the Brocade switch to create a new zone with the `zonecreate` command.

   Example

   ```
   brcd_sw_1:root> zonecreate "sm_zone_1", "20:00:00:e0:8b:0a:aa:6d;
   20:00:00:e0:8b:14:70:af"
   brcd_sw_1:root> zonecreate "sm_zone_2", "20:01:00:e0:8b:2a:aa:6d;
   20:01:00:e0:8b:34:70:af"
   ```

7. Create a new zoning configuration by using the `cfgcreate` command.

   Example

   ```
   brcd_sw_1:root> cfgcreate "sm_zone_cfg", "sm_zone_1; sm_zone_2"
   ```

8. Enable the zoning configuration by using the `cfgenable` command.

   Example

   ```
   brcd_sw_1:root> cfgenable "sm_zone_cfg"
   zone config "sm_zone_cfg" is in effect
   Updating flash ...
   ```

9. Check the zoning configuration by using the `cfgshow` command.

Example

```
brcd_sw_1:root> cfgshow
Defined configuration:
cfg: sm_zone_cfg
sm_zone_1; sm_zone_2
zone: sm_zone_1
20:00:00:e0:8b:0a:aa:6d; 20:00:00:e0:8b:14:70:af
zone: sm_zone_2
20:01:00:e0:8b:2a:aa:6d; 20:01:00:e0:8b:34:70:af
Effective configuration:
cfg: sm_zone_cfg
zone: sm_zone_1
20:00:00:e0:8b:0a:aa:6d
20:00:00:e0:8b:14:70:af
zone: sm_zone_2
20:01:00:e0:8b:2a:aa:6d
20:01:00:e0:8b:34:70:af
```

10. Use the Cisco switch to define the two zones by using the following commands:

```
cisco_sw_1(config)# zone name sm_zone_1
cisco_sw_1(config-zone)# member pwwn 20:00:00:e0:8b:0a:aa:6d
cisco_sw_1(config-zone)# member pwwn 20:00:00:e0:8b:14:70:af
cisco_sw_1(config-zone)# zone name sm_zone_2
cisco_sw_1(config-zone)# member pwwn 20:01:00:e0:8b:2a:aa:6d
cisco_sw_1(config-zone)# member pwwn 20:01:00:e0:8b:34:70:af
```

11. Define the zoning configuration by using the following commands:

```
cisco_sw_1(config)# zoneset name sm_zone_cfg
cisco_sw_1(config-zoneset)# member sm_zone_1
```

12. Activate the zoning configuration by using the following commands:

```
cisco_sw_1(config-zoneset)# zoneset activate name sm_zone_cfg
Zoneset activation initiated. check zone status

cisco_sw_1(config-zoneset)# member sm_zone_2
```

13. Check zoning configuration status by using the following commands:

```
cisco_sw_1# show zoneset active
zoneset name sm_zone_cfg
zone name sm_zone_1
pwwn 20:00:00:e0:8b:0a:aa:6d
pwwn 20:00:00:e0:8b:14:70:af
zone name sm_zone_2
pwwn 20:01:00:e0:8b:2a:aa:6d
pwwn 20:01:00:e0:8b:34:70:af
```

14. Determine the IP address and net mask for each port. If you have an HA pair, you also need to decide the IP address of the partner port. These IP addresses must be within the private network IP address range.

    Note: You should configure different private subnets for each Fibre Channel port on the system.

15. Use the `setup` command to configure the IP address.

    This ensures that the changes are committed to non-volatile storage and persist after a system reboot.

Example

The following example shows an abbreviated `setup` command output displayed when configuring a Fibre Channel NIC adapter:

```
system_1> setup
The setup command will rewrite the /etc/rc, /etc/exports, /etc/
hosts, /etc/hosts.equiv, /etc/dgateways, /etc/nsswitch.conf, and /etc/
resolv.conf files, saving the original contents of these files
in .bak files (e.g. /etc/exports.bak).
Are you sure you want to continue? [yes] yes

Release 7.3: Wed Mar 31 02:47:49 PST 2009
System ID: 0033587346 (system_1); partner ID: 0033586737 (system_2)
    System Serial Number: 1022105 (system_1)
    System Rev: B0
    Backplane Part Number: 0
    Backplane Rev: B0
    Backplane Serial Number: 1022105
    slot 0: System Board
        Processors: 1
        Memory Size: 3072 MB
        CIOB Revision ID: 5
    slot 0: FastEnet-10/100 Ethernet Controller
        e0 MAC Address: 00:a0:98:00:f5:39 (auto-100tx-fd-up)
slot 4: FCVI Host Adapter 4b
    slot 4: FCVI Host Adapter 4b
    slot 6: Gigabit Ethernet Controller IV
        e6 MAC Address: 00:02:b3:aa:19:d6 (auto-1000sx-fd-up)
    slot 7: FC Host Adapter 7
        28 Disks: 1904.0GB
        2 shelves with LRC
    slot 8: FC Host Adapter 8
        28 Disks: 1904.0GB
        2 shelves with LRC
slot 10: VI Cluster Adapter
    slot 11: NVRAM
        Memory Size: 128 MB
Please enter the new hostname [system_1]:
Do you want to configure virtual network interfaces? [n]: . . . .
Please enter the IP address for FC Network Interface ql4a
[10.1.1.15]: 10.1.1.15
```

```
Please enter the netmask for FC Network Interface ql4a
[255.255.255.0]: 255.255.255.0
Should interface ql4a take over a partner IP address during failover?
[y]: y
Please enter the IP address or interface name to be taken over by
ql4a [10.1.1.16]: 10.1.1.16
Please enter the IP address for FC Network Interface ql4b
[10.1.2.15]: 10.1.2.15
Please enter the netmask for FC Network Interface ql4b
[255.255.255.0]: 255.255.255.0
Should interface ql4b take over a partner IP address during failover?
[y]: y
Please enter the IP address or interface name to be taken over by
ql4b [10.1.2.16]: 10.1.2.16 . . . .
```

You need to reboot the system for the changes to take effect. You must use the `reboot` command.

16. Reboot the systems and Fibre Channel switches.

17. Use the `ifconfig` command to verify the IP address configuration.

Note: Ensure that IP addresses for the ports on the same fabric have the same net number.

A port on the FC NIC adapter is named as qlxa or qlxb, where:

- ql—Indicates the card vendor. (At this point it is ql only, which stands for QLogic)
- x—The slot number in which the card is installed in the system
- a/b—The FC port number on the card

Example

The following example shows an output for the `ifconfig` command:

```
system_1> ifconfig ql4a
ql4a: flags=840041<UP,RUNNING,LINK_UP> mtu 8160
    inet 10.1.1.15 netmask 0xffffff00 broadcast 0.0.0.0
    partner inet 10.1.1.16 (not in use)
    ether 00:00:00:00:00:00 (VIA Provider)

system_1> ifconfig ql4b
ql4b: flags=840041<UP,RUNNING,LINK_UP> mtu 8160
    inet 10.1.2.15 netmask 0xffffff00 broadcast 0.0.0.0
    partner inet 10.1.2.16 (not in use)
    ether 00:00:00:00:00:00 (VIA Provider)
```

Note: Under normal operation, you should see `UP` and `LINK_UP` in the command output. `UP` means that this interface has been enabled and is working. `LINK_UP` means that the physical connection of the interface is online.

18. Use the `route` command to verify the IP routing table setup.

As this is on a private network, it has a special entry in the routing table.

Example

```
system_1> route -s
Routing tables

Internet:
Destination    Gateway           Flags  Refs    Use Interface
default     harley-29-19.lab.n UGS    4   319989  e0
10.1.1/24   system_1-ql4a      U      0    0      ql4a
10.1.2/24   system_1-ql4b      U      0    0      ql4b
127         localhost          UGS    0    0      lo
localhost   localhost          UH     1    0      lo
```

The preceding entries specify that for IPs within subnet 10.1.1.x, use the ql4a interface; for IPs within subnet 10.1.2.x, use the ql4b interface.

19. Use the `ping` command to verify the connectivity between two network interfaces.

Example

```
system_1> ping -s 10.1.1.26

64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=0 ttl=255 time=0.903 ms
64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=1 ttl=255 time=0.462 ms
64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=2 ttl=255 time=0.439 ms
64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=3 ttl=255 time=0.442 ms
--- 10.1.1.26 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.439/0.561/0.903 ms
```

Note: After the zoning configuration has been changed, you need to verify the connectivity again.

Related information

NetApp Support Site: support.netapp.com

Configuring Exchange-based SAN routing policies for Brocade SAN switches

You can configure the exchange-based SAN routing policies to perform all the SnapMirror features over Fibre Channel in a SAN environment. You can enable the out-of-order delivery (OOD) functionality by using the exchange-based SAN routing policies.

Steps

1. Enable OOD for all FCVI ports on the storage system by using the following command:

```
fcnic ood on <fcvi ql portnum>
```

2. Ensure that the default aptpolicy is the exchange-based SAN routing policy for all the switches and directors in SAN fabric by using the following command:

   ```
   aptpolicy
   ```

3. Enable IODset for all the switches and directors in SAN fabric by using the following command:

   ```
   Iodset
   ```

4. Enable dls for all the switches and directors in SAN fabric by using the following command:

   ```
   dlsset --enable –lossless
   ```

Configuring Port-based SAN routing policies for Brocade SAN switches

You can configure the port-based SAN routing policies to perform all the SnapMirror features over Fibre Channel in a SAN environment. The out-of-order delivery (OOD) functionality is not supported with port-based SAN routing policies.

Steps

1. Set the port-based routing policy for all the switches and directors in SAN fabric by using the following command:

   ```
   aptpolicy 1
   ```

2. Enable IODset for all the switches and directors in SAN fabric by using the following command:

   ```
   Iodset
   ```

3. Disable dls for all the switches and directors in SAN fabric by using the following command:

   ```
   dlsreset
   ```

Configuring Exchange-based SAN routing policies for Cisco SAN switches

You can configure the exchange-based SAN routing policies to perform all the SnapMirror functions over Fibre Channel in a SAN environment. You can enable the out-of-order delivery (OOD) functionality by using the exchange-based SAN routing policies.

About this task

These exchange-based SAN routing policies must be performed on the appropriate vsan (using vsan 10 in the example that follows).

Steps

1. Enable OOD for all FCVI ports on the storage system by using the following command:

   ```
   fcnic ood on <fcvi ql portnum>
   ```

If OOD is enabled on the storage system, the following is the recommended setting for load balancing:

```
Source Destination ExchangeID
switch(config)# vsan database
switch(config-vsan-db)# vsan 10 loadbalancing src-dst-ox-id
```

If OOD is enabled on the storage system, you do not need to enable in-order-guarantee.

2. To check if in-order-guarantee is enabled, enter the following command:

   ```
   show in-order-guarantee
   ```

3. To disable in-order-guarantee for a vsan, enter the following command:

   ```
   no in-order-guarantee vsan 10
   ```

4. Copy the changes to the startup configuration by using the following command:

   ```
   switch# copy running-config startup-config
   ```

Configuring Port-based SAN routing policies for Cisco SAN switches

You can configure the port-based SAN routing policies to perform all the SnapMirror features over Fibre Channel in a SAN environment. The out-of-order delivery (OOD) functionality is not supported with port-based SAN routing policies.

About this task

These port-based SAN routing policies must be performed on the appropriate vsan (using vsan 10 for this example).

Steps

1. Enable in-order-guarantee on the vsan by using the following command:

   ```
   switch# config t
   switch(config)# in-order-guarantee vsan 10
   ```

2. Enable the source and destination routing by using the following command:

   ```
   switch(config)# vsan database
   switch(config-vsan-db)# vsan 10 loadbalancing src-dst-id
   ```

3. Copy the changes to the startup configuration by using the following command:

   ```
   switch# copy running-config startup-config
   ```

Enabling or disabling support for out-of-order frame delivery for SnapMirror over Fibre Channel

Enabling support for out-of-order frame delivery on the SnapMirror source and destination systems changes the default system behavior to allow processing of out of order frames. When enabled, the out-of-order frame delivery makes use of multiple Inter-Switch Links for frame transfers and ensures uninterrupted SnapMirror transfers regardless of the order in which the frames are delivered.

Before you begin

You must ensure that SnapMirror transfer is not in progress.

About this task

Out-of-order frame delivery is not supported with QLogic ISP23xx FCVI cards.

Note: If you want to enable exchange-based policy on the Fibre Channel switches, you must enable out-of-order frame delivery on both the source and destination systems.

Steps

1. Quiesce the SnapMirror destination by entering the following command:

   `snapmirror quiesce`

2. Enable or disable out-of-order frame delivery on SnapMirror source and destination by entering the following command:

   `fcnic ood [on|off|status]` ***`interface`***

   `on`

   Enables support for out-of-order frame delivery.

   Enable out-of-order frame delivery on both the source and destination systems.

   `off`

   Disables support for out-of-order frame delivery.

   Disable out-of-order frame delivery on both the source and destination systems.

   Note: By default, out-of-order frame delivery is disabled.

   `status`

   Displays the status of support for out-of-order frame delivery.

After you finish

Ignore the NIC reset notifications that are displayed after enabling out-of-order frame delivery.

Troubleshooting issues related to SnapMirror over Fibre Channel

You might come across certain issues when setting up a SnapMirror over Fibre Channel connection. The following table lists some of the problems you might encounter and recommends ways to tackle these issues.

Problem	Solution

Connection failure	**1.** Check the cabling between the system and switch and make sure there are no loose connections.
	2. Ping the ports. If you have a good connection, you should see something similar to the following output.
	`filer_1*> ping -s 10.1.1.26`
	`64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=0 ttl=255 time=0.903 ms`
	`64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=1 ttl=255 time=0.462 ms`
	`64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=2 ttl=255 time=0.439 ms`
	`64 bytes from 10.1.1.26 (10.1.1.26): icmp_seq=3 ttl=255 time=0.442 ms`
	`--- 10.1.1.26 ping statistics ---`
	`4 packets transmitted, 4 packets received, 0% packet loss`
	`round-trip min/avg/max = 0.439/0.561/0.903 ms`
	In case of a connection failure, you see something similar to this.
	`filer_1*> ping 10.1.1.26`
	`no answer from 10.1.1.26`

Problem	Solution

The local Fibre Channel NIC port might be offline	**1.** Enter the `ifconfig` command to exam the Fibre Channel NIC port state.
	2. If you do not see the `UP` state in the `ifconfig` output, as shown here, that means the interface has been taken down.
	`filer_1*> ifconfig ql4a`
	`ql4a: flags=800040<RUNNING,LINK_UP> mtu 8160`
	`inet 1.1.1.15 netmask 0xffffff00 broadcast 0.0.0.0`
	`partner inet 1.1.1.16 (not in use)`
	`ether 00:00:00:00:00:00 (VIA Provider)`
	3. Enter the following command to correct this problem.
	`ifconfig` ***`interface_name`*** **`up`**
	4. If you do not see the `LINK_UP` state in the output, the physical connections is offline.
	`filer_1*> ifconfig ql4a`
	`ql4a: flags=40041<UP,RUNNING> mtu 8160`
	`inet 1.1.1.15 netmask 0xffffff00 broadcast 0.0.0.0`
	`partner inet 1.1.1.16 (not in use)`
	`ether 00:00:00:00:00:00 (VIA Provider)`
	5. In this case, check the physical connections including the fiber-optic cable and the FC switch side configuration.

Problem	Solution

The remote Fibre Channel NIC port information might not be correct, and the local port might not “see” the remote port.	**1.** Use the `fcnic` command to verify that a local FC NIC port can “see” remote ports that are zoned together. This command requires diagnostic privilege (`priv set diag` command). A sample output is shown here.
	
	`filer_1*> fcnic show fabric ql4a`
	
	`  pid = 0xa01000 wwn = 20:00:00:e0:8b:14:67:af IP addr(s) = 10.1.1.13`
	`* pid = 0xa01200 wwn = 20:00:00:e0:8b:0a:a8:6d IP addr(s) = 10.1.1.15`
	`  pid = 0xa01300 wwn = 20:01:00:e0:8b:2a:aa:6d IP addr(s) = 10.1.1.16`
	`  pid = 0xa01800 wwn = 20:00:00:e0:8b:14:70:af IP addr(s) = 10.1.1.26`
	
	`Port Login database:`
	
	`  pid = 0xfffffe, lid = 0x7e`
	`  pid = 0xfffffc, lid = 0x80`
	`  pid = 0xa01800, lid = 0xee`
	`  pid = 0xfffffa, lid = 0xef`
	
	2. The entry prefixed by an asterisk (*) is the local entry specified in the command line. This output displays all the FC NIC ports (remote and local) that are zoned together. Any missing entries or any missing information within an entry indicates a connection problem.

Problem	**Solution**
The Fibre Channel switch might not be configured correctly.	**1.** To pinpoint a Fibre Channel connection problem, check the Fibre Channel switch and the fabric to which the system is connected. **2.** Under normal working conditions, an FC NIC port should communicate with the FC switch as an F-Port. The following example shows an output of the `switchshow` command on a Brocade switch. **3.** If the corresponding port does not show up as an F-Port, there are some negotiation problems between the Fibre Channel NIC port on the system and the switch port. Check the switch port configuration, in particular, the port topology and speed configurations.

```
brcd_sw_1:root> switchshow

switchName:     brcd_sw_1
switchType:     9.2
switchState:    Online
switchMode:     Native
switchRole:     Principal
switchDomain:   160
switchId:       fffca0
switchWwn:      10:00:00:60:69:51:58:d9
switchBeacon:   OFF
Zoning:         ON (sm_zone_cfg)

port  0: id N2 Online       F-Port 20:00:00:e0:8b:14:67:af
port  1: id N1 Online       F-Port 20:00:00:e0:8b:0a:16:6e
port  2: id N2 Online       F-Port 20:00:00:e0:8b:0a:a8:6d
port  3: id N2 Online       F-Port 20:00:00:e0:8b:0a:aa:6d
port  4: id N2 No_Light
port  5: id N2 Online       F-Port 21:00:00:e0:8b:0a:15:6e
port  6: id N1 Online       F-Port 20:00:00:e0:8b:14:7c:af
port  7: id N2 Online       F-Port 21:00:00:e0:8b:14:9c:af
port  8: id N2 Online       F-Port 20:00:00:e0:8b:14:70:af
port  9: id N2 No_Light
port 10: id N2 No_Light
port 11: id N2 No_Light
port 12: id N2 No_Light  L2
port 13: id N2 No_Light
port 14: id N2 No_Light
port 15: id N2 No_Light
```

Problem	Solution

The switch name server might not have all the Fibre Channel NIC port entries.	**1.** Make sure that the switch name server database has all the FC NIC port entries. On a Brocade switch, use the `nsallshow` command, the output of which is shown here.
	`brcd_sw_1:root> nsallshow`
	`{`
	`a01000 a01100 a01200 a01300 a01500 a01600 a01700 a01800`
	`8 Nx_Ports in the Fabric }`
	2. This command displays the 24-bit Fibre Channel addresses (PIDs) of all the devices in the fabric. Ensure that this list includes all the FC NIC ports.

Note: If you observe very low throughput compared to the available physical network bandwidth, contact technical support.

Troubleshooting of SnapMirror issues

When using SnapMirror, you might face issues when you change the name of the destination volume. There can also be issues when SnapMirror Snapshot copies are deleted.

What happens if you change a SnapMirror destination volume name

If you change the name of a SnapMirror destination volume, you need to manually correct the SnapMirror relationships affected by the change. SnapMirror is unable to replicate source volume data to a newly named destination volume whose configuration information is incomplete.

In the following case, the destination, `volJobak`, was renamed to `volStatbak`. After the renaming, the `snapmirror status` command does not display the source. Instead, the entry is shown with a dash (–) in the source column.

```
systemB> vol rename volJobak volStatbak

volJobak renamed to volStatbak
you may need to update /etc/exports

systemB> snapmirror status volJobak
Snapmirror is on.

systemB>snapmirror status volStatbak
Snapmirror is on.
Source       Destination         State         Lag        Status
-            systemB:volStatbak  Snapmirrored  -00:03:22  Idle
```

If you change the volume name of a SnapMirror source or destination, you need to make the following changes.

1. Update the `snapmirror.conf` file, if there is an old entry.
2. Use the `snapmirror release` command to update the old destination name, and SnapMirror releases the soft lock and the old Snapshot copy.
3. Use the `snapmirror update` command on the new volume name, and status registry is updated with the new volume name.
4. Update the `/etc/exports` file.

Note: If a system is running at its limit of concurrent transfers, and you attempt to initiate more transfers through by using the `snapmirror update` command, the attempted transfer will fail.

Accidental deletion of SnapMirror Snapshot copies

SnapMirror Snapshot copies stored on either the SnapMirror source or destination location must not be deleted. If the base Snapshot copy (most common Snapshot copy) is accidentally deleted from either the source or destination location, you can attempt recovery.

You might be able to recover without reinitializing the destination by breaking the SnapMirror relationship and then resynchronizing the source and the destination.

As long as there is at least one Snapshot copy common to both the source and the destination, resynchronization will succeed.

If there is no Snapshot copy common to both the source and the destination, you need to use the `snapmirror initialize` command over the network. Or, if the source and destination are volumes, you must use the `smtape backup` command to store the source volume on tape and then use the `smtape restore` command to restore the volume from the tape to the destination.

For more information about `smtape` commands, see the *Data ONTAP Data Protection Tape Backup and Recovery Guide for 7-Mode*.

Related concepts

Conversion of a destination to a writable volume or qtree on page 168

Related tasks

Resynchronizing a SnapMirror relationship on page 187

Space issues when volume space guarantee is enabled for a destination volume

When volume space guarantee is enabled on a SnapMirror destination volume, if the destination volume is larger than the source volume, the destination volume consumes more space than the source volume. You can reduce the size of the destination volume by using the `vol size` command.

Before you reduce the size of the destination volume, compare the size of the destination and source volumes by using the `vol status -b` command. This is because if volume space guarantee is enabled on a volume, space is allocated to that volume based on the size of that volume.

For example, if you create a destination volume that is larger than the source volume or you reduce the size of the source volume after a volume SnapMirror relationship is established, the destination volume is allocated more space than the source volume. In such a scenario, the **`Vol Size`** of the destination volume is larger than the **`FS Size`** or the size of the file system on the destination volume.

Related tasks

Verifying the size of each volume on page 303

Data protection using SnapVault

SnapVault protects data on both NetApp and non-NetApp primary systems by maintaining a number of read-only versions of that data on a SnapVault secondary system and the SnapVault primary system.

What SnapVault is

SnapVault is a disk-based storage backup feature of Data ONTAP. SnapVault enables data stored on multiple systems to be backed up to a central, secondary system quickly and efficiently as read-only Snapshot copies.

In the event of data loss or corruption on a system, backed-up data can be restored from the SnapVault secondary system with less downtime and uncertainty than is associated with conventional tape backup and restore operations.

The following terms are used to describe the SnapVault feature:

- Primary system—a system whose data is to be backed up
- Secondary system—a system to which data is backed up
- Primary system qtree—a qtree on a primary system whose data is backed up to a secondary qtree on a secondary system
- Secondary system qtree—a qtree on a secondary system to which data from a primary qtree on a primary system is backed up
- Open systems platform—a server running AIX, Solaris, HP-UX, Red Hat Linux, SUSE Linux, or Windows, whose data can be backed up to a SnapVault secondary system
- Open Systems SnapVault agent—a software agent that enables the system to back up its data to a SnapVault secondary system
- SnapVault relationship—the backup relationship between a qtree on a primary system or a directory on an open systems primary platform and its corresponding secondary system qtree
- SnapVault Snapshot copy—the backup images that SnapVault creates at intervals on its primary and secondary systems

  SnapVault Snapshot copies capture the state of primary qtree data on each primary system. This data is transferred to secondary qtrees on the SnapVault secondary system. The secondary system creates and maintains versions of Snapshot copies of the combined data for long-term storage and possible restore operations.
- SnapVault Snapshot basename—a name that you assign to a set of SnapVault Snapshot copies using the `snapvault snap sched` command. As incremental Snapshot copies for a set are

taken and stored on both the primary and secondary systems, the system appends a number (0, 1, 2, 3, and so on) to the basenames to track the most recent and earlier Snapshot updates.

- SnapVault baseline transfer—an initial complete backup of a primary storage qtree or an open systems platform directory to a corresponding qtree on the secondary system
- SnapVault incremental transfer—a follow-up backup to the secondary system that contains only the changes to the primary storage data between the current and last transfer actions

Advantages of using SnapVault

The SnapVault disk-based backup and restore system enables you to perform fast and simple data restore operations.

You can also perform the following operations:

- Browse backed-up files online.
- Schedule frequent and efficient backup of large amounts of data.
- Minimize media consumption and system overhead through incremental backup.
- If tape backup is necessary, offload the tape backup task from the primary storage systems to the SnapVault secondary storage system, which centralizes the operation and saves resources.
- Configure and maintain a single storage system for backing up data stored on multiple platforms: Data ONTAP, AIX, Solaris, HP-UX, Linux, Windows, or VMware ESX server systems.

Note: SnapVault behavior is independent of the Data ONTAP version that is installed on the primary or secondary system, and the aggregate type. For example, you can back up data from a primary system that has Data ONTAP 7.3 installed to a secondary system that has Data ONTAP 7.1 installed. You can also back up data from a primary system that has Data ONTAP 7.1 installed to a secondary system that has Data ONTAP 7.3 installed.

What data gets backed up and restored through SnapVault

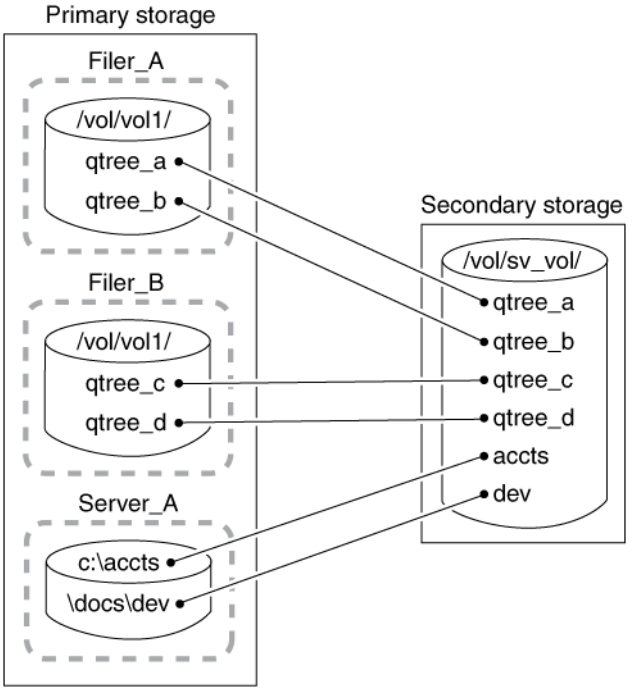
The data structures that are backed up and restored through SnapVault depend on the primary system.

- On systems running Data ONTAP, the qtree is the basic unit of SnapVault backup and restore.
  SnapVault backs up specified qtrees on the primary system to associated qtrees on the SnapVault secondary system. If necessary, data is restored from the secondary qtrees back to their associated primary qtrees.
- On open systems storage platforms, the directory is the basic unit of SnapVault backup.
  SnapVault backs up specified directories from the native system to specified qtrees in the SnapVault secondary system.
  If necessary SnapVault can restore an entire directory or a specified file to the open systems platform.

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

Note: You can back up the qtrees from multiple primary systems, or directories from multiple open systems storage platforms, to associated qtrees on a single SnapVault secondary volume.

The following illustration shows the backup of qtrees and directories on different systems to a single secondary volume:



Note: Starting with Data ONTAP 7.3.2, SnapVault does not support the use of source Snapshot copies created by releases prior to Data ONTAP 6.5.

Types of SnapVault deployment

You can deploy SnapVault in three ways as per business requirements.

- Basic SnapVault deployment
- Primary to secondary to tape backup variation
- Primary to secondary to SnapMirror variation

What basic SnapVault deployment is

The basic SnapVault backup system deployment consists of a primary system and a secondary system.

Primary storage systems

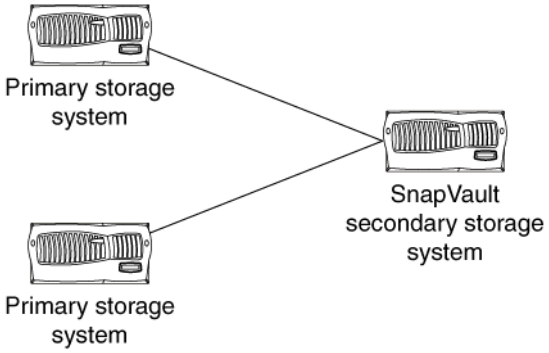
Primary systems are the platforms that run Data ONTAP and open systems storage platforms to be backed up.

