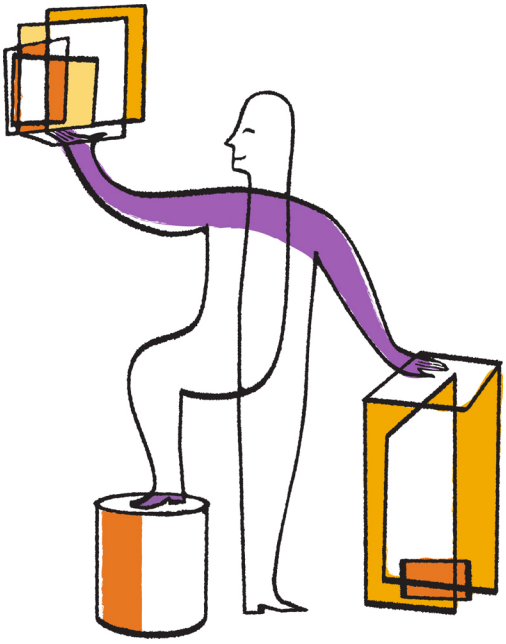
Updated for 8.2.1

Clustered Data ONTAP® 8.2

Physical Storage Management Guide



NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089
U.S.

Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 463-8277
Web: www.netapp.com
Feedback: doccomments@netapp.com

Part number: 215-08511_B0
February 2014

Contents

Managing disks using Data ONTAP

Disks (sometimes also called *drives*) provide the basic unit of storage for storage systems running Data ONTAP that use native storage shelves. Understanding how Data ONTAP uses and classifies disks will help you manage your storage more effectively.

How Data ONTAP reports disk types

Data ONTAP associates a type with every disk. Data ONTAP reports some disk types differently than the industry standards; you should understand how Data ONTAP disk types map to industry standards to avoid confusion.

When Data ONTAP documentation refers to a disk type, it is the type used by Data ONTAP unless otherwise specified. *RAID disk types* denote the role a specific disk plays for RAID. RAID disk types are not related to Data ONTAP disk types.

For a specific configuration, the disk types supported depend on the storage system model, the shelf type, and the I/O modules installed in the system. For more information about the types of disks supported by your configuration, see the *Hardware Universe* at *hwu.netapp.com*.

The following tables show how Data ONTAP disk types map to industry standard disk types for the SAS and FC storage connection types, storage arrays, and for virtual storage (Data ONTAP-v):

Table 1: SAS-connected storage

Data ONTAP disk type	Disk class	Industry standard disk type	Description
BSAS	Capacity	SATA	Bridged SAS-SATA disks with added hardware to enable them to be plugged into a SAS-connected storage shelf.
FSAS	Capacity	NL-SAS	Near Line SAS
MSATA	Capacity	SATA	SATA disk in multi-disk carrier storage shelf
SAS	Performance	SAS	Serial-Attached SCSI
SSD	Ultra-performance	SSD	Solid-state drives

Table 2: FC-connected storage

Data ONTAP disk type	Disk class	Industry standard disk type	Description
ATA	Capacity	SATA	
FCAL	Performance	FC	

Table 3: Storage arrays

Data ONTAP disk type	Disk class	Industry standard disk type	Description
LUN	N/A	LUN	A logical storage device backed by storage arrays and used by Data ONTAP as a disk. These LUNs are referred to as *array LUNs* to distinguish them from the LUNs that Data ONTAP serves to clients.

Table 4: Virtual storage (Data ONTAP-v)

Data ONTAP disk type	Disk class	Industry standard disk type	Description
SAS	N/A	VMDK	Virtual disks that are formatted and managed by VMware ESX.

For information about best practices for working with different types of disks, see *Technical Report 3437: Storage Subsystem Resiliency Guide*.

Related concepts

Rules for mixing HDD types in aggregates on page 110
Storage connection types and topologies supported by Data ONTAP on page 11

Related references

Understanding RAID disk types on page 16

Related information

TR 3437: Storage Subsystem Resiliency Guide

Storage connection types and topologies supported by Data ONTAP

Data ONTAP supports two storage connection types: Serial-Attached SCSI (SAS) and Fibre Channel (FC). The FC connection type supports three topologies: arbitrated loop, switched, and point-to-point.

- SAS, BSAS, FSAS, SSD, and MSATA disks use the SAS connection type.
  SAS-connected storage shelves are connected to the controller on a daisy chain called a *stack*.
- FC and ATA disks use the FC connection type with an arbitrated-loop topology (FC-AL).
  FC-connected storage shelves are connected to the controller on a loop.
- Array LUNs use the FC connection type, with either the point-to-point or switched topology.

You cannot combine different connection types in the same loop or stack.

How disks can be combined for the SAS storage connection type

You can combine SAS-connected storage shelves containing performance disks and SAS-connected storage shelves containing capacity disks within the same stack, although this configuration is not recommended.

Each SAS-connected storage shelf can contain only one class of disk (capacity, performance, or SSDs). The only exception to this rule is if the shelf is being used for a Flash Pool aggregate. In that case, for some SSD sizes and shelf models, you can combine SSDs and HDDs in the same shelf. For more information, see the *Hardware Universe*.

How disks can be combined for the FC-AL storage connection type

You cannot combine storage shelves containing FC disks and storage shelves containing ATA disks in the same loop.

Methods of calculating aggregate and system capacity

You use the physical and usable capacity of the drives you employ in your storage systems to ensure that your storage architecture conforms to the overall system capacity limits and the size limits of your aggregates.

To maintain compatibility across different brands of drives, Data ONTAP rounds down (*right-sizes*) the amount of space available for user data. In addition, the numerical base used to calculate capacity (base 2 or base 10) also impacts sizing information. For these reasons, it is important to use the correct size measurement, depending on the task you want to accomplish:

- For calculating overall system capacity, you use the physical capacity of the drive, and count every drive that is owned by the storage system.
- For calculating how many drives you can put into an aggregate before you exceed its maximum size, you use the right-sized, or usable, capacity of all data drives in that aggregate. Parity, dparity, and cache drives are not counted against the maximum aggregate size.

To see the physical and usable capacity for a specific drive, see the *Hardware Universe* at *hwu.netapp.com*.

Disk speeds supported by Data ONTAP

For hard disk drives, which use rotating media, speed is measured in revolutions per minute (RPM). Faster drives provide more input/output operations per second (IOPS) and faster response time.

It is best to use disks of the same speed in an aggregate.

Data ONTAP supports the following rotational speeds for hard disk drives:

- Performance disks (SAS-connected)
  - 10K RPM
  - 15K RPM
- Capacity disks (SAS-connected)
  - 7.2K RPM
- Performance disks (FC-connected)
  - 10K RPM
  - 15K RPM
- Capacity disks (FC-connected)
  - 7.2K RPM

Solid-state drives, or SSDs, are flash media-based devices and therefore the concept of rotational speed does not apply to them.

For more information about which disks are supported with specific hardware configurations, see the *Hardware Universe* at *hwu.netapp.com*.

How drive checksum types affect aggregate and spare management

There are two checksum types available for drives used by Data ONTAP: BCS (block) and AZCS (zoned). Understanding how the checksum types differ and how they impact storage management enables you to manage your storage more effectively.

Both checksum types provide the same resiliency capabilities. BCS optimizes for data access speed, and reserves the smallest amount of capacity for the checksum for drives with 520-byte sectors.

AZCS provides enhanced storage utilization and capacity for drives with 512-byte sectors. You cannot change the checksum type of a drive.

To determine the checksum type of a specific drive model, see the *Hardware Universe*.

Aggregates have a checksum type, which is determined by the checksum type of the drives or array LUNs that compose the aggregate. The following configuration rules apply to aggregates, drives, and checksums:

- Checksum types cannot be combined within RAID groups.
  This means that you must consider checksum type when you provide hot spare drives.
- When you add storage to an aggregate, if it has a different checksum type than the storage in the RAID group to which it would normally be added, Data ONTAP creates a new RAID group.
- An aggregate can have RAID groups of both checksum types.
  These aggregates have a checksum type of `mixed`.
- Drives of a different checksum type cannot be used to replace a failed drive.
- You cannot change the checksum type of a drive.

Drive name formats

Each drive has a name that differentiates it from all other drives. Drive names have different formats, depending on the connection type (FC or SAS) and how the drive is attached.

Each drive has a universal unique identifier (UUID) that differentiates it from all other drives in the cluster.

The names of unowned drives (broken or unassigned drives) display the alphabetically lowest node name in the cluster that can see that drive.

The following table shows the various formats for drive names, depending on how they are connected to the storage system.

Note: For internal drives, the slot number is zero, and the internal port number depends on the system model.

Drive connection	Drive name	Example
SAS, direct-attached	<node>:<slot><port>.<shelfID>.<bay>	The drive in shelf 2, bay 11, connected to onboard port 0a and owned by node1 is named node1:0a.2.11. The drive in shelf 6, bay 3, connected to an HBA in slot 1, port c, and owned by node1 is named node1:1c.6.3.
SAS, direct-attached in multi-disk carrier disk shelf	<node>:<slot><port>.<shelfID>.<bay>L<carrierPosition>	Carrier position is 1 or 2.
SAS, direct-attached, for systems running Data ONTAP-v	<slot><port>.<ID>	The third virtual disk connected to the first port is named 0b.3. The second virtual disk connected to the third port is named 0d.2. The range of ports is b through e, and the range of disks is 0 through 15.

Drive connection	Drive name	Example
FC, direct-attached	<node>:<slot><port>.<loopID>	The drive with loop ID 19 (bay 3 of shelf 1) connected to onboard port 0a and owned by node1 is named node1:0a.19. The drive with loop ID 34 connected to an HBA in slot 8, port c and owned by node1 is named node1:8c.34.
FC, switch-attached	<node>:<switch_name>.<switch_port>.<loopID>	The drive with loop ID 51 connected to port 3 of switch SW7 owned by node1 is named node1:SW7.3.51.

You can use the `storage disk show` command to see the drive name for a particular drive using the initiator, shelf, and bay.

Related concepts

Array LUN name format on page 61

Loop IDs for FC-AL connected disks

For disks connected using Fibre Channel-Arbitrated Loop (FC-AL or FC), the loop ID is an integer between 16 and 126. The loop ID identifies the disk within its loop, and is included in the disk name, which identifies the disk uniquely for the entire system.

The loop ID corresponds to the storage shelf number and the bay in which the disk is installed. The lowest loop ID is always in the far right bay of the first storage shelf. The next higher loop ID is in the next bay to the left, and so on. You can view the device map for your storage shelves by using the `fcadmin device_map` command, available through the nodeshell.

For more information about the loop ID map for your storage shelf, see the hardware guide for the storage shelf.

Understanding RAID disk types

Data ONTAP classifies disks as one of four types for RAID: data, hot spare, parity, or dParity. The RAID disk type is determined by how RAID is using a disk; it is different from the Data ONTAP disk type.

Data disk	Holds data stored on behalf of clients within RAID groups (and any data generated about the state of the storage system as a result of a malfunction).
Spare disk	Does not hold usable data, but is available to be added to a RAID group in an aggregate. Any functioning disk that is not assigned to an aggregate but is assigned to a system functions as a hot spare disk.
Parity disk	Stores row parity information that is used for data reconstruction when a single disk drive fails within the RAID group.
dParity disk	Stores diagonal parity information that is used for data reconstruction when two disk drives fail within the RAID group, if RAID-DP is enabled.

Related concepts

How Data ONTAP reports disk types on page 9

How disk sanitization works

Disk sanitization is the process of physically obliterating data by overwriting disks or SSDs with specified byte patterns or random data so that recovery of the original data becomes impossible. You use the sanitization process to ensure that no one can recover the data on the disks. This functionality is available through the nodeshell.

Related tasks

Using disk sanitization to remove data from disks on page 36

Disk sanitization process

Understanding the basics of the disk sanitization process helps you understand what to anticipate during the sanitization process and after it is complete.

The disk sanitization process uses three successive default or user-specified byte overwrite patterns for up to seven cycles per operation. The random overwrite pattern is repeated for each cycle.

Depending on the disk capacity, the patterns, and the number of cycles, the process can take several hours. Sanitization runs in the background. You can start, stop, and display the status of the sanitization process.

The sanitization process contains two phases:

1. Formatting phase
   The operation performed for the formatting phase depends on the class of disk being sanitized, as shown in the following table:

Disk class	Formatting phase
Capacity HDDs	Skipped
Performance HDDs	SCSI format operation
SSDs	SCSI sanitize operation

2. Pattern overwrite phase
   The specified overwrite patterns are repeated for the specified number of cycles.

When the sanitization process is complete, the specified disks are in a sanitized state. They are not returned to spare status automatically. You must return the sanitized disks to the spare pool before the newly sanitized disks are available to be added to another aggregate.

When disk sanitization cannot be performed

Disk sanitization is not supported for all disk types. In addition, there are times when disk sanitization cannot be performed.

You should be aware of the following facts about the disk sanitization process:

- It is not supported on all SSD part numbers.
  For information about which SSD part numbers support disk sanitization, see the *Hardware Universe* at *hwu.netapp.com*.
- It is not supported in takeover mode for systems in an HA pair.
- It cannot be performed on disks that were failed due to readability or writability problems.
- It does not perform its formatting phase on ATA drives.
- If you are using the random pattern, it cannot be performed on more than 100 disks at one time.
- It is not supported on array LUNs.
- If you sanitize both SES disks in the same ESH shelf at the same time, you see errors on the console about access to that shelf, and shelf warnings are not reported for the duration of the sanitization.
  However, data access to that shelf is not interrupted.
- You can perform disk sanitization on disks using Storage Encryption.
  However, there are other methods to obliterate data on disks using Storage Encryption that are faster and do not require an operational storage system.

What happens if disk sanitization is interrupted

Disk sanitization is a long-running operation. If disk sanitization is interrupted by user intervention or an unexpected event such as a power outage, Data ONTAP takes action to return the disks that were

being sanitized to a known state, but you must also take action before the sanitization process can finish.

If the sanitization process is interrupted by power failure, system panic, or manual intervention, the sanitization process must be repeated from the beginning. The disk is not designated as sanitized.

If the formatting phase of disk sanitization is interrupted, Data ONTAP must recover any disks that were corrupted by the interruption. After a system reboot and once every hour, Data ONTAP checks for any sanitization target disk that did not complete the formatting phase of its sanitization. If any such disks are found, Data ONTAP recovers them. The recovery method depends on the type of the disk. After a disk is recovered, you can rerun the sanitization process on that disk; for HDDs, you can use the `-s` option to specify that the formatting phase is not repeated again.

Tips for creating and backing up aggregates containing data to be sanitized

If you are creating or backing up aggregates to contain data that might need to be sanitized, following some simple guidelines will reduce the time it takes to sanitize your data.

- Make sure your aggregates containing sensitive data are not larger than they need to be.
  If they are larger than needed, sanitization requires more time, disk space, and bandwidth.
- When you back up aggregates containing sensitive data, avoid backing them up to aggregates that also contain large amounts of nonsensitive data.
  This reduces the resources required to move nonsensitive data before sanitizing sensitive data.

How Data ONTAP monitors disk performance and health

Data ONTAP continually monitors disks to assess their performance and health. When Data ONTAP encounters certain errors or behaviors from a disk, it takes the disk offline temporarily or takes the disk out of service to run further tests.

What happens when Data ONTAP takes disks offline

Data ONTAP temporarily stops I/O activity to a disk and takes a disk offline when Data ONTAP is updating disk firmware in background mode or when disks become non-responsive. While the disk is offline, Data ONTAP performs a quick check on it to reduce the likelihood of forced disk failures.

A disk can be taken offline only if its containing RAID group is in a normal state and the plex or aggregate is not offline.

While the disk is offline, Data ONTAP reads from other disks within the RAID group while writes are logged. When the offline disk is ready to come back online, Data ONTAP resynchronizes the RAID group and brings the disk online. This process generally takes a few minutes and incurs a negligible performance impact.

How Data ONTAP reduces disk failures using Rapid RAID Recovery

When Data ONTAP determines that a disk has exceeded its error thresholds, Data ONTAP can perform Rapid RAID Recovery by removing the disk from its RAID group for testing and, if necessary, failing the disk. Spotting disk errors quickly helps prevent multiple disk failures and allows problem disks to be replaced.

By performing the Rapid RAID Recovery process on a suspect disk, Data ONTAP avoids three problems that occur during sudden disk failure and the subsequent RAID reconstruction process:

- Rebuild time
- Performance degradation
- Potential data loss due to additional disk failure during reconstruction

During Rapid RAID Recovery, Data ONTAP performs the following tasks:

1. Places the suspect disk in pre-fail mode.
2. Selects a hot spare replacement disk.

   Note: If no appropriate hot spare is available, the suspect disk remains in pre-fail mode and data continues to be served. However, a suspect disk performs less efficiently. Impact on performance ranges from negligible to worse than degraded mode. For this reason, hot spares should always be available.

3. Copies the suspect disk's contents to the spare disk on the storage system before an actual failure occurs.
4. After the copy is complete, attempts to put the suspect disk into the maintenance center, or else fails the disk.

Note: Tasks 2 through 4 can occur only when the RAID group is in normal (not degraded) mode.

If the suspect disk fails on its own before copying to a hot spare is complete, Data ONTAP starts the normal RAID reconstruction process.

A message is sent to the log file when the Rapid RAID Recovery process is started and when it is complete. The messages are tagged `raid.rg.diskcopy.start:notice` and `raid.rg.diskcopy.done:notice`.

Related concepts

About degraded mode on page 95
When Data ONTAP can put a disk into the maintenance center on page 20
How Data ONTAP works with hot spare disks on page 93

How the maintenance center helps prevent drive errors

Data ONTAP provides a mechanism to test drives called the maintenance center. Sometimes Data ONTAP puts drives into the maintenance center automatically; you can also put a suspect drive into

the maintenance center manually. Knowing how the maintenance center works helps you manage your storage effectively.

When a disk is in the maintenance center, it is subjected to a number of tests. If the disk passes all of the tests, it is redesignated as a spare. Otherwise, Data ONTAP fails the disk.

The maintenance center is controlled by the `disk.maint_center.enable` option. It is on by default.

Data ONTAP puts disks into the maintenance center only if there are two or more spares available for that disk.

You can control the number of times a disk is allowed to go to the maintenance center by using the `disk.maint_center.allowed_entries` option. The default value for this option is `1`, which means that if the disk is ever sent back to the maintenance center, it is automatically failed.

You can also put a disk into the maintenance center manually by using the `disk maint start` command. If the target disk is in use, it does not enter the maintenance center until its contents have been copied to another disk (unless you include the `-i` option).

Data ONTAP informs you of these activities by sending messages to the following destinations:

- The console
- A log file at `/etc/log/maintenance.log`

When Data ONTAP puts a disk into the maintenance center and that disk is housed in a storage shelf that supports automatic power cycling, power to that disk might be turned off for a short period of time. If the disk returns to a ready state after the power cycle, the maintenance center tests the disk. Otherwise, the maintenance center fails the disk immediately.

You can see the power-cycle status for ESH4 storage shelves by using the `environment shelf_power_status` command.

You can access the options and commands to control the maintenance center by using the nodeshell. For more information about the nodeshell, see the man page for the `system node run` command.

For information about best practices for working with the maintenance center, see *Technical Report 3437: Storage Best Practices and Resiliency Guide*.

Related information

TR 3437: Storage Best Practices and Resiliency Guide

When Data ONTAP can put a disk into the maintenance center

When Data ONTAP detects certain disk errors, it tries to put the disk into the maintenance center for testing. Certain requirements must be met for the disk to be put into the maintenance center.

If a disk experiences more errors than are allowed for that disk type, Data ONTAP takes one of the following actions:

- If the `disk.maint_center.spares_check` option is set to `on` (the default) and two or more spares are available (four for multi-disk carriers), Data ONTAP takes the disk out of service and assigns it to the maintenance center for data management operations and further testing.
- If the `disk.maint_center.spares_check` option is set to `on` and fewer than two spares are available (four for multi-disk carriers), Data ONTAP does not assign the disk to the maintenance center.

  It fails the disk and designates the disk as a broken disk.
- If the `disk.maint_center.spares_check` option is set to `off`, Data ONTAP assigns the disk to the maintenance center without checking the number of available spares.

Note: The `disk.maint_center.spares_check` option has no effect on putting disks into the maintenance center from the command-line interface.

Data ONTAP does not put SSDs into the maintenance center.

How Data ONTAP uses continuous media scrubbing to prevent media errors

The purpose of the continuous media scrub is to detect and correct media errors to minimize the chance of storage system disruption due to a media error while a storage system is in degraded or reconstruction mode.

By default, Data ONTAP runs continuous background media scrubbing for media errors on all storage system disks. If a media error is found, Data ONTAP uses RAID to reconstruct the data and repairs the error.

Media scrubbing is a continuous background process. Therefore, you might observe disk LEDs blinking on an apparently idle storage system. You might also observe some CPU activity even when no user workload is present.

How continuous media scrubbing impacts system performance

Because continuous media scrubbing searches only for media errors, its impact on system performance is negligible. In addition, the media scrub attempts to exploit idle disk bandwidth and free CPU cycles to make faster progress. However, any client workload results in aggressive throttling of the media scrub resource.

If needed, you can further decrease the CPU resources consumed by a continuous media scrub under a heavy client workload by increasing the maximum time allowed for a media scrub cycle to complete. You can do this by using the `raid.media_scrub.rate` option.

Why continuous media scrubbing should not replace scheduled RAID-level disk scrubs

Because the continuous media scrub process scrubs only media errors, you should continue to run the storage system's scheduled complete RAID-level scrub operation. The RAID-level scrub finds and corrects parity and checksum errors as well as media errors.

Related concepts

How you schedule automatic RAID-level scrubs on page 97

Increasing storage availability by using ACP

ACP, or Alternate Control Path, is a protocol that enables Data ONTAP to manage and control a SAS-connected storage shelf subsystem. It uses a separate network (alternate path) from the data path, so management communication is not dependent on the data path being intact and available.

You do not need to actively manage the SAS-connected storage shelf subsystem. Data ONTAP automatically monitors and manages the subsystem without operator intervention. However, you must provide the required physical connectivity and configuration parameters to enable the ACP functionality.

Note: You can install SAS-connected storage shelves without configuring ACP. However, for maximum storage availability and stability, you should always have ACP configured and enabled.

After you enable ACP, you can use the `storage show acp` and `acpadmin list_all` commands, available through the nodeshell, to display information about your ACP subsystem.

Because ACP communication is on a separate network, it does not affect data access in any way.

Enabling ACP

ACP can increase your storage availability when you use SAS-connected storage shelves. If your storage system model has a dedicated port for ACP, then ACP is enabled by default and you do not need to explicitly enable ACP.

Before you begin

- The ACP subnet must be cabled on an isolated network, with no switches or hubs.
  For more information, see the *Installation and Service Guide* for your storage shelf.
- If you are configuring ACP for storage shelves attached to an HA pair, you must have recorded the domain name and network mask to ensure that they are the same for both nodes.

About this task

The ACP subnet is a private Ethernet network that enables the ACP processor in the SAS module to communicate both with Data ONTAP and the SAS IOMs in the storage shelves.

The ACP subnet is separate from the I/O data path that connects the storage shelves to the HBA on the storage controller. When you configure ACP on one of the system's network interfaces, you must supply a private domain name that conforms to the standard for private internet addresses (RFC1918). You can use the system default domain or another network name (that is, an IP address ending in 0) that conforms to the standard.

If your system has a dedicated port for ACP (e0P), you must use it. If you previously configured ACP to use a different port, when you run the `acpadmin configure` command, ACP is updated to use the dedicated port.

Some of the commands used in this procedure are available only through the nodeshell.

Steps

1. If your system does not have a dedicated port for ACP (e0P), ensure that the port you are assigning to ACP has no LIFs homed or hosted on it by completing the following steps:
   a) Determine whether a LIF is currently hosted on the target port:

      network interface show -curr-node *node* -curr-port *port*
   b) If any LIFs are hosted on the target port, migrate them away:

      network interface migrate -vserver *vserver_name* -lif *lif* -dest-node *dest_node* -dest-port *dest_port*
   c) Determine whether a LIF is currently homed on the target port:

      network interface show -home-node *node* -home-port *port*
   d) If any LIFs are homed on the target port, modify their home port to another port:

      network interface modify -vserver *vserver_name* -lif *lif* -home-port *new_home_port*
   e) Determine whether any failover groups are configured to use the target port:

      network interface failover-group show -node *node* -port *port*
   f) If the port is used by a failover group, delete the port from the failover group:

      network interface failover-group delete -failover-group *failover_group* -node *node* -port *port*
   g) Determine whether any failover rules are configured to use the target port:

      network interface failover show
   h) If the port is used in a failover rule, delete the port from the failover rule:

      network interface failover delete -server *vserver_name* -lif *lif* -priority *priority_number*
2. At the Data ONTAP command line, enter the following command:

   acpadmin configure

   If you have not previously configured the networking information for ACP, you are prompted for that information. When you select a domain name and network mask for the ACP interface, Data ONTAP automatically assigns IP addresses for the ACP interface on the storage controller and both I/O modules on each storage shelf on the ACP subnet.

3. If you configured ACP to use a non-dedicated port, complete the following steps:
   a) Reboot the node.
   b) Enter advanced privilege mode:

      set advanced

   c) Delete the port you configured for use by ACP from the resource database:

      network port delete -node *node* -port *port*

   d) Return to administrative privilege mode:

      set admin

4. Verify your ACP connectivity by entering the following command:

   storage show acp

   The ACP Connectivity Status should show “Full Connectivity”.

Example

For example, with a dedicated ACP port, 192.168.0.0 as the ACP domain, and 255.255.252.0 as the network mask for the ACP subnet, the `storage show acp` command output looks similar to the following:

```
my-sys-1> storage show acp

 Alternate Control Path:            Enabled
 Ethernet Interface:                e0p
 ACP Status:                        Active
 ACP IP address:                    192.168.2.61
 ACP domain:                        192.168.0.0
 ACP netmask:                       255.255.252.0
 ACP Connectivity Status:           Full Connectivity
 ACP Partner Connectivity Status:   Full Connectivity

Shelf Module      Reset Cnt    IP address      FW Version    Module Type   Status
------------------------------------------------------------------------------
7a.001.A          002          192.168.0.145    01.05         IOM6          active
7a.001.B          003          192.168.0.146    01.05         IOM6          active
7c.002.A          000          192.168.0.206    01.05         IOM6          active
7c.002.B          001          192.168.0.204    01.05         IOM6          active
```

How you use SSDs to increase storage performance

Solid-state drives (SSDs) are flash media-based storage devices that provide better overall performance than hard disk drives (HDDs), which are mechanical devices using rotating media. You should understand how Data ONTAP manages SSDs and the capability differences between SSDs and HDDs.

Depending on your storage system model, you can use SSDs in two ways:

- You can create Flash Pool aggregates—aggregates composed mostly of HDDs, but with some SSDs that function as a high-performance cache for your working data set.

- You can create aggregates composed entirely of SSDs, where the SSDs function as the persistent storage for all data in the aggregate.

You manage Flash Pool aggregates and aggregates composed entirely of SSDs the same way you manage aggregates composed entirely of HDDs. However, there are some differences in the way you manage SSDs from the way you manage disks. In addition, some Data ONTAP capabilities are not available on SSDs and Flash Pool aggregates.

SSDs are not supported on all storage system models. For information about which models support SSDs, see the *Hardware Universe* at *hwu.netapp.com*.

Related concepts

How Flash Pool aggregates work on page 105

How Data ONTAP manages SSD wear life

Solid-state disks (SSDs) have a different end-of-life behavior than rotating media (hard disk drives, or HDDs). Data ONTAP monitors and manages SSDs to maximize storage performance and availability.

In the absence of a mechanical failure, rotating media can serve data almost indefinitely. This is not true for SSDs, which can accept only a finite (though very large) number of write operations. SSDs provide a set of internal spare capacity, called *spare blocks*, that can be used to replace blocks that have reached their write operation limit. After all of the spare blocks have been used, the next block that reaches its limit causes the disk to fail.

Because a drive failure is an undesirable occurrence, Data ONTAP replaces SSDs before they reach their limit. When a predetermined percentage of the spare blocks have been used (approximately 90%), Data ONTAP performs the following actions:

1. Sends an AutoSupport message.
2. If a spare SSD is available, starts a disk copy to that spare.
3. If no spare is available, starts a periodic check for a spare so that the disk copy can be started when a spare becomes available.
4. When the disk copy finishes, fails the disk.

Note: You do not need to replace SSDs before they are failed by Data ONTAP. However, when you use SSDs in your storage system (as for all disk types), it is important to ensure that you have sufficient hot spares available at all times.

