Creating and using Snapshot copies in SnapDrive for UNIX

Snapdrive for Unix

NetApp
February 12, 2024

Table of Contents

Creating and using Snapshot copies in SnapDrive for UNIX

SnapDrive for UNIX enables you to create Snapshot copies of host data.

What Snapshot operations are in SnapDrive for UNIX

SnapDrive for UNIX enables you to use Data ONTAP Snapshot technology to make an image (Snapshot copy) of host data that is stored on a storage system.

This Snapshot copy provides you with a copy of that data, which you can restore later. The data in the Snapshot copy can exist on one storage system or span multiple storage systems and their volumes. These storage systems can be in HA pair or node-local file systems or disk groups or LUNs in a host cluster environment. On a nonclustered UNIX host with SnapDrive for UNIX installed, you can create a Snapshot copy of one or more volume groups on a storage system.

The Snapshot copy can contain file systems, logical volumes, disk groups, LUNs, and NFS directory trees. After you create a Snapshot copy, you can rename, restore, or delete it. You can also connect to a different location on the same host or to a different host. After you connect, you can view and modify the content of the Snapshot copy, or you can disconnect the Snapshot copy. In addition, SnapDrive for UNIX enables you to display information about the Snapshot copy that you have created. On a clustered UNIX host with SnapDrive for UNIX installed, you can perform Snapshot operations on the host cluster nodes that includes disk groups and file systems. The Snapshot operations include create, rename, restore, connect, disconnect, display, and delete.

Considerations when working with Snapshot copies

There are certain considerations that you should keep in mind when using a Snapshot copy, such as the fact that SnapDrive for UNIX works only with Snapshot copies that it creates, or that Snapshot copies are automatically replicated from the storage system to the host, and so on.

When working with Snapshot operations, you should consider the following:

- SnapDrive for UNIX works only with Snapshot copies that it creates.

  It cannot restore Snapshot copies that it did not create.

- When you create a Snapshot copy on a destination storage system, the Snapshot copy is automatically replicated from the source storage system on which it is created to the destination storage system.

  SnapDrive for UNIX enables you to restore the Snapshot copy on the destination storage system as well.

- Connecting to the originating host occurs when you use the `snapdrive snap connect` command to connect to a Snapshot copy at a new location on the same host where it was last connected (or is still connected).
- Snapshot support for storage entities spanning multiple storage system volumes or multiple storage systems is limited on configurations that do not allow a freeze operation in the software stack.
- When you export the volume through the NFS protocol, you must set the Anonymous User ID option to 0 for the SnapDrive for UNIX commands to work.

- When there are two JFS file systems (JFS and JFS2) in a single disk group, SnapDrive for UNIX supports operations only for the JFS2 file system.
- SnapDrive for UNIX enables you to create Snapshot copies of a concurrent volume group, and to clone or restore using the same properties.
- A Single-File SnapRestore (SFSR) operation followed immediately by the creation of a Snapshot copy fails.

  You must retry the operation after some time passes. For more information, see the Clustered Data ONTAP Logical Storage Management Guide.

Snapshot copy operations

You can create Snapshot copies by using the `snapdrive snap create` command.

Crash-consistent Snapshot copies

You might have to create a crash-consistent Snapshot copies of your file system or disk groups. SnapDrive for UNIX creates Snapshot copies that contain the image of all the storage system volumes specified in the entity.

When you create a Snapshot copy of a storage entity, such as a file system or disk group, SnapDrive for UNIX creates a Snapshot copy that contains the image of all the storage system volumes that comprise the entity you specified using a `file_spec` argument. The `file_spec` argument specifies the storage entity, such as the file system, LUN, or NFS directory tree that SnapDrive for UNIX uses to create the Snapshot copy.

SnapDrive for UNIX makes consistent storage components that compose the entity you requested in the Snapshot copy. This means that LUNs or directories being used outside those specified by the `snapdrive snap create` command `file_spec` argument might not have consistent images in the Snapshot copy. SnapDrive for UNIX enables you to restore only the entities specified by the `file_spec` argument that are consistent in the Snapshot copy.

Snapshot copies of entities contained on a single storage system volume are always crash-consistent. SnapDrive for UNIX takes special steps to ensure that Snapshot copies that span multiple storage systems or storage system volumes are also crash-consistent. The method that SnapDrive for UNIX uses to ensure crash consistency depends on the Data ONTAP version used where the storage entities in your Snapshot copy resides.

Crash consistency with Data ONTAP 7.2 and later

SnapDrive for UNIX uses the support for consistency groups provided by Data ONTAP 7.2 and later versions, such that all Snapshot copies that span multiple volumes are crash consistent.

Data ONTAP versions 7.2 and greater provides support for consistency groups and storage system fencing. SnapDrive for UNIX uses these features to ensure that all Snapshot copies that span multiple volumes are crash consistent.

To create a crash consistent Snapshot copy across multiple volumes, SnapDrive for UNIX does the following:

- Fences (freezes) I/O to every volume that contains a storage entity.
- Takes a Snapshot copy of each volume.

The time it takes to fence the volume and create the Snapshot copy is limited, and is controlled by Data ONTAP.

The **`snapcreate-cg-timeout`** parameter in the `snapdrive.conf` file specifies the amount of time, within Data ONTAP limitations, that you wish to allow for storage system fencing. You can specify an interval that is urgent, medium, or relaxed. If the storage system requires more time than allowed to complete the fencing operation, SnapDrive for UNIX creates the Snapshot copy using the consistency methodology for previous Data ONTAP 7.2 versions. You can also specify this methodology by using the `-nofilerfence` option when you create the Snapshot copy.

If you request a Snapshot copy for a storage entity that spans storage systems with both Data ONTAP 7.2 and previous Data ONTAP versions, SnapDrive for UNIX also creates the Snapshot copy using the consistency method for Data ONTAP versions before 7.2.

Consistency group Snapshot copies in SnapDrive for UNIX

Consistency Group Snapshot is a Snapshot copy of a set of volumes that span different Vservers or nodes of a cluster, which is managed as a single entity.

SnapDrive for UNIX captures crash-consistent Snapshot copies across all volumes spanning different Vservers or nodes of a cluster. You can also configure the time period within which the Snapshot copy is to be captured.

SnapDrive for UNIX captures consistency group Snapshot copies by default. You can disable this feature and revert to capturing Snapshot copies in best-effort mode.

> SnapDrive for UNIX 5.2 supports consistency group Snapshot copies for clustered Data ONTAP only in Data ONTAP 8.2 or later versions.

Related information

Creating a consistency group Snapshot copy

Disabling consistency group Snapshots copies

Creating a consistency group Snapshot copy

You can configure SnapDrive for UNIX to create a consistency group Snapshot copy.

Steps

1. Enter the following command on the host:

   `snapdrive snap create -fs /mnt/test -snapname` ***`snapshotname`*** **`-f -noprompt`**.

   `snapshotname` is the name specified for the consistency group Snapshot copy.

Example

The following is an example of the command:

```
snapdrive snap create -fs /mnt/test -snapname snap_123 -f -noprompt
```

The consistency group Snapshot copy is successfully created.

Disabling consistency group Snapshots copies

You can configure SnapDrive for UNIX to disable a consistency group Snapshot copy.

Steps

1. Enter:

   ```
   snapdrive snap create -fs /mnt/fs3 -snapname nfs_snap -nofilerfence
   ```

   The consistency group Snapshot copy is successfully disabled.

Application-consistent Snapshot copies

To make an application-consistent Snapshot copy, you should halt the application before the Snapshot operation.

To ensure that a Snapshot copy is application-consistent, you might need to stop or perform the necessary steps to quiesce the application before taking the Snapshot copy. Note that database hot backup facilities depend on the methods used by the DBMS, and do not always quiesce I/O to database files.

If the application has not completed its transactions and written data to the storage system, the resulting Snapshot copy might not be application-consistent.

> If your application can recover from a crash-consistent Snapshot copy, you do not need to stop it. Consult the documentation for your application. For more information about taking application-consistent Snapshot copies.

You should take a new Snapshot copy whenever you add or remove a host volume, LUN, or NFS directory tree, or resize host volumes or file systems. This ensures that you have a consistent copy of the newly configured disk group that you can use if you need to restore the disk group.

Snapshot copies that span storage systems or volumes

SnapDrive for UNIX enables you to take Snapshot copies that reside in multiple storage system volumes on the same or different storage systems.

SnapDrive for UNIX allows you to take Snapshot copies that span across multiple storage system volumes or multiple storage systems. These volumes can reside on the same storage system or different storage systems. Although the `snapdrive snap create` command creates a Snapshot copy of all the volumes that comprise the entity you request, SnapDrive for UNIX restores only the entities that you specify in the `snapdrive snap create` command.

When you use the `snapdrive snap create` command to make a Snapshot copy that spans multiple volumes, you do not need to name the volumes on the command prompt. SnapDrive for UNIX gets this information from the `file_spec` argument that you specify.

- If the `file_spec` you enter requests a disk group, or a file system or host volume that resides on a disk group, SnapDrive for UNIX automatically creates a Snapshot copy that includes all the storage system volumes for the disk group, volume, or file system you specified.

- If the `file_spec` you enter requests a LUN, SnapDrive for UNIX takes a Snapshot copy of the storage system volume that contains the LUN.
- If the `file_spec` you enter requests a file system that resides directly on a LUN, SnapDrive for UNIX takes a Snapshot copy of the storage system volume that contains the LUN and file system that you specified.
- If the `file_spec` you enter requests an NFS directory, SnapDrive for UNIX creates a Snapshot copy of the volume that contains the NFS directory tree.

In addition to using a `file_spec` argument that is built on entities from multiple storage systems and storage system volumes, you can also use a combination of `file_spec` arguments where each value is based on single storage system or storage system volume. For example, suppose you have a setup where the disk group dg1 spans the storage systems storage system2 and storage system3, dg2 is on storage system2, and dg3 is on storage system3. In this case, any of the following command lines would be correct:

```
snapdrive snap create -dg dg1 -snapname snapdg1

snapdrive snap create -dg dg2 dg3 -snapname snapdg23

snapdrive snap create -dg dg1 dg2 dg3 -snapname snapdg123
```

Something to keep in mind when creating Snapshot copies that span storage systems and volumes is that SnapDrive for UNIX creates the Snapshot copy on each storage systems volume using a short name. It does not include the full path name in the name, even if the volumes are on different storage system. This means that if you later rename the Snapshot copy, you must go to each storage system and volume and rename it there as well.

Creating Snapshot copies of unrelated entities

SnapDrive for UNIX creates Snapshot copies of unrelated entities by maintaining individual crash-consistent Snapshot copies.

Unless you specify otherwise, SnapDrive for UNIX assumes that all entities that you specify on a particular snapdrive snap create command line are related; in other words the validity of updates to one entity can depend on updates to the other entities specified. When storage entities have dependent writes in this way, SnapDrive for UNIX takes steps to create a Snapshot copy that is crash consistent for all storage entities as a group.

The following example shows how SnapDrive for UNIX creates a Snapshot copy of storage entities that may have dependent writes. In the following example, the snapdrive snap create command specifies a file system on a LUN and also a disk group. The disk group consists of LUNs residing on a single storage system (see Creating a Snapshot copy). The file system on a LUN resides on a different storage system and storage system volume. As a group, the file system and the disk group span multiple storage system volumes; individually they do not.

The following command specifies a Snapshot copy that contains both the file system /mnt/fs1 and the disk group dg1: `snapdrive snap create -fs /mnt/fs1 -dg dg1 -snapname fs1_dg1`

Because these storage entities can have dependent writes, SnapDrive for UNIX attempts to create a crash-consistent Snapshot copy, and treats the file system /mnt/fs1 and the disk group dg1 as a group. This means SnapDrive for UNIX is required to freeze I/O operations to the storage system volumes before creating the Snapshot copy.

Creating crash-consistent Snapshot copies for multiple storage entities across volumes takes extra time, and is not always possible if SnapDrive for UNIX cannot freeze I/O operations. Because this is so, SnapDrive for UNIX allows you to create Snapshot copies of unrelated storage entities. Unrelated storage entities are entities that you can specify that have no dependent writes when the Snapshot copy is taken. Because the entities have no dependent writes, SnapDrive for UNIX does not take steps to make the entities consistent as a group. Instead, SnapDrive for UNIX creates a Snapshot copy in which each of the individual storage entities is crash-consistent.