- On primary systems, SnapVault backs up primary qtree data, non-qtree data, and entire volumes, to qtree locations on the SnapVault secondary systems.
- Supported open systems storage platforms include Windows servers, Solaris servers, AIX servers, Red Hat Linux servers, SUSE Linux servers, and HP-UX servers. On open systems storage platforms, SnapVault can back up directories to qtree locations on the secondary system. SnapVault can restore directories and single files. For more information, see the *Open Systems SnapVault Installation and Administration Guide*.

Secondary storage system

The SnapVault secondary system is the central disk-based unit that receives and stores backup data from the system as Snapshot copies. You can configure any system as a SnapVault secondary system; however, it is best if you enable `NearStore` option.

The following figure shows a basic SnapVault deployment:



Primary to secondary to tape backup variation

A common variation to the basic SnapVault backup deployment adds a tape backup of the SnapVault secondary system.

This deployment can serve two purposes:

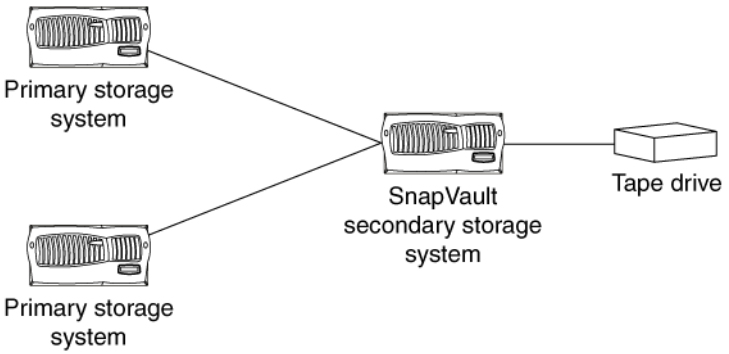
- It enables you to store an unlimited number of network backups offline while keeping the most recent backups available online in secondary storage. This can help in the quick restoration of

data. If you run a single tape backup off the SnapVault secondary storage system, the storage platforms are not subject to the performance degradation, system unavailability, and complexity of direct tape backup of multiple systems.

- It can be used to restore data to a SnapVault secondary system in case of data loss or corruption on that system.

Note: Some UNIX attributes are not preserved using this method; notably, UNIX access control lists (ACLs).

The following figure shows a basic SnapVault deployment with tape backup:



Primary to secondary to SnapMirror variation

In addition to the basic SnapVault deployment, you can replicate the SnapVault secondary using SnapMirror. This protects the data stored on the SnapVault secondary against problems with the secondary system itself.

The data backed up to SnapVault secondary storage is replicated to a SnapMirror destination.

If the secondary system fails, the data mirrored to the SnapMirror destination can be converted to a secondary system and used to continue the SnapVault backup operation with minimum disruption.

How SnapVault backup works

Backing up qtrees using SnapVault involves starting the baseline transfers, making scheduled incremental transfers, and restoring data upon request.

How to start the baseline transfers:

- In response to command-line input, the SnapVault secondary system requests initial base transfers of qtrees specified for backup from a primary storage volume to a secondary storage volume. These transfers establish SnapVault relationships between the primary and secondary qtrees.
- Each primary system, when requested by the secondary system, transfers initial base images of specified primary qtrees to qtree locations on the secondary system.

How to make scheduled incremental transfers:

- Each primary system, in response to command-line input, creates sets of scheduled SnapVault Snapshot copies of the volumes containing the qtrees to be backed up. For tracking purposes, you might name according to frequency, for example, `sv_hourly`, `sv_nightly`, and so on.

  For each Snapshot set, SnapVault saves the number of primary storage Snapshot copies you specify and assigns each Snapshot a version number (0 for most current, 1 for second most recent, and so on).

- The SnapVault secondary system, in response to command-line input, carries out a specified set of scheduled data transfer and Snapshot actions. For each of its secondary qtrees on a given volume, SnapVault retrieves, from the Snapshot data of each corresponding primary qtree, the incremental changes to the primary qtrees made since the last data transfer.

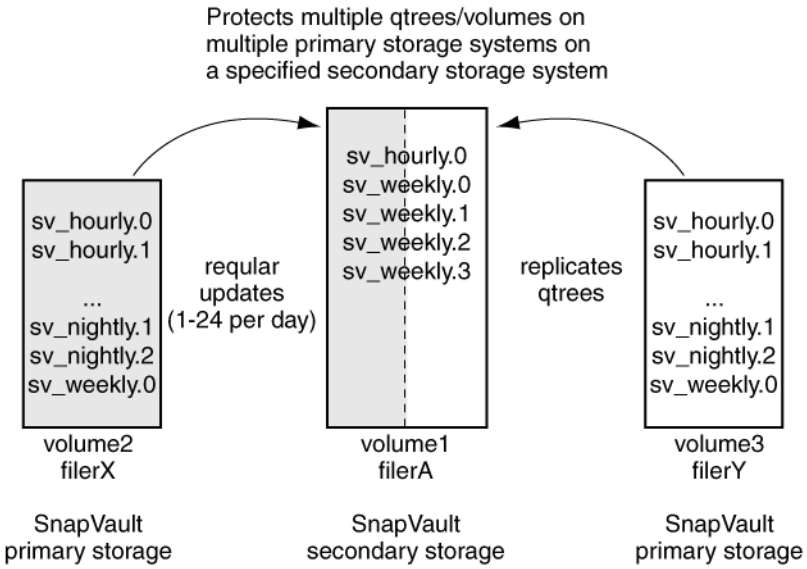
  Then SnapVault creates a volume Snapshot copy of the changes in the secondary qtrees.

  For each transfer and Snapshot set, SnapVault saves the number of secondary storage Snapshot copies that you specify and assigns each Snapshot copy a version number (0 for most current, 1 for second most recent, and so on).

Restoration upon request:

- If data needs to be restored to the primary system, SnapVault transfers the specified versions of the qtrees back to the primary system that requests them.

The following diagram illustrates the SnapVault functionality:



How SnapVault backup works for open systems

The process of SnapVault backup of open systems platform directories involves starting the baseline transfers, scheduling incremental transfers, and restoring data upon request.

Starting the baseline transfers:

- In response to CLI input, the SnapVault secondary system requests from an open systems platform initial baseline image transfers of directories specified for backup. These transfers establish SnapVault relationships between the open systems platform directories and the SnapVault secondary qtrees.
- Each open systems platform, when prompted by the secondary system, transfers initial base images of specified directories to qtree locations on the secondary system.

  Note: There are no primary-side Snapshot copies in Open Systems SnapVault.

Scheduling incremental transfers:

- The SnapVault secondary storage system, in response to CLI input, follows a set of scheduled data transfers (to which, for tracking purposes, you can assign names like “sv_hourly,” “sv_nightly,” and so on).
  To each secondary qtree on a given volume, from a corresponding primary directory on the open systems storage platform, SnapVault transfers the files that have been added or modified since the previous data transfer.
  For each set of scheduled data transfers, SnapVault creates a set of incremental Snapshot copies that capture the changes to the secondary qtrees after each transfer. For each set of Snapshot copies, the SnapVault secondary system saves the number of secondary storage Snapshot copies you specify and assigns each Snapshot copy in the set a version number (0 for most current, 1 for second most recent, and so on).
- Restore upon request: If directory or file data needs to be restored to the open systems storage platform, SnapVault retrieves the data from one of the retained Snapshot copies and transfers the data back to the open systems storage platform that requests it.
  For more information, see the *Open Systems SnapVault Installation and Administration Guide*.

SnapVault support for IPv6

SnapVault supports the use of IPv6 addresses to specify source and destination systems. However, there are some differences between the specification of IPv6 and IPv4 addresses.

When using an IPv6 address with the `snapvault` command, you need to enclose the IPv6 address within square brackets. The usage is shown in the following example.

Use of IPv6 address with the snapvault start command

```
snapvault start -S [2001:0:0:0:0:FFD3:0:57ab]:/vol/vol1/qtree2
system2:/vol/vol1/qtree2
```

You can also specify whether to use an IPv6 connection or an IPv4 connection, for the following three commands by using the `-p` option.

- `snapvault start`
- `snapvault modify`
- `snapvault restore`

The use of the `-p` option is shown in the following command entry.

`snapvault start [-p { inet6 | inet | unspec }] -S system1:/vol/vol1/qtree3 system2:/vol/vol1/qtree3`

- `unspec` is the default value. An IPv6 connection is attempted first. If an IPv6 connection is not established, then an IPv4 connection is attempted.
- `inet6` specifies the use of an IPv6 connection.
- `inet` specifies the use of an IPv4 connection.

Use of IPv6 connection with the snapvault modify command

`snapvault modify -p inet6 -S system1:/vol/vol1/qtree3 system2:/vol/vol1/qtree3`

Data compression with SnapVault

SnapVault operates at the logical level and when data compression is enabled on the source system, the data is uncompressed in memory on the source system before it is backed up.

When data compression is enabled on the source volume, no bandwidth savings are achieved over the network because the data is uncompressed on the source volume before it is sent for replication. If inline compression is enabled on the destination volume, the data is compressed inline at the destination before it is written to the disk. If inline compression is not enabled on the destination volume, you must manually compress the data after the SnapVault transfer is completed to achieve storage space savings in the destination volume.

Note: Inline compression does not guarantee compression of all the data that is being transferred using SnapVault. The space savings at the destination and the source systems are the same if inline compression is enabled on the source system.

For more information about inline compression and how data compression works with SnapVault, see the *Data ONTAP Storage Management Guide for 7-Mode*.

SnapVault considerations when using Data ONTAP Edge storage systems

If you are using SnapVault with Data ONTAP Edge systems installed with the Value bundle, you need to be aware of a few limitations.

Note: These limitations and considerations do not apply to Data ONTAP Edge systems installed with the Premium bundle because those systems can function as a both a SnapVault primary or as a SnapVault secondary.

Consider the following when using SnapVault on Data ONTAP Edge storage systems with the Value bundle:

- These systems can be configured only as a SnapVault primary system, not as a SnapVault secondary system.
- SnapVault primary is supported only when paired with a NetApp storage system configured as a SnapVault secondary system.
- After you start Data ONTAP Edge the first time, you must establish a connection to the SnapVault secondary system within 72 hours or the system will be shut down.

See the *Data ONTAP Edge Installation and Administration Guide* for more information about Data ONTAP Edge storage systems.

Planning SnapVault backups

Before starting SnapVault backups, you need to plan your primary system qtrees or open systems directories and their corresponding secondary system qtrees. You also need to plan the SnapVault backup schedule and Snapshot copy retention, and estimate the initial backup time.

Planning primary and secondary qtree locations

Planning the location of primary system qtrees or open systems directories and their corresponding secondary system qtrees is helpful for better storage management. You can back up multiple qtrees from different volumes and multiple open systems directories to corresponding qtrees in a single volume.

Step

1. List the primary system qtrees or open systems directories and their corresponding secondary system qtrees in a table. For an example, see the following table.

Primary system qtree or directory location	Corresponding secondary system qtree location
systemA:`/vol/vol1/qtreeAA`	sv_secondary:`/vol/sv_vol/qtreeAA`

Primary system qtree or directory location	Corresponding secondary system qtree location
systemA:/vol/vol1/qtreeAB	sv_secondary:/vol/sv_vol/qtreeAB
systemB:/vol/vol1/qtreeBB	sv_secondary:/vol/sv_vol/qtreeBB
winsrvrA:c:\melzdir	sv_secondary:/vol/sv_vol/melzdir
ux_srvrB:/usrs/moz_acct	sv_secondary:/vol/sv_vol/moz_acct

Note: The maximum number of secondary system qtrees per volume is 255. SnapVault backup of qtrees over a Fibre Channel network is not supported.

SnapVault primary and secondary on the same system

In Data ONTAP 7.3 and later, the SnapVault primary and secondary features can be on the same storage system.

The system can be used in the following ways:

- SnapVault destination for one or multiple backup relationships.
- Both SnapVault source and SnapVault destination for the same backup relationship. For example, by using SnapVault you can back up data from FC aggregates to ATA aggregates connected to the same system.

  Note: The source and destination qtrees cannot be within the same volume.

Planning SnapVault backup schedule and Snapshot copy retention

It is important to plan the SnapVault backup schedule and number of Snapshot copies to be retained.

Before you begin

Before you start SnapVault configuration, you should create a table to plan how many Snapshot copies you want per volume, when you want them updated, 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 create a Snapshot copy every hour, every two hours, or every four hours?
- Nightly
  Do you want to create a Snapshot copy every night or just workday nights?
- Weekly
  How many weekly Snapshot copies is it useful to keep?

On storage-based primary systems and SnapVault secondary systems, the data to be backed up is captured and preserved in Snapshot copies.

Steps

1. On the Data ONTAP primary systems, plan the intervals at which to create SnapVault Snapshot copies of your primary system qtrees. The maximum frequency at which you can take Snapshot copies is 1 hour.

   Note: On primary storage platforms not running Data ONTAP, Snapshot copy planning and creation does not apply. For more information, see the Open Systems SnapVault documentation.

2. On the SnapVault secondary system, plan the intervals at which you want to update the secondary system qtrees with data transferred from primary storage platforms, and create SnapVault Snapshot copies to retain that information.

3. Plan how to limit the combined total of Snapshot copies retained on any one volume of the SnapVault secondary system to 255 or fewer.

   Attention: The combined total of Snapshot copies retained on each volume of the SnapVault secondary system cannot exceed 255. If the number of Snapshot copies per volume limit is reached and the old Snapshot copies are not deleted, SnapVault will not create new Snapshot copies on that volume.

Example

In this example, the user is supposed to have 12 qtrees on the secondary system volume.

Snapshot intervals	Primary storage: when created	Primary storage: Snapshot copies retained	Secondary storage: when created	Secondary storage: Snapshot copies retained
weekly	sat @19	4	sat @21	8
nightly	mon-fri @19	10	mon-fri @20	60
hourly	@7-18	11	@8-19	120
Total	n/a	21	n/a	188

On the secondary system, the user schedules the following:

- A weekly update every Saturday at 9:00 p.m. and keeps 8 of them
- A daily update every Monday through Friday at 8:00 p.m. and keeps 60 of them
- An hourly update every hour from 8:00 a.m. to 7:00 p.m. and keeps 120 of them

The result in this example is that 188 Snapshot copies are being kept in the SnapVault secondary system volume.

The limit on Snapshot copies per volume is 251, so the 188 Snapshot copies scheduled in this example do not exceed the volume limit.

If you need to retain more than 251 Snapshot copies on the SnapVault secondary system, you can configure additional volumes on the secondary system. Each additional volume can support 251 additional Snapshot copies.

Estimating the initial backup time

The backup time required for the initial transfer of data from the primary storage system to the secondary storage system depends on the inode count of the data to be backed up, size of the dataset, and capabilities of the network.

SnapVault can perform initial backup transfers at an approximate rate of 7 million inodes per hour (110,000 inodes per minute).

Limit on the number of concurrent SnapVault targets

Starting with Data ONTAP 7.3, the maximum number of concurrent SnapVault transfers of individual qtrees on a storage system is increased. However, the maximum number of concurrent SnapVault targets has not changed.

Before Data ONTAP 7.3, the maximum number of concurrent SnapVault targets supported by a system was equal to the maximum number of concurrent SnapVault transfers possible for the system.

A SnapVault target controls the creation of a scheduled SnapVault Snapshot copy on a SnapVault destination volume. For each SnapVault destination volume that has qtrees being updated, there is a SnapVault target.

The maximum number of concurrent SnapVault targets for each platform is described in the following table. At any point in time, only the number of volumes listed can have their qtrees updated concurrently. If the number of SnapVault targets exceeds the limit, the excess SnapVault targets are queued and executed after the active SnapVault targets complete their backups.

Model	Recommended maximum number of concurrent SnapVault targets
FAS2220	128
FAS2240	128
3140	128
3160	256
3170	256
6040	192

Model	Recommended maximum number of concurrent SnapVault targets
6080	256
3210	128
3220	256
3240	256
3250	256
3270	256
6210	256
6220	256
6240	256
6250	256
6280	256
6290	256
FAS8020	256
FAS8040	256
FAS8060	256
FAS8080	256
FAS2554	192
FAS2552	192
FAS2520	192
CBvM100, FDvM100	32

Note: These maximum numbers apply only to SnapVault targets, and therefore to SnapVault qtrees. There is no restriction on the number of volumes that can be updated concurrently for SnapMirror qtrees.

Although there is a maximum number of concurrent SnapVault targets, you can configure SnapVault relationships in as many volumes as required. However, at any point in time, only the qtrees in the limited number of volumes can be updated.

If the SnapVault backup throughput is limited by the bandwidth or latency of individual links between multiple primary systems and the secondary system, you can use the enhanced number of concurrent transfer limits available in Data ONTAP 7.3 and later releases to get higher overall throughput. You can use the higher limits by spreading the qtrees in the specified number of volumes.

Related references

Maximum number of concurrent replication operations on page 123

Enabling SnapVault

Setting up SnapVault backup on the primary systems means preparing the primary storage system and SnapVault secondary storage system to perform their backup tasks. In Data ONTAP 8.2 and later, a single SnapVault license is used for SnapVault primary and SnapVault secondary instead of two separate SnapVault licenses. You must license and prepare your storage system before you can use SnapVault to back up data.

Steps

1. Enabling licenses for SnapVault on page 242
2. Setting the snapvault.enable option on page 242
3. Setting the ndmpd option on page 243
4. Setting the snapvault.access option on page 243
5. Turning off boot time cleanup of stale entries on page 244

Enabling licenses for SnapVault

You need to enable the SnapVault license for the SnapVault primary and secondary system. If you are using an HA pair, you must enable the SnapVault license on both the nodes.

Step

1. To enter the SnapVault license key, on the system, enter the following command:

   `license add` ***`xxxxxxx`***

   xxxxxxx is the license key you purchased.

   This setting persists across reboots.

   For more information about entering license keys, see the information about licensing in the *Data ONTAP System Administration Guide for 7-Mode*.

Setting the snapvault.enable option

You can set the `snapvault.enable` option to perform SnapVault data transfers and to create Snapshot copies.

Before you begin

You must have entered the SnapVault license key on the SnapVault secondary and primary systems.

Step

1. On both the primary and secondary systems, enter the following command:

   `options snapvault.enable on`

   This option persists across reboots.

Setting the ndmpd option

The `ndmpd` option enables the NDMP service on each primary system to be backed up.

Step

1. To set the `ndmpd` option, enter the following command:

   `ndmpd on`

   This option persists across reboots. To enable and configure NDMP, see the NDMP management chapter in the *Data ONTAP Data Protection Tape Backup and Recovery Guide for 7-Mode*.

Setting the snapvault.access option

The `snapvault.access` option controls which systems can request data transfers. This option persists across reboots.

Steps

1. **On the primary system**: To set the primary systems to grant access only to the secondary systems, enter the following command:

   `options snapvault.access host=`*`snapvault_secondary`*

   Note: In the `snapvault.access` option, up to 255 characters are supported after `host=`.

   Setting this option on the SnapVault primary system determines which secondary system can access data from that primary system.

2. **On the secondary system**: To allow the primary systems to restore data from the secondary system, enter the following command:

   `options snapvault.access host=`*`snapvault_primary1,snapvault_primary2,...`*

   Setting this option on the SnapVault secondary system determines which SnapVault primary systems can access the secondary system.

   The system must be able to resolve the host name entered as *`snapvault_primary`* to an IP address in the `/etc/hosts` file, or else the system needs to be running DNS or NIS. You can also use the literal IP address instead of the host name. The syntax for specifying which systems are allowed access to the secondary system is described in the na_protocolaccess(8) man page. For more information about the `options` command, see the na_options(1) man page.

The system must be able to resolve the host name entered as *snapvault_secondary* to an IP address in the `/etc/hosts` file, or else the system needs to be running DNS or NIS. You can also use the literal IP address instead of the host name. For details, see the na_protocolaccess(8) man page. For more information about the `options` command, see the na_options(1) man page.

Note: To grant access to any requester, enter `options snapvault.access all`. For security reasons, you should avoid using this option setting.

Example

```
systemB> options snapvault.access host=systemA
```

Example

```
systemA> options snapvault.access host=systemB,systemC
```

Turning off boot time cleanup of stale entries

You can turn off the boot time cleanup of stale entries of the SnapVault configurations. This will prevent the deletion of stale SnapVault schedules and relationships, at boot time.

Step

1. To turn off the boot time cleanup of stale entries, enter the following command:

   ```
   options snapvault.stale_config_cleanup_enable off
   ```

   By default, the value for this option is `on`.

After you finish

You must reboot the storage system after setting this option.

How to start a SnapVault backup relationship

After you have enabled SnapVault on both the primary and secondary storage systems and have provided the primary and secondary storage systems access to each other, you must specify the qtrees or volumes whose data you want transferred from the primary storage system to the SnapVault secondary storage system. You must then perform a complete (baseline) transfer of data from the primary storage system to secondary storage system.

The `snapvault start` command configures the SnapVault relationship between primary qtrees or volumes and secondary qtrees by specifying the following:

- The primary storage system qtrees or volumes to be backed up on the storage system qtrees
- The parameters for the updates from the primary storage system qtrees to the secondary storage system qtrees

These parameters include transfer speed and try count. Try count is the number of times SnapVault attempts to start a transfer before stopping the operation.

- The `snapvault start` command also initiates the baseline transfer of the primary storage system qtree data to the secondary storage system qtree

Guidelines for creating a SnapVault relationship

You need to follow certain guidelines when creating a SnapVault relationship.

When you create a SnapVault relationship, you must be aware of the following guidelines for volumes and qtrees:

- You must establish a SnapVault relationship between volumes that have the same `vol lang` settings.
- After you establish a SnapVault relationship, you must not change the language assigned to the destination volume.
- You must avoid white space (spaces and tab characters) in names of source and destination qtrees.
- You must not rename volumes or qtrees after establishing a SnapVault relationship.
- The qtree cannot exist on the secondary system before the baseline transfer.

Backing up qtree data

You should run a complete SnapVault transfer, by using the `snapvault start` command, for the qtrees to be backed up.

Step

1. At the console for the secondary system, enter the following command:

   `snapvault start -S` ***`prim_system:prim_qtree_path sec_host:sec_qtree_path`***

   The `-S` option specifies the primary system and path. This option must be set the first time the command is run for each primary system qtree you want to copy. It is optional when restarting an initial transfer for a previously configured qtree.

   `prim_system:prim_qtree_path` specifies the qtree on the primary system to be backed up.

   `sec_host` is the name of the destination (secondary system) to which the data from the primary system qtree is transferred. If no secondary system is specified, the local host's name is used.

   `sec_qtree_path` is the path to and includes the name of the qtree on the secondary system.

   Note: The qtree specified for *`sec_qtree_name`* must not exist on the secondary system before you run the `snapvault start` command.

   For information about `snapvault start` command options, see the na_snapvault(1) man page.

Backing up qtree data

```
systemB> snapvault start -S systemA:/vol/vol2/qtree3 /vol/vol1/qtree3
```

SnapVault creates a Snapshot copy of qtree3 in vol2 of the primary system (systemA), copies the data in it to the secondary system (systemB) at `vol/vol1/qtree3`, and configures the qtree for future updates.

Note: The time required for this baseline transfer is limited by the total amount of primary system data to be backed up and by the inode count. SnapVault can carry out initial backup at an approximate rate of 7 million inodes per hour (110,000 inodes per minute). In the first phase of a large transfer, SnapVault creates inodes, and it might appear that no activity is taking place.

What non-qtree data is

Non-qtree data is any data on a storage system that is not contained in its qtrees.

Non-qtree data can include the following items:

- Configuration and logging directories (for example, `/etc` or `/logs`) that are not normally visible to clients
- Directories and files on a volume that has no qtree configured

Backing up non-qtree data

You can replicate and protect non-qtree data in a primary system.

Step

1. From the secondary system, enter the following command:

   `snapvault start -S` ***`prim_system`*****`:/vol/`*****`vol_name/- /vol/vol_name/qtree_name`**

   `-S` ***`prim_system`*****`:/vol/`*****`vol_name/-`** specifies the volume on the primary system whose non-qtree data you want to backup.

   The dash (-) indicates all non-qtree data in the specified volume.

   `/vol/`*`vol_name/qtree_name`* specifies the qtree in the secondary system where you want to store this data.

   Note: The qtree that is specified for `/vol/`*`vol_name/qtree_name`* must not exist on the secondary system before you run the `snapvault start` command.

   Note: The non-qtree part of the primary system volume can be replicated only to the SnapVault secondary system. The data can be restored to a qtree on the primary system, but cannot be restored as non-qtree data.

Example

```
systemB> snapvault start -S systemA:/vol/vol1/- /vol/vol4/
non_qtree_data_in_vol7
```

SnapVault transfers the non-qtree data on primary `systemA` to the qtree called `non_qtree_data_in_vol7` in `vol4` on `systemB` (the secondary system). It also configures the qtree for future updates.

What volume data backup involves

When you back up a source volume using SnapVault, the volume is backed up to a qtree on the secondary system; therefore, any qtrees in the source volume become directories in the destination qtree.

Reasons for backing up a volume using SnapVault

- You want to back up a volume that contains many qtrees.
- You want the Snapshot copy management that SnapVault provides.
- You want to consolidate the data from several source volumes on a single destination volume.

Limitations to backing up a volume to a qtree

Before you perform a volume-to-qtree backup, consider the following limitations:

- You lose the qtree as a unit of quota management.
  Quota information from the qtrees in the source volume is not saved when they are replicated as directories in the destination qtree.
- You lose qtree security information.
  If the qtrees in the source volume had different qtree security styles, those security styles are lost in the replication to the destination qtree and are replaced by the security style of the volume.
- The use of SnapVault for backing up volumes to qtrees is not integrated with the NetApp Management Console data protection capability.
- It is not a simple process to restore data.
  SnapVault cannot restore the data back to a volume. When restoring data, the original source volume is restored as a qtree. Also, incremental restores are not supported.
- Volume-to-qtree backup is not supported for volumes containing Data ONTAP LUNs.

SnapVault volume limits are subject to the volume limits supported by WAFL. For information about WAFL volume limits on the SnapVault secondary system, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Related concepts

How to start a SnapVault backup relationship on page 244

Related tasks

Unscheduling SnapVault Snapshot copies on page 256

Backing up volume data

You can replicate and protect volume data in a primary system using SnapVault.

Step

1. From the secondary system, enter the following command:

   snapvault start -S *prim_system*:/vol/*volume_name* /vol/*volume_name*/*qtree_name*

   `-S` *`prim_system`*`:/vol/`*`volume_name`* specifies the volume on the primary system whose data you want to backup.

   `/vol/`*`volume_name/qtree_name`* specifies the qtree in the secondary system where you want to store this data.

   Note: The qtree that is specified for `/vol/`*`volume_name/qtree_name`* must not exist on the secondary system before you run the `snapvault start` command.

Example

```
systemB> snapvault start -S systemA:/vol/vol1 /vol/vol4/vol1_copy
```

SnapVault transfers the data in vol1 on primary systemA to the qtree called vol1_copy in vol4 on systemB (the secondary system). It also configures the qtree for future updates.

Restoring a qtree to the original volume structure

You can use the `snapvault restore` command so that the source volume you backed up to a qtree is restored as a qtree on the primary system.

Steps

1. To restore the backed-up qtree to the original volume structure with multiple qtrees on the primary system, re-create all of the qtrees in the volume on the primary system by using the `qtree create` command.

   ```
   pri_system> qtree create /vol/projs/project_x
   ```

2. Restore the data for each qtree by using the `ndmpcopy` command.

   The following command restores data from the backed-up `project_x` directory on the secondary system to the re-created `project_x` qtree on the primary system.

   pri_system> ndmpcopy -sa *username*:*password* sec_system:/vol/vol1/projs/project_x /vol/projs/project_x

For more information about the `ndmpcopy` command, see the *Data ONTAP Data Protection Tape Backup and Recovery Guide for 7-Mode*.

3. Stop qtree updates and remove the qtree on the secondary system by using the `snapvault stop` command. The following command removes the `projs` qtree from the secondary system:

   ```
   sec_system> snapvault stop /vol/vol1/projs
   ```

4. Reinitialize a baseline copy of each qtree to the secondary system by using the `snapvault start` command. The following command reinitializes the SnapVault backup:

   ```
   sec_system> snapvault start -S pri_system:/vol/projs /vol/vol1/projs
   ```

What SnapVault Snapshot copy update schedules are

After you have completed the initial baseline backup of qtrees on the primary system to qtrees on the SnapVault secondary system, you must use the `snapvault snap sched` command to schedule a set of Snapshot copies on the SnapVault primary system. You can also specify the volume to create Snapshot copies for, the Snapshot copy basename, how many versions of the Snapshot copies to retain, and the days and hours to create this set of Snapshot copies.