Capability differences between SSDs and HDDs

Usually, you manage SSDs the same as HDDs, including firmware updates, scrubs, and zeroing. However, some Data ONTAP capabilities do not make sense for SSDs, and SSDs are not supported on all hardware models.

SSDs cannot be combined with HDDs within the same RAID group. When you replace an SSD in an aggregate, you must replace it with another SSD. Similarly, when you physically replace an SSD within a shelf, you must replace it with another SSD.

The following capabilities of Data ONTAP are not available for SSDs:

- Disk sanitization is not supported for all SSD part numbers.
  For information about which SSD part numbers support sanitization, see the *Hardware Universe*.
- The maintenance center
- FlexShare

SSDs are not supported on all storage system models. For information about which models support SSDs, see the *Hardware Universe* at *hwu.netapp.com*.

Guidelines and requirements for using multi-disk carrier storage shelves

Data ONTAP automatically handles most of the extra steps required to manage disks in multi-disk carriers. However, there are some extra management and configuration requirements that you must understand before incorporating multi-disk carrier disk shelves in your storage architecture.

When using storage from multi-disk carrier disk shelves such as the DS4486, you must familiarize yourself with the guidelines and requirements governing the following topics:

- The process that Data ONTAP uses to avoid impacting any RAID groups when a multi-disk carrier needs to be removed
- When it is safe to remove a multi-disk carrier after a disk failure
- The minimum required number of spares for multi-disk carrier disks
- Multi-disk carrier disk shelf configuration
- Aggregate configuration requirements when using multi-disk carrier disk shelves
- Guidelines and best practices for using disks from a multi-disk carrier disk shelf in an aggregate

How Data ONTAP avoids RAID impact when a multi-disk carrier must be removed

Data ONTAP takes extra steps to ensure that both disks in a carrier can be replaced without impacting any RAID group. Understanding this process helps you know what to expect when a disk from a multi-disk carrier storage shelf fails.

A multi-disk carrier storage shelf, such as the DS4486, has double the storage density of other SAS-connected storage shelves. It accomplishes this by housing two disks per disk carrier. When two disks share the same disk carrier, they must be removed and inserted together. This means that when one of the disks in a carrier needs to be replaced, the other disk in the carrier must also be replaced, even if it was not experiencing any issues.

Removing two data or parity disks from an aggregate at the same time is undesirable, because it could leave two RAID groups degraded, or one RAID group double-degraded. To avoid this situation, Data ONTAP initiates a storage evacuation operation for the carrier mate of the failed disk, as well as the usual reconstruction to replace the failed disk. The disk evacuation operation copies the contents of the carrier mate to a disk in a different carrier so that the data on that disk remains available when you remove the carrier. During the evacuation operation, the status for the disk being evacuated is shown as `evacuating`.

In addition, Data ONTAP tries to create an optimal layout that avoids having two carrier mates in the same RAID group. Depending on how the other disks are laid out, achieving the optimal layout can require as many as three consecutive disk evacuation operations. Depending on the size of the disks and the storage system load, each storage evacuation operation could take several hours, so the entire swapping process could take an entire day or more.

If insufficient spares are available to support the swapping operation, Data ONTAP issues a warning and waits to perform the swap until you provide enough spares.

How to determine when it is safe to remove a multi-disk carrier

Removing a multi-disk carrier before it is safe to do so can result in one or more RAID groups becoming degraded, or possibly even a storage disruption. Data ONTAP provides several indications of when it is safe to remove a multi-disk carrier.

When a multi-disk carrier needs to be replaced, the following events must have occurred before you can remove the carrier safely:

- An AutoSupport message must have been logged indicating that the carrier is ready to be removed.
- An EMS message must have been logged indicating that the carrier is ready to be removed.
- Both disks in the carrier must be displayed in the list of broken disks.

  You can see the list of broken disks by using the `storage disk show -broken` command.

  The disk that was evacuated to allow the carrier to be removed shows the outage reason of `evacuated`.
- The amber LED on the carrier must be lit continuously.

- The green LED on the carrier must show no activity.

  Attention: You cannot reuse the carrier mate of a failed disk. When you remove a multi-disk carrier that contains a failed disk, you must replace and return the entire carrier.

Spare requirements for multi-disk carrier disks

Maintaining the proper number of spares for disks in multi-disk carriers is critical for optimizing storage redundancy and minimizing the amount of time Data ONTAP must spend copying disks to achieve an optimal disk layout.

You must maintain a minimum of two hot spares for multi-disk carrier disks at all times. To support the use of the Maintenance Center, and to avoid issues caused by multiple concurrent disk failures, you should maintain at least four hot spares for steady state operation, and replace failed disks promptly.

If two disks fail at the same time with only two available hot spares, Data ONTAP might not be able to swap the contents of both the failed disk and its carrier mate to the spare disks. This scenario is called a *stalemate*. If this happens, you are notified through EMS messages and AutoSupport messages. When the replacement carriers become available, you must follow the instructions provided by the EMS messages or contact technical support to recover from the stalemate.

Shelf configuration requirements for multi-disk carrier storage shelves

You can combine multi-disk carrier disk shelves with single-disk carrier disk shelves (standard disk shelves) on the same storage system and within in the same stack.

Aggregate requirements for disks in multi-disk carrier storage shelves

Aggregates composed of disks in multi-disk carrier disk shelves must conform to some configuration requirements.

The following configuration requirements apply to aggregates composed of disks in multi-disk carrier disk shelves:

- The RAID type must be RAID-DP.
- The format must be 64-bit.
- All HDDs in the aggregate must be the same Data ONTAP disk type.
  The aggregate can be a Flash Pool aggregate.

Related concepts

How Flash Pool aggregates work on page 105

Considerations for using disks from a multi-disk carrier storage shelf in an aggregate

Observing the requirements and best practices for using disks from a multi-disk carrier disk shelf in an aggregate enables you to maximize storage redundancy and minimize the impact of disk failures.

Disks in multi-disk carriers always have the Data ONTAP disk type of MSATA. MSATA disks cannot be mixed with HDDs from a single-carrier disk shelf in the same aggregate.

The following disk layout requirements apply when you are creating or increasing the size of an aggregate composed of MSATA disks:

- Data ONTAP prevents you from putting two disks in the same carrier into the same RAID group.
- Do not put two disks in the same carrier into different pools, even if the shelf is supplying disks to both pools.
- Do not assign disks in the same carrier to different nodes.
- For the best layout, do not name specific disks; allow Data ONTAP to select the disks to be used or added.

  If the operation cannot result in an optimal layout due to the placement of the disks and available spares, Data ONTAP automatically swaps disk contents until an optimal layout is achieved. If there are not enough available spares to support the swaps, Data ONTAP issues a warning and waits to perform the disk swaps until you provide the necessary number of hot spares. If you name disks and an optimal layout cannot be achieved, you must explicitly force the operation; otherwise, the operation fails.

Aggregate creation example

To create an aggregate using MSATA disks, you can specify the disk type and size but leave the disk selection and layout to Data ONTAP by using a command like this:

```
storage aggregate create -aggregate c1n1_aggr1 -node node1 -
disktype MSATA -diskcount 14
```

Adding disks to a storage system

You add disks to a storage system to increase the number of hot spares, to add space to an aggregate, or to replace disks.

Before you begin

You must have confirmed that your storage system supports the type of disk you want to add. For information about supported disk drives, see the *Hardware Universe* at *hwu.netapp.com*.

About this task

You use this procedure to add physical disks to your storage system. If you are administering a storage system that uses virtual disks, for example, a system based on Data ONTAP-v technology, see the installation and administration guide that came with your Data ONTAP-v system for information about adding virtual disks.

Steps

1. Check the NetApp Support Site for newer disk and shelf firmware and Disk Qualification Package files. If your system does not have the latest versions, update them before installing the new disk.
2. Install one or more disks according to the hardware guide for your disk shelf or the hardware and service guide for your storage system.

   The new disks are not recognized until they are assigned to a system. You can assign the new disks manually, or you can wait for Data ONTAP to automatically assign the new disks if your system follows the rules for disk autoassignment.
3. After the new disks have all been recognized, verify their addition and their ownership information by entering the following command:

   `storage disk show -spare`

   You should see the new disks, owned by the correct system, listed as hot spare disks.
4. You can zero the newly added disks now, if needed, by entering the following command:

   `storage disk zerospares`

   Note: Disks that have been used previously in a Data ONTAP aggregate must be zeroed before they can be added to another aggregate. Zeroing the disks now can prevent delays in case you need to quickly increase the size of an aggregate. The disk zeroing command runs in the background and can take hours to complete, depending on the size of the non-zeroed disks in the system.

Result

The new disks are ready to be added to an aggregate, used to replace an existing disk, or placed onto the list of hot spares.

Related concepts

How automatic ownership assignment works for disks on page 43
Guidelines for assigning ownership for disks on page 44

Related information

Disk Qualification Package Instructions: support.netapp.com/NOW/download/tools/diskqual/
Disk Drive & Firmware Matrix: support.netapp.com/NOW/download/tools/diskfw/

When you need to update the Disk Qualification Package

The Disk Qualification Package (DQP) adds full support for newly qualified drives. Before you update drive firmware or add new drive types or sizes to a cluster, you must update the DQP. A best practice is to also update the DQP regularly; for example, every quarter or semi-annually.

You can obtain the DQP from the NetApp Support Site. You need to download and install the DQP in the following situations:

- Whenever you add a new drive type or size to the node
  For example, if you already have 1-TB drives and add 2-TB drives, you need to check for the latest DQP update.
- Whenever you update the disk firmware
- Whenever newer disk firmware or DQP files are available

Related information

Disk Qualification Package Instructions: support.netapp.com/NOW/download/tools/diskqual
Disk Drive & Firmware Matrix: support.netapp.com/NOW/download/tools/diskfw

Replacing disks that are currently being used in an aggregate

You can use the `storage disk replace` command to replace disks that are part of an aggregate without disrupting data service. You do this to swap out mismatched disks from a RAID group. Keeping your RAID groups homogeneous helps optimize storage system performance.

Before you begin

You should already have an appropriate hot spare disk of the correct type, size, speed, and checksum type installed in your storage system. This spare disk must be assigned to the same system and pool as the disk it will replace. For multi-disk carrier disks, you should have at least two hot spare disks available, to enable Data ONTAP to provide an optimal disk layout.

About this task

If you need to replace a disk—for example a mismatched data disk in a RAID group—you can replace the disk. This operation uses Rapid RAID Recovery to copy data from the specified old disk in a RAID group to the specified spare disk in the storage system. At the end of the process, the spare disk replaces the old disk as the new data disk, and the old disk becomes a spare disk in the storage system.

Note: If you replace a smaller disk with a larger disk, the capacity of the larger disk is downsized to match that of the smaller disk; the usable capacity of the aggregate is not increased.

Step

1. Enter the following command:

   storage disk replace -disk *old_disk_name* -replacement *new_spare_disk_name* -action start

   If you need to stop the disk replace operation, you can use the `-action stop` option. If you halt a disk replace operation, the target spare disk needs to be zeroed before it can be used in another aggregate.

Result

The old disk is converted to a spare disk, and the new disk is now used in the aggregate.

Replacing a self-encrypting disk

Replacing a self-encrypting disk (SED) is similar to replacing a regular disk, except that there are some extra steps you must take to reenable Storage Encryption after you replace the disk.

Before you begin

You should know the key used by the SEDs on your storage system so that you can configure the replacement SED to use the same key.

Steps

1. Ensure that reconstruction has started by entering the following command:

   aggr status -r

   The status of the disk should display as "Reconstructing".

2. Remove the failed disk and replace it with a new SED, following the instructions in the hardware guide for your disk shelf model.

3. Assign ownership of the newly replaced SED by entering the following command:

   disk assign *disk_name*

4. Confirm that the new disk has been properly assigned by entering the following command:

   disk encrypt show

   You should see the newly added disk in the output.

5. Encrypt the disk by entering the following command:

   disk encrypt rekey *key_id disk_name*

6. Finalize the replacement process by entering the following command:

   disk encrypt lock *disk_name*

The newly replaced SED is ready for use, and Storage Encryption is enabled and working on this system.

Converting a data disk to a hot spare

Data disks can be converted to hot spares by destroying the aggregate that contains them.

Before you begin

The aggregate to be destroyed cannot contain volumes.

About this task

Converting a data disk to a hot spare does not change the ownership information for that disk. You must remove ownership information from a disk before moving it to another storage system.

Step

1. Destroy the aggregate that contains the disk by entering the following command:

   `storage aggregate delete -aggregate` ***`aggr_name`***

   All disks in use by that aggregate are converted to hot spare disks.

Removing disks from a storage system

How you remove a disk from your storage system depends how the disk is being used. By using the correct procedure, you can prevent unwanted AutoSupport notifications from being generated and ensure that the disk functions correctly if it is reused in another storage system.

You cannot reduce the number of disks in an aggregate by removing data disks. The only way to reduce the number of data disks in an aggregate is to copy the data and transfer it to a new aggregate that has fewer data disks.

Removing a failed disk

A disk that is completely failed is no longer counted by Data ONTAP as a usable disk, and you can immediately disconnect the disk from the disk shelf. However, you should leave a partially failed disk connected long enough for the Rapid RAID Recovery process to complete.

About this task

If you are removing a disk because it has failed or because it is producing excessive error messages, you should not use the disk again in this or any other storage system.

Steps

1. Find the disk ID of the failed disk by entering the following command:

   storage disk show -broken

   If the disk does not appear in the list of failed disks, it might be partially failed, with a Rapid RAID Recovery in process. In this case, you should wait until the disk is present in the list of failed disks (which means that the Rapid RAID Recovery process is complete) before removing the disk.

2. Determine the physical location of the disk you want to remove by entering the following command:

   storage disk set-led -disk *disk_name* 2

   The fault LED on the face of the disk is lit for 2 minutes.

3. Remove the disk from the disk shelf, following the instructions in the hardware guide for your disk shelf model.

Removing a hot spare disk

Removing a hot spare disk requires you to remove ownership information from the disk. This prevents the disk from causing problems when it is inserted into another storage system, and notifies Data ONTAP that you are removing the disk to avoid unwanted AutoSupport messages.

About this task

Removing a hot spare disk does not make the contents of that disk inaccessible. If you need absolute assurance that the data contained by this disk is irretrievable, you should sanitize the disk instead of completing this procedure.

Steps

1. Find the disk name of the hot spare disk you want to remove:

   storage disk show -spare

2. Determine the physical location of the disk you want to remove:

   storage disk set-led -disk *disk_name*

   The fault LED on the face of the disk is lit.

3. If disk ownership automatic assignment is on, turn it off:

   storage disk option modify -node *node_name* -autoassign off

4. Repeat the previous step for the node's HA partner.

5. Remove the software ownership information from the disk:

   storage disk removeowner *disk_name*

6. Remove the disk from the disk shelf, following the instructions in the hardware guide for your disk shelf model.
7. If you turned off disk ownership automatic assignment previously, turn it on now:

   ```
   storage disk option modify -node node_name -autoassign on
   ```

Related concepts

How to determine when it is safe to remove a multi-disk carrier on page 27

Related tasks

Using disk sanitization to remove data from disks on page 36

Removing a data disk

The only time that you should remove a data disk from a storage system is if the disk is not functioning correctly. If you want to remove a data disk so that it can be used in another system, you must convert it to a hot spare disk first.

About this task

You can cause Data ONTAP to fail the disk immediately or allow a disk copy to finish before the disk is failed. If you do not fail the disk immediately, you must wait for the disk copy to finish before physically removing the disk. This operation might take several hours, depending on the size of the disk and the load on the storage system.

Do not immediately fail a disk unless it is causing immediate performance or availability issues for your storage system. Depending on your storage system configuration, additional disk failures could result in data loss.

Steps

1. Determine the name of the disk you want to remove.

   If the disk is reporting errors, you can find the disk name in the log messages that report disk errors. The name is prefixed with the word “Disk”.
2. Determine the physical location of the disk you want to remove by entering the following command:

   ```
   storage disk set-led -disk disk_name 2
   ```

   The red LED on the face of the disk is lit for 2 minutes.
3. Take the appropriate action based on whether you need to fail the disk immediately.

If you...	Then...
Can wait for the copy operation to finish (recommended)	Enter the following command to pre-fail the disk: **`storage disk fail disk_name`** Data ONTAP pre-fails the specified disk and attempts to create a replacement disk by copying the contents of the pre-failed disk to a spare disk. If the copy operation is successful, then Data ONTAP fails the disk and the new replacement disk takes its place. If the copy operation fails, the pre-failed disk fails and the storage system operates in degraded mode until the RAID system reconstructs a replacement disk.
Need to remove the disk immediately	Enter the following command to cause the disk to fail immediately: **`storage disk fail -disk disk_name -immediate`** The disk fails and the storage system operates in degraded mode until the RAID system reconstructs a replacement disk.

4. Ensure that the disk you want to remove is shown as failed by entering the following command, and looking for its disk name:
   `storage disk show -broken`
   Do not remove the disk until its name appears in the list of failed disks.
5. Remove the failed disk from the disk shelf, following the instructions in the hardware guide for your disk shelf model.

Related concepts

About degraded mode on page 95
How to determine when it is safe to remove a multi-disk carrier on page 27

Using disk sanitization to remove data from disks

Disk sanitization enables you to remove data from a disk or set of disks so that the data can never be recovered.

Before you begin

The disks that you want to sanitize must be spare disks; they must be owned by but not used in an aggregate.

About this task

When disk sanitization is enabled, it disables some Data ONTAP commands. Once disk sanitization is enabled on a storage system, it cannot be disabled.

If you need to remove data from disks using Storage Encryption, do not use this procedure. Use the procedure for destroying data on disks using Storage Encryption.

Steps

1. Enter the nodeshell for the system that owns the disks you want to sanitize by entering the following command:

   system node run -node *node_name*

2. Enable the disk sanitization option to be modified by entering the following command:

   options nodescope.reenabledoptions
   licensed_feature.disk_sanitization.enable

3. Enable disk sanitization by entering the following command:

   options licensed_feature.disk_sanitization.enable on

   You are asked to confirm the command because it is irreversible.

4. Sanitize the specified disks by entering the following command:

   disk sanitize start [-p *pattern1*|-r [-p *pattern2*|-r [-p *pattern3*|-r]]]
   [-c *cycle_count*] *disk_list*

   Attention: Do not turn off the storage system, disrupt the storage connectivity, or remove target disks while sanitizing. If sanitizing is interrupted during the formatting phase, the formatting phase must be restarted and allowed to finish before the disks are sanitized and ready to be returned to the spare pool.

   If you need to abort the sanitization process, you can do so by using the `disk sanitize abort` command. If the specified disks are undergoing the formatting phase of sanitization, the abort does not occur until the phase is complete. At that time, Data ONTAP displays a message telling you that the sanitization process was stopped.

   `-p` *`pattern1`* `-p` *`pattern2`* `-p` *`pattern3`* specifies a cycle of one to three user-defined hex byte overwrite patterns that can be applied in succession to the disks being sanitized. The default pattern is three passes, using 0x55 for the first pass, 0xaa for the second pass, and 0x3c for the third pass.

   `-r` replaces a patterned overwrite with a random overwrite for any or all of the passes.

   `-c` *`cycle_count`* specifies the number of times that the specified overwrite patterns are applied. The default value is one cycle. The maximum value is seven cycles.

   `disk_list` specifies a space-separated list of the IDs of the spare disks to be sanitized.

5. If you want to check the status of the disk sanitization process, enter the following command:

   disk sanitize status [*disk_list*]

6. After the sanitization process is complete, return the disks to spare status by entering the following command for each disk:

   disk sanitize release *disk_name*

7. Return to the clustered Data ONTAP CLI by entering the following command:

```
exit
```

8. Determine whether all of the disks were returned to spare status by entering the following command:

   ```
   storage disk show -spare
   ```

If...	Then...
All of the sanitized disks are listed as spares	You are done. The disks are sanitized and in spare status.

Some of the sanitized disks are not listed as spares	Complete the following steps:
	a. Enter advanced privilege mode:
	`set -privilege advanced`
	b. Assign the unassigned sanitized disks to the appropriate node by entering the following command for each disk:
	`storage disk assign -disk disk_name -owner system_name`
	c. Return the disks to spare status by entering the following command for each disk:
	`storage disk unfail -disk disk_name -s`
	d. Return to administrative mode:
	`set -privilege admin`

Result

The specified disks are sanitized and designated as hot spares. The serial numbers of the sanitized disks are written to `/etc/log/sanitized_disks`.

Related concepts

How disk sanitization works on page 16

Stopping disk sanitization

You can use the `disk sanitize abort` command to stop an ongoing sanitization process on one or more specified disks.

Step

1. Enter the following command:

   ```
   disk sanitize abort disk_list
   ```

This command is available through the nodeshell.

If the specified disks are undergoing the disk formatting phase of sanitization, the process does not stop until the disk formatting is complete.

Data ONTAP displays the message `Sanitization abort initiated`. After the process stops, Data ONTAP displays another message for each disk to inform you that sanitization is no longer in progress.

Commands for managing disks

Data ONTAP provides the `storage disk` command for managing disks.

If you want to...	Use this command...
Display a list of failed disks	`storage disk show -broken`
Display a list of spare disks	`storage disk show -spare`
Display a list of disks in the maintenance center	`storage disk show -maintenance`
Display the RAID type of each disk in an aggregate	`storage disk show -aggregate` *`aggr_name`* `-raid`
Display the RAID type, current usage, aggregate and RAID group for disks	`storage disk show -raid`
Display the checksum type for a specific disk	`storage disk show -fields checksum-compatibility`
Display the checksum type for all spare disks	`storage disk show -fields checksum-compatibility -container-type spare`
Display disk connectivity and placement information	`storage disk show -fields disk,primary-port,secondary-name,secondary-port,shelf,bay`
Zero all non-zeroed disks	`storage disk zerospares`

See the man page for each command for more information.

Commands for displaying space information

You can see how space is being used in your aggregates and volumes and their Snapshot copies.

To display information about...	Use this command...
Aggregates, including details about used and available space percentages, Snapshot reserve size, and other space usage information	`storage aggregate show -aggregate` `storage aggregate show-space -snap-size-total,-used-including-snapshot-reserve`
How disks and RAID groups are used in an aggregate and RAID status	`system node run -node <node_name> aggr status -r`
The amount of disk space that would be reclaimed if you deleted a specific Snapshot copy	`volume snapshot compute-reclaimable` (advanced)
The amount of space used by a volume	`volume show -fields size,used,available,percent-used` `volume show-space -vserver <vserver_name>, -volume <volume_name>`
The amount of space used by a volume in the containing aggregate	`volume show-footprint -vserver <vserver_name>, -volume <volume_name>`

For detailed information about these commands, see the appropriate man page.

Managing ownership for disks

Disk ownership determines which node owns a disk. Data ONTAP stores ownership information directly on the disk.

Reasons to assign ownership of disks and array LUNs

Storage system ownership must be assigned for disks and array LUNs before they become an effective part of your system. You must explicitly assign ownership for array LUNs. Disks can be automatically or manually assigned.

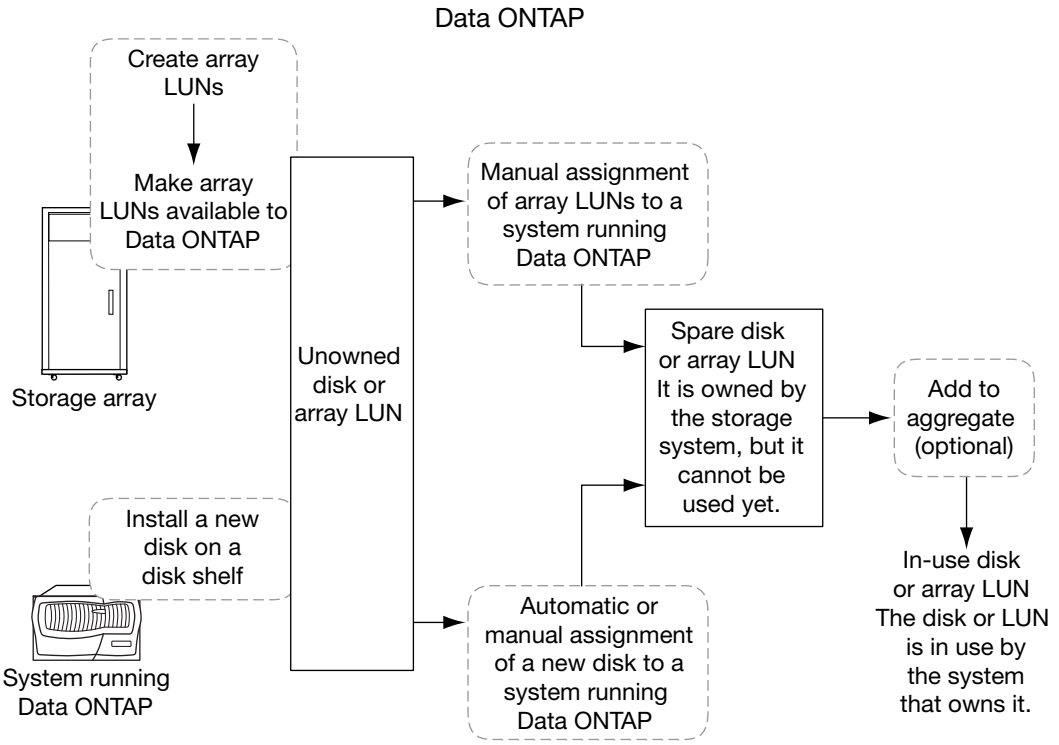
You assign ownership of a disk or array LUN to accomplish the following actions:

- Associate the disk or array LUN with a specific storage system.
  For a stand-alone system, all disks and array LUNs are owned by that system. In an HA configuration, the disks and array LUNs can be owned by either system.
- Enable the disk or array LUN to be used and managed by the system that owns it.
  Unowned disks cannot be used as spares and do not receive the automatic firmware updates that owned disks do.

How disks and array LUNs become available for use

When you add a disk or array LUN to a system running Data ONTAP, the disk or array LUN goes through several stages before it can be used by Data ONTAP to store data or parity information.

The process for making a disk available for use differs slightly from the process for making an array LUN available for use. Both processes are shown in the following diagram:



The process for disks includes the following actions:

1. The administrator physically installs the disk into a disk shelf.
   Data ONTAP can see the disk, but the disk is still unowned.
2. If the system is configured to support disk autoassignment, Data ONTAP assigns ownership for the disk; otherwise, the administrator must assign ownership of the disk manually.
   The disk is now a spare disk.
3. The administrator or Data ONTAP adds the disk to an aggregate.
   The disk is now in use by that aggregate. It could contain data or parity information.

The process for array LUNs includes the following actions:

1. The storage array administrator creates the array LUN and makes it available to Data ONTAP.
   Data ONTAP can see the array LUN, but the array LUN is still unowned.
2. The Data ONTAP administrator assigns ownership of the array LUN to a Data ONTAP system.
   The array LUN is now a spare array LUN.
3. The Data ONTAP administrator adds the array LUN to an aggregate.
   The array LUN is now in use by that aggregate and is storing data.

How automatic ownership assignment works for disks

If your configuration follows some basic rules to avoid ambiguity, Data ONTAP can automatically assign ownership for disks. Automatic ownership assignment is not available for array LUNs or virtual disks.

If you decide to change the way Data ONTAP has assigned the disks, you can do so at any time.

If you need to temporarily remove disk ownership for a disk while you perform an administrative task, you must disable automatic disk ownership first to prevent Data ONTAP from immediately reassigning ownership for that disk.

What automatic ownership assignment does

When automatic disk ownership assignment runs, Data ONTAP looks for any unassigned disks and assigns them to the same system as all other disks on their loop, stack, or shelf.

Note: If a single loop or stack has disks assigned to multiple systems, Data ONTAP does not perform automatic ownership assignment on that loop or stack. Automatic assignment works only when it is clear which system to assign unowned disks to. For this reason, always follow the disk assignment guidelines for your automatic assignment configuration.

You configure Data ONTAP to automatically assign disks at the stack or shelf level, depending on your system requirements and configuration. By default, autoassignment is at the stack or loop level. Data ONTAP automatically assigns the unowned disks to the system that owns the rest of the disks in that stack or loop.

If you need to have the disks in a single stack owned by more than one system, you can configure Data ONTAP to perform automatic disk assignment at the shelf level. In this case, Data ONTAP automatically assigns the unowned disks to the same owner as the already assigned disks on that shelf.

When automatic ownership assignment is invoked

Automatic disk ownership assignment does not happen immediately after disks are introduced into the storage system.