The following command specifies a Snapshot copy of the file system on a LUN and the disk group described previously. Because the -unrelated option is specified, SnapDrive for UNIX creates a Snapshot copy in which the file system /mnt/fs1 and the disk group dg1 are crash-consistent as individual storage entities, but are not treated as a group. The following command does not require SnapDrive for UNIX to freeze I/O operations on the storage system volumes: `snapdrive snap create -fs /mnt/fs1 -dg dg1 -unrelated -snapname fs1_dg1`

Related information

Creating a Snapshot copy

Guidelines for Snapshot copy creation

Consider the following while creating Snapshot copies using SnapDrive for UNIX: you can keep maximum 255 Snapshot copies per volume, SnapDrive for UNIX supports only the Snapshot copies that it creates, you cannot create Snapshot copies of root disk groups, and boot device or swap device, and SnapDrive for UNIX requires a freeze operation to maintain crash-consistency.

Follow these guidelines when you enter commands that create Snapshot copies:

- You can keep a maximum of 255 Snapshot copies per storage system volume. This limit is set by the storage system. The total number can vary depending on whether other tools use these Snapshot copies.

  When the number of Snapshot copies has reached the maximum limit, the Snapshot create operation fails. You must delete some of the old Snapshot copies before you can use SnapDrive for UNIX to take anymore.

- SnapDrive for UNIX does not support Snapshot copies that it does not create. For example, it does not support Snapshot copies that are created from the storage system console, because such a practice can lead to inconsistencies within the file system.
- You cannot use SnapDrive for UNIX to create Snapshot copies of the following:
  - Root disk groups

    The Snapshot create operation fails when you try to take a Snapshot copy of a root disk group for an LVM.

  - Boot device or swap device

    SnapDrive for UNIX does not take a Snapshot copy of a system boot device or a system swap device.

- When a Snapshot copy spans multiple storage systems or storage system volumes, SnapDrive for UNIX requires a freeze operation to guarantee crash-consistency. For information about creating Snapshot copies on configurations for which a freeze operation is not provided.

Information required for using the snapdrive snap create command

When you create a Snapshot copy, you should determine the storage entity that you want to capture and specify a name of the Snapshot copy.

The following table provides the information you need when you use the `snapdrive snap create` command:

Requirement/Options	Argument
Determine the type of storage entity you want to capture in the Snapshot copy. You can specify NFS entities, LUNs, file systems created directly on LUNs, and LVM entities on a single command line. Supply that entity's name with the appropriate argument. This is the value for the `file_spec` argument. • If you specify a disk group that has a host volume or file specification, the argument translates into a set of disk groups on the storage system. SnapDrive for UNIX creates the entire disk group containing the entity, even if the entity is a host volume or file system. • If you specify a file specification that is an NFS mount point, the argument translates to the directory tree on the storage system volume. • If you specify a LUN, or a LUN that has a file system, the argument translates to the LUN, or to the LUN that has the file system. ⓘ You cannot specify special characters with the storage entities such as `-vg`, `-dg`, `-fs`, `-lvol`, and `-hostvol`. However, slash (/) is allowed for storage entities such as `-fs`, `-lvol`, and `-hostvol`.	
LUN (`-lun file_spec`)	Name of the LUN You must include the name of the storage system, volume, and LUN.
Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	Name of the disk or volume group
File system (`-fs file_spec`)	filesystem_name
Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	Name of the host or logical volume ⓘ You must supply both the requested volume and the disk group containing it; for example, `-hostvol dg3/acct_volume`.
Snapshot copy name (`-snapname snap_name`)	Snapshot copy_name

Requirement/Options	Argument
Specify the name for the Snapshot copy. This can be either the long version of the name that includes the storage system and volume with the Snapshot copy name or the short version that is the Snapshot copy name. You must ensure that the value specified for `snapname` does not exceed 200 characters.	
`-unrelated`	~
Optional: To create a Snapshot copy of storage entities that have no dependent writes when the Snapshot copy is created, SnapDrive for UNIX creates a crash-consistent Snapshot copy of the individual storage entities, but does not try to make the entities consistent with each other.	
`-force`	~
`-noprompt`	~
Optional: Decide if you want to overwrite an existing Snapshot copy. Without this option, this operation halts if you supply the name of an existing Snapshot copy. When you supply this option and specify the name of an existing Snapshot copy, the command prompts you to confirm that you want to overwrite the Snapshot copy. To prevent SnapDrive for UNIX from displaying the prompt, include the `-noprompt` option also. (You must always include the `-force` option if you want to use the `-noprompt` option.)	
`-fstype`	*type*
`-vmtype`	*type*
Optional: Specify the type of file system and volume manager to be used for SnapDrive for UNIX operations.	

Creating a Snapshot copy

To create a Snapshot copy, run the `snapdrive snap create` command.

Before you execute this syntax, you must understand the options, keywords, and arguments mentioned in this command.

Steps

1. Enter the following command syntax to create a Snapshot copy:

```
snapdrive snap create {-lun | -dg | -fs | -hostvol } file_spec[file_spec …] [
{-lun |-dg | -fs | -hostvol } file_spec [file_spec…]] -snapname snap_name [
-force [-noprompt]][-unrelated] [-nofilerfence] [-fstype type][-vmtype type]
```

The *file_spec* arguments represent a set of storage entities on one or more storage systems. The Snapshot create operation takes a Snapshot copy of the storage system volume containing those entities and gives it the name specified in the `snap_name` argument.

Related information

Creating Snapshot copies of unrelated entities

Snapshot copies information display

You can display Snapshot copies for a storage system, a storage system volume, LUNs, and other storage entities. Use the command `snapdrive snap show` (or `list`) to display a Snapshot copy.

Command to use to display Snapshot copy information

Use the `snapdrive snap show` (or `list`) command to display information about the Snapshot copies.

You can use the `snapdrive snap show` (or list) command to display information about each Snapshot copy that is created by SnapDrive for UNIX. You can use this command to display information about the following:

- Storage systems
- Volumes on storage systems
- Storage entities such as NFS files and directory trees, volume groups, disk groups, file systems, logical volumes, and host volumes
- Snapshot copies

> The `show` and `list` forms of this command are synonymous. For SnapDrive 2.0 for UNIX and later, you must use the long form of the Snapshot copy name when you display information about Snapshot copies.

Guidelines for displaying Snapshot copies

You can use wildcards to display the Snapshot copies. You can display all the Snapshot copies present in a specific object.

Keep the following points in mind, before working with Snapshot copies:

- You can use the wildcard (*) character in Snapshot copy names. The Snapshot show operation lets you use the wildcard character to show all Snapshot copy names that match a certain pattern or all Snapshot copy names on a particular volume. The following rules apply while using wildcard in Snapshot copy names:
  - You can use a wildcard at the end of the name only. You cannot use the wildcard at the beginning or the middle of a Snapshot copy name.
  - You cannot use the wildcard in the storage system or storage system volume fields of a Snapshot copy name.
- You can also use this command to list all of the Snapshot copies on specific objects, including storage systems and their volumes, disk groups, host volume groups, file systems, host volumes, and logical volumes.
- If you enter a `snapdrive snap show` command and SnapDrive for UNIX does not locate any Snapshot copies, it displays the message “no matching Snapshot copies.” If you specify arguments on the command

line, and some portions of them do not exist, SnapDrive for UNIX returns a partial listing of those for which Snapshot copies are found. It also lists the arguments that were invalid.

- If the `snapdrive snap create` command is abruptly aborted, an incomplete `.stoc.xml` file is stored in the volume on the storage system. Due to this, all scheduled Snapshot copies made by the storage system will have a copy of the incomplete `.stoc.xml` file. For the `snapdrive snap list` command to work successfully, complete the following steps:

Steps

1. Delete the incomplete `.stoc.xml` file in the volume.
2. Delete the scheduled Snapshot copies made by the storage system containing the incomplete `.stoc.xml` file.

Information required for using the snapdrive snap show or list commands

You can use the command `snapdrive snap show` or `list` to display information about storage systems, storage system volumes, disks, or volume groups, file system, Snapshot copies, and so on.

The following table gives the information you need to supply when you use the `snapdrive snap show | list` command.

> You can use the same arguments regardless of whether you enter `snapdrive snap show` or `snapdrive snap list` as the command. These commands are synonyms.

Requirement/Option	Argument
Based on the command you enter, you can display information about any of the following: • Storage systems • Storage system volumes • Disk or volume groups • File systems • Host or logical volumes • Snapshot copies The value you enter for the `file_spec` argument must identify the storage entity about which you want to display information. The command assumes the entities are on the current host.	
Storage system (`-filer`)	*`filername`*
A volume on the storage system (`-filervol`)	*`filervol`*
Disk group (`-dg` *`file_spec`*) or volume group (`-vg` *`file_spec`*)	*`name of the disk or volume group`*
File system (`-fs` *`file_spec`*)	*`filesystem_name`*

Requirement/Option	Argument
Host volume (`-hostvol` *`file_spec`*) or logical volume (`-lvol` *`file_spec`*)	*`name of the host or logical volume`*
Snapshot copy name (`-snapname` *`long_snap_name`*)	*`long_snap_name`*
Additional Snapshot copy names	*`snap_name`* (long or short version)
If you want to display information about a Snapshot copy, specify the name for the Snapshot copy. For the first Snapshot copy, *`long_snap_name`*, enter the long version of the name, which includes the storage system name, volume, and Snapshot copy name. You can use the short version of the Snapshot copy name if it is on the same storage system and volume.	
`-verbose`	~
To display additional information, include the `-verbose` option.	

Displaying Snapshot copies residing on a storage system

You can use the SnapDrive command to display Snapshot copies on a storage system.

To display information about Snapshot copies residing on a storage system, use the following syntax:

```
snapdrive snap show -filer filername [filername…] [-verbose]
```

Displaying Snapshot copies of a storage system volume

You can use the SnapDrive command to display Snapshot copies on a storage system volume.

To display information about Snapshot copies of a storage system volume, use the following syntax:

```
snapdrive snap show -filervol filervol [filervol…] [-verbose]
```

Displaying a Snapshot copy

The command `snapdrive snap show` or `list` displays the name of the storage system where the Snapshot copy is created, the host name, date and time, and so on.

Steps

1. Enter the following command syntax to display a Snapshot copy:

   `snapdrive snap show [-snapname] long_snap_name [snap_name…] [-verbose]`

This operation displays, at a minimum, the following information about the Snapshot copy:

- The name of the storage system where the Snapshot copy is created
- The name of the host that took the Snapshot copy
- The path to the LUNs on the storage system
- The date and time the Snapshot copy was created
- The name of the Snapshot copy
- The names of the disk groups included in the Snapshot copy

Example: The following are examples of `snapdrive snap show` commands:

```
# snapdrive snap show -snapname toaster:/vol/vol2:snapA snapX snapY

# snapdrive snap show -verbose toaster:/vol/vol2:snapA /vol/vol3:snapB
snapC

# snapdrive snap show toaster:/vol/vol2:snapA

# snapdrive snap list -dg dg1 dg2
```

Example: This example uses a wildcard to display information about the available Snapshot copies on a particular volume:

```
# snapdrive snap show toaster:/vol/vol1:*
snap name host date
snapped
-----------------------------------------------------------------
-------------
toaster:/vol/vol1:sss1 DBserver Mar 12
16:19 dg1
toaster:/vol/vol1:testdg DBserver Mar 12
15:35 dg1
toaster:/vol/vol1:t1 DBserver Mar 10
18:06 dg1
toaster:/vol/vol1:hp_1 HPserver Mar 8
19:01 vg01
toaster:/vol/vol1:r3 DBserver Mar 8
13:39 rdg1
toaster:/vol/vol1:r1 DBserver Mar 8
13:20 rdg1
toaster:/vol/vol1:snap2 DBserver Mar 8
11:51 rdg1toaster:/vol/vol1:snap_p1 DBserver
Mar 8 10:18 rdg1
```

Example: In this example, the `-verbose` option is used in the command line in an AIX host:

```
# snapdrive snap list betty:/vol/vol1:testsnap -v
snap name host date
snapped
-----------------------------------------------------------------
-------------
betty:/vol/vol1:testsnap aix198-75 Jul 31
10:43 test1_SdDg
host OS: AIX 3 5
snapshot name: testsnap
Volume Manager: aixlvm 5.3
disk group: test1_SdDg
host volume: test1_SdHv
file system: test1_SdHv file system type: jfs2
mountpoint:
/mnt/test1
lun path dev paths
-------------------------------------------------------
betty:/vol/vol1/aix198-75_luntest1_SdLun /dev/hdisk32
```