You must use the `snapvault snap sched` command to schedule the following tasks:

- Regular SnapVault Snapshot copy times of volumes on the primary system to capture new and changed data in the qtrees that have a SnapVault relationship configured through the `snapvault start` command
- Regular transport of new or modified data in the primary qtrees to their associated secondary qtrees on the SnapVault secondary system
- Regular Snapshot copies of the volume containing the updated secondary qtrees for archiving by SnapVault

Note: For descriptions and procedures pertaining to SnapVault backup of open systems drives and directories, see the Open Systems SnapVault documentation.

How to avoid Snapshot copy schedule conflicts

If SnapVault is scheduled to perform Snapshot copy management at the same time as default `snap sched` activity, then the Snapshot copy management operations scheduled using the `snap sched` command might fail with syslog messages, `Skipping creation of hourly snapshot`, and `Snapshot already exists`.

To avoid this condition, you should disable the conflicting times using `snap sched`, and use the `snapvault snap sched` command to configure equivalent schedules to create Snapshot copies.

Note: You can disable the `snap sched` schedule and only use the `snapvault snap sched` command to create Snapshot copies. Therefore, to track the schedule for creating Snapshot copies, look at the `snapvault snap sched` output, and not the `snap sched` output.

Scheduling Snapshot copies on the SnapVault primary system

To schedule a set of Snapshot copies on the SnapVault primary system, you can use the `snapvault snap sched` command to specify the volume to create Snapshot copies for, the Snapshot copy basename, how many versions of the Snapshot copies to retain, and the schedule to execute this set of Snapshot copies.

Step

1. To set a schedule for the primary system, from the primary system, enter the following command:

   `snapvault snap sched` *`vol_name snap_name schedule_spec`*

   `vol_name` is the name of the volume on the primary system on which to create this set of Snapshot copies.

   `snap_name` is the basename of a Snapshot copy set, for example, `sv_nightly`. The name of this Snapshot copy must be the same on the primary and secondary systems. The *`snap_name`* must not be `hourly`, `nightly`, or `weekly` to avoid conflict with `snap sched` Snapshot copies.

   `schedule_spec` is made up of *`count`*[*`@day_list`*] [*`@hour_list`*].

   `count` is the number of Snapshot copies to retain for this Snapshot copy set. A zero (0) in this field means no new instance of this Snapshot copy will be created.

   Attention: The combined total of Snapshot copies retained cannot exceed 251 Snapshot copies per volume. If it does, SnapVault will not create new Snapshot copies.

   `@day_list` is a comma-separated list that specifies the days on which a new Snapshot copy for this set is created. Valid entries are `mon tue wed thu fri sat sun`. They are not case-sensitive. You can specify a range using a dash (-), for example, `mon-fri`. The dash (-) by itself means no Snapshot copy will be created automatically. You can create the Snapshot copy manually. The default value is `mon-sun`.

   `@hour_list` specifies the hours at which a new Snapshot copy is created for this set. Valid entries are whole numbers from 0 to 23. You can specify a range using a dash (-), or use a comma-separated list, for example, `7, 8-17, 19, 21, 23`. The default is midnight (0).

Scheduling Snapshot copies on the SnapVault primary system

The following three `snapvault snap sched` command lines schedule three sets of SnapVault Snapshot copies on volume `vol1` of the primary system:

```
systemB> snapvault snap sched vol1 sv_weekly 1@sat@19

systemB> snapvault snap sched vol1 sv_nightly 2@mon-fri@19
```

systemB> snapvault snap sched vol1 sv_hourly 11@mon-fri@7-18

Result: SnapVault primary system creates Snapshot copies on the specified volume, as follows:

- SnapVault creates *sv_weekly.0* every Saturday at 7:00 p.m., and keeps one copy.
- SnapVault creates *sv_nightly.0* every Monday through Friday at 7:00 p.m., and keeps two copies.
- SnapVault creates *sv_hourly.0* every Monday through Friday, every hour from 7:00 a.m. to 6:00 p.m., and keeps eleven copies.

Scheduling Snapshot copy backups to the SnapVault secondary system

You can schedule the backup of SnapVault Snapshot copies from the primary systems to the secondary system, by using the `snapvault snap sched -x` command.

Step

1. From the secondary system, enter the following command on a single line:

   snapvault snap sched -x *sec_vol snap_name schedule_spec*

   The `-x` portion of the command is required on the secondary system. This parameter specifies that the SnapVault secondary qtrees on the specified volume are updated from their associated primary system qtrees just before the new Snapshot copy of the specified volume is created.

   sec_vol is the name of the volume on the secondary system for which this Snapshot copy is scheduled.

   snap_name is the basename of the set of Snapshot copies to create, for example, sv_nightly. The basename of this Snapshot set must match the basename of the corresponding Snapshot copy set configured on the primary system volume. Each new Snapshot copy created for this set is numbered 0, the number of each previous Snapshot copy in this set is increased by 1, and the oldest Snapshot copy in this set is deleted. The *snap_name* must not be `hourly`, `nightly`, or `weekly` to avoid conflict with regular Data ONTAP `snap sched` Snapshot copies.

   schedule_spec is made up of *count*[*@day_list*][*@hour_list*].

   - *count* is the number of Snapshot copies to retain for this set. A zero (0) in this field means no new secondary system Snapshot copy will be created for this set, although the qtrees on the secondary system will be updated by the transfers from the primary systems.

     Attention: The combined total of Snapshot copies retained for this and other Snapshot sets cannot exceed 251 Snapshot copies per volume. If it does, SnapVault will not create new Snapshot copies.

- *@day_list* is a comma-separated list that specifies the days on which a new Snapshot copy is created for this set. Valid entries are mon, tue, wed, thu, fri, sat, sun. They are not case-sensitive. You can specify a range using a dash (-), for example, `mon-sun`. The dash (-) by itself means no Snapshot copy will be created automatically. You can create the Snapshot copy manually.
- *@hour_list* specifies the hours at which a new Snapshot copy is created for this set. Valid entries are whole numbers from 0 to 23. You can specify a range using a dash (-), or use a comma-separated list, for example, `6, 8-17, 19, 21, 23`. The default is midnight (0).

  Note: SnapVault transfers scheduled on the secondary system with the `snapvault snap sched -x` command are started five minutes after the hour you specify, to give SnapVault on the primary systems enough time to create Snapshot copies before the secondary system starts the update.

Note: You can turn off the SnapVault Snapshot copy schedule on the primary or secondary system at any time with the `snapvault snap unsched` command. You can also use the `options snapvault.enable off` to stop all SnapVault transfers.

Scheduling Snapshot copy backups to the SnapVault secondary system

The following three `snapvault snap sched` command lines schedule three sets of SnapVault updates and Snapshot copies on volume `vol1` of the secondary systems.

```
systemA> snapvault snap sched -x vol1 sv_weekly 5@sat@21
systemA> snapvault snap sched -x vol1 sv_nightly 5@mon-fri@20
systemA> snapvault snap sched -x vol1 sv_hourly 4@mon-fri@8-19
```

Result: SnapVault transfers qtree data from the primary systems Snapshot copy as follows:

- SnapVault transfers *sv_weekly.0* to the secondary storage system every Saturday at 9:00 p.m., makes a new Snapshot copy with the same name containing all the transferred data, and keeps five copies.
- SnapVault transfers *sv_nightly.0* to the secondary system every Monday through Friday at 8:00 p.m., makes a new Snapshot copy with the same name containing all the transferred data, and keeps five copies.
- SnapVault transfers *sv_hourly.0* to the secondary storage system every hour from 8:00 a.m. to 7:00 p.m., Monday through Friday, makes a new Snapshot copy with the same name containing all the transferred data, and keeps four copies.

Scheduling Snapshot copies on the secondary system for archiving

You might want to schedule some Snapshot copies on the secondary storage system that do not require a transfer from the primary storage system. For example, you might want to maintain hourly and nightly Snapshot copies on the primary storage system and you might want to keep only weekly Snapshot copies on the secondary storage system.

Step

1. To schedule Snapshot copies on the secondary storage system without having to create a corresponding Snapshot copy on the primary storage system, enter the following command:
   `snapvault snap sched` ***`sec_vol snap_name schedule_spec`***

   Note: The `snapvault snap sched` command is used because it waits for any active SnapVault transfers to finish before creating the Snapshot copy.

Example

The following command schedules a weekly Snapshot copy at 11 p.m. on Saturdays and keeps the last five Snapshot copies:

```
snapvault snap sched vol2 sv_weekly 5@sat@22
```

Displaying the currently configured Snapshot copy schedule

You can use the `snapvault snap sched` command without a schedule specification to show the current schedule values.

Step

1. Enter the following command.

   ```
   snapvault snap sched
   ```

   Note: If no *`snap_name`* is given, the command displays the basenames of all Snapshot copy sets scheduled for the specified volume. If no *`volume_name`* is given, the command shows the basenames of all Snapshot copy sets for all SnapVault volumes.

Displaying the SnapVault Snapshot copy schedule

A sample output of the `snapvault snap sched` command is given in the following example.

```
systemA> snapvault snap sched

xfer vol1 sv_weekly 5@sat@21
xfer vol1 sv_nightly 5@mon-fri@20
xfer vol1 sv_hourly 4@mon-fri@8-19
xfer vol2 sv_nightly 5@mon-fri@20
xfer vol2 sv_hourly 4@mon-fri@8-19
```

Preserving older SnapVault Snapshot copies on SnapVault secondary volumes

Data ONTAP 7.3.2 and later enable you to preserve the older SnapVault Snapshot copies on the SnapVault secondary volumes. The preserved SnapVault Snapshot copies are not deleted automatically even if the maximum limit for a specified schedule is reached. If required, delete them manually.

About this task

- In Data ONTAP 7.3.1 and earlier, when the number of SnapVault Snapshot copies reaches the configured limit of copies for the specified schedule, SnapVault deletes the older SnapVault Snapshot copies of the specified schedule to make space available for the new SnapVault Snapshot copies.

  Suppose you want to preserve 250 nightly SnapVault Snapshot copies for five years and the number of SnapVault scheduled Snapshot copies reaches 250, SnapVault automatically deletes the older Snapshot copies to make space for the new SnapVault Snapshot copy. This behavior is not what you want. You want to preserve all the SnapVault Snapshot copies and display a message to indicate that you have reached the limit of SnapVault Snapshot copies for a specified SnapVault Snapshot copy schedule.

- In Data ONTAP 7.3.2 and later, you can preserve the SnapVault Snapshot copies by referring to the following steps:

  1. Enable the `preservesnap` option at the system level or the `preserve` option at an individual schedule level to preserve all SnapVault Snapshot copies on the SnapVault secondary volumes.

     When you enable this option, SnapVault does not delete an older SnapVault Snapshot copy to create a new SnapVault Snapshot copy when the maximum limit of the schedule is reached.

     Note: Once the retention limit for the SnapVault Snapshot copies is reached, further SnapVault updates for the volume will not take place.

  2. Create space for new Snapshot copies when the maximum capacity is reached by doing any of the following tasks:

     - Manually delete some SnapVault Snapshot copies to make space for new Snapshot copies.

- Clone the volume to create new SnapVault Snapshot copies while preserving the older SnapVault Snapshot copies intact on the old base volume.

Step

1. You can prevent automatic deletion of SnapVault Snapshot copies at the backup schedule level or at the global level in a storage system.

If you want to...	Then enter the following command...
Prevent automatic deletion of the SnapVault Snapshot copies at the backup schedule level	**`snapvault snap sched -x -o preserve=on,warn=`*`warn_count vol_name snap_name n`*`@mon-fri@0-23`** • `-x`—specifies the transfer of new data from all primary paths before creating the SnapVault Snapshot copy. • `-o`—sets user-configurable options for this SnapVault Snapshot copy schedule. • `preserve`—can take one of the following values. ◦ `on`—Creates and preserves the SnapVault archive Snapshot copies until the retention limit is reached. Thereafter, SnapVault does not create archive Snapshot copies. ◦ `off`—Automatically deletes the oldest archived SnapVault Snapshot copy to create a new Snapshot copy when the number of Snapshot copies reaches the configured limit in the SnapVault schedule. ◦ `default`—The preserve setting depends on the global `snapvault.preservesnap` setting. • `warn`—sends out the warning EMS and SNMP alerts when the remaining number of SnapVault scheduled Snapshot copies for a target reaches the *`warn_count`* limit set. This option can have a value from 0 to *`n`* - 1. • *`n`*@mon-fri@0-23 is the number of SnapVault Snapshot copies to preserve along with the list of hours or days and is the schedule to trigger the SnapVault target snapshot creation. **Note:** During an upgrade from an earlier Data ONTAP release, the `preserve` option at the individual backup schedule level is set to `default` and `warn` to 0. During a revert operation, all the unknown options are ignored and cleaned up from the registry. For information about `snapvault sched` command options, see the na_snapvault(1) man page.

If you want to...	Then enter the following command...
Prevent auto deletion of SnapVault Snapshot copies at the global level in a system	**`options snapvault.preservesnap on`** `on`—After reaching the retention limit, SnapVault does not automatically delete the oldest SnapVault Snapshot copies to make space available for new SnapVault Snapshot copies. Instead, SnapVault aborts the creation of new SnapVault Snapshot copies. SnapVault executes according to this global option setting when the `preserve` option is not specified or is set to `default` for a given SnapVault schedule. `off`—SnapVault executes according to the option set at the level of backup schedule. **Note:** A backup schedule-level setting of the `preserve` option takes precedence over the global `snapvault.preservesnap` setting.

Preserving SnapVault Snapshot copies on the SnapVault secondary system

For example, you want to create and preserve up to 250 SnapVault Snapshot copies and display a warning message when the number of SnapVault Snapshot copies reaches 240. Because the system does not backup any more SnapVault Snapshot copies once the configured limit for the specified schedule is reached, the warning message is required. The following command enables you to preserve 250 SnapVault Snapshot copies and issue the warning message when the number of SnapVault Snapshot copies reaches 240:

`snapvault snap sched -x -o preserve=on,warn=10 vol1 sv_nightly 250@-`

Unscheduling SnapVault Snapshot copies

You can unschedule a set of SnapVault Snapshot copies if the data in the qtrees you are backing up has been migrated to another location or is no longer useful.

Step

1. To turn off the SnapVault schedule for a set of Snapshot copies and stop the Snapshot copy process for the SnapVault primary system or secondary system, enter the following command at the console of the primary or secondary system:

   `snapvault snap unsched [-f] [`*`volume`*` [`*`snap_name`*`]]`

   `-f` forces the command to run without showing the list of Snapshot copies to stop creating and without asking for confirmation.

   `volume` is the name of the volume to stop creating Snapshot copies on.

   `snap_name` is the basename of the Snapshot copy set to stop creating.

If no value for *snap_name* is provided, SnapVault turns off the SnapVault Snapshot copy schedule for all Snapshot copy sets in the volume and does not update any more SnapVault Snapshot copies in the volume. If no volume is provided, SnapVault deletes the schedules for all SnapVault Snapshot copy sets and does not update any more SnapVault Snapshot copies in the system. If there is already a Snapshot copy being created in the volume, the command fails.

Example

```
systemB> snapvault snap unsched vol1 sv_nightly
```

Unless you used the `-f` option, SnapVault asks for confirmation. If you confirm the action, SnapVault unschedules all SnapVault Snapshot copies with the basename `sv_nightly` on `vol1` of `systemB`.

Disabling Snapshot copies temporarily without unscheduling

You can temporarily disable Snapshot copies without having to unschedule using the `snap sched` command and the `tries` option.

Step

1. To disable Snapshot copies temporarily, enter the following command:

   `snapvault snap sched -o tries=0 *volname snapname sched_spec*`

   Data ONTAP never takes a Snapshot copy because the `tries` option is set to 0.

Enabling Snapshot copies that are temporarily disabled

You can enable Snapshot copies that are temporarily disabled.

Step

1. Enter the following command:

   `snapvault snap sched -o [tries=*option*] *vol_name snap_name sched_spec*`

   `tries` specifies the number of times SnapVault should try to create a Snapshot copy. You can specify a value between 1 and 120, or `unlimited`. The default is `unlimited`. For more information, see the na_snapvault(1) man page.

Checking SnapVault transfers

To ensure SnapVault transfers are taking place as expected, you can check the transfer status using the `snapvault status` command.

Step

1. To check the status of a data transfer and see how recently a qtree has been updated, enter the following command:

   `snapvault status [-l|-s|-c|-t] [[[`*`system_name`*`:]`*`qtree_path`*`] ...]`

   - `-l` displays the long format of the output, which contains more detailed information.
   - `-s` displays the SnapVault Snapshot copy basename, status, and schedule for each volume.
   - `-c` displays the configuration parameters of all SnapVault qtrees on the system. This option can be run only from the secondary system.
   - `-t` displays the relationships that are active.

     Note: A relationship is considered *active* if the source or destination is involved in any one of the following activities: transferring data to or from the network, reading or writing to a tape device, waiting for a tape change, or performing local on-disk processing or clean-up.

   `system_name` is the name of the system for which you want to see the status of SnapVault operations.

   `qtree_path` is the path of the qtree or qtrees for which you want to see the status of SnapVault operations. You can specify more than one qtree path.

   The system displays a message showing whether a transfer is in progress, how much data has been transferred, the state of the destination, and how long ago the last successful transfer took place.

   - If [*`system_name`*:]*`qtree_path`* arguments are specified, then status is displayed only for the specified qtrees.
   - If the `-l` option is given, the output includes the more detailed information shown in Example 2.
   - If the `-s` option is given, the output displays Snapshot copy creation status, as shown in Example 3.
   - If the `-c` option is given, the output displays the parameter settings for the primary system configuration, as shown in Example 4.

- If the `-t` option is given, the output displays the list of relationships that have active transfers as shown in Example 5.

Data ONTAP allows you to set a maximum rate for transfers coming into a system and for transfers going out of a system.

Examples for checking the status

The following examples help you check the data transfer status pertaining to SnapVault backup of storage systems running Data ONTAP only.

Example 1

If you enter `snapvault status` with no option, you see SnapVault qtree relationships involved in the most recent transfer from a primary qtree to a secondary qtree:

```
systemA> snapvault status

Snapvault is ON.
Source                          Destination
State       Lag      Status
systemB:/vol/vol2/qtree3 systemA:/vol/sv_vol/qtree3 Snapvaulted 00:48:24 Idle
```

Example 2

The `snapvault status -l` command displays more detailed information on the most recent SnapVault transfer and Snapshot copy activity:

```
systemA> snapvault status -l

Snapvault is ON.
Source:                      systemA:/vol/vol2/qtree3
Destination             systemB:/vol/sv_vol/qtree3
Status                  Idle
Progress:               -
State:                         SnapVaulted
Lag:                           2:09:58
SnapVault Timestamp:    Thu Jan 31 12:30:01 PST 2002
Base Snapshot:               systemB(0016778780)_sv_vol.59
Current Transfer Type
Current Transfer Error:
Contents:               Replica
Last Transfer Type:        Scheduled
Last Transfer Size:        1680 KB
Last Transfer Duration: 00:02:08-
```

Example 3

The `snapvault status -s` command lists all the Snapshot copies scheduled on the primary or secondary storage system. Information includes volume, Snapshot copy basename, current status, and Snapshot schedule:

```
systemA> snapvault status -s

Snapvault is ON.
Volume     Snapshot   Status  Schedule
------     --------   ------  --------
sv_vol     sv_hourly  Idle    1@0-23
vol2       sv_weekly  Idle    8@fri
```

Example 4

The `snapvault status -c` command lists all the secondary system qtrees, their corresponding primary system qtrees, maximum speed of scheduled transfers, and maximum number of times SnapVault attempts to start a scheduled transfer before skipping that transfer:

```
systemA> snapvault status -c

/vol/sv_vol/db_qtree1 source=systemB:/vol/db1/s1       kbs=unlimited tries=20
/vol/sv_vol/db_qtree2 source=systemC:/vol/db2/s2       kbs=unlimited tries=20
/vol/sv_vol/qtree1    source=systemC:/vol/users/qtree1 kbs=10000     tries=10
/vol/sv_vol/qtree2    source=systemC:/vol/users/qtree2 kbs=10000     tries=10
/vol/sv_vol/qtree3    source=systemD:/vol/users/qtree1 kbs=10000     tries=10
/vol/sv_vol/qtree4    source=systemD:/vol/db/db_1      kbs=7000      tries=10

systemA> snapvault status -c /vol/sv_vol/qtree3

/vol/sv_vol/qtree3 source=systemA:/vol/vol2/qtree3     kbs=10000     tries=10
```

Example 5

The `snapvault status -t` command lists all the SnapVault relationships that have active transfers:

```
systemA> snapvault status -t

Snapvault is ON.
Source                   Destination                    State
Lag      Status
systemB:/vol/vol2/qtree3  systemA:/vol/sv_vol/qtree3   Snapvaulted
00:48:24  Transferring
```

If there are no active transfers, the following message appears:

```
systemA> snapvault status -t

Snapvault is ON.
There are no active transfers.
```

What the status fields mean

You can see the information fields that SnapVault can display for the `snapvault status` and `snapvault status -l` commands.

Field	Possible values that might be displayed
Source	*system:qtree_path*—The source is the primary storage system and qtree path listed.
Destination	*system:qtree_path*—The destination is the secondary storage system and qtree path listed.
State	Uninitialized—The destination SnapVault secondary storage qtree is not yet initialized through the `snapvault start` command. Snapvaulted—The qtree is a SnapVault secondary destination. Unknown—It is not known what state the secondary storage system qtree is in; the volume that contains the secondary storage system qtree could be offline or restricted. Source—This state is reported when the `snapvault status` command is run on the primary storage system. It also appears if `snapvault status` is run on secondary storage systems after the `snapvault restore` command was run on an associated primary storage system.
Lag	*hh*:*mm*:*ss* indicates the time difference between the data currently on the primary storage system and the latest data stored on the secondary storage system; that is, the difference between the current time and the timestamp of the Snapshot copy last successfully transferred to the secondary storage system. A dash (-) in this field means that the secondary storage system is not initialized.

Field	Possible values that might be displayed
Status	Aborting—A transfer is being aborted and cleaned up. Idle—No data is being transferred. Pending—The secondary storage system cannot be updated because of a resource issue; the transfer is retried automatically. Quiescing—The specified qtree is waiting for all existing transfers to complete. The destination is being brought into a stable state. Resyncing—The specified qtree is resynchronizing. Transferring—Transfer has been initiated, but has not yet started, or is just finishing.
Progress	Shows the number of KB transferred by the current transfer, or the restart check point if the status is Idle or Pending.
Mirror Timestamp	*hh*:*mm*:*ss* indicates the timestamp of the last Snapshot copy successfully transferred from the primary storage system to the secondary storage system. **Note:** A resynchronization (`snapvault start -r`) might change the base Snapshot copy to a Snapshot copy with a timestamp older than the original base.
Base Snapshot copy	The name of the base Snapshot copy for the corresponding qtree on the secondary storage system. For qtrees in a SnapVault relationship, the secondary storage side lists the name of the exported Snapshot copy for that qtree on the storage side. A resynchronization (`snapvault start -r`) might change the name of the base Snapshot copy.
Current Transfer Type	Indicates the type of the current transfer: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the destination side.
Current Transfer Error	Displays an error message if the latest transfer attempt failed.
Contents	Indicates whether the contents of the destination volume or qtree in the active file system are up-to-date replicas or are in transition. The field applies only to the destination side. Since a destination is read-only, its contents are always a replica.
Last Transfer Type	Indicates the type of the previous transfer: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the secondary storage side.
Last Transfer Size	Shows the number of KB transferred in the last successful transfer.

Field	Possible values that might be displayed
Last Transfer Duration	Shows the elapsed time for the last successful transfer. If the transfer failed and restarted, the time includes time waiting to restart the transfer. If a transfer aborted and was retried from the beginning, it includes only the time required for the final successful attempt.
Last Transfer From	This field applies only to the secondary storage side and shows the name of the primary system and volume or qtree (where the content is transferred from).

Note: If `snapvault status` displays a negative lag time, it means the clock on the destination storage system is ahead of the clock on the source storage system. The solution is to synchronize the clocks on both the storage systems.

Displaying SnapVault Snapshot copies

You can use the `snap list` command to display a list of Snapshot copies to confirm what versions of your primary qtree data have been backed up, or to locate by date or time a particular version of a qtree to retrieve.

Using the `snap list -q` command, you can see the following:

- A list of all Snapshot copies on the secondary storage system (not just SnapVault Snapshot copies)
- The qtrees in the Snapshot copies
- The primary storage system sources of those qtrees
- The timestamp of the primary storage system Snapshot copy that was the source for the data in the secondary storage system Snapshot copy

Using the `snap list -o` command, you can also list the Snapshot copy timestamps, primary qtree origin (if applicable), and Snapshot copy names stored for an individual qtree.

Note: The descriptions and procedures pertain to SnapVault backup of storage systems running Data ONTAP only. For descriptions and procedures pertaining to SnapVault backup of open systems drives and directories, see the Open Systems SnapVault documentation.

Displaying SnapVault Snapshot copies on a volume

You can display a list of the Snapshot copies and qtrees on your volumes, by using the `snap list` command.

Step

1. On the system for which you want to see the Snapshot copy information, enter the following command:

   `snap list -q [`*`vol_name`*`]`

   `vol_name` is the name of the volume for which you want to list the Snapshot copies.

   If no volume name is given, the Snapshot copies on all this system's volumes are displayed.

   Note: If the deduplication feature is enabled on the SnapVault secondary volume, a deduplication operation is run on the volume after the SnapVault target snapshot has been created. This operation eliminates duplicate data blocks from the volume. After the completion of the deduplication operation, the SnapVault target Snapshot copy that was created earlier is deleted and a new Snapshot copy with the same name is created.

   For more information about deduplication, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Displaying SnapVault Snapshot copies on a volume

Primary storage output example

If you specify the primary volume name, the command lists the information for each Snapshot copy in the volume. This output is from a volume used as a SnapVault primary system:

```
systemA> snap list -q vol2
Volume vol2
working...

    qtree                contents       timestamp     source
    -----                --------       ---------     ------
sv_hourly.0  (Jan 22 20:00)
    qtree1               Original       Jan 22 20:00  -
    qtree2               Original       Jan 22 20:00  -
    qtreeZ               Original       Jan 22 20:00  -
sv_hourly.1  (Jan 22 16:00)
    qtree1               Original       Jan 22 16:00  -
    qtree2               Original       Jan 22 16:00  -
    qtreeZ               Original       Jan 22 16:00  -
sv_hourly.2  (Jan 22 12:00)
    qtree1               Original       Jan 22 12:00  -
    qtree2               Original       Jan 22 12:00  -
    qtreeZ               Original       Jan 22 12:00  -
sv_hourly.3  (Jan 22 08:00)
    qtree1               Original       Jan 22 08:00  -
    qtree2               Original       Jan 22 08:00  -
```

```
    qtreeZ                Original       Jan 22 08:00  -
sv_nightly.0 (Jan 22 00:00)
    qtree1                Original       Jan 22 00:00  -
    qtree2                Original       Jan 22 00:00  -
    qtreeZ                Original       Jan 22 00:00  -
sv_hourly.4  (Jan 21 20:00)
    qtree1                Original       Jan 21 20:00  -
    qtree2                Original       Jan 21 20:00  -
            qtreeZ                 Original       Jan 21 20:00  -
sv_hourly.5  (Jan 21 16:00)
    qtree1                Original       Jan 21 16:00  -
    qtree2                Original       Jan 21 16:00  -
    qtreeZ                Original       Jan 21 16:00  -
sv_nightly.1 (Jan 21 00:00)
    qtree1                Original       Jan 21 00:00  -
    qtree2                Original       Jan 21 00:00  -
    qtreeZ                Original       Jan 21 00:00  -
```

This output displays which qtrees were writable and therefore have original content (the timestamp in these cases is the same as for the Snapshot copy as a whole). It also displays whether any qtrees were transitioning and are therefore neither a faithful replica nor original content. Instead of a timestamp, transitioning qtrees are shown with a dash (-).