Automatic ownership assignment is invoked at the following times:

- Every five minutes during normal system operation
- Ten minutes after the initial system initialization

  This delay enables the person configuring the system enough time to finish the initial disk assignments so that the results of the automatic ownership assignment are correct.
- Whenever you enable automatic ownership assignment.

How disk ownership works for platforms based on Data ONTAP-v technology

You manage ownership for virtual disks by using the same commands you use for physical disks. However, automatic ownership assignment works differently for virtual disks.

Storage systems based on Data ONTAP-v technology, for example, Data ONTAP Edge systems, create the root volume in aggregate 0 and create spare disks for the virtual data disks you defined during the initial system setup.

For information about adding virtual disks and managing a storage system based on Data ONTAP-v technology, see the *Data ONTAP Edge Installation and Administration Guide*.

Guidelines for assigning ownership for disks

When you assign ownership for disks, you need to follow certain guidelines to keep automatic ownership assignment working and to maximize fault isolation.

Use these guidelines for configuring automatic disk ownership at the stack or loop level:

- Always assign all disks on the same loop or stack to the same system.
- Always assign disks in the same multi-disk carrier to the same system.

Use these guidelines for configuring automatic disk ownership at the shelf level:

- Always assign all disks on the same shelf to the same system.
- On storage systems that contain two controllers but only a single stack, if the stack contains more than one shelf, you can use shelf-level assignment.

Assigning ownership for disks

Disks must be owned by a node before they can be used in an aggregate. If your cluster is not configured to use automatic disk ownership assignment, you must assign ownership manually.

About this task

You can use the wildcard character to assign more than one disk at once.

If you are reassigning a spare disk that is already owned by a different node, you must use the `-force` option for the `storage disk assign` command.

You cannot reassign a disk that is in use in an aggregate.

Steps

1. Display all unowned disks by entering the following command:

   storage disk show -container-type unassigned

2. Assign each disk by entering the following command:

   storage disk assign -disk *disk_name* -owner *owner_name*

Related concepts

How automatic ownership assignment works for disks on page 43

Removing ownership from a disk

Data ONTAP writes disk ownership information to the disk. Before a spare disk or its shelf is removed from a storage system, you should remove its ownership information so that it can be properly integrated into another system.

Before you begin

The disk you want to remove ownership from must meet the following requirements:

- It must be a spare disk.
  You cannot remove ownership from a disk that is being used in an aggregate.
- It cannot be in the maintenance center.
- It cannot be undergoing sanitization.
- It cannot be failed.
  It is not necessary to remove ownership from a failed disk.

About this task

If you have automatic disk assignment turned on, the system could automatically reassign ownership before you remove the disk. For this reason, you turn off automatic ownership until the disk is removed, and then turn it back on.

Steps

1. If disk ownership automatic assignment is on, turn it off:

   storage disk option modify -node *node_name* -autoassign off

2. Repeat the previous step for the node's HA partner.

3. Remove the software ownership information from the disk:

   storage disk removeowner *disk_name*

   If you are removing ownership from a disk that is owned by a different node, you add the `-force` parameter.

To remove ownership information from multiple disks, use a comma-separated list.

Example

storage disk removeowner sys1:0a.23,sys1:0a.24,sys1:0a.25

4. If you turned off disk ownership automatic assignment previously, turn it on after the disk has been removed or reassigned:

   `storage disk option modify -node` *`node_name`* `-autoassign on`

5. Repeat the previous step for the node's HA partner.

Configuring automatic ownership assignment of disks

If you have unowned disks on a stack, loop, or shelf, you can configure Data ONTAP to automatically assign disk ownership at the stack or shelf level.

Before you begin

- Your system must adhere to the requirements for automatic disk ownership.
- If you have multiple stacks or shelves that must have different ownership, one disk must have been manually assigned on each stack or shelf so that automatic ownership assignment will work on each stack or shelf.

About this task

For most system configurations, you can use automatic assignment at the stack or loop level; for smaller configurations, you can use automatic assignment at shelf level.

You can use automatic assignment at the shelf level for the following system configurations:

- Your storage system supports only one stack or loop.
- Your storage system contains two controllers but only a single stack with more than one shelf.
- Your MetroCluster configuration consists of one stack or loop for each node and two shelves. With shelf level automatic assignment, you can have one shelf for one pool and one shelf for the other pool.
- You have any other configuration for which you cannot assign an entire stack or loop to a single system or pool.

If the disks on a single shelf have ownership split between two systems, you need to manually assign those disks; you cannot use automatic disk assignment for the disks on that shelf.

Steps

1. The action you take depends on whether you want to set up automatic ownership assignment at the stack, loop, or shelf level:

If you want to...	Then use the following command...
Configure automatic ownership assignment at the stack or loop level	**storage disk option modify -autoassign on**
Configure automatic ownership assignment at the shelf level	**storage disk option modify -autoassign on -autoassign-shelf on** **Note:** The `-autoassign-shelf` parameter is ignored if `-autoassign` is set to `off`.
Turn off automatic ownership assignment	**storage disk option modify -autoassign off -autoassign-shelf off**

Example

For example, you have a shelf whose disks are owned by one system and another shelf on the same loop whose disks are owned by a different system. In this case, you would configure automatic ownership assignment at the shelf level.

```
storage disk option modify -autoassign-shelf on -node *
```

Data ONTAP automatically assigns unowned disks on the stack, loop, or shelf, depending on the command you entered.

2. Verify the automatic assignment settings for the disks:

   storage disk option show

   Example

```
cluster1::> storage disk option show

Node    BKg. FW. Upd.  Auto Copy  Auto Assign     Auto Assign Shelf
------  -------------  ---------  -----------     -----------------
node0   on             on         on                    on
node1   on             on         on                    on
node2   on             on         on                    on
node3   on             on         on                    on
```

How you use the wildcard character with the disk ownership commands

You can use the wildcard character ("*") with some commands, including commands to manage disk ownership. However, you should understand how Data ONTAP expands the wildcard character.

You can use the wildcard character with the following disk ownership commands:

- storage disk modify
- storage disk assign
- storage disk show
- storage disk removeowner

When you use the wildcard character with these commands, Data ONTAP expands it with zero or more characters to create a list of disk names that will be operated on by the command. This can be very useful when you want to assign all of the disks attached to a particular port or switch, for example.

Note: Be careful when you use the wildcard character. It is accepted anywhere in the disk name string, and is a simple string substitution. Therefore, you might get unexpected results.

For example, to assign all disks on port 1 of the switch brocade23 to node03, you would use the following command:

```
storage disk assign -disk brocade23:1.* -owner node03
```

However, if you left off the second ".", as in the following command, you would assign all disks attached to ports 1, 10, 11, 12, and so on:

```
storage disk assign -disk brocade23:1* -owner node03
```

Managing array LUNs using Data ONTAP

For Data ONTAP to be able to use storage on a storage array, some tasks must be done on the storage array and some tasks must be done in Data ONTAP.

For example, the storage array administrator must create array LUNs for Data ONTAP use and map them to Data ONTAP. You can then assign them to nodes running Data ONTAP.

If the storage array administrator wants to make configuration changes to an array LUN after it is assigned to a node, for example to resize it, you might need to perform some activities in Data ONTAP before it is possible to reconfigure the LUN on the storage array.

Related concepts

How ownership for disks and array LUNs works on page 53

Data ONTAP systems that can use array LUNs on storage arrays

V-Series (“V”) systems and new FAS platforms released in Data ONTAP 8.2.1 and later can use array LUNs if the proper license is installed. In discussions in the Data ONTAP and FlexArray Virtualization documentation, these systems are collectively referred to as Data ONTAP systems when it is necessary to make it clear which information applies to them and what information applies to storage arrays.

Note: Starting with Data ONTAP 8.2.1, the capability of using LUNs on a storage array, formerly identified as V-Series functionality, has a new name—*Data ONTAP FlexArray Virtualization Software*. Although the V-Series product name has been retired as of 8.2.1, Data ONTAP can access LUNs on all the same storage arrays as before, with the same functionality that V-Series (the product) provided.

Systems prior to Data ONTAP 8.2.1 that can use array LUNs

The only systems released prior to Data ONTAP 8.2.1 that can use array LUNs are V-Series systems —systems with a “V” or “GF” prefix. A V-Series system is an open storage controller that virtualizes storage from storage array vendors, native disks, or both into a single heterogeneous storage pool.

Note: Almost all Data ONTAP platforms released prior to Data ONTAP 8.2.1 were released with FAS and V-Series equivalent models (for example, a FAS6280 and a V6280). (For a few systems, there were no “V” equivalent models.) Although both types of models could access native disks, only the V-Series systems (a “V” or “GF” prefix) could attach to storage arrays.

Systems in Data ONTAP 8.2.1 and later that can use array LUNs

Starting with Data ONTAP 8.2.1, the model for how platforms are released and the storage they can use changes. Attaching to storage arrays is no longer limited to V-Series systems.

Starting with Data ONTAP 8.2.1, all new platforms are being released as a single hardware model. This single hardware model has a FAS prefix; there are no longer separate "V" and FAS models for new platforms. If the V_StorageAttach license package is installed on a new FAS model, it can attach to storage arrays. (This is the same license required on a V-Series system.) Clustered Data ONTAP systems must have disks, even if they are using array LUNs for storage.

Important: FAS systems released prior to Data ONTAP 8.2.1 cannot use LUNs on storage arrays, even if they are upgraded to 8.2.1; only the "V" equivalent of a platform can use array LUNs.

Overview of setting up Data ONTAP to use array LUNs

You should complete basic setup of the cluster to work with native disks before setting up the cluster nodes to use LUNs on storage arrays. The storage array administrator must present the array LUNs to Data ONTAP and configure the required storage array parameters before you can configure the nodes to use the array LUNs.

Note: Storage array administrators can prepare storage for Data ONTAP any time before you assign the array LUNs to nodes running Data ONTAP.

Setup task	Who typically performs the task	Where to find information
1. Configure a cluster, join nodes to it, and verify basic setup.	Data ONTAP administrator	*Clustered Data ONTAP Software Setup Guide*
2. Set up Data ONTAP features and test them.	Data ONTAP administrator	Various Data ONTAP guides
3. Create LUNs and make them available to Data ONTAP.	Storage array administrator or vendor	*FlexArray Virtualization Installation Requirements and Reference Guide* (Data ONTAP guidelines and requirements) Storage array documentation (how to create LUNs and make them available to hosts)
4. Configure parameters on the storage array so that it can work with Data ONTAP.	Storage array administrator or vendor	*FlexArray Virtualization Implementation Guide for Third-Party Storage* (for storage array parameters that must be set to work with Data ONTAP)

Setup task	Who typically performs the task	Where to find information
5. Connect the nodes to the storage array.	Data ONTAP administrator	*FlexArray Virtualization Installation Requirements and Reference Guide* (for connection procedures)
6. Install the license for accessing LUNs on storage arrays, if it has not yet been installed. The license must be installed on every storage system running Data ONTAP with which you want to use array LUNs.	Data ONTAP administrator	*Clustered Data ONTAP Physical Storage Management Guide* *Clustered Data ONTAP System Administration Guide for Cluster Administrators*
7. Verify that there are no errors that would prevent the nodes from using the array LUNs, and that the configuration is set up as intended.	Data ONTAP administrator and the storage array administrator or vendor	*FlexArray Virtualization Installation Requirements and Reference Guide*
8. Assign ownership of array LUNs to specific nodes.	Data ONTAP administrator	*Clustered Data ONTAP Physical Storage Management Guide*
9. Create additional aggregates and assign additional LUNs to nodes as needed.	Data ONTAP administrator	*Clustered Data ONTAP Physical Storage Management Guide*

See the *FlexArray Virtualization Installation Requirements and Reference Guide* for general requirements for setting up storage arrays to work with Data ONTAP and the *FlexArray Virtualization Implementation Guide for Third-Party Storage* for required storage array-specific parameters to work with Data ONTAP.

Related tasks

Assigning ownership of array LUNs on page 58

Installing the license for using array LUNs

The V_StorageAttach license must be installed on each Data ONTAP node that you want to use with array LUNs. It is *not* a single license for the cluster. Array LUNs cannot be used in aggregates until a license is installed.

Before you begin

- The cluster must be installed.
- You must have the license key for the V_StorageAttach license.
  The license key is available on the NetApp Support Site at *support.netapp.com*.

  Note: The license key is different for releases prior to Data ONTAP 8.2.

About this task

You do not need to perform this procedure if the license key for the V_StorageAttach package is already installed. Clustered Data ONTAP systems must be ordered with disks, and typically the factory installs the license package for you. Alternatively, many customers install all necessary licenses early in the installation process.

Steps

1. For each Data ONTAP node in the cluster for use with array LUNs, enter the following command on the node:

   `system license add` ***`license key`***

   Example

   ```
   vgv3170f41a> license
   Serial Number: nnnnnnnn
   Owner: mysystem1a
   Package           Type    Description           Expiration
   ----------------- ------- --------------------- --------------------
   V_StorageAttach   license Virtual Attached Storage
   ```

2. Look at the output to confirm that the V_StorageAttach package is shown.

How ownership for disks and array LUNs works

Disk and array LUN ownership determines which node owns a disk or array LUN. Understanding how ownership works enables you to maximize storage redundancy and manage your hot spares effectively.

Data ONTAP stores ownership information directly on the disk or array LUN.

Reasons to assign ownership of disks and array LUNs

Storage system ownership must be assigned for disks and array LUNs before they become an effective part of your system. You must explicitly assign ownership for array LUNs. Disks can be automatically or manually assigned.

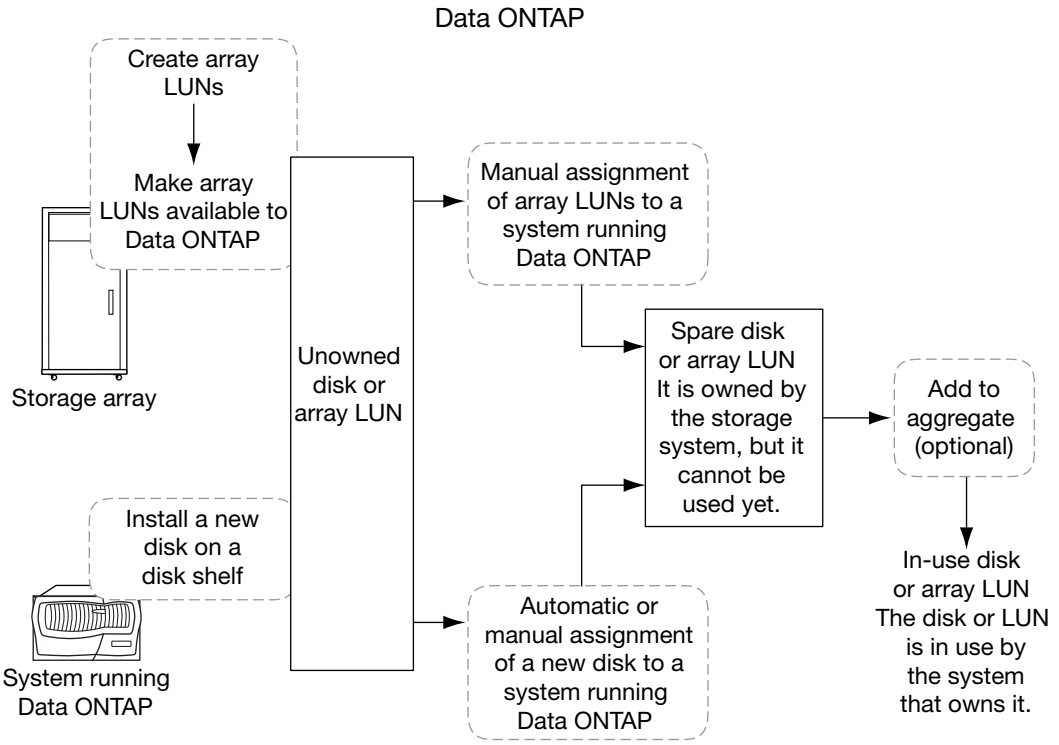
You assign ownership of a disk or array LUN to accomplish the following actions:

- Associate the disk or array LUN with a specific storage system.
  For a stand-alone system, all disks and array LUNs are owned by that system. In an HA configuration, the disks and array LUNs can be owned by either system.
- Enable the disk or array LUN to be used and managed by the system that owns it.
  Unowned disks cannot be used as spares and do not receive the automatic firmware updates that owned disks do.

How disks and array LUNs become available for use

When you add a disk or array LUN to a system running Data ONTAP, the disk or array LUN goes through several stages before it can be used by Data ONTAP to store data or parity information.

The process for making a disk available for use differs slightly from the process for making an array LUN available for use. Both processes are shown in the following diagram:



The process for disks includes the following actions:

1. The administrator physically installs the disk into a disk shelf.
   Data ONTAP can see the disk, but the disk is still unowned.
2. If the system is configured to support disk autoassignment, Data ONTAP assigns ownership for the disk; otherwise, the administrator must assign ownership of the disk manually.
   The disk is now a spare disk.
3. The administrator or Data ONTAP adds the disk to an aggregate.
   The disk is now in use by that aggregate. It could contain data or parity information.

The process for array LUNs includes the following actions:

1. The storage array administrator creates the array LUN and makes it available to Data ONTAP.
   Data ONTAP can see the array LUN, but the array LUN is still unowned.
2. The Data ONTAP administrator assigns ownership of the array LUN to a Data ONTAP system.
   The array LUN is now a spare array LUN.
3. The Data ONTAP administrator adds the array LUN to an aggregate.
   The array LUN is now in use by that aggregate and is storing data.

What it means for Data ONTAP to own an array LUN

Data ONTAP cannot use an array LUN presented to it by a storage array until you configure a logical relationship in Data ONTAP that identifies a specific system running Data ONTAP as the *owner* of the array LUN.

A storage array administrator creates array LUNs and makes them available to specified FC initiator ports of storage systems running Data ONTAP. (The process for how to do this varies among storage array vendors.) When you assign an array LUN to a system running Data ONTAP, Data ONTAP writes data to the array LUN to identify that system as the *owner* of the array LUN. Thereafter, Data ONTAP ensures that only the owner can write data to and read data from the array LUN.

From the perspective of Data ONTAP, this logical relationship is referred to as *disk ownership* because Data ONTAP considers an array LUN to be a virtual disk. From the perspective of Data ONTAP, you are assigning disks to a storage system.

An advantage of the disk ownership scheme is that you can make changes through the Data ONTAP software that, on typical hosts, must be done by reconfiguring hardware or LUN access controls. For example, through Data ONTAP you can balance the load of requests among a group of systems running Data ONTAP by moving data service from one system to another, and the process is transparent to most users. You do not need to reconfigure hardware or the LUN access controls on the storage array to change which system running Data ONTAP is the owner and, therefore, servicing data requests.

Attention: The Data ONTAP software-based scheme provides ownership control only for storage systems running Data ONTAP; it does not prevent a different type of host from overwriting data in an array LUN owned by a system running Data ONTAP. Therefore, if multiple hosts are accessing array LUNs through the same storage array port, be sure to use LUN security on your storage array to prevent the systems from overwriting each other's array LUNs.

Array LUN reconfiguration, such as resizing the array LUN, must be done from the storage array. Before such activities can occur, you must release Data ONTAP ownership of the array LUN.

Why you might assign array LUN ownership after installation

For a Data ONTAP system ordered with disk shelves, you are not required to set up the system to work with array LUNs during initial installation. In the case of a Data ONTAP system using only array LUNs, you need to assign only two array LUNs during initial installation.

If you ordered your Data ONTAP system with disk shelves, you do not need to assign any array LUNs initially because the factory installs the root volume on a disk for you. If you are using only array LUNs, you must configure one array LUN for the root volume and one array LUN as a spare for core dumps during initial installation. In either case, you can assign ownership of additional array LUNs to your system at any time after initial installation.

After you configure your system, you might assign ownership of an array LUN in the following circumstances:

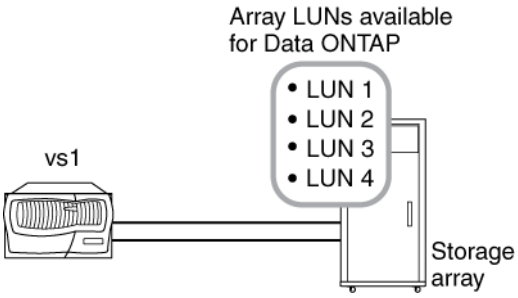
- You ordered your Data ONTAP system with native disk shelves and did not set up your system to work with array LUNs initially
- You left some LUNs that the storage array presented to Data ONTAP unowned, and you now need to use the storage
- Another system released the ownership of a particular array LUN and you want this system to use the LUN
- The storage array administrator has not made the LUNs available to Data ONTAP during the initial system configuration, and you want to use the storage.

Examples showing when Data ONTAP can use array LUNs

After an array LUN has been assigned to a storage system, it can be added to an aggregate and used for storage or it can remain a spare LUN until it is needed for storage.

No storage system owns the LUNs yet

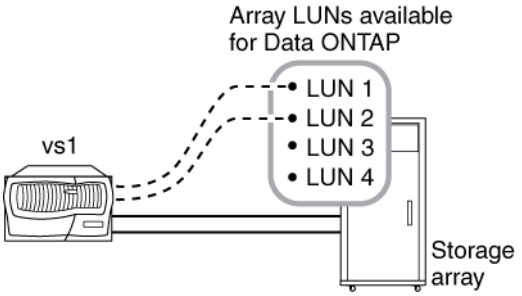
In this example, the storage array administrator made the array LUNs available to Data ONTAP. However, system vs1 has not yet been configured to "own" any of the LUNs. Therefore, it cannot read data from or write data to any array LUNs on the storage array:



Only some array LUNs are owned

In this example, vs1 was configured to own array LUNs 1 and 2, but not array LUNs 3 and 4. LUNs 3 and 4 are still available to Data ONTAP, however, and can be assigned to a storage system later:

Data ONTAP used LUN 1 for the root volume. System vs1 can read data from and write data to LUN 1, because LUN 1 is in an aggregate. LUN 2 remains a spare LUN because it has not yet been added to an aggregate. System vs1 cannot read data from and write data to LUN 2 while it is a spare.

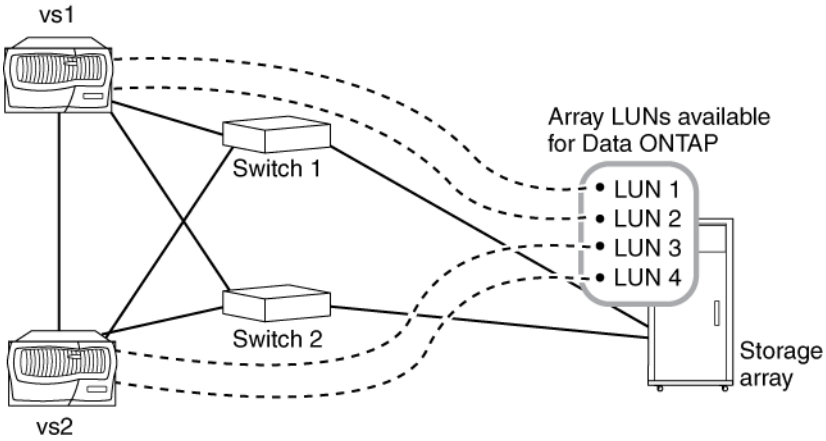


After you perform initial setup of the storage system, you could configure vs1 to also own LUN 3, LUN 4, both, or neither, depending on your storage needs.

Ownership of LUNs in an HA pair

In this example, two storage systems running Data ONTAP are configured in an HA pair. In an HA pair, only one node can be the owner of a particular LUN, but both nodes must be able to see the same LUNs so that the partner can take over if the owning node becomes unavailable.

LUN 1 through LUN 4 were created on the storage array and mapped to the ports on the storage array to which the storage systems are connected. All four LUNs are visible to each node in the HA pair.



Assume that during initial setup vs1 was assigned ownership of LUN 1 and LUN 2. LUN 1 was automatically added to the root volume, so LUN 1 is now “in use” by vs1. LUN 2 remains a spare until it is explicitly added to an aggregate on vs1. Similarly, assume that during initial setup vs2 was assigned ownership of LUN 3 and LUN 4, with LUN 3 assigned to the root volume. LUN 4 remains a spare LUN until it is explicitly added to an aggregate.

The key points of this example are as follows:

- By deploying the storage systems in an HA pair, one system can take over services for its partner if the partner becomes unavailable.
- Only one storage system can own a specific array LUN.

  However, all array LUNs assigned to a node in an HA pair must be visible to—but not assigned to or owned by—the other node in the HA pair.
- By deploying two switches, if one switch fails, the other switch provides the alternate path to the storage array.
- Both switches must be zoned correctly so that each storage system in the HA pair can see the array LUNs owned by its partner.

Assigning ownership of array LUNs

Array LUNs must be owned by a node before they can be added to an aggregate to be used as storage.

Before you begin

- Back-end configuration testing (testing of the connectivity and configuration of devices behind the Data ONTAP systems) must be completed.
- Array LUNs that you want to assign must be presented to the Data ONTAP systems.

About this task

You can assign ownership of array LUNs that have the following characteristics:

- They are unowned.
- They have no storage array configuration errors, such as the following:
  - The array LUN is smaller than or larger than the size that Data ONTAP supports.
  - The LDEV is mapped on only one port.
  - The LDEV has inconsistent LUN IDs assigned to it.
  - The LUN is available on only one path.

Data ONTAP issues an error message if you try to assign ownership of an array LUN with back-end configuration errors that would interfere with the Data ONTAP system and the storage array operating together. You must fix such errors before you can proceed with array LUN assignment. See the *FlexArray Virtualization Installation Requirements and Reference Guide* for information about how to fix these types of errors.

Data ONTAP alerts you if you try to assign an array LUN with a redundancy error: for example, all paths to this array LUN are connected to the same controller or only one path to the array LUN. You can fix a redundancy error before or after assigning ownership of the LUN.

Steps

1. Enter the following command to see the array LUNs that have not yet been assigned to a node:

   `storage disk show -container-type unassigned`

2. Enter the following command to assign an array LUN to this node:

   `storage disk assign -disk` *`arrayLUNname`* `-owner` *`nodename`*

   If you want to fix a redundancy error after disk assignment instead of before, you must use the `-force` parameter with the `storage disk assign` command.

   If you want the array LUN to be designated as an AZCS checksum type, you must add `-c zoned` to your command.

   The default and recommended checksum type is block. For a description of the block (BCS) and advanced checksum (AZCS) types, see the *FlexArray Virtualization Installation Requirements and Reference Guide*.

Related concepts

How ownership for disks and array LUNs works on page 53

Related tasks

Modifying assignment of spare array LUNs on page 60

Verifying back-end configuration

It is important to detect and resolve any configuration errors before you bring the configuration online in a production environment. You start installation verification by using `storage array config show` command.

The `storage array show config` command shows how storage arrays connect to the cluster. If Data ONTAP detects an error in the back-end configuration, the following message is displayed at the bottom of the `storage array show config` output:

```
Warning: Configuration errors were detected.  Use 'storage errors show'
for detailed information.
```

You then use the `storage errors show` output to see details of the problem, at the LUN level. You must fix any errors shown by `storage errors show`.

For detailed information about what back-end configuration you need to verify and how to do it, see the *FlexArray Virtualization Installation Requirements and Reference Guide*.

Modifying assignment of spare array LUNs

You can change the ownership of a *spare* array LUN to another node. You might want to do this for load balancing over the nodes.

Steps

1. At the console of the node that owns the array LUN you want to reassign, enter the following command to see a list of spare array LUNs on the node:

   `storage disk show -owner local`

   The array LUNs owned by the node, both spares and LUNs in aggregates, are listed.
2. Confirm that the LUN you want to reassign to another node is a spare LUN.
3. Enter the following command to assign ownership of the array LUN to another node:

   `storage disk assign` ***`arrayLUNname`*** **`-owner`** ***`new_owner_name`*** **`-force`**

   Note: The array LUN ownership is not changed if the `-force` option is not used or if the array LUN was already added to an aggregate.
4. Enter the following command to verify that the ownership of the spare array LUN was changed to the other node:

   `storage disk show -owner local`

   The spare array LUN that you changed to the new owner should no longer appear in the list of spares. If the array LUN still appears, repeat the command to change ownership.
5. On the destination node, enter the following command to verify that the spare array LUN whose ownership you changed is listed as a spare owned by the destination node:

   `storage disk show -owner local`

After you finish

You must add the array LUN to an aggregate before it can be used for storage.

Related concepts

How ownership for disks and array LUNs works on page 53

Related tasks

Assigning ownership of array LUNs on page 58

Array LUN name format

The array LUN name is a path-based name that includes the devices in the path between the Data ONTAP system and the storage array, ports used, and the SCSI LUN ID on the path that the storage array presents externally for mapping to hosts.