Example: This example includes messages about Snapshot copies on an AIX host that are not created with SnapDrive for UNIX:

```
# snapdrive snap show -filer toaster
snap name host date
snapped
-----------------------------------------------------------------
-------------
toaster:/vol/vol1:hourly.0 non-snapdrive snapshot
toaster:/vol/vol1:hourly.0 non-snapdrive snapshot
toaster:/vol/vol1:snap1 DBserver1 Oct 01 13:42 dg1
dg2
toaster:/vol/vol1:snap2 DBserver2 Oct 10 13:40
DBdg/hvol1
toaster:/vol/vol1:snap3 DBserver3 Oct 31 13:45
DBdg
```

This example displays a Snapshot copy of an LVM-based file system on an AIX host using the `snapdrive snap show` command and the `- verbose` option:

```
# snapdrive snap show -v -fs /mnt/check_submit/csdg2/hv3_0
snapname host date snapped
-----------------------------------------------------------------
-------------
toaster:/vol/vol1:mysnapshot sales-aix Aug 24 10:55 csdg2
host OS: AIX 1 5
snapshot name: mysnapshot
Volume Manager: aixlvm 5.1
disk group: csdg2
host volume: csdg2_log
host volume: csdg2_hv3_0
file system: csdg2_hv3_0 file system type: jfs2 mountpoint:
/mnt/check_submit/csdg2/hv3_0
lun path dev paths
-------------------------------------------------------
spinel:/vol/vol1/check_submit_aix-4 /dev/hdisk4
```

Example: The following examples use wildcard:

```
# snapdrive snap show toaster:/vol/volX:*
# snapdrive snap show -v toaster:/vol/volX:DB1*
filer1:/vol/volY:DB2*
# snapdrive snap show toaster:/vol/vol2:mysnap* /vol/vol2:yoursnap*
hersnap*
# snapdrive snap show toaster:/vol/volX:*
```

Example: In this example use of a wildcard is invalid because the wildcard is in the middle of the name instead of being placed at the end:

```
# snap show toaster:/vol/vol1:my*snap
```

Other ways to get Snapshot copy names

Use the `snapdrive snap list` command to display the Snapshot copy name.

Another way to get a Snapshot copy name is to log in to the storage system and use the `snapdrive snap list` command there. This command displays the names of the Snapshot copies.

> The `snapdrive snap show` command is equivalent to the storage system `snapdrive snap list` command.

Snapshot copy rename

You can change the name of a Snapshot copy by using the `snapshot snap rename` command. A Snapshot copy that is across multiple storage systems or storage system volumes can also be renamed.

Command to use to rename a Snapshot copy

Use the `snapshot snap rename` command to rename a Snapshot copy.

Renaming a Snapshot copy that spans storage systems or volumes

For Snapshot copies that cross multiple storage systems or storage system volumes, rename all the related Snapshot copies.

You can also use this command to rename a Snapshot copy that is across multiple storage systems or multiple storage system volumes.

If you rename one of these Snapshot copies, you must also rename all the related Snapshot copies using the same name. This is because SnapDrive for UNIX uses a short name when it creates the Snapshot copy, even though it spans multiple storage systems or volumes. The rename command changes the name of the current Snapshot copy but it does not change the name of the related Snapshot copies in the other locations.

Guidelines for renaming Snapshot copies

While renaming Snapshot copies, ensure that the two Snapshot copies do not have the same name.

Follow these guidelines when you use the `snapdrive snap rename` command:

- An error message occurs if you try to rename a Snapshot copy to a different storage system volume.
- An error message occurs if the new name for the Snapshot copy already exists. You can use the `-force` option to force SnapDrive for UNIX to change the name without generating an error message.

Changing a Snapshot copy name

You can change the name of a Snapshot copy by using the `snapdrive snap rename` command. The Snapshot rename operation changes the name of the source Snapshot copy to the name specified by the target argument.

Before you execute this syntax, you must understand the options, keywords, and arguments mentioned in this command.

Steps

1. Enter the following command to change the Snapshot copy name:

   `snapdrive snap rename [-snapname] old_long_snap_name new_snap_name [-force [-noprompt]]`

The following are examples of the `snapdrive snap rename` command. The first command line includes the `-force` option because a Snapshot copy named `new snapshot` copy already exists. In the second example, both Snapshot copy names use the long form of the name, but they both resolve to the same storage system volume.

```
snapdrive snap rename -force filer1:/vol/vol1:oldsnap new snapshot
```

```
snapdrive snap rename filer1:/vol/vol1:FridaySnap
filer1:/vol/vol1:Snap040130
```

Restoring a Snapshot copy

You can restore a Snapshot copy of a single storage entity or multiple storage entities.

Command to use to restore Snapshot copies

Use the command `snapdrive snap restore` to restore a Snapshot copy.

The `snapdrive snap restore` command restores data from the Snapshot copy you specify at the command prompt to the storage system. This operation replaces the contents of the `file_spec` arguments (for example disk groups, NFS files, NFS directory trees, file systems created directly on LUNs) that you specified on the `snapdrive snap restore` command with the contents of the `file_spec` arguments located in the specified Snapshot copy.

You can also restore Snapshot copies for non-existent `file_spec` arguments. This happens when the value you specify no longer exists on the host, but existed when you took the Snapshot copy. For example, it might be a file system that you have now unmounted or a disk group that you have removed.

Normally, you restore Snapshot copies from the host where you took the Snapshot copies (in other words, the originating host).

- In a NFS entity, when the same volume is mounted on two different IPs, snap restore operations can be performed only for one file specification at a time for these volumes.
- If you are a non-root user then, from SnapDrive 4.1 for UNIX, you should have storage write capability on global in addition to GlobalSDSnapshot capability for Snapshot restore to work.

Restoring Snapshot copies on a destination storage system

You can restore Snapshot copies on the storage system from which it is created.

When you create a Snapshot copy on a destination storage system, the Snapshot copy is automatically replicated, from the source system, where it is created to the destination storage system. SnapDrive for UNIX allows you to restore the Snapshot copy on the source storage system. You can also restore the Snapshot copy on the destination storage system.

If you are performing a single file snap restore in a SnapMirror relationship, then the source and destination volume names should not be the same. SnapDrive for UNIX displays the following error message if the source and destination volume name are the same:

```
0001-636 Command error: Snapdrive cannot restore LUNs on SnapMirror
destination filer volumes: <filer-vol-name>
```

In SnapDrive 4.0 for UNIX and later, if Role Based Access Control is enabled, then you can perform snap restore on vFiler unit only when you have the Snapshot Restore capability on the vFiler unit.

Related information

Role-based access control in SnapDrive for UNIX

Restoring multiple storage entities

You can restore a Snapshot copy that contains multiple storage entities.

To restore a Snapshot copy that contains storage entities that reside on multiple destination storage systems, you must meet the following requirements:

- The storage entities you specify at the command prompt must reside on a single storage system, or on a HA pair.
- The name of the volume of the source storage system must match the name of the volume of the destination storage system.
- You must set the `snapmirror-dest-multiple-filervolumes-enabled` argument in the `snapdrive.conf` file to on.

You can use one command to restore storage entities that reside on a single storage system or on a HA pair.

Considerations for restoring a Snapshot copy

Before restoring Snapshot copies, ensure that you are not using the file system, and that you do not interrupt the restore process.

Before restoring a Snapshot copy, consider the following important information:

- Ensure you are not in any directory on a file system that you want to restore. You can perform the `snapdrive snap restore` command from any directory except the one on a file system to which you want to restore the information.
- Do not interrupt the restore operation by entering Ctrl-C. Doing so could leave your system in an unusable configuration. If that happens, you might have to contact the NetApp technical support to recover.
- When exporting the NFS entities to a volume, set the Anonymous User ID option to “0” for the `snapdrive snap restore` command to work successfully.

Information required for using the snapdrive snap restore command

To restore a Snapshot copy, determine which storage entity you want to restore, specify the name of the Snapshot copy, and so on.

The following table gives the information you need to supply when you use the `snapdrive snap restore` command.

Requirement/Option	Argument
Decide the type of storage entity that you wish to restore and enter that entity's name with the appropriate argument. • If you specify a host volume or file system to be restored, the argument you give is translated to the disk group containing it. SnapDrive for UNIX then restores the entire disk group. SnapDrive for UNIX freezes any file systems in host volumes in those disk groups and takes a Snapshot copy of all storage system volumes containing LUNs in those disk groups. • If you specify a file specification that is an NFS mountpoint, the argument translates to a directory tree. SnapDrive for UNIX restores only the NFS directory tree or file. Within the directory tree, SnapDrive for UNIX deletes any new NFS files or directories that you created after you created the Snapshot copy. This ensures that the state of the restored directory tree will be the same as when the Snapshot copy of the tree was made. • If you restore a LUN, SnapDrive for UNIX restores the LUN you specify. • If you restore a file system that is created directly on a LUN, SnapDrive for UNIX restores the LUN and the file system. • If the Snapshot copy contains storage entities that span multiple storage system volumes, you can restore any of the entities in that Snapshot copy.	
LUN (`-lun` *`file_spec`*)	*name of the LUN. You must include the name of the storage system, volume, and LUN.*
Disk group (`-dg` *`file_spec`* `c`) or volume group (`-vg` *`file_spec`*)	*name of the disk or volume group*
File system (`-fs` *`file_spec`*)	*name of the file system*
File (`-file` *`file_spec`*)	*name of the NFS file*
Host volume (`-hostvol` *`file_spec`*) or logical volume (`-lvol` *`file_spec`*)	*name of the host or logical volume. You must supply both the requested volume and the disk group containing it; for example, - hostvol dg3/acct_volume.*

Requirement/Option	Argument
Specify the name for the Snapshot copy. If any of the *file_spec* arguments you supply at the command prompt currently exist on the local host, you can use a short form of the Snapshot copy name. If none of the *file_spec* arguments exist on the host, you must use a long form of the Snapshot copy name where you enter the storage system name, volume, and Snapshot copy name. If you use a long name for the Snapshot copy and the path name does not match the storage system and/or storage volume information at the command prompt, SnapDrive for UNIX fails. The following is an example of a long Snapshot copy name: `big_filer:/vol/account_vol:snap_20031115` Sometimes, the value supplied with the *file_spec* argument might not exist on the host. For example, if you had unmounted a file system or removed a disk group by exporting, deporting, or destroying, that file system or disk group could still be a value for the *file_spec* argument. It would, however, be considered a non-existent value. SnapDrive for UNIX can restore Snapshot copies for such a non-existent *file_spec*, but you must use the long Snapshot copy name. When you restore Snapshot copies that span multiple storage systems and volumes, and contain a nonexistent *file_spec* argument, SnapDrive for UNIX permits an inconsistency in the command line. It does not allow for existing *file_spec* arguments. If you want to restore only one storage entity from a multiple storage system Snapshot copy, the Snapshot copy you specify does not need to be on the same storage system as the storage system containing the storage entity. The short form of the same Snapshot copy name would omit the storage system and storage system volume name, so it would appear as: *snap_20031115*	
Snapshot copy name (`-snapname`)	*snap_name*
It can be either a short name, such as *mysnap1*, or a long name that includes the storage system name, volume, and Snapshot copy name. Generally, NetApp recommends that you use the short name. If the *file_spec* argument is non-existent: that is, it no longer exists on the host; see the explanation of the *file_spec* argument. Then you must use the long name for the Snapshot copy.	
`-reserve | -noreserve`	
Optional: If you want SnapDrive for UNIX to create a space reservation when you restore the Snapshot copy.	
`-force`	~
`-noprompt`	~

Requirement/Option	Argument
Optional: Decide if you want to overwrite an existing Snapshot copy. Without this option, this operation halts if you supply the name of an existing Snapshot copy. When you supply this option and specify the name of an existing Snapshot copy, it prompts you to confirm that you want to overwrite the Snapshot copy. To prevent SnapDrive for UNIX from displaying the prompt, include the `-noprompt` option also. (You must always include the `-force` option if you want to use the -noprompt option.) You must include the `-force` option at the command prompt if you attempt to restore a disk group where the configuration has changed since the last Snapshot copy. For example, if you changed the way data is striped on the disks since you took a Snapshot copy, you would need to include the `-force` option. Without the `-force` option, this operation fails. This option asks you to confirm that you want to continue the operation unless you include the `-noprompt` option with it. If you added or deleted a LUN, the restore operation fails, even if you include the `-force` option.	
`mntopts`	~
Optional: If you are creating a file system, you can specify the following options: • Use `-mntopts` to specify options that you want to pass to the host mount command (for example, to specify host system logging behavior). The options you specify are stored in the host file system table file. Allowed options depend on the host file system type. • The *`-mntopts`* argument is a file system `-type` option that is specified using the mount command `-o` flag. Do not include the -o flag in the *`-mntopts`* argument. For example, the sequence -mntopts tmplog passes the string `-o tmplog` to the `mount` command, and inserts the text tmplog on a new command line. If you pass any invalid *`-mntopts`* options for storage and snap operations, SnapDrive for UNIX does not validate those invalid mount options.	

