Data ONTAP® 7.3
Archive and Compliance Management Guide

NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089 U.S.A.
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: http://www.netapp.com

Part number: 210-04827_A0
Updated for Data ONTAP 7.3.3 on 4 March 2010

Contents

Copyright information

Trademark information

About this guide

You can use your product more effectively when you understand this document's intended audience and the conventions that this document uses to present information.

This guide describes how to archive and protect data for compliance purposes.

Next topics

Audience

This document is written with certain assumptions about your technical knowledge and experience.

This guide is intended for system administrators familiar with operating systems, such as UNIX, Windows 95, Windows NT, and Windows 2000.

You should be familiar with how to configure storage system and how the NFS, CIFS, and HTTP protocols are used for file sharing or transfers. This guide does not cover basic system or network administration topics, such as IP addressing, routing, and network topology; it emphasizes the archival and compliance characteristics of the storage system.

Accessing Data ONTAP man pages

You can use the Data ONTAP manual (man) pages to access technical information.

About this task

Data ONTAP manual pages are available for the following types of information. They are grouped into sections according to standard UNIX naming conventions.

Types of information	Man page section
Commands	1

Types of information	Man page section
Special files	4
File formats and conventions	5
System management and services	8

Step

1. View man pages in the following ways:
   - Enter the following command at the storage system command line:

     `man` ***`command_or_file_name`***
   - Click the manual pages button on the main Data ONTAP navigational page in the FilerView user interface.
   - Use the *Commands: Manual Page Reference*, Volumes 1 and 2 (which can be downloaded or ordered through the NOW site).

     Note: All Data ONTAP man pages are stored on the storage system in files whose names are prefixed with the string "na_" to distinguish them from client man pages. The prefixed names are used to distinguish storage system man pages from other man pages and sometimes appear in the NAME field of the man page, but the prefixes are not part of the command, file, or services.

Terminology

To understand the concepts in this document, you might need to know how certain terms are used.

Storage terms

array LUN Refers to storage that third-party storage arrays provide to storage systems running Data ONTAP software. One array LUN is the equivalent of one disk on a native disk shelf.

LUN (logical unit number) Refers to a logical unit of storage identified by a number.

native disk Refers to a disk that is sold as local storage for storage systems that run Data ONTAP software.

native disk shelf Refers to a disk shelf that is sold as local storage for storage systems that run Data ONTAP software.

storage controller	Refers to the component of a storage system that runs the Data ONTAP operating system and controls its disk subsystem. Storage controllers are also sometimes called *controllers*, *storage appliances*, *appliances*, *storage engines*, *heads*, *CPU modules*, or *controller modules*.
storage system	Refers to the hardware device running Data ONTAP that receives data from and sends data to native disk shelves, third-party storage, or both. Storage systems that run Data ONTAP are sometimes referred to as *filers*, *appliances*, *storage appliances*, *V-Series systems*, or *systems*.
third-party storage	Refers to the back-end storage arrays, such as IBM, Hitachi Data Systems, and HP, that provide storage for storage systems running Data ONTAP.

Cluster and high-availability terms

active/active configuration	In the Data ONTAP 7.2 and 7.3 release families, refers to a pair of storage systems (sometimes called *nodes*) configured to serve data for each other if one of the two systems stops functioning. Also sometimes referred to as *active/active pairs*. In the Data ONTAP 7.1 release family and earlier releases, this functionality is referred to as a *cluster*.
cluster	In the Data ONTAP 7.1 release family and earlier releases, refers to a pair of storage systems (sometimes called *nodes*) configured to serve data for each other if one of the two systems stops functioning. In the Data ONTAP 7.3 and 7.2 release families, this functionality is referred to as an *active/active configuration*.

Where to enter commands

You can use your product more effectively when you understand how this document uses command conventions to present information.

You can perform common administrator tasks in one or more of the following ways:

- You can enter commands either at the system console or from any client computer that can obtain access to the storage system using a Telnet or Secure Shell (SSH) session.
  In examples that illustrate command execution, the command syntax and output shown might differ from what you enter or see displayed, depending on your version of the operating system.
- You can use the FilerView graphical user interface.

- You can enter Windows, ESX, HP-UX, AIX, Linux, and Solaris commands at the applicable client console.
  In examples that illustrate command execution, the command syntax and output shown might differ from what you enter or see displayed, depending on your version of the operating system.
- You can use the client graphical user interface.
  Your product documentation provides details about how to use the graphical user interface.

- You can enter commands either at the switch console or from any client that can obtain access to the switch using a Telnet session.
  In examples that illustrate command execution, the command syntax and output shown might differ from what you enter or see displayed, depending on your version of the operating system.

Keyboard and formatting conventions

You can use your product more effectively when you understand how this document uses keyboard and formatting conventions to present information.

Keyboard conventions

Convention	What it means
The NOW site	Refers to *NetApp On the Web at http://now.netapp.com/*.
Enter, *enter*	• Used to refer to the key that generates a carriage return; the key is named Return on some keyboards. • Used to mean pressing one or more keys on the keyboard and then pressing the Enter key, or clicking in a field in a graphical interface and then typing information into the field.
hyphen (-)	Used to separate individual keys. For example, Ctrl-D means holding down the Ctrl key while pressing the D key.
type	Used to mean pressing one or more keys on the keyboard.

Formatting conventions

Convention	What it means
Italic font	• Words or characters that require special attention. • Placeholders for information that you must supply. For example, if the guide says to enter the `arp -d` *`hostname`* command, you enter the characters "arp -d" followed by the actual name of the host. • Book titles in cross-references.
`Monospaced` font	• Command names, option names, keywords, and daemon names. • Information displayed on the system console or other computer monitors. • Contents of files. • File, path, and directory names.

Convention	What it means
`Bold monospaced` font	Words or characters you type. What you type is always shown in lowercase letters, unless your program is case-sensitive and uppercase letters are necessary for it to work properly.

Special messages

This document might contain the following types of messages to alert you to conditions that you need to be aware of.

Note: A note contains important information that helps you install or operate the system efficiently.

Attention: An attention notice contains instructions that you must follow to avoid a system crash, loss of data, or damage to the equipment.

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback.

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments by e-mail to *doccomments@netapp.com*. To help us direct your comments to the correct division, include in the subject line the name of your product and the applicable operating system. For example, *FAS6070—Data ONTAP 7.3*, or *Host Utilities—Solaris*, or *Operations Manager 3.8—Windows*.

What SnapLock is

SnapLock is an alternative to the traditional optical "write once, read many" (WORM) data. SnapLock is used for the storage of read-only WORM data.

SnapLock is a license-based, disk-based, open-protocol feature that works with application software to administer non-rewritable storage of data. The primary objective of this Data ONTAP feature is to provide storage-enforced WORM and retention functionality by using open file protocols such as CIFS and NFS. SnapLock can be deployed for protecting data in strict regulatory environments in such a way that even the storage administrator is considered an untrusted party. An example of such an environment is the broker or dealer market that is regulated by the U.S. Securities and Exchange Commission (SEC) rule 240.17a-4. Alternative configurations of SnapLock can be deployed for unregulated or more flexible regulated environments.

SnapLock provides special purpose volumes in which files can be stored and committed to a non-erasable, non-rewritable state either forever or for a designated retention period. SnapLock allows this retention to be performed at the granularity of individual files through standard open file protocols such as CIFS and NFS (versions 2, 3, and 4). The retention of these files is enforced by Data ONTAP, through which all the file protocols or administrative access to the data must pass.

Next topics

Related concepts

What SnapLock volumes are

SnapLock volumes are of two types—SnapLock Compliance volume and SnapLock Enterprise volume.

The SnapLock Compliance volume provides WORM protection for files and also restricts the storage administrator's ability to perform any operations that might modify or erase retained WORM records. SnapLock volumes use a secure ComplianceClock to enforce the retention periods. Use SnapLock Compliance in strictly regulated environments that require information to be retained for a specified period of time, such as those governed by SEC Rule 17a-4.

The SnapLock Enterprise volume provides WORM protection for files with a trusted model of operation to manage the systems. SnapLock Enterprise allows the administrator to destroy SnapLock Enterprise volumes before all locked files on the volume reach their expiry date.

You cannot use a SnapLock volume as a regular volume for data storage. In most cases, SnapLock volumes behave identically to regular volumes, but there are some specific and critical differences as far as functionality and administration are concerned that make the SnapLock volume unsuitable for use as regular volumes. Specific examples include the following:

- Renaming directories on SnapLock volumes are not allowed.
- Transition of the file attribute from writable to read-only commits a file to the WORM state.
- Administrative interfaces are restricted (drastically for SnapLock Compliance volumes).

Related concepts

What ComplianceClock is on page 26

How SnapLock works

The WORM data on SnapLock volumes is administered in the same way as data on regular (non-WORM) volumes. SnapLock volumes operate in WORM mode and support standard file system semantics. You can create data on a SnapLock volume and commit it to the WORM state by transitioning the file from a writable state to a read-only state.

Marking an active writable file as read-only on a SnapLock volume commits the data to WORM. When a file is committed to WORM, it cannot be altered or deleted by applications, users, or administrators until the file retention date is reached. The exception is in SnapLock Enterprise volumes, where you can delete a file before it reaches the retention date by using the privileged delete feature.

The data that is committed to the WORM state on a SnapLock volume cannot be changed or deleted before its retention date. However, you can change or delete the empty directories and files that are not committed to a WORM state. Directories do not behave any differently than they would on regular volumes, with the exception that they cannot be renamed or moved once created. It is a requirement for regulatory compliance that WORM data be not only non-erasable and non-rewritable, but it must also be locked down in the same location at which it was created. In the case

of WORM implementation, this means that the directory path to WORM files must be locked down and should never change.

In Data ONTAP 7.0 and later, WORM files can be deleted after their retention dates have been reached. The retention date on a WORM file is set when the file is committed to the WORM state, but it can be extended at any time. The retention period can never be shortened for any WORM file.

Issues in reverting to Data ONTAP 7.3 if you have SnapLock volumes

SnapLock is disabled in the first release of Data ONTAP 7.3. Therefore, if you revert from a later release in the Data ONTAP 7.3 release family, any SnapLock Compliance or SnapLock Enterprise license that has been installed on the later release will either be disabled or your storage system will halt.

If your storage system contains SnapLock Compliance or Enterprise volumes or aggregates and you attempt to revert to Data ONTAP 7.3, the system halts. The SnapLock disks that cause the system to halt are listed on the console. Following is an example of such a message on the console:

```
SnapLock disk : 0d.40 Use fcadmin device_map for shelf and slot info
SnapLock disk : 0d.43 Use fcadmin device_map for shelf and slot info
SnapLock disk : 0d.42 Use fcadmin device_map for shelf and slot info
This release does not support SnapLock.  Halting the system !!!  To
recover - boot with a release that supports SnapLock or unplug the
SnapLock disks.
```

If you revert to Data ONTAP 7.3 with an installed SnapLock Compliance or Enterprise license without any SnapLock Compliance or Enterprise volumes or aggregates on the system, the system does not halt, but the SnapLock license is disabled. In such a scenario, the license is displayed if you use the `license` command. However, all operations that require a SnapLock license fail. Therefore, you cannot create new SnapLock volumes on Data ONTAP 7.3. The error message `snaplock.unsupported.version` is generated when you boot the storage system.

If you connect disks that were part of a SnapLock Compliance or Enterprise aggregate into a system running Data ONTAP 7.3, the system halts and displays the following message on the console:

```
SnapLock disk : 0d.40, Shelf : 2, Slot : 8
 SnapLock disk : 0d.43, Shelf : 2, Slot :11
 SnapLock disk : 0d.42, Shelf : 2, Slot: 10
 This release does not support SnapLock.
 Halting the system !!!
 To recover - boot with a release that supports SnapLock or unplug the
SnapLock disks
```

Starting in Data ONTAP 7.3.1, SnapLock is supported. Therefore, if you are upgrading to Data ONTAP 7.3.1 or a later release in the 7.3 release family, the system will not halt and you can install the SnapLock Compliance or SnapLock Enterprise license.

Recovering from the halt caused by SnapLock

If the storage system halts because you have SnapLock volumes on the system and you attempted to revert to Data ONTAP 7.3, you can recover from the halt by using a release that supports SnapLock or by disconnecting disks that have SnapLock installed.

Step

1. Recover from the halt in Data ONTAP by rebooting or disconnecting disks (if rebooting does not work):

If...	Then...
You have a release that supports SnapLock	Boot using that release
You have a backup release that supports SnapLock	Boot using the following boot menu command: `boot_backup`
Booting does not work	Disconnect the SnapLock disks. To find the location of the SnapLock disks, boot in maintenance mode and use the following command: `fcadmin device_map`

Hardware platforms supported for SnapLock

SnapLock is exclusively a licensed feature of Data ONTAP and is supported on almost all NetApp hardware platforms. V-Series supports SnapLock Enterprise with both native and third-party storage, however, SnapLock Compliance is supported only on native disks. SnapLock is not a "software only" solution; it is part of an integrated hardware and software solution.

How to license SnapLock

SnapLock is a persistent property of both the SnapLock volumes and the files on SnapLock volumes. It is strictly enforced regardless of the state of the licensing. SnapLock requires a separate license. You need a license only for the creation of new SnapLock volumes and for the committing of files to the WORM state on SnapLock volumes.

However, even if a license is uninstalled, the existing SnapLock volumes are accessible for general reading and writing.

You can unlicense SnapLock, but files that are already in SnapLock state retain their immutable WORM protection. The only disadvantage of removing the SnapLock license is that you can no longer put new files into the WORM state or create new SnapLock volumes.

Licensing SnapLock functionality

You must license SnapLock Compliance, SnapLock Enterprise, or both before you can use the SnapLock feature.

Steps

1. Check the existence of the SnapLock licenses on the storage system by entering the following command:

   `license`

   Following is the output:

   ```
   snaplock site XXXXXXX
   snaplock_enterprise site XXXXXXX
   ```

2. Install a license for SnapLock Compliance, SnapLock Enterprise, or both.

   - To use the SnapLock Compliance feature, enter the following command:

     `license add` ***`snaplock_license`***

     where *`snaplock_license`* is the actual site license.

     The following output is displayed:

     ```
     A snaplock site license has been installed.
     SnapLock(tm) Compliance enabled.
     Wed Jan 24 00:23:35 GMT [rc:notice]: snaplock licensed
     ```

   - To use the SnapLock Enterprise feature, enter the following command:

     `license add` ***`snaplock_enterprise`***

     where *`snaplock_enterprise`* is the actual site license.

     The following output is displayed:

     ```
     A snaplock_enterprise site license has been installed.
     SnapLock Enterprise enabled.
     ```

SnapLock and AutoSupport messages

If you enable the AutoSupport feature, the storage system sends AutoSupport messages to technical support.

AutoSupport event messages include event and log-level descriptions as well as SnapLock volume state and options. The messages also display the value of ComplianceClock and the expiry date of all volumes on the storage system.

Note: AutoSupport messages do not include options such as a privileged delete setting.

Creating SnapLock volumes

You can create SnapLock traditional volumes or SnapLock FlexVol volumes. Before you create a SnapLock FlexVol volume, however, you must create an aggregate with SnapLock as an attribute of that aggregate.

Next topics

Creating SnapLock traditional volumes on page 24
Creating SnapLock aggregates and their flexible volumes on page 25

Creating SnapLock traditional volumes

You can create SnapLock traditional volumes to retain compliance or archival data.

Before you begin

Ensure that you have licensed SnapLock Compliance, SnapLock Enterprise, or both.

Steps

1. To initialize ComplianceClock, enter the following command:

   `date -c initialize`

2. To create a SnapLock traditional volume, enter the following command:

   `vol create` ***`trad_vol`*** **`-v -L`** ***`snaplock_type ndisks`*****`[@disksize]`**

   `trad_vol` is the new traditional volume name.

   `ndisks` is the number of disks.

   `disksize` is the size of the disk.

   `snaplock_type` can be `compliance` or `enterprise`. If only one SnapLock type is enabled, you need not specify the attribute; however, you must use the attribute if you have enabled both SnapLock version licenses.

3. Verify that the newly created SnapLock volume exists by entering the following command:

   `vol status`

   Example

```
sys1> vol status
      Volume State     Status        Options
      vol0 online      raid4, trad   root
      wormvol online   raid4, trad   no_atime_update=on,
                                     snaplock_compliance
```

Creating SnapLock aggregates and their flexible volumes

If you are using the FlexVol feature of Data ONTAP, you can create SnapLock aggregates and flexible volumes to retain compliance or archival data.

Before you begin

Ensure that you have licensed SnapLock Compliance, SnapLock Enterprise, or both. In addition, you must initialize the ComplianceClock before you can create the SnapLock aggregates and their flexible volumes.

About this task

SnapLock is an attribute of the aggregate; therefore, the volume contained in that aggregate inherits the aggregate's SnapLock attributes. Every FlexVol volume created in a SnapLock aggregate is, by definition, a SnapLock volume. A SnapLock Compliance aggregate contains only SnapLock Compliance flexible volumes, and a SnapLock Enterprise aggregate contains only SnapLock Enterprise flexible volumes.

Steps

1. To initialize ComplianceClock, enter the following command:

   `date -c initialize`

2. To create a SnapLock aggregate, enter the following command:

   `aggr create` ***`aggrname`*** **`-L`** ***`snaplock_type ndisks`*****`[@`*****`disksize`*****`]`**

   `aggrname` is the new SnapLock aggregate name.

   `snaplock_type` can be `Compliance` or `Enterprise`. If only one SnapLock version is licensed, you need not specify the attribute; however, you must specify the attribute if both the SnapLock versions are licensed.

   `ndisks` is the number of disks.

   `disksize` is the size of the disk, for example, 72 GB.

3. Verify that the newly created SnapLock aggregate exists by entering the following command:

   `aggr status`

   Example

```
aggr status aggr1
      Aggr State      Status           Options
      aggr1 online  raid_dp, aggr      snaplock_enterprise
      Volumes: vol1
```

4. To create a SnapLock FlexVol volume, enter the following command:

   `vol create wormvol` ***`aggrname size`*****`[k|m|g|t]`**

SnapLock Compliance write verification option

You can use the SnapLock Compliance write verification option when each write operation to the disk media must be immediately read back and verified for integrity.

This feature adds a level of data integrity checking to the already robust data protection and resiliency features in Data ONTAP. Although SnapLock write verification can affect performance, the performance impact should not affect archival throughput.

SnapLock write verification is applicable only for SnapLock Compliance volumes. The SnapLock write verification option provides limited benefit beyond the advanced, high-performance data protection and integrity features already provided by nonvolatile RAM (NVRAM), checksums, RAID scrubs, media scans, and double-parity RAID.

Using the SnapLock Compliance write verification option

You enable the SnapLock write verification option by using the `snaplock.compliance.write_verify` option.

Step

1. To enable SnapLock write verification, enter the following command:

   `options snaplock.compliance.write_verify on`

How SnapLock Compliance meets WORM data requirements

SnapLock Compliance prevents anyone, including the system administrator, from destroying or altering files, volumes, or aggregates before they reach their expiry date.

To enforce WORM data retention, you require a secure time base, which ensures that the retained data cannot be deleted prematurely by changing the regular clock of the storage system.

SnapLock Compliance meets these requirements by using the secure ComplianceClock feature. ComplianceClock is implemented in software and runs independently of the system clock.

What ComplianceClock is

ComplianceClock is a secure time base that prevents compliant data from being tampered with. ComplianceClock makes it impossible to change the system clock to prematurely alter or remove the compliant data.

You can view ComplianceClock by using the `date -c` command, and you can initialize it (once only) by using the `date -c initialize` command.

The ComplianceClock is stored on all the volumes of a storage system on which ComplianceClock is initialized. There is also a central Data ONTAP ComplianceClock value, which is the primary reference point. Data ONTAP periodically updates the ComplianceClock. The ComplianceClock value is also placed in qtrees that are replicated to provide synchronization between the source and destination storage systems.

Before Data ONTAP 7.1, ComplianceClock drifted toward the system clock at a rate of one day per year. In Data ONTAP 7.1 and later, the drift rate is one week per year. However, there is no way to alter ComplianceClock manually.

How ComplianceClock is used

You use ComplianceClock for performing various SnapLock operations.

The following table describes these operations.

Operations	Description
Checking the expiry status of a WORM file	To check whether a WORM file has expired, you compare the retention time of the file with the ComplianceClock. If ComplianceClock is not set or is less than the retention time, the file has not expired.
Committing a file to WORM (ctime)	When you commit a file to WORM state, ComplianceClock or system clock is written to the `ctime` field of the file. If the ComplianceClock is initialized in the system, the ComplianceClock value is set in the `ctime` field. If the ComplianceClock is not initialized in the system (SnapLock Enterprise volume creation do not necessarily need the ComplianceClock to be initialized on the system), the system clock is written to the `ctime` field. **Note:** The `ctime` value is different than `atime` value which is used as the retention date of the file.
Destroying a SnapLock Compliance volume	To check whether a SnapLock Compliance volume can be destroyed, you compare its destroy time (equivalent to retention time) with ComplianceClock. If ComplianceClock is not set or is less than the volume destroy time, the volume cannot be destroyed.
Creating a SnapLock Compliance volume	You must initialize ComplianceClock before creating a SnapLock Compliance volume.
SnapLock for SnapVault	The ComplianceClock must be initialized before you can commit Snapshot copies to WORM.

Operations	Description
Volume SnapMirror	Volume SnapMirror works at the block level. Therefore, ComplianceClock value is automatically transferred from the volume SnapMirror source volume to the volume SnapMirror destination volume.
Volume SnapMirror to a SnapLock Compliance volume	In Data ONTAP 7.2.5, Data ONTAP 7.3.1, and later releases, ComplianceClock must be initialized on the destination storage system before you can do a volume SnapMirror transfer that has a SnapLock Compliance volume as the destination.
Qtree SnapMirror	In Data ONTAP 7.2.5, Data ONTAP 7.3.1, and later releases, the qtree SnapMirror protocol is upgraded to transfer ComplianceClock from the source to the destination. This adjusts ComplianceClock of the destination storage system when the qtree SnapMirror relationship is broken using the `snapmirror break` command.

Initializing the ComplianceClock feature

You must initialize the ComplianceClock feature if you want to enforce WORM data retention on SnapLock volumes. If ComplianceClock is not initialized, you cannot delete the files after their expiry date, even on SnapLock Enterprise volumes.

Before you begin

Ensure that a SnapLock Compliance or SnapLock Enterprise license is installed on the storage system. In addition, ensure that the system time and time zone are set correctly.