On a Data ONTAP system that supports array LUNs, each array LUN can have multiple names because there are multiple paths to each LUN.

Array LUN name format for clustered Data ONTAP systems

Configuration	Array LUN name format	Component descriptions
Direct-attached	*node-name.adapter.idlun-id*	*node-name* is the name of the clustered node. With clustered Data ONTAP, the node name is prepended to the LUN name so that the path-based name is unique within the cluster. *adapter* is the adapter number on the Data ONTAP system. *id* is the channel adapter port on the storage array. *lun-id* is the array LUN number that the storage array presents to hosts. Example: *node1.0a.0L1*

Configuration	Array LUN name format	Component descriptions
Fabric-attached	*node-name*`:`*switch-name*`:`*port*`.`*id*`L`*lun-id*	*node-name* is the name of the node. With clustered Data ONTAP, the node name is prepended to the LUN name so that the path-based name is unique within the cluster. *switch-name* is the name of the switch. *port* is the switch port that is connected to the target port (the end point). *id* is the device ID. *lun-id* is the array LUN number that the storage array presents to hosts. Example: *node1*`:`*brocade3*`:`*6*`.`*126L1*

Related concepts

Drive name formats on page 13

Guidelines for adding storage to a storage system that uses array LUNs

Starting with Data ONTAP 8.2.1, memory is allocated dynamically for storage devices as you present them to a storage system that uses array LUNs. Depending on the memory configuration of the particular storage system, you should set the *bootarg.disk.init.dynamic.allocation* variable to `true` or `false` at boot time.

If you set the *bootarg.disk.init.dynamic.allocation* variable to `true` (for example **`setenv bootarg.disk.init.dynamic.allocation? true`**) at boot time, memory is allocated only for the number of storage devices visible to the system at that time. Memory is allocated dynamically if and when you present more storage devices to the system at run time.

If you set the *bootarg.disk.init.dynamic.allocation* variable to `false` (for example **`setenv bootarg.disk.init.dynamic.allocation? false`**) at boot time, memory is allocated for the maximum number of storage devices allowed for that platform.

By default, the *bootarg.disk.init.dynamic.allocation* variable is set to `true`.

Checking the checksum type of spare array LUNs

If you plan to add a spare array LUN to an aggregate by specifying its name, you need to make sure that the checksum type of the array LUN you want to add is the same as the aggregate checksum type.

About this task

You cannot mix array LUNs of different checksum types in an array LUN aggregate. The checksum type of the aggregate and the checksum type of the array LUNs added to it must be the same.

If you specify a number of spare array LUNs to be added to an aggregate, by default Data ONTAP selects array LUNs of the same checksum type as the aggregate.

Note: Data ONTAP 8.1.1 and later supports a new checksum scheme called *advanced zoned checksum* (AZCS). Existing zoned checksum aggregates are still supported. The checksum type of all newly created aggregates using zoned checksum array LUNS is AZCS, which provides more functionality than the "version 1" zoned checksum type that was supported in previous releases and continues to be supported for existing zoned aggregates. Zoned checksum spare array LUNs added to an existing zoned checksum aggregate continue to be zoned checksum array LUNs. Zoned checksum spare array LUNs added to an AZCS checksum type aggregate use the AZCS checksum scheme for managing checksums.

Step

1. Check the checksum type of the spare array LUNs by entering the following command:

For...	The command is...
Clustered Data ONTAP	`storage disk show -fields checksum-compatibility -container-type spare` You can add a block checksum array LUN to a block checksum aggregate and a zoned array LUN to either a zoned checksum aggregate or an AZCS checksum aggregate.

Related tasks

Changing the checksum type of an array LUN on page 64

Changing the checksum type of an array LUN

You must change the checksum type of an array LUN if you want to add it to an aggregate that has a different checksum type than the checksum type of the LUN.

Before you begin

You should have reviewed the tradeoffs between performance in certain types of workloads and storage capacity utilization of each checksum type. The *FlexArray Virtualization Installation Requirements and Reference Guide* contains information about checksum use for array LUNs. You can also contact your Sales Engineer for details about using checksums.

About this task

You must assign a `zoned` checksum type to an array LUN that you plan to add to a zoned checksum aggregate or an advanced zoned checksum (AZCS) aggregate. When a zoned checksum array LUN is added to an AZCS aggregate, it becomes an advanced zoned checksum array LUN. Similarly, when a zoned checksum array LUN is added to a zoned aggregate, it is a zoned checksum type.

Step

1. Enter the following command to change the checksum type:

   `storage disk assign -disk` *`LUN-path`* `-o` *`owner`* `-c` *`new_checksum_type`*

   `LUN-path` is the <current-owner>:<LUNname> of the array LUN whose checksum type you want to change.

   `owner` is the current owner.

   `new_checksum_type` can be `block` or `zoned`.

   #### Example

   ```
   storage disk assign -disk system147b:vgbr300s181:5.126L2 -o system147b -
   c block
   ```

   The checksum type of the array LUN is changed to the new checksum type you specified.

Related tasks

Checking the checksum type of spare array LUNs on page 63

Prerequisites to reconfiguring an array LUN on the storage array

If an array LUN has already been assigned (through Data ONTAP) to a particular Data ONTAP system, the information that Data ONTAP wrote to the array LUN must be removed before the storage administrator attempts to reconfigure the array LUN on the storage array.

When the storage array presents an array LUN to Data ONTAP, Data ONTAP collects information about the array LUN (for example, its size) and writes that information to the array LUN. Data ONTAP cannot dynamically update information that it wrote to an array LUN. Therefore, before the storage array administrator reconfigures an array LUN, you must use Data ONTAP to change the state of the array LUN to *unused*. The array LUN is unused from the perspective of Data ONTAP.

While changing the state of the array LUN to unused, Data ONTAP does the following:

- Terminates I/O operations to the array LUN
- Removes the label for RAID configuration information and the persistent reservations from the array LUN, which makes the array LUN unowned by any Data ONTAP system

After this process finishes, no Data ONTAP information remains in the array LUN.

You can do the following after the array LUN's state is changed to unused:

- Remove the mapping of the array LUN to Data ONTAP and make the array LUN available to other hosts.
- Resize the array LUN or change its composition.

If you want Data ONTAP to resume using the array LUN after its size or composition is changed, you must present the array LUN to Data ONTAP again, and assign the array LUN to a Data ONTAP system again. Data ONTAP is aware of the new array LUN size or composition.

Related tasks

Changing array LUN size or composition on page 65

Changing array LUN size or composition

Reconfiguring the size or composition of an array LUN must be done on the storage array. If an array LUN has already been assigned to a Data ONTAP system, you must use Data ONTAP to change the state of the array LUN to unused before the storage array administrator can reconfigure it.

Before you begin

The array LUN must be a spare array LUN before you can change its state to unused.

Steps

1. On the Data ONTAP system, enter the following command to remove ownership information:

   `storage disk removeowner -disk LUNfullname`

2. On the storage array, complete the following steps:

   a) Unmap (unpresent) the array LUN from the Data ONTAP systems so that they can no longer see the array LUN.
   b) Change the size or composition of the array LUN.
   c) If you want Data ONTAP to use the array LUN again, present the array LUN to the Data ONTAP systems again.

      At this point, the array LUN is visible to the FC initiator ports to which the array LUN was presented, but it cannot be used by any Data ONTAP systems yet.

3. Enter the following command on the Data ONTAP system that you want to be the owner of the array LUN:

   `storage disk assign -disk arrayLUNname -owner nodename`

   If you want the array LUN to be designated as an AZCS checksum type, you must add `-c zoned` to your command.

   After the ownership information is removed, the array LUN cannot be used by any Data ONTAP system until the array LUN is assigned again to a system. You can leave the array LUN as a spare or add it to an aggregate. You must add the array LUN to an aggregate before the array LUN can be used for storage.

Related concepts

Prerequisites to reconfiguring an array LUN on the storage array on page 65

Removing one array LUN from use by Data ONTAP

If the storage array administrator no longer wants to use a particular array LUN for Data ONTAP, you must remove the information that Data ONTAP wrote to the LUN (for example, size and ownership) before the administrator can reconfigure the LUN for use by another host.

Before you begin

If the LUN that the storage array administrator no longer wants Data ONTAP to use is in an aggregate, you must take the aggregate offline and destroy the aggregate before starting this procedure. Taking an aggregate offline and destroying it changes the data LUN to a spare LUN.

Step

1. Enter the following command:

   `storage disk removeowner -disk LUN_name`

LUN_name is the name of the array LUN.

Preparing array LUNs before removing a Data ONTAP system from service

You must release the persistent reservations on all array LUNs assigned to a Data ONTAP system before removing the system from service.

About this task

When you assign Data ONTAP ownership of an array LUN, Data ONTAP places persistent reservations (ownership locks) on that array LUN to identify which Data ONTAP system owns the LUN. If you want the array LUNs to be available for use by other types of hosts, you must remove the persistent reservations that Data ONTAP put on those array LUNs; some arrays do not allow you to destroy a reserved LUN if you do not remove the ownership and persistent reservations that Data ONTAP wrote to that LUN.

For example, the Hitachi USP storage array does not have a user command for removing persistent reservations from LUNs. If you do not remove persistent reservations through Data ONTAP before removing the Data ONTAP system from service, you must call Hitachi technical support to remove the reservations.

Contact technical support for instructions about how to remove persistent reservations from LUNs before removing a Data ONTAP system from service.

Introduction to Storage Encryption

Overview of Storage Encryption concepts, functionality, benefits, and limitations.

What Storage Encryption is

Storage Encryption is an optional feature that you can enable for additional data protection. It is available on certain supported storage controllers and disk shelves that contain disks with built-in encryption functionality.

In a standard storage environment, data is written to disk in cleartext format. This makes the data vulnerable to potential exposure to unauthorized users when disks removed from a storage system are lost or stolen.

When you enable Storage Encryption, the storage system protects your data at rest by storing it on self-encrypting disks.

The authentication keys used by the self-encrypting disks are stored securely on external key management servers.

Purpose of the external key management server

An external key management server is a third-party system in your storage environment that securely manages authentication keys used by the self-encrypting disks in the storage system. You link the external key management server to other systems that use authentication or encryption keys such as your storage system.

The storage system uses a secure SSL connection to connect to the external key management server to store and retrieve authentication keys. The communication between the storage system and key management server uses the Key Management Interoperability Protocol (KMIP).

The external key management server securely stores authentication or encryption keys entrusted to it and provides them upon demand to authorized linked systems. This provides an additional level of security by storing authentication keys separate from the storage system. Additionally, authentication keys are always handled and stored securely. The keys are never displayed in cleartext.

You must link at least one key management server to the storage system during the Storage Encryption setup and configuration process. You should link multiple key management servers for redundancy. If the only key management server in the environment becomes unavailable, access to protected data might become unavailable until the key management server is available again. For example, when the storage system needs to unlock self-encrypting disks but cannot retrieve the authentication key from the key management server because it is unavailable.

You can specify up to four key servers during or after setup for redundancy.

For a list of supported key management servers, see the Interoperability Matrix.

How Storage Encryption works

Storage Encryption occurs at the firmware level of disks that are equipped with special firmware and hardware to provide the additional security, also known as *self-encrypting disks (SEDs)*. SEDs can operate either in unprotected mode like regular disks, or in protected mode requiring authentication after the power-on process.

SEDs always encrypt data for storage. In unprotected mode, the encryption key needed to decrypt and access the data is freely available. In protected mode, the encryption key is protected and requires authentication to be used.

When you first enable and configure Storage Encryption on a storage system using SEDs, you create an authentication key that the storage system uses to authenticate itself to the SEDs. You configure the storage system with the IP address to one or more external key management servers that securely stores the authentication key.

The storage system communicates with the key management servers at boot time to retrieve the authentication keys. Data ONTAP requires the authentication keys to authenticate itself to the SEDs any time after the SEDs are power-cycled.

If the authentication is successful, the SEDs are unlocked. The SEDs use the authentication key to decrypt the data encryption keys stored inside the disk. When presented with a read request, SEDs automatically decrypt the stored data before passing it on to the storage system. When presented with a write request from the storage system, SEDs automatically encrypt the data before writing the data to the disk's storage platters. When the SEDs are *locked*, Data ONTAP must successfully authenticate itself to the disk before the SEDs allow data to be read or written. When locked, SEDs require authentication each time the disk is powered on.

Encryption and decryption happens without a perceptible disk performance decrease or boot time increase. Storage Encryption does not require a separate license key. The only additional required component is an external key management server.

When you halt and power down the storage system, including the disk shelves containing SEDs, the disks are locked again and the data becomes inaccessible.

Disk operations with SEDs

Most of the disk-related operations are identical for SEDs and regular disks.

Because storage encryption happens at a very low level, specifically the disk firmware, it does not affect any higher level functionality. The storage controller sees SEDs the same as regular disks, and all functionality remains the same.

There are some additional options and requirements with SEDs:

- Sanitizing disks
  There are additional options to sanitize disks when using SEDs.

- Destroying disks
  An additional option enables you to make the disks permanently inaccessible.

Benefits of using Storage Encryption

There are several scenarios where using Storage Encryption provides significant benefits by protecting data from unauthorized access when disks removed from a storage system have fallen into the wrong hands.

Data protection in case of disk loss or theft

Storage Encryption protects your data if disks are lost or stolen.

Someone who comes into possession of disks that store data using Storage Encryption cannot access the data. Without the authentication key that is required to authenticate and unlock the disks, all attempts to read or write data result in an error message returned by the SEDs.

Circumventing the disk authentication by moving the platters into another disk without encryption firmware would be unsuccessful as well. The data stored on the platters appears as ciphertext and is fully protected from unauthorized access.

Data protection when returning disks to vendors

Storage Encryption protects your data when you return disks to vendors.

The following three options are available to protect data on disks that are removed from a storage system and returned to a vendor:

- If the SED is owned by a storage system, it requires authentication to access the data. Since the vendor does not know, or have access to, the authentication key, the vendor cannot access data on the disk.
- If you sanitize the disk before returning it to a vendor, it changes the encryption key to a new unknown key. Any subsequent attempts to read data from the disk result in random data.
- If you "destroy" the disk, it changes the encryption key to a random unknown key, it changes the authentication key to a random unknown key, and permanently locks the disk, preventing any further decryption of the data and access to the disk.

Related tasks

Sanitizing disks using Storage Encryption before return to vendor on page 84

Data protection when moving disks to end-of-life

Storage Encryption protects your data when moving a disk to an end-of-life state.

You can protect data on a disk by changing the authentication key to a random value that is not stored and permanently locking the drive. This prevents any further decryption of the data and access to the disk.

Related tasks

Setting the state of disks using Storage Encryption to end-of-life on page 85

Data protection through emergency data shredding

Storage Encryption protects your data in emergency situations by allowing you to instantaneously prevent access to the data on the disk.

This might include extreme scenarios where power to the storage system or the key management server (or both) is not available, or one or both have fallen into possession of a hostile third-party.

Related tasks

Emergency shredding of data on disks using Storage Encryption on page 86

Limitations of Storage Encryption

You must keep certain limitations in mind when using Storage Encryption.

- For the latest information about which storage systems, disk shelves, and key management servers are supported with Storage Encryption, see the Interoperability Matrix.
- All disks in the storage system and optional attached disk shelves must have encryption functionality to be able to use Storage Encryption. You cannot mix regular non-encrypting disks with self-encrypting disks.
- Storage Encryption is not supported with Flash Pool aggregates.
- Storage Encryption `key_manager` commands are only available for local nodes.
  They are not available in takeover mode for partner nodes.
- Do not configure Storage Encryption to use 10 Gigabit network interfaces for communication with key management servers. This limitation does not apply to serving data.
- Storage Encryption supports a maximum of 128 authentication keys per key management server.
  You receive a warning when the number of stored authentication keys reaches 100. You cannot create new authentication keys when the number of stored authentication keys reaches the limit of 128. You must then delete unused authentication keys before you can create new ones.

Related information

Interoperability Matrix: support.netapp.com/matrix

Managing Storage Encryption

You can perform various tasks to manage Storage Encryption, including viewing and removing key management servers, and creating, deleting, restoring and synchronizing authentication keys.

Displaying Storage Encryption disk information

You can display information about self-encrypting disks by using the `disk encrypt show` command. This command displays the key ID and lock status for each self-encrypting disk.

About this task

The key ID displayed in the command output is an identifier used by Storage Encryption and key management servers as a reference to the authentication key. It is not the actual authentication key or the data encryption key.

Steps

1. Access the nodeshell by entering the following command:

   `system node run -node` *`node_name`*

2. To display information about SEDs, enter the following command:

   `disk encrypt show`

   The `disk encrypt show`, `lock`, and `rekey` commands support extended wildcard matching. For more information, see the `disk encrypt show` man page.

3. Exit the nodeshell and return to the clustershell by entering the following command:

   `exit`

Example

The following command displays the status of each self-encrypting disk:

```
storage-system> disk encrypt show
Disk       Key
ID
Locked?
0c.00.1
080CF0C8000000000100000000000000A948EE8604F4598ADFFB185B5BB7FED3
Yes
0c.00.0
080CF0C8000000000100000000000000A948EE8604F4598ADFFB185B5BB7FED3
Yes
0c.00.3
080CF0C8000000000100000000000000A948EE8604F4598ADFFB185B5BB7FED3
```

```
Yes
0c.00.4
080CF0C8000000000100000000000000A948EE8604F4598ADFFB185B5BB7FED3
Yes
0c.00.2
080CF0C8000000000100000000000000A948EE8604F4598ADFFB185B5BB7FED3
Yes
0c.00.5
080CF0C8000000000100000000000000A948EE8604F4598ADFFB185B5BB7FED3
Yes
```

Displaying key management server information

You can display information about the external key management servers associated with the storage system by using the `key_manager show` command.

Steps

1. Access the nodeshell by entering the following command:

   `system node run -node` ***`node_name`***

2. To display external key management servers, enter the following command:

   `key_manager show`

   All external key management servers associated with the storage system are listed.

3. Exit the nodeshell and return to the clustershell by entering the following command:

   `exit`

Example

The following command displays all external key management servers associated with the storage system:

```
storage-system> key_manager show
  172.18.99.175
```

Verifying key management server links

You use the `key_manager status` or `key_manager query` commands to verify that all key management servers are successfully linked to the storage system. These commands are useful for verifying proper operation and troubleshooting.

About this task

Both commands display whether key management servers are responding.

Steps

1. Access the nodeshell by entering the following command:

   system node run -node *node_name*

2. Perform one of the following actions:

If you want to...	Then enter the following command:
Check the status of a specific key management server	**key_manager status -key_server *key_server_ip_address***
Check the status of all key management servers	**key_manager status**
Check the status of all key management servers and view additional server details.	**key_manager query** The `key_manager query` command displays additional information about key tags and key IDs.

3. Check the output to verify that all of the appropriate keys are available in the Data ONTAP key table.

   If the output of the `key_manager query` command displays key IDs marked with an asterisk (*), those keys exist on a key server but are not currently available in the Data ONTAP key table. To import those keys from the key management server into the key table, enter the following command:

   key_manager restore

4. Exit the nodeshell and return to the clustershell by entering the following command:

   exit

Examples

The following command checks the status of all key management servers linked to the storage system:

```
storage-system> key_manager status
Key server                      Status
172.16.132.118                   Server is responding
172.16.132.211                   Server is responding
```

The following command checks the status of all key management servers linked to the storage system and displays additional information:

```
storage-system> key_manager query
Key server 172.16.132.118 is responding.
Key server 172.16.132.211 is responding.

Key server 172.16.132.118 reports 4 keys.

Key tag                          Key ID
--------                         -------
storage-system                      080CDCB20...
storage-system                      080CDCB20...
storage-system                      080CDCB20...
storage-system                      080CDCB20...

Key server 172.16.132.211 reports 4 keys.

Key tag                          Key ID
--------                         -------
storage-system                    *080CDCB20...
storage-system                     080CDCB20...
storage-system                     080CDCB20...
storage-system                    *080CDCB20...
```

Adding key management servers

You can use the `key_manager add` command to link key management servers to the storage system. This enables you to add additional key management servers for redundancy after initial setup or to replace existing key management servers.

Before you begin

You must first install the required storage system and key management server SSL certificates. If they are not present, the command fails.

You must know the IP address for each key management server you want to link.

Steps

1. Access the nodeshell by entering the following command:
   `system node run -node` ***`node_name`***
2. To add a key management server, enter the following command:

```
key_manager add -key_server key_server_ip_address
```

3. Exit the nodeshell and return to the clustershell by entering the following command:
   ```
   exit
   ```

Example

The following command adds a link from the storage system to the key management server with the IP address 172.16.132.118:

```
storage-system> key_manager add -key_server 172.16.132.118
Found client certificate file client.pem.
Registration successful for client.pem.
Found client private key file client_private.pem.
Is this file protected by a passphrase? [no]: no
Registration successful for client_private.pem.
Registering 1 key servers...
Found client CA certificate file 172.16.132.118_CA.pem.
Registration successful for 172.16.132.118_CA.pem.
Registration complete.
```

Removing key management servers

If you no longer want to use a key management server to store authentication keys used by self-encrypting disks in the storage system, you can remove the key management server link to the storage system by using the `key_manager remove` command.

Before you begin

You must know the IP address for each key management server that you want to remove.

About this task

Storage Encryption requires at least one key management server linked to the storage system to operate. If you want to replace a single key management server with another one, you must first add the new one before removing the old one.

Steps

1. Access the nodeshell by entering the following command:
   ```
   system node run -node node_name
   ```
2. To remove key management servers, enter the following command:
   ```
   key_manager remove -key_server key_server_ip_address
   ```
   `-key_server` *`key_server_ip_address`* specifies the IP address of the key management server you want to remove.

3. Exit the nodeshell and return to the clustershell by entering the following command:

   exit

Example

The following command removes the link between the storage system and the key management server with the IP address 172.18.99.175:

```
storage-system> key_manager remove -key_server 172.18.99.175
Key server 172.18.99.175 will be unregistered from service.
Unregistration successful.
```

What happens when key management servers are not reachable during the boot process

Data ONTAP takes certain precautions to avoid undesired behavior in the event that the storage system cannot reach any of the specified key management servers during the boot process.

If the storage system is configured for Storage Encryption, the SEDs have been rekeyed and locked, and the SEDs are powered on, the storage system must retrieve the required authentication keys from the key management servers to authenticate itself to the SEDs before it can access the data.

The storage system attempts to contact the specified key management servers for up to three hours. If the storage system cannot reach any of them after that time, the boot process stops and the storage system halts.

If the storage system successfully contacts any specified key management server, it then attempts to establish an SSL connection for up to 15 minutes. If the storage system cannot establish an SSL connection with any specified key management server, the boot process stops and the storage system halts.

While the storage system attempts to contact and connect to key management servers, it displays detailed information about the failed contact attempts at the CLI. You can interrupt the contact attempts at any time by pressing Ctrl-C.

As a security measure, SEDs allow only a limited number of unauthorized access attempts, after which they permanently disable access to the existing data. If the storage system cannot contact any specified key management servers to obtain the proper authentication keys, it can only attempt to authenticate with the default key which leads to a failed attempt and a panic. If the storage system is configured to automatically reboot in case of a panic, it would enter a boot loop which results in continuous failed authentication attempts on the SEDs.

Halting the storage system in these scenarios is by design to prevent the storage system from entering a boot loop and unintended data loss as a result of the SEDs locked permanently due to exceeding the safety limit of 1024 consecutive failed authentication attempts.

If you encounter this scenario where the storage system is halted due to failure to reach any specified key management servers, you must first identify and correct the cause for the communication failure before you attempt to continue booting the storage system.

Changing the authentication key

You can change the authentication key at any time by using the `key_manager rekey` command. You might want to change the authentication key as part of your security protocol or when moving an aggregate to another storage system.

Steps

1. Access the nodeshell by entering the following command:

   `system node run -node` ***`node_name`***

2. Perform one of the following actions:

If you want to...	Then...
Change the authentication key and enter a new one manually	**a.** Enter the following command at the storage system prompt: **`key_manager rekey -manual -key_tag`** ***`key_tag`*** **b.** When prompted, enter the new authentication key. It must be 20 to 32 characters long.
Change the authentication key and have the system generate a new one automatically	Enter the following command at the storage system prompt: **`key_manager rekey -key_tag`** ***`key_tag`***

   `key_tag` is the label used to associate keys with a particular storage system. If you do not specify a key tag, the storage system uses the key tag specified when you set up Storage Encryption. If you did not specify this key tag during setup, it uses the parent key tag as the default. Each node has a parent key tag. HA pair members share the same parent key tag.

3. Exit the nodeshell and return to the clustershell by entering the following command:

   `exit`

Example

The following command changes the authentication key and prompts you to enter a new one manually. You can run the `disk encrypt show` command after completion to verify the results.

```
storage-system> key_manager rekey -manual
Please enter a new passphrase:
Please reenter the new passphrase:
```

```
New passphrase generated.
Key ID:
080CDCB2000000000100000000000000B0A11CBF3DDD20EFB0FBB5EE198DB22A
Key tag: storage-system

Notice: Remember to store the passphrase and the Key ID in a secure
location.

Passphrase, key ID, and key tag synchronized with the following key
server(s):
 172.16.132.118
 172.16.132.211
Completed rekey on 4 disks: 4 successes, 0 failures, including 0
unknown key and 0 authentication failures.
```

Retrieving authentication keys

You can use the `key_manager restore` command to retrieve authentication keys from a key management server to a storage system. For example, when you created authentication keys on a node, you use this command to retrieve the keys for use on the partner node.

Before you begin

You must know the IP address for each key management server that you want to retrieve authentication keys from.

Steps

1. Access the nodeshell by entering the following command:

   system node run -node *node_name*

2. To retrieve authentication keys from a key management server to the storage system, enter the following command:

   key_manager restore -key_server *key_server_ip_address* -key_tag *key_tag*

   If all specified key management servers are available, you can use the `-all` option instead of the `-key_server` option to clear out the current Data ONTAP key table and retrieve all keys matching the specified key tag from all specified key management servers.

3. Exit the nodeshell and return to the clustershell by entering the following command:

   exit

Examples

The following command restores keys with the key tag storage-system from the key management server with the IP address 172.18.99.175:

```
storage-system> key_manager restore -key_server 172.18.99.175 -
key_tag storage-system
```

The following command restores all keys with the key tag storage-system from all key management servers linked to the storage system:

```
storage-system> key_manager restore -all -key_tag storage-system
```

Deleting an authentication key

You can delete an authentication key that is no longer needed by removing it from the external key management server.

Before you begin

Verify that the authentication key is no longer needed before deleting it. Deleting an authentication key that is still in use can permanently prevent access to data on a storage system.

Step

1. Refer to the documentation for the external key management server for details on how to delete stored authentication keys.

SSL issues due to expired certificates

If the SSL certificates used to secure key management communication between the storage system and key management servers expire, the storage system can no longer retrieve authentication keys from the key management server at bootup. This issue can cause data on SEDs to be unavailable. You can prevent this issue by updating all SSL certificates before their individual expiration dates.

SSL certificates have a limited lifespan because they have an expiration date. After the SSL certificates reach their expiration dates, the certificates are no longer valid. When this happens, SSL connections that use expired certificates fail.

For Storage Encryption, this means that the SSL connections between the storage system and the key management servers fail, the storage system no longer can retrieve authentication keys when needed, and data access to the SEDs fails, resulting in storage system panic and downtime.

To prevent this issue from occurring, you must keep track of the expiration dates of all installed SSL certificates so that you can obtain new SSL certificates before they expire.

After you have obtained the new certificate files, you must first remove the existing certificate files from the storage system, and then install the new certificates on the storage system.

Steps

1. Removing old SSL certificates before installing new ones on page 81
   If you want to update or reinstall the SSL certificates used by Storage Encryption, you must first manually remove the old ones to ensure that the new ones are used.
2. Installing replacement SSL certificates on the storage system on page 81
   After you remove the old certificates, you create the new replacement SSL certificates, save them with the proper file name and format, and then install them on the storage system.

Removing old SSL certificates before installing new ones

If you want to update or reinstall the SSL certificates used by Storage Encryption, you must first manually remove the old ones to ensure that the new ones are used.

Steps

1. Access the nodeshell by entering the following command:
   `system node run -node node_name`
2. Remove the IP addresses of all key management servers by entering the following command for each key management server:
   `key_manager remove -key_server key_server_ip_address`
3. Remove the storage system's client certificates by entering the following commands:
   `keymgr delete cert client_private.pem`
   `keymgr delete cert client.pem`
4. Remove all installed key management server certificates by entering the following commands for each key management server:
   `keymgr delete cert key_server_ip_address_CA.pem`
5. Exit the nodeshell and return to the clustershell by entering the following command:
   `exit`

Installing replacement SSL certificates on the storage system

After you remove the old certificates, you create the new replacement SSL certificates, save them with the proper file name and format, and then install them on the storage system.