Restoring a Snapshot copy

You can restore a Snapshot copy by using the `snapdrive snap restore` command. The restore operation can take several minutes, depending on the type, and amount of data being restored.

Steps

1. Enter the following command to restore a Snapshot copy:

```
snapdrive snap restore -snapname snap_name {-lun |-dg |-fs | - hostvol |-file
}file_spec [file_spec…] [{ -lun | -dg | -fs |-hostvol |-file } file_spec
[file_spec …] [-force [-noprompt]] [-mntopts options]][{-reserve |-noreserve}]
```

SnapDrive for UNIX replaces the contents of the LUNs you specify in the `snapdrive snap restore` command line with the contents of the LUNs in the Snapshot copy you specify. This operation can take several minutes. When the operation is complete, SnapDrive for UNIX displays a message similar to the following:

```
snap restore <filespec list> succeeded
```

In the following example, file system 1 (fs1) resides on storage system1, and file system 2 (fs2) resides on storage system1 and also on storage system 2, which is the partner storage system. File system 3 (fs3) resides on storage system1, partner storage system 2, and storage system3, which is not part of the HA pair. An additional file system, fs4, resides entirely on storage system 4.

The following command creates a Snapshot copy of `fs1`, `fs2`, `fs3`, and, `fs4`:

```
# snapdrive snap create -fs /mnt/fs1 /mnt/fs2 /mnt/fs3 /mnt/fs4
-snapname fs_all_snap
```

The next command restores fs1 and fs2 on the destination storage system. Both fs1 and fs2 reside on a HA pair, so you can restore them with one command:

```
# snapdrive snap restore -fs /mnt/fs1 /mt/fs2 -snapname fs_all_snap
```

The following command restores fs4:

```
snapdrive snap restore -fs /mnt/fs4 -snapname fs_all_snap
```

SnapDrive for UNIX cannot restore fs3 on the destination storage system, because this file system resides on storage system1, storage system 2, and storage system 3.

Restoring a Snapshot copy from a different host

Use the `snapdrive snap restore` command to restore a Snapshot copy from a different host.

Usually, you can restore a Snapshot copy from the host where you took the Snapshot copy. Occasionally, you might need to restore a Snapshot copy using a different or non-originating host. To restore a Snapshot copy using a non-originating host, use the same `snapdrive snap restore` command that you would normally use. If the Snapshot copy you restore contains NFS entities, the non-originating host must have permission to access the NFS directory.

Volume-based SnapRestore

SnapDrive 4.0 for UNIX and later provides Snapshot restore capability at a volume level. This explains the various storage operations you can carry out using volume-based Snapshot restore.

What volume-based SnapRestore is

Volume-based SnapRestore (VBSR) restores the volume with all its storage objects. The volume-based restore is faster than each storage object restored individually. VBSR also works with FlexClone volumes and vFiler configurations. VBSR for vFiler is available for

Data ONTAP 7.3 and later.

SnapDrive 3.0 for UNIX and earlier can restore LUNs for a host-side entity like file system, disk groups, and host volumes, or normal files created over NFS from an application consistent snapshot. SnapDrive for UNIX uses Single File Snap Restore SFSR implemented in Data ONTAP. SFSR works as following:

- For normal files while the SFSR is proceeding, any operation which tries to change the file is suspended until SFSR is complete.
- For LUNs, when SFSR is proceeding the LUN is available and I/Os (both reads and writes) are allowed. SFSR for normal files as well as LUNs may take a long time depending on the size of the LUN or the file being restored.

Therefore, for some environments SFSR is an obstruction.

SnapDrive 4.0 for UNIX and later enables you to take volume-based Snapshot copies. VBSR is much faster and requires less CPU and storage resources. It restores all the data on the active file system. This functionality can be used if a user wants to restore all LUNs or normal files on a volume from the same Snapshot copy.

Volume-based Snapshot copy should be used with caution because all Snapshot copies taken after volume Snapshot copy that is used for the restore operation are deleted. All the new files and new LUNs created on this volume must be deleted.

Considerations for using volume-based SnapRestore

You can use volume-based SnapRestore (VBSR) with keeping few points in mind. Keeping these points in mind helps you to use the VBSR feature safely.

You must keep the following points in mind:

- VBSR reverts the entire volume to the state of the time when the Snapshot copy is created, which is used for VBSR. VBSR includes the following:
  - All the files and LUNs for the specified host filespec during `snap create` operation.
  - All the files and LUNs that are part of the application consistent Snapshot copy during `snap create` operation.
- VBSR removes all the newer files and LUNs created on the current volume after the Snapshot copy used for restore.
- VBSR removes all the newer Snapshot copies that are taken after the Snapshot copy which is being used for restore.
- It is recommended that you run `-vbsr preview` command before using `-vbsr execute` command.

Mandatory checks for volume-based SnapRestore

Before volume based SnapRestore is carried out, SnapDrive for UNIX does some mandatory checks with the system. These checks are required so that volume based SnapRestore is used safely. The mandatory checks cannot be over-ridden by the user.

The following are the mandatory checks which SnapDrive for UNIX makes before implementing volume-based SnapRestore:

- Volume-based SnapRestore works only with Snapshots which are created using SnapDrive for UNIX.
- The volume in volume-based SnapRestore should not be a storage system's root volume.
- SnapDrive for UNIX checks for volume clones. It does not allow the volume restore operation if there are any volume clones from new Snapshot copies. This is a limitation imposed by Data ONTAP.
- The volume in volume-based SnapRestore should not have any mapped LUNs apart from the LUNs specified (raw LUN or LUNs present in file system, disk group or host volume) for restore.
- SnapDrive for UNIX checks whether the volume exists in a SnapMirror relationship.
- SnapDrive for UNIX checks whether the volume exists in a SnapVault relationship.

The SnapMirror and SnapVault checks can be overridden if SnapDrive for UNIX is using Operations Manager for RBAC and user has SD.SnapShot.DisruptBaseline capability on volume. For more information about the specific RBAC capability for user to override these checks, you can see Role Based Access Control in SnapDrive for UNIX.

Related information

Role-based access control in SnapDrive for UNIX

Checks that can be overridden by the user

Before volume-based SnapRestore is carried out, SnapDrive for UNIX performs some checks that a user can override using `-force` option. These checks are required so that volume-based SnapRestore is used safely.

It is recommended that you follow the various checks that the system performs, but you can override these checks by using the `-force` option.

You can override the following SnapDrive for UNIX for volume-based SnapRestore checks:

- SnapDrive for UNIX checks for LUNs in the Snapshot copy which are not crash consistent. If it finds an application inconsistent LUN in the Snapshot copy, it warns you about it.
- SnapDrive for UNIX checks whether there are additional LUNs in the active volume which were created after the Snapshot copy was taken. If SnapDrive for UNIX finds additional LUNs, it warns you that those additional LUNs in the active volume is lost.
- SnapDrive for UNIX checks for new Snapshot copies. These new Snapshot copies do not get restored and are lost.
- SnapDrive for UNIX checks for normal file (files that are visible from the local host) in a volume.
- SnapDrive for UNIX checks for NFS exports.
- SnapDrive for UNIX checks for CIFS shares.

If Snapshot copies were created using SnapDrive 3.0 for UNIX or earlier, volume-based SnapRestore preview is not able to perform the first two checks in the earlier mentioned list. If you have provided `-force` option, then during volume-based SnapRestore execute, a prompt message is displayed to over-ride these checks and proceed.

Volume-based SnapRestore command

This section describes the commands and the options to use volume based SnapRestore.

`-vbsr` option is added in snap restore CLI to select volume based SnapRestore. Use the following command syntax to perform restore using volume based SnapRestore:

```
snapdrive snap restore {-lun | -dg | -vg | -hostvol |
-lvol | -fs |-file} file_spec [file_spec ...] [{-lun | -dg |
-vg | -hostvol | -lvol | -fs -file} file_spec [file_spec ...]
...] -snapname snap_name [-force [-noprompt]][{-reserve |
-noreserve}]
[-vbsr [execute | preview]]
```

If no argument is supplied with the `-vbsr`, the default output is that of the `preview` option. A `-verbose` option is used which enables detailed output of all the mandatory checks that can be over-ridden by the user. The default output when `-verbose` option is not provided displays the results of the checks that is failed.

If you do not want any confirmation message to prompt, while executing volume based SnapRestore, you can use `-noprompt` and `-force` option with `snap restore -vbsr execute` command. The following table describes SnapDrive for UNIX behavior depending on the options provided by you.

S.N.	-vbsr execute	-force	-noprompt	Result
1.	No	NA	NA	Preview mode is the default mode. All the checks are done and report for each check is generated.
2.	Yes	No	No	All checks are done. If any mandatory checks that a user can override fails, SnapDrive for UNIX displays an error message.
3.	Yes	Yes	No	All the checks are done. If any mandatory checks fail, SnapDrive for UNIX displays an error message. If any check that a user can override fails, SnapDrive for UNIX prompts you.

S.N.	-vbsr execute	-force	-noprompt	Result
4.	Yes	Yes	Yes	All the checks are done. If any mandatory checks fail, SnapDrive for UNIX displays an error message. If any check that a user can override fails, SnapDrive for UNIX does not prompt you.

Information about LUNs mapped to local or remote hosts

The only mapping information available from the volume to SnapDrive for UNIX during volume based SnapRestore is the initiator group information for a LUN. If the initiator groups used by you are always created by SnapDrive for UNIX, then the fully qualified domain name of the host is part of the initiator group name.

If SnapDrive for UNIX administrator specifies the `-igroup` CLI option or if you use manually created initiator groups, then the `igroup` name need not necessarily have the host name. For all the earlier reasons, SnapDrive for UNIX cannot reliably detect local or remote bindings for a LUN. Therefore, SnapDrive for UNIX displays the full LUN initiator group and initiator information as part of the volume based SnapRestore.

Host filespec information for a particular volume

SnapDrive for UNIX as part of volume restore preview report displays the LUN mapping information. This displayed information is relevant for the checks and the normal files are reverted. Finding out all the host filespecs based on LUNs on a particular volume is a time consuming process and slows down the volume restore process.

If you want to know that the host filespec information for the local host mapped to a particular storage system volume, you can use `snapdrive storage show -filervol <full-volume-name>`. An example of this is shown in the following.

```
#snapdrive storage show -filervol bart:/vol/volusecase2

Connected LUNs and devices:

 device filename        adapter path    size    proto   state   clone
lun path           backing snapshot
----------------        ------- ----    ----    -----   -----   -----
--------                        ----------------
/dev/sdg                  -     P       100m    iscsi   online   No
bart:/vol/volusecase2/lun5          -

Host devices and file systems:

dg: vbsrfs_1_SdDg               dgtype lvm
hostvol: /dev/mapper/vbsrfs_1_SdDg-vbsrfs_1_SdHv        state: AVAIL
fs: /dev/mapper/vbsrfs_1_SdDg-vbsrfs_1_SdHv     mount point: /mnt/vbsrfs_1
(persistent) fstype jfs2

 device filename     adapter path    size    proto   state   clone   lun
path           backing snapshot
----------------        ------- ----    ----    -----   -----   -----
--------                        ----------------
/dev/sdbe                 -     P       100m    iscsi   online   No
bart:/vol/volusecase1/lun9_0        -
/dev/sdbf                 -     P       100m    iscsi   online   No
bart:/vol/volusecase2/lun4_0        -

raw device: /dev/sdbr1  mount point: /mnt/fs11 (persistent) fstype jfs2

 device filename     adapter path    size    proto   state   clone   lun
path           backing snapshot
----------------        ------- ----    ----    -----   -----   -----
--------                        ----------------
/dev/sdbr                 -     P       200m    iscsi   online   No
bart:/vol/volusecase2/fs11_SdLun    -

NFS device: bart:/vol/volusecase1        mount point: /mnt/volusecase1
(non-persistent)

LUNs not connected to this host:

 lun path                                            size   state
 ------------------------------               ------ ------
 bart:/vol/volusecase2/lunotherhost 20m   online
```

Volume-based SnapRestore for space reservation

For space reservation with volume based snap reserve you need to set *space-reservations-volume-enabled* option in the `snapdrive.conf` file.