About this task

Data ONTAP checks ComplianceClock when determining or enforcing retention periods.

For example, Data ONTAP checks ComplianceClock when determining whether a file has expired; to enforce the volume's minimum, maximum, or default retention period when committed to the WORM state; or to determine whether a volume’s destroy date has expired and the volume can therefore be destroyed.

Attention: ComplianceClock must be initialized before you create SnapLock Compliance volumes or aggregates. You can initialize ComplianceClock only once. You should exercise extreme care when setting ComplianceClock. Ensure that the system clock is as close to the accurate time as possible. The initial setting of ComplianceClock is based on the current system clock.

Steps

1. Initialize ComplianceClock by using the following command:

   `date -c initialize`

   The system prompts you to confirm the current local time and to verify that you want to initialize ComplianceClock.

2. Confirm that the system clock is correct and that you want to initialize ComplianceClock.

ComplianceClock initialization message

```
*** WARNING: YOU ARE INITIALIZING THE SECURE COMPLIANCE CLOCK ***

You are about to initialize the secure Compliance Clock of this
system to the current value of the system clock. This procedure
can be performed ONLY ONCE on this system so you should ensure
that the system time is set correctly before proceeding.

The current local system time is: Wed Feb 4 23:38:58 GMT 2007

Is the current local system time correct? y
Are you REALLY sure you want initialize the Compliance Clock? y

Compliance Clock: Wed Feb 4 23:39:27 GMT 2007
```

Viewing the ComplianceClock time

You can view the ComplianceClock time to verify that it is synchronized with the system clock.

Step

1. Enter the following command:

   `date -c`

   Example

   ```
   date -c
   Compliance Clock: Wed Feb 4 23:42:39 GMT 2007
   ```

How ComplianceClock stays synchronized with the system clock

The ComplianceClock, which is implemented in software, can lag behind the system clock and must realign itself with the system clock periodically.

The ComplianceClock might lag behind the system clock in the following circumstances:

- The entire storage system is down (Data ONTAP is not running).
- A volume is taken offline and then brought online again.
- A volume is taken offline and the system is rebooted.
- One of several SnapMirror configurations exist:
  - The destination volume of the SnapMirror relationship is removed from the SnapMirror schedule. However, the SnapMirror relationship is not broken, and therefore the destination volume is not brought online. The ComplianceClock can leap backward each time the system is rebooted.
  - The SnapMirror destination volume is updated relatively infrequently (for example, monthly), and the system is rebooted well into the period between SnapMirror updates.

Note: You can minimize the time by which ComplianceClock lags behind the system clock by limiting the amount of time your system is down or your volumes are offline.

Some SnapMirror configurations might cause ComplianceClock to lag behind the system clock. Breaking a volume SnapMirror relationship to a WORM volume can cause ComplianceClock to lag behind the system clock contained in a volume SnapMirror destination. Therefore, you must perform a volume SnapMirror update operation before breaking the SnapMirror relationship.

The ComplianceClock is programmed to align with the system clock at a particular rate, thereby making up for small discrepancies automatically. The rate at which ComplianceClock aligns with the system clock depends on the version of Data ONTAP your system is running:

- For releases earlier than Data ONTAP 7.1, the rate is one day per year.
- For Data ONTAP 7.1 and later, the rate is seven days per year.

How a WORM file works

The abbreviation *WORM* stands for "write once, read many." A WORM file has the property that data can be written only once to any area of the file and can never be overwritten or erased before the retention period expires.

In some cases, this feature is a property of the physical media (such as with WORM optical platters). In other cases, the physical media is rewritable, but the integrated hardware and software code controlling access to the media prevent such overwrites (such as with WORM magnetic tape and disk-based SnapLock).

In regulated environments, the regulating body, such as the SEC, normally requires that all business records that fall under the umbrella of the regulations be archived on WORM media. This is to maintain a non-erasable and non-rewritable electronic paper trail that can be used for discovery or investigation.

After a file on a SnapLock volume is committed to the WORM state, there is only one attribute that can be modified—the retention date of the file can be extended. The retention period can never be shortened.

Next topics

How to manage WORM data on page 31
How SnapLock automatically commits files to WORM state on page 35

How to manage WORM data

After creating SnapLock volumes, you can transition a file to the WORM state and set the retention date on the file, determine whether a file is in the WORM state, or extend the retention date of a WORM file.

Next topics

Transitioning data to the WORM state and setting the retention date on page 32
Extending the retention date of a WORM file on page 32
Determining the WORM status of a file on page 33
What the WORM append file is on page 34
Creating a WORM append file on page 34

Transitioning data to the WORM state and setting the retention date

You must transition a file from the writable to read-only state in the SnapLock volume for the file to be committed to a WORM state. You can transition data to the WORM state interactively or automatically. In addition, you must set the retention date for the file.

Steps

1. Set the retention date using the command or program available to the file access protocol (CIFS, NFS, and so on) and client operating system you are using.

   Example

   In a UNIX shell, use the following commands to set the `document.txt` file with a retention date of 21 November 2020:

   ```
   touch -a -t 202011210600 document.txt
   ```

2. Commit the `document.txt` file to the WORM state by using the following command:

   ```
   chmod -w document.txt
   ```

   Note: The last accessed timestamp of the file at the time it is committed to the WORM state becomes its retention date unless it is limited by the minimum or maximum retention period of the SnapLock volume. If no retention date was set for the file, the default retention period of the SnapLock volume is used. The file's retention date is stored in the `atime` field. The option `no_atime_update` is always set to `on` for SnapLock volumes. Therefore, `atime` is not updated when files are accessed.

Related concepts

How the SnapLock volume retention period works on page 37

Extending the retention date of a WORM file

A WORM file can remain in WORM state longer than the current retention date if you extend the retention date of the WORM file beyond that of the current retention date.

About this task

For extending the retention date, you can use the command or program available to the file access protocol (CIFS, NFS, and so on) and client operating system you are using. You can extend the retention date interactively or programmatically.

Note: The SnapLock volume maximum or minimum retention period restrictions are not applied when the retention date of a WORM file is being extended.

Steps

1. To extend the retention date, use the command or program available to you.

   Example
   In a UNIX shell, you can use the following command to extend the `document.txt` file to a WORM state with a retention date of 15 December 2020:

   ```
   touch -a -t 202012150600 document.txt
   ```

   Note: The retention date of a WORM file can never be changed to earlier than its current setting.

2. To check the retention date on a file, enter the following command and note the value of the access time:

   `stat document.txt`

Related concepts

How the SnapLock volume retention period works on page 37

Determining the WORM status of a file

You can check whether the files placed in a SnapLock volume are committed to the WORM state.

About this task

To determine whether a file is in the WORM state, you must determine whether the file has transitioned from a writable state to a read-only state in the SnapLock volume. You can check the WORM state in one of the following ways:

- By using NFS/CIFS: Attempt to change the last accessed timestamp of the file to a date earlier than its current setting.

  This operation fails if the file is in the WORM state.

  Note: The new date must be later than 2003 or it will be taken as a wraparound date and the retention date can be extended.

- By using the fingerprint of the file. You can compute the fingerprint of the file by entering the following command:

  `file fingerprint -x /vol/sle0/file`

  A file has WORM protection if the file type is `worm` and the volume is a SnapLock volume. The file's expiry date is present in the `retention-time` and `formatted-retention-time` fields. You should check this date against the ComplianceClock time (specified in the `snaplock-compliance-clock` and `formatted-snaplock-compliance-clock` fields) to determine whether the file has expired.

What the WORM append file is

You can append data to a WORM append file. The WORM operations are done in 256-KB segments, so that when byte 256KB+1 is written, the previous segment automatically becomes immutable.

It is sometimes useful to have a SnapLock file that accepts appended data. An example would be a log file that you must retain. Rather than filling a log and copying it to a SnapLock file, you can create the file as a SnapLock file, append data to it, and then lock the file in place.

Creating a WORM append file

The WORM append file procedure is used to create the SnapLock for SnapVault Compliance Log. The WORM append file works with chunks of 256 KB. As each chunk is filled, it is locked. The file must be made read-only for it to be locked in its entirety.

Steps

1. To create a WORM file that can be appended, create a zero length file with the desired retention date inside a WORM volume.

   `touch -a -t` *`202012150600 file`*

2. Update the file access time to indicate the file's desired expiry date.
3. Make the file read-only.

   `chmod 444` *`file`*

   Note: This step is not deemed a compliance risk because there is no data in the file.

4. Make the file writable again.

   `chmod 644` *`file`*

   Note: Data is committed to WORM state in 256-KB chunks. Data does not have to be written sequentially to the active 256-KB chunk. When the data is written to byte n*256KB+1 of the file, the previous 256-KB segment becomes WORM and cannot be rewritten.

5. Start writing data to the file.

   `echo test data >>` *`file`*

   As each chunk is filled, it is locked. At this stage, you have a WORM appendable file.

6. When you have finished entering data into the file, make the file read-only again.

   `chmod 444` *`file`*

   The entire file is now in the WORM state and therefore is immutable.

How SnapLock automatically commits files to WORM state

The SnapLock autocommit feature, available in Data ONTAP 7.2 and later, is an adjustable timer that commits files to the WORM state if the file does not change during a specified delay period. You can automatically commit files on a SnapLock volume to the WORM state without using any other application. This way of committing files permits existing applications to be used with SnapLock without any change.

The minimum delay that can be specified is two hours. The autocommit operation does not take place instantly when the delay period ends; instead, the specified delay is the minimum amount of time that will pass before files become eligible for locking. The autocommit operation might not follow the delay period instantly because of the additional processing needed to commit multiple files to the WORM state.

Note: A file that is automatically committed to the WORM state gets a retention period equal to the volume's default retention period. However, if you explicitly change the `atime` value of the file before it is automatically committed, the file's retention date will be the `atime` value set by you.

When you use the autocommit feature to commit a file to WORM, the value (file's `ctime` value plus autocommit period) is compared with the value of the system clock. If the system clock value is greater, the file is committed to WORM.

Next topics

Setting the time delay before a file is automatically committed to the WORM state on page 35
Displaying the time delay before files are automatically committed on page 36

Setting the time delay before a file is automatically committed to the WORM state

You can set a global option that determines how long a file must remain unchanged in the SnapLock volume before it is committed to the WORM state.

Step

1. To set the autocommit time delay, enter the following command:

   `options snaplock.autocommit_period none|`*`count`*`(|h|d|m|y)`

   `autocommit_period` is the time delay in hours, days, months, or years.

   ### Example

   The following example sets the autocommit period to 24 days.

   `options snaplock.autocommit_period 24d`

   Note: You can turn off the autocommit feature by setting the time delay to `none`.

Examples of file access before and after an autocommit operation

The following example shows a file with read-write permissions and autocommit set.

```
$ stat *
  File: "new.txt"
  Size: 11 Blocks: 0 IO Block: 8192   Regular File
Device: 1eh/30d Inode: 1512 Links: 1
Access: (0664/-rw-rw-r--) Uid: (15883/  djv) Gid: ( 30/    engr)
```

The following example shows how permissions become read-only after the autocommit period has elapsed.

```
$ stat *
  File: "new.txt"
  Size: 11 Blocks: 0 IO Block: 8192   Regular File
Device: 1eh/30d Inode: 1512 Links: 1
Access: (0444/-r--r--r--) Uid: (15883/ djv) Gid: ( 30/ engr)
```

After the SnapLock retention period has elapsed, no changes will be made to the file. The file will continue to look as it does in this example. If you want to delete the file, you must manually change permissions to read-write after the retention period has ended. There is no automatic reset feature.

Displaying the time delay before files are automatically committed

You can view the time delay used to automatically commit files to WORM state. This information helps you to know the autocommit time delay, which you can change if the current delay is too short or too long.

About this task

The autocommit time delay option considers WORM appendable files the same as regular files and maintains the commitment to WORM state as long as the retention period has not elapsed.

Step

1. To display the autocommit time delay, enter the following command:

   `options snaplock.autocommit_period`

   Example
   If the autocommit time delay is 120 hours, the result is as follows:

   `snaplock.autocommit_period 120h`

What the retention period is

A retention date is the time period after which Data ONTAP permits the deletion of a write once, read many (WORM) file on a SnapLock volume. A retention period is the duration for which you can retain a file in a SnapLock volume.

Regulatory environments require that records be retained for a long period. Every record committed to the WORM state on a SnapLock volume can have an individual retention period associated with it. Data ONTAP enforces retention of these records until the retention period ends. After the retention period is over, the disposition of the records automatically changes to allow deletion, but not modification. Data ONTAP does not automatically delete any record. All records must be deleted using an application or manually.

SnapLock allows the retention period to be extended. However, it does not allow the retention period to be shortened.

How the SnapLock volume retention period works

After creating the SnapLock volume, you can set the volume retention periods. The SnapLock volume retention periods can be specified in days, months, or years.

Data ONTAP applies the retention period in a calendar-correct method. For example, if a WORM file or Snapshot copy created on 1 February has a retention period of one month, the retention period will expire on 1 March.

A SnapLock Compliance or Enterprise volume has three retention period values:

- Minimum retention period
- Maximum retention period
- Default retention period

The storage system checks for the minimum and maximum values when a file is committed to the WORM state. You can extend the retention date beyond the date set by the volume's maximum retention period value after the file has been locked initially.

The following table shows the default maximum and minimum retention periods for SnapLock Enterprise and SnapLock Compliance volumes.

Option	**SnapLock Enterprise default**	**SnapLock Compliance default**
`snaplock_minimum_period`	0	0
`snaplock_maximum_period`	30 years	30 years
`snaplock_default_period`	min (equal to snaplock_minimum_period)	max (equal to snaplock_maximum_period)

Next topics

Viewing the retention period of a volume

You can view the retention period of a volume.

Step

1. To view the retention period of a SnapLock volume, enter the following command:

   `vol status -w`

   Example

   Following is an example of command output that displays the expiry date of a SnapLock Compliance volume.

```
vol status -w
       Volume   Expiry Date
       ------   -----------
       vol_slc1 Wed Sep  8 03:31:34 GMT 2038
           vol0 -
       vol_sle1 Thu Sep  4 14:32:54 GMT 2008
       vol_reg1 -
```

   If the SnapLock volume does not have any retention time set, the command displays the expiry date as `none`.

```
vol status -w
       Volume  Expiry Date
       ------  -----------
         sle0  none
         vol0  -
         vol1  -
```

   Note: In a SnapLock Enterprise volume, the volume destroy time might be incorrect. This is because certain operations such as privileged delete and Single File SnapRestore (SFSR) might not update the volume's expiry date. However, you can destroy a SnapLock Enterprise volume irrespective of the volume's expiry date.

What the minimum retention period is

The minimum retention period is the shortest amount of time that a WORM file can be retained in a SnapLock volume. If the application sets the retention period shorter than the minimum retention

period, Data ONTAP adjusts the retention period of the file to the volume's minimum retention period.

The minimum retention period has the following characteristics:

- The existing files that are already in the WORM state are not affected by changes in this volume retention period.
- The minimum retention period takes precedence over a default period or over a retention date that was explicitly set by the application.
- Until you explicitly reconfigure the retention period, the minimum retention period is 0.

Setting the minimum retention period

You can set the SnapLock volume minimum retention period to ensure that applications or users do not assign retention periods that do not conform to the minimum retention period requirement.

Step

1. To set the SnapLock volume minimum retention period, enter the following command:

   vol options *vol_name* snaplock_minimum_period {*period*|infinite}

   vol_name is the SnapLock volume name.

   period is the retention period specified by a numeral, followed by days (d), months (m), or years (y). Alternatively, you can specify `infinite`, which means that all the files on the volume are retained forever. To learn about retention period values, see the na_vol(1) man page.

   Note: To set the minimum retention period to `infinite`, you must set the default and maximum retention periods also as `infinite`. For details, see the na_vol(1) man page.

   ### Example

   The following command sets a minimum retention period of six months.

   vol options wormvol1 snaplock_minimum_period 6m

What the maximum retention period is

The maximum retention period is the longest retention period a file can have at the time it is committed to WORM. After a file is committed to WORM, its retention period can be extended beyond this limit.

The maximum retention period has the following characteristics:

- The existing files that are already in the WORM state are not affected by changes in this volume retention period.
- The maximum retention period takes precedence over a default period or over a retention date that was explicitly set by the application.
- The maximum allowed retention period is 70 years.

- Until you explicitly reconfigure the retention period, the maximum retention period is 30 years.

Setting the maximum retention period

You can set the SnapLock volume maximum retention period to ensure that applications or users do not assign excessive retention periods that do not conform to the required maximum retention period values. You can also set the maximum retention period to infinite.

Step

1. To set the SnapLock volume maximum retention period, enter the following command:

   vol options *vol_name* snaplock_maximum_period [*period* | infinite]

   vol_name is the SnapLock volume name.

   period is the retention period specified by a numeral, followed by days (d), months (m), or years (y). Alternatively, you can specify infinite, which means that all files on the volume are retained forever.

   See the na_vol(1) man page for details.

   Example
   The following command sets a maximum retention period of three years:

   vol options wormvol1 snaplock_maximum_period 3y

What the default retention period is

The default retention period is assigned to a WORM file on a SnapLock volume if the file was not explicitly assigned a retention period.

The initial value of the default retention period is equal to the maximum retention period on SnapLock Compliance volumes and the minimum retention period on SnapLock Enterprise volumes. Any changes in these values change the default retention period.

You can set infinite retention for a file by setting the default retention period of the volume to infinite. You cannot explicitly assign an infinite retention period to the file.

Note: Any attempt to explicitly set the retention date of the file before committing it to the WORM state will cause the file to be retained for that specified period. The file will not be retained for an infinite period.

Setting the default retention period

You can reset the default retention period of a SnapLock volume. You might want to do this to ensure that a retention period is assigned to all the WORM files on the volume even if users or the application failed to assign a retention period.

Step

1. To reset the SnapLock volume default retention period, enter the following command:

   vol options *vol_name* snaplock_default_period [*period*|min|max|infinite]

   vol_name is the SnapLock volume name.

   period is the retention period specified by a numeral, followed by days (d), months (m), or years (y). Alternatively, you can specify `infinite`, which means that the files are committed to WORM state and retained forever. For details, see the na_vol(1) man page.

   `min` is the retention period specified by the `snaplock_minimum_period` option.

   `max` is the retention period specified by the `snaplock_maximum_period` option.

   Example

   The following command sets the default retention period equal to the minimum retention period:

   vol options wormvol1 snaplock_default_period min

   Note: The value of the SnapLock default period must be greater than the value of the SnapLock minimum retention period and less than the value of the SnapLock maximum retention period.

What the privileged delete feature is

The privileged delete functionality of SnapLock allows a privileged user to delete a file that is otherwise immutable because of a retention policy. A privileged user should be part of the "Compliance Administrators" group. On a SnapLock Enterprise volume, a privileged user can delete a file irrespective of its retention period. This feature is available only on a SnapLock Enterprise volume; it is not available on a SnapLock Compliance volume. The delete operation is recorded in the SnapLock log file. All the privileged delete operations are logged in the configured SnapLock log volume. Therefore, the privileged delete feature is also known as "Auditable Delete."

When a privileged user deletes a WORM file, the event needs to be reliably logged for auditing and inspection later. The following SnapLock operations are logged in the SnapLock log volume:

- Privileged delete of a retained WORM file
- Changes to privileged delete state of the SnapLock Enterprise volume
- Changes to the SnapLock log volume
- Addition of a user to and removal of a user from the Compliance Administrators group.

In the SnapLock log file, you can find details related to the privileged deletion of a WORM file, such as whether a file has been deleted, when it was deleted, and who deleted the file.

The privileged delete feature enables the compliance administrator to delete a WORM file that is retained on a SnapLock Enterprise volume. The delete operation deletes only the specified file. You can also use the privileged delete functionality on a WORM file that has infinite retention.

Note: You cannot use a privileged delete operation to delete an expired WORM file that has exceeded its retention period.

Next topics

How privileged delete works

By default, a newly created SnapLock Enterprise volume does not allow privileged delete operations. The initial state of the privileged delete feature in a SnapLock Enterprise volume is

`uninitialized`. Therefore, to use privileged delete, you need to enable the privileged delete option in the SnapLock Enterprise volume.

Each volume has one of the following options for the privileged delete functionality:

- `on`—The privileged delete feature is turned on and deletions are allowed on that SnapLock Enterprise volume.
- `off`—The feature is turned off and no privileged delete operations are allowed on that SnapLock Enterprise volume.
- `disallowed`—The feature is disabled for this volume and can never be turned on for this volume. If this option is set, you cannot delete any files from the volume before the expiry date.

You can transition the volume option between the three values. However, you cannot enable the privileged delete functionality on a volume for which the volume option is `disallowed`. In certain scenarios, where the volume option for the privileged delete feature is set to `on`, the privileged delete operations might fail. This happens because the SnapLock log file is unavailable. For example, it can happen when the volume is offline.

Note: The SnapLock log captures any change made to the privileged delete configuration, along with the delete events. Therefore, before transitioning between the privileged delete options on the volume, you need to assign a SnapLock log volume in the system.

The privileged delete state of the SnapLock Enterprise volume is a property of the data contained in the SnapLock Enterprise volume. Consequently, in the case of a volume SnapMirror relationship, the state of the privileged delete option on the source volume is transparently transferred to the destination volume. For SnapLock Enterprise volumes, the initial state of the privileged delete feature is `uninitialized`; for SnapLock Compliance volumes, the initial state of the privileged delete feature is `disallowed`; for non-SnapLock volumes, the initial state of the privileged delete feature is `uninitialized`. In addition, when you upgrade a SnapLock Enterprise volume to Data ONTAP 7.3 for the first time, the initial state of the privileged delete feature is `uninitialized`. You can transition the privileged delete state to `on`, `off`, or `disallowed` by using the `snaplock options [-f] privdel on|off|disallowed` command. However, if a SnapLock Enterprise volume in `disallowed` state is a volume SnapMirror destination and you use a `vsm transfer`, `vol copy`, or `aggr copy` command to transfer data from the source volume, the SnapLock Enterprise volume can transition out of the `disallowed` privileged delete state to the privileged delete state of the source.

Note: When you transition a SnapLock Enterprise volume to `disallowed` privileged delete state, you cannot use the `snaplock` command to change the state again.

The SnapLock privileged delete functionality was introduced in the Data ONTAP 7.3 release family starting with release Data ONTAP 7.3.1; it was not available in earlier releases. If you revert to any release earlier than Data ONTAP 7.3, the privileged delete state of the SnapLock Enterprise volume is maintained. When the system is upgraded again to a Data ONTAP 7.3 release, the saved privileged delete state of the SnapLock Enterprise volume is restored from the earlier saved volume metadata. The Data ONTAP system can detect whether the volume is upgraded to a Data ONTAP 7.3 release

for the first time or is upgraded after having already been in a Data ONTAP 7.3 release and having been reverted to a previous release.