Before you begin

- You must have removed the old certificates that are about to expire from the storage system.
- You must have obtained the replacement public and private certificates for the storage system and the public certificate for the key management server, and named them as required.
  For more information, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.
- You must have installed the appropriate new certificates on the key management server.

For more information, see the documentation for your key management server.

Steps

1. Access the nodeshell by entering the following command:

   system node run -node *node_name*

2. Copy the certificate files to a temporary location on the storage system.
3. Install the public certificate of the storage system by entering the following command:

   keymgr install cert */path/*client.pem

4. Install the private certificate of the storage system by entering the following command:

   keymgr install cert */path/*client_private.pem

5. Install the public certificate of all key management servers by entering the following command for each key management server:

   keymgr install cert */path/key_management_server_ipaddress_*CA.pem

6. Add all key management servers by entering the following command for each key management server:

   key_manager add -key_server *key_server_ip_address*

7. Verify connectivity between the storage system and key management servers by entering the following command:

   key_manager query

   You should see a list of existing key IDs retrieved from the key management servers.

8. Exit the nodeshell and return to the clustershell by entering the following command:

   exit

Returning SEDs to unprotected mode

If your storage system is configured to use Storage Encryption but you decide to stop using this feature, you can do so by returning the SEDs to unprotected mode. You cannot disable Storage Encryption altogether because SEDs always encrypt data for storage. However, you can return them to unprotected mode where they no longer use secret authentication keys, and use the default MSID instead.

Steps

1. Access the nodeshell by entering the following command:

   system node run -node *node_name*

2. To change the authentication key for all SEDs on the storage system back to the default MSID, enter the following command:

disk encrypt rekey * 0x0

3. If you expect to operate the storage system in unprotected mode permanently, you should also remove all key management servers by entering the following command for each one:

   key_manager remove -key_server *key_server_ip_address*

   -key_server *key_server_ip_address* specifies the IP address of the key management server you want to remove.

   The storage system displays two kmip_init errors during every bootup after you remove all key management servers. These errors are normal in this situation and you can disregard them.

4. If you expect to operate the storage system in unprotected mode permanently and you removed all key management servers in the preceding step, you should view the list of installed Storage Encryption related SSL certificates, and then remove all key management server SSL certificates:

   keymgr cert list

   keymgr delete cert client.pem

   keymgr delete cert client_private.pem

   keymgr delete cert *key_management_server_ipaddress*_CA.pem

   If you had multiple key management servers linked to the storage system, repeat the last command for each public certificate of each key management server.

5. Exit the nodeshell and return to the clustershell by entering the following command:

   exit

Destroying data on disks using Storage Encryption

You can destroy data stored on disks using Storage Encryption for security reasons, including sanitizing the disks, setting the disk state to end-of-life, and emergency shredding of the data.

Sanitizing disks using Storage Encryption before return to vendor

If you want to return a disk to a vendor but do not want anyone to access sensitive data on the disk, you can sanitize it first by using the `disk encrypt sanitize` command. This renders the data on the disk inaccessible, but the disk can be reused. This command only works on spare disks.

Steps

1. Migrate any data that needs to be preserved to a different aggregate.
2. Destroy the aggregate.
3. Access the nodeshell by entering the following command:

   system node run -node *node_name*
4. Identify the disk ID for the disk to be sanitized by entering the following command:

   disk encrypt show
5. Enter the following command:

   disk encrypt sanitize *disk_ID*
6. Exit the nodeshell and return to the clustershell by entering the following command:

   exit

Example

The following command sanitizes a self-encrypting disk with the disk ID 0c.00.3. You can run the `sysconfig -r` command before and after the operation to verify the results.

```
storage-system> sysconfig -r
Aggregate aggr0 (online, raid_dp) (block checksums)
  Plex /aggr0/plex0 (online, normal, active)
    RAID group /aggr0/plex0/rg0 (normal)

      RAID Disk    Device     HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Phys (MB/blks)
      ---------    ------     ------------ ---- ---- ---- ----- --------------    --------------
      dparity     0c.00.0    0c    0   0   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
      parity      0c.00.1    0c    0   1   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
      data        0c.00.2    0c    0   2   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
```

```
Spare disks

RAID Disk    Device     HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Phys (MB/blks)
---------    ------     ------------- ---- ---- ---- ----- --------------    --------------
Spare disks for block or zoned checksum traditional volumes or aggregates
spare       0c.00.3    0c    0   3   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
spare       0c.00.4    0c    0   4   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
spare       0c.00.5    0c    0   5   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688

storage-system> disk encrypt sanitize 0c.00.3
storage-system> Wed Jun 30 17:49:16 PDT [disk.failmsg:error]: Disk 0c.00.3 (3SL04F3V00009015WTHU):
message received.
Wed Jun 30 17:49:16 PDT [raid.disk.unload.done:info]: Unload of Disk 0c.00.3 Shelf 0 Bay 3 [SYSTEM
X415_S15K7560A15 NQS3] S/N [3SL04F3V00009015WTHU] has completed successfully

storage-system> Wed Jun 30 17:49:25 PDT [disk.sanit.complete:info]: Disk 0c.00.3 [S/N
3SL04F3V00009015WTHU] has completed sanitization.

storage-system> sysconfig -
r
Aggregate aggr0 (online, raid_dp) (block checksums)
  Plex /aggr0/plex0 (online, normal, active)
    RAID group /aggr0/plex0/rg0 (normal)

      RAID Disk    Device     HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Phys (MB/blks)
      ---------    ------     ------------- ---- ---- ---- ----- --------------    --------------
      dparity     0c.00.0    0c    0   0   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
      parity      0c.00.1    0c    0   1   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
      data        0c.00.2    0c    0   2   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688


Spare disks

RAID Disk    Device     HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Phys (MB/blks)
---------    ------     ------------- ---- ---- ---- ----- --------------    --------------
Spare disks for block or zoned checksum traditional volumes or aggregates
spare       0c.00.4    0c    0   4   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
spare       0c.00.5    0c    0   5   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688

Maintenance disks

RAID Disk    Device     HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Phys (MB/blks)
---------    ------     ------------- ---- ---- ---- ----- --------------    --------------
sanitized    0c.00.3    0c    0   3   SA:B   -  SAS  15000 560000/1146880000 560208/1147307688
storage-system>
```

Setting the state of disks using Storage Encryption to end-of-life

If you want to render a disk permanently unusable and the data on it inaccessible, you can set the state of the disk to end-of-life by using the `disk encrypt destroy` command. This command only works on spare disks.

Steps

1. Remove any data from the aggregate containing the disk.
2. Migrate any data that needs to be preserved to a different aggregate.
3. Destroy the aggregate.
4. Access the nodeshell by entering the following command:
   `system node run -node node_name`
5. Enter the following command:
   `disk encrypt destroy disk_ID`

6. Exit the nodeshell and return to the clustershell by entering the following command:

   ```
   exit
   ```

Result

The disk's encryption key is set to an unknown random value and the disk is irreversibly locked. The disk is now completely unusable and can be safely disposed of without risk of unauthorized data access.

Emergency shredding of data on disks using Storage Encryption

In case of a security emergency, you can instantly prevent access to data on disks using Storage Encryption, even if power is not available to the storage system or the external key server.

Before you begin

You must configure the external key server so that it only operates if an easily destroyed authentication item (for example, a smart card or USB drive) is present. See the documentation for the external key management server for details.

About this task

The steps for emergency shredding vary depending on whether power is available to the storage system and the external key server.

Step

1. Perform one of the following actions:

If...	Then...
Power is available to the storage system and you have time to gracefully take the storage system offline	**a.** If the storage system is a node in an HA pair, disable takeover. **b.** Take all aggregates offline and destroy them. **c.** Halt the storage system. **d.** Boot into maintenance mode. **e.** Enter the following command: `disk encrypt sanitize -all` This leaves the storage system in a permanently disabled state with all data erased. To use the storage system again, you must set it up from the beginning.

If...	Then...

Power is available to the storage system and you must shred the data immediately; time is critical	**a.** If the storage system is a node in an HA pair, disable takeover.
	b. Access the nodeshell by entering the following command:
	`system node run -node node_name`
	c. Set the privilege level to advanced.
	d. Enter the following command:
	`disk encrypt sanitize -all`
	The storage system panics, which is expected due to the abrupt nature of the procedure. It leaves the storage system in a permanently disabled state with all data erased. To use the storage system again, you must set it up from the beginning.

Power is available to the external key server but not to the storage system	**a.** Log in to the external key server. **b.** Destroy all keys associated with the disks containing data to protect.
Power is not available to the external key server or the storage system	Destroy the authentication item for the key server (for example, the smart card). If power to the systems is restored, the external key server cannot operate due to the missing authentication item. This prevents access to the disk encryption keys by the storage system, and therefore access to the data on the disks.

How Data ONTAP uses RAID to protect your data and data availability

Understanding how RAID protects your data and data availability can help you administer your storage systems more effectively.

For native storage, Data ONTAP uses RAID-DP (double-parity) or RAID Level 4 (RAID4) protection to ensure data integrity within a RAID group even if one or two of those drives fail. Parity drives provide redundancy for the data stored in the data drives. If a drive fails (or, for RAID-DP, up to two drives), the RAID subsystem can use the parity drives to reconstruct the data in the drive that failed.

For array LUNs, Data ONTAP stripes data across the array LUNs using RAID0. The storage arrays, not Data ONTAP, provide the RAID protection for the array LUNs that they make available to Data ONTAP.

RAID protection levels for disks

Data ONTAP supports two levels of RAID protection for aggregates composed of disks in native disk shelves: RAID-DP and RAID4. RAID-DP is the default RAID level for new aggregates.

For more information about configuring RAID, see *Technical Report 3437: Storage Subsystem Resiliency Guide*.

Related information

TR 3437: Storage Subsystem Resiliency Guide

What RAID-DP protection is

If an aggregate is configured for RAID-DP protection, Data ONTAP reconstructs the data from one or two failed disks within a RAID group and transfers that reconstructed data to one or two spare disks as necessary.

RAID-DP provides double-parity disk protection when the following conditions occur:

- There is a single-disk failure or double-disk failure within a RAID group.
- There are media errors on a block when Data ONTAP is attempting to reconstruct a failed disk.

The minimum number of disks in a RAID-DP group is three: at least one data disk, one regular parity disk, and one double-parity (dParity) disk. However, for non-root aggregates with only one RAID group, you must have at least 5 disks (three data disks and two parity disks).

If there is a data-disk failure or parity-disk failure in a RAID-DP group, Data ONTAP replaces the failed disk in the RAID group with a spare disk and uses the parity data to reconstruct the data of the

failed disk on the replacement disk. If there is a double-disk failure, Data ONTAP replaces the failed disks in the RAID group with two spare disks and uses the double-parity data to reconstruct the data of the failed disks on the replacement disks.

RAID-DP is the default RAID type for all aggregates.

What RAID4 protection is

RAID4 provides single-parity disk protection against single-disk failure within a RAID group. If an aggregate is configured for RAID4 protection, Data ONTAP reconstructs the data from a single failed disk within a RAID group and transfers that reconstructed data to a spare disk.

The minimum number of disks in a RAID4 group is two: at least one data disk and one parity disk. However, for non-root aggregates with only one RAID group, you must have at least 3 disks (two data disks and one parity disk).

If there is a single data or parity disk failure in a RAID4 group, Data ONTAP replaces the failed disk in the RAID group with a spare disk and uses the parity data to reconstruct the failed disk's data on the replacement disk. If no spare disks are available, Data ONTAP goes into degraded mode and alerts you of this condition.

Attention: With RAID4, if there is a second disk failure before data can be reconstructed from the data on the first failed disk, there will be data loss. To avoid data loss when two disks fail, you can select RAID-DP. This provides two parity disks to protect you from data loss when two disk failures occur in the same RAID group before the first failed disk can be reconstructed.

RAID protection for array LUNs

Storage arrays provide the RAID protection for the array LUNs that they make available to Data ONTAP; Data ONTAP does not provide the RAID protection.

Data ONTAP uses RAID0 (striping) for array LUNs. Data ONTAP supports a variety of RAID types on the storage arrays, except RAID0 because RAID0 does not provide storage protection.

When creating *RAID groups* on storage arrays, you need to follow the best practices of the storage array vendor to ensure that there is an adequate level of protection on the storage array so that disk failure does not result in loss of data or loss of access to data.

Note: A *RAID group* on a storage array is the arrangement of disks that together form the defined RAID level. Each RAID group supports only one RAID type. The number of disks that you select for a RAID group determines the RAID type that a particular RAID group supports. Different storage array vendors use different terms to describe this entity—RAID groups, parity groups, disk groups, Parity RAID groups, and other terms.

Data ONTAP supports RAID4 and RAID-DP on the native disk shelves connected to a V-Series system but does not support RAID4 and RAID-DP with array LUNs.

RAID protection for Data ONTAP-v storage

Because Data ONTAP-v storage is connected to the host server, rather than a storage system running Data ONTAP, the host server provides the RAID protection for the physical disks. Data ONTAP uses RAID0 for the virtual disks to optimize performance.

See the *Data ONTAP Edge Installation and Administration Guide* for more information.

Understanding RAID disk types

Data ONTAP classifies disks as one of four types for RAID: data, hot spare, parity, or dParity. The RAID disk type is determined by how RAID is using a disk; it is different from the Data ONTAP disk type.

Data disk	Holds data stored on behalf of clients within RAID groups (and any data generated about the state of the storage system as a result of a malfunction).
Spare disk	Does not hold usable data, but is available to be added to a RAID group in an aggregate. Any functioning disk that is not assigned to an aggregate but is assigned to a system functions as a hot spare disk.
Parity disk	Stores row parity information that is used for data reconstruction when a single disk drive fails within the RAID group.
dParity disk	Stores diagonal parity information that is used for data reconstruction when two disk drives fail within the RAID group, if RAID-DP is enabled.

How RAID groups work

A RAID group consists of one or more data disks or array LUNs, across which client data is striped and stored, and up to two parity disks, depending on the RAID level of the aggregate that contains the RAID group.

RAID-DP uses two parity disks to ensure data recoverability even if two disks within the RAID group fail.

RAID4 uses one parity disk to ensure data recoverability if one disk within the RAID group fails.

RAID0 does not use any parity disks; it does not provide data recoverability if any disks within the RAID group fail.

How RAID groups are named

Within each aggregate, RAID groups are named rg0, rg1, rg2, and so on in order of their creation. You cannot specify the names of RAID groups.

About RAID group size

A RAID group has a maximum number of disks or array LUNs that it can contain. This is called its maximum size, or its size. A RAID group can be left partially full, with fewer than its maximum number of disks or array LUNs, but storage system performance is optimized when all RAID groups are full.

Related references

Storage limits on page 148

Considerations for sizing RAID groups for drives

Configuring an optimum RAID group size for an aggregate made up of drives requires a trade-off of factors. You must decide which factor—speed of recovery, assurance against data loss, or maximizing data storage space—is most important for the aggregate that you are configuring.

You change the size of RAID groups on a per-aggregate basis. You cannot change the size of an individual RAID group.

HDD RAID groups

You should follow these guidelines when sizing your RAID groups composed of HDDs:

- All RAID groups in an aggregate should have the same number of disks.
  If this is impossible, any RAID group with fewer disks should have only one less disk than the largest RAID group.
- The recommended range of RAID group size is between 12 and 20.
  The reliability of performance disks can support a RAID group size of up to 28, if needed.
- If you can satisfy the first two guidelines with multiple RAID group sizes, you should choose the larger size.

SSD RAID groups in Flash Pool aggregates

The SSD RAID group size can be different from the RAID group size for the HDD RAID groups in a Flash Pool aggregate. Usually, you should ensure that you have only one SSD RAID group for a Flash Pool aggregate, to minimize the number of SSDs required for parity.

SSD RAID groups in SSD-only aggregates

You should follow these guidelines when sizing your RAID groups composed of SSDs:

- All RAID groups in an aggregate should have the same number of drives.

If this is impossible, any RAID group with fewer drives should have only one less drive than the largest RAID group.

- The recommended range of RAID group size is between 20 and 28.

Related references

Storage limits on page 148

Considerations for Data ONTAP RAID groups for array LUNs

Setting up Data ONTAP RAID groups for array LUNs requires planning and coordination with the storage array administrator so that the administrator makes the number and size of array LUNs you need available to Data ONTAP.

For array LUNs, Data ONTAP uses RAID0 RAID groups to determine where to allocate data to the LUNs on the storage array. The RAID0 RAID groups are not used for RAID data protection. The storage arrays provide the RAID data protection.

Note: Data ONTAP RAID groups are similar in concept to what storage array vendors call RAID groups, parity groups, disk groups, Parity RAID groups, and other terms.

Follow these steps when planning your Data ONTAP RAID groups for array LUNs:

1. Plan the size of the aggregate that best meets your data needs.
2. Plan the number and size of RAID groups that you need for the size of the aggregate.

   Follow these guidelines:

   - RAID groups in the same aggregate should be the same size with the same number of LUNs in each RAID group. For example, you should create four RAID groups of 8 LUNs each, not three RAID groups of 8 LUNs and one RAID group of 6 LUNs.
   - Use the default RAID group size for array LUNs, if possible. The default RAID group size is adequate for most organizations.

     Note: The default RAID group size is different for array LUNs and disks.
3. Plan the size of the LUNs that you need in your RAID groups.

   - To avoid a performance penalty, all array LUNs in a particular RAID group should be the same size.
   - The LUNs should be the same size in all RAID groups in the aggregate.
4. Ask the storage array administrator to create the number of LUNs of the size you need for the aggregate.

   The LUNs should be optimized for performance, according to the instructions in the storage array vendor documentation.
5. Create all the RAID groups in the aggregate at the same time.

   Note: Do not mix array LUNs from storage arrays with different characteristics in the same Data ONTAP RAID group.

Note: If you create a new RAID group for an existing aggregate, be sure that the new RAID group is the same size as the other RAID groups in the aggregate, and that the array LUNs are the same size as the LUNs in the other RAID groups in the aggregate.

How Data ONTAP works with hot spare disks

A hot spare disk is a disk that is assigned to a storage system but is not in use by a RAID group. It does not yet hold data but is ready for use. If a disk failure occurs within a RAID group, Data ONTAP automatically assigns hot spare disks to RAID groups to replace the failed disks.

How many hot spares you should have

Having insufficient spares increases the risk of a disk failure with no available spare, resulting in a degraded RAID group. The number of hot spares you should have depends on the Data ONTAP disk type.

MSATA disks, or disks in a multi-disk carrier, should have four hot spares during steady state operation, and you should never allow the number of MSATA hot spares to dip below two.

For RAID groups composed of SSDs, you should have at least one spare disk.

For all other Data ONTAP disk types, you should have at least one matching or appropriate hot spare available for each kind of disk installed in your storage system. However, having two available hot spares for all disks provides the best protection against disk failure. Having at least two available hot spares provides the following benefits:

- When you have two or more hot spares for a data disk, Data ONTAP can put that disk into the maintenance center if needed.

  Data ONTAP uses the maintenance center to test suspect disks and take offline any disk that shows problems.
- Having two hot spares means that when a disk fails, you still have a spare available if another disk fails before you replace the first failed disk.

A single spare disk can serve as a hot spare for multiple RAID groups.

Related concepts

Spare requirements for multi-disk carrier disks on page 28

What disks can be used as hot spares

A disk must conform to certain criteria to be used as a hot spare for a particular data disk.

For a disk to be used as a hot spare for another disk, it must conform to the following criteria:

- It must be either an exact match for the disk it is replacing or an appropriate alternative.
- The spare must be owned by the same system as the disk it is replacing.

What a matching spare is

A matching hot spare exactly matches several characteristics of a designated data disk. Understanding what a matching spare is, and how Data ONTAP selects spares, enables you to optimize your spares allocation for your environment.

A matching spare is a disk that exactly matches a data disk for all of the following criteria:

- Effective Data ONTAP disk type
  The effective disk type can be affected by the value of the `raid.mix.hdd.performance` and `raid.mix.hdd.capacity` options, which determine the disk types that are considered to be equivalent.
- Size
- Speed (RPM)
- Checksum type (BCS or AZCS)

Related concepts

How Data ONTAP reports disk types on page 9

What an appropriate hot spare is

If a disk fails and no hot spare disk that exactly matches the failed disk is available, Data ONTAP uses the best available spare. Understanding how Data ONTAP chooses an appropriate spare when there is no matching spare enables you to optimize your spare allocation for your environment.

Data ONTAP picks a non-matching hot spare based on the following criteria:

- If the available hot spares are not the correct size, Data ONTAP uses one that is the next size up, if there is one.
  The replacement disk is downsized to match the size of the disk it is replacing; the extra capacity is not available.
- If the available hot spares are not the correct speed, Data ONTAP uses one that is a different speed.
  Using drives with different speeds within the same aggregate is not optimal. Replacing a disk with a slower disk can cause performance degradation, and replacing a disk with a faster disk is not cost-effective.

If no spare exists with an equivalent disk type or checksum type, the RAID group that contains the failed disk goes into degraded mode; Data ONTAP does not combine effective disk types or checksum types within a RAID group.

Related concepts

How Data ONTAP reports disk types on page 9

About degraded mode

When a disk fails, Data ONTAP can continue to serve data, but it must reconstruct the data from the failed disk using RAID parity. When this happens, the affected RAID group is said to be in *degraded mode*. The performance of a storage system with one or more RAID groups in degraded mode is decreased.

A RAID group goes into degraded mode in the following scenarios:

- A single disk fails in a RAID4 group.
  After the failed disk is reconstructed to a spare, the RAID group returns to normal mode.
- One or two disks fail in a RAID-DP group.
  If two disks have failed in a RAID-DP group, the RAID group goes into *double-degraded mode*.
- A disk is taken offline by Data ONTAP.
  After the offline disk is brought back online, the RAID group returns to normal mode.

Note: If another disk fails in a RAID-DP group in double-degraded mode or a RAID4 group in degraded mode, data loss could occur (unless the data is mirrored). For this reason, always minimize the amount of time a RAID group is in degraded mode by ensuring that appropriate hot spares are available.

Related concepts

How Data ONTAP handles a failed disk that has no available hot spare on page 96

How low spare warnings can help you manage your spare drives

By default, Data ONTAP issues warnings to the console and logs if you have fewer than one hot spare drive that matches the attributes of each drive in your storage system. You can change the threshold value for these warning messages to ensure that your system adheres to best practices.

To make sure that you always have two hot spares for every drive (a best practice), you can set the `min_spare_count` RAID option to `2`.

Setting the `min_spare_count` RAID option to `0` disables low spare warnings. You might want to do this if you do not have enough drives to provide hot spares (for example, if your storage system does not support external disk shelves). You can disable the warnings only if the following requirements are met:

- Your system has 16 or fewer drives.
- You have no RAID groups that use RAID4.

  Note: You cannot create aggregates that use RAID4 protection while the `raid.min_spare_count` option is set to `0`. If either of these requirements is no longer met after this option has been set to `0`, the option is automatically set back to `1`.

How Data ONTAP handles a failed disk with a hot spare

Using an available matching hot spare, Data ONTAP can use RAID to reconstruct the missing data from the failed disk onto the hot spare disk with no data service interruption.

If a disk fails and a matching or appropriate spare is available, Data ONTAP performs the following tasks:

- Replaces the failed disk with a hot spare disk.
  If RAID-DP is enabled and a double-disk failure occurs in the RAID group, Data ONTAP replaces each failed disk with a separate spare disk.
- In the background, reconstructs the missing data onto the hot spare disk or disks.

  Note: During reconstruction, the system is in degraded mode, and file service might slow down.
- Logs the activity in the `/etc/messages` file.
- Sends an AutoSupport message.

Attention: Always replace the failed disks with new hot spare disks as soon as possible, so that hot spare disks are always available in the storage system.

Note: If the available spare disks are not the correct size, Data ONTAP chooses a disk of the next larger size and restricts its capacity to match the size of the disk it is replacing.

Related concepts

How Data ONTAP handles a failed disk that has no available hot spare on page 96

How Data ONTAP handles a failed disk that has no available hot spare

When a failed disk has no appropriate hot spare available, Data ONTAP puts the affected RAID group into degraded mode indefinitely and the storage system automatically shuts down within a specified time period.

If the maximum number of disks have failed in a RAID group (two for RAID-DP, one for RAID4), the storage system automatically shuts down in the period of time specified by the `raid.timeout` option. The default timeout value is 24 hours.

To ensure that you are aware of the situation, Data ONTAP sends an AutoSupport message whenever a disk fails. In addition, it logs a warning message in the `/etc/message` file once per hour after a disk fails.

Attention: If a disk fails and no hot spare disk is available, contact technical support.

Related concepts

How Data ONTAP handles a failed disk with a hot spare on page 96
About degraded mode on page 95

Considerations for changing the timeout RAID option

The `raid.timeout` option controls how long a storage system runs after a RAID group goes into degraded mode or the NVRAM battery malfunctions or loses power. You can change the value of this option, but you should understand the implications of doing so.

The purpose for the system shutdown is to avoid data loss, which can happen if an additional disk failure occurs in a RAID group that is already running in degraded mode, or if a stand-alone system encounters a catastrophic error and has to shut down without NVRAM. You can extend the number of hours the system operates in these conditions by increasing the value of this option (the default value is `24`). You can even disable the shutdown by setting the option to `0`, but the longer the system operates with one or both of these conditions, the greater the chance of incurring data loss.

How RAID-level disk scrubs verify data integrity

RAID-level scrubbing means checking the disk blocks of all disks in use in aggregates (or in a particular aggregate, plex, or RAID group) for media errors and parity consistency. If Data ONTAP finds media errors or inconsistencies, it uses RAID to reconstruct the data from other disks and rewrites the data.

RAID-level scrubs help improve data availability by uncovering and fixing media and checksum errors while the RAID group is in a normal state (for RAID-DP, RAID-level scrubs can also be performed when the RAID group has a single-disk failure).

RAID-level scrubs can be scheduled or run manually.

How you schedule automatic RAID-level scrubs

By default, Data ONTAP performs a weekly RAID-level scrub starting on Sunday at 1:00 a.m. for a duration of six hours. You can change the start time and duration of the weekly scrub, or add more automatic scrubs.

To schedule an automatic RAID-level scrub, you use the `raid.scrub.schedule` option.

To change the duration of automatic RAID-level scrubbing without changing the start time, you use the `raid.scrub.duration` option, specifying the number of minutes you want automatic RAID-level scrubs to run. If you set this option to `-1`, all automatic RAID-level scrubs run to completion.

Note: If you specify a duration using the `raid.scrub.schedule` option, that value overrides the value you specify with the `raid.scrub.duration` option.

Scheduling example

The following command schedules two weekly RAID scrubs. The first scrub is for 240 minutes (four hours) every Tuesday starting at 2 a.m. The second scrub is for eight hours every Saturday starting at 10 p.m.

```
storage raid-options modify -node nodename -name raid.scrub.schedule -
value 240m@tue@2,8h@sat@22
```

Verification example

The following command displays your current RAID-level automatic scrub schedule.

```
storage raid-options show raid.scrub.schedule
```

Reverting to the default schedule example

The following command reverts your automatic RAID-level scrub schedule to the default (Sunday at 1:00 a.m., for six hours):

```
storage raid-options modify -node nodename -name raid.scrub.schedule -
value ""
```

How you run a manual RAID-level scrub

You can manually run a RAID-level scrub on individual RAID groups, plexes, aggregates, or all aggregates using the `storage aggregate scrub` command. You can also stop, suspend, and resume manual RAID-level scrubs.

If you try to run a RAID-level scrub on a RAID group that is not in a normal state (for example, a group that is reconstructing or degraded), the scrub returns errors and does not check that RAID group. You can run a RAID-level scrub on a RAID-DP group with one failed disk.

Customizing the size of your RAID groups

You can customize the size of your RAID groups based on your requirements for data availability, performance, and disk utilization.

About this task

For standard aggregates, you change the size of RAID groups on a per-aggregate basis. For Flash Pool aggregates, you can change the RAID group size for the SSD RAID groups and the HDD RAID groups independently. You cannot change the size of individual RAID groups.

The following list outlines some facts about changing the RAID group size:

- If you increase the RAID group size, more disks or array LUNs will be added to the most recently created RAID group until it reaches the new size.

- All other existing RAID groups in that aggregate remain the same size, unless you explicitly add disks to them.
- You cannot decrease the size of already created RAID groups.
- The new size applies to all subsequently created RAID groups in that aggregate (or, in the case of a Flash Pool aggregate, all subsequently created RAID groups for the affected RAID group type —SSD or HDD).

