



Clustered Data ONTAP® 8.3

Physical Storage Management Guide

February 2017 | 215-10884_B0
doccomments@netapp.com

Updated for 8.3.2

Contents

Managing disks using Data ONTAP	9
How Data ONTAP reports disk types	9
Storage connection types and topologies supported by Data ONTAP	10
How disks can be combined for the SAS storage connection type	11
How disks can be combined for the FC-AL storage connection type	11
Methods of calculating aggregate and system capacity	11
Disk speeds supported by Data ONTAP	11
How drive checksum types affect aggregate and spare management	12
Drive name formats	12
Pre-cluster drive name formats	13
Loop IDs for FC-AL connected disks	15
Understanding RAID drive types	15
How disk sanitization works	16
Disk sanitization process	16
When disk sanitization cannot be performed	17
What happens if disk sanitization is interrupted	17
Tips for creating and backing up aggregates containing data to be sanitized	18
How Data ONTAP monitors disk performance and health	18
What happens when Data ONTAP takes disks offline	18
How Data ONTAP reduces disk failures using Rapid RAID Recovery	18
How the maintenance center helps prevent drive errors	19
When Data ONTAP can put a disk into the maintenance center	20
How Data ONTAP uses continuous media scrubbing to prevent media errors	20
How you can use ACP to increase storage availability for SAS-connected disk shelves	21
How you use SSDs to increase storage performance	21
How you manage All Flash FAS models differently from FAS models	21
How Data ONTAP manages SSD wear life	22
Capability differences between SSDs and HDDs	22
Guidelines and requirements for using multi-disk carrier storage shelves	23
How Data ONTAP avoids RAID impact when a multi-disk carrier must be removed	23
How to determine when it is safe to remove a multi-disk carrier	24
Spare requirements for multi-disk carrier disks	24
Shelf configuration requirements for multi-disk carrier storage shelves	24
Aggregate requirements for disks in multi-disk carrier storage shelves	24
Considerations for using disks from a multi-disk carrier storage shelf in an aggregate	25
Understanding root-data partitioning	25

How root-data partitioning works	25
How root-data partitioning affects storage management	26
Which drives are partitioned and used for the root aggregate	27
Standard root-data partitioning layouts	27
Requirements for using root-data partitioning	28
Initializing a node to configure root-data partitioning	29
Setting up an active-passive configuration on nodes using root-data partitioning	32
Adding disks to a node	34
When you need to update the Disk Qualification Package	35
Replacing disks that are currently being used in an aggregate	36
Converting a data disk to a hot spare	36
Removing disks from a node	37
Removing a failed disk	37
Removing a hot spare disk	38
Removing a data disk	39
Using disk sanitization to remove data from disks	40
Stopping disk sanitization	41
Commands for managing disks	42
Commands for displaying information about storage shelves	43
Commands for displaying space usage information	43
Managing ownership for disks	45
Types of disk ownership	45
Reasons to assign ownership of disks and array LUNs	45
How disks and array LUNs become available for use	45
How automatic ownership assignment works for disks	46
Which disk autoassignment policy to use	47
When automatic ownership assignment is invoked	47
How disk ownership works for platforms based on Data ONTAP-v technology	47
Guidelines for assigning ownership for disks	48
Assigning ownership for disks	48
Assigning ownership for disks partitioned for root-data partitioning	49
Removing ownership from a disk	49
Configuring automatic ownership assignment of disks	50
How you use the wildcard character with the disk ownership commands	51
Managing array LUNs using Data ONTAP	53
Data ONTAP systems that can use array LUNs on storage arrays	53
Installing the license for using array LUNs	54
How ownership for disks and array LUNs works	54
Reasons to assign ownership of disks and array LUNs	54
How disks and array LUNs become available for use	55
What it means for Data ONTAP to own an array LUN	56
Why you might assign array LUN ownership after installation	56
Examples showing when Data ONTAP can use array LUNs	57
Assigning ownership of array LUNs	58

Verifying the back-end configuration	59
Modifying assignment of spare array LUNs	60
Array LUN name format	60
Pre-cluster array LUN name format	61
Checking the checksum type of spare array LUNs	62
Changing the checksum type of an array LUN	63
Prerequisites to reconfiguring an array LUN on the storage array	63
Changing array LUN size or composition	64
Removing one array LUN from use by Data ONTAP	65
Preparing array LUNs before removing a Data ONTAP system from service	65
Securing data at rest with Storage Encryption	66
Introduction to Storage Encryption	66
What Storage Encryption is	66
How Storage Encryption works	66
Disk operations with SEDs	67
Benefits of using Storage Encryption	67
Limitations of Storage Encryption	68
Setting up Storage Encryption	69
Information to collect before configuring Storage Encryption	69
Using SSL for secure key management communication	70
Running the key management setup wizard	72
Running self-encrypting disks in FIPS compliance mode	73
What FIPS compliance mode means for self-encrypting disks	73
Considerations for running self-encrypting disks in FIPS compliance mode	73
Configuring self-encrypting disks to run in FIPS compliance mode	74
Taking self-encrypting disks out of FIPS compliance mode	74
Managing Storage Encryption	75
Adding key management servers	75
Verifying key management server links	76
Displaying key management server information	77
Removing key management servers	77
What happens when key management servers are not reachable during the boot process	78
Displaying Storage Encryption disk information	79
Changing the authentication key	80
Retrieving authentication keys	82
Deleting an authentication key	83
SSL issues due to expired certificates	83
Returning SEDs to unprotected mode	84
Returning self-encrypting disks to service when authentication keys are no longer available	85
Destroying data on disks using Storage Encryption	87
Sanitizing disks using Storage Encryption before return to vendor	87
Setting the state of disks using Storage Encryption to end-of-life	88

Emergency shredding of data on disks using Storage Encryption	88
What function the physical secure ID has for SEDs	89
SEDs that have PSID functionality	90
Resetting an SED to factory original settings	90
How Data ONTAP uses RAID to protect your data and data availability	92
RAID protection levels for disks	92
What RAID-DP protection is	92
What RAID4 protection is	93
RAID protection for array LUNs	93
RAID protection for Data ONTAP-v storage	93
Protection provided by RAID and SyncMirror	93
Understanding RAID drive types	96
How RAID groups work	96
How RAID groups are named	97
Considerations for sizing RAID groups	97
Customizing the size of your RAID groups	98
Considerations for Data ONTAP RAID groups for array LUNs	99
How Data ONTAP works with hot spare disks	99
Minimum number of hot spares you should have	99
What disks can be used as hot spares	100
What a matching spare is	100
What an appropriate hot spare is	100
About degraded mode	101
How low spare warnings can help you manage your spare drives	101
How Data ONTAP handles a failed disk with a hot spare	102
How Data ONTAP handles a failed disk that has no available hot spare	102
Considerations for changing the timeout RAID option	103
How RAID-level disk scrubs verify data integrity	103
Changing the schedule for automatic RAID-level scrubs	103
How you run a manual RAID-level scrub	104
Controlling the impact of RAID operations on system performance	104
Controlling the performance impact of RAID data reconstruction	105
Controlling the performance impact of RAID-level scrubbing	105
Controlling the performance impact of plex resynchronization	106
Controlling the performance impact of mirror verification	106
What aggregates are	108
How unmirrored aggregates work	108
How mirrored aggregates work	109
What a Flash Pool aggregate is	111
How Flash Pool aggregates work	111
Requirements for using Flash Pool aggregates	112
Considerations for RAID type and spare management for Flash Pool cache	112
Comparison of Flash Pool aggregates and Flash Cache	113

How the available Flash Pool cache capacity is calculated	113
How Flash Pool aggregate caching policies work	114
Modifying caching policies	116
How Flash Pool SSD partitioning increases cache allocation flexibility for Flash	
Pool aggregates	116
Considerations for when to use SSD storage pools	118
How you use SSD storage pools	118
Requirements and best practices for using SSD storage pools	119
Creating an SSD storage pool	119
Adding SSDs to an SSD storage pool	120
Considerations for adding SSDs to an existing storage pool versus	
creating a new one	121
Determining the impact to cache size of adding SSDs to an SSD storage	
pool	121
Commands for managing SSD storage pools	122
How the SVM affects which aggregates can be associated with a FlexVol	
volume	122
Understanding how Data ONTAP works with heterogeneous storage	123
How you can use disks with mixed speeds in the same aggregate	123
How to control disk selection from heterogeneous storage	123
Rules for mixing HDD types in aggregates	124
Rules for mixing drive types in Flash Pool aggregates	125
Rules for mixing storage in array LUN aggregates	125
How the checksum type is determined for array LUN aggregates	125
How to determine space usage in an aggregate	126
How you can determine and control a volume's space usage in the aggregate	127
How Infinite Volumes use aggregates	129
Aggregate requirements for Infinite Volumes	129
How FlexVol volumes and Infinite Volumes share aggregates	129
How storage classes affect which aggregates can be associated with	
Infinite Volumes	130
How aggregates and nodes are associated with Infinite Volumes	130
How space is allocated inside a new Infinite Volume	131
Relocating ownership of aggregates used by Infinite Volumes	132
Managing aggregates	134
Creating an aggregate using unpartitioned drives	134
Creating an aggregate using root-data partitioning	136
Increasing the size of an aggregate that uses unpartitioned drives	137
Increasing the size of an aggregate that uses root-data partitioning	139
Correcting misaligned spare partitions	142
What happens when you add storage to an aggregate	143
Creating a Flash Pool aggregate using physical SSDs	143
Creating a Flash Pool aggregate using SSD storage pools	145
Determining Flash Pool candidacy and optimal cache size	147

Determining and enabling volume write-caching eligibility for Flash Pool aggregates	149
Changing the RAID type of RAID groups in a Flash Pool aggregate	150
Determining drive and RAID group information for an aggregate	152
Relocating aggregate ownership within an HA pair	152
How aggregate relocation works	153
How root-data partitioning affects aggregate relocation	154
Relocating aggregate ownership	154
Commands for aggregate relocation	156
Key parameters of the storage aggregate relocation start command	156
Veto and destination checks during aggregate relocation	157
Assigning aggregates to SVMs	159
Methods to create space in an aggregate	160
Determining which volumes reside on an aggregate	161
Determining whether a Flash Pool aggregate is using an SSD storage pool	161
Commands for managing aggregates	161
Storage limits	163
Copyright information	165
Trademark information	166
How to send comments about documentation and receive update notifications	167
Index	168

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.

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).

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

FC-connected storage

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

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 <i>array LUNs</i> to distinguish them from the LUNs that Data ONTAP serves to clients.

Virtual storage (Data ONTAP-v)

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

Related concepts

[Rules for mixing HDD types in aggregates](#) on page 124

[Storage connection types and topologies supported by Data ONTAP](#) on page 10

Related references

[Understanding RAID drive types](#) on page 15

Related information

[NetApp Technical Report 3437: Storage Subsystem Resiliency Guide](#)

[NetApp Hardware Universe](#)

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. However, for MetroCluster configurations, the FC and SAS connection types can be combined in a bridged connection, with FC on the controller side and SAS on the shelf side. The bridged connection can be used in either a direct-attached or switched topology. For more information, see *Configuring a MetroCluster system with SAS disk shelves and FibreBridge 6500N bridges in 7-Mode*.

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 hww.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)
 - 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 hwwu.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**.
- For mirrored aggregates, both plexes must have the same checksum type.
- 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

When a node is part of a functioning cluster, the drives it is connected to are accessible using a simple, consistent drive name format. The drive name is independent of what nodes the drive is physically connected to and from which node you are accessing the drive.

The drive name format is a concatenation of four components:

<stack_id>.<shelf_id>.<bay>.<position>

Note: During system boot, before the node has joined the cluster, or if certain key cluster components become unavailable, drive names revert to the classic format based on physical connectivity.

- The stack ID is assigned by Data ONTAP.
Stack IDs are unique across the cluster, and start with 1.
- The shelf ID is set on the storage shelf when the shelf is added to the stack or loop.
If there is a shelf ID conflict for SAS shelves, the shelf id is replaced with the shelf serial number in the drive name.
- The bay is the position of the disk within its shelf.
You can find the bay map in the administration guide for your storage shelf.
- The position is used only for multi-disk carrier storage shelves.
For carriers that house two disks, it can be 1 or 2.

Related concepts

[Pre-cluster array LUN name format](#) on page 61

[Pre-cluster drive name formats](#) on page 13

Related references

[Commands for displaying information about storage shelves](#) on page 43

Related information

[SAS Disk Shelves Installation and Service Guide for DS4243, DS2246, DS4486, and DS4246](#)

[DS14mk2 FC, and DS14mk4 FC Hardware Service Guide](#)

[DiskShelf14mk2 AT Hardware Service Guide](#)

Pre-cluster drive name formats

During steady-state operations, drive names are independent of their physical connection and unaffected by the node from which they are viewed. However, during system boot, before a node has joined a cluster, and if certain system components become unavailable, drive names revert to the format used before Data ONTAP 8.3, the *pre-cluster* format.