Limitations of the privileged delete functionality

Privileged delete functionality has certain limitations.

- The privileged delete functionality is not available for SnapLock Compliance volumes.
- You cannot use the privileged delete functionality to delete a file that has exceeded its retention period.
- You cannot perform the privileged delete operation on alternate data streams because you cannot commit the alternate data streams to the WORM state.
- Privileged delete functionality is not available for use through standard protocols such as NFS or CIFS. It is available only through proprietary command-line interface.
- The vFiler units do not support privileged delete operations. Privileged delete is allowed only for the volumes owned by the default vFiler unit (vFiler0).
- If the privileged delete feature is set to `on` for a SnapLock Enterprise volume, you cannot move the volume from the default vFiler unit to any other vFiler unit.

Ensuring secure connection to the storage system

Before you start working with privileged delete, you must ensure that the session is secure and that the SSH and Telnet sessions are distinct sessions.

Step

1. To ensure that the SSH or Telnet sessions are kept separate from other sessions, perform the following steps:
   a. If the `telnet.distinct.enable` option is set to `off` at the time of logging in, set this option to `on`.

      Example

      ```
      options telnet.distinct.enable on
      ```

   b. Log out and log in again. This enables the `telnet.distinct.enable` option to take effect.

Enabling privileged delete functionality

To use the privileged delete functionality, you need to enable it in the SnapLock Enterprise volume.

Before you begin

You must configure the SnapLock log volume before you enable the privileged delete functionality in the volume.

Steps

1. To configure the SnapLock log volume, enter the following command:

   snaplock log volume *volume_name*

   The log volume should be a SnapLock Compliance volume.

2. To enable the privileged delete functionality in a SnapLock Enterprise volume, enter the following command:

   snaplock options *volume_name* privdel on

Disabling or disallowing privileged delete functionality

You can disable (turn off) or disallow (permanently remove) the privileged delete feature in a volume. To disallow the privileged delete feature, you must first disable the privileged delete feature in the volume.

Step

1. To disable or disallow the privileged delete feature in a SnapLock volume, perform one of the following steps:

If you want to...	Then...
Turn off the privileged delete feature	Enter the following command: **snaplock options *volume-name* privdel off**

If you want to...	Then...
Disallow privileged delete functionality in a volume	Turn off the privileged delete feature, and then enter the following command: **`snaplock options volume-name -f privdel disallowed`** **Attention:** The `disallowed` option deletes the privileged delete functionality from the volume. Therefore, when you change the privileged delete state of a volume to `disallowed`, you can never perform a privileged delete operation on that volume.

Deleting a WORM file using privileged delete

You can use the privileged delete feature to delete a WORM file in a SnapLock Enterprise volume before the expiry date of the file is reached. This functionality is supported only for SnapLock Enterprise volumes.

Before you begin

You need to configure a SnapLock log volume before enabling the privileged delete functionality. The log volume must be a SnapLock Compliance volume.

Steps

1. Configure a SnapLock log volume by using the following command:

   `snaplock log volume volume-name`

   where *`volume-name`* is the name of the SnapLock Compliance volume that you want to use for SnapLock logging.

2. In the SnapLock Enterprise volume, create a file with a retention period of, for example, 30 years by using the following command:

   `touch -a -t 203811210600 filename`

   Example

   `touch -a -t 203811210600 document.txt`

   The retention period of the `document.txt` file is set to 30 years.

3. Enable privileged delete functionality on the SnapLock Enterprise volume by using the following command:

   `snaplock options volume-name privdel on`

4. Delete the file by using the following command:

   `snaplock privdel [-f] path`

The `-f` flag allows the command to proceed without interactive confirmation from the user.

Note: Only a privileged user can perform the privileged delete operation. The user should be part of the "Compliance Administrators" group.

5. Check the log entry for this delete operation.

How privileged delete affects mirroring interactions

Using SnapMirror, you can replicate a SnapLock Enterprise volume to a non-WORM or another SnapLock Enterprise volume.

If privileged delete is enabled on a SnapLock Enterprise source volume, and the destination is also a SnapLock Enterprise volume, the transfer of the privileged delete state depends on the type of mirroring used. To use the privileged delete functionality at the destination, you need to break the SnapMirror relationship and make the destination volume writable.

In the case of a volume SnapMirror relationship, the privileged delete state of the source volume is transparently transferred to the volume SnapMirror destination volume irrespective of the type of the source and destination volumes. For a SnapLock Enterprise volume and a regular volume, the initial state is `uninitialized`; for a SnapLock Compliance volume the initial state is `disallowed`. In this type of mirroring, the state of the privileged delete feature on the final destination is the same as that on the source volume.

In the case of a qtree SnapMirror relationship, the privileged delete state of the source is not transferred to the destination.

Note: You must separately replicate the log volume to the SnapMirror destination for maintaining complete information about all the privileged delete activity associated with the replicated SnapLock Enterprise volume.

Considerations when using the privileged delete feature

You can perform privileged delete operations over RLM, console, and secure shell (SSH) connections. However, there are certain issues to consider.

Keep in mind the following considerations while using RLM and the console:

- You cannot perform privileged delete operations if both the RLM and console sessions are active.
- You cannot perform privileged delete operations if you had a shared login session previously for the RLM and console sessions.

What SnapLock logging is

SnapLock logging is an infrastructure for logging SnapLock events. The SnapLock events are recorded in the SnapLock log file.

Following are the properties of SnapLock log files:

- The log files are protected from tampering. They are not writable by any external applications.
- SnapLock log files are WORM files that reside on a SnapLock Compliance volume.

In SnapLock logging, you cannot delete the log files until their retention period expires. The retention period of the SnapLock log file is derived from the events that are logged in. The minimum and default retention period of the log file is six months. You can determine the default retention period by using the option `snaplock.log.default_retention`.

When a new record is added in the log file, the retention period of the log file is modified. The retention period of the log file is compared with the retention period of the new event logged in the file. If the retention period of the logged event is longer, the retention period of the log file is made equal to the retention period of the logged event.

In the case of the privileged delete record, the retention period is derived from the file that is deleted. For other events (addition or deletion of compliance users, enabling or disabling of the privileged delete feature, and changes to the log volume), the retention period is equal to the value obtained from the `snaplock.log.default_retention` option.

A SnapLock log file name contains a base name along with starting date and time, and closing date and time, for example, `priv_delete.20080825_161858_GMT-20080826_121346_GMT`. However, for the active log files, the names do not have an end time specified, for example, `system_log.20080909_131521_GMT-present`.

You can use SnapLock logging for any audits or log. SnapLock logging is also useful for privileged delete operations.

Next topics

Related concepts

What the privileged delete feature is on page 43

Types of SnapLock log files

SnapLock log files are of two types, `system_log` and `priv_delete`.

The `system_log` file logs system-related SnapLock events such as the addition of a compliance user and changes to the SnapLock log volume.

The `priv_delete` log file logs the privileged delete operations performed on a SnapLock file.

The SnapLock log file name contains the following parameters:

- Base name (`system_log` or `priv_delete`)
- Starting date and time (date and time when the SnapLock log file was created)
- End date and time (present only when the SnapLock log file is archived)

For example, a `priv_delete` log file could have the following name: `priv_delete.20080825_161858_GMT-20080826_121346_GMT`.

A `system_log` file could have the following name: `system_log.20080825_161858_GMT-20080826_121346_GMT`.

Note: The log file that is being used for logging is called the active log file. An active log file name does not have the end date and time specified in it. It contains only the start date and time. For example, an active `system_log` file could have the following name: `system_log.20080909_131521_GMT-present`.

Advantages of SnapLock logging

SnapLock logging has several advantages.

- SnapLock logging logs all the privileged delete operations that affect a WORM file.
- The SnapLock log resides in the SnapLock Compliance volume. Therefore, you cannot delete the log file until the retention period expires. You can extend the retention period of a log file, but you cannot shorten it.
- The log files are protected from tampering. SnapLock files are not writable by any external applications.
- The SnapLock log file contains the version number; therefore, it is possible to check the compatibility between the log files in different Data ONTAP releases.
- SnapLock log files can be parsed by an XML parser.

Limitations of SnapLock logging

SnapLock logging has certain limitations.

- SnapLock logging fails if there is not enough space in the logging volume.
- SnapLock logging fails if the required licenses are not installed. The log volume is a SnapLock Compliance volume; therefore, you need to install the SnapLock Compliance license.
- SnapLock logging fails if the volume used for logging is not available (is offline or does not exist).
- You cannot move a SnapLock log volume from the default vFiler unit to any other vFiler unit.
- If you want to use NTFS or a mixed security style for SnapLock logging, you must upgrade the storage system to Data ONTAP 7.3.2 or later. In such a case, be sure to upgrade both the nodes in an active/active configuration to Data ONTAP 7.3.2. This ensures that the SnapLock logging operation works correctly during a takeover.

Assigning a SnapLock log volume

When you assign a SnapLock log volume, a `system_log` file is created in the volume. The `system_log` file stores the system-related SnapLock events such as the addition of a compliance user or a change to the SnapLock log volume.

Before you begin

You need to have a SnapLock license and a SnapLock Compliance volume.

Steps

1. To check the existing volume, enter the following command:

   `vol status`

   This command lists all the volumes. You need to select a SnapLock Compliance volume.

2. To list the system compliance log volume, enter the following command:

   `snaplock log volume`

   This command gets the name of the current SnapLock log volume.

3. To set the SnapLock log file to another volume, enter the following command:

   `snaplock log volume -f` ***`worm_log_volume`***

   `worm_log_volume` is the name of the log volume. You can assign any SnapLock Compliance volume on a storage system as the log volume.

Example

snaplock log volume -f *logvol*

logvol is the name of the SnapLock log volume that you want to set.

Note: The `-f` option force-changes the SnapLock log volume.

4. To check the log status of all the active log files on a volume, enter the following command:

   snaplock log status *worm_log_volume*

   Example

   snaplock log status *logvol*

   logvol is the name of the SnapLock log volume.

How archiving a log file works

Archiving a file means committing the log file to the WORM state. You cannot use a log file after it is archived.

You can archive a log file automatically or manually. The logs are archived automatically when the size of the log file exceeds the size that is specified in the `snaplock.log.maximum_size` option.

However, you can also manually archive log files. Manual archiving is required in scenarios where you want to keep the privileged delete log records of certain files separately. You can use the `snaplock log archive` command to create a new log file, perform privileged delete, and then again archive the log file.

Archiving log files

You can archive a single active log file or all active log files on a volume.

Steps

1. To get the maximum permissible size of the log file, enter the following command:

   options snaplock.log.maximum_size

2. To archive the file, enter the following command:

   snaplock log archive *vol* [*basename*]

   vol is the name of the SnapLock volume that contains the active SnapLock log files.

   basename is the base name of the log file that needs to be archived. Currently, there are only two log files—`system_log` and `priv_delete`. If you do not specify the base name, all active log files in the SnapLock volume are archived.

Note: If the archiving operation fails, the `snaplock log status` command displays the following error message:

```
Log files on volume are inconsistent. Use snaplock log archive
command or change the log volume.
```

Finding the status of the SnapLock log file

You can get the status of either a single active SnapLock log file or all log files in a volume.

Step

1. To get the status of a SnapLock log file, enter the following command:

   `snaplock log status` ***`vol`*** **`[`*** ***`basename`*** **`]`**

   `vol` is the name of the SnapLock volume that contains the active SnapLock files.

   `basename` is the base name of the log file for which you want the status.

Result

The `snaplock log status` command displays the following information:

- Log volume name
- Total log files: The command displays the number of active SnapLock log files in a volume. For each active SnapLock log file, it displays the following information:
  - Base name
  - Complete path
  - Expiry date
  - Size
  - SnapLock error

Following is an example of the output of the `snaplock log status` command:

```
system_a> snaplock log status vol_slc
Log Volume Name : vol_slc, Total active logfiles in volume : 2
-----------------------------------------------------------------
Basename: system_log
Complete_path: /vol/vol_slc/system_log.20090330_152241_GMT-present
Expiry: Wed Sep 30 15:22:41 GMT 2009
Size: 503

Basename: priv_delete
Complete_path: /vol/vol_slc/priv_delete.20090330_152915_GMT-present
Expiry: Wed Sep 30 15:29:15 GMT 2009
Size: 104KB
```

What the SnapLock log file contains

When an unexpired SnapLock file is deleted, an entry is made in the log file. The log file captures all information about the SnapLock file.

This log record serves as an audit trail for the file during an audit. The log entry is evidence that the deleted file existed. If the file expiry time of the deleted SnapLock file exceeds the file expiry time of the SnapLock log file, the file expiry time of the log file is automatically extended to be the same as the file expiry time of the deleted SnapLock file. This ensures that the log record for file deletion is retained at least until the expiry date of the deleted file. The SnapLock Compliance volume contains the SnapLock log files. The `priv_delete` log file logs in the privileged delete operations.

The SnapLock log file logs in the following events:

- snaplock.log.volume.changed
- snaplock.pre.privileged.delete
- snaplock.post.privileged.delete
- snaplock.pre.option.privileged.delete
- snaplock.post.option.privileged.delete
- snaplock.user.added.deleted

Note: The log entries log in the following fields: LogEntry date (System Date), CCdate (ComplianceClock date), node (name of the system), vFilerName, and vFilerUUID.

When you use the privileged delete feature, the `snaplock.pre.privileged.delete` log entry is logged immediately before the file is deleted, and the `snaplock.post.privileged.delete` log entry is logged immediately after the file is deleted. The following table lists the fields that are logged for `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete` events in the SnapLock log file with the base name `priv_delete`.

Field	Events	Description
sequence_number	Logged for both `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete`	The sequence number maps to the `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete` log entries.
file_pathname	Logged for both `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete`	The complete path of the file on which the privileged delete operation is performed.
file_expires	Logged for both `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete`	Expiry date of the file.

Field	Events	Description
file_fingerprint	Logged for both `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete`	XML-formatted fingerprint using a secure hash of the file's metadata and contents.
client_user	Logged for both `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete`	User name of the client who performs the operation.
client_ip_address	Logged for both `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete`	The IP address of the client who performs the delete operation. This field is not valid if client_transportclient_transport' is CONSOLE.
client_transport	Logged for both `snaplock.pre.privileged.delete` and `snaplock.post.privileged.delete`	Transport used by client request. It can have the following values: • SSH • HTTPS • CONSOLE **Note:** The client_ip_address field is not valid if this field has value 'CONSOLE'.
result	Logged only for `snaplock.post.privileged.delete`	Describes the success or failure of the delete operation.

The `snaplock.pre.option.privileged.delete` log entry is logged immediately before the privileged delete option is modified on the volume, and the `snaplock.post.option.privileged.delete` log entry is logged immediately after the privileged delete option is modified on the volume. The following table lists the fields that are logged for `snaplock.pre.option.privileged.delete` and `snaplock.post.option.privileged.delete` events in the SnapLock log file with the base name `priv_delete`.

Field	Events	Description
sequence_number	Logged for both `snaplock.pre.option.privileged.delete` and `snaplock.post.option.privileged.delete`	The sequence number maps to the `snaplock.pre.option.privileged.delete` and `snaplock.post.option.privileged.delete` log entries.

Field	Events	Description
volume	Logged for both `snaplock.pre.option.privileged.delete` and `snaplock.post.option.privileged.delete`	The name of the SnapLock volume.
changed_from	Logged for both `snaplock.pre.option.privileged.delete` and `snaplock.post.option.privileged.delete`	The starting value for the option.
changed_to	Logged for both `snaplock.pre.option.privileged.delete` and `snaplock.post.option.privileged.delete`	The ending value for the option.
result	Logged for `snaplock.post.option.privileged.delete`	Describes the success or failure of the option change operation.

The following table lists the fields that are logged for the `snaplock.user.added.deleted` event in the SnapLock log file with the base name `system_log`.

Field	Description
whodidit	The name of the user who modified the compliance group.
username	The name of the user who was added or deleted.
action	The action that occurred to the subject. This can be one of the following values: "added" or "deleted."

The following table lists the fields that are logged for the `snaplock.log.volume.changed` event in the SnapLock log file with the base name `system_log`.

Field	Description
old_log_volume	The name of the previous log volume.
new_log_volume	The name of the new log volume.

Upgrade and revert considerations for SnapLock logging

If you upgrade to Data ONTAP 7.3.1 or later with support for the SnapLock logging feature, you will have access to the new configurable options and commands required for this feature.

If you revert to a release that does not support the SnapLock logging feature, all the active SnapLock log files will be archived. The SnapLock log file is committed to the WORM state and acts as a simple WORM file. You will lose all the state information about the active log file, for example, the sequence number. The SnapLock log files that already exist are protected against any modification and deletion. All the existing WORM files will be retained until the expiry date of the file.

How Data ONTAP tracks the deleted files on SnapLock volumes

Data ONTAP constantly tracks the retention status of WORM files on SnapLock Compliance volumes. It allows you to destroy a SnapLock Compliance volume depending on the status of the WORM files on the volume. You can destroy a SnapLock Compliance volume if the volume does not contain any unexpired WORM files.

Note: An administrator can destroy SnapLock Enterprise volumes at any time.

Next topics

Destroying a SnapLock volume on page 59
Destroying aggregates on page 59

Destroying a SnapLock volume

You must check the status of the volume and the WORM files before attempting to destroy the volume.

Before you begin

Ensure that the volume contains no unexpired WORM files. If the SnapLock volume contains unexpired WORM files, Data ONTAP will not allow you to destroy the volume. It returns a message stating that the `vol destroy` command cannot destroy the volume.

Steps

1. Take the volume offline by entering the following command:

   `vol offline` *`vol_name`*

2. To destroy the SnapLock volume, enter the following command:

   `vol destroy` *`vol_name`*

   If a SnapLock Compliance volume cannot be destroyed, it remains offline. You should immediately bring it online by using the `vol online` *`vol_name`* command. This step ensures that ComplianceClock does not fall behind.

Destroying aggregates

An aggregate must be taken offline for destroying. You can take SnapLock aggregates offline only when they do not contain flexible volumes. You must destroy the flexible volumes contained in an

aggregate before you can destroy the aggregate. This applies to all flexible volumes within SnapLock aggregates.

Steps

1. Destroy all flexible volumes contained within the aggregate you want to destroy, by using the `vol destroy` command.
2. To take the aggregate offline, enter the following command:

   `aggr offline` ***`aggr_name`***

   `aggr_name` is the name of the aggregate that you intend to destroy and whose disks you are converting to hot spares.
3. Destroy the aggregate by entering the following command:

   `aggr destroy` ***`aggr_name`***

   `aggr_name` is the name of the aggregate that you are destroying and whose disks you are converting to hot spares.

Related tasks

Destroying a SnapLock volume on page 59

How SnapLock uses fingerprints

A fingerprint is useful for checking data integrity. The fingerprint operation allows you to generate a fingerprint on a per-file basis by using either of the following hashing algorithms—MD5 or SHA-256.

The fingerprint accounts for data set in place before migration and compares the data before and after migration. In addition, fingerprinting enables log tracking, thus extending the logging functionality.

For example, the privileged delete feature needs the ability to disambiguate (in a crash situation) between a file that was not deleted and a file that is deleted and replaced with a new file with the same name or contents.

You can create fingerprints for both WORM and non-WORM data files. You can generate fingerprints in XML format that can be viewed using any XML browser.

The SnapLock fingerprint operation supports the following algorithms:

- MD5
- SHA-256

By default, SnapLock uses the SHA-256 algorithm.

Next topics

How a fingerprint is calculated on page 61
Input parameters for the fingerprint operation on page 62
Calculating the fingerprint of a file on page 62
Output parameters for the fingerprint operation on page 63

How a fingerprint is calculated

A fingerprint is calculated based on the file data and metadata.

- File data: Contents of the file
- File metadata:
  - File type
  - File flags
  - File size
  - File user ID
  - File group ID
  - File change time (the time when the inode was changed)
  - File modification time

- File creation time
- File retention time

Input parameters for the fingerprint operation

You need to provide the input parameters for the fingerprint operation, including path, algorithm, and scope of the fingerprint operation.

The following table describes the input parameters.

Input parameter	Description
Path	The path of the file for which the fingerprint is calculated.
Fingerprint scope	The scope of fingerprint calculation. It can have the following values: • Data: to be calculated on the data of a file only. • Metadata: to be calculated on the metadata only. • Data and metadata: to be calculated on the data and metadata of a file.
Fingerprint algorithm	The following algorithms are used for fingerprint calculation: • MD5 (hashing algorithm) • SHA-256 (hashing algorithm)

Calculating the fingerprint of a file

You can create a fingerprint for a file by using the `file fingerprint` command.

Step

1. To perform the fingerprint operation, enter the following command:

   `file fingerprint [-a {MD5|SHA-256}] [-m] [-d] [-x]` ***`path`***

   `-a` specifies the algorithm for fingerprint calculation. It can be either MD5 or SHA-256. By default, file fingerprinting uses the SHA-256 algorithm.

   `-m` specifies the metadata fingerprint calculation.

   `-d` specifies the data fingerprint calculation. By default, the fingerprint is calculated for both data and metadata.

   `-x` displays the output in XML format.

path is the path of the file for which the fingerprint operation is required.

Example:

To display the regular output, enter the following command:

```
file fingerprint -a MD5 -m -d /vol/vol_worm/myfile
```

To display the XML output, include -x in the command:

```
file fingerprint -a MD5 -m -d -x /vol/vol_worm/myfile
```

Output parameters for the fingerprint operation

The fingerprint operation provides the output parameters of a file, such as general information about the fingerprint operation, volume information, file information, and system information.

The following table describes the output parameters.

Output parameter	Description
General information about the fingerprint operation	• Fingerprint algorithm: the algorithm that was used for calculation. • File fingerprint scope: The scope provided for the input parameter. It can be data, metadata or both (data and metadata). • File fingerprint version: the version of the file fingerprint. Currently, Data ONTAP uses version 1. • File fingerprint start and stop time: the start and stop time of the file fingerprint calculation. • Readable format of the fingerprint start and stop times. • Path: the path of the file for which the file fingerprint calculation was done.
System information	• FilerID: the NVRAM ID of the storage system. • Filer-name: the name of the storage system. • SnapLock license: the type of SnapLock license (SnapLockCompliance or SnapLock Enterprise). • ComplianceClock time: the time of the ComplianceClock in the system. It is valid only if the ComplianceClock is set on the storage system. • Readable format of ComplianceClock time.