Step

1. Use the applicable command:

If you want to...	Enter the following command...
Change the RAID group size for the SSD RAID groups of a Flash Pool aggregate	**storage aggregate modify -aggregate *aggr_name* -cache-raid-group-size *size***
Change the size of any other RAID groups	**storage aggregate modify -aggregate *aggr_name* -maxraidsize *size***

Examples

The following command changes the maximum RAID group size of the aggregate n1_a4 to 20 disks or array LUNs:

```
storage aggregate modify -aggregate n1_a4 -maxraidsize 20
```

The following command changes the maximum RAID group size of the SSD cache RAID groups of the Flash Pool aggregate n1_cache_a2 to 24:

```
storage aggregate modify -aggregate n1_cache_a2 -cache-raid-group-size
24
```

Controlling the impact of RAID operations on system performance

You can reduce the impact of RAID operations on system performance by decreasing the speed of the RAID operations.

You can control the speed of the following RAID operations with RAID options:

- RAID data reconstruction
- Disk scrubbing

The speed that you select for each of these operations might affect the overall performance of the storage system. However, if the operation is already running at the maximum speed possible and it is fully utilizing one of the three system resources (the CPU, disks, or the disk-to-controller connection

bandwidth), changing the speed of the operation has no effect on the performance of the operation or the storage system.

If the operation is not yet running, you can set a speed that minimally slows storage system network operations or a speed that severely slows storage system network operations. For each operation, use the following guidelines:

- If you want to reduce the performance impact on client access to the storage system, change the specific RAID option from medium to low. Doing so also causes the operation to slow down.
- If you want to speed up the operation, change the RAID option from medium to high. Doing so might decrease the performance of the storage system in response to client access.

Controlling the performance impact of RAID data reconstruction

Because RAID data reconstruction consumes CPU resources, increasing the speed of data reconstruction sometimes slows storage system network and disk operations. You can control the speed of data reconstruction with the `raid.reconstruc.perf_impact` option.

About this task

When RAID data reconstruction and plex resynchronization are running at the same time, Data ONTAP limits the combined resource utilization to the greater impact set by either operation. For example, if `raid.resync.perf_impact` is set to `medium` and `raid.reconstruct.perf_impact` is set to `low`, the resource utilization of both operations has a medium impact.

The setting for this option also controls the speed of Rapid RAID Recovery.

Step

1. Enter the following command:

   `storage raid-options modify -node` ***`node_name`*** **`raid.reconstruct.perf_impact`** ***`impact`***

   `impact` can be `high`, `medium`, or `low`.

   `high` means that the storage system uses most of the system resources available for RAID data reconstruction; this setting can heavily affect storage system performance, but reconstruction finishes sooner, reducing the time that the RAID group is in degraded mode.

   `low` means that the storage system uses very little of the system resources; this setting lightly affects storage system performance. However, reconstruction takes longer to complete, increasing the time that the storage system is running in degraded mode.

   The default impact is `medium`.

Controlling the performance impact of RAID-level scrubbing

When Data ONTAP performs a RAID-level scrub, it checks the disk blocks of all disks on the storage system for media errors and parity consistency. You can control the impact this operation has on system performance with the `raid.verify.perf_impact` option.

About this task

When RAID-level scrubbing and mirror verification are running at the same time, Data ONTAP limits the combined resource utilization to the greater impact set by either operation. For example, if `raid.verify.perf_impact` is set to `medium` and `raid.scrub.perf_impact` is set to `low`, the resource utilization by both operations has a medium impact.

If there are times during the day when the load on your storage system is decreased, you can also limit the performance impact of the automatic RAID-level scrub by changing the start time or duration of the automatic scrub.

Step

1. Enter the following command:

   storage raid-options modify -node *node_name* raid.scrub.perf_impact *impact*

   impact can be `high`, `medium`, or `low`.

   `high` means that the storage system uses most of the system resources available for scrubbing; this setting can heavily affect storage system performance, but the scrub finishes sooner.

   `low` means that the storage system uses very little of the system resources; this setting lightly affects storage system performance, but the scrub takes longer to complete.

   The default impact is `low`.

What aggregates are

To support the differing security, backup, performance, and data sharing needs of your users, you can group the physical data storage resources on your storage system into one or more aggregates. You can then design and configure these aggregates to provide the appropriate level of performance and redundancy.

Each aggregate has its own RAID configuration, plex structure, and set of assigned drives or array LUNs. The aggregate provides storage, based on its configuration, to its associated FlexVol volumes or Infinite Volume.

Aggregates have the following characteristics:

- They can be composed of drives or array LUNs.
- They can be in 64-bit or 32-bit format.
- If they are composed of drives, they can be single-tier (composed of only HDDs or only SSDs) or they can be Flash Pool aggregates, which include both HDD RAID groups and an SSD cache.

The cluster administrator can assign one or more aggregates to a Storage Virtual Machine (SVM), in which case you can use only those aggregates to contain volumes for that SVM.

For information about best practices for working with aggregates, see *Technical Report 3437: Storage Subsystem Resiliency Guide*.

Related information

Technical Report 3437: Storage Subsystem Resiliency Guide

How the SVM affects which aggregates can be associated with a FlexVol volume

FlexVol volumes are always associated with one Storage Virtual Machine (SVM), and one aggregate that supplies its storage. The SVM can limit which aggregates can be associated with that volume, depending on how the SVM is configured.

When you create a FlexVol volume, you specify which SVM the volume will be created on, and which aggregate that volume will get its storage from. All of the storage for the newly created FlexVol volume comes from that associated aggregate.

If the SVM for that volume has aggregates assigned to it, then you can use only one of those assigned aggregates to provide storage to volumes on that SVM. This can help you ensure that your SVMs are not sharing physical storage resources inappropriately. This segregation can be important in a multi-tenancy environment, because for some space management configurations, volumes that share the same aggregate can affect each other's access to free space when space is constrained for the

aggregate. Aggregate assignment requirements apply to both cluster administrators and SVM administrators.

Volume move and volume copy operations are not constrained by the SVM aggregate assignments, so if you are trying to keep your SVMs on separate aggregates, you must ensure that you do not violate your SVM aggregate assignments when you perform those operations.

If the SVM for that volume has no aggregates assigned to it, then the cluster administrator can use any aggregate in the cluster to provide storage to the new volume. However, the SVM administrator cannot create volumes for SVMs with no assigned aggregates. For this reason, if you want your SVM administrator to be able to create volumes for a specific SVM, then you must assign aggregates to that SVM (`vserver modify -aggr-list`).

Changing the aggregates assigned to an SVM does not affect any existing volumes. For this reason, the list of aggregates assigned to an SVM cannot be used to determine the aggregates associated with volumes for that SVM.

For more information about configuring and managing SVMs, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.

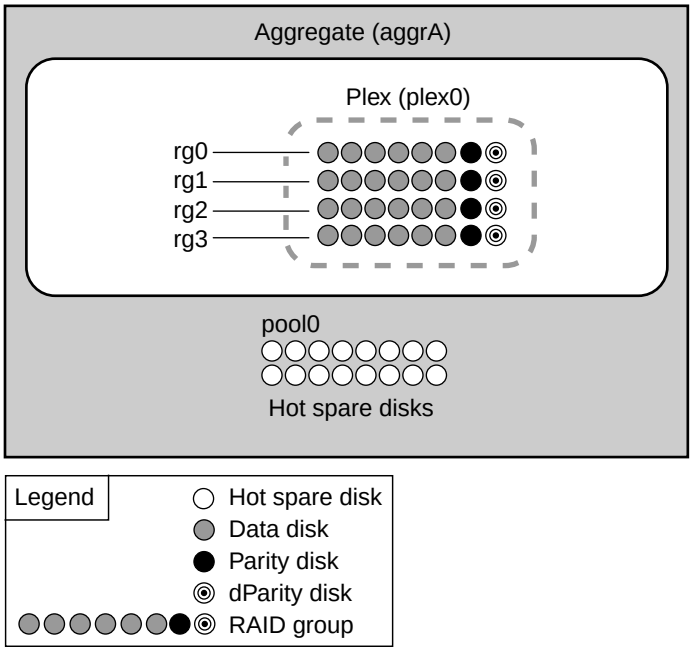
Related tasks

Assigning aggregates to SVMs on page 144

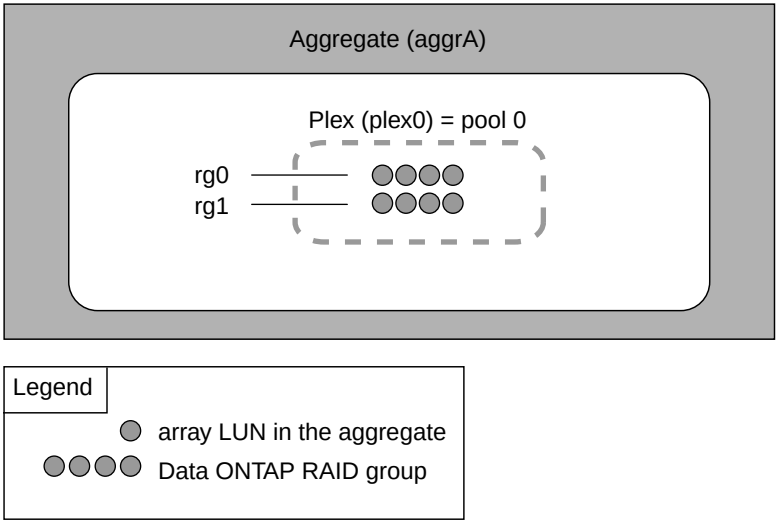
How aggregates work

Aggregates have a single copy of their data, or *plex*, which contains all of the RAID groups belonging to that aggregate. Mirrored aggregates, which have two plexes, are not currently supported in clustered Data ONTAP.

The following diagram shows an unmirrored aggregate with disks, with its one plex:



The following diagram shows an unmirrored aggregate with array LUNs, with its one plex:



Introduction to 64-bit and 32-bit aggregate formats

Aggregates are either 64-bit or 32-bit format. 64-bit aggregates have much larger size limits than 32-bit aggregates. 64-bit and 32-bit aggregates can coexist on the same storage system or cluster.

32-bit aggregates have a maximum size of 16 TB; 64-bit aggregates' maximum size depends on the storage system model. For the maximum 64-bit aggregate size of your storage system model, see the *Hardware Universe* at *hwu.netapp.com*.

When you create a new aggregate, it is a 64-bit format aggregate.

You can expand 32-bit aggregates to 64-bit aggregates by increasing their size beyond 16 TB. 64-bit aggregates, including aggregates that were previously expanded, cannot be converted to 32-bit aggregates.

You can see whether an aggregate is a 32-bit aggregate or a 64-bit aggregate by using the `storage aggregate show -fields block-type` command.

Related tasks

Increasing the size of an aggregate on page 130

What a Flash Pool aggregate is

A Flash Pool aggregate combines both SSDs and HDDs (performance or capacity) to provide a high-performance aggregate more economically than an SSD-only aggregate.

The SSDs provide a high-performance cache for the active data set of the data volumes provisioned on the Flash Pool aggregate, offloading random read operations and repetitive random write operations to improve response times and overall throughput for disk I/O-bound data access operations. (Performance is not significantly increased for predominately sequential workloads.)

Related tasks

Creating a Flash Pool aggregate on page 123

How Flash Pool aggregates work

The Flash Pool technology enables you to add one or more RAID groups composed of SSDs to an aggregate that consists of HDD RAID groups.

The SSD cache does not contribute to the size of the aggregate as calculated against the maximum aggregate size. For example, even if an aggregate is at the maximum aggregate size, you can add an SSD RAID group to it. The SSDs *do* count toward the overall (node or HA pair) drive limit.

The HDD RAID groups in a Flash Pool aggregate behave the same as HDD RAID groups in a standard aggregate, following the same rules for mixing disk types, sizes, speeds, and checksums.

For example, you cannot combine performance and capacity disks in the HDD RAID groups of a Flash Pool aggregate.

The checksum type, RAID type, and RAID group size values can be configured for the SSD cache RAID groups and HDD RAID groups independently.

There is a platform-dependent maximum size for the SSD cache. For information about this limit for your platform, see the *Hardware Universe*.

Related concepts

Rules for mixing drive types in Flash Pool aggregates on page 111
Rules for mixing HDD types in aggregates on page 110
How the available Flash Pool cache capacity is calculated on page 108

Related tasks

Changing the RAID type of RAID groups in a Flash Pool aggregate on page 127

Requirements for using Flash Pool aggregates

The Flash Pool technology has some configuration requirements that you should be aware of before planning to use it in your storage architecture.

Flash Pool aggregates cannot be used in the following configurations:

- 32-bit aggregates
- Aggregates composed of array LUNs
- Aggregates that use the ZCS checksum type

You can use Flash Pool aggregates and the Flash Cache module (WAFL external cache) in the same system. However, data stored in a Flash Pool aggregate is not cached in the Flash Cache module. Flash Cache is reserved for data stored in aggregates composed of only HDDs. For more information about Flash Cache and WAFL external cache, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.

You can use data compression on volumes associated with a Flash Pool aggregate. However, compressed blocks are not cached in the Flash Pool cache for either read or write operations.

Read-only volumes, such as SnapMirror or SnapVault destinations, are not cached in the Flash Pool cache.

For a list of the platforms that support Flash Pool aggregates, and for minimum numbers of SSDs, see the *Hardware Universe*.

If you create a Flash Pool aggregate using an aggregate that was created using Data ONTAP 7.1 or earlier, the volumes associated with that Flash Pool aggregate will not support write caching.

For more information about the types of workloads that benefit from using Flash Pool aggregates, see *Technical Report 4070: NetApp Flash Pool Design and Implementation Guide*.

Related information

TR 4070: NetApp Flash Pool Design and Implementation Guide

How Flash Pool aggregates and Flash Cache compare

Both the Flash Pool technology and the family of Flash Cache modules (Flash Cache and Flash Cache 2) provide a high-performance cache to increase storage performance. However, there are differences between the two technologies that you should understand before choosing between them.

You can employ both technologies on the same system. However, data stored in volumes associated with a Flash Pool aggregate (or an SSD aggregate) is not cached by Flash Cache.

Criteria	Flash Pool aggregate	Flash Cache
Scope	A specific aggregate	All aggregates assigned to a node
Caching types supported	Read and write	Read
Cached data availability during and after takeover events	Cached data is available and unaffected by either planned or unplanned takeover events.	Cached data is not available during takeover events. After giveback for a planned takeover, previously cached data that is still valid is re-cached automatically.
PCIe slot on storage controller required?	No	Yes
Supported with array LUNs?	No	Yes
Supported with Storage Encryption?	No	Yes. Data in the cache is not encrypted.

For more information about Flash Cache, see the *Clustered Data ONTAP System Administration Guide for Cluster Administrators*.

About read and write caching for Flash Pool aggregates

The Flash Pool technology provides both read caching and write caching for random I/O workloads. You can configure Flash Pool caching on the volume, but for most workloads, the default caching policies result in optimal performance.

Some volumes cannot be enabled for write caching. When you attempt to use an aggregate associated with one or more of these volumes as a Flash Pool aggregate, you must force the operation. In this case, writes to that volume would not be cached in the SSD cache, but otherwise the Flash Pool aggregate would function normally. You can get more information about why a volume cannot be enabled for write caching by using the `volume show -instance` command.

For more information about read and write caching policies, see *Technical Report 4070: NetApp Flash Pool Design and Implementation Guide*.

Related tasks

Determining and enabling volume write-caching eligibility on page 125

Related information

TR 4070: NetApp Flash Pool Design and Implementation Guide

How the available Flash Pool cache capacity is calculated

Knowing the available cache capacity enables you to determine how many data SSDs you can add before reaching the limit, because Flash Pool cache capacity cannot exceed a platform-dependent limit for the system or HA configuration.

The current cache capacity is the sum of the "used size" capacity (as reported by the `sysconfig -r` command) of all of the data SSDs used in Flash Pool aggregates on the system. Parity SSDs are not included.

For systems in an HA configuration, the cache size limits apply to the HA configuration as a whole, and can be split arbitrarily between the two nodes, provided that the total limit for the HA configuration is not exceeded.

If Flash Cache modules are installed in a system, the available cache capacity for Flash Pool use is the Flash Pool cache capacity limit minus the sum of the Flash Cache module cache installed on the node. (In the unusual case where the size of the Flash Cache modules is not symmetrical between the two nodes in an HA configuration, the available Flash Pool cache capacity is decreased by the size of the larger Flash Cache module.)

For information about cache size limits, see the *Hardware Universe*.

Example calculation with Flash Cache modules

For an HA configuration composed of two storage controllers with a maximum cache capacity of 12 TB and 2 TB of Flash Cache installed on each node, the maximum Flash Pool aggregate cache capacity for the HA pair would be 12 TB minus 2 TB, or 10 TB.

Example calculation with asymmetrically sized Flash Cache modules

For an HA configuration composed of two storage controllers with a maximum cache capacity of 12 TB and 2 TB of Flash Cache installed on one node and 3 TB of Flash Cache installed on the other node, the maximum Flash Pool aggregate cache capacity for the HA pair would be 12 TB minus 3 TB, or 9 TB.

Understanding how Data ONTAP works with heterogeneous storage

When you have disks with different characteristics (type, speed, size, checksum) or have both disks and array LUNs attached to your storage system, you have heterogeneous storage. Understanding how Data ONTAP works with heterogeneous storage helps you ensure that your aggregates and RAID groups follow best practices and provide maximum storage availability.

How you can use disks with mixed speeds in the same aggregate

Whenever possible, you should use disks of the same speed in an aggregate. However, if needed, you can configure Data ONTAP to allow mixed speed aggregates based on the disk class.

To configure Data ONTAP to allow mixed speed aggregates, you use the following RAID options:

- `raid.mix.hdd.rpm.performance`
- `raid.mix.hdd.rpm.capacity`

When these options are set to `on`, Data ONTAP allows mixing speeds for the designated disk class. Performance disk types are FC and SAS; capacity disk types are BSAS, FSAS, MSATA, and ATA.

By default, `raid.mix.hdd.rpm.performance` is set to `off`, and `raid.mix.hdd.rpm.capacity` is set to `on`.

Even if Data ONTAP is not configured to allow mixing speeds, you can still create aggregates out of disks with different speeds by setting the `-allow-mixed` parameter to `true`.

How to control disk selection from heterogeneous storage

When disks with different characteristics coexist on the same node, or when both disks and array LUNs are attached to the same node, the system has heterogeneous storage. When you create an aggregate from heterogeneous storage, you should take steps to ensure that Data ONTAP uses the disks you expect.

If your node has heterogeneous storage and you do not explicitly specify what type of disks to use, Data ONTAP uses the disk type (including array LUNs) with the highest number of available disks. When you create or add storage to an aggregate using heterogeneous storage, you should use one of the following methods to ensure that Data ONTAP selects the correct disks or disk types:

- Through disk attributes:
  - You can specify disk size by using the `-disksize` option.
    Disks within 20% of the specified size are selected.
  - You can specify disk speed by using the `-diskrpm` option.
  - You can specify disk type by using the `-disktype` option.
- Through an explicit disk list.

You can list the names of specific disks you want to use.

Note: For unplanned events such as disk failures, which cause Data ONTAP to add another disk to a RAID group automatically, the best way to ensure that Data ONTAP chooses the best disk for any RAID group on your system is to always have at least one spare (and preferably two) available to match all disk types and sizes in use in your system.

Rules for mixing HDD types in aggregates

You can mix disks from different loops or stacks within the same aggregate. Depending on the value of the `raid.mix.hdd.disktype` RAID options, you can mix certain types of HDDs within the same aggregate, but some disk type combinations are more desirable than others.

When the appropriate `raid.mix.hdd.disktype` option is set to `off`, HDD RAID groups can be composed of only one Data ONTAP disk type. This setting ensures that your aggregates are homogeneous, and requires that you provide sufficient spare disks for every disk type in use in your system.

The default value for the `raid.mix.hdd.disktype.performance` option is `off`, to prevent mixing SAS and FCAL disks.

The default value for the `raid.mix.hdd.disktype.capacity` option is `on`. For this setting, the BSAS, FSAS, and ATA disk types are considered to be equivalent for the purposes of creating and adding to aggregates, and spare management.

To maximize aggregate performance and for easier storage administration, you should avoid mixing FC-connected and SAS-connected disks in the same aggregate. This is because of the performance mismatch between FC-connected storage shelves and SAS-connected storage shelves. When you mix these connection types in the same aggregate, the performance of the aggregate is limited by the presence of the FC-connected storage shelves, even though some of the data is being served from the higher-performing SAS-connected storage shelves.

MSATA disks cannot be mixed with any other disk type in the same aggregate.

Disks using Storage Encryption have a Data ONTAP disk type of SAS. However, they cannot be mixed with any other disk type, including SAS disks that are not using Storage Encryption. If any disks on a storage system use Storage Encryption, all of the disks on the storage system (and its high-availability partner node) must use Storage Encryption.

Note: If you set a `raid.mix.hdd.disktype` option to `off` for a system that already contains aggregates with more than one type of HDD, those aggregates continue to function normally and accept both types of HDDs. However, no other aggregates composed of the specified disk type will accept mixed HDD types as long as that option is set to `off`.

For information about best practices for working with different types of disks, see *Technical Report 3437: Storage Best Practices and Resiliency Guide*.

Related concepts

How Data ONTAP reports disk types on page 9

Related information

TR 3437: Storage Best Practices and Resiliency Guide

Rules for mixing drive types in Flash Pool aggregates

By definition, Flash Pool aggregates contain more than one drive type. However, the HDD RAID groups follow the same drive-type mixing rules as single-tier aggregates. For example, you cannot mix performance and capacity disks in the same Flash Pool aggregate. The SSD cache can contain only SSDs.

Rules for mixing storage in array LUN aggregates

When planning for aggregates, you must consider the rules for mixing storage in aggregates. You cannot mix different storage types or array LUNs from different vendors or vendor families in the same aggregate.

Adding the following to the same aggregate is not supported:

- Array LUNs and disks
- Array LUNs with different checksum types
- Array LUNs from different drive types (for example, FC and SATA) or different speeds
- Array LUNs from different storage array vendors
- Array LUNs from different storage array model families

Note: Storage arrays in the same family share the same performance and failover characteristics. For example, members of the same family all perform active-active failover, or they all perform active-passive failover. More than one factor might be used to determine storage array families. For example, storage arrays with different architectures would be in different families even though other characteristics might be the same.

How the checksum type is determined for array LUN aggregates

Each Data ONTAP aggregate has a checksum type associated with it. The aggregate checksum type is determined by the checksum type of the array LUNs that are added to it.

The checksum type of an aggregate is determined by the checksum type of the first array LUN that is added to the aggregate. The checksum type applies to an entire aggregate (that is, to all volumes in the aggregate). Mixing array LUNs of different checksum types in an aggregate is not supported.

- An array LUN of type *block* must be used with block checksum type aggregates.
- An array LUN of type *zoned* must be used with advanced zoned checksum (AZCS or advanced_zoned) type aggregates.

  Note: Prior to Data ONTAP 8.1.1, zoned checksum array LUNs were used with ZCS (zoned) type aggregates. Starting in 8.1.1, any new aggregates created with zoned checksum array

LUNs are AZCS aggregates. However, you can add zoned checksum array LUNs to existing ZCS aggregates.

Before you add array LUNs to an aggregate, you must know the checksum type of the LUNs you want to add, for the following reasons:

- You cannot add array LUNs of different checksum types to the same aggregate.
- You cannot convert an aggregate from one checksum type to the other.

When you create an aggregate you can specify the number of array LUNs to be added, or you can specify the names of the LUNs to be added. If you want to specify a number of array LUNs to be added to the aggregate, the same number or more array LUNs of that checksum type must be available.

How to determine space usage in an aggregate

You can view space usage by all volumes in one or more aggregates with the `aggregate show-space` command. This helps you see which volumes are consuming the most space in their containing aggregates so that you can take actions to free more space.

The used space in an aggregate is directly affected by the space used in the FlexVol volumes and Infinite Volume constituents it contains. Measures that you take to increase space in a volume also affect space in the aggregate.

When the aggregate is offline, no values are displayed. Only non-zero values are displayed in the command output. However, you can use the `-instance` parameter to display all possible feature rows regardless of whether they are enabled and using any space. A value of `-` indicates that there is no data available to display.

The following rows are included in the `aggregate show-space` command output:

- Volume Footprints

  The total of all volume footprints within the aggregate. It includes all of the space that is used or reserved by all data and metadata of all volumes in the containing aggregate. It is also the amount of space that is freed if all volumes in the containing aggregate are destroyed. Infinite Volume constituents appear in the output of space usage commands as if the constituents were FlexVol volumes.
- Aggregate Metadata

  The total file system metadata required by the aggregate, such as allocation bitmaps and inode files.
- Snapshot Reserve

  The amount of space reserved for aggregate Snapshot copies, based on volume size. It is considered used space and is not available to volume or aggregate data or metadata. The aggregate's Snapshot reserve is set to 0 percent by default.
- Total Used

  The sum of all space used or reserved in the aggregate by volumes, metadata, or Snapshot copies.

There is never a row for Snapshot spill.

The following example shows the `aggregate show-space` command output for an aggregate whose Snapshot reserve was increased to 5%. If the Snapshot reserve was 0, the row would not be displayed.

```
cluster1::> storage aggregate show-space

                        Aggregate : wqa_gx106_aggr1

      Feature                                   Used      Used%
      --------------------------------    ----------     ------
      Volume Footprints                      101.0MB         0%
      Aggregate Metadata                       300KB         0%
      Snapshot Reserve                        5.98GB         5%

      Total Used                              6.07GB         5%
```

How you can determine and control a volume's space usage in the aggregate

You can determine which FlexVol volumes and Infinite Volume constituents are using the most space in the aggregate and specifically which features within the volume. The `volume show-footprint` command provides information about a volume's footprint, or its space usage within the containing aggregate.

The `volume show-footprint` command shows details about the space usage of each volume in an aggregate, including offline volumes. This command does not directly correspond to the output of the `df` command, but instead bridges the gap between the output of `volume show-space` and `aggregate show-space` commands. All percentages are calculated as a percent of aggregate size.

Only non-zero values are displayed in the command output. However, you can use the `-instance` parameter to display all possible feature rows regardless of whether they are enabled and using any space. A value of `-` indicates that there is no data available to display.

Infinite Volume constituents appear in the output of space usage commands as if the constituents were FlexVol volumes.

The following example shows the `volume show-footprint` command output for a volume called testvol:

```
cluster1::> volume show-footprint testvol

      Vserver : thevs
      Volume  : testvol

      Feature                                   Used    Used%
      --------------------------------    ----------    -----
```

```
      Volume Data Footprint                  120.6MB       4%
      Volume Guarantee                        1.88GB      71%
      Flexible Volume Metadata               11.38MB       0%
      Delayed Frees                           1.36MB       0%
      Total Footprint                         2.01GB      76%
```

The following table explains some of the key rows of the output of the `volume show-footprint` command and what you can do to try to decrease space usage by that feature:

Row/feature name	Description/contents of row	Some ways to decrease
Volume Data Footprint	The total amount of space used in the containing aggregate by a volume's data in the active file system and the space used by the volume's Snapshot copies. This row does not include reserved space, so if volumes have reserved files, the volume's total used space in the `volume show-space` command output can exceed the value in this row.	• Deleting data from the volume. • Deleting Snapshot copies from the volume.
Volume Guarantee	The amount of space reserved by the volume in the aggregate for future writes. The amount of space reserved depends on the guarantee type of the volume.	Changing the type of guarantee for the volume to `none`. This row will go to `0`. If you configure your volumes with a volume guarantee of `none`, you should refer to Technical Report 3965 or 3483 for information about how a volume guarantee of `none` can affect storage availability.
Flexible Volume Metadata	The total amount of space used in the aggregate by the volume's metadata files.	No direct method to control.

Row/feature name	Description/contents of row	Some ways to decrease
Delayed Frees	Blocks that Data ONTAP used for performance and cannot be immediately freed. When Data ONTAP frees blocks in a FlexVol volume, this space is not always immediately shown as free in the aggregate because operations to free the space in the aggregate are batched for increased performance. Blocks that are declared free in the FlexVol volume but that are not yet free in the aggregate are called “delayed free blocks” until the associated delayed free blocks are processed. For SnapMirror destinations, this row has a value of `0` and is not displayed.	No direct method to control.
Total Footprint	The total amount of space that the volume uses in the aggregate. It is the sum of all of the rows.	Any of the methods used to decrease space used by a volume.