The pre-cluster 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 pre-cluster 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>	<p>The pre-cluster name for the drive in shelf 2, bay 11, connected to onboard port 0a and owned by node1 is node1:0a.2.11.</p> <p>The pre-cluster name for the drive in shelf 6, bay 3, connected to an HBA in slot 1, port c, and owned by node1 is node1:1c.6.3.</p>
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>	<p>The pre-cluster name for the third virtual disk connected to the first port is 0b.3.</p> <p>The pre-cluster name for the second virtual disk connected to the third port is 0d.2.</p> <p>The range of ports is b through e, and the range of disks is 0 through 15.</p>
SAS, bridge-attached (FibreBridge, used for MetroCluster configurations)	<slot><port>.<loopID>L<LUN>	The pre-cluster name for the drive with LUN 2 behind the bridge connected to port a in slot 3, loop ID 125, is 3a.125L2.
SAS, bridge-attached through a switch (FibreBridge, used for MetroCluster configurations)	<switch_name>:<switch_port>.<loopID>L<LUN>	The pre-cluster name for the drive with LUN 5 behind the bridge connected to port 2 of switch brcd44, loop ID 126, is brcd44:2.126L5.

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

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

Related concepts

[Drive name formats](#) on page 12

[Pre-cluster 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 drive types

Data ONTAP classifies drives (or, for partitioned drives, *partitions*) as one of four types for RAID: data, hot spare, parity, or dParity. You manage disks differently depending on whether they are spare or being used in an aggregate.

The RAID type is determined by how RAID is using a drive or partition; it is different from the Data ONTAP disk type.

You cannot affect the RAID type for a drive. The RAID type is displayed in the `Position` column for many storage commands.

For drives using root-data partitioning and SSDs in storage pools, a single drive might be used in multiple ways for RAID. For example, the root partition of a partitioned drive might be a spare partition, whereas the data partition might be being used for parity. For this reason, the RAID drive type for partitioned drives and SSDs in storage pools is displayed simply as `shared`.

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 40

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 hww.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.

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 101

[When Data ONTAP can put a disk into the maintenance center](#) on page 20

[How Data ONTAP works with hot spare disks](#) on page 99

How the maintenance center helps prevent drive errors

Drives can sometimes display small problems that do not interfere with normal operation, but which could be a sign that the drive might fail sometime soon. The *maintenance center* provides a way to put these drives under increased scrutiny.

When a suspect drive is in the maintenance center, it is subjected to a number of tests. If the drive passes all of the tests, Data ONTAP redesignates it as a spare; if it fails any tests, Data ONTAP fails the drive.

By default, Data ONTAP puts a suspect drive into the maintenance center automatically only if there are two or more spares available for that drive. If that drive is housed in a shelf that supports automatic power-cycling, power to that drive might be turned off for a short time. If the drive returns to a ready state after the power-cycle, the maintenance center tests the drive. Otherwise, the maintenance center fails the drive immediately.

You can put a suspect drive into the maintenance center manually regardless of how many spares are available. You can also specify the number of times a drive is allowed to go to the maintenance center and whether it should go immediately or only after copying its contents to a spare drive.

- The `disk.maint_center.enable` option controls whether the maintenance center is on or off. The default value is **on**.
- The `disk.maint_center.allowed_entries` option controls the number of times a suspect drive is allowed to go to the maintenance center. The default value is **1**, which means that if the drive is sent to the maintenance center more than once, it is automatically failed.
- The `disk.maint_center.spares_check` option controls whether Data ONTAP requires that sufficient spares are available before it puts a drive into the maintenance center.
- The `disk maint start` command, available through the nodeshell, enables you to put a suspect drive into the maintenance center manually.
If the target drive is in use, the default is that it does not enter the maintenance center until its contents have been copied to a spare drive. You can put the drive into the maintenance center immediately by including the `-i` option.

Related information

[NetApp Technical Report 3437: Storage Subsystem 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.
Instead, Data ONTAP submits the disk to a small number of tests. If the disk fails those tests, Data ONTAP fails the disk regardless of the number of spares available, 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.

How you can use ACP to increase storage availability for SAS-connected disk shelves

The Alternate Control Path (ACP) protocol enables Data ONTAP to manage and control SAS-connected storage shelves, which can increase storage availability. After you ensure that ACP is properly enabled and configured, you do not need to actively manage it.

You can install SAS-connected storage shelves without configuring ACP. However, for maximum storage availability and stability, you should always have ACP enabled and configured, including providing the required physical connectivity and configuration parameters to enable the ACP functionality.

ACP is enabled by default. If you need to enable it or change its configuration, you can use the `storage shelf acp configure` and `storage shelf acp show` commands.

You do not need to actively manage the SAS-connected storage shelf subsystem. Data ONTAP automatically monitors and manages the subsystem without operator intervention.

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.

Related concepts

[*How Flash Pool aggregates work*](#) on page 111

[*How Flash Pool SSD partitioning increases cache allocation flexibility for Flash Pool aggregates*](#) on page 116

How you manage All Flash FAS models differently from FAS models

All Flash FAS models provide extremely high performance; they also introduce some management restrictions.

You cannot attach HDDs or array LUNs to an All-Flash FAS node; no drives other than SSDs are recognized as supported.

All volumes created in aggregates owned by an All Flash FAS node have inline compression enabled by default.

You cannot pair an All Flash FAS node with a regular FAS node in an HA pair.

The root-data partitioning capability is enabled by default on an All Flash FAS node.

All-Flash FAS nodes contain `All-Flash Optimized` in the output of the `system node show - instance` command.

Related information

[NetApp Hardware Universe](#)

[Clustered Data ONTAP 8.3 Logical Storage Management Guide](#)

[Using All Flash FAS with Clustered Data ONTAP](#)

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.
- The maintenance center

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.
- All HDDs in the aggregate must be the same Data ONTAP disk type.
The aggregate can be a Flash Pool aggregate.

- If the aggregate is mirrored, both plexes must have the same Data ONTAP disk type (or types, in the case of a Flash Pool aggregate).

Related concepts

[How Flash Pool aggregates work](#) on page 111

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
```

Understanding root-data partitioning

Some platform models use partitioning to enable the root aggregate to use less space, leaving more space for the data aggregate and improving storage utilization. The root-data partitioning capability, also called *Advanced Drive Partitioning*, changes the way you view and administer your disks and aggregates.

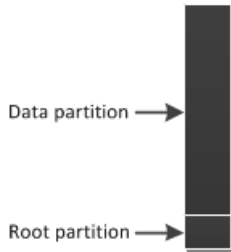
How root-data partitioning works

For entry-level platforms, All Flash FAS (AFF) platforms, and FAS platforms with only SSDs attached, aggregates can be composed of parts of a drive rather than the entire drive.

The root-data partitioning capability is usually enabled and configured by the factory. It can also be established by initiating system initialization using option 4 from the boot menu. Note that system

initialization erases all data on the disks of the node and resets the node configuration to the factory default settings.

When a node has been configured to use root-data partitioning, partitioned disks have two partitions:



The smaller partition is used to compose the root aggregate. The larger partition is used in data aggregates. The size of the partitions is set by Data ONTAP, and depends on the number of disks used to compose the root aggregate when the system is initialized. (The more disks used to compose the root aggregate, the smaller the root partition.) After system initialization, the partition sizes are fixed; adding partitions or disks to the root aggregate after system initialization increases the size of the root aggregate, but does not change the root partition size.

The partitions are used by RAID in the same manner as physical disks are; all of the same requirements apply. For example, if you add an unpartitioned drive to a RAID group consisting of partitioned drives, the unpartitioned drive is partitioned to match the partition size of the drives in the RAID group and the rest of the disk is unused.

If a partitioned disk is moved to another node or used in another aggregate, the partitioning persists; you can use the disk only in RAID groups composed of partitioned disks.

Related information

[Clustered Data ONTAP 8.3 System Administration Guide](#)

How root-data partitioning affects storage management

Generally, you administer a node that is using root-data partitioning the same way you administer a node that uses only physical disks. However, partitioned disks are displayed slightly differently, and you need to take their presence into account for some management tasks.

Root and data partitions are used by the RAID subsystem the same way that physical disks are, and the same requirements and best practices apply. You follow the same best practices for spare management, and disk names are not impacted by root-data partitioning.

When you are working with a node that is using root-data partitioning, the following tasks are slightly different:

- **Creating and adding storage to aggregates**
Spare partitions are displayed differently—as either `root` or `data`. In addition, you need to ensure that a root and data spare are provided on the same disk to support core file creation. Finally, you should understand whether you are mixing physical disks and partitions in the same RAID group, and the impact of doing so.
- **Displaying drive and RAID group information for an aggregate**
Partitioned disks display `shared` for the Position column, rather than their RAID disk type.
- **Assigning and removing ownership**
Automatic ownership assignment works for partitioned disks. If you need to manually assign or remove ownership for partitioned disks, you must do so for both partitions as well as the container disk.

- Removing and replacing disks
Removing a partitioned disk can potentially affect more than one aggregate.

Related concepts

[Which drives are partitioned and used for the root aggregate](#) on page 27

[Removing disks from a node](#) on page 37

[Requirements for using root-data partitioning](#) on page 28

Related tasks

[Creating an aggregate using root-data partitioning](#) on page 136

[Increasing the size of an aggregate that uses root-data partitioning](#) on page 139

[Determining drive and RAID group information for an aggregate](#) on page 152

[Assigning ownership for disks partitioned for root-data partitioning](#) on page 49

Related references

[Commands for managing disks](#) on page 42

Which drives are partitioned and used for the root aggregate

The drives that are partitioned for use in the root aggregate depend on the system configuration. Knowing how many drives are used for the root aggregate helps you to determine how much of the drives' capacity is reserved for the root partition, and how much is available for use in a data aggregate.

The root-data partitioning capability is supported for entry-level platforms, All Flash FAS platforms, and FAS platforms with only SSDs attached.

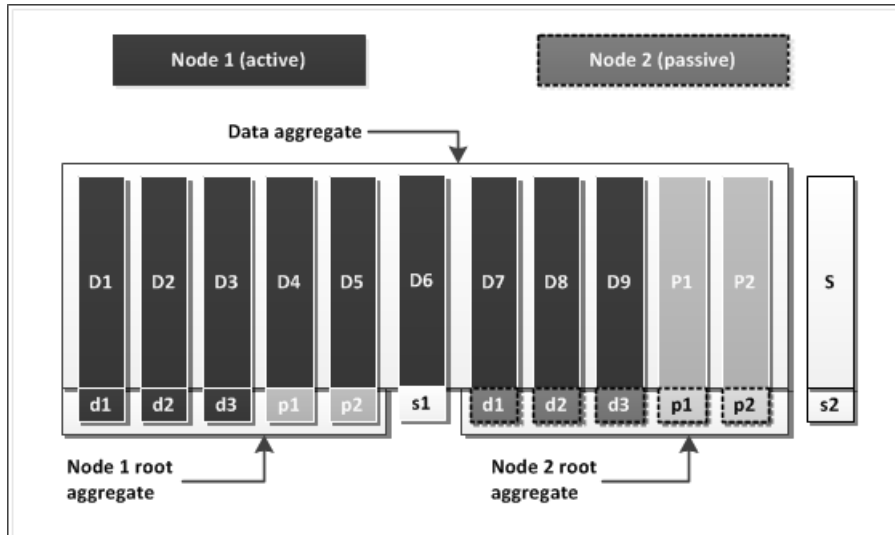
For entry-level platforms, only the internal drives are partitioned.

For All Flash FAS platforms and FAS platforms with only SSDs attached, all drives that are attached to the controller when the system is initialized are partitioned, up to a limit of 24 per node. Drives that are added after system configuration are not partitioned.

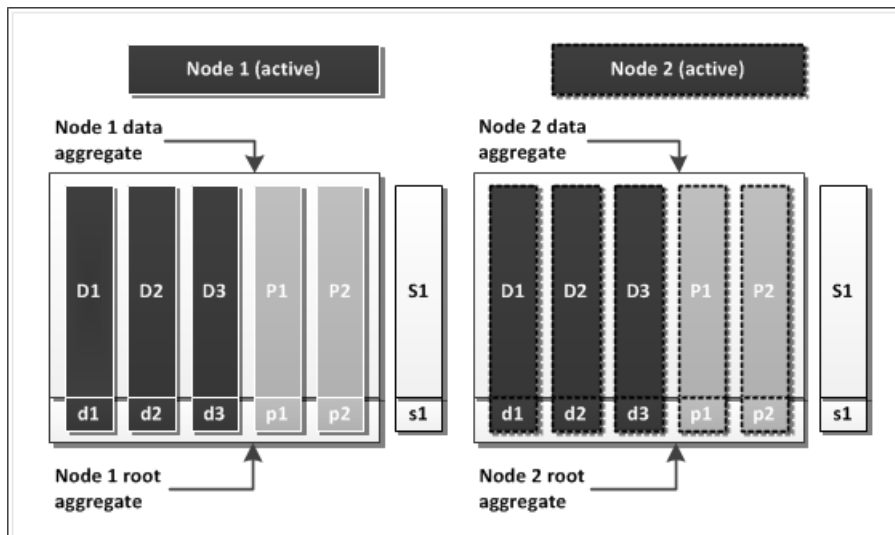
Standard root-data partitioning layouts

The root aggregate is configured by the factory; you should not change it. However, you can use the data partitions in a few different configurations, depending on your requirements.

The following diagram shows one way to configure the partitions for an active-passive configuration with 12 partitioned disks. There are two root aggregates, one for each node, composed of the small partitions. Each root aggregate has a spare partition. There is just one RAID-DP data aggregate, with two parity disk partitions and one spare partition.



The following diagram shows one way to configure the partitions for an active-active configuration with 12 partitioned disks. In this case, there are two RAID-DP data aggregates, each with their own data partitions, parity partitions, and spares. Note that each disk is allocated to only one node. This is a best practice that prevents the loss of a single disk from affecting both nodes.



The disks used for data, parity, and spare partitions might not be exactly as shown in these diagrams. For example, the parity partitions might not always align on the same disk.

Requirements for using root-data partitioning

In most cases, you can use drives that are partitioned for root-data partitioning exactly as you would use a physical, unshared drive. However, you cannot use root-data partitioning in certain configurations.

The following storage devices cannot be partitioned:

- Array LUNs
- Virtual disks as created by Data ONTAP-v
- HDD types that are not available as internal drives: ATA, FCAL, and MSATA

- 100-GB SSDs

You cannot use root-data partitioning with the following technologies:

- MetroCluster
- Data ONTAP-v
- RAID4

Aggregates composed of partitioned drives must have a RAID type of RAID-DP.

Initializing a node to configure root-data partitioning

If you are initializing a node whose platform model supports root-data partitioning, you must complete some steps before performing the initialization to ensure that root-data partitioning is configured correctly.

Before you begin

- **Attention:** All data that you need to retain **must** have been migrated to another node.
All data on the HA pair is erased during this procedure and cannot be retrieved.
- Your node configuration must support root-data partitioning.
If the node does not satisfy the requirements for root-data partitioning, root-data partitioning is not configured.
- The node or HA pair you are initializing should not be joined with more nodes in a cluster, and should not be serving data.
- Your nodes must be running Data ONTAP 8.3 or later.
- Neither node is in takeover or giveback.

About this task

This procedure is designed to be run on an HA pair, so the nodes are called Node A and Node B. However, you can still use this procedure if you have only one controller; ignore instructions that are specifically for Node B.

This procedure can take many hours to complete, depending on the amount and type of storage attached to the HA pair.

Steps

1. Record your node configuration, including network configuration, license values, and passwords, and if you want to restore the same storage architecture as you have now, record that also.

All of the current node configuration information will be erased when the system is initialized.

2. If you are performing this procedure on a two-node cluster, disable the HA capability:

```
cluster ha modify -configured false
```

This can be done from either node.

3. Disable storage failover for both nodes:

```
storage failover modify -enabled false -node nodeA,nodeB
```

4. On both nodes, boot into maintenance mode:

- a. Halt the system:

```
system node halt -node node_name
```

You can ignore any error messages about epsilon. For the second node, you can include the `-skip-lif-migration-before-shutdown` flag if prompted to do so.

- b. At the LOADER prompt, boot Data ONTAP:

```
boot_ontap
```

- c. Monitor the boot process, and when prompted, press Ctrl-C to display the boot menu.

Example

```
*****
*                                     *
* Press Ctrl-C for Boot Menu.      *
*                                     *
*****
```

- d. Select option 5, Maintenance mode boot.

Example

```
Please choose one of the following:

(1) Normal Boot.
(2) Boot without /etc/rc.
(3) Change password.
(4) Clean configuration and initialize all disks.
(5) Maintenance mode boot.
(6) Update flash from backup config.
(7) Install new software first.
(8) Reboot node.
Selection (1-8)? 5
```

5. For both nodes, if there are external disks connected to the node, destroy all aggregates, including the root aggregate:

- a. Display all aggregates:

```
aggr status
```

- b. For each aggregate, take the aggregate offline:

```
aggr offline aggr_name
```

- c. For each aggregate, destroy the aggregate:

```
aggr destroy aggr_name
```

All disks are converted to spares.

6. Ensure that no drives in the HA pair are partitioned:

- a. Display all drives owned by both nodes:

```
disk show
```

If any of the drives show three entries, that drive has been partitioned. If your nodes have no partitioned disks, you can skip to step 7.

Example

The following example shows partitioned drive 0a.10.11:.

DISK	OWNER	POOL	SERIAL NUMBER	HOME
0a.10.11	sys1(536880559)	Poo10	N11YE08L	sys1(536880559)
0a.10.11P1	sys1(536880559)	Poo10	N11YE08LNP001	sys1(536880559)
0a.10.11P2	sys1(536880559)	Poo10	N11YE08LNP002	sys1(536880559)

- b. Note any partitions that do not have the same owner as their container disk.

Example

In the following example, the container disk (0a.10.14) is owned by sys1, but partition one (0a.10.14P1) is owned by sys2.

DISK	OWNER	POOL	SERIAL NUMBER	HOME
0a.10.14	sys1(536880559)	Poo10	N11YE08L	sys1(536880559)
0a.10.14P1	sys2(536880408)	Poo10	N11YE08LNP001	sys2(536880408)
0a.10.14P2	sys1(536880559)	Poo10	N11YE08LNP002	sys1(536880559)

- c. Update the ownership of all partitions owned by a different node than their container disk:

```
disk assign disk_partition -f -o container_disk_owner
```

Example

You would enter a command like the following example for each disk with partitions owned by a different node than their container disk:

```
disk assign 0a.10.14P1 -f -o sys1
```

- d. For each drive with partitions, on the node that owns the container disk, remove the partitions:

```
disk unpartition disk_name
```

disk_name is the name of the disk, without any partition information, such as “0a.10.3”.

Example

You would enter a command like the following example for each disk with partitions:

```
disk unpartition 0a.10.14
```

7. On both nodes, remove disk ownership:

```
disk remove_ownership
```

8. On both nodes, verify that all drives connected to both nodes are unowned:

```
disk show
```

Example

```
*> disk show
Local System ID: 465245905
disk show: No disk match option show.
```

9. On both nodes, return to the LOADER menu:

```
halt
```

10. On Node A *only*, begin zeroing the drives:

- a. At the LOADER prompt, boot Data ONTAP:

```
boot_ontap
```

- b. Monitor the boot process, and when prompted, press Ctrl-C to display the boot menu.

- c. Select option **4**, Clean configuration and initialize all disks.

When the drives begin to be zeroed, a series of dots are printed to the console.

11. After the drives on Node A have been zeroing for a few minutes, repeat the previous step on Node B.

The drives on both nodes that will be partitioned are zeroing. When the zeroing process is complete, the nodes return to the Node Setup wizard.

12. Restore your system configuration.
13. Confirm that the root aggregate on both nodes is composed of partitions:

storage aggregate show-status

The **Position** column shows **shared** and the usable size is a small fraction of the physical size:

```
Owner Node: sys1
Aggregate: aggr0_1 (online, raid_dp) (block checksums)
Plex: /aggr0_1/plex0 (online, normal, active, pool0)
RAID Group /aggr0_1/plex0/rg0 (normal, block checksums)
```

Position	Disk	Pool	Type	RPM	Usable Size	Physical Size	Status
shared	0a.10.10	0	BSAS	7200	73.89GB	828.0GB	(normal)
shared	0a.10.0	0	BSAS	7200	73.89GB	828.0GB	(normal)
shared	0a.10.11	0	BSAS	7200	73.89GB	828.0GB	(normal)
shared	0a.10.6	0	BSAS	7200	73.89GB	828.0GB	(normal)
shared	0a.10.5	0	BSAS	7200	73.89GB	828.0GB	(normal)

14. Run System Setup to reconfigure the HA pair or rejoin the cluster, depending on your initial configuration.
15. Ensure that your failover and HA capabilities are properly configured.

For a two-node cluster, both cluster HA and storage failover must be enabled.

For all other HA pairs, storage failover must be enabled.

Related concepts

[Which drives are partitioned and used for the root aggregate](#) on page 27

[Requirements for using root-data partitioning](#) on page 28

Related information

[Clustered Data ONTAP 8.3 Software Setup Guide](#)

[Clustered Data ONTAP 8.3 High-Availability Configuration Guide](#)

Setting up an active-passive configuration on nodes using root-data partitioning

When an HA pair is configured to use root-data partitioning by the factory, ownership of the data partitions is split between both nodes in the pair, for use in an active-active configuration. If you want to use the HA pair in an active-passive configuration, you must update partition ownership before creating your data aggregate.

Before you begin

- You should have decided which node will be the active node and which node will be the passive node.
- Storage failover must be configured on the HA pair.

About this task

This task is performed on two nodes: Node A and Node B.

All commands are input at the clustershell.

This procedure is designed for nodes for which no data aggregate has been created from the partitioned disks.

Steps

1. View the current ownership of the data partitions:

```
storage aggregate show-spare-disks
```

Example

You can see that half of the data partitions are owned by one node and half are owned by the other node. All of the data partitions should be spare.

```
cluster1::> storage aggregate show-spare-disks

Original Owner: cluster1-01
Pool0
Partitioned Spares

Disk              Type      RPM  Checksum      Local Data Usable      Local Root Usable      Physical Size
-----
1.0.0             BSAS      7200 block      753.8GB      0B      828.0GB
1.0.1             BSAS      7200 block      753.8GB      73.89GB  828.0GB
1.0.5             BSAS      7200 block      753.8GB      0B      828.0GB
1.0.6             BSAS      7200 block      753.8GB      0B      828.0GB
1.0.10            BSAS      7200 block      753.8GB      0B      828.0GB
1.0.11            BSAS      7200 block      753.8GB      0B      828.0GB

Original Owner: cluster1-02
Pool0
Partitioned Spares

Disk              Type      RPM  Checksum      Local Data Usable      Local Root Usable      Physical Size
-----
1.0.2             BSAS      7200 block      753.8GB      0B      828.0GB
1.0.3             BSAS      7200 block      753.8GB      0B      828.0GB
1.0.4             BSAS      7200 block      753.8GB      0B      828.0GB
1.0.7             BSAS      7200 block      753.8GB      0B      828.0GB
1.0.8             BSAS      7200 block      753.8GB      73.89GB  828.0GB
1.0.9             BSAS      7200 block      753.8GB      0B      828.0GB
12 entries were displayed.
```

2. Enter the advanced privilege level:

```
set advanced
```

3. For each data partition owned by the node that will be the passive node, assign it to the active node:

```
storage disk assign -force -data true -owner active_node_name -disk disk_name
```

You do not need to include the partition as part of the disk name.

Example

You would enter a command similar to the following example for each data partition you need to reassign:

```
storage disk assign -force -data true -owner cluster1-01 -disk 1.0.3
```

4. Confirm that all of the partitions are assigned to the active node.

Example

```
cluster1::*> storage aggregate show-spare-disks
```

Original Owner: cluster1-01
Pool0
Partitioned Spares

Disk	Type	RPM	Checksum	Local Data Usable	Local Root Usable	Physical Size
1.0.0	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.1	BSAS	7200	block	753.8GB	73.89GB	828.0GB
1.0.2	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.3	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.4	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.5	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.6	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.7	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.8	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.9	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.10	BSAS	7200	block	753.8GB	0B	828.0GB
1.0.11	BSAS	7200	block	753.8GB	0B	828.0GB

Original Owner: cluster1-02
Pool0
Partitioned Spares

Disk	Type	RPM	Checksum	Local Data Usable	Local Root Usable	Physical Size
1.0.8	BSAS	7200	block	0B	73.89GB	828.0GB

13 entries were displayed.

Note that cluster1-02 still owns a spare root partition.

- Return to administrative privilege:

```
set admin
```

- Create your data aggregate, leaving at least one data partition as spare:

```
storage aggregate create new_aggr_name -diskcount number_of_partitions -
node active_node_name
```

The data aggregate is created and is owned by the active node.

Related information

[Clustered Data ONTAP 8.3 High-Availability Configuration Guide](#)

Adding disks to a node

You add disks to a node 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 platform model supports the type of disk you want to add.

About this task

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

Steps

- Check the NetApp Support Site for newer disk and shelf firmware and Disk Qualification Package files.

If your node does not have the latest versions, you must update them before installing the new disk.

2. Install the disks according to the hardware guide for your disk shelf or the hardware and service guide for your platform.

The new disks are not recognized until they are assigned to a node and pool. You can assign the new disks manually, or you can wait for Data ONTAP to automatically assign the new disks if your node follows the rules for disk autoassignment.

3. After the new disks have all been recognized, verify their addition and their ownership information:

```
storage aggregate show-spare-disks
```

You should see the new disks, owned by the correct node and in the correct pool.

4. Optional: Zero the newly added disks:

```
storage disk zerospares
```

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 node.

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 46

[Guidelines for assigning ownership for disks](#) on page 48

Related information

[NetApp Hardware Universe](#)

[NetApp Downloads: Disk Drive and Firmware](#)

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 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
- Whenever you upgrade to a new version of Data ONTAP.
The DQP is not updated as part of a Data ONTAP upgrade.

Related information

[NetApp Downloads: Disk Qualification Package](#)

[NetApp Downloads: Disk Drive and Firmware](#)

Replacing disks that are currently being used in an aggregate

You replace disks that are currently being used in an aggregate to swap out mismatched disks from a RAID group. Keeping your RAID groups homogeneous helps to 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. If the disk to be replaced is partitioned or in a storage pool, the spare disk (or its partitions, if applicable) can be owned by either node in the HA pair.

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 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. Replace the disk:

```
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` parameter.

Result

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

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 node

How you remove a disk from a node 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 node.

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.

If you are removing a partitioned drive, you must account for how both partitions are being used, and unpartition the drive before you remove it.

If you are removing a disk in a multi-disk carrier, you must take special precautions.

Related concepts

[Understanding root-data partitioning](#) on page 25

[How to determine when it is safe to remove a multi-disk carrier](#) on page 24

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 node, and notifies Data ONTAP that you are removing the disk, which prevents unwanted AutoSupport messages.

Before you begin

If the disk you are removing is partitioned for root-data partitioning, you must unpartition it before removing it from the node.