Output parameter	**Description**
Volume information	• Volume name: the name of the volume to which a file belongs. • Volume-uuid: the universal unique identifier for the volume. • Volume type: the type of volume (traditional or flexible). • Aggregate name for flexible volume: the name of the containing aggregate if it is a flexible volume. • Aggregate uuid: the universal unique identifier for the aggregate if the volume is a flexible volume. • Volume expiry date: the expiry date of the volume. • Fsid: the file system ID for the file. • is-volume-expiry-date-wraparound: volume-expiry-date is a wraparound format. The wraparound format indicates that dates after 01/19/2038 are mapped from 01/01/1970 - 12/31/2002 to 01/19/2038 - 01/19/2071. The field is not included if volume-expiry-date is not included. It is present only for SnapLock volumes. • formatted-volume-expiry-date: The expiry date of the SnapLock volume is in the format *<day> <month> <day of month> <hour> : <min>: <sec> <year>* in the GMT time zone. A value of `infinite` indicates that the volume has an infinite expiry date. A value of `scan_in_progress` indicates that the expiry date is not displayed because a WORM scan is in progress on the volume. A value of `no_expiry_date` indicates that the expiry date is not displayed because the SnapLock volume has no WORM files and WORM Snapshot copies. This field is not included if the volume is offline or the volume is a non-SnapLock volume. • SnapLock-volume-type: the type of SnapLock volume. This information is present for SnapLock volumes only. Possible values: `compliance` or `enterprise`. • Readable format of volume expiry date and time.

Output parameter	**Description**
File information	• Path: the path of the file for which the fingerprint calculation was performed. • Fileid: the unique ID of the file within the file system. • File type: the file type, for example, WORM, WORM appendable, WORM log, or regular file. • File size: the size of the file in bytes. • creation time: creation time of the file in seconds in standard UNIX format. Time is taken from system clock. • Formatted-creation-time: creation time of the file formatted in human-readable format *<day><month><day of month><hour>:<min>:<sec> <year>* in GMT timezone. • Modified time: last modification time of file in seconds in the standard UNIX format. Time is taken from system clock. • Formatted-modified-time: last modification time of the file formatted in human-readable format *<day><month><day of month><hour>:<min>:<sec> <year>* in GMT timezone. • access time: the last access time of the file attributes in seconds in the standard UNIX format. This will be displayed only for non-WORM files and files on non-SnapLock volumes. The time is taken from the system clock. • Formatted-access-time: last access time of the file formatted in a human-readable format *<day><month><day of month><hour>:<min>:<sec> <year>* in GMT timezone. This field is included for regular files and files on non- SnapLock volumes. • Retention time: retention time of the file in seconds in standard UNIX format. This field is not included for regular files, files on non-SnapLock volumes and files with infinite retention. The flag is-wraparound indicates that retention is in the wraparound format. The time is taken from ComplianceClock. • is-wraparound: is displayed if the date represented in retention-time is in wraparound format. The wraparound format indicates that dates after 01/19/2038 are mapped from 01/01/1970-12/31/2002 to 1/19/2038-01/19/2071. • Formatted-retention-time: Expiry date of the WORM file formatted in a human-readable format. This takes care of wraparound dates and prints the expiry date of a file in the format *<day><month><day of month><hour>:<min>:<sec> <year>* in GMT timezone. A value as "infinite" indicates that the file has infinite retention time. This field is not included for regular files and files on a non-SnapLock volumes. • Changed-time: last changed time of the file attributes in seconds in standard UNIX format. For WORM files last change time for file attributes occurs when file is committed to WORM. Time is taken from system clock. However, for WORM files, time is taken from ComplianceClock. • Formatted change time: last changed time of the file in human-readable format *<day><month><day of month><hour>:<min>:<sec> <year>* in GMT timezone. • Owner ID: the ID of the owner for the file. • Group ID: the group ID for the file. • Owner SID: the owner's Secure ID if Windows NT security is present for the file. • Data fingerprint: the string format of the fingerprint calculated for the file data. Valid if the scope selected is for data or for both data and metadata. • Metadata fingerprint: the string format for the fingerprint for the file metadata. Not valid if the scope selected is only data. • The output parameter also indicates the retention time and wraparound information. This field is valid only for WORM files.

Note:

- You cannot compute a fingerprint on the symbolic link of the file.
- Currently, fingerprint operation is supported only on one file.

SnapLock interaction with a vFiler unit

SnapLock disallows vFiler unit creation if the root of the vFiler unit is on a SnapLock volume.

In releases earlier than Data ONTAP 7.3.1, the vFiler units can be created with the root on a SnapLock volume. After you upgrade to Data ONTAP 7.3.1 or later, these vFiler units continue to operate. However, an error message is raised when the vFiler units are started. The intention of the error message is to warn about the unsupported configuration.

In Data ONTAP 7.3.1 and later, SnapVault is vFiler unit enabled. Because the SnapLock for SnapVault feature is a SnapVault configuration with SnapVault secondary on the SnapLock volume, the following SnapVault options are also available on the vFiler unit:

- `snapvault.access`
- `snapvault.enable`

Next topics

Creating the root of a vFiler unit from a SnapLock volume to a non-SnapLock volume on page 67

Limitations of vFiler units on SnapLock volumes on page 68

Creating the root of a vFiler unit from a SnapLock volume to a non-SnapLock volume

You cannot move a vFiler unit from a SnapLock volume to a non-SnapLock volume. You can only re-create a vFiler unit (with the same configuration as in the SnapLock volume) in the non-SnapLock volume. Therefore, to recover from a situation where the root of a vFiler unit is in a SnapLock volume, you need to re-create the root of the vFiler unit in the non-SnapLock volume.

Step

1. To re-create a vFiler unit in a non-SnapLock volume, enter the following command:

   `vfiler create` ***`vfilername`*** **`-r path`**

   where *`vfilername`* is the vFiler unit with the root on the non-SnapLock volume.

Re-creating the vFiler unit in a non-SnapLock volume

The following example shows how to re-create the vFiler unit (with the same configuration as in the SnapLock volume) in a non-SnapLock volume.

1. Consider a vFiler unit, `vfiler1`, having its root in the SnapLock volume `sle0` (at `/vol/sle0/etc`). Copy `/etc` from `sle0` to `vfvol0`, where `vfvol0` is a non-SnapLock volume.
2. Rename SnapLock volume `sle0` to `sle0_temp` and non-SnapLock volume `vfvol0` to `sle0`.
3. Enter the following command:

   `vfiler create vfiler1 -r /vol/sle0`

   This command re-creates the vFiler unit with its root on `/vol/sle0/etc` (which is actually the non-SnapLock volume).
4. Rename `sle0` to `vfvol0` in the non-SnapLock volume and `temp_sle0` to `sle0` in the SnapLock volume.

   The vFiler unit, `vfiler1`, is created with its root directory switched from `/vol/sle0/etc/` in the SnapLock volume to `/vol/vfvol0/etc/` in the non-SnapLock volume.

Limitations of vFiler units on SnapLock volumes

Following are the limitations of vFiler units on SnapLock volumes:

- You cannot create new vFiler units on a SnapLock volume.
- The SnapLock commands work only for the volumes that are owned by the root vFiler unit.
- The SnapLock log volume must be completely owned by the default vFiler unit (vfiler0).
- The SnapLock log volume cannot be moved to a nondefault vFiler unit.
- Privileged delete operations cannot be performed on a volume that is not completely owned by the default vFiler unit.
- If a SnapLock Enterprise volume has the `privdel` option set to `on`, you must disable the option before moving the volume to a nondefault vFiler unit.

SnapLock interaction with active/active configuration

In an active/active configuration pair, both the nodes should be running on the same Data ONTAP version. SnapLock volumes cannot be created in an active/active configuration if one node is running on the first release of Data ONTAP 7.3 and the other node is running on Data ONTAP 7.3.1 or later. This is because the first release of Data ONTAP 7.3 does not support SnapLock.

For example, consider an active/active configuration with two nodes, node A running on Data ONTAP 7.3.1 and node B running on Data ONTAP 7.3. In addition, node A contains SnapLock volumes. If node A goes down, the disks of node A are taken over by the disks of node B. However, due to the presence of the SnapLock volumes, node B halts.

Therefore, to enable the creation of new SnapLock volumes in an active/active configuration, you need to upgrade both the nodes to Data ONTAP 7.3.1 or later.

For proper functioning of wormlog commit operations, both the nodes in an active/active configuration should use Data ONTAP 7.3.2 or later.

SnapLock interaction with MetroCluster

As with any volume on a mirrored aggregate, MetroCluster enables SnapLock volumes to be mirrored from one site to another while retaining the SnapLock characteristics, assuming that the sites have been properly configured.

However, if you are planning to use both SnapLock and MetroCluster together, see the *MetroClusters and SnapLock volumes* section of *Data ONTAP Active/Active Configuration Guide*.

Protecting your SnapLock volumes with SnapMirror

You can protect backups on SnapLock volumes by using the SnapMirror relationship.

If you want to protect backups on SnapLock Enterprise volumes with SnapMirror, you can use the same procedures that you use for non-SnapLock volumes.

However, if you are protecting backups on SnapLock Compliance volumes with SnapMirror, some operations, such as the `snapmirror resync` command, cannot be used. This is because a resynchronization operation might alter data and, by definition, compliance data must not be altered. To learn how to set up a SnapMirror relationship, see the *Data ONTAP Data Protection Online Backup and Recovery Guide*.

In a qtree SnapMirror relationship, the resynchronization of qtree SnapMirror (using the `snapmirror resync` command) and resynchronization of SnapVault (using the `snapvault start -r` command) to SnapLock Compliance volumes is allowed. This is because Data ONTAP provides a mechanism to store the data altered by a resynchronization for subsequent retrieval.

The following table is a quick reference for which SnapLock volumes you can use the `resysnc` option on.

	SnapLock Compliance destination volume	**SnapLock Enterprise destination volume**
Qtree SnapMirror resynchronization	Yes	Yes
Volume SnapMirror resynchronization	No	Yes

Next topics

SnapLock qtree SnapMirror resynchronization restrictions

Beginning with Data ONTAP 7.2.5.1, qtree SnapMirror resynchronization is not supported for mirroring volumes from a less strict qtree to a SnapLock Compliance qtree. When the source is a

non-SnapLock qtree or a SnapLock Enterprise qtree, and the destination is a SnapLock Compliance qtree, resynchronization is not allowed.

The following table shows all the possible combinations and whether resynchronization is allowed.

If SnapMirror resync source is...	And SnapMirror resync destination is...	Resynchronization is...
SnapLock Compliance volume	SnapLock Compliance volume	Allowed
SnapLock Enterprise volume	SnapLock Compliance volume	Not allowed
Non-SnapLock volume	SnapLock Compliance volume	Not allowed
SnapLock Compliance volume	SnapLock Enterprise volume	Allowed
SnapLock Enterprise volume	SnapLock Enterprise volume	Allowed
Non-SnapLock volume	SnapLock Enterprise volume	Allowed
SnapLock Compliance volume	Non-SnapLock volume	Allowed
SnapLock Enterprise volume	Non-SnapLock volume	Allowed
Non-SnapLock volume	Non-SnapLock volume	Allowed

In addition, to allow a qtree SnapMirror resynchronization between SnapLock Compliance qtrees, you must make at least one transfer from the SnapLock Compliance qtree source to the SnapLock Compliance qtree destination after you upgrade to Data ONTAP 7.2.5.1 or a later release and before you break the relationship. If you do not make the transfer after the upgrade, the resynchronization is disallowed.

What the dump file is

The dump file is created when a `snapmirror resync` is initiated to synchronize the SnapLock destination volume with the SnapLock source volume.

The dump file contains the files changed or added on the SnapLock destination volume after the SnapMirror relationship was broken. The data is written to the dump file in the BSD dump format.

Extracting files from the dump file after a qtree SnapMirror resynchronization

After the qtree SnapMirror resynchronization between the SnapLock Compliance source and destination volume pairs, any new data created on the destination qtree will be archived in a dump

file. These files are stored in the following directory on the destination SnapLock Compliance volume:`/etc/logs/snapmirror_resync_archive/volname_UUID_qtree`.The image and log files are named `dump_image_YYMMDD_HHMMSS` and `dump_log_YYMMDD_HHMMSS`, respectively.

Steps

1. Create a directory with the name `temp` in the SnapLock volume.
2. Copy the dump file into this directory and provide the new name `dumpfile` to the file.
3. To view the files contained in the dump file, run the following command:

   `restore -tf /vol/volume_name/temp/dumpfile`
4. To restore the files contained in the dump file to their original location, enter the following command:

   `restore -rfQ /vol/volume_name/temp/dumpfile`
5. To restore the files contained in the dump file to a different location, enter the following command:

   `restore -rfQD /vol/volume_name/temp/dumpfile /vol/volume_name/temp`
   The files extracted will be in their original SnapLock state irrespective of the approach used.

How to set an end-to-end SnapLock Compliance volume SnapMirror relationship

To create an end-to-end relationship between SnapLock Compliance volumes by using SnapMirror, you need to create both the source and the destination volumes as SnapLock Compliance volumes and then initialize the mirroring relationship by using the `snapmirror initialize` command. The other commands you use differ depending on whether you are creating a volume SnapMirror relationship for a traditional volume or a FlexVol volume.

Note: In a volume SnapMirror relationship between two SnapLock Compliance volumes, the exact status of the SnapLock log volume on the source is transferred to the destination volume. The active log files remain active on the destination. The WORM log metafile is also replicated to the destination, and it points to the active WORM log files on the destination. Therefore, you can use the WORM log file from the same state as it was at the source, before the migration.

Limitations of the SnapMirror relationship

The SnapMirror relationship in SnapLock has certain limitations.

- The SnapMirror relationship is not allowed if the source is a non-WORM volume and the destination is a SnapLock Compliance volume.

- The SnapMirror relationship is not allowed if the source is a SnapLock Enterprise volume and the destination is a SnapLock Compliance volume.
- A volume SnapMirror relationship is not allowed between two SnapLock Compliance volumes if the destination volume has unexpired files.
- Deletion of a Snapshot copy on a SnapLock Compliance volume is disallowed if the volume is part of a SnapMirror relationship and the Snapshot copy is locked by SnapMirror. This behavior protects you from accidentally deleting the base Snapshot copy of a volume SnapMirror relationship on a SnapLock Compliance volume.
- On a SnapLock Compliance volume, if the base Snapshot copy is deleted, the volume SnapMirror relationship is broken and it can never be reestablished.

Creating a volume SnapMirror relationship for a FlexVol volume

You can create a volume SnapMirror relationship for a FlexVol volume.

Steps

1. To create a SnapLock Compliance aggregate source, enter the following command:

   aggr create *aggr_name* [-L [*compliance*|*enterprise*]] *ndisks[@disksize]*

   `-L` is specified to create a SnapLock aggregate. The aggregate can either be a SnapLock Compliance aggregate or a SnapLock Enterprise aggregate.

   Example

   aggr create wormaggrsrc -L compliance 3

2. To create a volume in that aggregate, enter the following command:

   vol create *vol_name aggr_name size[k|m|g|t]*

   Example

   vol create src wormaggrsrc 20g

3. To create a SnapLock Compliance aggregate destination, enter the following command:

   aggr create *aggr_name* [-L [*compliance*|*enterprise*]] *ndisks[@disksize]*

   `-L` is specified to create a SnapLock aggregate. The aggregate can either be a SnapLock Compliance aggregate or a SnapLock Enterprise aggregate.

   Example

   aggr create wormaggrdst -L compliance 3

4. To create a volume in that aggregate, enter the following command

   vol create *vol_name aggr-name size[k|m|g|t]*

Example

```
vol create dst wormaggrdst 20g
```

5. On the destination system, restrict the volume by using the `vol restrict` command.

   ```
   vol restrict vol_name
   ```

   Example

   ```
   vol restrict dst
   ```

6. To initialize the SnapMirror relationship between the source and the destination, enter the following command:

   ```
   snapmirror initialize -S src_system:src_vol dst_system:dst_vol
   ```

   src_system is the name of the source system.

   src_vol is the volume you want to copy.

   dst_system is the name of the destination system.

   dst_vol is the destination volume.

   Example

   ```
   snapmirror initialize -S src dst
   ```

   The `snapmirror initialize` command initializes the SnapMirror relationship between the source volume *src* and destination volume *dst*.

Creating a volume SnapMirror relationship for a traditional volume

You can create a volume SnapMirror relationship for a traditional volume.

Steps

1. To create a SnapLock Compliance volume, enter the following command:

   ```
   vol create vol_name [-L [compliance|enterprise]] ndisks[@disk-size]
   ```

   Example

   ```
   vol create src_vol -L compliance 3
   ```

2. To create a SnapLock Compliance destination volume, enter the following command:

   ```
   vol create vol_name [-L  [compliance|enterprise]] ndisks[@disk-size]
   ```

   Example

   ```
   vol create dst_vol -L compliance 3
   ```

3. On the destination system, to mark the destination volume as restricted, enter the following command:

   vol restrict *vol_name*

   Example

   vol restrict dst_vol

4. To initialize the SnapMirror relationship between the source and the destination, enter the following command:

   snapmirror initialize -S *src_system:src_vol dst_system:dst_vol*

   src_system is the name of the source system.

   src_vol is the volume you want to copy.

   dst_system is the name of the destination system.

   dst_vol is the destination volume.

   Example

   snapmirror initialize -S src_vol dst_vol

   The `snapmirror initialize` command initializes the SnapMirror relationship between the source volume *src* and destination volume *dst*.

The SnapLock for SnapVault feature—secure SnapVault destination

You can use the SnapLock for SnapVault feature to back up data to SnapLock volumes. The SnapLock for SnapVault feature of Data ONTAP uses SnapVault to back up data, whether it is WORM data or not, to SnapLock Compliance or Enterprise volumes. To keep data on the SnapLock volume compliant, you put the log files on a separate log volume for the SnapLock for SnapVault feature.

Before using the SnapLock for SnapVault feature, you need to ensure that the log volume for the SnapLock for SnapVault feature is configured. To configure the log volume for the SnapLock for SnapVault feature, you use the `snapvault.lockvault.log_volume` option. For added data protection, create a copy of the log volume for the SnapLock for SnapVault feature by using SnapMirror.

The process of using the SnapLock for SnapVault feature to back up data to SnapLock volumes is similar to the process of backing up data to non-SnapLock volumes.

- You can back up the primary system data using the SnapLock for SnapVault feature .
- For additional protection, if you want a copy of the backup of the SnapLock for SnapVault feature, copy the destination volume of the SnapLock for SnapVault feature by using SnapMirror.

In the SnapLock for SnapVault relationship, you cannot have both the source and the destination as SnapLock Compliance volumes.

Note: Open Systems SnapVault 2.1 and later support the SnapLock for SnapVault feature.

Upgrade and revert issues related to the SnapLock for SnapVault feature

When you upgrade your storage system to Data ONTAP 7.3.1 or later, there are certain scenarios in which the SnapLock for SnapVault relationship might not work.

The following table lists these scenarios.

If...	Then...
The storage system for both SnapVault source and destination are on Data ONTAP 7.3.1 or later	The SnapVault relationship between SnapLock Compliance as the source and SnapLock Compliance as the destination is disallowed.

If...	Then...
	The SnapVault relationship between the SnapLock Compliance volume (source) and SnapLock Enterprise volume (destination) is allowed.
	The SnapVault relationship between the SnapLock Compliance volume (source) and a regular volume (destination) is allowed.
	The SnapVault relationship between a regular volume (source) and a SnapLock Enterprise volume (destination) is allowed.
	The SnapVault relationship between regular volume to regular volume is allowed.
	The SnapVault relationship between regular volume (source) and SnapLock Compliance volume (destination) is allowed.
The storage system for the SnapVault source is in Data ONTAP 7.1.3, Data ONTAP 7.2.5.1, or Data ONTAP 7.0.7 and the destination is Data ONTAP 7.3.1 or later	The SnapVault relationship between SnapLock Compliance (source) and SnapLock Compliance (destination) is disallowed.
	In this scenario, all the other SnapVault relationships are allowed between all types of volume.
The storage system for the SnapVault source is upgraded to Data ONTAP 7.3.1 or later and the destination is Data ONTAP 7.1.3, Data ONTAP 7.2.5.1, or Data ONTAP 7.0.7	The SnapVault relationship is allowed between all types of volumes.
The storage system for the SnapVault source and destination are on Data ONTAP 7.2.5.1 and you upgrade the destination to Data ONTAP 7.3.1 or later	Ensure that the source and destination volumes are not SnapLock Compliance volumes. If the destination volume is a SnapLock Compliance volume, the `snapvault update` command fails.

Guidelines for using the SnapLock for SnapVault feature

You should follow certain guidelines to back up data to SnapLock volumes by using the SnapLock for SnapVault feature.

These guidelines cover the following areas:

- Capacity planning
- Management of WORM Snapshot copies from SnapVault
- Management of SnapVault log files

Next topics

Aspects of capacity planning

You should plan the capacity of your storage system and the secondary storage system before performing the backup.

The following are the aspects of capacity planning:

- SnapVault secondary storage system volume size
- The SnapLock for SnapVault Log volume size
- Number of qtrees backed up to each volume
- SnapVault schedule

Next topics

Guidelines for estimating SnapVault secondary storage system volume size

There are guidelines you can follow to estimate the size of SnapVault secondary storage system volumes for growth over one year, depending on how often you back up data.

- Three percent (3%) growth every day

- Five percent (5%) growth every week
- Ten percent (10%) growth every month
- Seven percent (7%) monthly growth rate, compounded to 100 percent growth every year

Following is an example of how to estimate the daily backup.

Example: Estimating daily backup

Assume a starting baseline transfer of 1 GB and 100 percent growth of new data for the year (1 GB). The changed size due to daily growth if changes are made every weeknight is as follows:

- Calculate daily backup by using the following equation: Size due to daily growth = (number of days x .03) - (new data)
  For example, size due to daily growth = (250 x .03) - 1 GB = 7.5 GB - 1 GB = 6.5 GB of change
  Therefore, for 1 GB of source data, you need 1 GB baseline + 1 GB new data + 6.5 GB of changed data = 8.5 GB secondary system volume. In other words, you must be able to scale the secondary system volume to about 8 times the current data set size within one year.

Estimating the log volume size

You can use simple equations to estimate the size of the log data associated with each SnapLock volume and create a log volume equal to the total of the combined estimated sizes.

Steps

1. Calculate the baseline transfer as follows:

   ```
   Baseline transfer = number of inodes per volume x 256 bytes
   ```

   where 256 bytes is an estimated size of the log entry.