Secondary storage output example

If you specify the volume name (in this example, `sv_vol`) and are running the command from a system used as a SnapVault secondary system, you see a list of all the SnapVault Snapshot copies retained on volume `sv_vol` and the details of the qtrees contained in those Snapshot copies:

```
systemB> snap list -q sv_vol
Volume sv_vol
working...

qtree      contents   date        source
--------   ---------  --------    -------
sv_hourly.0 (Jan 31 20:00)
   qtree1  Replica   Jan 22 20:40 systemA:/vol/vol2/qtree1
   qtree2  Replica   Jan 22 20:40 systemA:/vol/vol2/qtree2
   qtreeZ  Replica   Jan 22 20:40 systemA:/vol/vol2/qtreeZ
sv_hourly.1 (Jan 22 16:00)
   qtree1  Replica   Jan 22 16:00 systemA:/vol/vol2/qtree1
   qtree2  Replica   Jan 22 16:00 systemA:/vol/vol2/qtree2
   qtreeZ  Replica   Jan 22 16:00 systemA:/vol/vol2/qtreeZ
sv_hourly.2 (Jan 22 12:00)
   qtree1  Replica   Jan 22 12:00 systemA:/vol/vol2/qtree1
   qtree2  Replica   Jan 22 12:00 systemA:/vol/vol2/qtree2
   qtreeZ  Replica   Jan 22 12:00 systemA:/vol/vol2/qtreeZ
.....
```

This output displays which qtrees are replicas of another qtree, and the timestamp of the source Snapshot copy.

Note: Qtrees that are transitioning appear with a dash (-) instead of a timestamp. In general, you should not attempt to restore Snapshot copy versions of qtrees that are transitioning.

If you specify no arguments, the output displays the information for each Snapshot copy in each volume.

Listing Snapshot copies for qtrees

You can use the `snap list` command to see a list of Snapshot copies associated with a qtree and, if applicable, the Snapshot copies' primary qtree origins.

Step

1. On the system for which you want to see the information, enter the following command:

   `snap list -o [`*`qtree_path`*`]`

   `qtree_path` displays one qtree. If no qtree name is given, information about all the qtree names on the volume is displayed.

Sample snap list -o output

If you specify the `-o` parameter with a qtree path, the `snap list` output includes the dates, sources (if any), and names of associated SnapVault Snapshot copies that are retained on the system, for example:

```
systemB> snap list -o /vol/sv_vol/qtree3
working...
Qtree /vol/sv_vol/qtree3
date          source                     name
------------  --------                   --------
Jan 31 18:00  systemA:/vol/vol2/qtree3  hourly.0
Jan 31 17:00  systemA:/vol/vol2/qtree3  hourly.1
Jan 31 16:00  systemA:/vol/vol2/qtree3  hourly.2
Jan 31 15:00  systemA:/vol/vol2/qtree3  hourly.3
Jan 30 14:00  systemA:/vol/vol2/qtree3  hourly.4
Jan 30 13:00  systemA:/vol/vol2/qtree3  hourly.5
Jan 30 12:00  systemA:/vol/vol2/qtree3  hourly.6
Jan 30 11:00  systemA:/vol/vol2/qtree3  hourly.7
Jan 31 10:00  systemA:/vol/vol2/qtree3  hourly.8
Jan 31 9:00   systemA:/vol/vol2/qtree3  hourly.9
Jan 31 8:00   systemA:/vol/vol2/qtree3  hourly.10
Jan 30 20:00  systemA:/vol/vol2/qtree3  nightly.0
Jan 29 20:00  systemA:/vol/vol2/qtree3  nightly.1
Jan 26 16:00  systemA:/vol/vol2/qtree3  weekly.0
```

About LUN clones and SnapVault

A LUN clone is a space-efficient copy of another LUN. Initially, the LUN clone and its parent share the same storage space. More storage space is consumed only when one LUN or the other changes.

In releases prior to Data ONTAP 7.3, SnapVault considers each LUN clone as a new LUN. Therefore, during the initial transfer of the LUN clone, all data from the clone and the backing LUN is transferred to the secondary system.

Note: LUNs in this context refer to the LUNs that Data ONTAP serves to clients, not to the array LUNs used for storage on a storage array.

For descriptions of data backup and restore on volumes containing LUNs, see the *Data ONTAP SAN Administration Guide for 7-Mode*.

Starting with Data ONTAP 7.3, SnapVault can transfer LUN clones in an optimized way by using SnapDrive for Windows. To manage this process, SnapDrive for Windows creates two Snapshot copies:

- Backing Snapshot copy, which contains the LUN to be cloned
- Backup Snapshot copy, which contains both the LUN and the clone

Modes of transfer

Starting with Data ONTAP 7.3, a SnapVault transfer with LUN clones can run in two modes:

- In non-optimized mode, a LUN clone is replicated as a LUN. Therefore, a LUN clone and its backing LUN get replicated as two separate LUNs on the destination. SnapVault does not preserve space savings that come from LUN clones.
- In optimized mode, a LUN clone is replicated as a LUN clone on the destination. Transfers of LUN clones to the secondary system in optimized mode are possible only with SnapDrive for Windows.

These modes apply to newly created LUN clones. On successive update transfers, only the incremental changes are transferred to the destination in both modes.

Note: A single relationship must either be optimized or non-optimized. Switching between the two modes is not allowed.

LUN clone transfer in non-optimized mode

LUN clones are typically replicated as complete LUNs on the destination system, because non-optimized transfers are used by default.

You must consider the following points before transferring the data in non-optimized mode:

- The backing Snapshot copy must be present on the source system.

- There is no space saving on the destination. The entire LUN clone, and all of the data from the backing LUN, is replicated to the destination.
- The destination qtree should not have LUN clones. If there are LUN clones on the destination qtree, the transfer fails. Before you perform the transfer again, you must delete the LUN clones.

Data restoration for qtrees with LUN clones

If the source qtree has LUN clones, SnapVault does not support in-place restores.

To recover data from the destination qtree using SnapVault, you can use one of the following options for a qtree with LUN clones.

- Delete the LUN clones within the source qtree, and then perform an in-place restore, using the `snapvault restore` command.

  Note: If you attempt an in-place restore for a qtree with LUN clones, the system displays the following error message.

  ```
  Qtree has lun clones
  ```

- Restore the data to a new qtree, by using the `snapvault restore` command.

  Attention: For a qtree with LUN clones, ensure that the volume has enough free space to store the LUN clones as complete LUNs before you initiate data recovery using SnapVault.

LUN clones transfer in optimized mode using SnapDrive for Windows

LUN clones can be replicated in optimized mode when you use SnapDrive for Windows to manage the SnapVault relationship. You need to keep certain conditions in mind when transferring LUN clones.

Consider the following requirements before using SnapVault with LUN clones in optimized mode:

- SnapDrive for Windows only transfers optimized LUN clones supporting Microsoft Windows Volume Shadow Copy Services for TxF Recovery. Volume Shadow Copy Services helps create application-consistent Snapshot copies. SnapDrive for Windows automatically creates these Snapshot copies if the backup application requests TxF Recovery data during Snapshot creation. For more information on Volume Shadow Copy Services and TxF Recovery data, see the Microsoft documentation.
- SnapVault creates LUN clones on the secondary system if both the primary and secondary storage systems are running Data ONTAP 7.3 or later, and optimized transfers occur using SnapDrive for Windows.
- After the relationship has been handed off to SnapDrive for Windows, the relationship must be left there.

Attention: Do not run manual updates from the CLI. This might cause data corruption.

- When creating a SnapVault relationship, ensure that the SnapVault primary system does not contain any LUN clones.
- An initial LUN clone transfer requires the backing LUN and the cloned LUN to be in the same qtree.
- The backing Snapshot copy must exist on the secondary system. It can be missing on the primary system for optimized transfers.
- If the backing Snapshot copy is deleted on the secondary storage system, the transfer of the backup Snapshot copy fails.
  If the backing Snapshot copy is deleted on the primary system, but exists on the secondary system, the transfer succeeds.
- The backing Snapshot copies on the secondary system are locked after the backup Snapshot copy is transferred. To delete a backing Snapshot copy, you should first delete any backup Snapshot copy that depends on it. You can run the `lun snap usage` command to find the backup Snapshot copy.
- During a restore operation, the backing Snapshot copy must be on the primary storage system before the backup Snapshot copy can be transferred. If the backing Snapshot copy already exists on the primary system, there is no need to transfer it again.
- If you have a SnapVault transfer followed by a volume SnapMirror cascade, you should not run volume SnapMirror updates while backing Snapshot copies are being transferred. This might cause SnapVault to use a sub-optimal backing Snapshot copy when the LUN clones are created. No data corruption is possible, but extra Snapshot copies might be locked.
- You should not create Snapshot copies manually when backing Snapshot copies are being transferred.

Note: If SnapDrive for Windows uses the SnapVault relationship running Data ONTAP 7.3 for Volume Shadow Copy Service-based backup, you should not revert to a release earlier than Data ONTAP 7.3.

For more details on how to transfer Volume Shadow Copy Service-based Snapshot copies using SnapDrive for Windows, see the *SnapDrive for Windows Installation and Administration Guide*.

How to change SnapVault settings

You can use the `snapvault modify` command to change the primary system (source) qtree that you specified using the `snapvault start` command. You can change the SnapVault settings for

transfer speed and number of tries before quitting. You might need to make these changes if there are hardware or software changes to the systems.

The meaning of the options is the same as for the `snapvault start` command. If an option is set, it changes the configuration for that option. If an option is not set, the configuration of that option is unchanged.

Note: The descriptions and procedures in this section pertain to SnapVault backup of systems running Data ONTAP only. For descriptions and procedures pertaining to SnapVault backup of open systems drives and directories, see the Open Systems SnapVault documentation.

The `snapvault modify` command is available only from the secondary system. You can also use this command to modify the tries count after the relationship has been set up. This is useful when there is a planned network outage.

You use the `snapvault modify` command to change the source if the primary system, volume, or qtree is renamed. This ensures the continuity of the existing SnapVault relationship between the primary and secondary systems. However, you cannot copy a primary qtree to another volume or system and use this command to take backups from that new location.

If you need to change the SnapVault schedule, use the `snapvault snap sched` command.

Related concepts

How to start a SnapVault backup relationship on page 244

Changing settings for SnapVault backup relationships

You can change the settings for SnapVault backup relationships that you entered with the `snapvault start` command, by using the `snapvault modify` command.

Step

1. From the secondary system, enter the following command on a single line:

   `snapvault modify [-k kbs] [-t n] [-o options] [-S [pri_system:]pri_qtree_path] [sec_system:]sec_qtree_path`

   `-k kbs` specifies a value in kilobytes per second for the throttle (transfer speed) for the primary system. A value of `unlimited` lets the transfer run as fast as it can. Other valid values are whole positive numbers.

   `-t n` specifies the number of times to try the transfer before giving up. The default is 2.

   If set to 0, the secondary system does not update the qtree. This is one way to temporarily stop updates to a qtree.

   options is *opt_name=opt_value* [[, *opt_name=opt_value*]...]. For more details about the available options, see the SnapVault man page.

Note: To ensure that SnapVault does not transfer inodes because of access time changes when a client reads a file, set `-o ignore_atime=on`.

`-S` [*pri_system*:] *pri_qtree_path* specifies the primary storage system for the qtree. [*sec_system*:] *sec_qtree_path* specifies the secondary system for the update.

The following example shows how to modify SnapVault so that it continues backing up the data on the primary system `systemB:/vol2/qtree3` after the name `qtree3` on the primary system changes to `qtreeBob`.

```
systemA> snapvault status

Snapvault server is ON.
Source                   Destination                  State        Lag       Status
systemB:/vol/vol2/qtree3  systemA:/vol/sv_vol/qtree3  Snapvaulted  02:48:24  Idle

[The qtree3 on systemB is renamed qtreeBob.]
```

```
systemA> snapvault modify -S systemB:/vol/vol2/qtreeBob /vol/sv_vol/qtree3
```

```
systemA> snapvault status

Snapvault server is ON.
Source                     Destination                  State        Lag      Status
systemB:/vol/vol2/qtreeBob  systemA:/vol/sv_vol/qtree3  Snapvaulted  0:00:31  Idle
```

Why you manually update a qtree on the secondary system

You can use the `snapvault update` command to manually update the SnapVault qtree on the secondary system from a Snapshot copy on the primary system. You might want to update at an unscheduled time to protect the primary system data.

Manual updates are useful in the following situations:

- A disk failed on the primary system and you want extra protection for the data.
- The nightly backup failed due to a network problem.
- The primary system hardware is going to be reconfigured.
- You want to transfer a Snapshot copy of a quiesced database.

Note: The descriptions and procedures in this section pertain to SnapVault backup of systems running Data ONTAP only. For SnapVault backup of open systems drives and directories, see the Open Systems SnapVault documentation.

Manually updating individual secondary system qtrees

You can manually update a SnapVault qtree on the secondary system, by using the `snapvault update` command.

Step

1. To manually update a SnapVault qtree on the secondary system, enter the following command from the secondary system:

   **`snapvault update [`*`options`*`] [`*`sec_system`*`:]`*`sec_qtree_path`*

   `options` can be one or more of the following:

   - `-k` *`kbs`* overrides the configured rate and specifies a value in kilobytes per second for the throttle (transfer speed) for this Snapshot copy transfer. The default value, `unlimited`, lets the transfer run as fast as it can.
   - `-s` *`snapname`* enables you to specify a primary system Snapshot copy that is more recent than the current base Snapshot copy.

   [*`sec_system`*:]*`sec_qtree_path`* is the name of the secondary system qtree that you want to update.

Examples of how to update the Snapshot copy on the secondary system

The following examples show how to update the Snapshot copy on the secondary system.

Example 1

```
systemB> snapvault update /vol/vol2/qtree3
```

SnapVault updates the qtree on the secondary system (`systemB`) with the data from a new Snapshot copy of the qtree it creates on the primary system (`systemA`). You do not have to specify where the primary system data is, because you already did so when you set up the SnapVault relationship using the `snapvault start` command.

Example 2

To update `qtree3` on the secondary system (`systemB`) with a particular Snapshot copy of `qtree3` on the primary system (`systemA`), you would use the `-s` option to specify the Snapshot copy on the primary system, as in the following example.

```
systemB> snapvault update -s my_snap systemB:/vol/vol0/qtree3
```

SnapVault updates the qtree on the secondary system (`systemB`) with the data from a Snapshot copy of `qtree3` (`my_snap`) that you created earlier on the primary system (`systemA`).

Note: The `snapvault update` command does not create a new Snapshot copy on the secondary system. You need to use the `snapvault snap create` command if you want to create a new Snapshot copy on the secondary system.

Related concepts

Why you create a Snapshot copy manually on page 273

Why you create a Snapshot copy manually

In certain cases, you might want to create a manual (unscheduled) Snapshot copy.

Creating a manual Snapshot copy is useful in these situations:

- You anticipate planned downtime or you need to recover from downtime (during which a Snapshot copy was not taken on time).
- You have just carried out a manual update of a secondary qtree, and you want to immediately incorporate that update into the retained Snapshot copies on the secondary system.

Note: The descriptions and procedures pertain to SnapVault backup of systems running Data ONTAP only. For descriptions and procedures pertaining to SnapVault backup of open systems drives and directories, see the Open Systems SnapVault documentation.

Related tasks

Manually updating individual secondary system qtrees on page 272

Creating a Snapshot copy manually

You can manually create a SnapVault Snapshot copy on the SnapVault primary or secondary system by using the `snapvault snap create` command. For example, if you need to recover from downtime during which a Snapshot copy was not taken on time, you can manually create a Snapshot copy.

Step

1. To create a manual Snapshot copy of a volume, from the primary system or secondary system, enter the following command:

   `snapvault snap create` *`vol_name snap_name`*

   `vol_name` is the name of the volume where the Snapshot copy to be created will reside.

snap_name is the basename of the Snapshot copy to create.

If there is already a Snapshot copy being created in the volume at the time this command is invoked, this command is carried out after the other Snapshot copy is completed.

```
systemB> snapvault snap create vol1 sv_nightly
```

SnapVault creates a new Snapshot copy and, based on the specified Snapshot copy basename, numbers it just as if that Snapshot copy had been created by the SnapVault schedule process. SnapVault names the new Snapshot copy `sv_nightly.0`, renames the older Snapshot copies, and deletes the oldest `sv_nightly` Snapshot copy.

Note: Unlike the `snapvault snap sched -x` command, the `snapvault snap create` command does not update the data in the secondary qtree from the data in the primary qtree prior to creating the new Snapshot copy. If you want to update your secondary qtrees before using the `snapvault snap create` command, use the `snapvault update` command.

Related tasks

Manually updating individual secondary system qtrees on page 272

Specifying a single try for SnapVault Snapshot copy creation

When a Snapshot copy creation fails because the volume is out of space, SnapVault puts the request for the Snapshot copy in a queue and keeps trying until the attempt is successful. Some applications have a limited backup time frame and cannot support retries. In such a case, you might want to stop SnapVault from trying to create the Snapshot copy again.

Step

1. To stop SnapVault from repeatedly trying to create a Snapshot copy, enter the following command:

   `snapvault snap create -o tries=1 *vol_name snap_name*`

   vol_name is the name of the volume where the Snapshot copy is created.

   snap_name is the basename of the Snapshot copy that is created.

   `tries=1` tries to create a Snapshot copy only once.

   For more information about the `tries` option, see the na_snapvault(1) man page.

Renaming a SnapVault or Open Systems SnapVault secondary volume

If you rename a volume involved in a SnapVault relationship, you need to update the SnapVault configurations with the new name.

About this task

If the qtree is not configured in SnapVault, using the `snapvault start` command to configure the qtree gives the following error:

```
Error: Not a snapvaulted qtree, ignoring.
```

This error means that the qtree is not a SnapVault replica.

Steps

1. To rename a volume, enter the following command:

   `vol rename` *`oldvolname newvolname`*

2. To verify the changes, enter the following command:

   `snapvault status -c`

   `snapvault status -c` does not show the new path yet.

3. Enter the following command:

   `snapvault start -S` *`pri_filer`*`:`*`pri_qtree sec_filer`*`:`*`sec_qtree`*

   ```
   Snapvault configuration for the qtree has been set. Qtree /vol/
   newvolname/sec_qtree is already a replica.
   ```

4. Enter the following command:

   `snapvault status -c`

   `snapvault status -c` now shows the new path.

5. Enter the following command to verify that the change was successful:

   `snapvault update` *`sec_qtree`*

6. The output of `snapvault status -c` will contain entries referencing the old volume name in addition to the new volume name. Remove the old entries by using the `snapvault stop` command:

   `snapvault stop /vol/`*`oldvolname`*`/`*`sec_qtree`*

```
Snapvault configuration for the qtree has been deleted. Could not delete
qtree: destination qtree does not exist
```

The output reflects that the configuration information is deleted and that the qtree did not exist on disk. This is normal because the volume name has changed.

Restoring SnapVault data to the primary system

In the event of data loss on a primary system, you might need to restore data from the secondary system.

About this task

Restoring data from the SnapVault secondary system involves the following command-line operations.

- You use the `snapvault restore` command to restore a backed-up qtree saved to the secondary system. Starting with Data ONTAP 7.3, you can restore the data to an existing qtree on the primary system using baseline restore or incremental restore.
  - **Baseline restore**: The primary system must be running Data ONTAP 7.3 or later, and the secondary system can be running any Data ONTAP version.
  - **Incremental restore**: Both the primary and secondary systems must be running Data ONTAP 7.3 or later.

    Note: Starting with Data ONTAP 7.3, the SCSI connectivity of applications to all LUNs within the qtree being restored is maintained throughout the restore process in order to make the restore operation nondisruptive to applications. However, I/O operations are not allowed during the restore operation. Only in-place baseline restores and incremental restores can be nondisruptive.
- After successfully restoring data, you use the `snapvault start -r` command to resume the SnapVault relationship between the restored qtree and its backup qtree on the secondary system (assuming you want to continue SnapVault protection of the data). If you do not want to continue the backup relationship, you use the `snapvault release` command to cancel any further backups of the restored qtree and to release the resources on the secondary system that were used in the SnapVault relationship.
- When several SnapVault restore operations to different primary qtrees in the same volume are running concurrently, Snapshot cleanup might fail due to Snapshot copies being locked by various restore operations. Therefore, some unwanted Snapshot copies might be left behind in the volume.

  You can manually delete these unwanted Snapshot copies. Make sure that you do not delete the base Snapshot copy of the restore operation.

Note: If you do not want to resume the relationship, you can delete the Snapshot copy created by the restore operation. If you want to resume the SnapVault operation, you can delete the Snapshot copy after you have successfully resumed the relationship.

Steps

1. If you intend to restore a primary qtree to the exact qtree location on the primary system from which you backed it up, you can perform Baseline restore or Incremental restore.

   - Baseline restore—The baseline restore can be to an existing qtree or to a non-existing qtree.

     Note: In case of a baseline restore to an existing qtree, the restore operation overwrites the qtree data.

   - Incremental restore—The restore operation transfers only incremental changes from the secondary qtree to the specified primary qtree.

   Note: When restoring an existing primary qtree, an incremental restore is more efficient. If the incremental restore fails, then you can attempt an in-place baseline restore.

2. Enter the following command on the primary system on a single line:

   `snapvault restore [-f] [-k n] [-r] [-w] [-s snapname] -S sec_system:sec_qtree_path [prim_system:]prim_qtree_path`

   `-S` [*`sec_system`*:]*`sec_qtree_path`* specifies the secondary system and qtree path from which you want to restore the data.

   The `-f` option forces the command to proceed without first asking for confirmation from the user.

   The `-k` option sets the maximum transfer rate in kilobytes per second.

   The `-r` option attempts an incremental restore. The incremental restore can be used to revert the changes made to a primary qtree since any backed-up version on the secondary system.

   The `-w` option causes the command not to return after the baseline transfer starts. Instead, it waits until the transfer completes (or fails). At that time, it prints the completion status and then returns. If the command returns successfully, the status of the transfer can be idle or quiescing.

   The `-s` option specifies that the restore operation must use the specified (*`snapname`*) Snapshot copy on the secondary system.

   `prim_system` is the name of the primary system that you want to restore to. If specified, this name must match the name of the host system.

   `prim_qtree_path` is the name of the primary system qtree that you want to restore to.

   For more information about the `snapvault restore` command options, see the na_snapvault(1) man page.

3. As required, choose one of the actions from the following table.

If you want to...	Then...
Resume SnapVault backups of the newly restored qtree on the primary system	Go to Step 4.
Discontinue the backup relationship between the newly restored qtree on the primary system and its secondary qtree partner	Go to Step 6.

4. To resume the SnapVault relationship between the restored qtree and its backup qtree on the secondary system, enter the following command on a single line:

   snapvault start -r [*options*] -S [*prim_system*:]*prim_qtree_path* [*sec_system*:]*sec_qtree_path*

   `-r` is required to restart backups from a restored primary system. For details about the `snapvault` command syntax, see the na_snapvault(1) man page.

5. To discontinue the SnapVault relationship between the restored qtree and its backup qtree on the secondary system, enter the following command on the secondary system:

   snapvault release *sec_qtree_path* [*prim_system*:]*prim_qtree_path*

   sec_qtree_path is the name of the secondary system qtree that you want to release from a SnapVault relationship.

   [*prim_system*:]*prim_qtree_path* is the name of the primary system qtree that you want to release.

Examples of restoring SnapVault data

The following examples show how to restore SnapVault data on the primary storage system.

Example 1

This example shows the primary storage system (systemA) requesting data to be restored from the secondary storage system (systemB) and then the secondary storage system restarting the SnapVault backup relationship.

For incremental restore:

```
systemA> snapvault restore -r -s sv_backup.0 -S systemB:/vol/sv_vol/
qtree3 systemA:/vol/vol1/qtree3
Restore will overwrite existing data in /vol/vol1/qtree3.
Are you sure you want to continue? yes
Transfer started.
Monitor progress with 'snapvault status' or the snapmirror log.
```

For baseline restore:

```
systemA> snapvault restore -s sv_backup.0  -S systemB:/vol/sv_vol/
qtree3 systemA:/vol/vol1/qtree3
Restore will overwrite existing data in /vol/vol1/qtree3.
Are you sure you want to continue? yes
Transfer started.
Monitor progress with 'snapvault status' or the snapmirror log.
```

After the restore operation, you can restart the SnapVault relationship:

```
systemB> snapvault start -r -S systemA:/vol/vol1/qtree3
systemB:/vol/sv_vol/qtree3
The resync base snapshot will be: sv_backup.0
Resync may alter the data in this qtree.
Are you sure you want to resync the qtree? yes
Transfer started.
Monitor progress with 'snapvault status' or the snapmirror log.
```

Example 2

This example shows the primary storage system (systemA) requesting data to be restored from the secondary storage system (systemB) and then the secondary storage system canceling the SnapVault backup relationship on both storage systems to release the resources used.

systemA> snapvault restore -S systemB:/vol/sv_vol/qtree3 /vol/vol1/qtree3

systemB> snapvault release /vol/sv_vol/qtree3 systemA:/vol/vol1/qtree3

Deleting the residual Snapshot copy

When you use the `snapvault restore` command to restore a primary qtree, SnapVault places a residual SnapVault Snapshot copy on the volume of the restored primary qtree. This Snapshot copy is not automatically deleted. However, you can delete this Snapshot copy manually.

About this task

If you have configured this volume to retain the maximum 251 Snapshot copies allowed by Data ONTAP, you must manually delete this residual Snapshot copy, or else no new Snapshot copies can be created.

Steps

1. To display all Snapshot copies (including the residual Snapshot copy) on the volume of the restored qtree, enter the following command:

   snap list *primaryvolume*

   The residual Snapshot copy is distinguished by the following syntax:

primaryhost* (*nvram_id*)*_primaryvolume_restoredqtree*-dst.*x

Example

prim_system (1880911275)_vol1_mytree-dst.2

2. To delete the residual Snapshot copy, enter the following command:

   snap delete *primaryvolume extrasnapshotname*

   Example

   snap delete vol1 prim_system (1880911275)_vol1_mytreedst.2

How to abort SnapVault transfers

You can use the `snapvault abort` command to halt an ongoing SnapVault transfer if a later transfer is more useful or if an immediate shutdown or restart is necessary.

This command can halt ongoing SnapVault transfers from primary to secondary storage system (invoked by the `snapvault start` or `snapvault update` commands or scheduled through the `snapvault snap sched -x` command), or from secondary back to primary storage system (invoked by the `snapvault restore` command).

You can enter the `snapvault abort` command at the primary storage system or at the secondary storage system.

Aborting primary-to-secondary storage transfers

The `snapvault abort` command can halt ongoing SnapVault transfers from the primary to the secondary storage system that were invoked by the `snapvault start` or `snapvault update` commands, or that were scheduled through the `snapvault snap sched -x` command.

Step

1. To abort a primary to secondary storage system transfer, at the console of either the primary or secondary storage system, enter the following command:

   snapvault abort [-f] [-h] [*sec_system*:]/vol/*volx/sec_qtree*

   The `-f` option forces the `abort` command to proceed without first asking confirmation from the user.

   The `-h` option causes a hard abort. It is only effective on the SnapVault secondary storage system. This option is ignored if specified on the primary storage system.

   Note: If you use the `-h` (hard abort) option with the `snapvault abort` command, you cannot restart the transfer.

sec_system is the name of the SnapVault secondary storage system.

sec_qtree is the name of the secondary qtree to which the data is being transferred through SnapVault `start` or `update` commands.

Aborting secondary-to-primary storage transfers

The `snapvault abort` command can halt ongoing SnapVault transfers from secondary to primary storage that were invoked by the `snapvault restore` command.

Step

1. To abort a secondary to primary storage transfer, at the console of either the primary or secondary storage system, enter the following command:

   snapvault abort [*prim_system*:]/vol/*volx*/*prim_qtree*

   prim_system is the name of the primary storage system.

   prim_qtree is the name of the primary qtree to which the data is being restored.

   Note: If you use the `-h` option (hard abort) with the `snapvault abort` command, you cannot restart the transfer.

Aborting SnapVault Snapshot copy creation

You can abort the ongoing creation of a SnapVault Snapshot copy on the secondary system.

About this task

You can obtain the secondary system volume and Snapshot copy basenames from the `snapvault status -s` output.

Step

1. At the console of the secondary system, enter the following command:

   snapvault abort -s *volx sv_snapname*

   The `-s` option aborts the attempt to create a Snapshot copy with basename of *sv_snapname* on volume *volx*.

Ending SnapVault backups for a qtree

You can use the `snapvault stop` command to end the SnapVault backup process for a qtree when you no longer need the data in the primary system qtree to be protected.

About this task

After you use the `snapvault stop` command, SnapVault stops updating the qtree on the secondary system and deletes the qtree. Existing Snapshot copies on the secondary system are unaffected, but as new Snapshot copies replace the old ones, the data from the qtree whose backup was stopped disappears.