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 aggregate show-spare-disks`
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. If disk ownership automatic assignment is on for the node's HA partner, turn it off on the partner node also.
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`
8. If you turned off disk ownership automatic assignment previously on the node's HA partner, turn it on for the partner node also.

Related concepts

[How to determine when it is safe to remove a multi-disk carrier](#) on page 24

Related tasks

[Using disk sanitization to remove data from disks](#) on page 40

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.

Before you begin

If the disk you are removing is partitioned for root-data partitioning, you must unpartition it before removing it from the node.

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:

```
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)	<p>Pre-fail the disk:</p> <pre>storage disk fail disk_name</pre> <p>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.</p> <p>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.</p>
Need to remove the disk immediately	<p>Cause the disk to fail immediately:</p> <pre>storage disk fail -disk disk_name -immediate</pre> <p>The disk fails and the storage system operates in degraded mode until the RAID system reconstructs a replacement disk.</p>

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 101

[How to determine when it is safe to remove a multi-disk carrier](#) on page 24

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 must be spare disks; they must be owned by a node, but not used in an aggregate. (If the disk is partitioned, neither partition can be in use in an aggregate.)
- The disks cannot be part of a storage pool.

About this task

When disk sanitization is enabled, it disables some Data ONTAP commands. After disk sanitization is enabled on a node, it cannot be disabled.

Steps

1. Enter the nodeshell for the node that owns the disks you want to sanitize:

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

2. Enable disk sanitization:

```
options licensed_feature.disk_sanitization.enable on
```

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

3. If the disks you want to sanitize are partitioned, unpartition each disk:

```
disk unpartition disk_name
```

4. Sanitize the specified disks:

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

Attention: Do not turn off power to the node, 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.

`-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:

```
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:

```
exit
```

8. Determine whether all of the disks were returned to spare status:

```
storage aggregate show-spare-disks
```

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	<p>Complete the following steps:</p> <ol style="list-style-type: none"> 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 node_name c. Return the disks to spare status by entering the following command for each disk: storage disk unfail -disk disk_name -s -q 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

You use the `storage disk` and `storage aggregate` commands to manage your disks.

If you want to...	Use this command...
Display a list of spare disks, including partitioned disks, by owner	<code>storage aggregate show-spare-disks</code>
Display the disk RAID type, current usage, and RAID group by aggregate	<code>storage aggregate show-status</code>
Display the RAID type, current usage, aggregate, and RAID group, including spares, for physical disks	<code>storage disk show -raid</code>
Display a list of failed disks	<code>storage disk show -broken</code>
Display the pre-cluster (nodescope) drive name for a disk	<code>storage disk show -primary-paths</code> (advanced)
Illuminate the LED for a particular disk or shelf	<code>storage disk set-led</code>
Display the checksum type for a specific disk	<code>storage disk show -fields checksum-compatibility</code>
Display the checksum type for all spare disks	<code>storage disk show -fields checksum-compatibility -container-type spare</code>
Display disk connectivity and placement information	<code>storage disk show -fields disk,primary-port,secondary-name,secondary-port,shelf,bay</code>
Display pre-cluster disk names for specific disks	<code>storage disk show -disk -fields diskpathnames</code>
Display a list of disks in the maintenance center	<code>storage disk show -maintenance</code>
Display SSD wear life	<code>storage disk show -ssd-wear</code>
Unpartition a disk	<code>system node run -node local -command disk unpartition</code>
Zero all non-zeroed disks	<code>storage disk zerospares</code>

Related information

[Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#)

Commands for displaying information about storage shelves

You use the `storage shelf show` command to display configuration and error information for your disk shelves.

If you want to display...	Use this command...
General information about shelf configuration and hardware status	<code>storage shelf show</code>
Detailed information for a specific shelf, including stack ID	<code>storage shelf show -shelf</code>
Unresolved, customer actionable, errors by shelf	<code>storage shelf show -errors</code>
Bay information	<code>storage shelf show -bay</code>
Connectivity information	<code>storage shelf show -connectivity</code>
Cooling information, including temperature sensors and cooling fans	<code>storage shelf show -cooling</code>
Information about I/O modules	<code>storage shelf show -module</code>
Port information	<code>storage shelf show -port</code>
Power information, including PSUs (power supply units), current sensors, and voltage sensors	<code>storage shelf show -power</code>

Related information

[Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#)

Commands for displaying space usage information

You use the `storage aggregate` and `volume` commands to 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	<code>storage aggregate show</code> <code>storage aggregate show-space -fields snap-size-total,used-including-snapshot-reserve</code>
How disks and RAID groups are used in an aggregate, and RAID status	<code>storage aggregate show-status</code>
The amount of disk space that would be reclaimed if you deleted a specific Snapshot copy	<code>volume snapshot compute-reclaimable</code> (advanced)
The amount of space used by a volume	<code>volume show -fields size,used,available,percent-used</code> <code>volume show-space</code>

To display information about...	Use this command...
The amount of space used by a volume in the containing aggregate	<code>volume show-footprint</code>

Related information

[Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#)

Managing ownership for disks

Disk ownership determines which node owns a disk and what pool a disk is associated with. Data ONTAP stores ownership information directly on the disk.

Types of disk ownership

The HA or Disaster Recovery (DR) state of the system that owns a disk can affect which system has access to the disk. This means that there are several types of ownership for disks.

Disk ownership information is set either by Data ONTAP or by the administrator, and recorded on the disk, in the form of the controller module's unique system ID (obtained from a node's NVRAM card or NVMEM board).

Disk ownership information displayed by Data ONTAP can take one or more of the following forms. Note that the names used vary slightly depending on the context.

- Owner (or Current owner)
This is the system that can currently access the disk.
- Original owner (or Home owner)
If the system is in HA takeover, then Owner is changed to the system that took over the node, and Original owner or Home owner reflects the system that owned the disk before the takeover.
- DR home owner
If the system is in a MetroCluster switchover, DR home owner reflects the value of the Home owner field before the switchover occurred.

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.

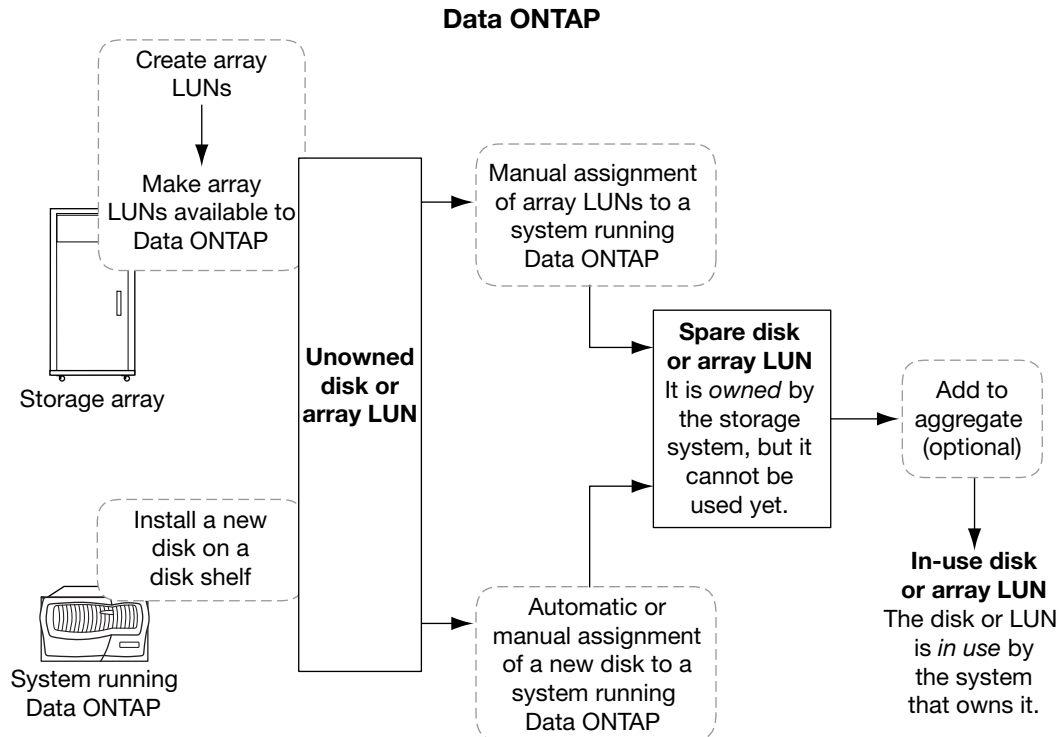
You assign ownership of a disk or array LUN to accomplish the following actions:

- 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.
- Associate the disk or array LUN with a specific SyncMirror pool (when SyncMirror is in use).
If SyncMirror is not in use, all disks and array LUNs are in pool0.

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 and pool membership for disks.

Automatic disk autoassignment uses the configured autoassignment policy to determine the correct ownership for disks. The autoassignment policy can be **stack**, **shelf**, **bay**, or **default**. In most cases, all disks in a stack (or loop) can be assigned to the same node and pool.

For mid- and high-end platforms, the **default** autoassignment policy is equivalent to the **stack** policy. For entry-level platforms (FAS2xxx), the **default** autoassignment policy is equivalent to the

bay policy, which supports two controllers sharing the same shelf. For this policy, disks in odd bay numbers are assigned together, and disks in even bay numbers are assigned together.

For partitioned disks, autoassignment operates on the container disk and also both of the partitions.

If you decide to change the way Data ONTAP has assigned the disks, you can do so at any time. You can also change autoassignment to use a different assignment policy, although doing so does not affect currently assigned disks.

If you need to temporarily remove disk ownership for one or more disks while you perform an administrative task, you must disable automatic ownership assignment first to prevent Data ONTAP from immediately reassigning ownership for those disks.

Automatic ownership assignment is not available for array LUNs or virtual disks.

Which disk autoassignment policy to use

Usually, you can use the default autoassignment policy, which is equivalent to the **stack** policy for most systems, and to the **bay** policy for entry level systems (FAS2xxx). However, for some configurations, you might need to change the autoassignment policy.

Use the appropriate autoassignment, based on your configuration:

If you are using...	Then use the autoassignment policy value of...
Stand-alone entry level system	stack
Entry level systems in an HA configuration with a single, shared shelf	bay
Entry level systems in an HA configuration with one stack of two or more shelves	shelf
MetroCluster configurations with one stack per node, two or more shelves	shelf
All other configurations	stack

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, automatically assign ownership for all the virtual data disks you defined during the initial system

setup. Automatic assignment is not available for any virtual disks you add after 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 maximize fault isolation and to keep automatic ownership assignment working. The guidelines are impacted by your autoassignment policy.

Use these guidelines when you assign ownership for disks:

- Assign disks so that your autoassignment policy group is homogenous.
For example, if your autoassignment policy is **stack** (the default for most platforms), assign all disks on the same stack or loop to the same node and pool. However, if your autoassignment policy is **bay** (the default for FAS2xxx models), keep your disks in even-numbered bays and odd-numbered bays homogenous by node and pool.
- If your autoassignment policy is **stack**, assign all stacks connected to the same adapter to the same pool.
- Assign disks in the same multi-disk carrier to the same node and pool.

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

This procedure is only for disks that are not partitioned.

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:

```
storage disk show -container-type unassigned
```

2. Assign each disk:

```
storage disk assign -disk disk_name -owner owner_name
```

Related concepts

[How automatic ownership assignment works for disks](#) on page 46

Related tasks

[Assigning ownership for disks partitioned for root-data partitioning](#) on page 49

Assigning ownership for disks partitioned for root-data partitioning

For partitioned disks, there are three different entities for ownership: the container disk, the data partition, and the root partition. You can set the ownership of the container disk or the partitions manually or by using autoassignment—just as you do for unpartitioned disks.

About this task

You can think of the three owned entities (the container disk and the two partitions) as being collectively owned by the HA pair. The container disk and the two partitions do not all need to be owned by the same node in the HA pair as long as they are all owned by one of the nodes in the HA pair. However, when you use a partition in an aggregate, it must be owned by the same node that owns the aggregate.

Steps

1. Display the current ownership for the partitioned disk:
`storage disk show -disk disk_name -partition-ownership`
2. Enter the appropriate command, depending on which ownership entity you want to assign ownership for:

If you want to assign ownership for the...	Use this command...
Container disk	<code>storage disk assign -disk <i>disk_name</i> -owner <i>owner_name</i></code>
Data partition	<code>storage disk assign -disk <i>disk_name</i> -owner <i>owner_name</i> -data true</code>
Root partition	<code>storage disk assign -disk <i>disk_name</i> -owner <i>owner_name</i> -root true</code>

If any of the ownership entities are already owned, then you must include the `-force` option.

Related concepts

[How automatic ownership assignment works for disks](#) on page 46

Related tasks

[Assigning ownership for disks](#) on page 48

Removing ownership from a disk

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

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 enabled, Data ONTAP could automatically reassign ownership before you remove the disk from the node. For this reason, you disable automatic ownership assignment until the disk is removed, and then reenabling it.

Steps

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

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

2. If needed, repeat the previous step for the node's HA partner.

3. Remove the software ownership information from the disk:

```
storage disk removeowner disk_name
```

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 the disk is partitioned for root-data partitioning, remove ownership from the partitions by entering both of the following commands:

```
storage disk removeowner disk_name -root true
```

```
storage disk removeowner disk_name -data true
```

Both partitions are no longer owned by any node.