2. Calculate the incremental transfers as follows:

   ```
   Incremental transfers = number of inodes x .03 x 250 snapshot copies x
   256 bytes
   ```

   where .03 represents the rate of change.

   Note: If the estimate is too low, you can add disks to the volume; alternatively, you can allocate a second log volume if the first becomes full and configure the system to use the new log volume.

How to set up SnapVault backups

To set up SnapVault backups, you must prepare the primary storage systems and SnapVault secondary storage systems to perform backup tasks. These tasks are almost identical to those for non-SnapLock volumes.

Note: To use SnapVault backup, you must have separate SnapVault licenses for the primary and the secondary storage systems.

- On the primary storage system, use console commands to activate the SnapVault primary license and specify the SnapVault secondary storage host.
- On the SnapVault secondary storage system, use console commands to license and enable SnapVault, initialize ComplianceClock, configure the log volume for the SnapLock for SnapVault feature, specify the primary storage systems to back up, and start the initial Snapshot copy backup.
- On the primary storage system, schedule the time for local SnapVault Snapshot copies to occur. On the SnapVault secondary storage system, schedule the time for the primary Snapshot copies to be backed up to secondary storage.

Next topics

Configuring a primary storage system for SnapVault on page 83
Configuring a SnapVault secondary storage system on page 84
Scheduling SnapVault update backups on the primary storage system on page 85
Scheduling SnapVault update backups on the SnapVault secondary storage system on page 86
Scheduling SnapVault update backups on the SnapVault primary and secondary storage system schedules on page 87
Guidelines for backing up qtrees to a volume using SnapVault on page 87
Guidelines for scheduling SnapVault transfers on page 87

Configuring a primary storage system for SnapVault

To back up a primary storage system to the SnapVault secondary storage system, you need to log in to the primary system's console and enable SnapVault.

Steps

1. Set up the SnapVault primary license on each primary storage system to be backed up. In the console, enter the following command:

   `license add sv_primary_license`

2. Enable SnapVault on each primary storage system to be backed up. In the console, enter the following command:

   `options snapvault.enable on`

3. Specify the name of the SnapVault secondary storage system. To do this, enter the following command:

   options snapvault.access host=*snapvault_secondary*

   The system must be able to resolve the host name (*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 enter the 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.

Configuring a SnapVault secondary storage system

To configure a SnapVault secondary storage system, you need to log in to the secondary system's console and enable the SnapVault license.

Steps

1. Set up the SnapVault secondary license. In the console of the SnapVault secondary system, enter the following command:

   license add *sv_secondary_license*

2. To enable SnapVault, enter the following command:

   options snapvault.enable on

3. Initialize ComplianceClock if you have not already done so. In the console, enter the following command:

   date -c initialize

   The system prompts you to confirm the current local time and that you want to initialize ComplianceClock.

4. Create the log volume for the SnapLock for SnapVault feature, a SnapLock volume that contains Operations log files and Files-transferred log files.

5. Configure the log volume for the SnapLock for SnapVault feature. Enter the following command:

   options snapvault.lockvault_log_volume *volume_name*

   volume_name is the log volume for the SnapLock for SnapVault feature .

   Note: You must use the name of a previously created SnapLock volume for this command to succeed.

6. To specify the names of the primary storage systems to back up and restore, enter the following command:

   options snapvault.access host=*snapvault_primary1,snapvault_primary2* ...

7. For each qtree on the primary storage systems to be backed up, execute an initial baseline copy of the qtree from the primary to the secondary storage system.

On each command line, specify the primary storage system, volume, qtree, and the secondary storage host, SnapLock volume, and qtree. You need to use the `-S` prefix to indicate the source qtree path.

Example

To start a baseline copy of qtrees `tree_a` , `tree_b`, and `tree_c` to a SnapLock Compliance volume called `sv_vol`, use the following commands:

```
snapvault start -S system_a:/vol/vol1/tree_a sv_systemb:/vol/sv_vol/
tree_a


snapvault start -S system_a:/vol/vol1/tree_b sv_systemb:/vol/sv_vol/
tree_b


snapvault start -S system_a:/vol/vol1/tree_c sv_systemb:/vol/sv_vol/
tree_c
```

Note: Enter each command on a single line.

Related tasks

Configuring the log volumes of the SnapLock for SnapVault feature on page 105

Scheduling SnapVault update backups on the primary storage system

You can configure a schedule for Snapshot copies on both the primary and the SnapVault secondary storage systems. You can set the schedules to an hourly, weekly, or nightly basis.

Step

1. On each primary storage system that contains qtrees to be backed up to a SnapVault secondary storage system, schedule the Snapshot copies on each volume that contains the backed-up qtrees by using the following command:

   ```
   snapvault snap sched volume_name snap_name schedule_spec
   ```

   For each set of Snapshot copies, specify the volume name, Snapshot copy base name (for example, “sv_hourly” or “sv_nightly”), number of SnapVault Snapshot copies to store locally, and the days and hours to execute the Snapshot copies.

Examples of scheduling the Snapshot copies

Some of the specific examples of the command follow:

```
snapvault snap sched vol1 sv_weekly 1@sat@19


snapvault snap sched vol1 sv_nightly 2@mon-fri@19
```

```
snapvault snap sched vol1 sv_hourly 11@mon-fri@7-18
```

Note: When specifying the SnapVault Snapshot copy base name, avoid using "hourly," "nightly," or "weekly." Such naming conflicts with the non-SnapVault `snap sched` Snapshot copies.

Scheduling SnapVault update backups on the SnapVault secondary storage system

For each SnapVault volume Snapshot copy set that you scheduled on your primary storage system, you can schedule a set of transfers and the subsequent Snapshot copies of the SnapVault secondary storage system.

Step

1. Enter the following command:

   `snapvault snap sched -x` ***`sec_vol snap_name count`*** **`[@`*** ***`day_list`*** **`][@`*** ***`hour_list`*** **`]`**

   Note: Before scheduling SnapVault Snapshot copies, ensure that the SnapLock default retention period is set correctly or that you explicitly set the retention period when you schedule SnapVault Snapshot copies.

   The `-x` parameter causes SnapVault to copy new or modified data from the primary qtrees to their associated qtrees on the secondary storage system. After all the secondary qtrees on the specified volume have been updated, SnapVault creates a Snapshot copy of this volume for archiving.

   `count` is the number of Snapshot copies you want to retain for this set. If the value is 0, no Snapshot copies are taken. If the value is not 0, Snapshot copies are taken and no Snapshot copies are automatically deleted.

   `@day_list` is a comma-separated list that specifies the days on which a new Snapshot copy is created for this set.

   `@hour_list` specifies the hours at which a new Snapshot copy is created for this set.

   Snapshot copy base names on the primary and secondary systems must match, but Snapshot copy times and number of stored Snapshot copies can differ.

Related concepts

Management of WORM Snapshot copies by using SnapVault on page 87

Scheduling SnapVault update backups on the SnapVault primary and secondary storage system schedules

If SnapVault is scheduled to perform Snapshot copy management at the same time as the default `snapshot sched` activity, the Snapshot copy management operations scheduled using the `snap sched` command might fail with syslog messages.

About this task

The Snapshot copy management operations scheduled using the `snap sched` command might fail with the syslog messages “Skipping creation of hourly snapshot” or “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.

Step

1. To turn off the regular schedule of Snapshot copies, enter the following command:

   ```
   snap sched volume 0 0 0
   ```

Guidelines for backing up qtrees to a volume using SnapVault

Certain guidelines are helpful when you want to use SnapVault to back up qtrees to a volume.

- Due to performance constraints, a maximum of 16 primary qtrees should be backed up to a single secondary volume.
- For optimum performance, the number of qtrees backed up to a volume should be approximately 6.
- If you are using SnapVault to back up volumes to qtrees, a maximum of 16 volumes should be backed up to a volume.

Guidelines for scheduling SnapVault transfers

The only guideline for scheduling SnapVault transfers is not to overload the primary storage system or the network. You must consider the overall load and time to complete all transfers.

Management of WORM Snapshot copies by using SnapVault

You can manage WORM Snapshot copies by using SnapVault.

Next topics

How retention of Snapshot copies works on SnapLock volumes on page 88
How Snapshot copies are named on SnapLock volumes on page 88
Retention period for WORM Snapshot copies created by SnapVault on page 88

How retention of Snapshot copies works on SnapLock volumes

SnapVault uses the `count` field of the SnapVault schedule on the secondary storage system to determine the number of Snapshot copies that are retained.

When the maximum count of Snapshot copies to be retained is reached, the oldest retained Snapshot copies are deleted when new Snapshot copies are added. For SnapLock volumes, older WORM Snapshot copies cannot be deleted until their retention period has expired.

WORM Snapshot copies are not deleted automatically. You must delete WORM Snapshot copies when they expire.

How Snapshot copies are named on SnapLock volumes

Snapshot copies created by SnapVault on SnapLock volumes use a different naming scheme from that used by Snapshot copies for regular volumes.

The SnapLock volumes use the time and date of the Snapshot copy creation as a suffix. The Snapshot copy name contains the system clock time and date as a suffix. Following is the format of the Snapshot copy name: *snapname*`.`*yyyymmdd*`_`*hhmmss*`_`*zzz*.

snapname is the Snapshot copy name specified in the schedule.

yyyymmdd is the year, month, and day.

hhmmss is the hour, minute, and second.

zzz is the current time zone setting.

Example

```
%/used %/total date name
---------- ---------- ------------ --------
2% ( 2%) 0% ( 0%) Feb 04 02:20 svhourly.20040104_120502_GMT
9% ( 3%) 1% ( 0%) Feb 04 02:15 svhourly.20040104_180601_GMT
```

Retention period for WORM Snapshot copies created by SnapVault

You can configure the schedules on the SnapVault secondary storage system to create WORM Snapshot copies and specify a retention period.

For WORM Snapshot copy creation, the volume must exist on the secondary storage system and it must be a SnapLock volume.

Note: If you change the retention period in the schedule, WORM Snapshot copies created under the previous schedule retain their retention period, and WORM Snapshot copies created under the changed schedule use the new retention period.

Next topics

Default SnapVault settings for the WORM Snapshot copies retention period on page 89
Specifying retention periods for WORM Snapshot copies on page 89
Extending the retention period of WORM Snapshot copies on page 90

Default SnapVault settings for the WORM Snapshot copies retention period

When you configure a SnapVault Snapshot copy schedule for a SnapLock volume on the secondary storage system, the Snapshot copies created for that volume are WORM Snapshot copies. By default, SnapVault uses the retention period set by the `snaplock_default_period vol` option as the retention period for the WORM Snapshot copies.

You should ensure that the retention periods you configured when creating a SnapLock Compliance volume are correct.

Note: The default retention period for volumes of the SnapLock for SnapVault feature is 30 years if the volume is a SnapLock Compliance volume. If the volume is a SnapLock Enterprise volume, the default retention period is 0 years. Unless you want all the SnapVault WORM Snapshot copies created in a SnapLock Compliance volume to have a 30-year retention period, be sure to reset the default retention period.

Specifying retention periods for WORM Snapshot copies

You can specify retention periods for different WORM Snapshot copies created in a SnapLock volume.

Step

1. To specify the retention period for WORM Snapshot copies, enter the following command:

   `snapvault snap sched -x -o retention_period=`*`period sec_vol snapname count@day_list@hour_list`*

   `period` is the retention period specified by a numeral followed by days (d), months (m), or years (y).

   `sec_vol` is the name of the volume where the Snapshot copy resides.

   `snapname` is the name of the Snapshot copy.

   `count` is the number of Snapshot copies you want to retain for this set, although this value is ignored.

   `@day_list` is a comma-separated list that specifies the days on which a new Snapshot copy is created for this set.

`@hour_list` specifies the hours at which a new Snapshot copy is created for this set.

Example

WORM Snapshot copies created on the secondary storage system in the `sv_proj` SnapLock volume have retention periods of 360 days from the time of their creation. Snapshot copies are created at noon and 8:00 p.m. every day.

```
snapvault snap sched -x -o retention_period=360d sv_proj sv_hourly
1@12,20
```

Extending the retention period of WORM Snapshot copies

You can extend the retention period of a WORM Snapshot copy.

Step

1. To extend the retention period of a WORM Snapshot copy, enter the following command:

   snapvault snap retain *volumesnapshot period*

   volume is the name of the WORM volume.

   snapshot is the name of the Snapshot copy.

   period is the retention period specified by a numeral followed by days (d), months (m), or years (y).

Example

The following command extends the shown *sv_hourly* Snapshot copy in the *wormvol* volume to two years:

```
snapvault snap retain wormvol sv_hourly.20050513_195442_GMT 2y

***WARNING: YOU ARE REQUESTING SNAPLOCK RETENTION OF A SNAPSHOT***

This operation will enforce the retention of the snapshot by
SnapLock for the specified retention period. You will NOT be able
to delete the retained snapshot until this retention period has
been satisfied. The relevant information for confirmation of this
operation is as follows:

      Volume: wormvol
    Snapshot: sv_hourly.20050513_195442_GMT
Retain until: Fri Feb 17 00:00:27 GMT 2006

Are you REALLY sure you want to retain this snapshot? Y
```

Listing Snapshot copies on the WORM volume

WORM Snapshot copies are identified by the word `snaplock` at the end of each Snapshot copy entry.

Step

1. To list all Snapshot copies, including WORM Snapshot copies, enter the following command:

   snap list *sec_vol*

   sec_vol is the name of the WORM volume.

Example

The following output lists Snapshot copies on the *wormvol* volume:

```
system> snap list wormvol
Volume wormvol
working...


%/used    %/total    date name
--------  ---------- ------------ --------
0% ( 0%)  0% ( 0%)   May 13 19:56 sv_hourly.20050513_195442_GMT
(snaplock)
3% ( 2%)  1% ( 0%)   May 13 19:51 sv_hourly.20050513_195006_GMT
(snaplock)
5% ( 2%)  1% ( 0%)   May 13 19:12 sv_hourly.20050513_191043_GMT
(snaplock)
```

Listing Snapshot copies and retention dates

You can list Snapshot copies and retention dates.

Step

1. To list Snapshot copies and retention dates, enter the following command:

   snap list -l *volume*

   volume is the name of the WORM volume.

Example

The following output lists Snapshot copies and retention dates:

```
system> snap list -l wormvol
Volume wormvol
working...
```

```
snapshot date              retention date name
-------------------------- ------------------------ --------
May 13 19:56:50 2005 +0000 May 13 19:59:42 2005 +0000
sv_hourly.20050513_195442_GMT
May 13 19:51:07 2005 +0000 May 13 19:55:08 2005 +0000
sv_hourly.20050513_195006_GMT
May 13 19:12:06 2005 +0000 May 13 19:15:43 2005 +0000
sv_hourly.20050513_191043_GMT
```

Snapshot copies with a dash (-) in the retention date column are not WORM Snapshot copies; therefore, they do not have retention periods.

Deleting expired WORM Snapshot copies

You can delete WORM Snapshot copies that are retained beyond their retention periods. WORM Snapshot copies are not deleted automatically. You must delete them when they expire.

Step

1. To delete an expired WORM Snapshot copy, enter the following command:

   snap delete *volume snapshot*

   volume is the name of the WORM volume.

   snapshot is the name of the Snapshot copy.

   Note: You cannot delete unexpired WORM Snapshot copies.

Example

```
system> snap delete wormvol slminutely.20040104_122040_GMT
Illegal operation on snapshot locked by SnapLock.
```

How to retain more than 255 SnapVault Snapshot copies

The limit imposed by Data ONTAP for retaining Snapshot copies is 255. However, if you need to retain more than 255 Snapshot copies, you can do so by creating a new volume.

The practical limit for WORM Snapshot copies is approximately 250 for the following reasons:

- A few non-WORM base Snapshot copies and temporary Snapshot copies are used by SnapVault.
- Snapshot copies are needed for managing SnapMirror relationships if you have SnapVault secondary volumes that are protected by SnapMirror.

If you need to retain more than 250 Snapshot copies, you can do so by creating a new volume. As subsequent volumes reach the limit, you can create additional new volumes. In this manner, you can use multiple volumes to retain a larger number of Snapshot copies.

Next topics

How to create a new volume to retain more Snapshot copies on page 93

How to create a new volume to retain more Snapshot copies

If you need to create a new volume to retain more Snapshot copies than the maximum allowed, you can do so in one of the two ways—by creating a volume clone or by copying a Snapshot copy.

- Create a clone of the volume and continue running SnapVault updates to the new volume.
- Copy one Snapshot copy in the volume to a new volume and continue running SnapVault updates to the new volume.

  Note: If you are using a version of Open Systems SnapVault prior to version 2.2, you cannot retain more than 255 SnapVault Snapshot copies without a new baseline. Open Systems SnapVault versions prior to 2.2 do not support the `snapvault start -r` command, which is needed to restart SnapVault relationships on the new volume. This command is supported on Open Systems SnapVault 2.2 and later.

Advantages of cloning Snapshot copies

Using the cloning approach for creating a new volume to retain more than the 255 Snapshot copies has many advantages.

- Less disk space is used: Initially, the new clone volume does not occupy any disk space. Only changes are recorded on the new volume.
- Speed: Volume cloning is almost instantaneous. Copying data from a Snapshot copy takes time.
- Breaking the relationship is easy: If you want to break the relationship between the original volume and the clone, you can do so by using the `vol clone split` command.

Note: At the time of the split, additional disk space will likely be used.

Advantages of copying Snapshot copies

Using the copying approach for creating a new volume to retain more than the 255 Snapshot copies has certain advantages.

- Each volume is completely independent. You need not keep the old volumes online. If the old volumes are damaged or destroyed, your more recent volumes can still be used.
- SnapVault relationships can be migrated to another machine with volume copying.

Creating a new volume for retaining Snapshot copies

You need to create a new volume to retain more than 250 Snapshot copies.

Steps

1. Ensure that everything on the old volume is in an appropriate state.
2. Remove the SnapVault schedules for the old volume.
3. Create a new volume by doing one of the following steps:
   - Create a volume clone to a new volume.
   - Copy the appropriate Snapshot copy to a new volume.
4. Check or set the retention periods on the new volume.
5. Check volume options on the new volume.
6. Restart all SnapVault relationships in the new volume.
7. Reconfigure the SnapVault schedules in the new volume.
8. Ensure that everything was migrated successfully. To do this, run a few Snapshot copy targets in the new volume.
9. Stop all SnapVault relationships in the old volume.

   Note: Free Snapshot copies might be required to complete some operations. If you get an error message stating that the system has reached the Snapshot copy limit per volume, you need to find unnecessary Snapshot copies and delete them.

Next topics

Verifying the state of the old volume on page 95
Removing the SnapVault schedules for the old volume on page 96
Creating a volume clone to a new volume on page 96
Copying the appropriate Snapshot copy to a new volume on page 97
Checking or setting the retention periods on the new volume on page 98
Checking volume options on the new volume on page 99
Restarting all SnapVault relationships in the new volume on page 99
Reconfiguring the SnapVault schedules in the new volume on page 100
Ensuring the migration of Snapshot copies on page 100
Stopping all SnapVault relationships in the old volume on page 100

Verifying the state of the old volume

Before creating a new volume for Snapshot copy retention, you need to ensure that all the files in the old volume are in the `Idle` and the `snapvaulted` state.

Steps

1. Check that all relationships for the volume that you will clone or copy are in the `Idle` and `snapvaulted` state by entering the following command:

   `snapvault status`

If...	Then.....
The volume is in this state	Go to Step 3.
The volume is not in this state, but transfers are proceeding normally	Wait for the transfers to finish and then repeat this step.
Transfers are not proceeding or completed	Go to Step 2.

2. To check and correct the configuration information, enter the following command:

   `snapvault status -c`

   You need to perform this step because occasionally, the `snapvault status` command might show a relationship for which there is no permanent configuration information.

   a. Compare the output from the `snapvault status -c` command to the output from the `snapvault status` command.

      You should see a one-to-one correspondence. If any relationships are missing, make the output from the two commands consistent.

   b. Create the configuration entry for the missing relationship using the `snapvault start` command or, if the relationship is not needed, use the `snapvault stop` command.

      The `snapvault stop` command destroys the relationship.

3. Ensure that all qtrees with SnapVault relationships are at the same base Snapshot copy by entering the following command:

   `snapvault status -l`

   Check the base Snapshot copy listed for each qtree in the destination volume to ensure that they refer to the same Snapshot copy.

If.....	Then...
They refer to the same Snapshot copy	You are finished.
They do not refer to the same Snapshot copy	Go to Step 4.

4. To create a new base Snapshot copy for all qtrees in the volume, enter the following command:

snapvault snap create -w *sec_vol* ""

sec_vol is the current secondary volume.

" " specifies that no `snapvault snap sched` Snapshot copy is created.

Note: All qtrees should be in the `snapvaulted` state after the transfer initiated by the `snapvault snap create` command.

5. Go back to Step 1 and ensure that everything is in an appropriate state.

Removing the SnapVault schedules for the old volume

Before you remove the schedules, make a note of the schedule configuration, including retention periods. Doing so enables you to set up the schedule in the same way on the new secondary volume.

Steps

1. View the existing schedules by entering the following command:

   snapvault snap sched *sec_vol*

2. Remove the schedules by entering the following command:

   snapvault snap unsched *sec_vol*

   Note: If transfers to the secondary volume are currently in process, the `snapvault snap unsched` command fails. Wait for the transfers to finish and then repeat the command.

3. Verify that the files in the volume are either in the `idle` or `snapvaulted` state.

Related tasks

Verifying the state of the old volume on page 95

Creating a volume clone to a new volume

You can create a volume clone from a base Snapshot copy.

Steps

1. Ensure that you have licensed the volume cloning feature (flex_clone license) by entering the following command:

   license

2. To create a volume clone, enter the following command:

   vol clone create *altvol* -b *sec_vol base_snapshot*

   altvol is the new secondary volume.

   sec_vol is the old secondary volume.

base_snapshot is the Snapshot copy that you get by using the `snapvault status -l` command.

When you clone a SnapLock Compliance volume, you get the following error message:

```
Error: vol clone create: The source volume is a strict worm volume. If
you create a clone of this volume, you will not be able to delete it
until all of the WORM files on this new clone have expired. Use force
option to create clones of worm volumes. If you use force option, the
destroy time of the clone volume would be: dd mm yy.
```

Note: Clone volumes are in the same aggregate as the parent volume.

When you create a volume clone, you can ignore the messages appearing on the console.