The *space-reservations-volume-enabled* option is used to set space guarantee policy on volume and can take the following three values.

- **Snapshot**: This is the default value. Space guarantee on the volume does not change.
- **volume**: Space guarantee on the volume is at the volume level.
- **none**: Space guarantee is set as none.

Following table describes the behavior of volume-based snap reserve for space reservation.

No space reserve CLI option used; `-vbsr execute` is specified	**space-reservations-volume-enabled=**	**Result**
none	snapshot	Space guarantee on the volume does not change.
none	none	Attempt to set space guarantee as “none” for the volumes.
`-reserve`	configuration value is over-ridden	Attempt to set space guarantee for volumes as "volume"
`-noreserve`	configuration value is over-ridden	Attempt to set space guarantee for volumes as "none"
none	volume	Attempt to set space guarantee for volumes as "volume"

`-vbsr preview` does not check for any of the space reservation options.

Connecting to a Snapshot copy

You can connect a Snapshot copy from one host to a different host.

SnapDrive for UNIX lets you connect a host to a Snapshot copy from a different location on a host. This new location can be on the host where you took the Snapshot copy (the originating host) or on a different host (the non-originating host).

Being able to set up the Snapshot copies in a new location means you can back up a Snapshot copy to another medium, perform maintenance on a disk group, or access the Snapshot copy data without disrupting the original copy of the data.

With this command, you can connect a host to a Snapshot copy that contains any of the following:

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes, and file systems created on LUNs
- NFS directory trees
- Disk groups, host volumes, and file systems on shared storage system

How the snapdrive snap connect command works

When you use the `snapdrive snap connect` command, it clones the storage for the entity you specify and imports it to the host:

- If you specify a Snapshot copy that contains a LUN (`-lun`), SnapDrive for UNIX maps a new copy of the LUN to the host. You cannot use the `snapdrive snap connect` command to specify a LUN on the same command line with other storage entities (`-vg`, `-dg`, `-fs`, `-lvol`, or `- hostvol`).
- If you specify a file system that resides directly on a LUN, SnapDrive for UNIX maps the LUN to the host and mounts the file system.
- If the source mount point is specified as the relative pathname in the `snap connect` command, SnapDrive for UNIX ignores the destination mount point specified in CLI and uses internal naming convention of the format `source_mount_point_<N>` to name the destination mount point.
- If you specify a Snapshot copy that contains a disk group, or a host volume or file system that is part of a disk group, the `snapdrive snap connect` command connects the entire target disk group. To make the connection, SnapDrive for UNIX re-activates all of the logical volumes for the target disk group and mounts all the file systems on the logical volumes.
- If you specify `autorename` option with the `snap connect` command, host volumes and file systems are always renamed. The disk groups are renamed only if they already exist on the host.
- If you specify a Snapshot copy that contains an NFS directory tree, SnapDrive for UNIX creates a clone of the FlexVol volume that contains the NFS directory tree. SnapDrive for UNIX then connects the volume to the host and mounts the NFS file system. Within the directory tree, SnapDrive for UNIX deletes any new NFS files or directories that you create after you created the Snapshot copy. SnapDrive for UNIX deletes any files or directories from the FlexVol volume that are outside the NFS directories that you connect, if the `snapconnect-nfs-removedirectories` configuration option is set to on.
- If you connect a Snapshot copy that contains NFS directory trees using the `-readonly` option, SnapDrive for UNIX mounts the Snapshot copy of the directory directly without creating a clone. You cannot use the `snapdrive snap connect` command to specify NFS mountpoints on the same command line as non-NFS entities; that is, using the options `-vg`, `-dg`, `-fs`, `-lvol`, or `- hostvol`.

> The `snap connect` operations with `-split` option in vFiler environment are supported with Data ONTAP 7.3 and later.

Connecting Snapshot copies on mirrored storage systems

In the case of a Snapshot copy on a mirrored storage system, you can connect the Snapshot copy on the source storage system and the destination system.

When you create a Snapshot copy on a mirrored storage system, the Snapshot copy is automatically replicated, from the source system where it is created, to the destination (mirrored) storage system. SnapDrive for UNIX allows you to connect the Snapshot copy on the source storage system. You can also connect the

Snapshot copy on the destination storage system.

Connecting multiple storage entities

You can connect a Snapshot copy containing multiple storage entities.

To connect a Snapshot copy that contains storage entities that reside on multiple destination storage systems you must meet the following requirements:

- The storage entities you specify at the command prompt must reside on a single storage system, or on a HA pair.
- The name of the volume of the source storage system must match the name of the volume of the destination storage system.
- You must set the *`snapmirror-dest-multiple-filervolumes-enabled`* variable in the snapdrive.conf file to “on”.

You can use one command to connect storage entities that reside on a single storage system or on a HA pair.

Snapshot connect and Snapshot restore operations

Snapshot copy clones the information when you connect to the Snapshot copy.

Unlike the Snapshot restore operation, the Snapshot connect operation does not replace the existing contents of the LUNs that make up the host entity with the Snapshot copy contents. It clones the information.

After the connection is established, both Snapshot connect and Snapshot restore operations perform similar activities:

- The Snapshot connect operation activates logical volumes for the storage entity, mounts file systems, and optionally adds an entry to the host file system table.
- The Snapshot restore operation activates the logical volumes for the storage entity, mounts the file systems, and applies the host file system mount entries that were preserved in the Snapshot copy.

Guidelines for connecting Snapshot copies

Follow the guidelines when connecting to Snapshot copies.

- The `snapdrive snap connect` command works only with Snapshot copies created in SnapDrive 4.2 for UNIX.
- On an originating host, SnapDrive for UNIX supports connecting and restoring Snapshot copies that are created by previous versions of SnapDrive for UNIX.
- For read and write access to NFS directory trees, the `snapdrive snap connect` command uses the Data ONTAP FlexVol volume feature, and therefore requires Data ONTAP 7.3 or later. Configurations with Data ONTAP 7.1 can connect NFS files or directory trees, but are provided with read-only access.
- If you set the *`enable-split-clone`* configuration variable value to “on” or “sync” during the Snapshot connect operation and “off” during the Snapshot disconnect operation, SnapDrive for UNIX does not delete the original volume or LUN that is present in the Snapshot copy.
- You have to set the value of Data ONTAP 7.2.2 configuration option vfiler. `vol_clone_zapi_allow` to “on” to connect to a Snapshot copy of a volume or LUN in a vFiler unit.

- The Snapshot connect operation is not supported on the hosts having different host configurations.
- The `snapdrive snap connect` command used to connect to a root volume of a physical storage system or a vFiler unit fails because Data ONTAP does not allow cloning of a root volume.

Information required for using the snapdrive snap connect command

To connect to a Snapshot copy, determine the type of storage entity, connect a Snapshot copy with the NFS directory tree to Data ONTAP 7.3 configurations, and so on.

The following table gives the information you need to supply when you use the `snapdrive snap connect` command.

> `snap connect` operations require Storage Foundation Enterprise License for vximport on AIX with Veritas.

Requirement/Option	Argument
Decide the type of storage entity that you want to use to attach the Snapshot copy and supply that entity's name with the appropriate argument. This is the value for the *`src_fspec`* argument. • If you connect a Snapshot copy of a LUN, SnapDrive for UNIX connects the LUN you specify. You cannot use the `-lun` option on the same command line with the `-vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol` options. You can specify the short name of the LUN in the lun_name or qtree_name/lun_name format. • If you connect a Snapshot copy of a file system that is created directly on a LUN, SnapDrive for UNIX connects the LUN that has the file system. • If you connect a Snapshot copy of a disk group that has a host volume or file specification, the argument translates into a set of disk groups on the storage system. SnapDrive for UNIX connects the entire disk group containing the entity, even if the entity is a host volume or file system. • If you connect a Snapshot copy of an NFS file system, the argument translates to the NFS directory tree. SnapDrive for UNIX creates a FlexClone of the volume, removes directory trees that are not specified in the Snapshot copy, and then connects and mounts the NFS directory tree. If you specify an NFS mount point, you cannot specify non-NFS entities (`-vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol`) on the same command line. > SnapDrive for UNIX does not support symbolic links at the mount point level.	
LUN (`-lun` *`file_spec`*)	*short name of the LUN.*
The *`s_lun_name`* specifies a LUN that exists in the `-snapname` *`long_snap_name`*. The short lun_name is required. You cannot include a storage system or storage system volume name. The *`d_lun_name`* specifies the name at which the LUN is connected. The short *`lun_name`* is required. You cannot include a storage system or storage system volume name. You must specify a *`d_lun_name`*	
Disk group (`-dg` *`file_spec`*) or volume group (`-vg` *`file_spec`*)	*name of the disk or volume group*
File system (`-fs` *`file_spec`*)	*name of the file system*

Requirement/Option	Argument
Host volume (`-hostvol` *`file_spec`*) or logical volume (`-lvol` *`file_spec`*)	*name of the host or logical volume*
Connect a Snapshot copy with an NFS directory tree to Data ONTAP 7.3 configurations. • If your configuration uses Data ONTAP 7.3 or a later version of Data ONTAP with traditional (not FlexVol) volumes, you must specify this option to connect the Snapshot copy with readonly access (required). • If your configuration uses Data ONTAP 7.3 and later and FlexVol volumes, SnapDrive for UNIX automatically provides read-write access. Specify this option only if you want to restrict access to read-only (optional).	
-readonly	*set read only permission*
Optional: Supply a name by which the target entity is accessible after the storage entity is connected. SnapDrive for UNIX uses this name to connect the destination entity. This is the *`dest_file_spec`* argument. If you omit this name, the `snap connect` command uses the value you supplied for *`src_fspec`*.	
Name of target entity	*`dest_file_spec`*
Optional: Specify the names for the destination storage entities. If you included this information as part of the *`dest_fspec/src_fspec`* pair, you do not need to enter it here. You can use the `-destxx` options to specify names for destination storage entities if this information is not part of the *`dest_fspec/src_fspec`* pair. For example, the `-fs` option names only a destination mount point so you can use the `-destdg` option to specify the destination disk group. If you do not specify the name needed to connect an entity in the destination disk group, the `snapdrive snap connect` command takes the name from the source disk group. If you do not specify the name needed to connect an entity in the destination disk group, the `snap connect` command takes the name from the source disk group. If it cannot use that name, the operation fails, unless you included `-autorename` at the command prompt.	
Destination disk group (`-destdg`) or destination volume group (`-destvg`)	*`dgname`*
Destination logical volume (`-destlv`) or destination host volume (`-desthv`)	*`lvname`*
Specify the name for the Snapshot copy. Use the long form of the name where you enter the storage system name, volume, and Snapshot copy name.	
Snapshot copy name (`-snapname`)	*`long_snap_name`*

Requirement/Option	Argument
`-nopersist`	~

Optional: Connect the Snapshot copy to a new location without creating an entry in the host file system table.
 	
• The `-nopersist` option allows you to connect a Snapshot copy to a new location without creating an entry in the host file system table. By default SnapDrive for UNIX creates persistent mounts. This means that:
  ◦ When you connect a Snapshot copy on a AIX host, SnapDrive for UNIX mounts the file system and then places an entry for the LUNs that comprise the file system in the host's file system table.
  ◦ You cannot use `-nopersist` to connect a Snapshot copy that contains an NFS directory tree.	

`-reserve | -noreserve`	~
Optional: Connect the Snapshot copy to a new location with or without creating a space reservation.	
Igroup name (`-igroup`)	*`ig_name`*
Optional: NetApp recommends that you use the default igroup for your host instead of supplying an igroup name.	
`-autoexpand`	~
To shorten the amount of information you must supply when connecting to a volume group, include the `-autoexpand` option at the command prompt. This option lets you name only a subset of the logical volumes or file systems in the volume group. It then expands the connection to the rest of the logical volumes or file systems in the disk group. In this manner, you do not need to specify each logical volume or file system. SnapDrive for UNIX uses this information to generate the name of the destination entity. This option applies to each disk group specified at the command prompt and all host LVM entities within the group. Without the `-autoexpand` option (default), you must specify all affected host volumes and file systems contained in that disk group to connect the entire disk group. ⓘ If the value you enter is a disk group, you do not need to enter all the host volumes or file systems because SnapDrive for UNIX knows what the disk group is connecting to. NetApp recommends that, if you include this option, you should also include the `-autorename` option. If the `-autoexpand` option needs to connect the destination copy of an LVM entity, but the name is already in use, the command fails unless the `-autorename` option is at the command prompt.	
The command fails if you do not include -autoexpand and you do not specify all the LVM host volumes in all the disk groups that is referred at the command prompt (either by specifying the host volume itself or the file system).	
`-autorename`	~

Requirement/Option	Argument
When you use the `-autoexpand` option without the `-autorename` option, the `snap connect` command fails if the default name for the destination copy of an LVM entity is in use. If you include the `-autorename` option, SnapDrive for UNIX renames the entity when the default name is in use. This means that with the -autorename option at the command prompt, the Snapshot connect operation continues regardless of whether all the necessary names are available. This option applies to all host-side entities specified at the command prompt. If you include the `-autorename` option at the command prompt, it implies the -autoexpand option, even if you do not include that option.	
`-split`	~
Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations.	
`mntopts`	~
Optional: If you are creating a file system, you can specify the following options: • Use `-mntopts` to specify options that you want to pass to the host mount command (for example, to specify host system logging behavior). The options you specify are stored in the host file system table file. Allowed options depend on the host file system type. • The *`-mntopts`* argument is a file system `-type` option that is specified using the mount command `-o` flag. Do not include the `-o` flag in the *`-mntopts`* argument. For example, the sequence -mntopts tmplog passes the string `-o tmplog` to the `mount` command, and inserts the text tmplog on a new command line. ⓘ If you pass any invalid *`-mntopts`* options for storage and snap operations, SnapDrive for UNIX does not validate those invalid mount options.	