Step

1. From the secondary system, enter the following command:

   **`snapvault stop [`*`sec_system`*`:]`*`sec_qtree_path`*

   [*`sec_system`*:]*`sec_qtree_path`* is the qtree that you no longer want to back up.

   ### Example

   ```
   systemB> snapvault stop systemB:/vol/sv_vol/qtree3
   ```

   Note: After you end the backup process from a SnapVault secondary system, you might want to release the obsolete Snapshot copies on the primary system.

Related tasks

Releasing SnapVault relationships on page 282

Releasing SnapVault relationships

There are two methods of releasing a SnapVault relationship between a primary qtree and its secondary qtree backup (originally defined through the `snapvault start` command) after the relationship is no longer needed.

About this task

You can release SnapVault relationships in the following scenarios:

- On a primary storage system, as a part of shutting down a SnapVault relationship after a `snapvault stop` command was completed on the secondary storage system.

- On the secondary storage system, after data is restored to a primary storage system and you do not want to reactivate the backup relationship between the primary and secondary qtrees.

Steps

1. On a primary storage system console, enter the following command:

   snapvault release *prim_qtree_path sec_system*:*sec_qtree_path*

2. On the secondary storage system console, enter the following command:

   snapvault release *sec_qtree_path prim_system*:*prim_qtree_path*

   Example

   ```
   systemB> snapvault release /vol/sv_vol/qtree3 systemA:/vol/vol1/qtree3
   ```

Turning SnapVault off

You can turn SnapVault off by using the `snapvault.enable` option if the files on the primary or secondary storage system are no longer important or current or have been moved to another location.

Step

1. To turn off SnapVault on a primary storage system or secondary storage system, enter the following command:

   options snapvault.enable off

   This option persists across reboots.

Compression feature of Open Systems SnapVault

The compression feature of Open Systems SnapVault enables data compression over the network. This feature helps optimize bandwidth usage for Open Systems SnapVault data transfers.

Data ONTAP 7.3 and later support bandwidth optimization for Open Systems SnapVault through the compression feature. However, SnapVault primary systems do not support bandwidth optimization.

When you want to compress the network data, you can use global or local compression options.

Enabling the compression feature globally for Open Systems SnapVault relationships

You can enable the compression feature globally, to optimize bandwidth usage for all Open Systems SnapVault relationships.

Step

1. To enable the compression feature globally, enter the following command on the secondary system:

   `options snapvault.ossv.compression on`

   `options snapvault.ossv.compression on` enables compression for the Open Systems SnapVault relationships for which the compression feature was not specified locally.

   Note: The compression feature is enabled only for those relationships for which the compression option is not individually specified.

Enabling the compression feature for a new Open Systems SnapVault relationship

You can enable the compression feature for bandwidth optimization for a new Open Systems SnapVault relationship.

Step

1. On the secondary storage system, when setting up an Open Systems SnapVault relationship, enter the following command to specify the compression option:

   `snapvault start -o compression=on -S` ***`prim_host`*****`:`*****`dirpath`**
   `sec_host`**`:`*****`sec_qtree_path`**

   `-o compression=on` enables the compression feature for bandwidth optimization.

   `-S` specifies the primary storage system and path. It must be given the first time to configure the qtree. It is optional when restarting an initial transfer for a previously configured qtree.

   ### Example

   `snapvault start -o compression=on -S systemA:C:\dir3 systemB:/vol/vol1/qtree3`

Enabling the compression feature for an existing Open Systems SnapVault relationship

You can enable the compression feature for bandwidth optimization for an existing Open Systems SnapVault relationship.

Step

1. To enable the compression feature for an existing Open Systems SnapVault relationship, enter the following command:

   snapvault modify -o compression=on [-S [*prim_host*:]*dirpath*] [*sec_system*:]*sec_qtree_path*

   `-S` specifies the primary storage system and path. It must be given the first time to configure the qtree. It is optional when restarting an initial transfer for a previously configured qtree.

   `compression=on` enables the compression feature for bandwidth optimization.

   ### Example

   snapvault modify -o compression=on /vol/vol1/qtree3

   Note: A `snapvault modify` command becomes effective only with the next transfer. Ongoing SnapVault transfers are not affected.

Disabling the compression feature globally for Open Systems SnapVault relationships

You can disable the compression feature for bandwidth optimization globally for all Open Systems SnapVault relationships.

Step

1. To disable the compression feature *globally*, enter the following command on the secondary storage system:

   options snapvault.ossv.compression off

   `options snapvault.ossv.compression off` disables compression for those Open Systems SnapVault relationships for which the compression feature is not specified locally.

   Note: The compression feature will be disabled only for those relationships for which per relationship compression option is not enabled or disabled.

Disabling the compression feature for a new Open Systems SnapVault relationship

You can disable the compression option locally for a new Open Systems SnapVault relationship.

Step

1. To disable the compression option locally for each SnapVault relationship, enter the following command on the secondary storage system:

   snapvault start -o compression=off -S *prim_host*:*dirpath* *sec_host*:*sec_qtree_path*

   `-S` specifies the primary storage system and path. It must be given the first time to configure the qtree. It is optional when restarting an initial transfer for a previously configured qtree.

   `-o compression=off` disables the compression feature.

Example

snapvault start -o compression=off -S systemA:C:\dir3 systemB:/vol/vol1/qtree3

Disabling the compression feature for an existing Open Systems SnapVault relationship

You can disable the compression feature for an existing Open Systems SnapVault relationship.

Step

1. To disable the compression feature for an existing Open Systems SnapVault relationship, enter the following command on the secondary storage system:

   snapvault modify -o compression=off [-S [*prim_system*:]*prim_qtree_path*] [*sec_system*:]*sec_qtree_path*

   `-S` specifies the primary storage system and path.

   `compression=off` disables the compression feature.

Disabling the compression feature for an existing Open Systems SnapVault relationship

snapvault modify -o compression=off /vol/vol1/qtree3

Note: To use the default value of the global compression option for an existing SnapVault relationship, enter the following command:

snapvault modify -o compression=default *sec_qtree_path*

A `snapvault modify` command becomes effective only with the next transfer. Ongoing SnapVault transfers are not affected.

Setting the default value for compression feature

You can set the default value for the compression feature for an existing Open Systems SnapVault relationship that has compression enabled or disabled locally.

Step

1. To set the default value, enter the following command:

   `snapvault modify -o compression=default` ***`sec_qtree_path`***

   - If the compression feature is enabled globally, then compression is enabled for this Open Systems SnapVault relationship.
   - If the compression feature is disabled globally, then compression is disabled for this Open Systems SnapVault relationship.

Viewing the compression status for Open Systems SnapVault relationships

You can view the compression status for Open Systems SnapVault relationships.

Step

1. To view the compression status for each Open Systems SnapVault relationship, enter the following command:

   `snapvault status -c`

   Either `compression=on` or `compression=off` is displayed for each relationship, if it is configured for that relationship.

   Note: To view the compression ratio for each Open Systems SnapVault relationship, run the `snapvault status -l` command.

SnapVault secondary system protection

By setting up a SnapMirror relationship between the SnapVault secondary storage system and a SnapMirror destination storage system, NearStore system, or tape backup unit, you can provide backup and standby service or backup and restore protection for the SnapVault secondary storage system data.

- SnapMirror backup and standby service for SnapVault uses the SnapMirror destination device as a standby device to be activated as an alternate SnapVault secondary storage system if the original secondary storage system goes down.

- SnapMirror backup and restore protection for SnapVault uses the SnapMirror destination device as a source from which you can restore backup data to a SnapVault secondary storage system that has suffered data loss or corruption.

How to use SnapMirror to replicate SnapVault data

The SnapVault secondary storage system carries out SnapVault operations on its sources as usual. Then on a scheduled per-volume basis, the system replicates the SnapVault data to its SnapMirror destination partner or tape backup unit.

In this configuration, SnapVault does not delete any Snapshot version on the primary storage systems until that version has been successfully replicated from the SnapVault secondary storage unit to its SnapMirror destination. This guarantees that a Snapshot version will always be retrievable even if the SnapVault secondary storage system is disabled.

Related concepts

Protection of SnapVault secondaries using volume SnapMirror on page 202

Using backup and standby service for SnapVault

You can set up SnapMirror to protect a SnapVault secondary system.

Steps

1. Use the `license` command to confirm that the SnapVault secondary storage device has both SnapVault secondary storage and SnapMirror features licensed.
2. Use the `license` command to confirm that the SnapMirror destination device has both the SnapVault secondary storage and SnapMirror features licensed.
3. Set up SnapMirror replication from the active SnapVault secondary system to a disk-based destination device (another system or NearStore system).
4. If the active SnapVault secondary system is damaged or destroyed, convert the SnapMirror destination device to an alternate SnapVault secondary system to carry on the task of backing up data from the primary system.

   #### Example

   You have the following configuration: `systemA` (primary) has a SnapVault backup to `systemB` (secondary) and `systemB` has a SnapMirror backup to `systemC` (tertiary). If `systemB` fails, break the SnapMirror backup between `systemB` and `systemC`, and re-create the SnapVault relationship such that `systemA` has a SnapVault backup to `systemC`.
5. After evaluating the reason for the data loss or corruption, either return the secondary system to the path as a tertiary system or add a new system as tertiary storage.

After adding `systemB`, you have the following configuration: `systemA` (primary) has a SnapVault backup to `systemC` (secondary) and `systemC` has a SnapMirror backup to `systemB` (tertiary). If you added a new system, it would replace `systemB` in the configuration.

6. As an optional step, you might want to return the systems to their original configuration.

Related tasks

Setting up a basic SnapMirror operation on page 104

Re-creating the SnapVault relationship

You can re-create the SnapVault relationship from the primary system to the tertiary system.

About this task

The original configuration in the following procedure is a SnapVault relationship between `systemA:/vol/vol1/qtree3` and `systemB:/vol/vol2/qtree3` with the SnapVault destination volume `systemB:vol2` backed up to another volume on a third system, `systemC:vol3`.

Steps

1. Break the volume SnapMirror relationship to the volume on the new secondary system (`systemC`) to make it writable.

   #### Example

   Perform the following steps from `systemC`.

   ```
   snapmirror quiesce vol3 snapmirror break vol3
   ```

2. Check the status of the SnapMirror relationship on the new secondary system (`systemC`), by using the `snapmirror status` command.

   The SnapMirror relationship should be `broken-off`.

3. Check the status of the SnapVault relationship on the new secondary system (`systemC`).

   #### Example

   ```
   snapvault status
   ```

   The SnapVault relationship should be `snapvaulted`.

4. Add SnapVault configuration information to the new secondary system using the `snapvault start` command.

   Note: This does not start a new baseline, it updates the registry.

Example

Perform the following step from `systemC` (the new secondary system).

```
snapvault start -S systemA:/vol/vol1/qtree3 systemC:/vol/vol3/qtree3
```

5. Check that the new SnapVault configuration is present.

Example

Perform the following step from `systemC`.

```
snapvault status -c
```

6. Test the new SnapVault relationship by manually updating `systemC`.

Example

Perform the following step from `systemC`.

```
snapvault update systemC:/vol/vol3/qtree3
```

7. Re-create any schedules used on the old secondary system on the new secondary system, and ensure access permissions are in place.
8. Release resources locked on `systemA` for the removed `systemA` to `systemB` SnapVault relationship.

Example

Perform the following step from `systemA`.

```
snapvault release /vol/vol1/qtree3 systemB:/vol/vol2/qtree3
```

Adding back the tertiary system for SnapMirror backup

Depending on the kind of backup interrupt you encountered, the old secondary system might be usable again as the tertiary system.

About this task

After adding the tertiary system, `systemA` has a SnapVault backup to `systemC`, and `systemC` has a SnapMirror backup to `system B` (or `systemD` if `systemB` cannot be reused).

Steps

1. Resynchronize the SnapMirror relationship between the new secondary system, `systemC`, and the new tertiary system, `systemB`.

Example

```
snapmirror resync -S systemC:vol3 systemB:vol2
```

2. Release resources that are locked for the old secondary system to tertiary system SnapMirror relationship.

Example

Perform the following step from `systemB`.

```
snapmirror release vol2 systemC:vol3
```

Related concepts

How the snapmirror resync command helps minimize data loss on page 190

Related tasks

Setting up a basic SnapMirror operation on page 104

Returning systems to the original configuration

In certain scenarios, you can return the systems to the original configuration after using backup and standby.

About this task

This is an optional procedure because you might not need to return the backup configuration to the original `systemA` to `systemB` to `systemC` configuration.

Steps

1. Ensure that the secondary system and tertiary system have the same data by temporarily blocking SnapVault updates to the secondary system.

Example

Perform the following step from the secondary system, `systemC`.

```
snapvault snap unsched vol3
```

2. Allow updates to the secondary system to be propagated to the tertiary system.

Example

Perform the following step from the tertiary system, `systemB`.

```
snapmirror update systemB:vol2
```

3. Reverse the roles of the secondary and tertiary systems, so that what was the secondary becomes the tertiary system and what was the tertiary becomes the secondary system.

Example

Perform the following step from the new tertiary system, `systemC`:

```
snapmirror resync systemC:vol3
```

4. Release resources locked on `systemC` for the removed `systemC` to `systemB` SnapMirror relationship.

   Example

   Perform the following step from the new tertiary system, `systemC`.

   ```
   snapmirror release vol3 systemB:vol2
   ```

5. Break the SnapMirror relationship on `systemB` to allow further SnapVault updates from the original SnapVault relationship.

   Example

   Perform the following step from `systemB`.

   ```
   snapmirror break vol2
   ```

6. Re-create the original SnapVault relationship from `systemA` to `systemB`.

   Example

   Perform the following step from `systemB` (the original secondary system).

   ```
   snapvault update /vol/vol2/qtree3
   ```

7. Re-create the original SnapMirror relationship from `systemB` to `systemC`.

   Example

   Perform the following step from `systemC` (the original tertiary system).

   ```
   snapmirror update vol3
   ```

8. Remove the SnapVault configuration from `systemC`.

   Example

   Perform the following step from `systemC`.

   ```
   snapvault stop [-f] /vol/vol3/qtree3
   ```

   This command is available on the secondary system only. The command unconfigures the qtree to ensure that are no more updates of the qtree and then deletes the qtree from the active file system.

   The `-f` option forces the `snapvault stop` command to proceed without first asking for confirmation.

How to use SnapVault to protect a volume SnapMirror destination

You can use SnapVault to protect a volume SnapMirror destination. You can perform SnapVault transfers from the volume SnapMirror destination when you want to retain the data for a longer period.

In this deployment scenario, data from various primary systems is replicated to a remote site for disaster recovery. Volume SnapMirror ensures identical data at the source and destination systems. If you want to retain the data for a longer duration (that is, 90 days and more) at the disaster recovery site, you can use SnapVault to back up data from the SnapMirror destination.

When using SnapVault to back up a SnapMirror destination volume or qtree, you need to ensure the following:

- The SnapMirror license is installed on the SnapMirror primary system and the SnapMirror secondary system (which also acts as the SnapVault primary system). The SnapVault license is installed on the SnapVault primary system (which also acts as the SnapMirror secondary system) and the SnapVault secondary system.
- The SnapVault operation occurs between SnapMirror scheduled updates. SnapMirror updates fail if a SnapVault operation is initiated or in progress.
- Before performing a SnapVault transfer for a particular Snapshot copy, this Snapshot copy must be preserved on the primary system.

To use the SnapVault backup schedule, you need to configure the SnapVault primary schedule at the volume SnapMirror primary system. Therefore, you need the SnapVault license installed on the volume SnapMirror primary system. However, you do not need the SnapVault license on the volume SnapMirror primary system if you want to use the Snapshot copies created by SnapMirror.

For more information about disaster recovery and long-term backup of data, see the technical report *SnapMirror Async Overview and Best Practices Guide*.

Related tasks

Planning SnapVault backup schedule and Snapshot copy retention on page 238

Related information

SnapMirror Async Overview and Best Practices Guide: media.netapp.com/documents/tr-3446.pdf

Preserving a Snapshot copy

The `snapvault snap preserve` command enables you to preserve the required Snapshot copy. This command prevents Data ONTAP features (such as `snap autodelete`) from deleting the Snapshot copy.

About this task

You might want to preserve the Snapshot copies in the following scenarios:

- You do not want a Snapshot copy created by SnapVault to be recycled.
- You want to preserve some Snapshot copies from getting deleted by the `snap autodelete` command.

  Note: The Snapshot copies locked by data replication features like SnapMirror and SnapVault are not deleted when the `snap autodelete` command uses the `commitment try` option.

- You want to preserve an application-consistent Snapshot copy.

Steps

1. To preserve a Snapshot copy at the volume SnapMirror primary system, enter the following command:

   snapvault snap preserve *vol_name snapshot_name* [*tag_name*]

   tag_name is the name of the preserve operation. It uniquely identifies this preserve operation. When the tag name is not specified, a tag name is added automatically.

   snapshot_name is the name of the Snapshot copy.

   Note: This command does not need a SnapVault license.

   Example

   snapvault snap preserve vol1 snap1 tag1

2. To list all preserved Snapshot copies, enter the following command:

   snapvault snap preservations [*vol_name*] [*snap_name*]

   If *snap_name* is not specified, then all the preserved Snapshot copies are listed.

   If *snap_name* is specified, then all preservations on the specified Snapshot copy are displayed.

   Example

   snapvault snap preservations vol1 snap1

Unpreserving a Snapshot copy

If you do not want to retain the Snapshot copy that you have preserved, you need to unpreserve the Snapshot copies.

Before you begin

You can list all preserved Snapshot copies by using the `snapvault snap preserve` command.

Step

1. To unpreserve a Snapshot copy, run the following command:

   `snapvault snap unpreserve` *`vol_name snapshot_name`* `[`*`tag_name`*`|-all]`

   If *`tag_name`* is specified, then the Snapshot copy that is preserved with the specified tag name is removed.

   If *`tag_name`* is not specified, then Snapshot copies preserved without a tag name are removed.

   If the `-all` option is specified, all preservations on a specified Snapshot copy are removed.

   #### Example

   ```
   snapvault snap unpreserve vol1 snap1 -all
   ```

SnapVault behavior when used for volume SnapMirror destination protection

When SnapVault is used to protect the volume SnapMirror destination, SnapVault uses the Snapshot copy based on the Data ONTAP version that is running on the SnapMirror destination system, when SnapVault schedule is used within Data ONTAP.

SnapVault behavior varies according to the Data ONTAP version running on the SnapMirror destination system.

Data ONTAP 7.3.2 and earlier: SnapVault ignores any specified (named) Snapshot copy and uses the most recent volume SnapMirror Snapshot copy for backup.

Data ONTAP 7.3.2 to 7.3.4 and Data ONTAP 8.0 7-Mode: SnapVault updates only from a specific named Snapshot copy and it does not use the Snapshot copy created by volume SnapMirror.

Data ONTAP 7.3.5 and later: You can choose the SnapVault behavior for updating the destination system. SnapVault can use either the Snapshot copy created by volume SnapMirror or a named Snapshot copy.

Setting the options to back up a volume SnapMirror destination using SnapVault

Starting with Data ONTAP 7.3.5, if you are backing up a volume SnapMirror destination using SnapVault, the SnapVault schedule updates can start from a named Snapshot copy, the Snapshot copy created by volume SnapMirror, or you can set a preference.

Step

1. From the SnapMirror destination system, enter any of the following commands:

If you want to…	Then, enter the following command…
Schedule the SnapVault transfer updates from a specific Snapshot copy	`options snapvault.snapshot_for_dr_backup named_snapshot_only`
Schedule the SnapVault transfer updates from Snapshot copy created by volume SnapMirror if the named Snapshot copy is not available	`options snapvault.snapshot_for_dr_backup named_snapshot_preferred`
Schedule the SnapVault transfer updates from the most recent Snapshot copy created by volume SnapMirror **Note:** This is the default option.	`options snapvault.snapshot_for_dr_backup vsm_base_only`

SnapVault and MultiStore

If you are using MultiStore software, you can use SnapVault to replicate the data for a vFiler unit.

The management of SnapVault secondary volume (creation or modification of SnapVault relationships and schedules at the SnapVault secondary) is supported from the default vFiler unit (vfiler0) and the nondefault vFiler unit.

The default vFiler unit manages the secondary volume of the default vFiler units and the nondefault vFiler units. The nondefault vFiler unit manages the secondary volume only of the nondefault vFiler units.

You can create relationships in the default vFiler unit and the nondefault vFiler unit. However, you cannot create the same relationship or schedule the same target in more than one vFiler units. When

you create a relationship in the default vFiler unit, a configuration entry is created in the registry of the default vFiler unit. You must either delete the relationship or run the `snapvault start –r –S srcfiler:`*`source_qtree_path`* *`destination_qtree_path`* command to move the relationship from the default vFiler unit to the nondefault vFiler unit. Also, at any instance, only one target can be active in a volume.

Note: It is best to create and manage all the relationships of the volume in either the default vFiler unit or the nondefault vFiler unit.

The following table shows vFiler unit support with SnapVault secondary volumes for different combinations:

Management of SnapVault secondary volume	**Ownership of SnapVault secondary volume**	
	Default vFiler unit (vfiler0)	Nondefault vFiler unit
Default vFiler unit (vfiler0)	Yes	Yes
Nondefault vFiler unit	No	Yes

By default, the default vFiler unit manages the secondary volume of the nondefault vFiler units.

The management of SnapVault primary in a vfiler context is supported.

The following table shows vFiler unit support with SnapVault primary volumes for different combinations:

Management of SnapVault primary volume	**Ownership of SnapVault primary volume**	
	Default vFiler unit (vFiler0)	Nondefault vFiler unit
Default vFiler unit (vFiler0)	Yes	Yes
Nondefault vFiler unit	No	Yes

For more information about vFiler units, see the *Data ONTAP MultiStore Management Guide for 7-Mode*.

DataFabric Manager support for the management of SnapVault relationships

DataFabric Manager supports the management of SnapVault relationships for volumes through the default vFiler (vFiler0) context only. When using DataFabric Manager, the following limitations apply for SnapVault relationships that involve nondefault vFiler units.

- You can only view SnapVault relationships configured through the default vFiler unit (vfiler0). You cannot view any SnapVault relationships configured through nondefault vFiler units.

- You can configure new SnapVault relationships for a volume only through the default vFiler unit (vfiler0), even if the volume belongs to a nondefault vFiler unit.

Moving SnapVault configurations across vFiler units

When you move a volume to a new vFiler unit, it also moves the SnapVault configurations and relationships to the new vFiler unit.

Before you begin

You must disable SnapVault on the vFiler unit before executing the `vfiler move` command to ensure that there are no transfers running on the vFiler unit.

About this task

If a relationship is managed by the default vFiler unit, then the SnapVault configurations for that particular relationship are not moved to the destination vFiler unit.

Step

1. Enter the following command on the vFiler unit source:

   vfiler move *source_vfiler destination_vfiler* [-f] [-i *ip_address*] [-i *ip_address*]...] [path [path...]]

   The following SnapVault configurations can move from one vFiler unit to another vFiler unit:

   - Qtree configuration
   - Schedules
   - Softlocks (both acs and qtree)
   - Language value for Open Systems SnapVault
   - Status

Moving a relationship between vFiler units

You can move a relationship between the default vFiler unit and the nondefault vFiler unit only if the volume, which contains the relationship, is part of the nondefault vFiler unit. You cannot move the SnapVault schedules using the `snapvault start –r –S srcfiler:`*`source_qtree_path destination_qtree_path`* command.

Step

1. Enter the following command on a vFiler unit:

   snapvault start –r –S srcfiler:*source_qtree_path destination_qtree_path*

Checking SnapVault transfers in vFiler units

The SnapVault status displays all the SnapVault relationships of a volume in a vFiler unit. You can use the `snapvault status` command to view the relationship established in a particular vFiler unit.

About this task

In the default vFiler unit, the `snapvault status` command displays all the relationships configured in the default vFiler unit and only the active transfers of relationships configured in the nondefault vFiler unit. In the nondefault vFiler unit, the `snapvault status` command displays all the relationships configured on all the volumes that belong to the nondefault vFiler unit.

Step

1. Enter the following command to check the status of active transfers:

   `vfiler run * snapvault status`

Error regarding language setting changes on volumes

A warning message is displayed on the Data ONTAP CLI when the language setting of the volume changes on volumes containing SnapVault or qtree SnapMirror destinations.

The following EMS error message appears if you update the SnapVault or qtree SnapMirror relationship after changing the language setting of the volume in the secondary system on which qtree replicas are configured.

```
Volume language was changed before this transfer or it was changed
before some previous update transfer for the relationship of qtree %s.
This might create some problems with further replication transfers of
the relationship.
```

To correct this problem, you need to create a new baseline for the transfer.

Data replication using volume copy

You can use the `vol copy` set of commands to copy all data from one volume to another, either on the same system or on a different system.

For using the `vol copy` command, the source and destination volumes must be of the same type: traditional volumes or FlexVol volumes.

You cannot use the `vol copy` command to copy data between a 7-Mode volume and a Cluster-Mode volume.

The `vol` family of commands manages volumes. A volume is a logical unit of storage, containing a file system image and associated administrative options such as Snapshot copy schedules. The disk space that a volume occupies (in addition to the characteristics of the RAID protection it receives) is provided by an aggregate (see the na_aggr(1) man page).

You can initiate a volume copy with the `vol copy start` command, which enables you to copy data from one volume to another volume, either on the same or on a different storage system. The result is a restricted volume containing the same data as the source volume at the time you initiated the copy operation.

Benefits of using volume copy

Although you can copy data from a system using client programs such as cpio or use the Data ONTAP `dump` and `restore` commands, the `vol copy` command set offers several benefits.

- When a `vol copy` command reads and writes data, Data ONTAP does not traverse directories on the system. Data is copied block for block directly from the disks, which means that Data ONTAP can finish the copying faster than it could with other methods.
- Using a `vol copy` command, Data ONTAP preserves the Snapshot copy data of the source volume. If, in the future, users might need to use Snapshot copies that were taken before data was copied from one volume to another, you can use a `vol copy` command for migrating data. For example, if users accidentally delete files and need to recover them, they can do so from the preserved data.
- You do not need any additional licenses to use the `vol copy` command.

Maximum number of concurrent volume copy operations: Volume copy has the same limit on concurrent replication operations as volume SnapMirror.

Related references

Maximum number of concurrent replication operations on page 123

When to copy volumes

You might find copying volumes useful under certain situations.

The following table describes some situations where you might find copying volumes useful.

Situation	Reasons for copying one volume to another
You want to migrate data from one storage system to another.	The destination storage system has more storage or is a model that supports newer technology.
You want to move a volume from one set of disks to another on the same storage system.	You want to • Split a volume • Expand storage **Examples**: You can copy a volume vol0 to another volume vol1 and then delete duplicate files and directories in these volumes so that the original contents of vol0 are split into two volumes. You have six 1 TB disks for vol0 and four 2 TB spare disks. You can migrate vol0 to the four 2 TB disks and replace all the 1 TB disks with larger capacity disks.
You want to copy data from one storage system to another regularly to ensure high data availability.	After you copy the data, clients can switch to the destination storage system in the following scenarios: • When you shut down the storage system for software or hardware upgrades, or when the storage system is not available for reasons such as natural disasters, you can put the destination volume online to continue file service. • If a network client process accidentally deletes a large number of files on the storage system, clients can continue to have access to the files on the destination storage system while you are restoring the files to the source system. **Note:** This scenario is also a good application for SnapMirror.

IPv6 support with volume copy

When copying volumes using the `vol copy` command, you can use IPv6 addresses to specify source and destination systems. However, there are some differences between the specification of IPv6 and IPv4 addresses.

Note: Before using IPv6 functionality for a system, ensure that the `ip.v6.enable` option is set to `on`.

The usage is shown in the following example.

Use of IPv6 functionality with the vol copy start command

**`vol copy start [-p {inet | inet6}] [`*`src_system`*`:]`*`src_vol`*
`[`*`dst_system`*`:]`*`dst_vol`*

- The `-p` option enables to specify an IPv4 or IPv6 connection mode, by using one of the following two options.
  - `inet` specifies the use of an IPv4 connection.
  - `inet6` specifies the use of an IPv6 connection.
- *`src_system`* is the name of the source system, and *`src_vol`* is the name of the source volume.
- *`dst_system`* is the name of the destination system, and *`dst_vol`* is the name of the destination volume.

If the IP addresses of the source or destination systems are specified, the `-p` option is not required. For example, if IPv6 addresses are specified, the connection mode is IPv6.

If host names are used instead of IP addresses without using the `-p` option, first an IPv6 connection is attempted. If an IPv6 connection is not established, an IPv4 connection is attempted.