5. 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
```

6. If needed, repeat the previous step for the node's HA partner.

Configuring automatic ownership assignment of disks

You can configure Data ONTAP to automatically assign disk ownership according to a disk's stack, shelf, or bay.

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

The behavior of the **default** autoassignment policy depends on the system model. For entry level models, the **default** policy is equivalent to the **bay** policy. For all other systems, it is equivalent to the **stack** policy.

Steps

1. The action you take depends on whether you want to set up automatic ownership assignment at the stack (or loop), shelf, or bay 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-policy stack
Configure automatic ownership assignment at the shelf level	storage disk option modify -autoassign-policy shelf
Configure automatic ownership assignment at the bay level	storage disk option modify -autoassign-policy bay
Turn off automatic ownership assignment	storage disk option modify -autoassign off

2. Verify the automatic assignment settings for the disks:

storage disk option show

Example

```
cluster1::> storage disk option show
```

Node Assign Policy	BKg. FW. Upd.	Auto Copy	Auto Assign	Auto
-----	-----	-----	-----	
cluster1-1	on	on	on	default
cluster1-2	on	on	on	default

Related concepts

[How automatic ownership assignment works for disks](#) on page 46

[Which disk autoassignment policy to use](#) on page 47

[When automatic ownership assignment is invoked](#) on page 47

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 to ensure that you operate on the correct set of disks.

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

- **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, for example, when you want to assign all of the disks attached to a particular port or switch.

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 operate on all disks on shelf 1 of stack 1, you would use the following syntax:

`1.1.*`

However, if you left off the second “.”, as in the following snippet, you would operate on all disks in stacks 1, 10, 11, and so on:

`1.1*`

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 54

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*. The capability of using array LUNs continues to be available as a licensed feature in Data ONTAP.

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 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.)

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

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.
[NetApp Support](#)

About this task

You need not perform this procedure if the license key for the V_StorageAttach package is already installed. If the Data ONTAP system is ordered with disks, the factory typically 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: mysystemla
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 and what pool a disk or array LUN is associated with. 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:

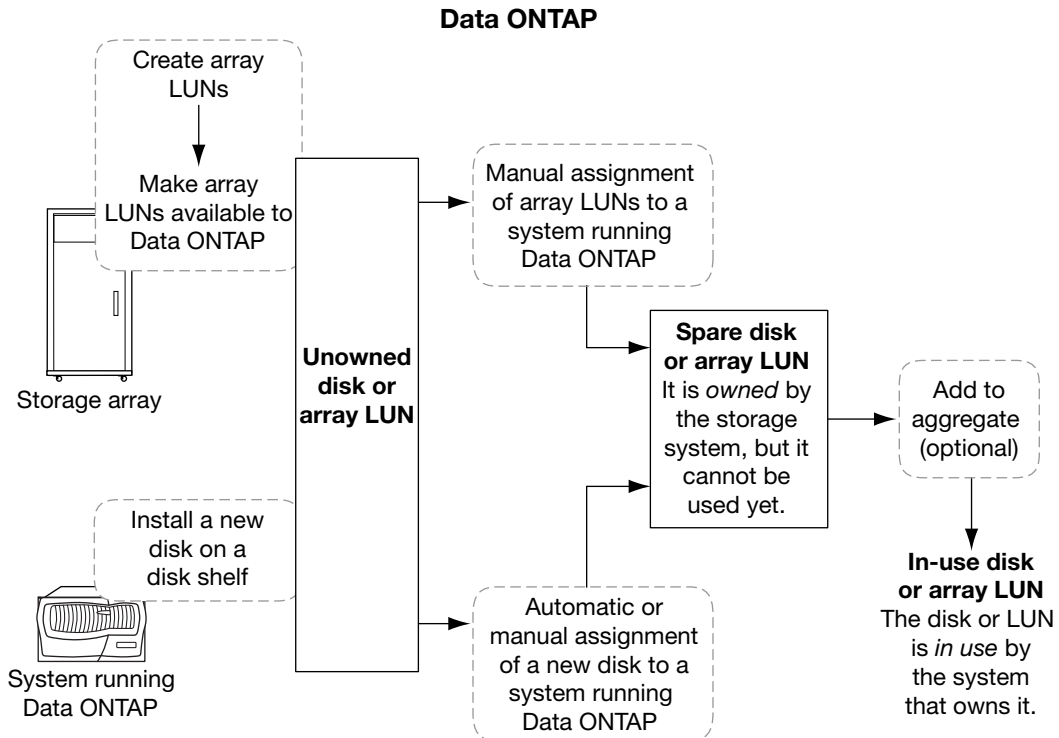
- Associate the disk or array LUN with a specific storage system.
For a stand-alone system, all disks and array LUNs are owned by that system. In an HA configuration, the disks and array LUNs can be owned by either system.

- 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.
- Associate the disk or array LUN with a specific SyncMirror pool (when SyncMirror is in use).
If SyncMirror is not in use, all disks and array LUNs are in pool0.

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:

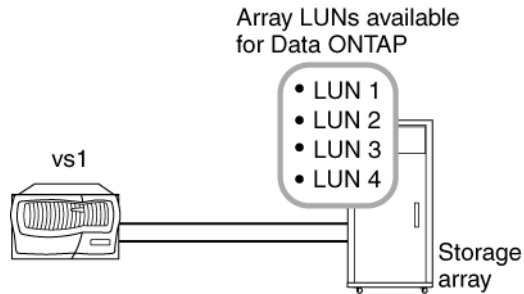
- You ordered your Data ONTAP system with native disk shelves and did not set up your system to work with array LUNs initially
- You left some LUNs that the storage array presented to Data ONTAP unowned, and you now need to use the storage
- Another system released the ownership of a particular array 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

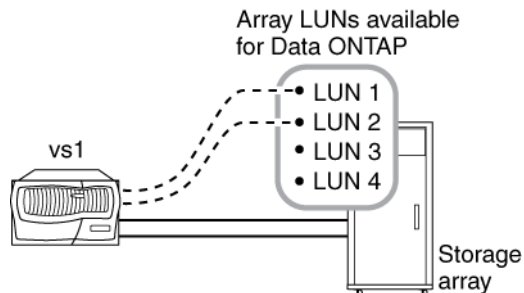
In this example, the storage array administrator made the array LUNs available to Data ONTAP. However, system vs1 has not yet been configured to “own” any of the LUNs. Therefore, it cannot read data from or write data to any array LUNs on the storage array:



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.

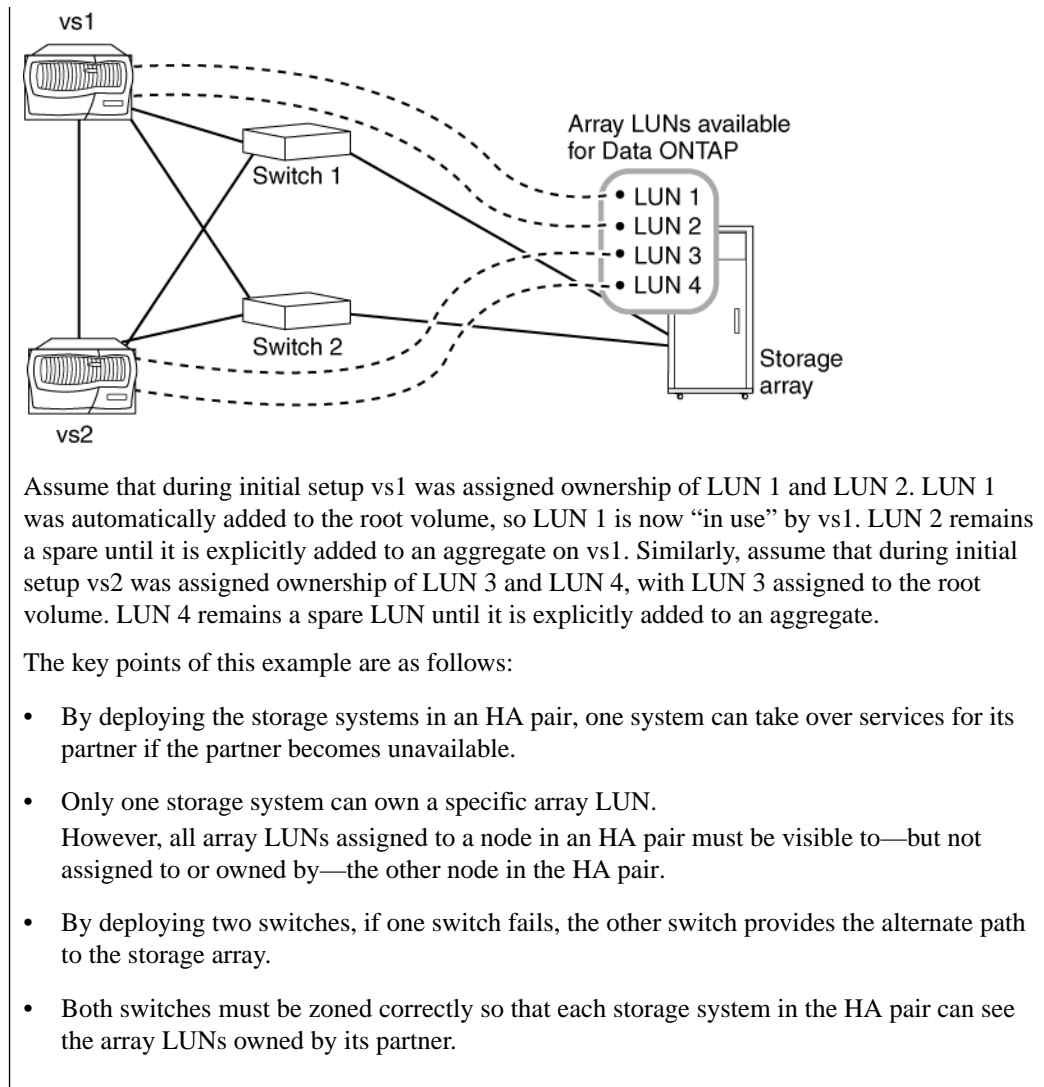


After you perform initial setup of the storage system, you could configure vs1 to also own LUN 3, LUN 4, both, or neither, depending on your storage needs.

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.



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.

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.

Related concepts

[How ownership for disks and array LUNs works](#) on page 54

Related tasks

[Modifying assignment of spare array LUNs](#) on page 60

Verifying the back-end configuration

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

The `storage array config show` 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 config show` 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 the `storage errors show` command.

Related information

[FlexArray Virtualization Implementation Guide for Third-Party Storage](#)

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 54

Related tasks

[Assigning ownership of array LUNs](#) on page 58

Array LUN name format

In Data ONTAP 8.3, the name assigned to an array LUN has a new format to ensure that the name is unique across a cluster.

The array LUN name consists of two components and looks like the following:

`<array_prefix>.<offset>`, for example `EMC-1.1`.

- The `array_prefix` is a unique prefix that Data ONTAP assigns by default to each storage array. This field is composed of `<array_name-array_instance>` (`EMC-1` in this case). `array_name` can be denoted by the first three letters of the name of the vendor.

If there is more than one array from the same vendor, the value of *array_instance* proceeds in an ascending order.

- The offset is the ascending virtual disk number that Data ONTAP assigns to each LUN. It is independent of the LUN ID of the host.

You can modify the *<array_prefix>* field by using the `storage array modify -name -prefix` command.

Pre-cluster array LUN name format

Before a node joins to a cluster or when the system is in the maintenance mode, the array LUN name follows a format used before Data ONTAP 8.3, the *pre-cluster* format.

In this 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 Data ONTAP systems

Configuration	Array LUN name format	Component descriptions
Direct-attached	<i>node-name.adapter.idlun-id</i>	<p><i>node-name</i> 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.</p> <p><i>adapter</i> is the adapter number on the Data ONTAP system.</p> <p><i>id</i> is the channel adapter port on the storage array.</p> <p><i>lun-id</i> is the array LUN number that the storage array presents to hosts.</p> <p>Example: <i>node1.0a.0L1</i></p>

Configuration	Array LUN name format	Component descriptions
Fabric-attached	<i>node-name:switch-name:port.idlun-id</i>	<p><i>node-name</i> 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.</p> <p><i>switch-name</i> is the name of the switch.</p> <p><i>port</i> is the switch port that is connected to the target port (the end point).</p> <p><i>id</i> is the device ID.</p> <p><i>lun-id</i> is the array LUN number that the storage array presents to hosts.</p> <p>Example: <i>node1:brocade3:6.126L1</i></p>

Related concepts

[Pre-cluster drive name formats](#) on page 13

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: The checksum type of all newly created aggregates using zoned checksum array LUNS is *advanced zoned checksum* (AZCS). Zoned checksum type 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:

```
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 an *advanced zoned checksum* (AZCS) aggregate.

Related tasks

[Changing the checksum type of an array LUN](#) on page 63

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 must 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 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.
- You cannot modify the checksum of array LUNs while assigning ownership. You can modify the checksum only on already assigned array LUNs.

Step

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

```
storage disk assign -disk disk name -owner owner -c new_checksum_type
```

disk name is the array LUN whose checksum type you want to change.

owner is the node to which the array LUN is assigned.

new_checksum_type can be **block** or **zoned**.