Connecting to a Snapshot copy that contains LUNs

You can connect to a Snapshot copy that contains LUNs by using the `snapdrive snap connect` command.

Steps

1. Enter the following command syntax to connect to a Snapshot copy that contains LUNs:

   `snapdrive snap connect -lun` ***`s_lun_name d_lun_name`*** **`[[-lun]`** ***`s_lun_name d_lun_name`*****`… -snapname`** ***`long_snap_name`*** **`[-igroup`** ***`ig_name`*** **`[`***`ig_name`***`…]] [-split]`**

   ⓘ The *s_lun_name* and *d_lun_name* should be in the format lun_name or qtree_name/lun_name.

SnapDrive for UNIX clones the LUNs you specify and connects them to a new location.

The following example connects the LUN mylun1, in hornet/vol/vol1/tuesdaysnapshot to mylun1copy:

```
# ./snapdrive snap connect -lun mylun1 mylun1copy -snapname
hornet:/vol/vol1:tuesdaysnapshot
connecting hornet:/vol/vol1/mylun1:
LUN copy mylun1copy ... created
(original: hornet:/vol/vol1/mylun1) mapping new lun(s) ... done
discovering new lun(s) ... done
```

The following example connects two LUNs, mylun1 and mylun2, to mylun1copy and mylun2copy, respectively:

```
# ./snapdrive snap connect -lun mylun1 mylun1copy -lun mylun2
mylun2copy -snapname hornet:/vol/vol1:tuesdaysnapshot
connecting hornet:/vol/vol1/mylun1:
LUN copy mylun1copy ... created
(original: hornet:/vol/vol1/mylun1)
mapping new lun(s) ... done
connecting hornet:/vol/vol1/mylun2:
LUN copy mylun2copy ... created
(original: hornet:/vol/vol1/mylun2)
mapping new lun(s) ... done
discovering new lun(s) ... done
```

Connecting to a Snapshot copy of storage entities other than LUNs

You can use the `snapdrive snap connect` command for connecting to a Snapshot copy that contains storage entities other than LUNs. This command cannot be used when destination names you provide are currently in use or, there is a file system name that is used as a mountpoint.

When you connect from a non-originating host to a Snapshot copy containing the VxFS file system mounted with the default mount `qio` option, you should have the Veritas license for Veritas File Device Driver (VxFDD) installed.

Steps

1. Enter the following command:

   snapdrive snap connect -snapname fspec_set [*fspec_set*…] -snapname *long_snap_name* [-igroup *ig_name* [*ig_name*…]] [-autoexpand] [-autorename] [-nopersist] [{-reserve | -noreserve}] [-readonly] [-split]

   In the preceding usage, *`fspec_set`* has the following format:

   {-dg | -fs | -hostvol} *src_file_spec* [*dest_file_spec*] [{-destdg | -destvg} *dgname*] [{-destlv | -desthv} lvname]

   This command must always start with the name of the storage entity you want to connect (for example,

-dg, -hostvol, or `-fs). If you specify an NFS mountpoint, you cannot specify non-NFS entities (-vg, -dg, -fs, -lvol or -hostvol) on the same command line.

SnapDrive for UNIX clones the LUNs you specify and connects them to a new location.

The following command line connects a disk group and uses the default names as the destination names (that is, it creates them from the source names):

```
# snapdrive snap connect -vg vg1 -snapname
filer1:/vol/vol1:vg1snapshot
connecting vg1:
LUN copy vg1_lun1_0 ... created
(original: filer1:/vol/vol1/vg1_lun1)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing vg1
```

The following command line connects a disk group with a single host volume. It also specifies a name for the destination host volume and disk group:

```
# snapdrive snap connect -lvol vg1/vol1 vg1copy/vol1copy -snapname
filer1:/vol/vol1:vg1snapshot
connecting vg1:
LUN copy vg1_lun1_0 ... created
(original: filer1:/vol/vol1/vg1_lun1)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing vg1copy
```

The following command line connects a disk group with two LUNs and two file systems. It specifies a destination name for each of the file systems, the host volume for one of the file systems, and the disk groups for both file systems:

```
# snapdrive snap connect -fs mnt/fs1 /mnt/fs1copy -destvg vg1copy \
-fs /mnt/fs2 /mnt/fs2copy -destlv vg1copy/vol2copy -destvg vg1copy
\ -snapname filer1:/vol/vol1:vg1snapshot
connecting vg1:
LUN copy vg1_lun1_0 ... created
(original: filer1:/vol/vol1/vg1_lun1)
LUN copy vg1_lun2_0 ... created
(original: filer1:/vol/vol1/vg1_lun2)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing vg1copy
```

The following command line includes the -autoexpand option as it connects a disk group with two file systems. It uses the default names as the destination names (that is, it creates them from the source names):

```
# snapdrive snap connect -lvol mnt/fs1 -snapname
filer1:/vol/vol1:vg1snapshot \
-autoexpand
connecting vg1:
LUN copy vg1_lun1_0 ... created
(original: filer1:/vol/vol1/vg1_lun1)
LUN copy vg1_lun2_0 ... created
(original: filer1:/vol/vol1/vg1_lun2)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing vg1
```

The following command line includes the `-autorename` option as it connects a disk group with two file systems and two LUNs:

```
# snapdrive snap connect -fs mnt/fs1 -snapname
filer1:/vol/vol1:vg1snapshot \
-autorename
connecting vg1:
LUN copy vg1_lun1_0 ... created
(original: filer1:/vol/vol1/vg1_lun1)
LUN copy vg1_lun2_0 ... created
(original: filer1:/vol/vol1/vg1_lun2)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing vg1_0
```

The following example connects to a Snapshot copy with file system, disk group created on LVM stack:

```
# snapdrive snap connect -fs /mnt/jfs1 /mnt/jfs1_clone -snapname
snoopy:/vol/vol1:snapLvm -autorename
connecting lvm1:
LUN copy lunLvm1_0 ... created
(original: snoopy:/vol/vol1/lunLvm1)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing lvm1_0
Successfully connected to snapshot snoopy:/vol/vol1:snapLvm
disk group lvm1_0 containing host volumes
jfs1_SdHv_0 (filesystem: /mnt/jfs1_clone)
```

In the following example, file system 1 (fs1) resides on storage system1, and file system 2 (fs2) resides on storage system1 and also on storage system2, which is the partner storage system. File system 3 (fs3) resides on storage system1, partner storage system 2, and storage system 3, which is not part of the HA pair. An additional file system, fs4, resides entirely on storage system 4.

The following command creates a Snapshot copy of fs1, fs2, fs3, and fs4:

```
snapdrive snap create -fs /mnt/fs1 /mnt/fs2 /mnt/fs3 /mnt/fs4
-snapname fs_all_snap
```

The next command connect fs1 and fs2 on the destination storage system. Both fs1 and fs2 reside on a HA pair, so you can restore them with one command:

```
snapdrive snap connect -fs /mnt/fs1 /mt/fs2 -snapname fs_all_snap
```

The following command restores fs4:

```
snapdrive snap connect -fs /mnt/fs4 -snapname fs_all_snap
```

SnapDrive for UNIX cannot connect fs3 on the destination storage system, because this file system resides on storage system1, storage system 2, and storage system 3.

Connecting to Snapshot copies of shared storage entities other than LUNs

You can use the `snapdrive snap connect` command to connect to a Snapshot copy of shared storage entities other than LUNs.

Steps

1. Enter the following command:

   snapdrive snap connect *fspec_set* [*fspec_set*…] -snapname *long_snap_name*

   ***fspec_set* [*fspec_set*…] -snapname *long_snap_name* [-devicetype shared] [-split]**

   In this syntax, *fspec_set* is: `{-dg | -fs}_src_file_spec_ [*dest_file_spec*] [-destdg *dgname*]

Splitting a volume clone or LUN clone

SnapDrive for UNIX enables you to split a volume clone or LUN clone. After the clone split is complete, the relationship between the parent volume and the clone is destroyed, and both the entities are independent of each other, and have their own individual storage space.

The following are the clone split operations:

- Estimate the disk space (in MB) for a volume clone or LUN clone.
- Split a volume clone or LUN clone.
- Stop the volume clone or LUN clone split.
- View the status of the clone split that is in progress, completed, or failed.

> - If a volume clone is split, all Snapshot copies in the cloned volume are deleted.
> - It is mandatory to run the clone split estimate command before splitting the file specification to determine, if there are any Snapshot copies taken in the cloned volume.
> - For all clone split commands, only long LUN name must be specified with `-lun` option. You cannot specify `-lun` option on the same command line along with other storage entities (-vg, -dg, -fs, - lvol, or -hostvol options).
> - It is always mandatory to use the absolute pathname for the file specifications with clone split commands.
> - The LUN clone split estimation using Snapshot is available only for Snapshot copies that are created from SnapDrive 4.2 for UNIX and later.

Estimating the storage space to split a volume clone

The clone split estimation helps you to estimate the required storage space (in MB) to split a volume clone. Depending on the clone split estimation provided by SnapDrive for UNIX, you can determine the space availability to split a volume clone.

Steps

1. Enter the following command to estimate the required storage space to split a volume clone.

   snapdrive clone split estimate [-lun] *long_lun_name* [long*lun_name*…] | *[{-dg* | *-vg* | *-fs* | *-hostvol* | *-lvol}* *_file_spec* [*file_spec*…]] | [-snapname *long_snap_name*] {-volclone|-lunclone}] [-v | -verbose] [-dump | -dumpall]

   This operation displays the following information:

   - Resource name
   - Container - Aggregate for a FlexClone
   - Required Space - space required to split the volume clone
   - Available space - space available on the container
   - Storage Status - indicates the space availability for a volume clone split
   - Owned Space - space occupied by the volume clone
   - Shared space - space occupied by the volume clone along with the parent

   The `Owned Space` and `Shared Space` is displayed when you use `-verbose` option.

The following example estimates the storage space to split a volume clone.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -fs /mnt/my_mnt2
Resource     Container  Required Available  Storage
                        Space(MB) Space(MB) Status
/mnt/my_mnt1  f3050-220  400      61500    AVAILABLE
              -111:aggr0
/mnt/my_mnt2  f3050-220  3292     1129     NOT AVAILABLE
              -112:aggr1
```

For every file specification, SnapDrive for UNIX determines the required space that is available in the storage system to split a volume clone. Here, the `/mnt/my_mnt1` file specification has the required space to split, and thus the storage status displays as AVAILABLE. Whereas, the `/mnt/my_mnt2` file specification does not have the required space to split, and so the storage status displays as NOT AVAILABLE.

The following is an example of using the `-verbose` option. Alternatively, you can use `-v` option.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -verbose
Resource    Container Owned    Shared   Required Available Storage
                      Space(MB)Space(MB)Space(MB)Space(MB)  Status
/mnt/my_mnt1 f3050-220 32365  403      403     55875 AVAILABLE
             -111:aggr0
```

Estimating the storage space to split a LUN clone

The clone split estimation helps you to estimate the required storage space (in MB) to split a LUN clone. Depending on the clone split estimation provided by SnapDrive for UNIX, you can determine the space availability to split a LUN clone.