If host names are used instead of IP addresses with the `-p` option, the connection is attempted as specified by the `-p` option.

Prerequisites before copying a volume

You should make sure that systems involved in a `vol copy` operation meet certain requirements.

Requirements

- The source and destination volumes must be of the same type: either both traditional or both FlexVol volumes.
- The capacity of the destination volume must be greater than or equal to the capacity of the source volume.
- The source and destination storage systems must have a trust relationship with each other.
- The destination volume must exist, and must not be the root volume.
- The source volume must be online and the destination volume must be restricted.
- Remote Shell access must be enabled.
- The destination volume must not contain data that you want to preserve.

The rest of this section provides more detailed information about verifying whether the source and destination volumes meet these requirements:

You should take care not to overwrite data that you need: If the destination volume is not a new volume, ensure that it does not contain data that you might need in the future. After Data ONTAP starts copying the source volume, it overwrites the entire destination volume. All data in the active file system and in the Snapshot copies of the destination volume is lost after Data ONTAP starts copying the data.

Where volume copies can reside: The source and destination volumes of the copy can reside on the same or on different storage systems.

Recommendation for copying a volume: When a system copies data between two volumes on separate systems, it floods the network between the two systems with packets. Users of the systems involved in a volume copy operation might notice a degradation in response time during the copy. A private network for copying between the source and destination systems helps circumvent network-related performance problems when copying to different systems.

Maximum number of simultaneous volume copies: Volume copy has the same limit of simultaneous copies that SnapMirror replications have.

Verifying the size of each volume

To see whether the data in one volume can be copied or replicated to another volume, you need to compare the file system size of the two volumes.

Steps

1. On the source storage system, enter the following command:

   vol status -b *volume_name*

   volume_name is the name of the source volume.

Result: Data ONTAP displays the block size of the volume (in bytes), the RAID volume size, and the Write Anywhere File Layout (WAFL) file system size. If no volume name is given, information for all volumes is displayed.

2. On the destination storage system, repeat Step 1, replacing *volume_name* with the name of the destination volume.

3. Compare the file system (FS) numbers. If the file system size of the destination is the same as or larger than the file system size of the source, you can use the `vol copy` command (or SnapMirror) to transfer data from the source to the destination.

Example

```
vol status -b
Volume       Block Size (bytes) Vol Size (blocks) FS Size (blocks)
------       ------------------ ----------------- ---------------
sourcevol  4096                  4346752             4346752
destvol    4096               4346752             4346752
```

Verifying the relationship between systems

You can verify the relationship between systems. If the source and destination volumes in a `vol copy` operation reside on two different systems, these systems must have a trust relationship with each other.

Steps

1. By mounting the system with NFS, enter the destination system host name in the `/etc/hosts.equiv` file of the source system, if it is not present already.

   The `/etc/hosts.equiv` file contains a list of host names, each of which is on a separate line. The presence of a host name in this file indicates that the system allows that host to perform remote operations.

2. Repeat Step 1 on the destination system, entering the source system host name in the `/etc/hosts.equiv` file, if it is not present already.

Verifying and changing the status of source and destination volumes

You can verify whether the source volume is online, and that the destination volume exists and is restricted. You can also change the status of a volume when necessary.

Before you begin

You must have ensured that the destination volume for a `vol copy` operation must not be the root volume. This is because the destination volume must be offline when Data ONTAP executes the `vol copy` command, and a root volume must always be online.

Steps

1. To verify that the destination volume exists and is restricted, enter the following command on the destination storage system:

   vol status *dest_volume*

   dest_volume is the name of the volume whose status you want to check.

   If you do not provide a volume name, the command displays the status of all volumes in the storage system.

   If the volume does not exist, Data ONTAP returns an error. For information about how to create a volume, see the *Data ONTAP Storage Management Guide for 7-Mode*.

2. To verify that the source volume is online, repeat Step 1 on the source storage system, replacing *dest_volume* with the name of the source volume.

3. If you need to change the status of a volume because of the results of Step 1, enter the following command on the destination storage system:

   vol restrict *dest_volume*

   dest_volume is the name of the destination volume.

4. If you need to change the status of a volume because of the results of Step 2, enter the following command on the source storage system:

   vol online *source_volume*

   source_volume is the name of the source volume.

   If you need to perform Step 3 or Step 4, you might want to perform Step 1 or Step 2 again to verify the changes that you made.

Example

```
systemA> vol status
    Volume State  Status   Options
    vol0 online normal   root
    vol1 online normal   raidsize=14
    vol2 online restricted
volextra offline
```

Enabling remote access

To perform a volume copy from one volume to another volume on the same storage system, Remote Shell services must be enabled or the volume copy fails.

Step

1. To enable Remote Shell services, enter the following command:

   ```
   options rsh.enable on
   ```

Copying volumes using the vol copy command

You can use the `vol copy start` command to generate volume copy operations, which produce screen messages that show the progress of the operations.

Each `vol copy start` command generates two volume copy operations, each of which is assigned a number:

- One operation is for reading data from the source volume. Screen messages displayed by a `vol copy` command refer to this operation as the `volcopy dump` operation.
- One operation is for writing data to the destination volume. Screen messages displayed by a `vol copy` command refer to this operation as the `volcopy restore` operation.

When to use the volume copy operation number: You need the volume copy operation number if you want to stop a volume copy operation or change the volume copy operation speed. To find the `vol copy` operation number, you can use the `vol copy status` command.

Related tasks

Checking the status of a volume copy operation on page 311

Number of vol copy operations supported

Whether Data ONTAP can execute a `vol copy start` command depends on how many volume copy operations are already in progress on the storage systems specified in the `vol copy start` command.

To copy volumes locally, you can enter the following `vol copy start` commands on a storage system:

```
vol copy start vol0 vol1

vol copy start vol2 vol3
```

When these commands are in progress, if you enter additional `vol copy start` commands, they will fail, because four volume copy operations are already running on the system. Two of the operations are for reading the vol0 and vol2 volumes, and two of the operations are for writing the vol1 and vol3 volumes.

Suppose you enter the following three `vol copy start` commands on a storage system named systemA to copy volumes to another storage system named systemB:

```
vol copy start systemA:vol0 systemB:vol0
vol copy start systemA:vol1 systemB:vol1
vol copy start systemA:vol2 systemB:vol2
```

When these commands are in progress, systemA runs three volume copy operations to read the volumes, and systemB runs three volume copy operations to write the volumes.

An additional `vol copy start` command to copy *between* systemA and systemB will succeed because the command adds one more volume copy operation to each storage system.

However, if you enter an additional `vol copy start` command to copy volumes *locally* on either systemA or systemB, it will fail. This is because the additional command creates two volume copy operations, one for reading and one for writing, on the storage system that performs the local copying.

Copying Snapshot copies with the vol copy start command

The following table describes the Snapshot copies that will be copied from the source volume and the resulting Snapshot copies on the destination volume, depending on the option you use with the `vol copy start` command.

Option	Snapshot copies to copy from the source volume	Snapshot copies in the Snapshot file system of the destination volume
None	No Snapshot copies are copied. Only the Snapshot copy taken after you enter the `vol copy start` command, are copied.	A Snapshot copy named snapshot_for_volcopy.*n* is created, where *n* is a number starting at 0 and incrementing by one whole number with each vol copy operation is created.

Option	Snapshot copies to copy from the source volume	Snapshot copies in the Snapshot file system of the destination volume
`-S`	All Snapshot copies in the Snapshot file system of the source volume, and the Snapshot copy taken after you enter the `vol copy start` command, are copied.	All Snapshot copies in the source volume, and snapshot_for_volcopy.*n*, where *n* is a number starting at 0 and incrementing by one whole number with each vol copy operation, are created.
`-s` followed by the name of the Snapshot copy	The specified Snapshot copy will be copied.	The specified Snapshot copy is created.

Note: The `vol copy start -S` command does not copy any Snapshot copies that are created while the copying is in progress. For example, if the copying lasts from 11:45 p.m. to 1:00 a.m. the next day and Data ONTAP creates a Snapshot copy named nightly.0 at midnight, Data ONTAP does not copy the nightly.0 Snapshot copy.

Copying one volume to another volume using the vol copy command

You can copy one volume to another volume, by using the `vol copy` command.

Step

1. Enter the following command on either the source or destination system:

   `vol copy start [-S | -s` *`snapshot_name`*`] source_volume dest_volume`

   The `-S` and `-s` arguments specify the Snapshot copies to copy.

   `source_volume` and *`dest_volume`* are the names of the source and destination volumes. If a volume is on a different storage system, precede the volume name with the system name and a colon. For examples illustrating how to specify volume names, see "Examples of the `vol copy start` command" in the following table.

   Note: If the copying takes place between two storage systems, you can enter the `vol copy start` command on either the source or destination storage system. However, you cannot enter the command on a third storage system that does not contain the source or destination volume.

Examples of the vol copy start command

The following table shows several examples of the `vol copy start` command.

If you want to...	Use...
Copy all Snapshot copies from the vol0 volume to the vol1 volume on the same storage system	**`vol copy start -S vol0 vol1`**
Copy a nightly Snapshot copy from the vol0 volume to the vol1 volume on the same storage system	**`vol copy start -s nightly.1 vol0 vol1`**
Create a Snapshot copy in the vol0 volume to be copied to the vol1 volume on the same storage system	**`vol copy start vol0 vol1`**
Copy all Snapshot copies from the vol0 volume to the vol1 volume on a different storage system named systemA	**`vol copy start -S vol0 systemA:vol1`**

Error messages generated by vol copy start commands

If your storage system does not meet the requirements for copying a volume, the `vol copy start` command generates one or more error messages.

The following table explains the possible error messages and their meanings.

Error message	Meaning
`Permission denied. VOLCOPY: Could not connect to system systemB`	The source system does not have permission to copy to the destination storage system. **Action**: Ensure that the storage systems have a trust relationship with each other.
`VOLCOPY: volcopy restore: volume is online, aborting`	The destination volume is online. **Action**: Take the destination volume offline.
`VOLCOPY: volcopy restore: volume is too small, aborting`	The destination volume is smaller than the source volume. **Action**: Add more disk space to the destination volume or choose another destination volume of sufficient capacity.
`write: setting up STDERR broken pipe`	A local volume copy tried to start, but Remote Shell access is not enabled on the storage system. **Action**: Enable Remote Shell access on the storage system so that it can receive `rsh` commands.

Using volume copy to copy LUNs

You can use the `vol copy` command to copy LUNs; however, this requires that applications accessing the LUNs are quiesced and offline prior to the copy operation.

Before you begin

The contents of the host file system buffers must be saved to disk before running `vol copy` commands on the storage system.

Note: The term *LUNs* in this context refers to the LUNs that Data ONTAP serves to clients, not to the array LUNs used for storage on a storage array.

About this task

The `vol copy` command enables you to copy data from one WAFL volume to another, either within the same storage system or to a different storage system. The result of the `vol copy` command is a restricted volume containing the same data that was on the source storage system at the time you initiate the copy operation.

Step

1. To copy a volume containing a LUN to the same or different storage system, enter the following command:

   `vol copy start -S` *`source`*`:`*`source_volume`* *`dest`*`:`*`dest_volume`*

   `-S` copies all Snapshot copies in the source volume to the destination volume. If the source volume has Snapshot copy-backed LUNs, you must use the `-S` option to ensure that the Snapshot copies are copied to the destination volume.

   If the copying takes place between two storage systems, you can enter the `vol copy start` command on either the source or destination storage system. You cannot, however, enter the command on a third storage system that does not contain the source or destination volume.

   #### Example

   ```
   vol copy start -S systemA:vol0 systemB:vol1
   ```

Checking the status of a volume copy operation

You can use the `vol copy status` command to check the status of volume copy operations.

About this task

This command displays the status for a specified volume copy operation. If you do not specify the operation number, the command displays the status of all volume copy operations in progress. In the command output, the operations are differentiated from one another with unique volume copy operation numbers.

Restrictions: Remember the following restrictions when checking volume copy status:

- If you start a volume copy operation from the system console, you can enter the `vol copy status` command only through the `rsh` command when the copy operation is in progress. This is because you do not have access to the system prompt on the console when Data ONTAP is copying the volume.
- If data is being copied between two storage systems, you can enter the `vol copy status` command through a remote shell connection to either system. The operation numbers displayed on the source system and the destination system are different because the reading and the writing are considered two different operations.

Step

1. Enter the following command:

   vol copy status [*operation_number*]

   operation_number is the specific volume copy operation.

   Omit *operation_number* to display the status of all current volume copy operations. The operations are numbered from 0 through 3.

Sample status message from the vol copy start command

The following example shows a `vol copy start` command that copies the vol0 volume to the vol1 volume on the same storage system. When the operation is in progress, it displays the volume copy operation status.

```
systemA>vol copy start -S vol0 vol1
Copy Volume: vol0 on machine 127.0.0.1 to Volume: vol1
Reading the dump stream
VOLCOPY: Starting on volume 1.
This dump contains 257 blocks
10:04 pm : volcopy restore 1 : begun.
10:04 pm : volcopy restore 1 : 5 % done. Estimate 3 minutes
remaining.
```

```
.
.
.
10:04 pm : volcopy restore 1 : 95% done. Estimate 1 minutes
remaining.
```

The following example shows a `vol copy` status command using `rsh`.

Before the prompt is displayed again, you can use the `vol copy status` command on a trusted host of the storage system, as shown in the following example:

```
rsh systemA vol copy status
10:04 pm : volcopy dump 0 : 99 % done. Estimate 1 minutes remaining.
10:04 pm : volcopy restore 1 : 99 % done. Estimate 1 minutes
remaining.
No operation 2 in progress.
No operation 3 in progress.
```

In the previous examples, volume copy operation 0, shown as `volcopy dump 0` in the display, is for reading the data from the vol0 volume; `volume copy operation 1`, shown as `volcopy restore 1` in the display, is for writing the data to the vol1 volume.

Displaying the current speed for copying a volume

You can display the speed for copying a volume when you want to determine the current setting, and to verify the speed before changing the setting. This procedure enables you to verify the default speed for all volume copy operations.

Step

1. To display the speed for copying a volume, enter the following command:

   `options vol.copy.throttle`

   Result: The value 10 (full speed) through 1 (one-tenth full speed) to be used by all volume copy operations is displayed. The default value is 10.

Controlling a volume copy operation speed

You can control the speed of a volume copy operation before you start the volume copy operation and during a volume copy operation.

About this task

The speed for reading data from the source volume and the speed for writing data to the destination volume can be different. The slower of the two values determines the time required for Data ONTAP to finish copying the data. You can change the speed of a volume copy operation when you suspect it might cause performance problems on your storage system.

Note: Changing the `vol.copy.throttle` option changes the default speed for all volume copy operations to follow.

Step

1. To control volume copy operation speed, choose one of the actions from the following table.

If you want to control the speed of the volume copy operation...	Then...
Before starting the copy operations	Enter the following command: **`options vol.copy.throttle`** ***`value`*** *`value`* is the specific speed you want.
During the copy operation	Enter the following command through a Remote Shell: **`vol copy throttle [`*`operation_number`*`] `*`value`*** *`operation_number`* is the specific volume copy operation whose speed you want to adjust. If you do not specify an operation number, the command applies to all volume copy operations that are in progress. *`value`* is the specific speed you want.

Example

The following example illustrates changing the speed of all volume copy operations in progress to one-tenth of full speed through a Remote Shell:

```
rsh systemA vol copy throttle 1
```

```
volcopy operation 0: Throttle adjusted from 100% to 10%.
volcopy operation 1: Throttle adjusted from 100% to 10%.
```

Aborting a volume copy operation

If data is being copied between two storage systems, you can stop copying by executing the `vol copy abort` command on either storage system. If you start the volume copying operation from the system console, you can enter the `vol copy abort` command only through the `rsh` command. This is because you do not have access to the system prompt on the console during the copying.

Before you begin

To stop a specific volume copy operation, you need to specify the operation number. You can obtain the operation number from the `vol copy status` output.

Step

1. To stop a volume copy operation, enter the following command:

   `vol copy abort [all | `*`operation_number`*`]`

   `operation_number` is the specific volume copy operation to be stopped. Specify `all` to stop all operations.

   Attention: An incomplete volume copy operation leaves unusable data in the destination volume.

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.

What SyncMirror is

SyncMirror is an optional feature of Data ONTAP. It is used to mirror data to two separate aggregates. It allows for real-time mirroring of data to matching aggregates physically connected to the same storage system.

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 directly connected to the same system.

The following provides information about the activities of SyncMirror:

- SyncMirror can be used to mirror aggregates and traditional volumes. (A traditional volume is essentially an aggregate with a single volume that spans the entire aggregate.)
- SyncMirror cannot be used to mirror FlexVol volumes. However, FlexVol volumes can be mirrored as part of an aggregate.
- SyncMirror is different from synchronous SnapMirror.

For more information about aggregates and volumes, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Related information

Data ONTAP Information Library page: support.netapp.com//documentation/productsatoz/index.html

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 third-party storage, 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) storage system.

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 need 2,880 GB of disk space to mirror a 1,440-GB aggregate—1,440 GB for each plex of the mirrored aggregate.

What mirrored aggregates are

A mirrored aggregate is a single WAFL storage file system with two physically separated and synchronously up-to-date copies on disks or array LUNs. These copies are called plexes. Data ONTAP typically names the first plex `plex0` and the second plex `plex1`.

Each plex is a physical copy of the same WAFL file system, and consists of one or more RAID groups. As SyncMirror duplicates complete WAFL file systems, you cannot use the SyncMirror feature with a FlexVol volume—only aggregates (including all contained FlexVol volumes) are supported.

Related references

Considerations for using mirrored aggregates on page 326

Requirements for using SyncMirror with disks

For using SyncMirror to mirror aggregates, you need systems that support the SyncMirror feature and an appropriate configuration of disk shelves.

The following are prerequisites for using SyncMirror:

- The systems, or the HA pair, should support the SyncMirror feature.
- You must connect disk shelves to the storage system in a configuration that supports mirrored aggregates.

For more information about configurations that support mirrored aggregates, see the *Data ONTAP High Availability and MetroCluster Configuration Guide for 7-Mode*.

For information about the disk shelves and other hardware supported with different V-Series systems, see the *Hardware Universe*.

Related references

Considerations before using synchronous SnapMirror on page 90

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.
  An entire site could fail because of a disaster or prolonged power failure. If this situation occurs, the site administrator enters a command to enable the surviving node to take over the functions of the partner. Data is accessed on the plex of the surviving node.

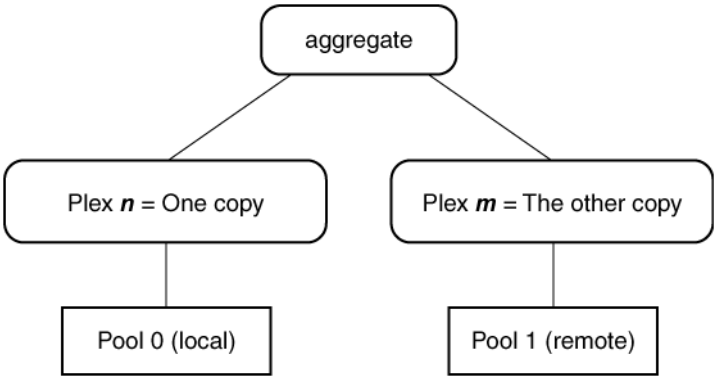
Each of the two plexes must be on a separate set of array LUNs. The plexes can be in two physically separate locations on the same storage array, or each of the two plexes can be on a different storage array.

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.

Data ONTAP needs to know whether a plex is local to the system on which the aggregate is configured or in a remote location. “Local” in the context of storage arrays means on the storage array connected to the Data ONTAP systems on which the aggregate is configured. The SyncMirror *pool* to which an array LUN is assigned provides the information that Data ONTAP needs to determine whether the plex is local or remote.

The following illustration shows the relationships of plexes and pools to an aggregate. One plex is associated with pool0 and one plex is associated with pool1. The number 0 is typically associated

with the local pool and the number 1 is typically associated with the remote pool. The remote plex is the mirror of the aggregate.



Implications of storage type when mirroring aggregates in a SyncMirror configuration

You can mirror data only between the same types of storage.

When planning for mirroring of aggregates for systems that can use both array LUNs and disks, keep the following in mind:

- 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.
- If your Data ONTAP system is using array LUNs, you can mirror the aggregate between two different sets of array LUNs that are in separate locations on the same storage array or on two storage arrays.
  You must follow the requirements for setting up SyncMirror with array LUNs, some of which are different from setting up SyncMirror with disks. After SyncMirror is set up properly, Data ONTAP can switch to using the mirrored storage without interruption in service if one or more LUNs on a storage array become unavailable.
- You cannot mirror an aggregate between a native disk shelf on a Data ONTAP system and a storage array.

Related concepts

Requirements for setting up SyncMirror with array LUNs on page 318
Example of SyncMirror pool assignments for array LUNs on page 322

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.

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 non-MetroCluster deployment with SyncMirror you can use one or two storage arrays.

For a MetroCluster configuration that uses array LUNs, you must use two separate storage arrays for the mirror.

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.
- 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 will be used for the aggregate and its mirror to the same Data ONTAP system. This system will *own* the aggregate.

Checksum consistency requirement

All array LUNs in both sets of LUNs that will be used for the plexes of the aggregate must be the same checksum type.

SyncMirror option requirements

The `cf.mode` option must be set to `ha` and the `cf.remote_syncmirror.enable` option must be set to `on`.

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.

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.

For performance reasons, in a MetroCluster configuration you want the read operations to be served from the local pool to avoid read operations over long distances.

Mirroring restrictions

You cannot mirror data between a native disk shelf on a Data ONTAP system and a storage array.

Specific requirements for a MetroCluster configuration that uses array LUNs

You can have both mirrored and unmirrored volumes in a MetroCluster configuration. However, the MetroCluster configuration can preserve data only if volumes are mirrored. Unmirrored volumes are lost if the storage on which they reside is destroyed.

Requirements for SyncMirror in a MetroCluster configuration that uses array LUNs include the following:

- You must configure each plex to be on a separate set of array LUNs on different storage arrays.
- You must mirror the root volume, with one plex at each site.
- You should mirror data volumes, with one plex at each site.
- You must connect unmirrored storage to both nodes, just as for mirrored storage. You cannot have storage that is connected to only one node in a MetroCluster configuration.

For more information about a MetroCluster configuration that uses array LUNs, see the *Data ONTAP High Availability and MetroCluster Configuration Guide for 7-Mode*.

Verification of pathing

Before you create your aggregate and mirror it, ensure that there are two paths to an array LUN.

For more information about checking paths to array LUNs, see the *FlexArray Virtualization Installation Requirements and Reference Guide*.

Related concepts

Implications of storage type when mirroring aggregates in a SyncMirror configuration on page 318
SyncMirror pool assignment planning for array LUNs on page 321
Example of SyncMirror pool assignments for array LUNs on page 322
Common errors when setting up SyncMirror pools with array LUNs on page 325
Troubleshooting errors with SyncMirror pool assignment for array LUNs on page 325
Rules for adding array LUNs to a mirrored aggregate on page 335

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 two storage arrays, or two locations on the same storage array, 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 need to ensure that each LUN set has the same number of LUNs and that the LUNs in each set are the same size.

If you are using one storage array, you need to ensure that each LUN is from a different disk group (RAID group) on the storage array.

Physical location of storage (assuming two storage arrays)	Pool to which the array LUNs need to be assigned	Command setting
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. (You can think of the Data ONTAP system as the owner of the aggregate.)	Local pool (pool0)	`disk assign -p 0`
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.)	Remote pool (pool1)	`disk assign -p 1`

Note: You use the `-p` parameter of the `disk assign` command to specify the SyncMirror pool assignment.

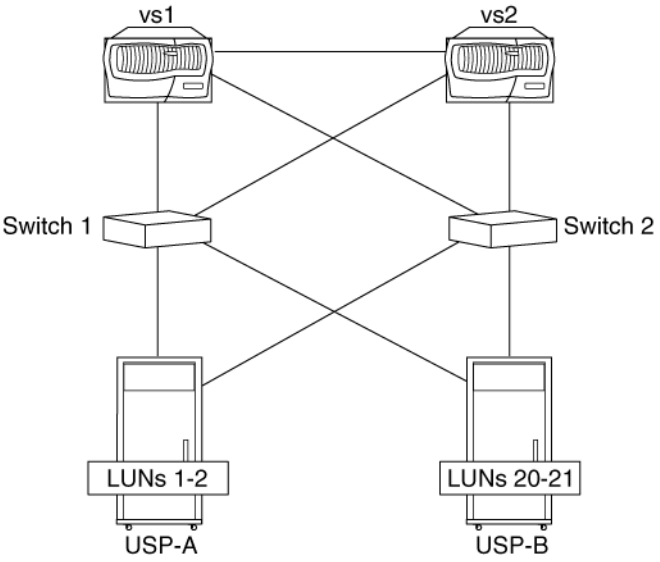
Related concepts

Requirements for setting up SyncMirror with array LUNs on page 318
Example of SyncMirror pool assignments for array LUNs on page 322

Example of SyncMirror pool assignments for array LUNs

A SyncMirror deployment with array LUNs requires careful planning so that you assign array LUNs on the local storage array to the local pool and array LUNs on the remote storage array to the remote pool.

The following illustration shows the configuration for this example. The storage arrays are the same model, which is required to set up SyncMirror with array LUNs.



For simplicity, this example discusses mirroring of just one aggregate on system vs1 (one-way mirroring). However, you can set up two-way mirroring; that is, you can mirror an aggregate owned by vs1 and also mirror an aggregate owned by vs2.

Assume that you want to create a 20-GB aggregate on system vs1 using two 10-GB LUNs, and you want to use SyncMirror to be able to mirror that aggregate. The total amount of storage that is required to set up mirroring is 40 GB, 20 GB from each storage array. The aggregate is created from the LUNs on USP-A (because USP-A is the storage that is local to vs1). To be able to mirror the aggregate, two 10-GB LUNs must be available on USP-B.

Availability of array LUNs

Assume that the storage array administrator made the following 10-GB LUNs available to Data ONTAP and that you are going to use these LUNs for mirroring:

- From USP-A: LUN1 and LUN2
- From USP-B: LUN20 and LUN21

Assignment of LUNs to a Data ONTAP system

System vs1 will be the owner of the aggregate. Therefore, you must assign both sets of LUNs that will be used for the plexes of the aggregate to vs1. You should assign the `-p` pool parameter when you initially assign each array LUN to vs1. If you assign array LUNs to pools later, you must unassign the LUNs in the remote location and assign them to vs1 again, this time with the remote LUNs assigned to pool1.

Pool (`-p`) parameter settings for this example are as follows:

- The array LUNs on storage array USP-A (the local storage array) must be assigned to the local pool—pool0
- The array LUNs on storage array USP-B (the remote storage array) must be assigned to the remote pool—pool1

The commands are as follows (assuming the checksum type is block):

- **disk assign *LUN1 LUN2* -p 0 -c block**
- **disk assign *LUN20 LUN21* -p 1 -c block**

System vs1 now owns two LUNs on each storage array. The pools are specified correctly so that when the mirrored aggregate is created, LUN20 and LUN21 can be the mirror of LUN1 and LUN2. The checksum type is the same for all the array LUNs that will be in the two plexes of the aggregate.

Note: Be sure to check that there are two paths to each LUN before creating the aggregate and mirroring it.

Creation of the aggregate and mirroring it

You can create the aggregate and mirror it all in the same command or create a mirrored aggregate later by adding a plex to an existing aggregate. The command formats are as follows:

To...	The command format to use is...
Create an aggregate and mirror it in the same step	**aggr create *aggrname* -m -d *disk-list_plex0* -d *disklist_plex1*** *disk-list* is the array LUNs to add to the plex.
Add a plex to an existing aggregate	**aggr *aggrname* mirror -d *disk-list_plex1*** *disk-list* is the array LUNs to add to the plex.

The following command creates the aggregate for this example and mirrors it. Plex 0 of the aggregate named vs1aggregate is created with LUN 1 and LUN2, and the other plex of the aggregate is created with LUN20 and LUN21:

aggr create vs1aggregate -m -d *LUN1 LUN2* -d *LUN20 LUN21*

During creating of a mirrored aggregate, the same number and size of array LUNs needs to be specified for each plex. Also the LUNs needed for plex 1 must exist on the storage array and have been assigned to pool1 (the remote pool).

Related concepts

Requirements for setting up SyncMirror with array LUNs on page 318

Implications of storage type when mirroring aggregates in a SyncMirror configuration on page 318
SyncMirror pool assignment planning for array LUNs on page 321

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:

Error	Result
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.	You cannot create the mirror for the aggregate. The mirror creation process does not allow mixed pools within a plex.
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 (`p1`) and assign the set of LUNs on the remote storage array to the local pool (`p0`)	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.
You plan to use two storage arrays for SyncMirror but you mistakenly create a mirrored aggregate with both pools from the same storage array.	Data is lost if the storage array fails.