Examples of possible messages

```
Reverting volume altvol to a previous snapshot.

Breaking snapmirrored qtree 1 in volume
altvol:
basesnapshot no longer exists.
Use snapmirror resync or initialize to re-establish the snapmirror.

Breaking snapmirrored qtree 2 in volume
altvol:
WAFL_check broke all snapmirrors in volume
wormclone1.Use snapmirror resync or
initialize to re-establish the snapmirror.
```

Copying the appropriate Snapshot copy to a new volume

When you are creating a copy of a volume, the copy must be of the same type (flexible or traditional) and same compliance type (SnapLock Compliance or SnapLock Enterprise) as the old volume.

Note: If the volume you are copying is a FlexVol volume, the copy does not have to be in the same aggregate as the old volume, nor does it have to be on the same storage system as the old volume.

Steps

1. To create a copy of the old volume, create the new volume by entering the following command:

   vol create *altvol*

   altvol is the new volume.

   Note: Include the appropriate volume type and sizing information.

2. Put the new volume in a restricted state by entering the following command:

   vol restrict *altvol*

3. Copy the base Snapshot copy from the old volume to the new volume by entering the following command:

   vol copy start -s *base_snapshot secsystem*:*secvol altvol*

   Note: The command assumes that the volume copy is initiated from the system that contains *altvol*.

4. Put the new volume online by entering the following command:

   vol online *altvol*

   When you create a volume copy, you can ignore the messages appearing on the console.

Examples of possible messages

```
Breaking snapmirrored qtree 1 in volume altvol: base snapshot no
longer exists. Use snapmirror resync or initialize to re-establish
the snapmirror.

Breaking snapmirrored qtree 2 in volume
altvol: WAFL_check broke all snapmirrors in volume wormclone1.Use
snapmirror resync or initialize to re-establish the snapmirror.
```

Checking or setting the retention periods on the new volume

Clone volumes and volume copies automatically inherit the retention periods from their parent volumes. For this reason, this step is usually a check. You can change the retention periods for the new volume.

Steps

1. To see the retention periods on the old volume, enter the following command:

   vol options *sec_vol*

2. To see the retention periods on the new volume, enter the same command for the new volume:

   vol options *altvol*

3. Compare the output from the two volumes to ensure that they are the same.

4. If you want to change the retention periods, enter one or more of the following commands:

   vol options *altvol* snaplock_default_period *period*

   vol options *altvol* snaplock_maximum_period *period*

   vol options *altvol* snaplock_miniumum_period *period*

   where *period* is the retention period.

   To learn about retention period values, see the na_vol(1) man page.

Checking volume options on the new volume

Clone volumes and volume copies automatically inherit all options from their parent volumes. Before starting any SnapVault activity on the new volume, you should check the volume options on the new volume against the volume options on the old volume.

Steps

1. To see all volume options for both volumes, enter the following commands:

   vol status -v *sec_vol*

   vol options *sec_vol*

   vol status -v *altvol*

   vol options *altvol*

   where *sec_vol* is the name of the old volume and *altvol* is the name of new volume.

2. Compare the output to ensure that all options, except for size-related options, are set to the same values.

   Note: It is especially important that the language settings be the same.

Restarting all SnapVault relationships in the new volume

The qtrees on the new volume are in the normal state, not the `snapvaulted` state (as you can see by using the `qtree status` command). You must restart the SnapVault relationships in the new volume.

Note: To support data integrity in a restrictive SnapLock Compliance environment, the `snapvault start -r` operation saves all data that was written after the common Snapshot copy to a directory on the volume; however, you typically cannot write data after the common Snapshot copy in this situation.

Steps

1. To see the SnapVault relationships in the old volume, enter the following command:

   snapvault status

   You can see the SnapVault relationships for all volumes. You can use the listed relationships to generate the relationships in the new volume.

2. To restart SnapVault relationships, enter the following command for each secondary qtree:

   snapvault start -r -S *pri_system*:/vol/*pri_vol*/*pri_qtree* /vol/*altvol*/*dest_qtree*

   For details about the `snapvault start` command, see the na_snapvault(1) man page.

Note: If you omit the `-r` option from the `snapvault start` command, the restart will fail.

Related concepts

How to resynchronize a broken SnapVault relationship on page 102

Reconfiguring the SnapVault schedules in the new volume

If you are not changing the SnapVault Snapshot schedule on the primary storage system, you can use the same names and schedule for Snapshot copies.

Step

1. To reconfigure the SnapVault schedules, enter the following command:

   snapvault snap sched -x -o retention_period=period *altvolsnapname sched_spec*

   To learn more about the `snapvault snap sched` command, see the na_snapvault man page.

   Note: Specifying the retention period is optional. If a retention period is not specified, the default retention period for the volume is used.

Ensuring the migration of Snapshot copies

You should run a few Snapshot copy targets in the new volume to ensure that everything was migrated successfully. Then, after the first scheduled Snapshot copy and transfer have occurred, look at the entries in the new volume.

Steps

1. To view the entries, enter the following commands:

   snapvault status

   snapvault status -c

2. Verify all the entries of Snapshot copies in the new volume.

Stopping all SnapVault relationships in the old volume

After verifying the migration of the Snapshot copy, you can delete the old SnapVault relationships.

Steps

1. From the system that contains the old volume, enter the following command for all the qtrees in the old volume:

   snapvault stop -f /vol/*sec_vol/dest_qtree*

2. From the primary storage system, enter the following command for all the qtrees backed up to the old volume:

   snapvault release /vol/*pri_vol*/*pri_qtree* *sec_system*:/vol/*sec_vol*/*dest_qtree*

   For details about the `snapvault release` command, see the na_snapvault(1) man page.

   The `snapvault stop` command deletes the qtree from the active file system on the old volume. Backup copies of the qtree are in the Snapshot copies on the old volume. You can access them if you want to browse the old backups or restore data from the old backups. The active image of the backup data is moved to the new volume when you restart the SnapVault relationship by using the `snapvault start -r` command.

Backup of the log volumes created by the SnapLock for SnapVault feature

Along with backing up SnapLock Compliance secondary storage volumes, you can provide additional protection for log volumes created by the SnapLock for SnapVault feature.

Next topics

Protecting a log volume of the SnapLock for SnapVault feature on page 101
Failing over to the standby system on page 101
Reestablishing standby protection on page 102

Protecting a log volume of the SnapLock for SnapVault feature

You can protect a log volume of the SnapLock for SnapVault feature. You do this by creating a SnapMirror backup with the same update schedule that you have for other SnapMirror relationships between the SnapVault secondary storage system and the SnapMirror destination.

Step

1. Create a SnapMirror relationship between the SnapLock for SnapVault log volume and a new SnapLock for SnapVault log volume on the new standby system.

   For more information about setting up a basic SnapMirror operation, see the *Data ONTAP Data Protection Online Backup and Recovery Guide*.

Failing over to the standby system

If you convert the standby system to a SnapVault secondary system, you need to configure the log volume for the SnapLock for SnapVault feature to continue with collecting log information.

Steps

1. Quiesce the SnapMirror relationship to the SnapLock for SnapVault log volume on the converted standby system by using the following command:

```
snapmirror quiesce volume_name
```

2. Break the SnapMirror relationship to the SnapLock for SnapVault log volume on the converted standby system by using the following command:

   ```
   snapmirror break volume_name
   ```

3. Configure the SnapLock for SnapVault log volume on the converted standby system by using the following command:

   ```
   options snapvault.lockvault_log_volume volume_name
   ```

Reestablishing standby protection

If you reestablish standby protection for new SnapVault secondary volumes, you should also reestablish new standby protection for the log volume of the SnapLock for SnapVault feature. As is the case with reestablishing standby protection for SnapVault secondary volumes, you need a new volume for the new standby log volume of the SnapLock for SnapVault feature.

Step

1. Create a SnapMirror relationship between the SnapLock for SnapVault log volume and a new SnapLock for SnapVault log volume on the new standby system.

How to resynchronize a broken SnapVault relationship

You can use the `snapvault start -r` command to reestablish a broken SnapVault relationship without a lengthy baseline transfer.

Typically, this command locates the most recent common Snapshot copy, discards any data written to the destination after that Snapshot copy, and begins to resynchronize content using the common Snapshot copy.

To support data integrity in a restrictive SnapLock Compliance environment, the `snapvault start -r` operation saves all data that was written after the common Snapshot copy to an image and log file in a directory on the volume. These files are then stored in the following directory on the SnapLock Compliance volume: `/etc/logs/snapmirror_resync_archive/volname_UUID_qtree`.

The image and log files are named `dump_image_`*`YYMMDD_HHMMSS`* and `dump_log_`*`YYMMDD_HHMMSS`*, respectively.

Note: The retention period for image and log files is equal to the longest retention period of any data file in the data set. This ensures that the image and log files are not be deleted before the retention period has passed.

If the save operation fails for any reason, the `snapvault start -r` transfer operation does not proceed.

Turning SnapVault off

You might want to turn SnapVault off because the files are no longer important or current or have been moved to another location.

Step

1. On both the primary storage system and the secondary storage system, enter the following command:

   ```
   options snapvault.enable off
   ```

   This option persists across reboots.

Management of SnapVault log files

You can manage SnapVault log files for compliance.

Next topics

Regulatory compliance and SnapVault log files

SnapVault creates log files that meet the requirements for regulatory compliance.

Following are the requirements for regulatory compliance:

- You cannot delete a log file.
- You cannot delete or overwrite the contents of log files.
- The log files must accurately record the exact series of events that occur during a time frame.

How SnapVault maintains compliance

SnapVault uses a SnapLock volume in which the log files are kept. This volume, called SnapLock for SnapVault log volume, is a standard WORM volume. SnapVault uses an append-only write

operation to write to the log files. This allows accurate record keeping but does not allow previous events to be overwritten.

SnapVault uses two types of log files to record events:

- SnapVault operations log files
- SnapVault files-transferred log files

Next topics

Operations log file on page 104
Files-transferred log files on page 104

Operations log file

Operations log files have a weekly rotation policy; SnapVault creates a new log file every Sunday morning at midnight. Any SnapVault operations that are active when the new log file is created are recorded in the new log file.

The following is the format for an operations log file name:

`snapvault.`*`yyyymmdd_zzz`*

`yyyy` is the year.

`mm` is the month (01 to 12).

`dd` is the date (01 to 31).

`zzz` is the current time zone setting.

The timestamp denotes when the log file is created. The timestamp is generated using the system clock.

Files-transferred log files

SnapVault creates a new files-transferred log file at the beginning of each SnapVault transfer.

Files-transferred log files are kept in the following directory structure in the log volume created by the SnapLock for SnapVault feature : `/etc/logs/`*`snapvault secondary vol/secondary qtree name/month/log file`*.

The `snapVault secondary vol` directories are the SnapLock volume names on the secondary storage system. The following is the format of the directory name: *`secondary volume name_volume id`*. The volume UUID uniquely identifies a volume globally.

The secondary qtree name directories are the qtrees within a SnapVault secondary vol directory. The following is the format of the secondary qtree directory name: *`secondary qtree name_treeid_inodenum_gennum`*. The `tree ID`, inode number, and generation number uniquely identify the qtrees within a volume.

The month directory denotes the month in which log files for a qtree were generated. The format for the month directory is as follows: *yyyymm*. The year (*yyyy*) and month (*mm*) are generated using the system clock.

The log file name uniquely identifies each transfer log file. The following is the format of the transfer log file name: `snapvault_filelog.yyyymmdd_hhmmss_zzz`.

The year (`yyyy`), month (`mm`), day (`dd`), hour (`hh`), minute (`mm`), second (`ss`), and time zone (`zzz`) are generated using the system clock.

Example

The following is a sample path for a transfer log file named `snapvault_filelog.20040123_000500_PST`:

```
/etc/log/vault_ ed0ad520-5f40-11d8-91ca-00a09800dcba/
users_19_1435628_1286197/200401/snapvault_filelog.20040123_000500_PST
```

Configuring the log volumes of the SnapLock for SnapVault feature

A log volume for the SnapLock for SnapVault feature must be a SnapLock volume. The SnapLock volume can be a traditional volume or a flexible volume. All SnapVault log entries are created in the log volume. This is a system-wide option; therefore, a SnapVault operation on any SnapLock volume is logged to the log volume. However, you need to set this option for SnapVault transfers to SnapLock volumes to succeed.

Before you begin

You must create a log volume for the SnapLock for SnapVault feature before starting an initial (level 0) SnapVault transfer.

Step

1. To configure the log volume, enter the following command:

   options snapvault.lockvault_log_volume *volume_name*

   volume_name is a SnapLock volume.

   Note: You must use the name of a previously created SnapLock volume, and this volume should not be used for any other purpose.

Where the log files are kept

All SnapVault log files are kept in the log directory of the SnapLock for SnapVault log volume that you created and configured.

The SnapVault log files are kept in the `/etc/log` directory of the SnapLock for SnapVault log volume. Log files in this directory inherit the default retention period of the log volume. Log files

cannot be deleted from the volume until their retention periods have expired. SnapVault does not remove expired log files automatically; you must delete them manually.

Note: Ensure that you set the default retention period on the log volume appropriately. The initial default retention period is 30 years for SnapLock Compliance volumes.

What files-transferred log files contain

A transfer log contains a header that describes the transfer and zero or more entries that describe the contents of the transfer.

A log file contains only a header and zero entries in two instances: when no data on the primary storage system has changed since the last SnapVault update or for a rollback transfer.

The header at the beginning of each log file contains the following information:

- Transfer type
- Source system and source path
- Destination system and destination path
- Date and time of the start of the transfer
- Date and time of the source system Snapshot copy

Note: Date and time of the source system Snapshot copy is interpreted according to the time zone settings on the secondary system, not the primary system.

Example

```
# Transfer type: Base Start
# From: sourcesystem:/vol/compat_data/myqtree1
# To: destinationsystem:/vol/logs/mult1
# Start time: Tue Mar 30 22:43:09 GMT 2004
# Source snapshot timestamp: Wed Mar 31 23:41:01 EST 2004
```

Types of log entries recorded

All create, delete, modify, and rename operations are recorded in the transfer log files.

- Create file (regular, special, or stream)
- Delete file (regular, special, or stream)
- Create directory
- Delete directory
- Rename from (regular file or directory)
- Rename to (regular file or directory)

- Modify file (regular, special, or stream file, but not directory)
- Modify attributes (regular, special, stream file, or directory)

Log entry format

The log entry format is as follows:

`date_and_time action_taken base_path_length`:*`stream_name_length  path_name`*

`date_and_time` is the `mtime` or `ctime`ctime value from the primary storage system, not the secondary storage system.

Note: The `mtime` value is used to create file and directory types, delete file and directory types, and modify file types because these entry types modify data. The `ctime` value is used to rename file and directory types and modify attribute types because these types modify the container for the data, not the data itself.

`action_taken` is one of the types of log entries.

`base_path_length`:*`stream_name_length`* is the length of the base path followed by a colon and the stream name in single byte characters. Only the base path length is shown if the file is not a stream file.

`path_name` is the file path relative to the base of the tree that is mirrored, not the absolute path name. The relative path name is the same on the primary and secondary storage systems.

Example

The following are examples of log entries:

```
Fri Mar 26 23:08:33 GMT 2004 Create Dir    7   ./mydir
Mon Oct  9 17:36:14 GMT 2000 Create File  14   ./mydir/myfile
Mon Jun 12 22:21:11 GMT 2000 Create File  14:8 ./mydir/myfile:strm
```

How log entries are created

Certain actions cause SnapVault to create a log entry type. The following table lists actions that cause SnapVault to create a log entry, the type of log entry created, and possibly a note about the action.

Action	Entry type	Note
The initial transfer of all files and directories.	Create File	
	Create Directory	
The data in a file was modified.	Modify File	A Modify Attributes entry is not created if a Modify File entry was.

Action	Entry type	Note
The attributes of a directory, such as permissions, were changed or any entries in the directory were added or deleted.	Modify Attributes (directory)	
An access control list (ACL) for a file or directory was created, deleted, or changed.	Modify Attributes (any file or directory associated with the ACL)	ACL creation, deletion, or modification is not explicitly logged.
A file or directory with only one link was renamed.	Rename From	Renaming creates two entries that appear together in the log.
	Rename To	
A file was renamed.	Modify Attributes or Modify File in addition to Rename From and Rename To	A Modify Attributes entry is created if no data was modified. A Modify file entry is created if data was modified.
A directory was renamed.	Modify Attributes in addition to Rename From and Rename To	
A hard link was added to an existing file.	Create File	
A hard link to a file was removed and the file still has one or more hard links.	Delete File	
A hard link was added to a file, but no content was changed.	Modify Attributes for all links to the file except the new link	
A hard link was added to a file and content was changed.	Modify File for all links to the file except the new link	
A hard link was deleted from a file, but no content was changed.	Modify Attributes for all links to the file except the deleted link	
A file with one or more hard links to it was created between the previous transfer and the current transfer.	Create File	No Modify File or Modify Attributes entry for the file.
A file and all its hard links were deleted between the previous transfer and the current transfer.	Delete File	No Modify File or Modify Attributes entry for any links.

Action	Entry type	Note
A file with multiple hard links was renamed.	Create File	A file with hard links is renamed by adding one link to the file and deleting one link from the file.
	Delete File	
	Modify Attributes or Modify File	
Data or attributes of a file were modified.	Modify File or Modify Attributes, respectively	A Modify File or Modify Attributes entry is created for each link to the file except links that were added or deleted by the transfer.

How to provide backup and standby protection using SnapMirror

By setting up a SnapMirror relationship between the SnapVault secondary storage system and a SnapMirror destination or NearStore system, you can provide backup and standby protection for the SnapVault secondary storage system data.

Following is the process for backup and standby protection for the secondary storage system data:

- Use the SnapMirror destination device as a standby device to be activated as an alternative SnapVault secondary storage if the original secondary storage device goes down.

  Note: The SnapMirror destination systems cannot take advantage of the disk-space-saving benefits of volume cloning. If you used volume cloning on the original SnapVault secondary volume to store more than 255 Snapshot copies, which is the limit imposed on volumes by Data ONTAP, the SnapMirror destination volumes might need to be larger than the SnapVault secondary volumes if they are going to store the same amount of data.

- Reestablishing backup and standby protection using this procedure, you have your original SnapMirror destination volumes as your SnapVault secondary volumes, and new volumes as your SnapMirror destination volumes.

  Note: You could use the original SnapVault secondary volumes when they become available after the retention periods of Snapshot copies and files on the volumes expire, but retention periods are usually long enough that the volumes are not likely to be available.

- An optional step, but one that you should avoid performing, is to return to the original backup and standby configuration. After you reestablish the backup and standby configuration, your SnapVault secondary volumes are protected with a SnapMirror replication, but the two backup systems are reversed (the system that was your standby is now your secondary and the system that was your secondary is now your standby). Instead of returning to the original configuration, it is better that the storage systems you use for your SnapVault secondaries and their associated

SnapMirror standbys have the same configuration such that their roles can be reversed, if necessary.

Next topics

Setting up backup and standby protection for SnapVault on page 110
Reestablishing backup and standby protection for SnapVault on page 111
Returning to the original backup and standby configuration on page 111
Limitations to compliance backup and standby service on page 112

Setting up backup and standby protection for SnapVault

You can set up the SnapMirror backup and standby protection for the SnapVault secondary storage system and fail over to the standby system, if needed.

Steps

1. To confirm that the SnapVault secondary storage device has both SnapVault secondary storage and SnapMirror features licensed, enter the following command:

   `license`

2. To confirm that the SnapMirror destination device has both the SnapVault secondary storage and SnapMirror features licensed, enter the following command:

   `license`

3. Set up SnapMirror replication from the active SnapVault secondary storage system to a disk-based destination device (another storage or NearStore system).

   For more information about setting up a SnapMirror relationship with a disk-based destination device, see the *Data ONTAP Data Protection Online Backup and Recovery Guide*.

4. If the active SnapVault secondary storage 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 storage systems.

   For details on converting a SnapMirror destination to a writable volume, see the *Data ONTAP Data Protection Online Backup and Recovery Guide*.

5. Activate the SnapVault license on the new SnapVault systems, and use the `snapvault start` and `snapvault snap sched` commands to complete configuration of the new SnapVault secondary storage system.

6. As an optional step, you can return the storage systems to their original configuration.

Reestablishing backup and standby protection for SnapVault

To reestablish SnapMirror backup and standby protection for the new SnapVault secondary storage system, you need to perform an initial SnapMirror transfer from the SnapVault secondary to a new volume.

Step

1. To perform an initial SnapMirror transfer from the new SnapVault secondary volume to the new volume, enter the following command:

   `snapmirror initialize`

   Attention: Using the `snapmirror initialize` command destroys all existing data on the new volumes. For more information about setting up a basic SnapMirror operation, see the *Data ONTAP Data Protection Online Backup and Recovery Guide*.

   If you are able to use your original SnapVault secondary volumes as the new SnapMirror destinations and you used volume cloning to go beyond 255 Snapshot copies, which is the limit imposed on a volume by Data ONTAP, the original SnapVault secondary volumes might not be large enough to accommodate the replication. If this is the case, you can add disk space to the original SnapVault secondary volumes.

Returning to the original backup and standby configuration

You can return to the original backup and standby configuration.

Steps

1. Break the SnapMirror relationship.
2. If the active SnapVault secondary storage system is damaged or destroyed, convert the SnapMirror destination device to an alternative SnapVault secondary system to carry on the task of backing up data from the primary storage systems.

   For details on converting a SnapMirror destination to a writable volume, see the *Data ONTAP Data Protection Online Backup and Recovery Guide*.
3. Activate the SnapVault license on the new SnapVault system, and use the `snapvault start` and `snapvault snap sched` commands to complete configuration of the new SnapVault secondary storage system.
4. Perform an initial SnapMirror transfer from the new SnapVault secondary volume to the new volume using the following command:

   `snapmirror initialize`

Attention: Using the `snapmirror initialize` command will destroy all existing data on the new volumes. For details on converting a SnapMirror destination to a writable volume or qtree, see the *Data ONTAP Data Protection Online Backup and Recovery Guide*.

If you are able to use your original SnapVault secondary volumes as the new SnapMirror destinations and you used volume cloning to go beyond 255 Snapshot copies, which is the limit imposed on a volume by Data ONTAP, the original SnapVault secondary volumes might not be large enough to accommodate the replication. If this is the case, you can add disk space to the original SnapVault secondary volumes.

Limitations to compliance backup and standby service

There are limitations to the backup and standby service.

- After you have failed over to the standby device and have begun to take backups, you cannot reuse the original SnapVault secondary storage disks for protecting the new SnapVault secondary volumes until the retention periods for all Snapshot copies and data on the original SnapVault secondary storage disks have expired.