Steps

1. Enter the following command to estimate the required storage space to split a LUN clone.

   snapdrive clone split estimate *long_lun_name* [*long_lun_name*…] | [{-dg | -vg | -fs | -hostvol | -lvol}*file_spec* [*file_spec*…]] | [-snapname *long_snap_name*] {-volclone|-lunclone}] [-v | -verbose]

   This operation displays the following information:

   - Resource name
   - Container- Volume for a LUN clone
   - Required Space - space required to split a LUN clone
   - Available space - space available on the container
   - Storage Status - indicates the space availability for a LUN clone split
   - Owned Space - space occupied by the LUN clone
   - Shared Space - space occupied by the LUN clone along with the parent

   The `Owned Space` and `Shared Space` is displayed when you use `-verbose` option.

The following example estimates the storage space to split a LUN clone.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1
Resource     Container Required Available Storage
                       Space(MB)Space(MB)  Status
/mnt/my_mnt1 f3050-220  5120    9986    AVAILABLE
             -112:/vol/vol_1
```

The following is an example of using the `-verbose` option. Alternatively, you can use `-v` option.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -verbose
Resource    Container  Owned    Shared   Required Available Storage
                       Space(MB)Space(MB)Space(MB)Space(MB)  Status
/mnt/my_mnt1 f3050-220  365   403      5120     9986 AVAILABLE
             -112:/vol/vol_1
```

Estimating the storage space using a Snapshot copy

The clone split estimation helps you to estimate the required storage space (in MB) using a Snapshot copy, when there is no clone available for a Snapshot copy in the storage system.

Steps

1. Enter the following command to estimate the required storage space.

   snapdrive clone split estimate -snapname [*long_snap_name*] {-volclone|-lunclone}] [-v | -verbose]

The following example estimates the storage space to split a LUN clone using a Snapshot copy.

```
snapdrive clone split estimate -snapname f3050-220-112:/vol/vol_1:snap_1
-lunclone
Resource       Container  Required Available Storage
                          Space(MB)Space(MB)  Status
f3050-220-112: f3050-220    5120       14078    AVAILABLE
/vol/vol_1:snap_1 -112:/vol/vol_1
```

The following example estimates the storage space to split a LUN clone using a Snapshot copy with the `-fs` option.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -snapname f3050-220-
112:/vol/vol_1:snap_1 -lunclone
Resource         Container Required Available Storage
                            Space(MB)Space(MB)  Status
f3050-220-112:   f3050-220  4120     14078  AVAILABLE
/vol/vol_1:snap_1 -112:/vol/vol_1
```

The following example estimates the storage space to split a volume clone using a Snapshot copy with the `-fs` option.

```
# snapdrive clone split estimate -fs /mnt/fs1 -snapname f3050-220-
112:/vol/vol_1:snap_1 -volclone
Resource      Container   Required Available Storage
                          Space(MB)Space(MB)  Status
f3050-220-112:  f3050-220 54019   54517  AVAILABLE
/vol/vol0:snap_1 112:aggr0
```

The following example estimates the storage space to split a volume clone using a Snapshot copy.

```
# snapdrive clone split estimate -snapname f3050-220-112:/vol/vol_1:snap_1
-volclone
Resource      Container   Required Available Storage
                          Space(MB)Space(MB)  Status
f3050-220-112:  f3050-220 54019    54517    AVAILABLE
/vol/vol0:snap_1 112:aggr0
```

- The "Resource" field contains the Snapshot copy name, if the clone split estimate is done for a Snapshot copy.
- If you provide any dead file specification along with the Snapshot copy with `-lunclone` option, the "Required Space" shows as 0.
- The LUN clone split estimation using Snapshot is available only for Snapshot copies that are created from SnapDrive 4.2 for UNIX and later.

Starting the volume clone or LUN clone split

You can start a volume clone or LUN clone split operation.

Steps

1. Enter the following command to start a volume clone or LUN clone split.

   # snapdrive clone split start [-lun] *long_lun_name* [*long_lun_name*…] | [{-dg | -vg | -fs | -hostvol | -lvol} *file_spec* [*file_spec* …]] [-force][-noprompt] [-dump | -dumpall]

The following options can be used when the storage status displays as NOT AVAILABLE.

- You can use the `-force` option to forcibly start the clone split operation and receive a confirmation message that the operation has started.
- You can use the `-noprompt` along with `-force` option to start the clone split start operation without receiving any confirmation message.

> When you start another clone split operation soon after stopping a clone split operation that was in progress, the operation might fail. This issue might occur if the delay between the starting and stopping of the clone split operation was not sufficient to allow the storage system to sync the stop operation.

The following example displays how to split a volume clone:

```
# snapdrive clone split start -fs /mnt/my_mnt4_0 /mnt/my_mnt3_0
Resource    Container  Required Available Storage
                       Space(MB)Space(MB) Status
----------------------------------------------------
/mnt/my_mnt4_0 f3050-220 3295   66033 AVAILABLE
             -111:aggr0
/mnt/my_mnt3_0 f3050-220  293   37707 AVAILABLE
             -112:aggr1

Job ID: B265Dbv8gh
Clone-Split for "/mnt/my_mnt4_0" is started
Clone-Split for "/mnt/my_mnt3_0" is started
```

The following example displays how to split a clone using the *`-force`* option:

```
# snapdrive clone split start -fs /mnt/my_mnt5 /mnt/my_mnt6 -force
Resource    Container Required  Available Storage
                      Space(MB) Space(MB) Status
----------------------------------------------------
/mnt/my_mnt5 f3050-220 1198   20033   AVAILABLE
             -111:aggr0
/mnt/my_mnt6 f3050-220  3294  2196  NOT AVAILABLE
             -112:aggr1
Not enough space available for Clone-Split.  Do you want to continue
(y/n)?y
Clone-Split for "/mnt/my_mnt5" is started
Clone-Split for "/mnt/my_mnt6" is started
```

The following example shows how to directly start a clone using the *`-noprompt`* option meaning there is no confirmation message:

```
# snapdrive clone split start -fs /mnt/my_mnt5 /mnt/my_mnt6 -force
-noprompt
Resource    Container  Required  Available Storage
                       Space(MB) Space(MB) Status
-----------------------------------------------------
/mnt/my_mnt5 f3050-220  1198     20033   AVAILABLE
             -111:aggr0
/mnt/my_mnt6 f3050-220  3294     2196  NOT AVAILABLE
             -112:aggr1
Clone-Split for "/mnt/my_mnt5" is started
Clone-Split for "/mnt/my_mnt6" is started
```

Viewing the status of a volume clone or LUN clone split

You can query the clone split status by using a job ID or file specification. SnapDrive for UNIX indicates the current status of the clone split as in-progress, failed, or complete.

Steps

1. Enter the following command to query the clone split status using a job ID or file specification.

   snapdrive clone split status [-lun] *long_lun_name* [*long_lun_name*…] [{-dg | -vg | -fs | -hostvol | -lvol} *file_spec* [*file_spec*…]][-job *<jobid>*] [-all]

The following example shows the clone split status using a job ID.

```
# snapdrive clone split status -job SVE2oxKXzH
Clone-Split-Status for /fs1-1_3 is 1% Complete
Clone-Split-Status for /fs1_0 is 73% Complete
Clone-Split-Status for /fs1_1 is 73% Complete
Clone-Split-Status for /fs1_2 is 74% Complete
Clone-Split-Status for /fs1_3 is 1% Complete
```

You can check the status of a clone split progress in either of the following ways:

- You can verify the clone by using the

  `snapdrive storage show -fs /mnt/my_mnt`

  or

  `snapdrive storage show -lun long_lun_pathname`

  commands. In either case, the clone type is displayed as a FlexClone or LUN clone if the split has not been completed.

- You can verify the clone split progress state by logging into the storage system and using the following commands in the storage system CLI:

  `vol clone split status vol_name lun clone split status lun_name`

The following example shows a clone split status query made using the file specification:

```
# snapdrive clone split status -fs /mnt/my_mnt3 -fs /mnt/my_mnt4
Clone-Split-Status for /mnt/my_mnt3 is 14% Complete
Clone-Split-Status for /mnt/my_mnt4 is 17% Complete
```

The following example shows a clone split status query that is running:

```
# snapdrive clone split status -all
Job ID: SVE2oxKXzH:
Clone-Split-Status for /fs1-1_3 is 100% Complete
Clone-Split-Status for /fs1_0 is 100% Complete
Clone-Split-Status for /fs1_1 is 100% Complete
Clone-Split-Status for /fs1_2 is 100% Complete
Clone-Split-Status for /fs1_3 is 100% Complete
```

- When a job is removed from the job set and you query the status of a clone split status using file specification, SnapDrive for UNIX displays the error message as

  `No split is currently in progress for the given resource`

- When a job is removed from the job set and you query the status of a clone split using a job ID, SnapDrive for UNIX displays the error message as

  `Job ID is not valid`

- When all file specifications are removed from a job and you query the status of a clone split using the Job ID, SnapDrive for UNIX displays as

  `Job ID is not valid`

because the job is removed from the job set.

- If any file specifications fail due to insufficient space in the storage system, the job still continues to split for the remaining file specifications. This means the job is not deleted from the job queue and the job status is retained until you query the overall job result.

Stopping the volume clone or LUN clone split operation

You can stop the clone split for a volume clone or LUN clone using the job ID or file specification.

Steps

1. Enter the following command:

   snapdrive clone split stop [-lun] *long_lun_name* [*long_lun_name*…] |[{-dg | -vg | -fs | -hostvol | -lvol} *file_spec* [*file_spec*…]] | [-job *<jobid>*]

   SnapDrive for UNIX stops the clone split stop operation that is in progress.

The following example shows the clone split operation that is stopped by using the file specification.

```
# snapdrive clone split stop -fs /mnt/my_mnt4 /mnt/my_mnt3
Clone-Split for "/mnt/my_mnt4" is 0% Completed and Stopped.
Clone-Split for "/mnt/my_mnt3" is 0% Completed and Stopped.
```

The following example shows the clone split operation that is stopped by using the job ID.

```
# snapdrive clone split stop -job B265Dbv8gh
Clone-Split for "/mnt/my_mnt3" is 14% Completed and Stopped.
Clone-Split for "/mnt/my_mnt4" is 17% Completed and Stopped.
```

The following example is a typical output that shows the clone split stop operation for a file specification that is already stopped.

```
# snapdrive clone split stop -fs /mnt/my_mnt4 /mnt/my_mnt3
Clone-Split for "/mnt/my_mnt3" is not stopped : No split is in progress
for this resource
Clone-Split for "/mnt/my_mnt4" is not stopped : No split is in progress
for this resource
```

- If the clone split is stopped for a particular file specification in the job id and the clone split stop is successful, then file specification is removed from the job.
- If the clone split is stopped for a job, and the clone split stop is successful for all the file specification in the job, the job is removed from the job set.

Viewing the result of a clone split operation using job ID or file specification

You can view the result of the completed clone split operation using job ID or file specification.

Steps

1. Enter the following command to view the clone split result using a file specification:

   snapdrive clone split result [-lun] *long_lun_name* [*long_lun_name*…] |[{-dg | -vg | -fs | -hostvol | -lvol} file_spec [file_spec…]] | [-job <*jobid*>]

   SnapDrive for UNIX displays the result of the clone split that is completed, or failed for a file specification, and then removes the file specification from the job, and removes the job from the job queue.

The following example shows the clone split result for a job ID that is completed successfully.

```
# snapdrive clone split result -job VT1ov6Q8vU
Clone-Split for "/mnt/my_mnt3" is 100% completed and succeeded
Clone-Split for "/mnt/my_mnt4" is 100% completed and succeeded
```

If there are two file specifications, and out of which one of the file specification fails due to insufficient space in the storage system, then the result of the clone split operation shows as one file specification failed and another file specification was completed successfully.

The following example shows the clone split result for a file specification that is completed successfully.

```
# snapdrive clone split result -fs /mnt/my_mnt3 /mnt/my_mnt4
Clone-Split for "/mnt/my_mnt3" is 100% completed and succeeded
Clone-Split for "/mnt/my_mnt4" is 100% completed and succeeded
```

The following example shows the clone split result when the clone split operation is still in progress and not yet completed.

```
# snapdrive clone split result -job R57aCzUaeG
Clone-Split for "/mnt/my_mnt3" is 0% completed and Split in progress
```

The following example shows a job that is permanently removed from the job set, and when you try to view the result using file specification, SnapDrive for UNIX encounters an error message as "does not belong to any job."

```
# snapdrive clone split result -fs /mnt/my_mnt2
Storage resource /mnt/my_mnt2 does not belong to any job
```

The following example shows a job that is permanently removed from the job set, and when you try to view the result using job ID, SnapDrive for UNIX encounters an error message as "Job ID is not valid".