Related concepts

Requirements for setting up SyncMirror with array LUNs on page 318
Troubleshooting errors with SyncMirror pool assignment for array LUNs on page 325
Rules for adding array LUNs to a mirrored aggregate on page 335

Troubleshooting errors with SyncMirror pool assignment for array LUNs

To troubleshoot SyncMirror pool assignment problems with array LUNs, you need to look at 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. If you are using just one storage array, you need to make sure that your pool assignments in Data ONTAP match the sets of LUNs that you determined were local and remote.

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:

- Look at switch zoning
- Look at the output of the Data ONTAP `storage show disk -p` command

You can use the following Data ONTAP commands to check the pool assignments of specific array LUNs:

- `aggr status -r`
- `sysconfig -r`

If you find problems, correct array LUN pool assignment errors as needed. In the Data ONTAP `disk assign -p` command, `-p0` represents the local pool and `-p1` represents the remote pool.

Related concepts

Common errors when setting up SyncMirror pools with array LUNs on page 325

Requirements for setting up SyncMirror with array LUNs on page 318

Considerations for using mirrored aggregates

If you want to use mirrored aggregates, you can either create a new aggregate with two mirrored plexes, or add a plex to an existing aggregate.

Note: A mirrored aggregate can have only two plexes.

The rules for the selection of disks or array LUNs, for using 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 the plexes.
- Disks are selected first on the basis of equivalent bytes per sector (bps) size, 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.

Note: When creating an aggregate, Data ONTAP selects disks from the plex which has the most available disks. You can override this selection policy by specifying the disks to use.

Related concepts

What mirrored aggregates are on page 316

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 a collection of disks to create 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 from the disks in pool1. Therefore, before enabling SyncMirror, you should ensure that the disks are installed in at least two shelves and the shelves are connected to the system with separate cables and adapters. 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.

For more information about V-Series system slot assignments, see the *Hardware Universe*.

Viewing plexes and spare pools

You can view the plexes and spare disks or array LUNs. It is useful to view the assignments when you are adding disks or array LUNs to an aggregate, or when you need to identify the pool used by each plex.

Step

1. Enter one of the following commands.

   `sysconfig -r`

   or

   `aggr status -r`

   or

   `vol status -r`

Example

In this example, the `aggr status -r` command is used to view the disks in plexes and spare disks in disk pools.

```
system1> aggr status -r

Aggregate vol0 (online, raid4) (block checksums)
Plex /vol0/plex0 (online, normal, active, pool1)
```

```
RAID group /vol0/plex0/rg0 (normal)

RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
--------- ------  ------------- ---- ---- ---- ----- --------------    ---
parity    9a.16   9a    1   0   FC:A   1  FCAL 10000 34000/69632000    34
data      9a.17   9a    1   1   FC:A   1  FCAL 10000 600/1228800       76
data      9a.20   9a    1   4   FC:A   1  FCAL 10000 34000/69632000    34

Aggregate GreG (online, raid4) (block checksums)
Plex /GreG/plex0 (online, normal, active, pool1)
RAID group /GreG/plex0/rg0 (normal)

RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
--------- ------  ------------- ---- ---- ---- ----- --------------    ---
parity    9a.18   9a    1   2   FC:A   1  FCAL 10000 34000/69632000    34
data      9a.19   9a    1   3   FC:A   1  FCAL 10000 34000/69632000    34


Pool1 spare disks

RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)
Ph)
---------       ------  ------------- ---- ---- ---- ----- --------------
---
Spare disks for block or zoned checksum traditional volumes or aggregates
spare           9a.24   9a    1   8   FC:A   1  FCAL 10000 34000/69632000    34
spare           9a.29   9a    1   13  FC:A   1  FCAL 10000 34000/69632000    34

Pool0 spare disks (empty)

Partner disks

RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)
Ph)
---------       ------  ------------- ---- ---- ---- ----- --------------
---
partner         9b.25   9b    1   9   FC:B   1  FCAL 10000 0/0               34
partner         9b.16   9b    1   0   FC:B   1  FCAL 10000 0/0               34
partner         9b.17   9b    1   1   FC:B   1  FCAL 10000 0/0               34
partner         9b.21   9b    1   5   FC:B   1  FCAL 10000 0/0               34
partner         9b.18   9b    1   2   FC:B   1  FCAL 10000 0/0               34
partner         9b.22   9b    1   6   FC:B   1  FCAL 10000 0/0               34

tpubs-cf1>
```

Example

In this example, the `vol status -r` command is used to view the disks in plexes and spare disks in disk pool.

```
system1> vol status -r

Volume vol0 (online, raid4) (block checksums)
  Plex /vol0/plex0 (online, normal, active, pool1)
    RAID group /vol0/plex0/rg0 (normal)

      RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)
Ph)
      --------- ------  ------------- ---- ---- ---- ----- --------------
---
      parity    9a.16   9a    1   0   FC:A   1  FCAL 10000 34000/69632000    34
```

```
      data      9a.17   9a    1   1   FC:A   1  FCAL 10000 600/1228800       76
      data      9a.20   9a    1   4   FC:A   1  FCAL 10000 34000/69632000    34

Aggregate aggrz (online, raid4) (block checksums)
  Plex /aggrz/plex0 (online, normal, active, pool1)
    RAID group /aggrz/plex0/rg0 (normal)

      RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)
Ph)
      --------- ------  ------------- ---- ---- ---- ----- --------------
---
      parity    9a.25   9a    1   9   FC:A   1  FCAL 10000 34000/69632000    34
      data      9a.26   9a    1   10  FC:A   1  FCAL 10000 34000/69632000    34
      data      9a.27   9a    1   11  FC:A   1  FCAL 10000 34000/69632000    34
      data      9a.28   9a    1   12  FC:A   1  FCAL 10000 34000/69632000    34

Pool1 spare disks
RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)
Ph)
---------       ------  ------------- ---- ---- ---- ----- --------------
---
Spare disks for block or zoned checksum traditional volumes or aggregates
spare           9a.24   9a    1   8   FC:A   1  FCAL 10000 34000/69632000    34
spare           9a.29   9a    1   13  FC:A   1  FCAL 10000 34000/69632000    34

Pool0 spare disks (empty)

Partner disks
RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)
Ph)
---------       ------  ------------- ---- ---- ---- ----- --------------
---
partner         9b.25   9b    1   9   FC:B   1  FCAL 10000 0/0               34
partner         9b.16   9b    1   0   FC:B   1  FCAL 10000 0/0               34
partner         9b.17   9b    1   1   FC:B   1  FCAL 10000 0/0               34
partner         9b.21   9b    1   5   FC:B   1  FCAL 10000 0/0               34
partner         9b.18   9b    1   2   FC:B   1  FCAL 10000 0/0               34
partner         9b.22   9b    1   6   FC:B   1  FCAL 10000 0/0               34
partner         9b.23   9b    1   7   FC:B   1  FCAL 10000 0/0               34
```

Related tasks

Converting an aggregate to a mirrored aggregate on page 332
Creating a mirrored aggregate on page 329

Creating a mirrored aggregate

You can use the `aggr create` command to create a mirrored aggregate. You have several options for how to specify the disks or array LUNs when you create a mirrored aggregate.

About this task

When you create an aggregate, you can specify the aggregate to use SyncMirror. This ensures that the aggregate is a mirrored one from the start.

There are three methods of creating a mirrored aggregate:

- Ascertaining the disks or array LUNs available, and specify which ones to use.

- Allowing Data ONTAP to automatically use the disks or array LUNs that are available.
- Previewing the disks or array LUNs that Data ONTAP has selected. Then, you can either use the same selection or modify the selection.

Steps

1. You can use the `aggr create` command to create a mirrored aggregate. As required, choose one of the actions from the following table.

If...	Then...
You know the disks or array LUNs to be used for the mirrored aggregate, and want to specify the disks	Enter the following command: **`aggr create aggr_name -m -d disk-list -d disk-list`** Ensure that you choose the correct number and size of disks for each set of disks. **Note:** Both the `-d` options must be used, one for each plex. If you specify only one disk set, the creation of a mirrored aggregate fails. This step completes the creation of a mirrored aggregate.
You want Data ONTAP to select the disks for the mirrored aggregate	Go to the next step. **Note:** Allowing Data ONTAP to select the disks is the easiest method of creating a mirrored aggregate.

aggr_name is the name of the mirrored aggregate.

`-m` specifies the creation of a mirrored aggregate.

disk-list consists of the disk IDs of two or more available disks. You should separate multiple disks with the space character.

Example

The following command creates a mirrored aggregate named `aggrA` with disks 6.1 and 6.2 on one plex, and disks 8.1 and 8.2 on the other plex:

```
aggr create aggrA -m -d 6.1 6.2 -d 8.1 8.2
```

2. As required, choose one of the actions from the following table.

If you want...	Then enter the following command...
Data ONTAP to automatically specify the disks for the mirrored aggregate	**`aggr create aggr_name -m ndisks[@disk-size]`** This step completes the creation of a mirrored aggregate.

If you want...	Then enter the following command...
To preview the disks selected by Data ONTAP for creating the mirrored aggregate	**aggr create *aggr_name* -n -m *ndisks*[@*disk-size*]** **Note:** The `-n` option instructs Data ONTAP not to create the aggregate, but to display the disks selected automatically for creating the mirrored aggregate. The system displays an `aggr create` command indicating the disks to be used for the mirrored aggregate. Go to the next step.

ndisks is the number of disks to use, which must be a minimum of four and an even number. Even numbers are required because the disks are equally divided between the two plexes.

disk-size specifies the disk capacity, in gigabytes. There must be enough disks of this size available.

Note: If you want to specify the disk size, first determine the size of spare disks in the disk pools by using the `vol status -r` command.

Example

The following command creates a mirrored aggregate named `aggrB`, with two plexes, each plex containing three disks.

aggr create aggrB -m 6

3. As required, choose one of the actions from the following table.

If you want to...	Then...
Use the selection provided in the preview	Enter the `aggr create` command, displayed by the system in the previous step.
Change one or more of the disks provided in the preview	Enter the `aggr create` command, displayed by the system in the previous step. Substitute one or more disks with other disks that you want to specify.

Example

The following command displays the `aggr create` command, with a preview of the disk selection for creating a four-disk mirrored aggregate named `aggrC`.

aggr create aggrC -n -m 4
The system returns the following command:

```
aggr create aggrC -m -d 5.1 5.3 -d 8.3 8.4
```

Use the preceding command to create a mirrored aggregate named `aggrC`, with two disks in each plex.

For more information about aggregates, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Related tasks

Viewing plexes and spare pools on page 327

Related references

Considerations for using mirrored aggregates on page 326

Converting an aggregate to a mirrored aggregate

You can convert an aggregate to a mirrored aggregate by adding a plex to the aggregate. Thus, you can use SyncMirror to mirror a previously unmirrored aggregate. You can check the details of an aggregate by using the `aggr status -r` command.

Before you begin

You must have ensured that a mirrored aggregate has only two plexes.

About this task

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.

The three methods of converting an aggregate to a mirrored aggregate are as follows:

- Ascertaining the disks or array LUNs available, and specify which ones to use.
- Allowing Data ONTAP to automatically use the disks or array LUNs that are available.
- Previewing the disks or array LUNs that Data ONTAP has selected. Then, you can either use the same selection or modify the selection.

Steps

1. Use the `aggr mirror` command to add a plex to an aggregate. As required, choose one of the actions from the following table.

If...	Then...
You know the disks or array LUNs to be used for the plex addition and want to specify the disks or array LUNs	Enter the following command: **aggr mirror *aggr_name* -d *disk-list*** Ensure that you choose the correct number and size of disks or array LUNs in the list. **Note:** These disks or array LUNs should be from a different pool than that already being used by the aggregate. This step adds a plex to the aggregate, making it a mirrored aggregate.
You want Data ONTAP to select the disks or array LUNs for the plex addition	Go to the next step. **Note:** Allowing Data ONTAP to select the disks or array LUNs is the easiest method of adding a plex to an aggregate.

aggr_name is the name of the mirrored aggregate.

disk-list is the list of IDs of two or more available disks or array LUNs. Separate the entries with the space character.

Example

The following command adds a plex to the aggregate `aggrD` with disks 7.1 and 7.2, and makes `aggrD` a mirrored aggregate.

```
aggr mirror aggrD -d 7.1 7.2
```

2. As required, choose one of the actions from the following table.

If you want...	Then enter the following command...
Data ONTAP to automatically specify the disks or array LUNs for the plex addition	**aggr mirror *aggr_name*** This step adds a plex to the aggregate, making it a mirrored aggregate.
To preview the disks or array LUNs selected by Data ONTAP for the plex addition	**aggr mirror *aggr_name* -n** **Note:** The `-n` option instructs Data ONTAP not to add the plex, but to display the disks or array LUNs selected automatically, for the plex addition. The system displays an `aggr mirror` command indicating the disks or array LUNs to be used for the plex addition. Go to the next step.

Example

The following command adds a plex to the aggregate `aggrE`, making `aggrE` a mirrored aggregate.

```
aggr mirror aggrE
```

3. As required, choose one of the actions from the following table.

If you want to...	Then...
Use the selection provided in the preview	Enter the `aggr mirror` command displayed by the system in the previous step.
Change one or more of the disks or array LUNs provided in the preview	Enter the `aggr mirror` command displayed by the system in the previous step. Substitute any disks or array LUNs with what you want to specify.

Example

The following command displays the `aggr mirror` command with a preview of the disk selection for adding a plex to an aggregate named `aggrF`.

```
aggr mirror aggrF -n
```

The system returns the following command:

```
aggr mirror aggrF -d 8.1 8.2
```

Use the preceding command to add a plex to the aggregate `aggrF` to make it a mirrored aggregate.

For more information about aggregates, see the *Data ONTAP Storage Management Guide for 7-Mode*.

Related tasks

Viewing plexes and spare pools on page 327

Related references

Considerations for using mirrored aggregates on page 326

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 remains identical.

Keep the following rules in mind when adding array LUNs to a mirrored aggregate:

- You must add an even number of array LUNs to the mirrored aggregate.
- You must divide the array LUNs you add equally between the two plexes.
- The array LUNs for each plex must come from a different LUN set. Do not mix the LUNs from the two LUN sets 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 318
Common errors when setting up SyncMirror pools with array LUNs on page 325

Adding disks to a mirrored aggregate, where Data ONTAP selects the disks

When you add disks to a mirrored aggregate, you can allow Data ONTAP to select the disks.

Step

1. Enter the following command:

   aggr add ***aggrname ndisks*****[****@disk-size****]**

   aggrname is the name of the aggregate to which you are adding disks or array LUNs.

ndisks is the number of disks or array LUNs to use. This number must be an even number, because half these disks or array LUNs must be in each pool.

disk-size specifies the disk or array LUN capacity, in gigabytes.

Note: If you use the `disk-size` option, you should first determine the size of spare disks or array LUNs in the pools using the `aggr status -r` or `vol status -r` command.

Adding disks or array LUNs to a mirrored aggregate, where the user selects the disks

If you select the new disks or array LUNs to add to the mirrored aggregate, you must ensure that you add the correct number of disks or array LUNs of the correct sizes. The disks or array LUNs must have the same checksum compatibility, and disks or array LUNs for each plex must be in different pools.

Steps

1. Enter the following command to list available spare disks or array LUNs.

   aggr status -r

2. Choose the spare disks or array LUNs you want to add.

3. Enter the following command:

   aggr add *aggrname* -d *disk-list* -d *disk-list*

   aggrname is the name of the aggregate to which you are adding disks or array LUNs.

   disk-list consists of disk or array LUNs IDs of one or more available disks; separate multiple disks or array LUNs with a space character. Both `-d` options must be used, one for each plex. Data ONTAP automatically assigns the disks or array LUNs to the appropriate plex.

   Note: Specifying only one disk or array LUN set using the `-d` option will fail.

In the following example, the `aggr add` command adds disk 3.1 to one plex and disk 8.1 to the other plex of the `aggrD` mirrored aggregate.

```
aggr add aggrD -d 6.1 -d 8.1
```

Related references

Rules for adding disks to a mirrored aggregate on page 335

Adding disks to a mirrored aggregate, where the user selects the disks with assistance from Data ONTAP

You can preview what disks Data ONTAP would select if it were to add new disks to the aggregate. Then, you can add the disks Data ONTAP selected or substitute other disks.

Steps

1. Enter the following command:

   aggr add *aggrname* -n *ndisks*[*@disksize*]

   aggrname is the name of the mirrored aggregate to which you are adding disks.

   ndisks is the number of disks to use. Adding new disks will fail if the number of disks is an odd number.

   disk-size specifies the disk capacity, in gigabytes. Half these disks must be in each disk pool.

   Note: If you use the `disk-size` option, first determine the size of spare disks in the disk pools using the `vol status -r` command.

   Data ONTAP returns an `aggr add` command that specifies the disks it would add to each plex of the mirrored aggregate.

2. As required, choose one of the actions from the following table.

If you want to...	Then...
Use the disks specified by Data ONTAP	Enter the `aggr add` command specified.
Substitute other disks for one or more disks specified by Data ONTAP	Enter the `aggr add` command and substitute other disks for one or more of the disks specified.

In the following example, the `aggr add` command provides a preview of the disks that Data ONTAP would use when adding two new 144-GB disks to the `aggrA` mirrored aggregate.

```
aggr add aggrA -n 2@144
```

Data ONTAP returns the following command:

```
aggr add aggrA -d 6.4 -d 8.6
```

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.

Viewing the status of plexes

At any point, you might want to view the status of a given plex.

Before you begin

To view the status of a plex, the plex must be online.

Step

1. To view the status of plexes, enter one of the following commands.

   `sysconfig -r`
   or
   `aggr status -r`
   or
   `vol status -r`

Changing the state of a plex

You can change the state of a plex in a mirrored aggregate, from `online` to `offline`, and `offline` to `online`.

About this task

Data ONTAP specifies the state of a plex as `resyncing` when synchronizing the two plexes of a mirrored aggregate. Also, when you create a mirrored aggregate by adding a plex to an unmirrored aggregate, Data ONTAP puts the added plex in a `resyncing` state.

Data ONTAP allows you to change the state of a plex from `offline` to `online`, and from `online` to `offline`.

Step

1. To change the state of a plex, enter the following command:

   `aggr { online | offline } `*`plexname`*

Splitting a mirrored aggregate

Splitting a mirrored aggregate removes the relationship between its two plexes and creates two independent unmirrored aggregates. After splitting, both the aggregates come online.

Before you begin

Ensure that both plexes of the mirrored aggregate you are splitting are online and operational.

About this task

You might split a mirrored aggregate for one of the following reasons.

- You want to stop mirroring an aggregate.
- You want to move a mirrored aggregate to another location.
- You want to modify the mirrored aggregate, and test the modification before applying it. You can apply and test the modifications on the split-off copy of the plex, then apply those changes to the untouched original plex.

Before splitting, a mirrored aggregate or traditional volume has two plexes, `plex0` and `plex1`. After splitting, the new unmirrored aggregate with the new name has one plex, `plex0`. The new unmirrored aggregate with the original name also has one plex, either `plex0` or `plex1`.

The plex name for an unmirrored aggregate is unimportant because the aggregate has only one plex. If you use SyncMirror to mirror one of these unmirrored aggregates, the resulting plex names will always be `plex0` and `plex1`.

Note: You do not need to stop applications that are using the aggregate, before splitting a mirrored aggregate.

Step

1. Enter the following command:

   `aggr split` *`aggrname/plexname new_aggr`*

   `aggrname` is the name of the mirrored aggregate.

   `plexname` is the name of one of the plexes in the mirrored aggregate.

   `new_aggr` is the name of the new aggregate that would be created.

   #### Example

   The following command splits `plex0` from mirrored aggregate `aggr0` and names the new aggregate `aggrNew`.

   `aggr split aggr0/plex0 aggrNew`

   After splitting, there are two unmirrored aggregates, `aggr0` and `aggrNew`.

Related tasks

Converting an aggregate to a mirrored aggregate on page 332
Rejoining split aggregates on page 340
Removing a plex from a mirrored aggregate on page 342
Changing the state of a plex on page 339

Rejoining split aggregates

You can rejoin split aggregates. You might want to do this if you have set up an HA pair in a MetroCluster configuration and a disaster breaks the HA pair.

About this task

There are additional considerations when planning to rejoin split aggregates that previously used MetroCluster to mirror SnapLock volumes.

Attention: When you rejoin split aggregates, Data ONTAP mirrors the data from one aggregate to the other and destroys data that existed on that aggregate before the rejoin.

You can use MetroCluster to mirror SnapLock volumes from one site to another. With proper configuration, the SnapLock volumes retain their characteristics at the mirror site. In case of a failure at the primary site, and if necessary, you can use the `cf forcetakeover –d` command to break the mirror relationship and to bring the mirror site online. Once the failure at the primary site is resolved, the MetroCluster mirror relationship can be reestablished. The mirrors can be resynchronized before resuming normal operation.

Attention: The primary node might have data that was not mirrored before using the `cf forcetakeover –d` command. For example, the data might have been written to the primary node while the link between the sites was inoperative. In such a case, you should back up the SnapLock volumes in the aggregate on the primary site, before resynchronizing the two mirror aggregates. This step of creating an additional backup for the SnapLock volumes is required to ensure the availability of all data.

For more information about backing up data in SnapLock volumes using SnapMirror, see the *Data ONTAP Archive and Compliance Management Guide for 7-Mode*.

For more information about MetroCluster deployment, see the *Data ONTAP High Availability and MetroCluster Configuration Guide for 7-Mode*.

Steps

1. Determine the aggregate whose data you want to keep and the aggregate whose data you want to be overwritten.
2. If the aggregate whose data is to be overwritten is online, take it offline by entering the following command.

   `aggr offline` ***`aggrname`***

   `aggrname` is the name of the aggregate.

   Note: An error message appears if the aggregate is already offline.
3. Re-create the mirrored aggregate by entering the following command.

   `aggr mirror` ***`aggrname1`*** **`-v`** ***`aggrname2`***

   `aggrname1` is the name of the aggregate whose data you want to keep.

   `aggrname2` is the name of the aggregate whose data you want to be overwritten by *`aggrname1`*.

Removing a plex from a mirrored aggregate

You can remove a plex from a mirrored aggregate. You might do this if you want to stop mirroring the aggregate, or if there is a problem with the plex. Removing a plex results in an unmirrored aggregate.

About this task

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 it using a different set of disks or array LUNs, if the problem cannot be fixed.

Steps

1. Take the selected plex offline by entering the following command:

   aggr offline *plex-name*

   plex-name is the name of one of the mirrored plexes.

   Note: Only one plex at a time can be taken offline.

2. Destroy the plex you took offline by entering the following command:

   aggr destroy *plex-name*

Result

Removing and destroying a plex from a mirrored aggregate results in an unmirrored aggregate, because the aggregate now has only one plex.

After removing the plex, Data ONTAP converts the disks or array LUNs used by the plex into hot spares.

Comparing plexes of a mirrored aggregate

The plexes of a mirrored aggregate are automatically synchronized. You can confirm this by using the `aggr verify` command to check for, and in rare cases, fix any inconsistencies.

Before you begin

The mirrored aggregate must be online and not mirror degraded before you can compare the plexes.

About this task

Comparing plexes might affect system performance.

When comparing the two plexes of a mirrored aggregate, you can choose one of the following options.

- Data ONTAP compares plexes without correcting differences. This is the default behavior.
- Data ONTAP compares plexes and corrects the differences it finds. To correct differences, you need to specify which plex to correct.
  The plex is specified as *plexnumber* (0,1, and so on.)

  Attention: This process might use advanced Data ONTAP commands. Contact technical support before correcting differences using this option.

Step

1. To compare the two plexes of a mirrored aggregate, choose one of the actions from the following table.

If...	Then...
You do not want Data ONTAP to correct differences	Enter the following command: **aggr verify start *aggrname* -n** *aggrname* is the name of the mirrored aggregate whose plexes you are comparing.
You want Data ONTAP to correct differences	Enter the following command: **aggr verify start *aggrname* -f *plexnumber*** *aggrname* is the name of the mirrored aggregate whose plexes you are comparing.

   Note: If *aggrname* is not specified, Data ONTAP compares the plexes of all mirrored aggregates that are online and not mirror degraded.

Stopping plex comparison

You can stop Data ONTAP from comparing the plexes of a mirrored aggregate. Such a step might be required if the comparison affects system performance.

Step

1. To stop Data ONTAP from comparing plexes, enter the following command:

   aggr verify stop *aggrname*

   Note: If *aggrname* is not specified, Data ONTAP stops comparing plexes for all mirrored aggregates.

Suspending plex comparison

You can suspend the comparison of the plexes of a mirrored aggregate, instead of stopping the comparison. The comparison remains suspended until you resume it, stop it, or reboot the system.

Step

1. To suspend a comparison of plexes, enter the following command:

   aggr verify suspend *aggrname*

   Note: If *aggrname* is not specified, Data ONTAP suspends the comparison of plexes for all mirrored aggregates.

Resuming plex comparison

After you have suspended a comparison of plexes, you can resume the comparison by using the `aggr verify resume` command. Otherwise, the comparison remains suspended until you stop it, or reboot the system.

Step

1. To resume a suspended comparison of plexes, enter the following command:

   aggr verify resume *aggrname*

   Note: If *aggrname* is not specified, Data ONTAP resumes the comparison of plexes for all mirrored aggregates.

Viewing the status of a plex comparison

You can view the status of a plex comparison operation. The status tells you what percentage of the plex comparison has been completed, and whether plex comparison of a mirrored aggregate is suspended.

Step

1. To view the status of a plex comparison, enter the following command:

   aggr verify status *aggrname*

   Note: If *aggrname* is not specified, Data ONTAP displays the status of all mirrored aggregates whose plexes are being compared.

How to avoid unnecessary RAID reconstruction when a plex fails

If a plex in a mirrored aggregate fails due to fabric failure such as a cable disconnection or a Fiber Channel switch power failure, you can manually take the failed plex offline before the cable connection or power supply is restored to avoid unnecessary RAID reconstruction.

You can take the failed plex offline by using the `aggr offline` *`aggr_name/plex-name`* command.

Database protection using NVFAIL

Data ONTAP provides database protection using the `nvfail` option. The `nvfail` option enables Data ONTAP to detect nonvolatile RAM (NVRAM) inconsistencies at boot time or while taking over in an HA pair.

You use this option to warn database administrators of NVRAM problems that can compromise database validity. If Data ONTAP finds any problems, database instances stop responding or shut down, and Data ONTAP sends error messages to the console to alert you to check the state of the database.

Note: The `nvfail` option is disallowed on SnapLock volumes.

How NVFAIL protects database files

When booting up or while taking over in a failover configuration, Data ONTAP checks for NVRAM errors. If no errors are detected, start the file service is started normally. However, if NVRAM errors are detected Data ONTAP stops database instances from responding.

When you enable the `nvfail` option, one of the following processes takes place during bootup.

If...	Then...
Data ONTAP detects no NVRAM errors	File service starts normally.
Data ONTAP detects NVRAM errors and you use the optional `nvfail_rename` file	1. Data ONTAP returns a stale file handle (`ESTALE`) error to NFS clients trying to access the database. This causes the application to stop responding, crash, or shut down. Data ONTAP then sends an error message to the system console and log file. 2. Data ONTAP renames database files specified in the `nvfail_rename` file by appending .nvfail to the original file names, making those files unavailable to both CIFS and NFS clients.

If...	Then...
Data ONTAP detects NVRAM errors and you do not use the optional `nvfail_rename` file	**1.** Data ONTAP returns a stale file handle (`ESTALE`) error to NFS clients trying to access the database, causing the application to stop responding, crash, or shut down. Data ONTAP then sends an error message to the system console and log file. **2.** No database files are renamed. When the application restarts, files are available to CIFS clients, even if you have not verified that they are valid. For NFS clients, files remain inaccessible as long as the file system is not remounted.
Data ONTAP detects NVRAM errors on a volume that contains LUNs and you use the optional `nvfail_rename` file **Note:** The term *LUNs* in this context refers to the LUNs that Data ONTAP serves to clients, not to the array LUNs used for storage on a storage array.	**1.** Data ONTAP takes the LUNs offline in the volume that had the NVRAM errors. **2.** Data ONTAP stops exporting those LUNs over iSCSI or FCP. **3.** Data ONTAP sends error messages to the system console and log file stating that Data ONTAP took the LUNs offline or that NFS file handles are stale (useful if the LUN is accessed over NAS protocols). **4.** Data ONTAP renames LUNs specified in the `nvfail_rename` file by appending `.nvfail` to the original LUN names.