  For all practical purposes, the original SnapVault secondary storage disks cannot be used to protect the new SnapVault secondary volumes and cannot be used to resume their original role as SnapVault secondary storage because typical retention periods are on the order of months to years.
- You cannot restore backup data to a SnapVault secondary storage if the secondary storage uses SnapLock Compliance volumes, because of the constraints put on SnapLock Compliance volumes to eliminate the possibility of losing compliance data.

  Note: Failing over to a standby device is complex, has the limitations mentioned above, and is costly in terms of time and disk space. Failing over should be done only in the event of a permanent loss of the original SnapVault secondary storage when the quick resumption of backups is critical enough to warrant the cost of reestablishing the original configuration.

Data ONTAP command-line interface commands in SnapLock

The following table lists the SnapLock command-line interface commands.

Input name	Description
`snaplock help [cmd]`	Shows the command-line interface help.
`vol create trad_vol -v -L snaplock_type ndisks [@disksize]`	Creates a SnapLock traditional volume.
`aggr create aggrname -L snaplock_type ndisks [@disksize]`	Creates a SnapLock aggregate.
`vol create wormvol aggrname size [k|m|g|t]`	Creates a SnapLock FlexVol volumes.
`vol status vol_name`	Displays the status of a volume.
`vol destroy vol_name`	Destroys the SnapLock volume.
`vol status -w`	Displays the expiry date of a SnapLock Compliance volume.
`vol offline vol_name`	Takes the volume offline.
`vol clone create -f`	The `-f` flag is used to force the creation of a clone on SnapLock volumes.
`aggr statusaggrname`	Displays the status of an aggregate.
`options snaplock.compliance.write_verify on`	Enables SnapLock write verification.
`date -c initialize`	Initializes the ComplianceClock.
`date -c`	Displays the ComplianceClock.
`options snaplock.autocommit_period none|count(h|d|m|y)`	Sets the autocommit time delay.
`options snaplock.autocommit_period`	Displays the autocommit time delay.
`snaplock log archive vol [basename]`	Archives active SnapLock log file *basename* if *basename* is specified. Otherwise archives all active SnapLock log files on volume *vol* and replaces with a new one.
`snaplock log volume [-f] vol`	Sets the SnapLock log volume to *vol*.
`snaplock log volume`	Displays the current SnapLock log volume.

Input name	Description
`snaplock log status` *`vol`* [*`basename`*]	Displays the status of active SnapLock log file *`basename`* if the *`basename`* is specified. Otherwise shows status of all active SnapLock log files on volume *`vol`*.
`options snaplock.log.maximum_size`	Displays the maximum permissible size of the log file.
`snaplock options` *`vol`* `privdel on`	Sets the state `on` of privileged delete option on the SnapLock Enterprise volume *`vol`*. Enables the privileged delete functionality.
`snaplock options` *`vol`* `privdeloff`	Sets the state `off` of privileged delete option on the SnapLock Enterprise volume *`vol`*.
`snaplock options -f` *`vol`* `privdel disallowed`	Sets the state `disallowed` of privileged delete option on the SnapLock Enterprise volume *`vol`*. The `-f` flag is required to set the state to disallowed to prevent operator error. The `-f` flag is ignored if it is used to set the option to any other state.
`snaplock options` *`vol`* `privdel`	Shows the state of the privileged delete option on the SnapLock Enterprise volume. The state can be either `on`, `off`, or `disallowed`.
`snaplock privdel [-f]` *`path`*	Deletes unexpired files on a SnapLock Enterprise volume. The `-f` flag causes the command to proceed without interactive confirmation from the user.
`file fingerprint [-a {MD5|SHA-256}] [-m ][-d][-x]` *`path`*	Calculates the fingerprint of the file using MD-5 or SHA-256 digest algorithm. `-a` specifies algorithm `-m` specifies metadata fingerprint calculation `-d` specifies data fingerprint calculation `-x` displays the output in XML format

How to manage SnapLock through Data ONTAP APIs

An application can integrate seamlessly with SnapLock using Data ONTAP APIs. SnapLock is an open solution that uses standard protocols to enable seamless integration with ISV archival applications as well as custom applications. This integration with specific vertical applications allows data immutability to become a part of your data workflow, thereby eliminating the need to separately manage WORM locking and the eventual deletion of data to meet regulations.

You can leverage the SnapLock features through the command line interface (CLI) or programmatically by using Data ONTAP APIs.

You can use the following two options to integrate with SnapLock:

- Data ONTAP APIs
- The native API available in the programming language that the application is written in

What ONTAPI is

You can use ONTAPI (also known as Data ONTAP APIs) are used to access and manage the storage system. Data ONTAP APIs are invoked in the form of XML. You can use the HTTP, HTTPS, and Windows DCE/RPC protocols to query Data ONTAP APIs.

Data ONTAP APIs can manage the following features of Data ONTAP:

- Setup and management of storage objects
- Quota/user management
- Device configuration
- Discovery of devices, aggregates, volumes
- Monitoring the health of the storage system, disk/volume capacity, performance
- Alerts/notifications
- License management
- Security
- Block protocols
- Data backup and recovery
- Data replication
- Archival and compliance of data
- File access protocols

The complete ONTAPI lists are available in the `doc/ontapi` directory of Manage ONTAP SDK installations, with subdirectories for each release. The ONTAPI versions pertaining to each major release of Data ONTAP are available in the Version Matrix provided with Manage ONTAP SDK.

You can invoke Data ONTAP APIs by using Manage ONTAP SDK. Manage ONTAP SDK contains sample code that demonstrates the use of ONTAPI in C/C++, Java, C#, VB.Net, Powershell, and Perl. For more information about using ONTAPI through Manage ONTAP SDK, see Manage ONTAP SDK.

Related information

Manage ONTAP SDK: http://communities.netapp.com/docs/DOC-1110

Setting up a client to use ONTAPI calls

Certain steps are involved in setting up a client to use ONTAPI calls.

About this task

Note: All the examples in this topic are shown in C/C++.

Steps

1. Use a client program to initialize the server context.

   In Perl and Java, initializing the server context occurs automatically apart from declaring the libraries and as part of creating a server object. In Perl, declaring the use of `NaElement` is optional because `NaServer` already does that as part of its own initialization.

   Example

   ```
   #include <netapp_api.h>

   ...

   char    err[256];

   if (!na_startup(err, sizeof(err))) {

       fprintf(stderr, "Error: %s\n", err);

       exit(-1);

   }
   ```

2. Identify which server (storage system or NetCache appliance) you want to communicate with, and the version of the ONTAPI library you expect.

   This setup gives you a pointer to an na_server_t* value (C/C++) or an object (Perl/Java) that is a server context used for subsequent ONTAPI invocations.

   Example

   ```
   na_server_t*    s;


   /*
    * server to talk to, and
    * ONTAPI version, in this case 1.3 or higher
    */
   ```

```
s = na_server_open("jetfighter",1,3);
```

3. Set the transport parameters.

   If you are writing a Windows application in C, you normally use Windows DCE/RPC, which furnishes native Windows authentication and authorization. On the other hand, if you must use HTTP (which you must always do with Perl and Java), you must furnish a user name and password for use by the server context, or have the server context authenticate against the `/etc/hosts.equiv` file on the storage system. This file consists of either host names, in which case everyone coming from that host is allowed in as root, or host name/user name pairs, in which case the named users are allowed in as root when they connect from the named host. The *Data ONTAP System Administration Guide* has more details on the `/etc/hosts.equiv` file. When using `HOSTSEQUIV` login style, you must set the `httpd.admin.hostsequiv.enable` option on the storage system to `on`.

   Example

```
bool          use_rpc = 1;
bool          use_hosts_equiv = 0;

if (use_rpc) {
    na_server_set_style(s, NA_STYLE_RPC);
}
else if (use_hosts_equiv) {
    na_server_set_style(s,
                    NA_STYLE_HOSTSEQUIV);
}
else {
    na_server_set_style(s,
                    NA_STYLE_LOGIN_PASSWORD);
    na_server_set_admin_user(s,
                    "root", "F18fan");
}
```

4. Check for the ONTAPI library version before invoking an ONTAPI library.

   Use `system-get-ontapi-version` to obtain the ONTAPI version of a storage system.

   Example

```
na_elem_t*      out;

/*

 * Make sure version is available

 */

out = na_server_invoke(s,

          "system-get-version", NULL);

if (na_results_status(out) != NA_OK) {
```

```
    printf("Version 1.3 is unsupported: %s\n",
           na_results_reason(out));
}


na_free(out);
```

5. Close the server context to free memory and resources. (The Perl bindings take care of this when the context goes out of scope.)

Example

```
na_server_close(s);
```

Setting up the client is complete.

Benefits of using the Data ONTAP API suite

The Data ONTAP API suite is a framework of methods that you can use from external applications to perform all the functions on a storage system running Data ONTAP.

Following are some of the advantages of using the Data ONTAP API suite:

- Access to Data ONTAP features through APIs
- Support for multiple platforms such as Windows, Linux, Solaris, VMware ESX, and so on
- Support for multiple languages such as C, C++, Java, Perl, and so on
- Support for multiple transport protocols such as HTTP, HTTPS, and Windows RPCs
- Support for different authentication mechanisms such as login/password, Windows RPC authentication, and so on
- Support for multithreading

List of SnapLock APIs

An application can use SnapLock–related Data ONTAP APIs to integrate with the SnapLock functionality.

Following is a list of SnapLock-related Data ONTAP APIs:

- volume-create
- file-get-snaplock-retention-time
- file-get-snaplock-retention-time-list-info-max
- file-set-snaplock-retention-time
- file-snaplock-retention-time-list-info
- snaplock-get-compliance-clock
- snaplock-get-log-volume
- snaplock-get-options
- snaplock-log-archive
- snaplock-log-status-list-info
- snaplock-privileged-delete-file
- snaplock-set-log-volume
- snaplock-set-options
- file-get-fingerprint

Next topics

volume-create

The `volume-create` API creates a new flexible, traditional, or sparse volume with the given name and characteristics. Freshly-created traditional volumes may not be operational immediately after the API returns. You can use the `volume-list-info` API to query the status of the newly-created volume in order to determine when it is fully operational.

For more information about the input name and the output name, refer to *Manage ONTAP SDK* .

file-get-snaplock-retention-time

The `file-get-snaplock-retention-time` API gets the SnapLock retention attributes of a file.

For more information about the input name and output name of this API's parameter, see the *Manage ONTAP SDK*.

file-get-snaplock-retention-time-list-info-max

The `file-get-snaplock-retention-time-list-info-max` API gets the maximum number of entries that can be processed and returned in one call to the `file-snaplock-retention-time-list-info` API.

For more information about the output name of this API's parameter, see the *Manage ONTAP SDK* .

file-set-snaplock-retention-time

The `file-set-snaplock-retention-time` API sets the SnapLock retention attributes of a file.

For more information about the input name of this API's parameter, see the *Manage ONTAP SDK*.

file-snaplock-retention-time-list-info

The `file-snaplock-retention-time-list-info` API gets the SnapLock retention attributes for a list of files.

For more information about the input name and output name of this API's parameter, see the *Manage ONTAP SDK*.

snaplock-get-compliance-clock

The `snaplock-get-compliance-clock` API gets the SnapLock ComplianceClock date and time.

For more information about the output name of this API's parameter, see the *Manage ONTAP SDK* .

snaplock-get-log-volume

The `snaplock-get-log-volume` API gets the active SnapLock log volume configuration.

For more information about the output name of this API's parameter, see the *Manage ONTAP SDK* .

snaplock-get-options

The `snaplock-get-options` API gets the value of a given SnapLock option on a volume.

For more information about the input name and output name of this API's parameter, see the *Manage ONTAP SDK* .

snaplock-log-archive

The `snaplock-log-archive` API archives the active SnapLock log file. This API closes the current log file for furthur updates and opens a new log file to write future log updates.

For more information about the input name of this API's parameter, see the *Manage ONTAP SDK* .

snaplock-log-status-list-info

The `snaplock-log-status-list-info` API provides the status of the active WORM log file.

For more information about the input name and output name of this API's parameter, see the *Manage ONTAP SDK* .

snaplock-privileged-delete-file

The `snaplock-privileged-delete-file` API executes a privileged delete on a SnapLock file.

For more information about the input name of this API's parameter, see the *Manage ONTAP SDK* .

snaplock-set-log-volume

The `snaplock-set-log-volume` API sets the active log volume configuration.

For more information about the input name of this API's parameter, see the *Manage ONTAP SDK*.

snaplock-set-options

The `snaplock-set-options` API sets the SnapLock options on a volume.

For more information about the output name of this API's parameter, see the *Manage ONTAP SDK* .

file-get-fingerprint

The `file-get-fingerprint` API gets the fingerprint or digest of the file. The fingerprint is calculated using MD5 or SHA-256 digest algorithm. The fingerprint is calculated over the file data or metadata or on both data and metadata depending on the scope that you have selected. The fingerprints are base64 encoded.

Data fingerprint is calculated over file contents and the metadata fingerprint is calculated over the selected attributes of the file. Following are the attributes used for metadata fingerprint calculations:

- file type (file-type)
- file size (file-size)
- file crtime (creation-time)
- file mtime (modified-time)
- file ctime (changed-time)
- file retention time (retention-time, is-wraparound)
- file uid (owner-id)
- file gid (group-id)

Note: The file retention time is applicable only to WORM protected files.

For more information about the input name and output name of this API's parameter, see the *Manage ONTAP SDK* .

What the extended date range mechanism is

Data ONTAP denotes time as a signed 32-bit integer that is interpreted as the number of seconds since 1 January 1970, 00 hours 00 minutes 00 seconds (GMT). This interpretation imposes an upper limit of 03 hours 14 minutes 07 seconds on 19 January 2038 (GMT). The extended date range mechanism remaps the dates in the range 2038 to 2071 to the date range 1970 to 2003.

To support an extended range for record retention dates, SnapLock provides a mechanism by which applications can specify retention dates up to 19 January 2071 (GMT). This is accomplished by defining a new epoch for the last access time of 1 January 2003 00 hours 00 minutes 00 seconds (GMT). Instead of setting off the entire time range, the last access times before 1 January 2003 (GMT) are interpreted as dates in the future, using a wraparound technique. Therefore, all retention dates between 1 January 2003 and 19 January 2003 remain identical to the regular format. However, the dates between 1 January 2038 and 19 January 2071 are encoded as past dates.

For example, a file with a record retention date of 1 January 2040 displays the retention date as 1 January 1972.

Setting files to WORM state from an application

An application can be integrated with SnapLock to the extent that it manages individual files, is able to set or extend the retention periods of the files, and is able to commit the files to WORM state by transitioning them from writable to read-only state. An application can set the retention period on a file by altering the atime parameter when using NFS and the Date Accessed parameter when using CIFS.

Before you begin

- SnapLock license is installed.
- ComplianceClock is initialized.
- SnapLock volume is created.
- Files in the SnapLock volume are in the writable state.

Steps

1. Create a storage system connection.
2. Set the retention period of a file in seconds or by dates.

   If you must provide the retention period in seconds, it should be in UNIX time_t format.

To calculate the retention period...	Then...
In seconds	**a.** Get the ComplianceClock time by using the `snaplock-get-compliance-clock` API. **b.** Verify if the range of the retention period needs an extension: • If the seconds <= (0x7FFFFFFF - ComplianceClock time), use the normal range (retention_period = seconds + ComplianceClock time). • If the seconds >= (0x7FFFFFFF - ComplianceClock time), use the extended range (retention_period = seconds + ComplianceClock time+1-0x7FFFFFFF).
By dates	**a.** Convert the date to UNIX time_t format (number of seconds since 1 January 1970). This date is the retention period. **b.** Verify if the range of this retention period needs an extension: • If the retention period > 0x7FFFFFFF, use the extended range (retention_period= retention_period + 1 - 0x7FFFFFFF).

3. Set the file to WORM state using the retention period calculated in Step 2.

4. Close the storage system connection (if required).

Result

The file is set to WORM state.

Related concepts

What the extended date range mechanism is on page 129

Using SnapLock volume defaults to set retention periods

If the requirement is to set the same retention period on all or a good majority of files in a SnapLock volume, you can use the SnapLock volume defaults—minimum retention period, maximum retention period, and default retention period—thereby minimizing changes to an application. You can use the `volume-set-option` API to set the three retention periods.

Before you begin

- SnapLock license is installed.
- ComplianceClock is initialized.
- SnapLock volume is created.

Steps

1. Create the storage system connection.
2. Set the default period for the volume by using the `volume-set-option` API and providing the following values for the corresponding input elements:
   - volume: the SnapLock volume name.
   - option-name: snaplock_default _period.
   - option-value: the default period value in days, months or years.

   Note: You can also use the `volume-set-option` to set the minimum and maximum retention period of the files.
3. Close the storage system connection (if required).

After you finish

You must convert the files to WORM state to get a retention period equal to the `snaplock_default_period`.

By setting the SnapLock volume options, you need not explicitly specify a retention period for a file unless a retention period other than `snaplock_default_period` is desired.

Related concepts

How the SnapLock volume retention period works on page 37

Using the SnapLock autocommit feature from an application

You can use the autocommit feature of SnapLock to automatically commit the files in SnapLock volumes to WORM state without having to change the application. A file that is automatically committed to WORM state gets a retention period equal to the SnapLock volume's default retention period. The autocommit feature is a system-wide option and, once enabled, affects every SnapLock volume on the storage system

Before you begin

- SnapLock license is installed.
- ComplianceClock is initialized.
- SnapLock volume is created.

Steps

1. Create the storage system connection.
2. Set the autocommit period by using the `options-set` API and providing the following values for the corresponding input elements:
   - name: snaplock.autocommit_period.
   - value: the value of the autocommit period in value [h|d|m|y] (hours, days, months or years) format.
3. Close the storage system connection (if required).

Related concepts

How SnapLock automatically commits files to WORM state on page 35

How to implement SnapLock features through Data ONTAP APIs

Applications can leverage the SnapLock features privileged delete and WORM append files by using Data ONTAP APIs. Applications can take advantage of these SnapLock features to further enhance your overall solution.

Using the SnapLock privileged delete feature from an application

You can set your application to perform privileged delete operations by using the Data ONTAP API suite.The privileged delete functionality is available only on a SnapLock Enterprise volume. It allows a privileged user to delete a file that is otherwise immutable because of a retention policy . (However, you cannot delete expired WORM files.) A privileged user should be part of the Compliance Administrators group.

Before you begin

- SnapLock license is installed.
- ComplianceClock is initialized.
- SnapLock volume is created.

Steps

1. Create a privileged user who is part of the Compliance Administrators group.

   You can use the `useradmin-user-add` Data ONTAP API and specify the new user information and membership to the Compliance Administrators group through the input parameters.

2. Enable a compliance log volume.

   The log volume must be of type SnapLock compliance. You can use the `volume-create` Data ONTAP API from an application and specify that the volume is a SnapLock Compliance volume.

3. Set the newly created volume as the compliance log volume.

   You can use the `snaplock-set-log-volume` Data ONTAP API and specify the newly created volume name through the input parameters.

4. Enable the privileged delete functionality on the SnapLock Enterprise volume from the CLI.

   You can use the `snaplock-set-options` Data ONTAP API and specify the option as `privdel` and the option value as `on`.

5. Log in over SSH.

   You must ensure that SSH or Telnet sessions are kept separate from other sessions.

6. Delete a file on the SnapLock Enterprise volume.

   You can use the `snaplock-privileged-delete-file` Data ONTAP API and specify the `do it` flag as `true` and the file path.

Related concepts

What the privileged delete feature is on page 43

Using the SnapLock logging feature from an application

SnapLock logging is an infrastructure for logging SnapLock events and is useful for audits. The log files behave like an audit trail providing track, trace, and reporting capabilities. SnapLock events are recorded in the SnapLock log files, which are protected from tampering, are not writable by any external applications, and reside on a SnapLock Compliance volume as WORM files. SnapLock logging occurs whenever there is privileged delete activity.

Before you begin

- SnapLock Compliance license is installed.
- ComplianceClock is initialized.
- SnapLock volume is created.

Steps

1. Assign a log volume.

   You can use the `snaplock-set-log-volume` Data ONTAP API and specify the name of the log volume through the input parameters.

2. Archive the log file.

   Archiving, in this context, means committing a log file to WORM state. You can use the `snaplock-log-archive` Data ONTAP API and specify the log basename and log volume through the input parameters.

3. View the status of the log file.

   You can use the `snaplock-log-status-list-info` Data ONTAP API and specify the log basename and log volume through the input parameters.

Related concepts

What SnapLock logging is on page 49

What event-based retention is

Event-based retention refers to retaining a record based on the occurrence of an event. A record should be deleted either when the event occurs or a certain number of years after the event occurs.

For example: If the retention period of an insurance contract is set for Date of the Insured's Death plus 10 years, the event is triggered when the insured dies on 5 March 2010. Therefore, based on the event, the contract must be retained until 4 March 2020 and can be deleted on or after 5 March 2020.

Note: While SnapLock does not have a native event-based retention capability, it is possible to implement this capability by using the native features of SnapLock.

Related tasks

Implementing event-based retention and legal hold on page 149

What legal hold is

Legal hold refers to retaining a record because a hold order is in place, probably because an inquiry is being conducted. The record must be retained until the hold expires. Legal hold supersedes event-based retention.

For example: If the retention period of an insurance contract is set for Date of the Insured's Death plus 10 years, and the insured dies on 5 March 2010, triggering the event, the contract must be retained until 4 March 2020. The record can only be deleted on or after 5 March 2020. If a legal hold is put on the contract on 15 August 2016 and the hold is still valid on 5 March 2020, you cannot delete the contract. The contract can be deleted only after the legal hold is removed.

Note: While SnapLock does not have native legal hold capability, it is possible to implement this capability by using the native features of SnapLock.

Related tasks

Implementing event-based retention and legal hold on page 149

Implementation of event-based retention and the legal hold feature using SnapLock

SnapLock does not have native event-based retention and legal hold capability; however, it is possible to implement these capabilities by using the native features of SnapLock.

You can use the SnapLock Enterprise volumes and assign a retention period of `infinite`, so that all files that are committed to the WORM state on the volume have a retention period of infinity and can never be deleted. Based on the workflow resulting from event-based retention, legal hold requirements, and actual events, you can delete files that require deletion using the privileged delete feature.