Related information

Technical Report: Thin Provisioning Deployment and Implementation Guide: media.netapp.com/documents/tr-3965.pdf

Technical Report: Thin Provisioning in a NetApp SAN or IP SAN Enterprise Environment: media.netapp.com/documents/tr3483.pdf

How Infinite Volumes use aggregates

Each Infinite Volume distributes data across multiple aggregates from multiple nodes. By understanding the way that Infinite Volumes use aggregates, you can plan your aggregates in a way that supports the Infinite Volumes that you want.

For more information about Infinite Volumes, see the *Clustered Data ONTAP Infinite Volumes Management Guide*.

Aggregate requirements for Infinite Volumes

The aggregates that are used by an Infinite Volume must be 64-bit aggregates. Aggregates should also be larger than 100 TB with a minimum of 1.1 TB of available space. If the Infinite Volume uses storage classes, the aggregates must also meet the requirements of the storage class.

The cluster that an Infinite Volume is in can contain 32-bit aggregates, but the aggregates that are associated with any Infinite Volume must all be 64-bit aggregates.

If an aggregate has less than 1.1 TB of available space, it is not used by the Storage Virtual Machine (SVM) with Infinite Volume.

If the Infinite Volume uses storage classes, aggregates must meet the requirements of the storage class to be used. For example, if the storage class is designated to use aggregates of type SAS, aggregates created for that storage class must consist entirely of SAS disks.

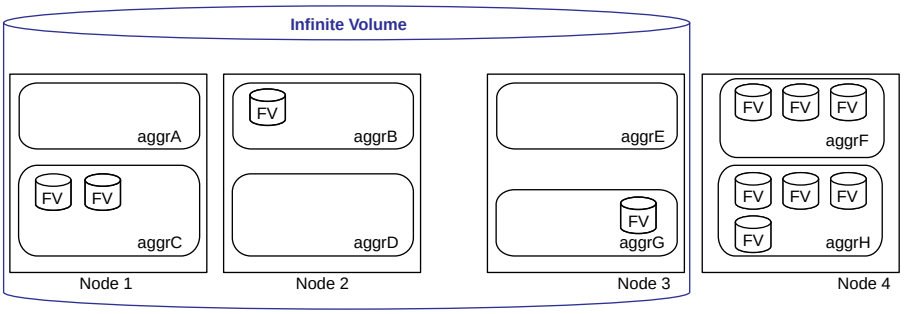
How FlexVol volumes and Infinite Volumes share aggregates

Aggregates can be shared among the volumes in a cluster. Each aggregate can contain multiple FlexVol volumes alongside multiple constituents of Infinite Volumes.

When you create an Infinite Volume, constituents of the Infinite Volume are placed on aggregates that are assigned to its containing Storage Virtual Machine (SVM). If the SVM with Infinite Volume includes aggregates that contain FlexVol volumes, one or more of the Infinite Volume's constituents might be placed on aggregates that already include FlexVol volumes, if those aggregates meet the requirements for hosting Infinite Volumes.

Similarly, when you create a FlexVol volume, you can associate that FlexVol volume with an aggregate that is already being used by an Infinite Volume.

The following diagram illustrates aggregate sharing in a four-node cluster that includes both FlexVol volumes and an Infinite Volume. The Infinite Volume uses the aggregates aggrA, aggrB, aggrC, aggrD, aggrE, and aggrG even though the aggregates aggrB, aggrC, and aggrG already provide storage to FlexVol volumes. (For clarity, the individual constituents that make up the Infinite Volume are not shown.)



How storage classes affect which aggregates can be associated with Infinite Volumes

Each storage class definition specifies an aggregate type. When you create an Infinite Volume with a storage class, only the type of aggregate specified for the storage class can supply storage for the volume. You must understand storage class definitions to create aggregates that are appropriate for the storage class.

Storage class definitions are available only in OnCommand Workflow Automation. After you understand the aggregate requirements for each storage class, you can use the command-line interface or OnCommand Workflow Automation to create aggregates for storage classes. However, you must use OnCommand Workflow Automation, not the command-line interface, to create an Infinite Volume with one or more storage classes.

When you use OnCommand Workflow Automation to create an Infinite Volume with a storage class, OnCommand Workflow Automation automatically filters the aggregates available in the cluster based on the storage class that you want to use. If no aggregates meet the requirements of the storage class, you cannot create an Infinite Volume with that storage class.

How aggregates and nodes are associated with Infinite Volumes

The aggregate list of the containing Storage Virtual Machine (SVM) with Infinite Volume determines which aggregates the Infinite Volume uses, as well as who can create an Infinite Volume and which nodes the Infinite Volume uses.

That aggregate list can be specified or unspecified, which is represented as a dash ("-"). By default, when a cluster administrator creates any SVM, its aggregate list is unspecified. After the SVM is created, the cluster administrator can specify the aggregate list by using the `vserver modify` command with the `-aggr-list` parameter.

Considerations when choosing to specify the aggregate list or leave it unspecified

If you are dedicating an entire cluster to the SVM with Infinite Volume, you can leave the aggregate list of an SVM with Infinite Volume unspecified. In most other situations, you should specify the aggregate list of an SVM with Infinite Volume.

Leaving the aggregate list of an SVM with Infinite Volume unspecified has the following outcomes:

- Only a cluster administrator can create the Infinite Volume, not an SVM administrator.
- When the Infinite Volume is created, it uses all nodes in the cluster.
- When the Infinite Volume is created, it can potentially use all of the aggregates in the cluster.

How the aggregate list contains candidate aggregates

The aggregate list of an SVM with Infinite Volume acts only as a candidate aggregate list for an Infinite Volume. An Infinite Volume uses aggregates according to various factors, including the following requirements:

- When an Infinite Volume is created, at least one data constituent is created on at least one aggregate from each node in the aggregate list.
- An Infinite Volume uses only the aggregates that it requires to meet the capacity requirements for its specified size.

  If the assigned aggregates have far greater capacity than the Infinite Volume requires when it is first created, some aggregates in the aggregate list might not contain any Infinite Volume constituents.

How the aggregate list determines the nodes

An Infinite Volume uses every node that has an aggregate in the aggregate list of an SVM with Infinite Volume.

When changes to the aggregate list take effect

Changes to the aggregate list do not have any immediate effect. The aggregate list is used only when the size of an Infinite Volume changes. For example, if you add an aggregate to the aggregate list of an SVM with Infinite Volume, that aggregate is not used until you modify the size of the Infinite Volume.

If you add aggregates from a new node to the aggregate list and then resize the Infinite Volume, whether the Infinite Volume uses the aggregates from the new node depends on several variables, including the size of existing constituents and how much the Infinite Volume was increased in size.

How the aggregate list can be filtered

You can filter the aggregate list for the SVM by using advanced parameters that control which aggregates are used for each type of constituent, such as data constituents. Unlike the aggregate list for the SVM, these aggregate-selection parameters apply only to a single operation. For example, if you use the parameter for data constituent aggregates when you create the Infinite Volume and then resize the Infinite Volume without using the parameter, the Infinite Volume uses the SVM aggregate list.

How space is allocated inside a new Infinite Volume

Several rules govern how space is allocated to constituents when an Infinite Volume is created. Understanding these rules can help you understand the best practices for configuring aggregates for an Infinite Volume.

The following rules govern how space is allocated to constituents when an Infinite Volume is created:

1. The namespace constituent and its mirror copies are created.

   Before any space is allocated for data, the namespace constituent and namespace mirror constituents are created as big as possible within their maximum sizes.

   a. The namespace constituent is placed on the aggregate with the most available space on any node that the Infinite Volume uses.

b. The first namespace mirror constituent is placed on the aggregate with the next most available space, as long as the aggregate is on a node that meets all of the following conditions:

- The node is used by the Infinite Volume.
- It does not already contain the namespace constituent.
- It is preferably not the partner node in the HA pair of the node that contains the namespace constituent.

c. If SnapDiff is enabled, additional namespace mirror constituents are placed on the aggregate with the most available space on each remaining node used by the Infinite Volume.

2. The data capacity is divided equally among the nodes that the Infinite Volume uses.
The data capacity of an Infinite Volume is balanced across nodes. Data capacity is the space remaining from the Infinite Volume's size after deducting the space required by the namespace-related constituents.

3. Within each node, individual data constituents are made as big as possible within a specified maximum.
Data constituents are always created as big as they are allowed to be within a specified maximum. Each time that Data ONTAP creates a data constituent, it evaluates all of the aggregates that the Infinite Volume uses on the node and selects the aggregate that has the most available space.

Relocating ownership of aggregates used by Infinite Volumes

If you want to relocate ownership of aggregates that are used by an Infinite Volume with SnapDiff enabled, you must ensure that the destination node has a namespace mirror constituent, and you must perform the aggregate reallocation in a specific order.

Before you begin

- You must know whether SnapDiff is enabled on the Infinite Volume.
  If you do not know, you can use the `volume show` command with the `-fields -enable-snapdiff` parameter.
- You must know the names of the aggregates on the source and destination nodes.

About this task

- Follow this procedure only if SnapDiff is enabled on an Infinite Volume that uses the node containing the aggregates that are affected by the ownership change. If SnapDiff is not enabled on the Infinite Volume, do not follow this procedure, because the presence of an Infinite Volume does not affect the aggregate relocation operation.
- For more information about SnapDiff and about how Infinite Volumes use aggregates, see the *Clustered Data ONTAP Infinite Volumes Management Guide*.

Steps

1. Determine whether the destination node is already used by the Infinite Volume by using the `vserver show` command with the `-instance` parameter.

- If the Aggregate List includes an aggregate from the destination node, the Infinite Volume already uses the destination node.
- If the Aggregate List does not include an aggregate from the destination node, the Infinite Volume does not currently use the destination node.

The destination node must have a namespace mirror constituent before you can relocate aggregate ownership.

2. If the destination node is not currently used by the Infinite Volume, add the destination node to the Infinite Volume.

a) Determine the size of the Infinite Volume's namespace constituent by using the `volume show` command with the `-is-constituent true` parameter and identifying the size of the constituents with “ns” in their names.

b) Identify an aggregate on the destination node that has available space to accommodate a namespace mirror constituent by using the `aggregate show` command.

c) Assign the aggregate from the new node to the Infinite Volume by using the `vserver modify` command with the `-aggr-list` parameter.

When you specify the aggregate list, you should include all the existing aggregates from existing nodes as well as the new aggregate from the new node.

d) Increase the size of the Infinite Volume by using the `volume modify` command with the `-size` parameter.

Increase the size of the Infinite Volume by an amount that is equal or larger than the size of the namespace constituent.

A namespace mirror constituent is created on the destination node.

3. Identify the type of Infinite Volume constituents contained by each aggregate on the source node by using the `volume show` command with the `-vserver` and `-is-constituent true` parameters.

Constituents with “data” in their names are data constituents. The constituent with a name ending in “_ns” is the namespace constituent. Constituents with “ns_mirror” in their names are namespace mirror constituents.

Example

In the following output, aggr3 contains a data constituent, aggr1 contains a namespace mirror constituent, and aggr2 contains a namespace mirror constituent:

```
cluster1::> volume show -vserver vs0 -is-constituent true
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
vs0       repo_vol_1024_data0001 aggr3 online  RW        100TB       95TB    5%
vs0       repo_vol_1024_data0002 vs_aggr online RW       100TB       95TB    5%
vs0       repo_vol_1024_data0003 aggr4 online  RW        100TB       95TB    5%
                                      ...
                                      ...
                                      ...
vs0       repo_vol_ns  aggr1        online     RW         10TB      9.5TB    5%
```

```
vs0       repo_vol_ns_mirror0001 aggr2 online  DP         10TB      9.5TB    5%
100 entries were displayed.
```

4. Perform the aggregate relocation in the following way:

a) Divide the aggregates into the following two categories:

- The single aggregate that contains either a namespace constituent or a namespace mirror constituent.

  It might also contain data constituents.
- All the other aggregates on the node that contain data constituents.

b) Relocate ownership of all aggregates that do not contain a namespace constituent or namespace mirror constituent.

> **Note:** If you want to relocate ownership of the aggregate that contains the namespace constituent without also relocating ownership of all of the aggregates that contain data constituents, you must contact technical support to create a namespace mirror constituent on the source node.

c) If the remaining aggregate contains only the namespace constituent or if the remaining aggregate contains more than one type of constituent, relocate ownership of the remaining aggregate.

   If the remaining aggregate contains only a namespace mirror constituent and no data constituents, you do not need to change ownership of the aggregate.

After you finish

You can consider contacting technical support to delete any excess namespace mirror constituents that are using space unnecessarily.

Managing aggregates

You create and manage your aggregates so that they can provide storage to their associated volumes.

Creating an aggregate

You create an aggregate to provide storage to one or more FlexVol volumes and Infinite Volumes. Aggregates are a physical storage object; they are associated with a specific node in the cluster.

Before you begin

You should know what drives or array LUNs will be used in the new aggregate.

If you have multiple drive types in your system (heterogeneous storage), you should understand how you can ensure that the correct drive type is selected.

About this task

Drives and array LUNs are owned by a specific node; when you create an aggregate, all drives in that aggregate must be owned by the same node, which becomes the home node for that aggregate.

Aggregate names must conform to the following requirements:

- Begin with either a letter or an underscore (_).
- Contain only letters, digits, and underscores.
- Contain 250 or fewer characters.

Steps

1. Display a list of available spares by entering the following command:
   `storage disk show -spare -owner` ***`node_name`***
2. Create the aggregate by using the `storage aggregate create` command.

   If you are logged in to the cluster on the cluster management interface, you can create an aggregate on any node in the cluster. To ensure that the aggregate is created on a specific node, use the `-nodes` parameter or specify drives that are owned by that node.

   You can specify the following options:

   - Aggregate's home node (that is, the node on which the aggregate is located unless the aggregate fails over to the node's storage failover partner)
   - List of specific drives or array LUNs that are to be added to the aggregate
   - Number of drives to include
   - Checksum style to use for the aggregate
   - Type of drives to use

- Size of drives to use
- Drive speed to use
- RAID type for RAID groups on the aggregate
- Maximum number of drives or array LUNs that can be included in a RAID group
- Whether drives with different RPM are allowed

For more information about these options, see the `storage aggregate create` man page.

3. Verify the RAID group and drives of your new aggregate by entering the following command:

   `storage aggregate show -aggregate` ***`aggr_name`***

Examples

The following command creates a 64-bit aggregate named aggr2 on node node1 that is composed of 6 SAS (or equivalent) drives. This example accepts the default values for RAID type and RAID group size.

```
storage aggregate create -aggregate aggr2 -node node1 -diskcount 6 -
disktype SAS
```

Creating a Flash Pool aggregate

You create a Flash Pool aggregate by enabling the feature on an existing 64-bit aggregate composed of HDD RAID groups, and then adding one or more SSD RAID groups to that aggregate. This results in two sets of RAID groups for that aggregate: SSD RAID groups (the SSD cache) and HDD RAID groups.

Before you begin

- You must have identified a valid 64-bit aggregate composed of HDDs to convert to a Flash Pool aggregate.
- You must have determined write-caching eligibility of the volumes associated with the aggregate, and completed any required steps to resolve eligibility issues.
- You must have determined the SSDs you will be adding, and these SSDs must be owned by the node on which you are creating the Flash Pool aggregate.
- You must have determined the checksum types of both the SSDs you are adding and the HDDs already in the aggregate.
- You must have determined the number of SSDs you are adding and the optimal RAID group size for the SSD RAID groups.

  Using fewer RAID groups in the SSD cache reduces the number of parity disks required.
- You must have determined the RAID level you want to use for the SSD cache.
- You must have familiarized yourself with the configuration requirements for Flash Pool aggregates.

About this task

After you add an SSD cache to an aggregate to create a Flash Pool aggregate, you cannot remove the SSD cache to convert the aggregate back to its original configuration.

You can change the RAID group size of the SSD cache, but you cannot make this change until after SSDs have been added. After disks have been added to a RAID group, they cannot be removed. If you know that you want to use a different RAID group size than the default SSD RAID group size, you can add a small number SSDs at first. Then, after you update the RAID group size, you can add the rest of the SSDs.

By default, the RAID level of the SSD cache is the same as the RAID level of the HDD RAID groups. You can override this default selection by specifying the `raidtype` option when you add the first SSD RAID groups. Although the SSD cache is providing caching for the HDD RAID groups, the SSD cache is integral to the health of the aggregate as a whole. An SSD RAID group that experiences a failure that exceeds the RAID protection capability of the RAID level in use takes the aggregate offline. For this reason, it is a best practice to keep the RAID level of the SSD cache the same as that of the HDD RAID groups.

There are platform- and workload-specific best practices for Flash Pool SSD cache size and configuration. For information about these best practices, see *Technical Report 4070: NetApp Flash Pool Design and Implementation Guide*.

Steps

1. Mark the aggregate as hybrid-enabled:

   storage aggregate modify -aggregate *aggr_name* -hybrid_enabled true

   If this step does not succeed, determine write-caching eligibility for the target aggregate.

2. Add the SSDs to the aggregate by using the `storage aggregate add-disks` command.

   You can specify the SSDs by ID or by using the `diskcount` and `disktype` parameters. You do not need to specify a new RAID group; Data ONTAP automatically puts the SSDs into their own RAID group.

   If you plan to change the RAID group size for the SSD cache, you should add only a small number SSDs in this step. (You must add at least three.)

   If the HDDs and the SSDs do not have the same checksum type, or if the aggregate is a mixed-checksum aggregate, then you must use the `checksumstyle` parameter to specify the checksum type of the disks you are adding to the aggregate.

   You can specify a different RAID type for the SSD cache by using the `raidtype` option.

3. If you want a different RAID group size for the SSD cache than for the HDD RAID groups, change the SSD RAID group size:

   storage aggregate modify -aggregate *aggr_name* -cache-raid-group-size *size*

4. If you did not add all of the required SSDs in the previous step, add the rest of the SSDs by using the `storage aggregate add-disks` command again.

Related concepts

How Flash Pool aggregates work on page 105
Requirements for using Flash Pool aggregates on page 106

Related tasks

Determining and enabling volume write-caching eligibility on page 125

Related information

TR 4070: NetApp Flash Pool Design and Implementation Guide

Determining and enabling volume write-caching eligibility

Understanding whether the FlexVol volumes associated with an aggregate are eligible for write caching can help you ensure that the volumes with high performance requirements can get the maximum performance improvement from having their associated aggregate converted to a Flash Pool aggregate.

About this task

Flash Pool aggregates employ two types of caching: *read caching* and *write caching*. Read caching is available for all volumes. Write caching is available for most volumes, but might be disabled for some volumes due to an internal ID collision. You can use this procedure to determine write caching eligibility to help you decide which aggregates are good candidates to become Flash Pool aggregates. You do not need any SSDs to complete this procedure.

If an aggregate you want to convert to a Flash Pool aggregate is associated with volumes that are ineligible for write caching, you have two choices:

- If you do not need write caching enabled on those volumes, you can go ahead and convert the aggregate to a Flash Pool aggregate by using the force option.
  You do not get the benefit of the Flash Pool cache for write operations on those volumes.
- If you need write caching enabled on those volumes, you can temporarily move those volumes to another aggregate and move them back, which resolves the ID collision.

Steps

1. Attempt to enable the Flash Pool capability on the aggregate:
   `storage aggregate modify aggr_name -hybrid-enabled true`
2. Take the applicable action based on the result of Step 1:

If...	Then...
The Flash Pool capability is successfully enabled	Disable the Flash Pool capability again: **`storage aggregate modify aggr_name -hybrid-enabled false`** You have completed this task. All of the volumes in the aggregate are eligible for write caching.
Data ONTAP displays an error message telling you that the aggregate cannot be converted to a Flash Pool aggregate	Determine which volumes are not eligible: **`volume show -volume * -fields hybrid-cache-write-caching-ineligibility-reason -aggregate aggr_name`** Each volume in the aggregate is listed, along with its reason for ineligibility if it is ineligible. Eligible volumes display a hyphen ("-").

3. Your next steps depend on your requirements for the ineligible volumes:

If you...	Then...
Do not need write caching enabled on the ineligible volumes	You have completed this task. You must use the force option when you convert the aggregate to a Flash Pool aggregate.
Need write caching enabled on the ineligible volumes	You must move (or copy and delete) all but one of each set of volumes with the same conflicting ID to another aggregate and then move them back until no more volumes show an ID conflict.

Example with ID collisions

The following example shows the system output when there are ID collisions:

```
clus1::> vol show -volume * -fields hybrid-cache-write-caching-
ineligibility-reason  -aggregate aggr1
  (volume show)
vserver volume   hybrid-cache-write-caching-ineligibility-reason
------- -------- -----------------------------------------------
vs0     root_vs0 -
vs0     vol1     -
vs0     vol2     "ID Collision(27216)"
vs0     vol3     "ID Collision(27216)"
4 entries were displayed.
```

Changing the RAID type of RAID groups in a Flash Pool aggregate

The SSD cache of a Flash Pool aggregate can have a different RAID type than the HDD RAID groups. You can change the RAID type of the SSD cache or HDD RAID groups independently of one another. All of the HDD RAID groups must have the same RAID type.

About this task

If the SSD cache RAID group goes into a failed state, the Flash Pool aggregate goes offline, just as it would if an HDD RAID group goes into a failed state. For this reason, you should use RAID-DP as the RAID type for the SSD cache whenever possible, and adhere to good hot spare practices for the SSD cache.

If the SSD cache has a different RAID type than the HDD RAID groups, the Flash Pool aggregate is considered to have a mixed RAID type, displayed as `mixed_raid_type` for the aggregate.

Steps

1. Change the RAID type of the SSD cache or HDD RAID groups of the Flash Pool aggregate:

   `storage aggregate modify -aggregate` ***`aggr_name`*** **`-raidtype`** ***`raid_type`*** **`-disktype`** ***`disk_type`***

   To change the RAID type of the SSD cache, use **`-disktype SSD`**. To change the RAID type of the HDD RAID groups, specify any disk type included in the HDD RAID groups.

2. Verify the RAID groups in your Flash Pool aggregate:

   `storage aggregate show -aggregate` ***`aggr_name`***

   You also can use the `aggr status -r` command to obtain more details about the RAID types of the HDD RAID groups and SSD cache of the Flash Pool aggregate.

Example

In this example, the HDD RAID groups and SSD cache of a Flash Pool aggregate named “test” initially have a RAID type of RAID4. The following command changes the RAID type of the SSD cache to RAID-DP, and converts the Flash Pool aggregate to the mixed RAID type:

`storage aggregate modify -aggregate test -raidtype raid_dp -disktype SSD`

The output from the `aggr status -r` command shows that the aggregate has a mixed RAID type, the HDD RAID groups have a RAID type of RAID4, and the SSD cache has a RAID type of RAID-DP.

```
run local aggr status -r test

Aggregate test (online, mixed_raid_type, hybrid) (block checksums)
  Plex /test/plex0 (online, normal, active)
    RAID group /test/plex0/rg0 (normal, block checksums, raid4)

RAID Disk Device   HA SHELF BAY CHAN Pool Type RPM  Used (MB/blks)    Phys (MB/blks)
--------- ------   -- ----- --- ---- ---- ---- ---  --------------    --------------
parity    6b.43.1  6b  43   1   SA:B -    BSAS 7200 847555/1735794176 847884/1736466816
data      6b.43.2  6b  43   2   SA:A -    BSAS 7200 847555/1735794176 847884/1736466816
data      6b.43.3  6b  43   3   SA:B -    BSAS 7200 847555/1735794176 847884/1736466816
data      6b.43.6  6b  43   6   SA:A -    BSAS 7200 847555/1735794176 847884/1736466816
data      6b.43.9  6b  43   9   SA:B -    BSAS 7200 847555/1735794176 847884/1736466816
data      6b.43.10 6b  43  10   SA:A -    BSAS 7200 847555/1735794176 847884/1736466816

    RAID group /test/plex0/rg1 (normal, block checksums, raid_dp)

RAID Disk Device   HA SHELF BAY CHAN Pool Type RPM  Used (MB/blks)    Phys (MB/blks)
--------- ------   -- ----- --- ---- ---- ---- ---  --------------    --------------
dparity   6b.43.15 6b  43   15  SA:B -    SSD  N/A  84574/173208064   84796/173663616
parity    6b.43.4  6b  43   4   SA:A -    SSD  N/A  84574/173208064   84796/173663616
data      6b.43.13 6b  43   13  SA:B -    SSD  N/A  84574/173208064   84796/173663616
data      6b.43.14 6b  43   14  SA:A -    SSD  N/A  84574/173208064   84796/173663616
```

Using the Automated Workflow Analyzer (AWA) feature to optimize Flash Pool cache size

The AWA feature analyzes the read-write workload mix and the percentage of reads and writes that are cacheable for an aggregate. This shows whether an HDD-only aggregate is a good candidate to be converted to a Flash Pool aggregate, and what an optimal cache size would be for your workload.

Before you begin

You should know approximately when the aggregate you are analyzing experiences its peak load.

About this task

The AWA feature collects data about the I/O workloads for all volumes associated with an aggregate over a period of time, and then uses that data to provide information about the effectiveness of various sizes of Flash Pool cache for that workload. Because AWA uses the highest load it sees during the data collection period, you should ensure that the data collection period includes at least one interval of peak load, but you do not need to ensure that the load is high for the entire data collection period.

AWA analyzes data up to one week in duration. Running AWA for more than one week causes the data collected earlier to be overwritten. Cache size estimates are based on the highest loads seen during the data collection period.

AWA can be run on HDD-only aggregates or on Flash Pool aggregates. For Flash Pool aggregates, AWA tells you whether the current cache size is optimal.

You should use the AWA results in concert with best practices and guidelines provided from all Flash Pool documentation.

Steps

1. Enter advanced mode:

   `set advanced`

2. Start AWA:

   `system node run -node` ***`node_name`*** **`wafl awa start`** ***`aggr_name`***

   AWA begins collecting workload data for the volumes associated with the specified aggregate.

3. Exit advanced mode:

   `set admin`

4. Allow AWA to run until one or more intervals of peak load have occurred, but no longer than one week.

   AWA collects workload statistics for the volumes associated with the specified aggregate.

5. Enter advanced mode:

   `set advanced`

6. Display the workload analysis:

   `system node run -node` ***`node_name`*** **`wafl awa print`**

   AWA displays the workload statistics and optimal Flash Pool cache size.

7. Stop AWA:

   `system node run -node` ***`node_name`*** **`wafl awa stop`**

   All workload data is flushed and is no longer available for analysis.

8. Exit advanced mode:

   `set admin`

Example

In the following example, AWA was run on an aggregate “aggr1”. Here is the output of the `awa print` command after AWA had been running for about 3 days (442 10-minute intervals):

```
Basic Information

                Aggregate aggr1
             Current-time Sun Dec  8 14:42:48 GMT 2013
               Start-time Thu Dec  5 13:11:26 GMT 2013
      Total runtime (sec) 264682
    Interval length (sec) 600
```

```
          Total intervals 442
        In-core Intervals 1024

Summary of the past 1023 intervals
                                    max
          Read Throughput        1.076 MB/s
         Write Throughput        5.777 MB/s
       Cacheable Read (%)       99.988 %
      Cacheable Write (%)       85.758 %
Max Projected Cache Size        50.459 GiB
   Projected Read Offload       99.000 %
  Projected Write Offload       87.000 %

Summary Cache Hit Rate vs. Cache Size

       Size        20%        40%        60%        80%       100%
   Read Hit     25.000     46.000     76.000     99.000     99.000
  Write Hit     53.000     64.000     77.000     87.000     87.000
```

The results provide the following pieces of information:

- Read Throughput and Write Throughput

  Approximate read and write throughput. For this example, the aggregate experienced a read throughput of 1.076 MB per second and a write throughput of 5.777 MB per second.
- Projected Cache Size

  The approximate optimal Flash Pool cache size for the captured workload. For this example, the suggested Flash Pool cache size is approximately 50 GB.
- Projected Read Offload and Projected Write Offload

  The approximate percentages of read and write operations that would have been handled by a Flash Pool cache of the optimal size rather than going to disk (projected cache hit rate). Note that this number is *not* a prediction of the performance increase you would see by converting the aggregate to a Flash Pool aggregate.
- Summary Cache Hit Rate vs. Cache Size

  The approximate cache hit rate for various smaller cache sizes

Increasing the size of an aggregate

You can add disks or array LUNs to an aggregate so that it can provide more storage to its associated volumes. If you need to add enough storage to a 32-bit aggregate to increase its size beyond 16 TB, you can do so; this operation expands the aggregate to 64-bit format.