```
# snapdrive clone split result -job T59aCzUaeG
Job ID is not valid
```

The following example displays the clone split result in which one of the clone split is in progress and another has failed.

```
# snapdrive clone split result -job qJrG8U59mg
Clone-Split for "/mnt/my_mnt4" is 100% completed and succeeded
Clone-Split for "/mnt/my_mnt5" is 0% completed and split failed
```

Deleting a Snapshot copy

You can delete a Snapshot copy on a storage system by using the `snapdrive snap delete` command.

Command to use to delete Snapshot copies

Use the `snapdrive snap delete` command to delete a Snapshot copy.

The `snapdrive snap delete` command removes the Snapshot copies you specify from a storage system. This command does not perform any operations on the host. It only removes the Snapshot copy from a storage system, if you have permission to do so. (If you want to keep the LUNs and mappings.)

Reasons to delete Snapshot copies

Delete older Snapshot copies to free space on the storage system volume, or to keep fewer Snapshot copies.

You might delete older Snapshot copies for the following reasons:

- To keep fewer stored Snapshot copies than the hard limit of 255 on a storage system volume. Once it reaches the limit, attempts to create new Snapshot copies fail.
- To free space on the storage system volume. Even before the Snapshot copy reaches the limit, a Snapshot copy fails if the disk does not have enough reserved space for it.
- You can also use the wildcard (*) character in Snapshot copy names. The Snapshot show operation enables you to use the wildcard character to show all Snapshot copy names that match a certain pattern. The following rules apply to using wildcard in Snapshot copy names:
  - You can use a wildcard at the end of the name only. You cannot use the wildcard at the beginning or the middle of a Snapshot copy name.
  - You cannot use the wildcard in the storage system or storage system volume fields of a Snapshot copy name.

Guidelines for deleting Snapshot copies

You cannot delete a Snapshot copy that is in use, or if the Snapshot copy is across

multiple storage system volumes.

Follow these guidelines when you use the `snapdrive snap delete` command:

- The Snapshot delete operation fails if any of the Snapshot copies you want to delete are in use, or not created by SnapDrive for UNIX. You can override this behavior by including the `-force` option with the `snapdrive snap delete` command.
- If you have a Snapshot copy that spans multiple storage system volumes, you must manually delete the Snapshot copy on each volume.

Information required for using the snapdrive snap delete command

To delete a Snapshot copy, specify the name of the Snapshot copy to be deleted. Using the `snapdrive snap delete` command you can view the list of Snapshot copies that are deleted.

The following table gives the information about the `snapdrive snap delete` command.

Requirement/Option	Argument
Specify the name for the Snapshot copy. Use the long form of the Snapshot copy name where you enter the storage system name, volume, and Snapshot copy name. The following is an example of a long Snapshot copy name: `big_filer:/vol/account_vol:snap_20031115` If you want to specify additional Snapshot copies, you can use the short form of the name if they are on the same storage system and volume as the first Snapshot copy. Otherwise, use the long form of the name again.	
Snapshot copy name (`-snapname`)	*`long_Snapshot copy_name`*
Additional Snapshot copies	*`Snapshot copy_name`* (either long or short form)
`-verbose`	~
To display a list of the Snapshot copies that is deleted, include the -verbose option. This option fills in the missing storage system and volume information in case where you used the short form of the Snapshot copy name.	
`-force`	~
`-noprompt`	~
Optional: Decide if you want to overwrite an existing Snapshot copy. Without this option, this operation halts if you supply the name of an existing Snapshot copy. When you supply this option and specify the name of an existing Snapshot copy, it prompts you to confirm that you want to overwrite the Snapshot copy. To prevent SnapDrive for UNIX from displaying the prompt, include the `-noprompt` option also. (You must always include the `-force` option if you want to use the `-noprompt` option.)	

Deleting a Snapshot copy

You can use the `snapdrive snap delete` command to delete a Snapshot copy.

Steps

1. Enter the following command:

   `snapdrive snap delete [-snapname] long_snap_name [snap_name…] [-verbose] [-force [-noprompt]]`

   > If the Snapshot copy you specify is in use, this operation fails. SnapDrive for UNIX reports that this operation has completed successfully only, if all the Snapshot copies are removed.

SnapDrive for UNIX deletes the existing contents of the LUNs you specify in the `snap delete` command line and replaces them with the contents of the LUNs in the Snapshot copy you specify.

This example displays a list of what is being deleted:

```
# snapdrive snap delete -v filer1:/vol/vol1/snap1 snap2 snap3
snapdrive: deleting
filer1:/vol/vol1/snap1
filer1:/vol/vol1/snap2
filer1:/vol/vol1/snap3
```

Disconnecting a Snapshot copy

You can disconnect a Snapshot copy from a LUN, a file system on a LUN, disk groups, NFS directories, or shared disk groups, host volumes, and file systems on a LUN.

> You can disconnect the split clone volumes from SnapDrive for UNIX 4.2 and later versions.

Using the Snapshot disconnect operation

Use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that is across multiple storage systems or storage system volumes.

You use the `snapdrive snap disconnect` command to remove the mappings for LUNs, or for storage entities and the underlying LUNs, or for NFS directories in the Snapshot copy.

You can use this command to disconnect Snapshot copies that span multiple storage system volumes or multiple storage systems. The storage entities and volumes can reside on the same storage system or different storage systems.

Use this command to disconnect any of the following:

- LUNs
- A file system created directly on a LUN

- Disk groups, host volumes, and file systems created on LUNs
- NFS directory trees
- Shared disk groups, host volumes, and file systems created on LUNs

The disconnect operation does not modify the connected Snapshot copy. However, by default, the operation does delete any temporary LUNs or clones created by the corresponding connect operation.

> For LUNs, file systems on LUNs, and LVM entities, this command is equivalent to `snapdrive storage delete`.

Guidelines for disconnecting Snapshot copies

The following are the guidelines to disconnect a Snapshot copy for LUNs, storage entities, or NFS directories.

- When you disconnect a file system, SnapDrive for UNIX always removes the mountpoint.
- To undo the effects of the Snapshot connect operation, use the Snapshot disconnect command.
- If you set the *`enable-split-clone`* configuration variable value to `on` or `sync` during the Snapshot connect operation and `off` during the Snapshot disconnect operation, SnapDrive for UNIX does not delete the original volume or LUN that is present in the Snapshot copy.

Guidelines for disconnecting Snapshot copies for NFS entities

The disconnect operation for a Snapshot copy can be executed from any node in a host cluster environment. Use command options to disconnect a storage entity from a specific node.

Follow these guidelines when disconnecting Snapshot copies that contain NFS entities:

- If you disconnect an NFS directory tree that you connected with read-only permission, SnapDrive for UNIX performs the following actions:
  - Unmounts the file system.
  - Removes the mount entry in the file system table file.
  - Removes the mountpoint.
  - Does not remove the export rules of the Snapshot copy directory, which was created while connecting NFS filespec from the secondary host (the host that does not have export permission on the parent volume).
- If you disconnect an NFS directory tree that you connected with read-write permission, SnapDrive for UNIX performs the following actions:
  - Unmounts the file system.
  - Removes the mount entry in the file system table file.
  - Deletes the NFS directory tree that corresponds to the file system in the FlexVol volume clone.
  - Destroys the underlying FlexVol volume clone (if it is empty).
  - Removes the mountpoint.

Information required for using the snapdrive snap disconnect command

To disconnect a Snapshot copy, specify the type of storage entity to be used, such as, LUN, disk groups, file systems, or host volume.

The following table gives the information you need to supply when you use the `snapdrive snap disconnect` command.

Requirement/Option	Argument
LUN (`-lun` *`file_spec`*)	*name of the LUN. Include the name of the filer, volume and LUN.*
Disk group (`-dg` *`file_spec`*) or volume group (`-vg` *`file_spec`*)	*name of the disk or volume group*
File system (`-fs` *`file_spec`*)	*`filesystem_name`*
Host volume (-hostvol *`file_spec`*) or logical volume (-lvol *`file_spec`*)	*name of the host or logical volume*
Specify the type of storage entity that you want to use to disconnect the Snapshot copy and supply that entity's name with the appropriate argument. This is the value for the *`file_spec`* argument.	
`-full`	~
Include the -full option on the command line if you want SnapDrive for UNIX to disconnect the objects from the Snapshot copy even if a host-side entity on the command line has other entities (such as a disk group that has one or more host volumes). If you do not include this option, you must specify only empty host-side entities.	
`-fstype`	*`type`*
`-vmtype`	*`type`*
Optional: Specify the type of file system and volume manager to be used.	
`-split`	~
Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations.	

Disconnecting Snapshot copy with LUNs and no storage entities

You can use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that contains LUNs having no storage entities.

Steps

1. Enter the following command syntax:

   snapdrive snap disconnect -lun *long_lun_name* [*lun_name*…]

SnapDrive for UNIX removes the mappings for the storage entities specified in the command line.

The following command removes the mappings to luna and lunb on the storage system toaster:

```
# snapdrive snap disconnect -lun toaster:/vol/vol1/luna lunb
```

Disconnecting Snapshot copy with storage entities

You can use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that contains storage entities.

Steps

1. Enter the following command:

   snapdrive snap disconnect {-dg | -fs | -hostvol} *file_spec* [*file_spec*…]{-dg | -fs | -hostvol} file_spec [file_spec…]] [-full] [-fstype type] [-vmtype type] [-split]

   This command must always start with the storage entity, for example, `-lun`, `-dg`, `-hostvol`, or `-fs`.

   - If you specify a LUN (`-lun`), you must enter the long LUN name. You cannot specify a LUN with the `-lun` option on the same command line as other storage entities (`-vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol` options).
   - If you specify an NFS mountpoint, you cannot specify non-NFS entities (`-vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol`) on the same command line. You must use a separate command to specify the NFS mountpoint.

> An error message occurs if the host entity is using LUNs that are not part of the Snapshot copy. An error also occurs if you specify a subset of the host volumes and/or file systems contained in each target disk group.

SnapDrive for UNIX removes the mappings for the storage entities specified in the command line.

This command line removes the mappings to all the LUNs underlying the host volume dg5/myvolume. It removes any temporary LUNs that creates with a Snapshot connect operation:

```
# snapdrive snap disconnect -hostvol dg5/myvolume
```

This command line removes the mappings to all the LUNs underlying the host volume dg5/myvolume. It removes any temporary LUNs that creates with a Snapshot connect operation:

```
# snapdrive snap disconnect -hostvol dg5/myvolume
```

This command disconnects the mapping to disk group 1 (dg1) and to the underlying LUN. It also removes any temporary LUNs that creates with the Snapshot connect operation:

```
# snapdrive snap disconnect -lun toaster:/vol/vol1/luna -dg dg1
```

This command line removes the mapping to the file system fs1, and to the LUN that underlies it. It also removes any temporary LUNs that creates with the Snapshot connect operation:

```
# snapdrive snap disconnect -fs mnt/fs1
```

This command line removes the mappings for disk groups dg1, dg2, and dg3. It removes any temporary LUNs that creates with the Snapshot connect operation:

```
# snapdrive snap disconnect -dg dg1 dg2 dg3
```

This example disconnects a Snapshot copy with file system, disk group on Veritas stack:

```
# snapdrive snap disconnect -fs /mnt/fs1_clone -fstype vxfs
delete file system /mnt/fs1_clone
- fs /mnt/fs1_clone ... deleted
- hostvol vxvm1_0/fs1_SdHv_0 ... deleted
- dg vxvm1_0 ... deleted
- LUN snoopy:/vol/vol1/lunVxvm1_0 ... deleted
```

This example disconnects a Snapshot copy with file system, disk group on LVM stack:

```
# snapdrive snap disconnect -fs /mnt/fs1_clone -fstype jfs2

delete file system /mnt/fs1_clone
- fs /mnt/fs1_clone ... deleted
- hostvol lvm1_0/fs1_SdHv_0 ... deleted
- dg lvm1_0 ... deleted
- LUN snoopy:/vol/vol1/lunLvm1_0 ... deleted
```

Disconnecting Snapshot copies with shared storage entities

You can use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that contains shared storage entities.

Steps

1. Enter the following command syntax:

   snapdrive snap disconnect {-dg | -fs} *file_spec* [*file_spec*…] {-dg | -fs} *file_spec* [*file_spec*…]…] *long_snap_name* [-full] [-fstype *type*] [-vmtype *type*] [-split]

This example disconnects shared file system:

```
# snapdrive snap disconnect -fs /mnt/oracle
```