For example: If the retention period of the contract is set for Date of the Insured's Death plus 10 years and the insured dies on 5 March 2010, the event is captured and examined by the application. The application infers that the contract needs to be retained for 10 years and flags the contract to be deleted, using privileged delete, on 5 March 2020. If nothing occurs until 5 March 2020, the application deletes the contract on 5 March 2020. However if a legal hold is put on the contract on 15 August 2016, the application captures this information and infers that the contract should not be deleted on 4 March 2020 but rather it should be deleted when the legal hold is removed. The legal hold is removed on 11 January 2021, and the application simply deletes the contract using the privileged delete feature.

Note: The application references time from the SnapLock ComplianceClock, not the system clock.

Related concepts

What the privileged delete feature is on page 43
What legal hold is on page 145
What event-based retention is on page 143

Implementing event-based retention and legal hold

You can capture an event (insured dies, a legal hold is placed, or a legal hold is removed), examine it, and take the necessary action on that event. Depending on the event, you can flag the record as DELETE or DONT_DELETE. You can delete the files that are marked as DELETE by using the SnapLock privileged delete feature.

Before you begin

- Privileged delete is enabled.
- The files are in WORM state.
- Files are stored in SnapLock Enterprise volume.
- The default retention period of the SnapLock enterprise volume is set to infinite.
- An internal database that is used by the application for implementing the event-based retention or legal hold functionality is in place.

Steps

1. Examine the event and get the record for which the event was triggerred.
2. Choose the event:

If the event is...	Then...
An event-based retention event occurs (for example, Insured dies)	**a.** Determine the retention period after which the file needs to be deleted. For example, if the retention period of a contract is set for Date of Insured's death plus 10 years and the insured dies on 5 March 2010, the application infers that the contract needs to be retained for 10 years. **b.** Calculate the deletion date based on the following formula: Deletion date = ComplianceClock date + retention period. **c.** Add a row to the database corresponding to this file and deletion date (record), and delete the file on the deletion date.
Legal hold	**a.** Find the record in the database. **b.** Edit the record to provide the path and deletion date and flag the record as DONT_DELETE.
Legal hold withdrawn	**a.** Find the record in the database. **b.** Edit the record to provide the path and deletion date and flag the record as DELETE.

3. Scan the database for all files that are marked as DELETE.
4. Delete the files using the privileged delete feature.

Deleting a record using the privileged delete feature

Based on the event triggered, you can delete a record that is marked as DELETE by using the SnapLock privileged delete feature.

Before you begin

- Files are stored in a SnapLock enterprise volume.
- Privileged delete feature is enabled on that volume.
- The default retention period of the volume is set to infinite.

Steps

1. Scan the database for all files that are marked as DELETE.
2. Delete the files using the privileged delete feature.

Related tasks

Deleting a WORM file using privileged delete on page 47

Examples for setting a file to WORM state using an application

You can set up an application to set a file to WORM state by using Data ONTAP APIs.

Example for setting the retention period

Using C#

```
using System;
using System.Text;
using NetApp.Manage;

void set_to_worm_with_retention_period(string serverip,
string serverusername, string serverpasswd, string path, long seconds)
{
    NaElement xi;
    NaElement xo;
    NaServer s;
    long complianceclocktime;
    long retentionperiod;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        s.SetAdminUser(serverusername, serverpasswd);
        //get the ComplianceClock time
        xi = new NaElement("snaplock-get-compliance-clock");
        //Invoke ONTAP API
        xo = s.InvokeElem(xi);
complianceclocktime = Convert.ToInt64(xo.GetChildContent("snaplock-
compliance-clock"));//long is of .NET framework type System.Int64
                                //check if retention period falls
within the normal range or
                                //requires the use of the extended
range
        if (seconds <= 0x7FFFFFFF - complianceclocktime)
        {
            //use of normal range
            retentionperiod = seconds + complianceclocktime;
        }
        else
        {
            //use of extended range
                                                retentionperiod =
seconds + complianceclocktime + 1 -   0x7FFFFFFF;
        }
        // set file to WORM state with this retention period
        xi = new NaElement("file-set-snaplock-retention-time");
```

```
        xi.AddNewChild("path", path);
        xi.AddNewChild("retention-time", retentionperiod.ToString());
        //Invoke ONTAP API
        xo = s.InvokeElem(xi);
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
    }
}
```

Example for setting the retention date

Using C#

```
void set_to_worm_with_retention_date(string serverip,
string serverusername, string serverpasswd, string path, DateTime
date)
{
    NaElement xi;
    NaElement xo;
    NaServer s;
    long retentionperiod;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        s.SetAdminUser(serverusername, serverpasswd);
        //convert date to UTC
                                retentionperiod =
(date.ToUniversalTime().Ticks - new
                                DateTime(1970,1,1,0,0,0,0).Ticks)/
10000000;
                                // if the retention period is not in
the normal date range,
                                //then use of the extended range will
be necessary
        if (retentionperiod > 0x7FFFFFFF)
            //use of extended date range
            retentionperiod = retentionperiod + 1-
0x7FFFFFFF;
        // set file to WORM state with this retention period
        xi = new NaElement("file-set-snaplock-retention-time");
        xi.AddNewChild("path", path);
        xi.AddNewChild("retention-time", retentionperiod.ToString());
        //Invoke ONTAP API
        xo = s.InvokeElem(xi);
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
```

```
    }
}
```

Examples for setting the SnapLock volume defaults

You can set up an application to set the SnapLock default period by using Data ONTAP APIs.

Using C#

```
using System;
using System.Text;
using NetApp.Manage;

void set_snaplock_default_period(string serverip,
string serverusername, string serverpasswd, string volume, long
numberofdays)
{
    NaElement xi;
    NaElement xo;
    NaServer s;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        s.SetAdminUser(serverusername, serverpasswd);
        //set option snaplock_default_period to numberofdays
        xi = new NaElement("volume-set-option");
        xi.AddNewChild("option-name", "snaplock_default_period");
        xi.AddNewChild("option-value", numberofdays.ToString()+"d");
        xi.AddNewChild("volume", volume);
        //Invoke ONTAP API
        xo = s.InvokeElem(xi);
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
    }
}
```

Examples for setting the autocommit feature and time intervals

You can set up an application to set the SnapLock autocommit feature and time intervals by using Data ONTAP APIs.

Using C#

```
using System;
using System.Text;
using NetApp.Manage;

void set_snaplock_autocommit_period(string serverip,
string serverusername, string serverpasswd, int autocommitperiodhrs)
{
    NaElement xi;
    NaElement xo;
    NaServer s;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        s.SetAdminUser(serverusername, serverpasswd);
        //set global option snaplock.autocommit_period to
        //
autocommitperiodhrs
                                                                    //
autocommitperiodhrs
        xi = new NaElement("options-set");
        xi.AddNewChild("name", "snaplock.autocommit_period");
        xi.AddNewChild("value", autocommitperiodhrs.ToString()+"h");
        //Invoke ONTAP API
        xo = s.InvokeElem(xi);
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
    }
}
```

Examples for creating a compliance administrator

You can create a compliance administrator by using Data ONTAP APIs.

Using C#

```
using System;
using System.Text;
using System.Collections.Generic;
using NetApp.Manage;

void create_compliance_user(string serverip, string serverusername,
string serverpasswd, string username, string passwd)
{
    NaElement xi;
    NaElement xo;
    NaElement user;
    NaElement userinfo;
    NaElement group;
    NaElement groupinfo;
    NaServer  s;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        s.SetAdminUser(serverusername, serverpasswd);
        //The ComplianceUser is part of group
'Compliance
        //Administrators'
        //Create useradmin-user-add ONTAPI API
        xi = new NaElement("useradmin-user-add");
        //Create useradmin-user structure
        user = new NaElement("useradmin-user");
        //Create useradmin-user-info structure
        userinfo = new NaElement("useradmin-user-info");
        //Add username
        userinfo.AddNewChild("name", username);
        //Create useradmin-groups structure
        group = new NaElement("useradmin-groups");
        //Create useradmin-group-info structure
        groupinfo = new NaElement("useradmin-group-info");
        //Add groupname "Compliance Administrators"
        groupinfo.AddNewChild("name", "Compliance Administrators");
        group.AddChildElement(groupinfo);
        userinfo.AddChildElement(group);
        user.AddChildElement(userinfo);
        //Add useradmin-user and password
        xi.AddChildElement(user);

xi.AddNewChild("password",passwd);
```

```
        //Invoke useradmin-user-list ONTAP API
        xo = s.InvokeElem(xi);
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
    }
}
```

Examples for setting a SnapLock log volume

You can set up an application to set a SnapLock log volume by using Data ONTAP APIs.

Using C#

```
void set_snaplock_log_volume(string serverip, string serverusername,
string serverpasswd, string snaplockcompliancevolume)
{
    NaElement xi;
    NaElement xo;
    NaServer  s;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        s.SetAdminUser(serverusername, serverpasswd);
        //Check to see if log volume has already been set
        //If not set, set the log volume, otherwise just return
        //Create snaplock-get-log-volume ONTAPI API
        xi = new NaElement("snaplock-get-log-volume");
        //Invoke snaplock-get-log-volume ONTAP API
        xo = s.InvokeElem(xi);
        if (string.Compare(xo.GetChildContent("log-volume"),"Not
configured")!=0)
            //Log volume already setup
            return;
        else
        {
            //Setup log volume
            xi = new NaElement("snaplock-set-log-volume");
            xi.AddNewChild("log-volume", snaplockcompliancevolume);
            //Invoke snaplock-set-log-volume ONTAP API
            xo = s.InvokeElem(xi);
        }
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
    }
}
```

Examples for enabling the privileged delete feature

You can enable the SnapLock privileged delete feature by using Data ONTAP APIs.

Using C#

```
void set_privdel_on(string serverip, string serverusername, string
serverpasswd, string snaplockenterprisevolume)
{
    NaElement xi;
    NaElement xo;
    NaServer s;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        s.SetAdminUser(serverusername, serverpasswd);
        //Set the privileged delete option ON
        xi = new NaElement("snaplock-set-options");
        xi.AddNewChild("option", "privdel");
        xi.AddNewChild("option-value", "on");
        xi.AddNewChild("volume", snaplockenterprisevolume);
        //Invoke snaplock-set-options ONTAP API
        xo = s.InvokeElem(xi);
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
    }
}
```

Examples for performing a privileged delete

You can perform a privileged delete by using Data ONTAP APIs.

Using C#

```
void privileged_delete(string serverip, string serverusername, string
serverpasswd, string path)
{
    NaElement xi;
    NaElement xo;
    NaServer s;

    try
    {
        //Initialize connection to server, and
        //request version 1.3 of the API set
        s = new NaServer(serverip, 1, 3);
        s.Style = NaServer.AUTH_STYLE.LOGIN_PASSWORD;
        //transport needs to be HTTPS
        s.TransportType =
NaServer.TRANSPORT_TYPE.HTTPS;
        s.SetAdminUser(serverusername, serverpasswd);
        //delete the file
        xi = new NaElement("snaplock-privileged-delete-file");
        xi.AddNewChild("do-it", "true");
        xi.AddNewChild("path", path);
        //Invoke snaplock-set-options ONTAP API
        xo = s.InvokeElem(xi);
    }
    catch (Exception e)
    {
        Console.Error.WriteLine(e.Message);
    }
}
```

Frequently asked questions

Frequently asked questions include general questions and questions related to SnapLock volumes, tampering scenarios, and the use of dump, NDMP, and WAFL.

Next topics

General FAQs related to SnapLock

Following are some general questions about SnapLock.

Is it possible to change any attributes on a WORM file before or after expiry date?

Before the retention date is reached:

- In Data ONTAP versions earlier than 7.0, no changes are possible.
- In Data ONTAP 7.0 or later, after a file on a SnapLock volume is committed to WORM status, there is only one attribute that can be modified—the retention date of the file can be extended. The retention period can never be shortened by pulling the retention date closer.

  Note: This does not compromise the WORM protection on the file in any way, but enables the retention period to be extended before the file expires. This is an important feature in cases such as legal holds on certain records.

After the retention date is reached (that is, the file has expired):

- In Data ONTAP versions earlier than 7.0, no changes are possible.
- In Data ONTAP 7.0 or later, after the file has expired (reached its retention date), two changes are possible:
  - The retention date of the file can be extended (pushed further out into the future). This will reenable WORM protection on the file as if it were being committed for the first time.
  - The write permissions on the file can be reenabled, and you can delete the file. However, you cannot modify the file contents.

Does SnapLock require any kind of proprietary API?

No. SnapLock is based purely on the open file protocol interfaces and does not require the use of any kind of proprietary API. You can perform SnapLock-specific operations, such as setting file retention periods and committing files to the WORM state, through regular file system operations available on

all clients. Similarly, applications can use the regular programmatic library interfaces they would use for file operations on any other kind of storage system. Optionally, the Data ONTAP API has a SnapLock interface that an application can use, if desired.

What types of SnapLock licenses are available in Data ONTAP?

The two types of licenses for SnapLock are:

- "snaplock" for SnapLock Compliance
- "snaplock_enterprise" for SnapLock Enterprise

These licenses do not determine the type of existing volumes. That is, volumes remember their SnapLock type regardless of the installed license.

FAQs related to SnapLock volumes

Following are some questions about SnapLock volumes.

Is it possible to convert a SnapLock Enterprise volume to a SnapLock Compliance volume?

You cannot convert a SnapLock Enterprise volume to a SnapLock Compliance volume. However, you can copy a SnapLock volume to another volume by using the `vol copy` command. You can copy between the SnapLock volumes or regular volumes. If the source volume is a SnapLock Enterprise volume and the destination volume is a SnapLock Compliance volume, the destination volume must be an equivalently sized or larger Compliance volume. However, if the destination volume is a SnapLock Compliance volume, you must ensure that the volume is empty, that is, it does not contain any unexpired files. The volumes, other than of the type SnapLock volume (Enterprise or Compliance), should have the same attributes of minimum, maximum, and default retention periods specified. The `vol copy` command retains the retention period attribute of the files and also preserves the WORM attribute of the files in the SnapLock Enterprise volume.

Use the `vol copy` command to copy between SnapLock volumes or regular volumes. The following table shows you all the possible combinations that are allowed for performing the `vol copy` command:

If the source is...	**The destination can be...**
SnapLock Enterprise volume	SnapLock Compliance volume
	SnapLock Enterprise volume
	Regular volume

If the source is...	The destination can be...
SnapLock Compliance volume	SnapLock Enterprise volume
	SnapLock Compliance volume
	Regular volume
Regular volume	SnapLock Enterprise volume
	SnapLock Compliance volume
	Regular volume

Is it possible to convert an existing volume to a SnapLock volume?

No. You cannot convert an existing non-SnapLock volume to a SnapLock volume or vice versa primarily because this opens up many potential security holes regarding the disposition of the volume's existing content. It would jeopardize compliance as well as complicate the user interfaces. The only way to place existing data on a SnapLock volume is to copy the files to a SnapLock volume and transition each individual file from read-write state to read-only state. The setting of the retention period is optional, because a file uses the volume's default retention period if no retention period is specified. Replicating a non-SnapLock volume to a SnapLock volume does not engage the SnapLock locking mechanism. Only the transition from read-write state to read-only state locks the file.

What is the way to identify SnapLock volumes in the system?

You can use the `vol status` command to identify any SnapLock volumes. This command identifies any SnapLock volumes with an entry in the `options` field for the volume. The `vol status` command displays `snaplock_compliance` for SnapLock Compliance volume and `snaplock_enterprise` for SnapLock Enterprise volumes. Following is sample output from a query of the `vol status` command:

```
sys1>aggr status
           Aggr  State     Status         Options
    wormaggr_sle online    raid_dp,aggr   snaplock_enterprise
wormvol_trad_slc online    raid4,trad     snaplock_compliance
wormvol_trad_sle online    raid4,trad     snaplock_enterprise
    wormaggr_slc online    raid_dp,aggr   snaplock_compliance
           aggr0 online    raid4,aggr     root
```

Is it possible to use Snapshot copies on a SnapLock volume?

There are no restrictions on creation, deletion, or number of Snapshot copies on SnapLock volumes beyond the normal system limits because there is no risk of data loss with these operations. This means that SnapMirror, dump, and all other Snapshot copy-based technologies will not be fundamentally impaired when dealing with SnapLock volumes. You can also use the `snap restore` command to restore a previous Snapshot copy on SnapLock Enterprise volumes even though doing so might involve data loss because the operating model of SnapLock Enterprise is that the storage administrator is trusted. However, you cannot use the `snap restore` command to restore a previous

Snapshot copy on SnapLock Compliance volumes. The whole premise of restoring to a previous Snapshot copy is that you are throwing away all modifications that occurred subsequent to the point in time at which the Snapshot copy was taken in order to revert to a previous consistent state of the file system. This is diametrically opposed to the permanence guarantees provided by the untrusted administrator model of SnapLock Compliance and cannot be allowed. To allow such an operation would violate regulatory compliance requirements.

Is it possible to destroy SnapLock volumes to get the disks back?

It depends on the type of SnapLock volume and the Data ONTAP release. Regulatory compliance requires preventing the destruction of any records that have not aged past their assigned retention period. This requirement puts heavy constraints on the allowable destructive administrative operations that can be performed on SnapLock Compliance volumes. The trusted administrator model of SnapLock Enterprise makes life much easier for SnapLock Enterprise volumes.

Following are the considerations for destroying SnapLock volumes in Data ONTAP 7.0 and later releases:

- SnapLock Compliance volumes can be destroyed only if all the WORM files on the volume have passed their specified retention date. Because the default retention date of files on SnapLock Compliance volumes can be very long, extreme care should be exercised by applications while assigning retention dates to WORM files.
- SnapLock Enterprise volumes can always be destroyed because the administrator is trusted.

How do SnapLock volumes work with virtual volumes?

SnapLock is the fundamental property of an aggregate and not a volume. All volumes created on a SnapLock aggregate automatically and permanently inherit the SnapLock type of the aggregate. This means that you cannot have SnapLock volumes on non-SnapLock aggregates and vice versa. You also cannot have SnapLock Enterprise and Compliance volumes on the same aggregate. The virtual volume type is inherited from the SnapLock aggregate type.

Is it possible to shrink a SnapLock FlexVol volume?

Yes, as long as you do not attempt to shrink SnapLock FlexVol volumes below the actual used capacity.

Is it possible to grow a SnapLock FlexVol volume?

Yes, it is possible to increase the size of a SnapLock FlexVol volume like a regular volume.

How do symbolic links work on SnapLock volumes?

Symbolic links do not have any permissions of their own. Therefore, the symbolic links cannot be made read-only and committed to the WORM state.

FAQs related to tampering scenarios

Following are some questions related to tampering scenarios.

Is it possible to deactivate SnapLock and start deleting WORM files?

You can remove the SnapLock license. However, files that are already in the SnapLock state retain their immutable WORM protection. The only disadvantage of removing the SnapLock license is that you can no longer put new files into the WORM state, or create new SnapLock volumes.

Is it possible to defeat the retention date protection and delete files by changing the system clock to the future?

No, file deletion is not possible on a Data ONTAP release that does not have a ComplianceClock. ComplianceClock is a secure time base that can be initialized only once. ComplianceClock is immune to tampering with the system clock.

Is it possible to get rid of incriminating records without trace by copying to a new SnapLock volume (minus the incriminating records) and redirecting the archival application to the new volume?

No, each SnapLock volume has a unique UUID, which is typically tracked by the archival application, or as part of the company's archival policies. A new SnapLock volume can never re-create the unique ID of the original volume with incriminating records, thus defeating any spoofing attempt.

Is it possible to take a SnapLock Compliance volume offline and move it to another storage controller with a different time set and then reinitialize ComplianceClock?

You can move the volume, but you cannot reinitialize ComplianceClock. If ComplianceClock is already initialized on the system, but does not match ComplianceClock stored in the volume that you have moved, the least of the two values is used as ComplianceClock time.

What needs to be done if the storage system is powered off for a while, and now ComplianceClock is behind the system clock?

You do not need to fix this issue because ComplianceClock catches up to the system clock at the rate of one week per year. This is to prevent any tampering with the retention date.

Is it possible to rename SnapLock volumes?

SnapLock Enterprise volumes can be renamed. However, for compliance reasons, you cannot rename SnapLock Compliance volumes.

FAQs related to dump, NDMP, and WAFL

Following are some questions related to dump, NDMP, and WAFL in SnapLock.

Is it possible to use dump, restore, or NDMP to back up SnapLock volumes?

Yes, but there are some restrictions for restores. Dump and NDMP backups work exactly as they work on regular volumes. However, there are two restrictions with restore and NDMP restore:

- True incremental restores are not supported on SnapLock volumes. The `restore` command automatically converts `restore r` to `restore x` if the destination volume is a SnapLock volume.

  This restriction is necessary because of the way the incremental restore process works. For example, moving directories around and renaming them is not allowed on WORM volumes and causes the incremental restore to break.
- Restore and NDMP do not get any free passes around the SnapLock protection and are bound by the same restrictions that apply to a regular client. This means that restore and NDMP cannot overwrite, modify, delete, or change attributes of any existing WORM files on a SnapLock volume.

Do dump, restore, and NDMP preserve the WORM attributes of the files?

Yes. The data streams of these features have been augmented to preserve the WORM attributes of files on a SnapLock volume when data is backed up, restored, or copied. However, for the WORM attributes to be meaningfully enforced, you must perform the restore to a SnapLock volume. If the backup from a SnapLock volume is restored to a non-SnapLock volume, the WORM attributes preserved are ignored, but they will not be enforced by Data ONTAP.

SnapLock error messages

This table lists common error messages in SnapLock, their probable cause, and the appropriate action to be taken, if any.

Error message	Cause	Appropriate action
`snaplock.log.mdata.IOerr`	This event is generated when there is a SnapLock logging metadata I/O error. The SnapLock logging system will not function until this condition is corrected.	SnapLock log file I/O errors can be a result of a full volume or a corrupted log file. Check that there is enough disk space to satisfy logging requirements. If a corrupt file is suspected, archive the log with the `snaplock log archive` command or change the log volume with the `snaplock log volume` command.
`snaplock.log.IO.error`	This event is generated when there is a SnapLock log file I/O error. The SnapLock logging system will not function until this condition is corrected.	SnapLock log file I/O errors can be a result of a full volume or a corrupted log file. Check that there is enough disk space to satisfy logging requirements. If a corrupt file is suspected, archive the log with the `snaplock log archive` command or change the log volume with the `snaplock log volume` command.
`snaplock.user`	This event is generated when the SnapLock logging system fails to log the addition or deletion of a compliance user to or from the compliance group. The actual operation might have succeeded. This event only applies when there is a failure to securely log the request.	No corrective action is necessary. However, a secure log of the request might not exist, and you should check the SnapLock log volume for correct operation.

Index

R

S

V

W