Example

```
storage disk assign -disk EMC-1.1 -owner 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 62

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 64

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 arrayLUNname
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

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 63

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.

Securing data at rest with Storage Encryption

You can protect your data at rest from unauthorized access with Storage Encryption. To use this feature, you must understand what Storage Encryption is and how it works, how to set it up and manage it, and how you can destroy data on disks using Storage Encryption.

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 (SSL in this context can refer to SSL or TLS, its successor, per common usage) 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 87

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 therefore 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 88

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 88

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 of an individual node or HA pair in the cluster must have encryption functionality to be able to use Storage Encryption.
You cannot mix regular non-encrypting disks with self-encrypting disks in a node or HA pair. However, Storage Encryption-enabled HA pairs can coexist with non-Storage Encryption-enabled HA pairs in the same cluster.

- Storage Encryption is not supported in a clustered Data ONTAP MetroCluster configuration.
- 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.
- You should use the network interface e0m for communication with key management servers.
Depending on the storage controller model, certain network interfaces might not be available during the boot process for communication with key management servers.
- Storage Encryption supports a maximum of 128 authentication keys per cluster.
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.
- Storage Encryption supports KMIP 1.0 and 1.1 for communication with key management servers.

Related information

Interoperability Matrix Tool: mysupport.netapp.com/matrix

Setting up Storage Encryption

During initial setup, your storage system checks whether it is properly configured with self-encrypting disks and is running a version of Data ONTAP that supports Storage Encryption. If the check is successful, you can then launch the Storage Encryption setup wizard after completion of the storage system setup wizard.

Information to collect before configuring Storage Encryption

You must gather certain information to successfully set up Storage Encryption on the cluster.

Information to collect	Details	Required	Optional
Network interface name	You must provide the name of the network interface the cluster should use to communicate with external key management servers. Note: Do not configure 10-Gigabit network interfaces for communication with key management servers. You should use the network interface e0m.	x	
Network interface IP address	You must provide the IP address of the network interface. This IP address can be in either IPv4 or IPv6 format.	x	
IPv6 network prefix length	For IPv6 addresses, you must provide the network prefix length. You can enter the network prefix length separately when prompted after entering the IPv6 address.	x	
Network interface subnet mask	You must provide the subnet mask of the network interface.	x	
Network interface gateway IP address	You must provide the IP address for the network interface gateway.	x	

Information to collect	Details	Required	Optional
IP addresses for external key management servers	You must link the cluster to at least one external key management server. You should add two or more external key management servers to prevent having a single point of failure. If you add only one external key management server and it fails, you can lose access to your data. If you specify IPv6 addresses for external key management servers, you must also provide an IPv6 address for the cluster network interface.	x	
IP address for additional external key management servers	You should link the cluster to multiple additional external key management servers during setup for redundancy. These key management servers should be in a clustered configuration.		x
Port number for each external key management server	You can provide a custom port number that each key management server listens on. The port number must be the same for all key management servers. If you do not provide a port number, it defaults to port 5696.		x
Public SSL certificate for cluster	You must provide a public SSL certificate for the cluster to link it to the external key management server.	x	
Private SSL certificate for cluster	You must provide a private SSL certificate for the cluster.	x	
Public SSL certificate for external key management servers	You must provide a public certificate for the root certificate authority (CA) for the SSL server of each external key management server to link it to the cluster.	x	
Key tag name	You can provide a name that is used to identify all keys belonging to a particular cluster. The default key tag name is the cluster name.		x

Using SSL for secure key management communication

The cluster and key management servers use SSL connections to keep the communication between them secure. This requires you to obtain and install various SSL certificates for the cluster and each key management server before you can set up and configure Storage Encryption.

The SSL communication between the cluster and key management servers is based on a client-server relationship. The cluster running Data ONTAP is the SSL client, and the key management servers are the SSL servers.

To avoid issues when installing SSL certificates, you should first synchronize the time between the following systems:

- The server creating the certificates
- The key management servers
- The cluster

Requirements for SSL certificates

Before obtaining and installing SSL certificates, you must understand what certificates are required and their requirements.

SSL certificates for Storage Encryption must use the Privacy Enhanced Mail (PEM) Base-64 encoded X.509 format and follow a strict naming convention. The following table describes the required certificate types and naming conventions:

Certificate for...	Certificate type	Certificate file name
Storage system	Public	<code>client.pem</code>
Storage system	Private	<code>client_private.pem</code>
Key management server	Public	<code>key_management_server_ipaddress_CA.pem</code> <i>key_management_server_ipaddress</i> must be identical to the IP address of the key management server that you use to identify it when running the Storage Encryption setup program.

These public and private certificates are required for the storage system and key management servers to establish secure SSL connections with each other and verify each other's identities.

The certificates for the storage system are only used by the storage system's KMIP client.

The private certificate can be passphrase protected during creation. In this case, the Storage Encryption setup program prompts you to enter the passphrase.

If your key management server does not accept self-signed certificates, you also need to include the necessary certificate authority (CA) public certificate.

In an HA pair, both nodes must use the same public and private certificates.

If you want multiple HA pairs that are connected to the same key management server to have access to each other's keys, all nodes in all HA pairs must use the same public and private certificates.

Installing SSL certificates on the cluster

You install the necessary SSL certificates on the cluster by using the `security certificate install` command. The SSL certificates are required for securing the communication between the cluster and key management servers.

Before you begin

You must have obtained the public and private certificates that Data ONTAP will use as an SSL client, and the public certificate for the root certificate authority (CA) of the SSL server (the key management server).

The SSL client certificate must not be password protected.

The common name of certificates cannot contain an underscore character (`_`).

Steps

1. Install the SSL client certificate for the cluster:

```
security certificate install -vserver admin_svm_name -type client -  
subtype kmip-cert
```

It is important to use the `-subtype kmip-cert` parameter to ensure the certificate is installed properly for use with key management servers.

2. Enter the public and private certificate as prompted.

3. Install the public certificate for the root certificate authority (CA) of the SSL server (the key management server):

```
security certificate install -vserver admin_svm_name -type server-ca -  
subtype kmip-cert -kmip-server-ip key_management_server_ipaddress
```

If you are using the same root CA for multiple key management servers with IPv4 addresses, enter the subnet address that covers all key management server IP addresses. If they are on completely different networks, you can use the subnet address 0.0.0.0 as a wildcard.

If your key management servers use IPv6 addresses, you must use a separate root CA for each one.

Example

If your key management server IP addresses are 172.18.12.9, 172.18.248.123, and 172.18.99.97, and they all use the same root CA, add them at the same time by using the subnet address 172.18.0.0 instead:

```
security certificate install -vserver svml -type server-ca -subtype  
kmip-cert -kmip-server-ip 172.18.0.0
```

4. If you are linking multiple key management servers to the cluster and they use individual root CAs, repeat the previous step for each public certificate of each key management server.

You can link up to four key management servers.

Related information

[Clustered Data ONTAP 8.3 System Administration Guide](#)

Running the key management setup wizard

You launch the key management setup wizard by using the `security key-manager setup` command. You should run the key management setup wizard after you complete setup of the cluster and the storage volumes or when you need to change Storage Encryption key management settings after initial setup.

Before you begin

The necessary SSL client and server certificates must be installed.

About this task

This wizard configures the network settings necessary for each node to contact key management servers at boot time.

This command is only available to cluster administrators.

Steps

1. Launch the key management setup wizard from the storage system prompt:

```
security key-manager setup
```

You can use the `-node` parameter to specify the name of the node you want to configure key management settings on.

2. Complete the steps in the wizard to configure Storage Encryption key management settings.
3. Repeat these steps for each node in the cluster.

Example

The following command launches the key management setup wizard on node2 of the cluster:


```
cluster1::> security key-manager setup -node node2
```

Welcome to the key manager setup wizard, which will lead you through the steps to add boot information.

Enter the following commands at any time
 "help" or "?" if you want to have a question clarified,
 "back" if you want to change your answers to previous questions, and
 "exit" if you want to quit the key manager setup wizard. Any changes you made before typing "exit" will be applied.

Restart the key manager setup wizard with "security key-manager setup". To accept a default or omit a question, do not enter a value.

You will now be prompted for a subset of your network configuration setup. These parameters will define a pre-boot network environment, allowing secure connections to the registered key server.

Enter the TCP port number for KMIP server [5696]:
 Enter the network interface: e0c
 Would you like to configure an IPv4 address? {yes, no} [yes]:

Enter the IP address: 10.63.55.152
 Enter the netmask: 255.255.192.0
 Enter the gateway: 10.63.0.1
 Would you like to configure an IPv6 address? {yes, no} [no]:

Running self-encrypting disks in FIPS compliance mode

You can optionally enable Federal Information Processing Standards (FIPS) compliance mode on self-encrypting disks (SEDs) to comply with security policies that require this additional level of security. To configure FIPS compliance mode on SEDs you should understand what FIPS is, considerations for running SEDs in FIPS compliance mode, and how to enable FIPS compliance mode on SEDs.

What FIPS compliance mode means for self-encrypting disks

The Federal Information Processing Standards (FIPS) is a set of security requirements for cryptographic modules defined by the Information Technology Laboratory National Institute of Standards and Technology (NIST). FIPS PUB 140-2 defines a set of rules that cryptographic modules, in this case self-encrypting disks (SEDs), have to follow when running in FIPS compliance mode.

FIPS PUB 140-2 details several levels of security requirements. When you configure SEDs to run in FIPS compliance mode, they comply with the requirements documented in level 2.

Related information

[US National Institute of Standards and Technology: Security requirements for cryptographic modules \(FIPS publication 140-2\)](#)

Considerations for running self-encrypting disks in FIPS compliance mode

If you decide to configure self-encrypting disks (SEDs) to run in FIPS compliance mode, you must be aware of certain requirements and limitations to ensure proper operation.

- FIPS compliance mode is only supported in clustered Data ONTAP 8.3.1 and later.

If you decide to downgrade or revert to an earlier release, you must first disable FIPS compliance mode for all SEDs by using the `storage encryption disk modify -disk * -fips-key-id 0x0` command at the advanced privilege level.

- Storage Encryption SEDs support FIPS 140-2 level 2 compliance.
- If SEDs are in FIPS mode, you cannot set the authentication key to MSID (0x0).
- FIPS compliance mode can only be managed from the clustershell.
- Enabling FIPS compliance mode on SEDs automatically enables power-on lock protection, meaning the disks require authentication after being power-cycled.
- The FIPS authentication key can be different from the data encryption authentication key; they do not have to be identical.
- Sanitizing SEDs or resetting SEDs to factory-original settings automatically takes the disk out of FIPS compliance mode.

Configuring self-encrypting disks to run in FIPS compliance mode

To put self-encrypting disks (SEDs) into FIPS compliance mode, you assign a FIPS authentication key to the disk.

About this task

Configuring SEDs to run in FIPS compliance mode automatically enables power-on lock protection.

Step

1. Configure SEDs to run in FIPS compliance mode:

```
storage encryption disk modify -disk disk_id -fips-key-id  
fips_authentication_key_id
```

Example

The following command enables FIPS compliance mode on all SEDs in disk stack 2.10:

```
cluster1::> storage encryption disk modify -disk 2.10.* -fips-key-id  
6A1E21D80000000001000000000000005A1FB4EE8F62FD6D8AE6754C9019F35A
```

```
Info: Starting modify on 14  
disks.
```

```
      |  
View the status of the operation by using  
the      |  
      storage encryption disk show-status command.
```

Taking self-encrypting disks out of FIPS compliance mode

There are several scenarios where self-encrypting disks (SEDs) leave FIPS compliance mode. This can happen either automatically when you sanitize SEDs or reset them to their factory-original settings, or manually when you decide to set the FIPS key ID back to the manufacturer's secure ID using the `storage encryption disk modify` command.

Steps

1. Display a list of SEDs that are currently in FIPS compliance mode:

```
storage encryption disk show -fips
```

2. Perform one of the following actions:

If you want to take SEDs out of FIPS compliance mode and...	Enter...
Keep the disk in service and data protection in place	a. <code>set -privilege advanced</code> b. <code>storage encryption disk modify -disk <i>disk_id</i> -fips-key-id 0x0</code> c. <code>set -privilege admin</code>
Sanitize the disks	a. Migrate data to preserve to another aggregate. b. Destroy the aggregate. c. <code>storage encryption disk sanitize -disk <i>disk_id</i></code>
Reset the disks to factory original settings (only disks that have a PSID)	a. <code>set -privilege advanced</code> b. <code>storage encryption disk revert-to-original-state -disk <i>disk_id</i> -psid <i>disk_physical_secure_id</i></code> c. <code>storage disk fail -disk <i>disk_id</i> -immediate true</code> d. <code>storage disk unfail -disk <i>disk_id</i></code> e. <code>set -privilege admin</code>

3. Verify that the modified SEDs are no longer in FIPS compliance mode:

```
storage encryption disk show -fips
```

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.

Adding key management servers

You can use the `security key-manager add` command to link key management servers to the cluster. 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 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.

Step

1. Add a key management server:

```
security key-manager add -address key_management_server_ipaddress
```

You can add up to four key management servers.

Example

The following command adds a link from the cluster to the key management server with the IP address 172.18.119.12:

```
cluster1:> security key-manager add -address 172.18.119.12
Node: 0
Registering 1 key servers...
Found client CA certificate.
Registration successful.
Registration complete.
Node: 1
Key manager "172.18.119.12" is a duplicate.
Node: 2
Error: Command failed. Missing SSL certificate. Install the SSL
certificate for each key server that requires one.
Use the "security certificate install" command to add the
certificate and try the command again.
Node: 3
Registering 1 key servers...
Found client CA certificate.
Registration successful.
Registration complete.
```

Verifying key management server links

You can use the `security key-manager show` command to verify that all key management servers are successfully linked to the cluster. This command is useful for verifying proper operation and troubleshooting.

Steps

1. Perform one of the following actions:

If you want to...	Enter...
See all key management servers linked to the cluster	<code>security key-manager show</code>
Check the status of a specific key management server	<code>security key-manager show -status -address <i>key_management_server_ipaddress</i></code>
Check the status of all key management servers	<code>security key-manager show -status</code>

2. Check the output to confirm that all key management servers are linked and available as desired.

To add missing or remove unwanted key management servers, use the `security key-manager add` and `security key-manager delete` commands, respectively.

Example

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

```
cluster1:> security key-manager show -status
```

Node	Registered Key Manager	Status
node0	172.18.119.78	available
node0	172.18.128.182	available
node0	172.18.128.183	available
node0	fd20:8b1e:b255:8175::33	available

node1	172.18.119.78	available
node1	172.18.128.182	available
node1	172.18.128.183	available
node1	fd20:8b1e:b255:8175::33	available

Displaying key management server information

You can display information about the external key management servers associated with the cluster by using the `security key-manager show` command.

Step

1. Display the external key management servers associated with the cluster:

```
security key-manager show
```

If you specify the `-status` parameter, the command displays the current status of each key management server.

All external key management servers associated with the cluster are listed.

Example

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

```
cluster1:> security key-manager show
```

Node	Registered Key Manager
node0	172.16.132.118
	172.16.132.211
node1	172.16.132.118
	172.16.132.211

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 cluster, you can remove the key management server link to the cluster by using the `security key-manager delete` 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 cluster 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.

Deleting the root CA certificate of a key management server by using the `security certificate delete` command does not delete the key management server. If you do so, the key management server continues to function.

Step

1. Remove a key management server:

```
security key-manager delete -address key_management_server_ipaddress
```

Example

The following command removes the link between the cluster and the key management server with the IP address 172.18.119.12:

```
cluster1:> security key-manager delete -address 172.18.119.12
Node: 0
Key manager 172.18.119.12 registration will be removed from service.
Key manager registration successfully removed.