If...	Then...
Data ONTAP detects NVRAM errors on a volume that contains LUNs and you do not use the optional `nvfail_rename` file	1. Data ONTAP takes the LUNs offline in the volume that had the NVRAM errors. 2. Data ONTAP stops exporting those LUNs over iSCSI or FCP. 3. Data ONTAP sends error messages to the system console and log file stating that Data ONTAP took the LUNs offline or that NFS file handles are stale (useful if the LUN is accessed over NAS protocols). 4. No database files are renamed. When the application restarts, files are available to CIFS clients, even if you have not verified that they are valid. For NFS clients, files remain inaccessible as long as the file system is not remounted.

Enabling database file protection

You can use the `nvfail` option of the `vol options` command to configure Data ONTAP to find NVRAM problems, stop database instances from responding or shut down, and send error messages to the console to alert the database administrator to check the state of the database. The `nvfail` option detects NVRAM inconsistencies at boot time or while taking over in an HA pair failover.

Step

1. Enter the following command:

   vol options *volume_name* nvfail [on|off]

   volume_name is the name of the volume.

   Use `on` to enable or `off` to disable protection. The default setting is `off`.

Where to look for database file verification instructions

For instructions about examining database file validity, see the documentation for the specific database software.

Adding more database file protection

In addition to protecting database files and LUNs using the `nvfail` option, you can have Data ONTAP rename database files and LUNs that you want protected. Renaming database files and LUNs prevents the database from restarting automatically and gives you an opportunity to examine the files for inconsistencies before clients can access them.

Before you begin

You should ensure that the `nvfail` option is enabled.

About this task

Data ONTAP requires a file called `/etc/nvfail_rename` in which to put the names of the files you want protected.

Steps

1. Use an editor to create or modify the nvfail_rename file in the system's `/etc` directory.
2. List the path name and file name of database files you want to protect, one file per line, within the `nvfail_rename` file. You can list any number of files.

   Example

   ```
   /vol/vol1/home/dbs/oracle-WG73.dbf
   ```

3. Save the file.

After you finish

For instructions about examining database file validity, see the documentation for your specific database software.

If your database uses LUNs, review the steps to make the LUNs accessible to the host after an NVRAM failure.

Related tasks

Making LUNs accessible to the host after an NVRAM failure on page 350

Making LUNs accessible to the host after an NVRAM failure

After an NVRAM failure, the host no longer has access to data on the LUNs. You must perform a number of actions before the database has access to the LUNs.

Steps

1. Examine the LUNs for any data inconsistencies and resolve them.
2. If you renamed LUNs using the `/etc/nvfail_rename` file, remove the `.nvfail` extension by renaming the LUNs using the `lun move` command.
3. Bring the LUNs online.
4. Export each LUN manually to the initiator.

Virus protection for CIFS

Data ONTAP allows virus-scanning PC clients running a compliant antivirus application to scan files before a CIFS client is allowed to open it.

How CIFS virus scanning works

CIFS virus scanning is carried out on dedicated PC clients running the Data ONTAP-compliant antivirus application of your choice.

When you enable the virus-scanning process through Data ONTAP on the storage system, the virus-scanning application tells the storage system to send file scanning requests.

The virus-scanning application watches for requests from the storage system. Whenever the types of files you specify are opened or changed on the storage system, Data ONTAP sends the PC client a request to scan the file.

The Data ONTAP virus-scanning process can scan multiple storage systems from a single PC client if your virus-scanning application performs this function. For more information about whether a specific virus-scanning application can accommodate scanning multiple storage systems, contact the manufacturer of your virus-scanning application.

File types scanned by default

Certain file types are scanned for viruses by default. You can add file types for better protection or remove file types for faster access.

The following table lists the file types scanned by default:

??_	DL?	IM?	OFT	SMM
ARJ	DOC	INI	OLE	SWF
ASP	DOT	JS?	OV?	SYS
BAT	DRV	LZH	PIF	VBS
BIN	EML	MD?	POT	VS?
CAB	EXE	MPP	PP?	VXD
CDR	GMS	MPT	RAR	WBK
CL?	GZ?	MSG	RTF	WPD
COM	HLP	MSO	SCR	XL?

CSC	HT?	OCX	SHS	XML

Note: In this table, the ? character is a wildcard that matches any character or no character. For example, C?? matches C, CL, CPP, C++, and so on.

Attention: To scan all files, you can use ???. However, this might severely degrade performance and is not recommended.

You can also configure Data ONTAP so that file extensions not in the default list are scanned, or only a subset of the default file extensions is scanned.

Setting up and starting virus scanning

You can set up one or more virus-scanning clients to ensure that files on your system are virus free.

About this task

The storage system anti-virus vscan feature requires NTLM or Kerberos authentication; it does not support Network Information Service (NIS) authentication.

The storage system validates any vscan server which connects to the storage system, and it requires the vscan server to connect as a user who is in the storage system's Backup Operators group.

Steps

1. Setting up PC clients as virus-scanning clients on page 352
2. Enabling virus scanning on the system on page 353
3. Setting up secondary scanning clients on page 353
4. Setting up McAfee scan detection properties for systems on page 354

Related information

Bug ID 139111: support.netapp.com/NOW/cgi-bin/bol?Type=Detail&Display=139111

Setting up PC clients as virus-scanning clients

You must set up virus-scanning software that Data ONTAP supports, and enable the Data ONTAP virus-scan feature before you can scan for viruses.

Steps

1. Make sure that the operators of the PC clients that you want to configure as virus-scanning clients are configured as “Backup Operators” or higher on the storage systems on which they will conduct virus scanning.

2. Install the Data ONTAP-customized commercial virus-scanning software on the PCs that you want to configure as virus-scanning clients. Follow the directions that accompany the virus-scanning software product.
3. After installation and configuration of the virus-scanning software on the PCs is complete, confirm the success of the installation by listing the IP addresses of the PCs now configured as virus-scanning clients. At the storage system console, enter the following command:

   vscan scanners

   The system displays a table listing the IP addresses of the active virus-scanning clients for this storage system.
4. Leave the virus-scanning client on and connected to the storage system or storage systems on which it is carrying out its virus scan operations.

Enabling virus scanning on the system

Before you use the virus-scanning software, you must start the virus scan feature of Data ONTAP.

Step

1. Enter the following command:

   vscan on [-f][on|off]

   `-f` forces virus scanning to be enabled even if no virus-scanning clients are available to scan files.

Setting up secondary scanning clients

If you configured more than one virus-scanning client to scan files on a system, you can place the additional clients on standby, not actively scanning for viruses unless the primary virus-scanning client becomes unavailable.

Steps

1. List the virus-scanning clients configured to scan on this system. On the system console enter the following command:

   vscan scanners

   #### Example

   ```
   >vscan scanners
   Virus scanners(IP and Name)      P/S ...
   ----------------------------------------
   132.132.59.12   \\XLAB-WTS      Pri ....
   ```
2. Specify, by IP address, the PC clients you want to serve as standby virus scanners by entering the following command:

vscan scanners secondary_scanners *scanner_ip*[, *scanner_ip...*]

scanner_ip can be either of the following:

- IP addresses of one or more of the configured virus-scanning clients displayed in Step 1
- IP addresses of PCs not yet configured as a virus-scanning client

Note: If the IP address you entered belongs to a PC not yet configured as a virus-scanning client for this system, you must configure it for this setting to take effect.

Example

```
vscan scanners secondary_scanners 132.132.59.14
```

3. Use the `vscan scanners` command to confirm your configuration.

Example

```
>vscan scanners
Virus scanners(IP and Name)     P/S ...
---------------------------------------
132.132.59.14   \\BORIS-PC      Sec ...
132.132.59.12   \\XLAB-WTS      Pri ....

Secondary scanners IP address list
132.132.59.14,10.20.30.40
```

Note: In this example, the address 10.20.30.40 belongs to a PC that is enabled as a standby scanner but is not turned on; therefore it is not listed in the Virus scanners (IP and Name) table, but it is listed in the Secondary scanners IP address list.

Setting up McAfee scan detection properties for systems

The anti-virus scanner is registered with the system, but the scanner scans the system share only when configured properly.

About this task

When setting up McAfee VirusScan for a system with Data ONTAP 7.1 and later, you need to specify the nature of detection.

Steps

1. Open a VirusScan console.
2. Click **Task > On-Access Scan Properties > All Processes > Go to 'Detection' Tab** .

   The Detection dialog opens.

3. In the Scan Files section select: **`When reading from disk`** and **`On network drives.`**

   Note: You must select both options to scan the system shares.

   This information can be found in the VSNA 7.1 help menu.

   Open **Help > On-Access Scanning > Configuring the on-access scanner > All processes and default processes > Detection Properties**.

Specifying file types to be scanned

If the default list of file types to be scanned is not inclusive enough or you plan to use an exclusion list in conjunction with the inclusion list, you can modify the list of file types to be scanned.

Displaying file types to be scanned

You can see the list of file types to determine if you need to add to the list, remove from the list, or replace the list with other file types that you want scanned.

Step

1. Enter the following command:

   `vscan extensions include`

   The current list of extensions to be scanned is displayed.

Adding file types to be scanned

You can add file types to the extensions list to include file types that you expect to store on the system.

About this task

A default list of file extensions is made available when you enable virus scanning; however, you can specify additional file extensions that are not in the default list.

Step

1. Enter the following command:

   `vscan extensions include add` ***`ext`*****`[,`*****`ext...`*****`]`**

   `ext` is the extension you want to add.

Example

`vscan extensions include add txt`

Note: Up to 255 file extensions can exist in the file extensions list.

Replacing file types to be scanned

You can replace the file types in the exclusion list with a short list if you are also specifying a list of file types to exclude from a scan.

Step

1. Enter the following command:

   vscan extensions include set *ext*[,*ext*...]

   ext is the extension you want to set.

Removing file types to be scanned

You can remove file types from the extension list if you are also specifying files types to exclude from a scan.

Step

1. Enter the following command:

   vscan extensions include remove *ext*[,*ext*...]

   ext is the extension you want to remove.

Resetting file types to be scanned

You can reset the list of file types to be scanned if you want to return to the default list of file types to be scanned.

Step

1. Enter the following command:

   vscan extensions include reset

   The list of file extensions is set to the default list.

Related references

File types scanned by default on page 351

Excluding file types to be scanned

You might prefer to specify what files types to exclude, rather than include, in a virus scan because of the proliferation of new file types (with new file name extensions) that might be stored on the storage system.

Displaying file types to exclude from scanning

You can see a list of file types excluded from scanning to determine if you need to add to the list or remove from the list.

Step

1. Enter the following command:

   `vscan extensions exclude`

   The current list of extensions to be excluded from virus scan is displayed.

Creating a list of file types to exclude from scanning

You can create a list of file types to exclude from virus scanning if you want to use the exclude feature either by itself or in combination with the include feature.

Step

1. Enter the following command:

   `vscan extensions exclude set` ***`ext`*****`[,`*****`ext`*****`...]`**

   `ext` is the extension or extensions that you want to set for the list of file types excluded from virus scan.

   Note: Using the `set` parameter will replace completely any existing file type extensions in the exclude list with the extensions you specified in this command. If an exclude list already exists and you merely want to add to it, use the `vscan extensions exclude add` command.

Adding file types to exclude from scanning

You can add file types to exclude from virus scanning if you decide that a file type is safe and does not need virus scanning.

Step

1. Enter the following command:

   `vscan extensions exclude add` ***`ext`*****`[,`*****`ext`*****`...]`**

   `ext` is the extension you want to add to the list of file types excluded from virus scan.

```
vscan extensions exclude add txt
```

Note: Up to 255 file extensions can exist in the file extensions list.

Removing file types to exclude from scanning

You can remove file types to exclude from virus scanning if you decide that a file type is not safe and requires virus scanning.

Step

1. Enter the following command:

   ```
   vscan extensions exclude remove ext[,ext...]
   ```

   ext is the extension you want to remove from the list of file types excluded from virus scan.

Resetting the exclude file types list to empty

You can reset the exclude file types list if you decide that none of the listed file types are safe or you want to remove the exclusion list.

Step

1. Enter the following command:

   ```
   vscan extensions exclude reset
   ```

   The list of file extensions is set to the default empty value.

Using an inclusion list in combination with an exclusion list

Because of the proliferation of new file types, you can allow any file types that you consider to be safe to go unscanned and at the same time ensure that any new, unfamiliar file type stored on the system does get scanned.

About this task

You can use the inclusion list to specify a general virus scan and use the exclusion list to specify the file types that are excluded from the general virus scan.

Steps

1. At the system console, enter the following command line specifying the extensions of all file types that you want to exclude from virus scan:

   ```
   vscan extensions exclude set ext[,ext...]
   ```

2. Enter the following command line to specify virus scan of all other file types:

   ```
   vscan extensions include set ???
   ```

   This command line instructs the virus scan program to scan all file types stored on the system. But the result of both the “exclude” and the “include” command lines together is that all file types are scanned except for the file types whose extensions have been specified in the vscan extensions exclude command line described in Step 1.

Specifying shares for scanning

You can turn off virus scanning for files in a share if the share is used only by trusted users, the files are restricted to read-only mode, or speed of access is more important than safety.

Turning virus scanning off for any access

You can turn off virus scanning for files in a share if the share is used only by trusted users, or if access speed is more important than safety.

About this task

Virus scanning for a share is turned on by default.

Step

1. Enter the following command:

   ```
   cifs shares -change share_name -novscan
   ```

   share_name is the name of the share for which you want to turn off virus scanning.

   The application does not perform a virus scan when clients access this share. The setting is persistent across reboots.

   Note: You can set these share attributes for all CIFS user home directories by using `cifs.homedir` as the share name, as given in the following example.

   ```
   cifs shares -change cifs.homedir -novscan
   ```

Turning scanning on for any access

You can turn on virus scanning for files in a share if the share is used by users other than trusted users or safety is more important than speed of access.

Step

1. Enter the following command:

   ```
   cifs shares -change share_name -vscan
   ```

   share_name is the name of the share for which you want to turn on virus scanning.

The application performs a virus scan when clients open files on this share. The setting is persistent after reboot.

Note: You can set these share attributes for all CIFS user home directories by using `cifs.homedir` as the share name, for example, `cifs shares -change cifs.homedir -vscan`

Turning scanning off for read-only access

You can turn off virus scanning for files in a share if the files are restricted to read-only mode or speed of access is more important than safety.

About this task

Virus scanning for a share is turned on by default.

Step

1. Enter the following command:

   `cifs shares -change` ***`share_name`*** **`-novscanread`**

   `share_name` is the name of the share for which you want to turn off virus scanning.

   The application does not perform a virus scan when clients open files on this share for read access. The setting is persistent after reboot.

Turning scanning on for read-only access

You can turn on virus scanning for files in a share if safety is more important than speed of access.

Step

1. Enter the following command:

   `cifs shares -change` ***`share_name`*** **`-vscanread`**

   `share_name` is the name of the share for which you want to turn on virus scanning.

   The application performs a virus scan when clients open files on this share for read access. The setting is persistent after reboot.

Adding shares with virus scanning turned off

You can add a share with virus scanning turned off if the share will be used only by trusted users, the files are restricted to read-only mode, or speed of access is more important than safety.

About this task

For backup purposes, you can create two shares on the same directory: one share with scanning disabled and a share-level Access Control List (ACL) that allows access only to a backup account;

the other share available to normal users and with scanning enabled. Backup can be performed on the share with no virus scanning and improved performance, while normal users continue to access the data through the regular share and get virus protection.

Note: Virus scanning for a share is turned on by default.

Step

1. Enter the following command:

   cifs shares -add *share_name* /*path* -novscan

   share_name is the name of the share with virus scanning turned off that you want to create.

   path specifies where you want the share created.

   Data ONTAP creates a share with virus scanning turned off.

Adding shares with virus scanning turned off for read-only access

You can add a share with virus scanning turned off if the share will be used only by trusted users, the files are restricted to read-only mode, or speed of access is more important than safety.

About this task

For backup purposes, you can create two shares on the same directory: one share with scanning disabled and a share-level Access Control List (ACL) that allows access only to a backup account; the other share available to normal users and with scanning enabled. Backup can be performed on the share with no virus scanning and improved performance, while normal users continue to access the data through the regular share and get virus protection.

Note: Virus scanning for a share is turned on by default.

Step

1. Enter the following command:

   cifs shares -add *share_name* /*path* -novscanread

   share_name is the name of the share with virus scanning turned off that you want to create.

   path specifies where you want the share created.

   Data ONTAP creates a share with virus scanning turned off for read-only access.

Displaying the scanner list

You can determine if scanner clients are receiving requests and scanning files by looking at the scanner list.

Step

1. Enter the following command:

   `vscan scanners`

   The system outputs the names of and information about the virus-scanning clients in a display like the one shown in the following paragraph.

Example

```
Virus scanners(IP and Name) Connect time Reqs Fails
                            (dd:hh:mm)
-------------------------------------------------------------------
10.61.155.118  \\WIN2K-NAN  00:02:23      2    0
10.60.129.152  \\WIN2K-RTB  00:00:01      0    0
```

The *`Connect time`* field displays the length of time the scanner has been connected to the system.

The *`Reqs`* field displays the total number of requests sent by the system to that scanner.

The *`Fails`* field displays the number of requests that failed.

Primary virus scanner not listed

Sometimes, after configuring a primary virus scanner, it no longer services client requests and does not appear in the list when you run `vscan scanners` on the system CLI. However, the previous configuration can be observed in old AutoSupport messages. To resolve this problem, perform the following steps.

Steps

1. Ping the virus scanner to determine whether network connectivity exists between the storage system and the virus scanner.

   If connectivity exists, proceed to Step 2.

2. Determine whether the virus scanning software is properly installed and configured. If it is not functioning properly, see the virus scanner documentation or contact the manufacturer's technical support staff to identify and correct any problems. If there are no problems, proceed to Step 3.

3. Ensure that the virus scanning software service is started. If it is not, restart the virus scanning application on the virus scan server.

   Note: Even if the service appears to be started, it might be frozen. Restarting the service causes it to re-register with the storage system. The primary scanner is not configured on the secondary storage system side. Secondary virus scanners are configured on the secondary storage system only by using the `vscan scanner secondary_scanners` command.

Checking vscan information

You can quickly determine whether the virus scanner is on, how well the virus scanner is working, and what files extensions are scanned and not scanned.

Step

1. Enter the following command:

   `vscan`

   The system displays the on/off status of vscan, information about the virus-scanning clients, the list of extensions to scan, and the number of files scanned and number of scan failures in a display like the one shown in the following paragraph.

Example

```
systemA> vscan

Virus scanning is enabled.

Virus scanners(IP and Name)  Connect time   Reqs  Fails
                             (dd:hh:mm)
-----------------------------------------------------------------
10.61.155.118  \\WIN2K-NAN    00:02:23       2     0
10.60.129.152  \\WIN2K-RTB    00:00:01       0     0

List of extensions to scan: ??_,ASP,BAT,CDR,COM,CSC,DL?,DOC,DOT,EXE,
GMS,GZ?,HLP,HT?,IM?,INI,JS?,MD?,MPP,MPT,MSG,MSO,OCX,OLE,OV?,POT,PP?,
RTF,SCR,SHS,SMM,SYS,VBS,VS?,VXD,WBK,WPD,XL?,XML

List of extensions not to scan:

Number of files scanned: 28
Number of scan failures: 0
```

Note: The number of scan requests and failures shown in the table at the beginning of the output represents this connection session. The second set of numbers at the end of the output reflects the total scans and failures since vscan was turned on.

Setting and resetting the request timeout for a virus scan

You can change how long Data ONTAP should wait for a virus scan to finish before requesting the status of the scan. You might need to do this when you have a slow network or your virus scanner is slow, and you want to avoid the scan request from timing out prematurely.

About this task

The request is repeated as often as necessary until the scan is complete or the host gives up. By default, the virus scan timeout is 10 seconds.

Step

1. You can either set a new scan request timeout value or reset a value that you previously set back to the default value.

If you want to...	Then enter the following command...
Set a new scan request timeout value.	**vscan options timeout set *value*** *value* is a setting from 1 to 45 seconds. The recommended setting is between 8 and 12 seconds.
Reset the scan request timeout value back to the default value of 10 seconds	**vscan options timeout reset**

Allowing file access when the scan cannot be performed

You can specify that files can be accessed if a scanner is not available or if scan requests time out.

About this task

The default setting is that file access is denied if a successful scan cannot be performed, that is, the option is set to `on`. When this option is set to `off`, access is allowed even if a scan cannot be performed.

Step

1. Enter the following command:

   vscan options mandatory_scan [on|off]

Controlling vFiler unit usage of host system's virus scanners

Depending on your security concerns, you can either have each vFiler unit register with the virus scanner or allow the vFiler units to use the host system's virus scanners.

About this task

By default, vFiler units can scan files using the virus scanners that are connected to the host system. For more information about vFiler units, see the *Data ONTAP MultiStore Management Guide for 7-Mode*.

If you have several vFiler units owned by separate departments, and you have security concerns, then you might want to have the virus scanner register with each vFiler unit. For more information, see the documentation for the virus-scanning software.

You can have the virus scanner register with only the physical system. In this way, only the system authenticates the virus scanner.

Step

1. You can register a virus scanner in one of two ways. As required, choose one of the actions from the following table.

If you want to...	Then from the vFiler context, enter the following command...
Specify a configuration for each vFiler unit	`vscan options use_host_scanners off`
Specify a configuration that applies to all vFiler units in the system	`vscan options use_host_scanners on`

Note: The `vscan options` command is not supported on the default vFiler unit.

Checking the status of virus-scanning options

You can quickly determine whether the virus-scanning options you want to use are set to the values you want.

Step

1. Enter the following command:

   `vscan options`

The storage system outputs the state of the virus-scanning options in a display like the following:

```
vscan options mandatory_scan     on
vscan options timeout:           12 sec
vscan options use_host_scanners  on
```

Note: The `use_host_scanners` option applies only to vFiler units and is displayed only if the `vscan options` command was run on a vFiler unit.

Stopping a virus scanner session

You can stop a scanner session if you have to terminate CIFS on a storage system or if you upgrade your antivirus program to a new version.

Step

1. Enter the following command:

   vscan scanners stop *scanner_IP*

   scanner_IP is the IP address of the virus scanner you want to stop.

Resetting the scanned files cache

If you have a new virus definitions file, you might want to clear the cache and rescan the files that were scanned using an old virus definitions file.

About this task

Data ONTAP caches information about previously scanned files to avoid rescanning those files.

Step

1. Enter the following command:

   vscan reset

Enabling virus scan messages to CIFS clients

You can send explicit virus-warning messages to inform CIFS clients, if the virus scanner detects a threat within a file that a CIFS client is trying to access.

About this task

If this feature is not enabled, CIFS clients attempting to access virus-infected files that have been detected by virus scanning will simply receive a general `file unavailable` message.

Step

1. Enter the following command:

   `vscan options` ***`client_msgbox`*** **`{on|off}`**

   `on` enables the display of virus warning messages.

   `off` disables the display of virus warning messages.

Resolving virus scan server connectivity issues

If the virus scan server disconnects often and displays an error message, you can try to resolve the issue by using certain checks.

Symptoms: Any one of the following symptoms is observed:

- The virus scan server is unable to scan a file on the storage system.
- The virus scan server disconnects and reconnects after a brief period of time.
- The virus scan server disconnects and reconnects often with port 139 error.

You can identify these symptoms by tracking the vscan (virus scanner) statistic counters. To do this, use the following commands.

- **`priv set advanced`**
- **`stats show vscan_stats`**
- **`stats show vscan_server_stat`**

Cause of this problem:

- NetBIOS over TCP is not enabled on the server side.
- The server service is stopped.

Solution:

1. Enable NetBIOS over TCP on the server.
   For more information, see the chapter on NetBIOS over TCP/IP in the Microsoft TechNet library
2. Make the server service automatic.
3. Start the service.

Related information

Microsoft TechNet Library: technet.microsoft.com/en-us/library/bb727013.aspx

Glossary

The following terms are used for Data ONTAP features:

Access control list (ACL)

A list that contains the users' or groups' access rights to each share.

active file system

A file system in use, excluding its Snapshot copies.

aggregate

A grouping of physical storage resources (disks or array LUNs) that provides storage to volumes associated with the aggregate. Aggregates provide the ability to control the RAID configuration for all associated volumes.

blocking factor

In a tape backup operation, the number of tape blocks that are transferred in each write operation.

CIFS

See *Common Internet File System (CIFS)*.

Common Internet File System (CIFS)

Microsoft's file-sharing networking protocol that evolved from SMB.

console

The physical or virtual terminal that is used to monitor and control a storage system.

DNS

See *Domain Name System (DNS)*.

Domain Name System (DNS)

An Internet service for finding computers on an IP network.

dump path

The entity that specifies one volume, qtree, or subtree to back up.

file mark

The data on a tape that signals the boundaries of a tape file.

HTTP

Hypertext Transfer Protocol.

increment chain

A series of incremental backups of the same path.

inode

A data structure containing information about files on a storage system and in a UNIX file system.

local tape device

A program-based functionality associated with a tape drive that is directly attached to a storage system that is performing a tape operation.

Message Digest 5 (MD5)

An algorithm for checksum generation for use in security applications.

mirror

Mirror protection is the periodic exact mirroring of all volume data (both active and protected) from a NetApp source storage system to a NetApp destination storage system. If data in the source storage system is lost or made unavailable (for example if the source system is damaged), then that same data can quickly be made available from the destination mirror site. Mirror operations are employed from primary to secondary storage and from secondary to tertiary storage, in circumstances where secure mirroring of that data, and in event of breakdown at the source site, quick availability of that data from a second site might be required. Mirror protection is based on NetApp Volume SnapMirror technology.

NDMP

Network Data Management Protocol. A protocol that allows storage systems to communicate with backup applications and provides capabilities for controlling the robotics of multiple tape backup devices.

NFS

Network File System.

NIS

Network Information Service, formerly called Yellow Pages. An administrative database for networks.

NVFAIL

Software that warns you of compromised database validity and automatically renames the database so that it does not restart automatically.

NVRAM

nonvolatile random-access memory.

NVRAM cache

Nonvolatile RAM in a storage system, used for logging incoming write data and NFS requests. Improves system performance and prevents loss of data in case of a storage system or power failure.

Open Systems platform

A system, such as a server running Solaris, HP-UX, or Windows, whose data can be backed up to a SnapVault secondary storage system.

Open Systems SnapVault

The software application that provides a disk-to-disk data protection solution that takes advantage of the NetApp SnapVault technology to protect data.

plex

A physical copy of a file system. An unmirrored volume has one plex; a mirrored volume has two identical plexes.

primary storage system

A system whose data is to be backed up by SnapVault.

quota

A limit placed on a file system that restricts disk space usage by files with a given User ID (UID) or group ID (GID).

qtree

A special subdirectory of the root of a volume that acts as a virtual subvolume with special attributes.

RAID

Redundant Array of Independent Disks. A technique that protects against disk failure by computing parity information based on the contents of all the disks in a storage array. Storage systems use either RAID4, which stores all parity information on a single disk, or RAID-DP, which stores all parity information on two disks.

Remote Shell

A program that enables a user on one system to execute a program on another system. Remote Shell connections are usually not interactive.

root volume

The volume that contains information that controls the entire storage system, usually in the /etc/rc file.

secondary storage system

A storage system to which data is backed up by SnapVault.

share

A directory or directory structure that has been made available to network users and can be mapped to a drive letter on a CIFS client. Also known as a *CIFS share*.

SnapMirror

A product that performs automated file system replication of a volume onto a separate disk or storage system.

SnapRestore

Software that restores an entire volume to the state recorded in a previously taken Snapshot copy.

Snapshot copy

An online, read-only copy of an entire file system that protects against accidental deletions or modifications of files without duplicating file contents. Snapshot copies enable users to restore files and to back up the storage system to tape while the storage system is in use.

Snapshot reserve

The portion of a volume's disk space that is reserved for Snapshot copies.

source storage system

The storage system from which you are replicating data.

subtree

A directory in a volume or qtree.

tape block

1,024 bytes of data.

tape device

A specific functionality of a physical tape drive that you create by specifying information in the tape device name when you install a tape drive or tape stacker.

tape file

Data on a tape delimited by file marks.

tape stacker

Hardware that can access tape cartridges from a stack.

volume

A file system.

volume copy

A way of copying both data in the active file system and data in Snapshot copies from one volume to another.

Copyright information

Trademark information

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

D

E

F

G

H

I

T

U

V