Before you begin

You must understand the following concepts:

- The requirement to add disks or array LUNs owned by the same system
- For aggregates composed of disks, you must understand the following:

- Benefits of keeping your RAID groups homogeneous for disk size and speed
- Which types of disks can be used together
- Checksum rules when disks of more than one checksum type are in use
- How to ensure that the correct disks are added to the aggregate (the disk addition operation cannot be undone)
- How to add disks to aggregates from heterogeneous storage
- Minimum number of disks to add for best performance
- Number of hot spares you need to provide for protection against disk failures
- Requirements for adding disks from multi-disk carrier disk shelves

About this task

When you add HDDs to an aggregate, you should add a complete RAID group. For information about adding SSDs to a Flash Pool aggregate, see *Technical Report 4070: NetApp Flash Pool Design and Implementation Guide*.

Steps

1. Verify that appropriate spare disks or array LUNs are available for you to add by entering the following command:

   storage disk show -spare -owner *node_name*

   For disks, make sure that enough of the spares listed are of the correct type, size, speed, and checksum type for the target RAID group in the aggregate to which you are adding the disks.

2. Add the disks or array LUNs by entering the following command:

   storage aggregate add-disks -aggregate *aggr_name* [-raidgroup *raid_group_name*] *disks*

   If you are adding disks with a different checksum than the aggregate, as when creating a Flash Pool aggregate, or if you are adding disks to a mixed checksum aggregate, you must either specify the disks to be added with a disk list or use the `-checksumstyle` parameter.

   If you are adding disks to a Flash Pool aggregate, you must either specify the disks to be added with a disk list or use the `-disktype` parameter to specify the disk type.

   If you specify the `-raidgroup` parameter, the storage is added to the RAID group you specify. *`raid_group_name`* is the name that Data ONTAP gave to the group—for example, rg0. If you are adding SSDs to the SSD cache of a Flash Pool aggregate, you do not need to specify the RAID group name; the SSD RAID group is selected by default based on the type of the disks you are adding.

   `disks` specifies the disks to be added in one of the following ways:

   - `-diskcount`, usually further qualified by disk type or checksum type
   - `-disklist` *`disk1`* [*`disk2`*...]

3. If the previous step was unsuccessful because you are adding disks to a 32-bit aggregate and the additional disks would cause its size to exceed 16 TB, complete the following steps to expand the aggregate to 64-bit:

   a) Repeat the `storage aggregate add-disks` command you entered before, with the `-64bit-upgrade normal` parameter added.

      Example

      For example, if you entered the **`storage aggregate add-disks -diskcount 10 -disktype SAS`** command, you would enter the following command:

      `storage aggregate add-disks -diskcount 10 -disktype SAS -64bit-upgrade normal`

      Data ONTAP checks each volume associated with the aggregate to ensure that it has enough free space to be expanded to 64-bit. If all of the volumes have enough free space, the disks are added and the aggregate is expanded to the 64-bit format. If any of the volumes are too full to be expanded, the command fails.

   b) If the previous command failed, run the command again, replacing the `-64-bit-upgrade normal` parameter with the `-64-bit-upgrade check` parameter and following the instructions in the output of that command.

   c) If you had to add more space to any volume, repeat the `storage aggregate -add-disks` command again, this time with the `-64bit-upgrade normal` parameter.

   d) If you want to ensure that the disk usage quota accounting for this aggregate is exactly correct, reinitialize quotas on all of its volumes.

      If you do not reinitialize quotas, quotas on volumes associated with this aggregate will remain active, but the disk usage accounting will be slightly lower than the actual usage until the next time quotas are reinitialized.

Related concepts

Best practices for expanding a 32-bit aggregate to 64-bit on page 133
How to control disk selection from heterogeneous storage on page 109

Related references

Storage limits on page 148

Related information

TR 4070: NetApp Flash Pool Design and Implementation Guide

What happens when you add storage to an aggregate

By default, Data ONTAP adds new drives or array LUNs to the most recently created RAID group until it reaches its maximum size. Then Data ONTAP creates a new RAID group. Alternatively, you can specify a RAID group that you want to add storage to.

When you create an aggregate or add storage to an aggregate, Data ONTAP creates new RAID groups as each RAID group is filled with its maximum number of drives or array LUNs. The last RAID group formed might contain fewer drives or array LUNs than the maximum RAID group size for the aggregate. In that case, any storage added to the aggregate is added to the last RAID group until the specified RAID group size is reached.

If you increase the RAID group size for an aggregate, new drives or array LUNs are added only to the most recently created RAID group; the previously created RAID groups remain at their current size unless you explicitly add storage to them.

If you add a drive to a RAID group that is larger than the drives already there, the new drive is capacity-limited to be the same size as the other drives.

Note: You are advised to keep your RAID groups homogeneous when possible. If needed, you can replace a mismatched drive with a more suitable drive later.

Best practices for expanding a 32-bit aggregate to 64-bit

You should be aware of certain best practices before expanding an aggregate from 32-bit to 64-bit format.

Following these suggestions ensures a smooth expansion operation:

- If the aggregate you are adding storage to contains FlexCache volumes, destroy the FlexCache volumes before initiating the expansion and re-create them after the operation is complete.
- If you are expanding aggregates that contain volumes in a SnapMirror relationship, expand the aggregate containing the source volume first whenever possible.

  Otherwise, expand the source aggregate as soon as possible after expanding the destination aggregate.
- If you are creating a FlexClone volume from a SnapMirror destination volume and you are expanding the aggregates containing the source and destination volumes, expand both source and destination, and use a base Snapshot copy that was created after the source volume was expanded.
- When you add storage to any aggregate, add an entire RAID group at a time to keep the size of your RAID groups homogeneous.

For more information about expanding a 32-bit aggregate to 64-bit, see TR-3978, *In-Place Expansion of 32-bit Aggregates to 64-bit Overview and Best Practices*.

Related information

TR 3978: In-Place Expansion of 32-bit Aggregates to 64-bit Overview and Best Practices

Expanding an aggregate to 64-bit without adding storage

You can expand a 32-bit aggregate to the 64-bit format without adding storage to it. This enables you to use capabilities that are supported only on 64-bit aggregates, and helps you remove 32-bit data from your storage system.

Before you begin

- If the aggregate contains destination volumes for a SnapMirror relationship with a 32-bit source volume, the aggregate containing the source volume must be expanded before expanding the aggregate containing the destination volume.
- All FlexCache volumes contained by the aggregate to be expanded must be destroyed before you initiate the expansion.
  They can be re-created after the expansion is complete.

About this task

For volumes in a SnapMirror relationship, the destination volume inherits the format of the source volume while the mirror is intact. If the aggregate you are expanding contains a destination volume whose source is a 32-bit volume and you break the mirror before expanding the aggregate, the destination volume will be expanded to the 64-bit format. However, if you reestablish the mirror and the source volume is still 32-bit, the destination volume returns to the 32-bit format. For this reason, you must expand the aggregate containing the source volume before reestablishing the SnapMirror relationship if you want to expand all 32-bit volumes in the aggregate to the 64-bit format.

Steps

1. Initiate the expansion by entering the following command:

   `storage aggregate 64bit-upgrade start -aggregate` ***`aggr_name`***

2. Depending on the result of the preceding step, take the appropriate action:

If the 64bit-upgrade command...	Then...
Initiates successfully	Proceed to the next step.
Indicates that one or more volumes could not be expanded because they did not have enough space	Retry the 64bit-upgrade command, adding the `grow-all` option.
Indicates that the expansion could not be completed for some other reason	Take the appropriate action, based on the issue outlined in the error message.

3. Display the status of the expansion by entering the following command:

   `storage aggregate 64bit-upgrade status -aggregate` ***`aggr_name`***

   The current status of the expansion is displayed. When the message that there is no upgrade in progress is displayed, the expansion is complete.

4. Optional: Confirm that all volumes in the aggregate are 64-bit format by entering the following command:

   `volume show -aggregate *aggr_name* -fields block-type`

Relocating aggregate ownership within an HA pair

You can change the ownership of aggregates among the nodes in an HA pair without interrupting service from the aggregates.

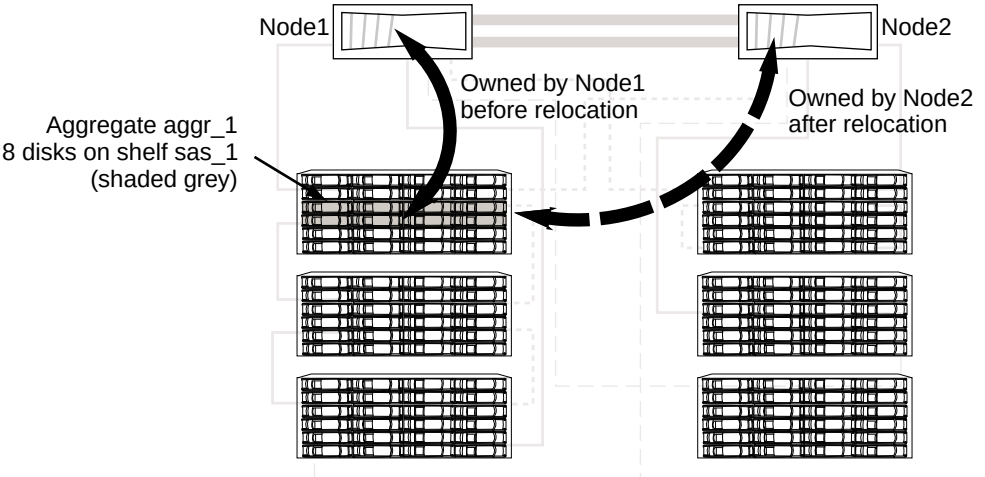
Both nodes in an HA pair are physically connected to each other's disks or array LUNs. Each disk or array LUN is owned by one of the nodes. While ownership of disks temporarily changes when a takeover occurs, the aggregate relocation operations either permanently (for example, if done for load balancing) or temporarily (for example, if done as part of takeover) change the ownership of all disks or array LUNs within an aggregate from one node to the other. The ownership changes without any data-copy processes or physical movement of the disks or array LUNs.

How aggregate relocation works

Aggregate relocation takes advantage of the HA configuration to move the ownership of storage aggregates within the HA pair. Aggregate relocation enables storage management flexibility not only by optimizing performance during failover events, but also facilitating system operational and maintenance capabilities that previously required controller failover.

Aggregate relocation occurs automatically during manually initiated takeovers to reduce downtime during planned failover events such as nondisruptive software upgrades. You can manually initiate aggregate relocation independent of failover for performance load balancing, system maintenance, and nondisruptive controller upgrades. However, you cannot use the aggregate relocation operation to move ownership of the root aggregate.

The following illustration shows the relocation of the ownership of aggregate aggr_1 from Node1 to Node2 in the HA pair:



The aggregate relocation operation can relocate the ownership of one or more SFO aggregates if the destination node can support the number of volumes in the aggregates. There is only a brief interruption of access to each aggregate. Ownership information is changed one by one for the aggregates.

During takeover, aggregate relocation happens automatically after you manually initiate takeover. Before the target controller is taken over, ownership of each of the controller's aggregates is moved, one at a time, to the partner controller. When giveback is initiated, ownership is automatically moved back to the original node. The `-bypass-optimization` parameter can be used with the `storage failover takeover` command to suppress aggregate relocation during the takeover.

Aggregate relocation and Infinite Volumes with SnapDiff enabled

The aggregate relocation requires additional steps if the aggregate is currently used by an Infinite Volume with SnapDiff enabled. You must ensure that the destination node has a namespace mirror constituent, and make decisions about relocating aggregates that include namespace constituents.

For information about Infinite Volumes, see the *Clustered Data ONTAP Infinite Volumes Management Guide*.

Related tasks

Relocating ownership of aggregates used by Infinite Volumes on page 119

Relocating aggregate ownership

You can change the ownership of an aggregate only between the nodes within an HA pair.

About this task

- Because volume count limits are validated programmatically during aggregate relocation operations, it is not necessary to check for this manually. If the volume count exceeds the supported limit, the aggregate relocation operation will fail with a relevant error message.
- You should not initiate aggregate relocation when system-level operations are in progress on either the source or the destination node; likewise, you should not start these operations during the aggregate relocation. These operations can include:
  - Takeover
  - Giveback
  - Shutdown
  - Another aggregate relocation operation
  - Disk ownership changes
  - Aggregate or volume configuration operations
  - Storage controller replacement
  - Data ONTAP upgrade
  - Data ONTAP revert
- You should not initiate aggregate relocation on aggregates that are corrupt or undergoing maintenance.
- If the source node is used by an Infinite Volume with SnapDiff enabled, you must perform additional steps before initiating the aggregate relocation and then perform the relocation in a specific manner. You must ensure that the destination node has a namespace mirror constituent and make decisions about relocating aggregates that include namespace constituents.

  For information about Infinite Volumes, see the *Clustered Data ONTAP Infinite Volumes Management Guide*.
- Before initiating the aggregate relocation, save any core dumps on the source and destination nodes.

Steps

1. View the aggregates on the node to confirm which aggregates to move and ensure they are online and in good condition:

   `storage aggregate show -node` *`source-node`*

   #### Example

   The following command shows six aggregates on the four nodes in the cluster. All aggregates are online. Node1 and Node 3 form an HA pair and Node2 and Node4 form an HA pair.

```
node1::> storage aggregate show
Aggregate     Size Available Used% State   #Vols  Nodes  RAID Status
--------- -------- --------- ----- ------- ------ ------ -----------
aggr_0     239.0GB   11.13GB   95% online       1 node1  raid_dp,
                                                        normal
aggr_1     239.0GB   11.13GB   95% online       1 node1  raid_dp,
                                                        normal
aggr_2     239.0GB   11.13GB   95% online       1 node2  raid_dp,
                                                        normal
aggr_3     239.0GB   11.13GB   95% online       1 node2  raid_dp,
                                                        normal
aggr_4     239.0GB   238.9GB    0% online       5 node3  raid_dp,
                                                        normal
aggr_5     239.0GB   239.0GB    0% online       4 node4  raid_dp,
                                                        normal
6 entries were displayed.
```

2. Issue the command to start the aggregate relocation:

   storage aggregate relocation start -aggregate-list *aggregate-1, aggregate-2...* -node *source-node* -destination *destination-node*

   The following command moves the aggregates aggr_1 and aggr_2 from Node1 to Node3. Node3 is Node1's HA partner. The aggregates can only be moved within the HA pair.

```
node1::> storage aggregate relocation start -aggregate-list aggr_1,
aggr_2 -node node1 -destination node3
Run the storage aggregate relocation show command to check relocation
status.
node1::storage aggregate>
```

3. Monitor the progress of the aggregate relocation with the `storage aggregate relocation show` command:

   storage aggregate relocation show -node *source-node*

   Example

   The following command shows the progress of the aggregates that are being moved to Node3:

```
node1::> storage aggregate relocation show -node node1
Source Aggregate   Destination    Relocation Status
------ ----------- -------------  ------------------------
node1
       aggr_1      node3          In progress, module: wafl
       aggr_2      node3          Not attempted yet
2 entries were displayed.
node1::storage aggregate>
```

   When the relocation is complete, the output of this command shows each aggregate with a relocation status of `Done`.

Related tasks

Relocating ownership of aggregates used by Infinite Volumes on page 119

Commands for aggregate relocation

There are specific Data ONTAP commands for relocating aggregate ownership within an HA pair.

If you want to...	Use this command...
Start the aggregate relocation process.	`storage aggregate relocation start`
Monitor the aggregate relocation process	`storage aggregate relocation show`

See the man page for each command for more information.

Key parameters of the storage aggregate relocation start command

The `storage aggregate relocation start` command includes several key parameters used when relocating aggregate ownership within an HA pair.

Parameter	Meaning
-node *nodename*	Specifies the name of the node that currently owns the aggregate
-destination *nodename*	Specifies the destination node where aggregates are to be relocated
-aggregate-list *aggregate name*	Specifies the list of aggregate names to be relocated from source node to destination node (This parameter accepts wildcards)
-override-vetoes *true*|*false*	Specifies whether to override any veto checks during the relocation operation
-relocate-to-higher-version *true*|*false*	Specifies whether the aggregates are to be relocated to a node that is running a higher version of Data ONTAP than the source node
-override-destination-checks *true*|*false*	Specifies if the aggregate relocation operation should override the check performed on the destination node

See the man page for more information.

Veto and destination checks during aggregate relocation

In aggregate relocation operations, Data ONTAP determines whether aggregate relocation can be completed safely. If aggregate relocation is vetoed, you must check the EMS messages to determine

the cause. Depending on the reason or reasons, you can decide whether you can safely override the vetoes.

The `storage aggregate relocation show` command displays the aggregate relocation progress and shows which subsystem, if any, vetoed the relocation. Soft vetoes can be overridden, whereas hard vetoes cannot be, even if forced. The following tables summarize the soft and hard vetoes, along with recommended workarounds.

Veto checks during aggregate relocation

Vetoing subsystem module	Workaround
Vol Move	Relocation of an aggregate is vetoed if any volumes hosted by the aggregate are participating in a volume move that has entered the cutover state. Wait for the volume move to complete. If this veto is overridden, cutover will resume automatically once the aggregate relocation completes. If aggregate relocation causes the move operation to exceed the number of retries (the default is 3), then the user needs to manually initiate cutover using the `volume move trigger-cutover` command.
Backup	Relocation of an aggregate is vetoed if a dump or restore job is in progress on a volume hosted by the aggregate. Wait until the dump or restore operation in progress is complete. If this veto is overridden, the backup or restore operation will be aborted and must be restarted by the backup application.
Lock manager	To resolve the issue, gracefully shut down the CIFS applications that have open files, or move those volumes to a different aggregate. Overriding this veto will result in loss of CIFS lock state, causing disruption and data loss.
Lock Manager NDO	Wait until the locks are mirrored. This veto cannot be overridden; doing so will disrupt Microsoft Hyper-V virtual machines.
RAID	Check the EMS messages to determine the cause of the veto: If disk add or disk ownership reassignment operations are in progress, wait until they complete. If the veto is due to a mirror resync, a mirror verify, or offline disks, the veto can be overridden and the operation will be restarted after giveback.

Destination checks during aggregate relocation

Vetoing subsystem module	Workaround
Disk Inventory	Relocation of an aggregate will fail if the destination node is unable to see one or more disks belonging to the aggregate. Check storage for loose cables and verify that the destination can access disks belonging to the aggregate being relocated. This check cannot be overridden.
WAFL	Relocation of an aggregate will fail if the relocation would cause the destination to exceed its limits for maximum volume count or maximum volume size. This check cannot be overridden.
Lock Manager NDO	Relocation of an aggregate will fail if: • The destination does not have sufficient lock manager resources to reconstruct locks for the relocating aggregate. • The destination node is reconstructing locks. Retry aggregate relocation after a few minutes. This check cannot be overridden.
Lock Manager	Permanent relocation of an aggregate will fail if the destination does not have sufficient lock manager resources to reconstruct locks for the relocating aggregate. Retry aggregate relocation after a few minutes. This check cannot be overridden.
RAID	Check the EMS messages to determine the cause of the failure: • If the failure is due to an aggregate name or UUID conflict, troubleshoot and resolve the issue. This check cannot be overridden. Relocation of an aggregate will fail if the relocation would cause the destination to exceed its limits for maximum aggregate count, system capacity, or aggregate capacity. You should avoid overriding this check.

Moving an aggregate composed of array LUNs

You might want to move an aggregate composed of array LUNs to a less-loaded system to balance the load processing over the systems.

Before you begin

- You should plan the number and size of your aggregates ahead of time so that you have flexibility in the amount of the workload that you can shift from one system to another.
- You should ensure that the *target* system meets the following requirements:
  - The target system must be running a version of Data ONTAP that is the same as or later than the version running on the source system.
  - The target system must support the size of the aggregate being moved.

Steps

1. Enter the following commands on the target system:
   a) Enter the following to access the nodeshell:

      system run -node *node_name*

      node_name is the name of the target system.
   b) Obtain the system ID of the target system by entering either of the following commands:

      disk show
      or

      sysconfig

      You need to provide the target system's ID on the source system when you assign each of the array LUNs to the target system.
2. Enter the following commands on the source system:
   a) Enter the following command to access the nodeshell:

      system run -node *node_name*

      node_name is the name of the source system.
   b) Enter the following command to display the array LUNs that the aggregate contains:

      aggr status *aggr_name* -r

      The array LUNs that are displayed are the LUNs that you need to reassign to the target system to be able to move the aggregate.
   c) Write down the names of the array LUNs in the aggregate that you want to move.
   d) Enter the following command to shut down the source system:

      halt

e) At the boot environment prompt, enter the following command to boot the source system:

bye

f) Interrupt the boot process by pressing Ctrl-C when you see the following message on the console:
`Press Ctrl-C for Boot menu`

g) Enter Maintenance mode.

h) When prompted whether you want to continue with booting, enter the following:

y

i) Enter the following command to take the aggregate offline:

aggr offline *aggr_name*

aggr_name is the name of the traditional volume or aggregate.

j) Enter the following and confirm that the aggregate is offline:

aggr status

k) In Maintenance mode, enter the following command *separately* for each array LUN in the aggregate that you are moving to the target system:

disk assign -s *system_id_target* *disk_id* -f

system_id_target is the system ID of the target system (the system to which you want to move the array LUN.)

disk_id is the ID of the array LUN you want to move.

Note: Entering this command automatically removes ownership of the array LUN from the source system and assigns it to the target system.

3. Enter the following commands on the target system.

a) Enter the following command to start a scan so that the target system can recognize the LUNs you moved to it as its own:

disk show

The target system should still be in the nodeshell

b) Enter the following command:

aggr status

The display shows the *foreign* aggregate as offline.

Note: The aggregate you are moving is a foreign aggregate to the target system.

If the foreign aggregate has the same name as an existing aggregate on the system, Data ONTAP renames it *aggr_name(1)*, where *aggr_name* is the original name of the aggregate.

Attention: If the foreign aggregate is incomplete, that is, if you have not moved all the array LUNs in the aggregate, go back to the source system to add the missing array LUNs to the aggregate you moved to the target system. Enter the following on the source system:

disk assign -s *system_id_target* *disk_id* -f

c) If Data ONTAP renamed the foreign aggregate because of a name conflict and you want to change the name, enter the following command to rename the aggregate:

aggr rename *aggr_name new_name*

aggr_name is the name of the aggregate you want to rename.

new_name is the new name of the aggregate.

Example

The following command renames the users(1) aggregate as newusers:

```
aggr rename users(1) newusers
```

d) Enter the following command to confirm that the aggregate you moved is online:

aggr status *aggr_name*

aggr_name is the name of the aggregate.

4. On the source system, reboot the system out of Maintenance mode.

Assigning aggregates to SVMs

If you assign one or more aggregates to a Storage Virtual Machine (SVM, formerly known as Vserver), then you can use only those aggregates to contain volumes for that SVM. Assigning aggregates to your SVMs is particularly important in a multi-tenancy environment or when you use Infinite Volumes.

Before you begin

The SVM and the aggregates you want to assign to that SVM must already exist.

About this task

Assigning aggregates to your SVMs helps you keep your SVMs isolated from each other; this is especially important in a multi-tenancy environment. If you use Infinite Volumes, or plan to use them in the future, you must assign aggregates to your SVMs to keep your Infinite Volumes from impacting each other and any FlexVol volumes on your cluster.

Steps

1. Check the list of aggregates already assigned to the SVM by entering the following command:

   vserver show -fields aggr-list

   The aggregates currently assigned to the SVM are displayed. If there are no aggregates assigned, "-" is displayed.

2. Assign one or more aggregates to the SVM by entering the following command:

   vserver modify -vserver *vserver_name* -aggr-list *aggr_name*

To assign more than one aggregate to the SVM, list all of the aggregate names separated by commas.

Note: If there is already one or more aggregates assigned to the SVM and you want those aggregates to continue to be assigned to that SVM, you must include their names in the list you provide. Otherwise, they will no longer be assigned to that SVM.

The aggregates you specified are assigned to the SVM. If the SVM already has volumes contained by aggregates that are not assigned to the SVM, a warning is displayed, but the command succeeds.

Example

In the following example, the aggregates aggr1 and aggr2 are assigned to SVM vs1:

```
vserver modify -vserver vs1 -aggr-list aggr1,aggr2
```

Related concepts

How the SVM affects which aggregates can be associated with a FlexVol volume on page 102

Methods to create space in an aggregate

If an aggregate runs out of free space, various problems can result that range from loss of data to disabling a volume's guarantee. There are multiple ways to make more space in an aggregate.

All of the methods have various consequences. Prior to taking any action, you should read the relevant section in the documentation.

The following are some common ways to make space in an aggregate, in order of least to most consequences:

- Add disks to the aggregate.
- Move some volumes to another aggregate with available space.
- Shrink the size of volumes whose guarantee type is `volume` in the aggregate.

  You can do this manually or with the `autoshrink` option of the autosize capability.
- Change volume guarantee types to `none` on volumes that are using large amounts of space (large volume-guaranteed volumes or file-guaranteed volumes with large reserved files) so that the volumes take up less space in the aggregate.

  A volume with a guarantee type of `none` has a smaller footprint in the aggregate than volumes with other guarantee types. The `Volume Guarantee` row of the `volume show-footprint` command output shows whether a volume is reserving a large amount of space in the aggregate due to its guarantee.

  If you configure your volumes with a volume guarantee of `none`, you should refer to Technical Report 3965 for information about how doing so can affect storage availability.
- Delete unneeded volume Snapshot copies if the volume's guarantee type is `none`.

- Delete unneeded volumes.
- Enable space-saving features, such as deduplication or compression.
- (Temporarily) disable features that are using a large amount of metadata (visible with the `volume show-footprint` command).

Related information

Technical Report: Thin Provisioning Deployment and Implementation Guide: media.netapp.com/documents/tr-3965.pdf

Technical Report: Thin Provisioning in a NetApp SAN or IP SAN Enterprise Environment: media.netapp.com/documents/tr3483.pdf

Determining which volumes reside on an aggregate

You might need to determine which FlexVol volumes or Infinite Volume constituents reside on an aggregate before performing operations on the aggregate, such as relocating it or taking it offline.

About this task

Infinite Volume constituents are somewhat similar to FlexVol volumes, but you usually do not manage them directly. For more information about Infinite Volumes and constituents, see the *Clustered Data ONTAP Infinite Volumes Management Guide*.

Step

1. Enter the appropriate command, depending on whether your system has Infinite Volumes:

If your system...	Then use this command...
Does not have Infinite Volumes	**`volume show -aggregate aggregate_name`**
Has Infinite Volumes	**`volume show -is-constituent * -aggregate aggregate_name`**

All volumes (and, if you have Infinite Volumes, constituents) that reside on the specified aggregate are displayed.

Commands for managing aggregates

There are specific commands for managing aggregates using the Data ONTAP CLI.

If you want to...	Use this command...
Bring an aggregate online	`storage aggregate online`

If you want to...	**Use this command...**
Delete an aggregate	`storage aggregate delete`
Determine the format of an aggregate	`storage aggregate show -fields block-type`
Put an aggregate into the restricted state	`storage aggregate restrict`
Rename an aggregate	`storage aggregate rename`
Take an aggregate offline	`storage aggregate offline`
Display the RAID usage of each disk in an aggregate	`storage disk show -aggregate <aggr_name> -fields position,raid_group`
Display the root aggregates in the cluster	`storage aggregate show -has-mroot true`

Storage limits

There are limits for storage objects that you should consider when planning and managing your storage architecture.

Limits are listed in the following sections:

- *Aggregate limits* on page 148
- *RAID group limits* on page 149

Aggregate limits

Limit	Native storage	Storage arrays	Virtual storage (Data ONTAP-v)	Notes
Aggregates Maximum per node	100	100	60	In an HA configuration, this limit applies to each node individually, so the overall limit for the pair is doubled.
Aggregates (32-bit) Maximum size	16 TB	16 TB	16 TB	
Aggregates (64-bit) Maximum size	Model-dependent	Model-dependent	16 TB	See the *Hardware Universe*.

Limit	Native storage	Storage arrays	Virtual storage (Data ONTAP-v)	Notes
Aggregates Minimum size	RAID-DP: 5 disks RAID4: 3 disks	Model-dependent	1 disk	For root aggregates, the minimum size is 3 disks for RAID-DP and 2 disks for RAID4. If you need a smaller non-root aggregate, you can use the force-small-aggregate option. See the *Hardware Universe* for the minimum aggregate size for storage arrays.
RAID groups Maximum per aggregate	150	150	60	

RAID group limits

For maximum and default RAID group sizes, see the *Hardware Universe*.

Limit	Native storage	Storage arrays	Virtual storage (Data ONTAP-v)	Notes
Maximum per system	400	400	60	
Maximum per aggregate	150	150	60	

Copyright information

Trademark information

How to send your comments

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

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments by email to *doccomments@netapp.com*. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

Index

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

Z