Node: 1
Error: Key manager "172.18.119.12" is not a registered key manager.

Node: 2
Error: Key manager "172.18.119.12" is not a registered key manager.

Node: 3
Key server 172.18.119.12 registration will be removed from service.
Key manager registration successfully removed.
```

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 are 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 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 enters 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 possible unintended data loss as a result of the SEDs locked permanently due to exceeding the safety limit of a certain number of consecutive failed authentication attempts. The limit and the type of lockout protection depends on the manufacturing specifications and type of SED:

SED type	Number of consecutive failed authentication attempts resulting in lockout	Lockout protection type when safety limit is reached
HDD	1024	Permanent. Data cannot be recovered, even when the proper authentication key becomes available again.
X440_PHM2800MCTO 800GB NSE SSDs with firmware revisions NA00 or NA01	5	Temporary. Lockout is only in effect until disk is power-cycled.
X577_PHM2800MCTO 800GB NSE SSDs with firmware revisions NA00 or NA01	5	Temporary. Lockout is only in effect until disk is power-cycled.
X440_PHM2800MCTO 800GB NSE SSDs with higher firmware revisions	1024	Permanent. Data cannot be recovered, even when the proper authentication key becomes available again.
X577_PHM2800MCTO 800GB NSE SSDs with higher firmware revisions	1024	Permanent. Data cannot be recovered, even when the proper authentication key becomes available again.
All other SSD models	1024	Permanent. Data cannot be recovered, even when the proper authentication key becomes available again.

For all SED types, a successful authentication resets the try count to zero.

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.

Displaying Storage Encryption disk information

You can display information about self-encrypting disks by using the `storage encryption disk 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.

The command is only available to cluster administrators.

Step

1. Display information about SEDs:

```
storage encryption disk show
```

By default, the command displays the data authentication key ID. If you have enabled FIPS-compliance mode and use the optional `-fips` parameter, the command displays the FIPS authentication key ID instead.

If you specify the optional `-disk` parameter and the specified disk path name matches both encrypting and non-encrypting disks, the command only displays encrypting disks.

The mode column in the output displays the data protection state of the disk. By specifying the `-protection-mode` parameter, you can limit the output to disks with the specified mode:

Protection mode	Description
open	The disk is not power-on lock protected.
data	The data is power-on lock protected but the disk is not in FIPS-compliance mode.
part	The disk is in FIPS-compliance mode but the data is not power-on lock protected.
full	The data is power-on lock protected and the disk is in FIPS-compliance mode.

The `storage encryption disk show` and `modify` commands support extended wildcard matching. For more information, see the `storage encryption disk show` man page.

Example

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

```
cluster::> storage encryption disk show

Disk      Mode Data Key ID
-----
0.0.0     open 0x0
0.0.1     data 0x0
0.0.2     data 0A9C9CFC000000000100000000000000345CFD1BAD310CA8EDB377D439FB5C9A
1.10.0    open 0A53ED2A000000000010000000000000BEDC1B27AD3F0DB8891375AED2F34D0B
1.10.1    part 0A9C9CFC00000000001000000000000000345CFD1BAD310CA8EDB377D439FB5C9A
1.10.2    full 0A9C9CFC00000000001000000000000000345CFD1BAD310CA8EDB377D439FB5C9A
```

Changing the authentication key

You can change the authentication key of self-encrypting disks (SEDs) at any time by using the `security key-manager create-key` command to create a new key and the `storage encryption disk modify` command to assign the new key to SEDs. You might want to change the authentication key as part of your security protocol or when moving an aggregate to another cluster.

Steps

1. Verify that key management servers are configured and available on all nodes in the cluster:
security key-manager show -status
2. If you want to create a new authentication key, perform one of the following actions; otherwise, to change to an existing authentication key, skip to step 3.

If you want to...	Then...
Create a new authentication key manually	<ol style="list-style-type: none"> a. Enter the following command: security key-manager create-key -prompt-for-key true -key_tag key_tag b. When prompted, enter the new authentication key. It must be 20 to 32 characters long.

If you want to...	Then...
Create a new authentication key automatically	Enter the following command: security key-manager create-key -key_tag <i>key_tag</i>

key_tag is the label used to associate keys with a particular cluster. If you do not specify a key tag, Data ONTAP uses the cluster name by default.

3. Make a note of the new authentication key ID.

4. Assign a new authentication key to disks:

```
storage encryption disk modify -disk disk_id -data-key-id  
authentication_key_id
```

5. Verify that the authentication key was changed successfully:

```
storage encryption disk show
```

6. Optional: Assign a new FIPS authentication key:

```
storage encryption disk modify -disk disk_id -fips-key-id  
fips_authentication_key_id
```

The FIPS authentication key ID can be different from the data authentication key ID above.

7. Verify that the FIPS authentication key was changed successfully:

```
storage encryption disk show -fips
```

Example

The following command creates a new authentication key automatically:

```
cluster1::> security key-manager create-key

Node: node0
Creating authentication key...
Authentication key creation successful.
Key ID:
6A7A9FEC6A7A9C3C0101000000000006B2CC4A92B8F6EB1FEBB6887C78D82A7

Node: node0
Key manager restore operation initialized.
Successfully restored key information.

Node: node1
Key manager restore operation initialized.
Successfully restored key information.
```

The following command changes the authentication key on all SEDs in disk stack 1 to the new authentication key:

```
cluster1::> storage encryption disk modify -disk 1.* -data-key-id  
6A7A9FEC6A7A9C3C0101000000000006B2CC4A92B8F6EB1FEBB6887C78D82A7
```

Retrieving authentication keys

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

About this task

When you create a new authentication key, Data ONTAP automatically initiates a key restore operation so that the new key is available on all nodes. You only have to retrieve authentication keys manually if you are adding a new node to the cluster or restoring a node that was temporarily down to ensure that the node has all the current authentication keys available.

Step

1. Retrieve authentication keys from all linked key management server to all nodes of the cluster:

`security key-manager restore`

Unless specified otherwise, by default the command retrieves authentication keys on all nodes of the cluster from all linked key management servers. If you want to retrieve keys only on a specific node, specify the `-node` parameter with the node name. If you want to retrieve keys only from a specific key management server, specify the `-address` parameter with the IP address of the key management server.

Unless specified otherwise, by default the command displays the results from all key management servers. If you specify the `-key-manager-server-status` parameter, the command output displays only the keys retrieved from the key management servers with the specified status

Examples

The following command restores keys on all nodes from the key management server with the IP address 172.18.119.12:

```
cluster1::> security key-manager restore -address 172.18.119.12
Node: 0
Key Manager: 172.18.119.12
Status: OK
Key IDs
-----
8C7CFD000000000000000008123456789ABCDEF0123456789ABCDEF456789ABCDEF
8C7CFD000000000000000008123456789ABBC0211124DE123FABB129412404EABAB

Node: 1
Key Manager: 172.18.119.12
Status: OK
Key IDs
-----
8C7CFD000000000000000008123456789ABCDEF0123456789ABCDEF456789ABCDEF
8C7CFD000000000000000008123456789ABBC0211124DE123FABB129412404EABAB

Node: 2
"172.18.119.12" is not a registered key manager.

Node: 3
"172.18.119.12" is not a registered key manager.
```

Deleting an authentication key

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

Steps

1. Ensure 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 cluster.
2. See the documentation for the external key management server for details about how to delete stored authentication keys.

Authentication keys can be created using Data ONTAP. However, they cannot be deleted from key management servers using Data ONTAP.

SSL issues due to expired certificates

If the SSL certificates used to secure key management communication between the cluster and key management servers expire, the cluster 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 cluster and the key management servers fail, the cluster no longer can retrieve authentication keys when needed, and data access to the SEDs fails, resulting in cluster 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 cluster, and then install the new certificates on the cluster.

Steps

1. [Removing old SSL certificates before installing new ones](#) on page 83
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 cluster](#) on page 84
After you remove the old certificates, you create the new replacement SSL certificates and then install them on the cluster.

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. Remove the IP address of the key management server whose SSL certificate you want to update:
`security key-manager delete -address key_management_server_ipaddress`
2. Remove the cluster's client certificates:

```
security certificate delete -type client -vserver admin_svm_name -
common-name fqdn_or_custom_common_name -ca certificate_authority -
subtype kmip-cert
```

3. Remove all installed key management server certificates by entering the following commands for each key management server:

```
security certificate delete -type server-ca -vserver admin_svm_name -
common-name fqdn_or_custom_common_name -ca certificate_authority -
subtype kmip-cert
```

Installing replacement SSL certificates on the cluster

After you remove the old certificates, you create the new replacement SSL certificates and then install them on the cluster.

Before you begin

- You must have removed the old certificates that are about to expire from the cluster.
- You must have obtained the replacement public and private certificates for the cluster and the public certificate for the key management server.
- 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. Install the SSL client certificate for the cluster:

```
security certificate install -vserver admin_svm_name -type client -
subtype kmip-cert
```

2. Enter the public and private certificate as prompted.

3. Install the public certificate of the key management server:

```
security certificate install -vserver admin_svm_name -type server-ca -
subtype kmip-cert -kmip-server-ip key_management_server_ipaddress
```

4. If you are linking multiple key management servers to the cluster, repeat the previous step for each public certificate of each key management server.

5. Add a key management server:

```
security key-manager add -address key_management_server_ipaddress
```

You can add up to four key management servers.

6. Verify connectivity between the cluster and key management servers:

```
security key-manager show -status
```

Related information

[Clustered Data ONTAP 8.3 System Administration Guide](#)

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. Set the privilege level to advanced:

```
set -privilege advanced
```

2. Optional: If SEDs are running in FIPS-compliance mode, you must first change the FIPS authentication key back to the default MSID:

```
storage encryption disk modify -disk * -fips-key-id 0x0
```

Otherwise, skip to the next step.

3. Change the authentication key for all SEDs on the storage system back to the default MSID:

```
storage encryption disk modify -disk * -data-key-id 0x0
```

4. 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:

```
security key-manager delete -address key_management_server_ipaddress
```

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.

5. 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:

```
security certificate show -subtype kmip-cert
```

```
security certificate delete -type server-ca -vserver admin_svm_name -  
common-name fqdn_or_custom_common_name -ca certificate_authority -  
subtype kmip-cert
```

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.

Returning self-encrypting disks to service when authentication keys are no longer available

If one or more self-encrypting disks (SEDs) that are set to non-MSID authentication keys become inaccessible due to the permanent loss of the authentication keys, you can recover the SEDs (though not the data stored on them) and return them to service.

About this task

Such SEDs will be in a broken state, and regardless of the method you use to return them to service, whatever data is currently stored on them will be lost afterward. Only use this process if you are absolutely sure that the authentication keys for the SEDs are permanently lost and that there is no chance to recover them.

There are several methods for returning such SEDs to service, depending on the following criteria:

- If the SEDs have not been put into FIPS compliance mode, or if the SEDs are in FIPS compliance mode and the FIPS authentication key is still available (only the data authentication key is lost), you can use the `storage encryption disk sanitize` command.
- If the SEDs are in FIPS compliance mode and the FIPS authentication key is not available either, you can use the `storage encryption disk revert-to-original-state` command, provided that the disk has a PSID printed on its label.

This requires accessing the physical disk to obtain the PSID from its printed label.

Step

1. Return the SEDs to service by using one of the following methods:

If the SEDs are...	Use these steps...
Not in FIPS compliance mode	<ol style="list-style-type: none"> Sanitize the broken disk: <code>storage encryption disk sanitize -disk <i>disk_id</i></code> Set the privilege level to advanced: <code>set -privilege advanced</code> Fail the sanitized disk: <code>storage disk fail -immediate true -disk <i>disk_id</i></code> Unfail the sanitized disk: <code>storage disk unfail -spare true -disk <i>disk_id</i></code> Verify that the SED is now a spare and ready to be reused in an aggregate: <code>storage disk show -disk <i>disk_id</i></code>
In FIPS compliance mode and the FIPS key is available	<ol style="list-style-type: none"> Sanitize the broken disk: <code>storage encryption disk sanitize -disk <i>disk_id</i></code>
In FIPS compliance mode, the FIPS key is not available, and the SEDs have a PSID printed on the label	<ol style="list-style-type: none"> Obtain the SED's PSID from its disk label. Set the privilege level to advanced: <code>set -privilege advanced</code> Reset the SED to its factory configured settings: <code>storage encryption disk revert-to-original-state -disk <i>disk_id</i> -psid <i>disk_physical_secure_id</i></code> Fail the sanitized disk: <code>storage disk fail -immediate true -disk <i>disk_id</i></code> Unfail the sanitized disk: <code>storage disk unfail -spare true -disk <i>disk_id</i></code> Verify that the SED is now a spare and ready to be reused in an aggregate: <code>storage disk show -disk <i>disk_id</i></code>

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 `storage encryption disk sanitize` command. This renders the data on the disk inaccessible, but the disk can be reused.

Steps

1. Migrate any data that needs to be preserved to a different aggregate.
2. Destroy the aggregate.
3. Identify the disk ID for the disk to be sanitized:

```
storage encryption disk show
```

4. Sanitize the disk:

```
storage encryption disk sanitize -disk disk_id
```

`disk_id` specifies the disks you want to sanitize. You can use wildcards to specify multiple disks.

By default, the command only sanitizes disks whose state is **spare** or **broken**. If you want to sanitize all specified disks regardless of state, specify the `-force-all-states true` parameter.

Example

The following command sanitizes a self-encrypting disk with the disk ID 1.10.20:

```
cluster1:>> storage encryption disk sanitize 1.10.20

Warning: This operation will cryptographically sanitize 1 spare or broken
self-encrypting disk on 1 node.
To continue, enter
  sanitize disk
:sanitize disk

Info: Starting sanitize on 1 disk.
View the status of the operation using the
storage encryption disk show-status command.
```

The following command sanitizes all self-encrypting disks in the cluster regardless of state:

```
cluster1:>> storage encryption disk sanitize -force-all-states -
disk *

Warning: This operation will cryptographically sanitize 96
self-encrypting disks on 4 nodes.
To continue, enter
  sanitize disk
:sanitize disk

Info: Starting sanitize on 96 disks.
View the status of the operation by using the
storage encryption disk show-status command.
```

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 `storage encryption disk destroy` command. This command only works on spare and broken disks.

About this task

The result of this operation is not reversible. After you perform this operation on a disk, you can no longer access the data on the disk or use the disk for data storage. You must use this command with extreme care.

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. Destroy the disk:

```
storage encryption disk destroy -disk disk_id
```

disk_id specifies the disks you want to destroy. You can use wildcards to specify multiple disks.

5. If the disk has a PSID printed on its label but you do not want the disk to be able to be reset to factory settings and returned to service at a later time, obliterate all instances of the PSID on the disk label (text or scannable code).

Be sure to also destroy any copies, scans, or photographs of the disk label.

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

The external key server must be configured 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	<ol style="list-style-type: none"> If the storage system is a node in an HA pair, disable takeover. Take all aggregates offline and destroy them. Halt the storage system. Boot into maintenance mode. Enter the following command: <pre>storage encryption disk sanitize -disk * - force-all-states true</pre> <p>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.</p>
Power is available to the storage system and you must shred the data immediately; time is critical	<ol style="list-style-type: none"> If the storage system is a node in an HA pair, disable takeover. Set the privilege level to advanced. Enter the following command: <pre>storage encryption disk sanitize -disk * - force-all-states true</pre> <p>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.</p>
Power is available to the external key server but not to the storage system	<ol style="list-style-type: none"> Log in to the external key server. Destroy all keys associated with the disks containing data to protect.
Power is not available to the external key server or the storage system	<p>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.</p>

What function the physical secure ID has for SEDs

Certain self-encrypting disks (SEDs) feature additional functionality to reset the disk to factory settings. These disks have a physical secure ID (PSID) printed on the disk label that is required to perform a factory reset.

The PSID is unique to each drive. It is printed on the disk label and visible to anyone with physical access to the SED. The PSID is not electronically retrievable from the SED. If the disk label is obliterated, the PSID is lost and cannot be recovered.

Using the PSID to perform a factory reset causes all disk parameters to be reset to factory original settings, including the following:

- The encryption key used to encrypt and decrypt the data on the media is changed to an unknown value.
If the SED contained data, access to the data is permanently lost. The new unknown encryption key cannot be retrieved. This operation cannot be undone.
- The authentication key required to authenticate to the SED is changed back to the MSID (the default security ID set by the manufacturer).

New data stored on the SED is not protected until the MSID is changed to a new secret authentication key.

- The SED can be returned into service, even if its state was previously set to end-of-life. If the SED was previously used but then set to end-of-life state using the `disk encrypt destroy` command, use of the PSID recovers the SED from this state and returns it to normal service state. However, it only returns it to factory original settings. It cannot in any way recover previously used encryption or authentication keys or restore access to previous data stored on the SED.
- The SED exits FIPS compliance mode.
Resetting the SED to factory original settings takes the SED out of FIPS compliance mode.

SEDs that have PSID functionality

There are several models of SEDs but only some of them have PSID functionality. Earlier SED models with PSID functionality do not have the PSID printed on the disk label and therefore cannot use it.

Use the following table for guidance to find out whether your SEDs have PSID functionality:

Disk model	Has PSID functionality	Has PSID printed on disk label	Comments
X414	No	No	SED cannot be reset to factory original settings using a PSID.
X415	No	No	SED cannot be reset to factory original settings using a PSID.
X416	Yes	Maybe	Check the label on the physical disk. If a PSID is printed on the label, the SED can be reset to factory original settings using the PSID.
X417	Yes	Maybe	Check the label on the physical disk. If a PSID is printed on the label, the SED can be reset to factory original settings using the PSID.
All other SEDs	Yes	Yes	Check the label on the physical disk to obtain the PSID.

Resetting an SED to factory original settings

If you previously set the state of an SED to end-of-life by using the `storage encryption disk destroy` command but now want to return it to service, you can reset it to its factory original settings by using the `storage encryption disk revert-to-original-state` command provided that the disk has a PSID printed on its label.

Before you begin

You must have obtained the SED's PSID from its disk label.

Steps

1. Set the privilege level to advanced:
`set -privilege advanced`
2. Reset the disk to its factory configured settings:

```
storage encryption disk revert-to-original-state -disk disk_id -psid  
disk_physical_secure_id
```

The PSID is the physical secure ID printed on the disk label that is required to perform a factory reset.

3. Make the disk available again to the storage system:

```
storage disk unfail -disk disk_id
```

4. Verify that the operation was successful:

```
storage encryption disk show
```

The disk that you reset shows up as a spare disk.

5. Return to the admin privilege level:

```
set -priv admin
```

Example

The following command returns the disk 1.10.0 to its original manufactured state:

```
cluster1::> storage encryption disk revert-to-original-state -disk  
01.10.0 -psid AC65PYF8CG45YZABUQJKM98WV2VZGRLD
```

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.

Note: Data ONTAP supports RAID4 and RAID-DP on native disk shelves but supports only RAID0 on 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.

Protection provided by RAID and SyncMirror

Combining RAID and SyncMirror provides protection against more types of drive failures than using RAID alone.

You can use RAID in combination with the SyncMirror functionality, which also offers protection against data loss due to drive or other hardware component failure. SyncMirror protects against data

loss by maintaining two copies of the data contained in the aggregate, one in each plex. Any data loss due to drive failure in one plex is repaired by the undamaged data in the other plex.

For more information about SyncMirror, see the *Clustered Data ONTAP Data Protection Guide*.

The following tables show the differences between using RAID alone and using RAID with SyncMirror:

Table 1: RAID-DP and SyncMirror

Criteria	RAID-DP alone	RAID-DP with SyncMirror
Failures protected against	<ul style="list-style-type: none"> Single-drive failure Double-drive failure within a single RAID group Multiple-drive failures, as long as no more than two drives within a single RAID group fail 	<ul style="list-style-type: none"> All failures protected against by RAID-DP alone Any combination of failures protected against by RAID-DP alone in one plex, concurrent with an unlimited number of failures in the other plex Storage subsystem failures (HBA, cables, shelf), as long as only one plex is affected
Failures <i>not</i> protected against	<ul style="list-style-type: none"> Three or more concurrent drive failures within a single RAID group Storage subsystem failures (HBA, cables, shelf) that lead to three or more concurrent drive failures within a single RAID group 	<ul style="list-style-type: none"> Three or more concurrent drive failures in a single RAID group on both plexes
Required resources per RAID group	n data drives + 2 parity disks	$2 \times (n \text{ data drives} + 2 \text{ parity drives})$
Performance cost	Almost none	Low mirroring overhead; can improve performance
Additional cost and complexity	None	SyncMirror license and configuration

Table 2: RAID4 and SyncMirror

Criteria	RAID4 alone	RAID4 with SyncMirror
Failures protected against	<ul style="list-style-type: none"> Single-disk failure Multiple-disk failures, as long as no more than one disk within a single RAID group fails 	<ul style="list-style-type: none"> All failures protected against by RAID4 alone Any combination of failures protected against by RAID4 alone in one plex, concurrent with an unlimited number of failures in the other plex Storage subsystem failures (HBA, cables, shelf), as long as only one plex is affected
Failures <i>not</i> protected against	<ul style="list-style-type: none"> Two or more concurrent drive failures within a single RAID group Storage subsystem failures (HBA, cables, shelf) that lead to two or more concurrent drive failures within a single RAID group 	<ul style="list-style-type: none"> Two or more concurrent drive failures in a single RAID group on both plexes
Required resources per RAID group	n data drives + 1 parity drive	$2 \times (n \text{ data drives} + 1 \text{ parity drive})$
Performance cost	None	Low mirroring overhead; can improve performance
Additional cost and complexity	None	SyncMirror configuration and extra storage requirement

Table 3: RAID0 and SyncMirror

Criteria	RAID0 alone	RAID0 with SyncMirror
Failures protected against	No protection against any failures RAID protection is provided by the RAID implemented on the storage array.	Any combination of array LUN, connectivity, or hardware failures, as long as only one plex is affected
Failures <i>not</i> protected against	No protection against any failures RAID protection is provided by the RAID implemented on the storage array.	Any concurrent failures that affect both plexes
Required array LUN resources per RAID group	No extra array LUNs required other than n data array LUNs	$2 \times n$ data array LUNs
Performance cost	None	Low mirroring overhead; can improve performance

Criteria	RAID0 alone	RAID0 with SyncMirror
Additional cost and complexity	None	SyncMirror configuration and extra storage requirement

Understanding RAID drive types

Data ONTAP classifies drives (or, for partitioned drives, *partitions*) as one of four types for RAID: data, hot spare, parity, or dParity. You manage disks differently depending on whether they are spare or being used in an aggregate.

The RAID type is determined by how RAID is using a drive or partition; it is different from the Data ONTAP disk type.

You cannot affect the RAID type for a drive. The RAID type is displayed in the `Position` column for many storage commands.

For drives using root-data partitioning and SSDs in storage pools, a single drive might be used in multiple ways for RAID. For example, the root partition of a partitioned drive might be a spare partition, whereas the data partition might be being used for parity. For this reason, the RAID drive type for partitioned drives and SSDs in storage pools is displayed simply as *shared*.

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.

Considerations for sizing RAID groups

Configuring an optimum RAID group size requires a trade-off of factors. You must decide which factors—speed of RAID rebuild, assurance against risk of data loss due to drive failure, optimizing I/O performance, and maximizing data storage space—are most important for the aggregate that you are configuring.

When you create larger RAID groups, you maximize the space available for data storage for the same amount of storage used for parity (also known as the “parity tax”). On the other hand, when a disk fails in a larger RAID group, reconstruction time is increased, impacting performance for a longer period of time. In addition, having more disks in a RAID group increases the probability of a multiple disk failure within the same RAID group.

HDD or array LUN RAID groups

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

- All RAID groups in an aggregate should have a similar number of disks.
The RAID groups do not have to be exactly the same size, but you should avoid having any RAID group that is less than one half the size of other RAID groups in the same aggregate when possible.
- 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 aggregates

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

- All RAID groups in an aggregate should have a similar number of drives.
The RAID groups do not have to be exactly the same size, but you should avoid having any RAID group that is less than one half the size of other RAID groups in the same aggregate when possible.
- For RAID-DP, the recommended range of RAID group size is between 20 and 28.

Related references

[Storage limits](#) on page 163

Customizing the size of your RAID groups

You can customize the size of your RAID groups to ensure that your RAID group sizes are appropriate for the amount of storage you plan to include for an aggregate.

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.

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

- By default, if the number of disks or array LUNs in the most recently created RAID group is less than the new RAID group size, 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 can never cause a RAID group to become larger than the current maximum RAID group size for the aggregate.
- You cannot decrease the size of already created RAID groups.
- The new size applies to all RAID groups in that aggregate (or, in the case of a Flash Pool aggregate, all 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 maximum RAID group size for the SSD RAID groups of a Flash Pool aggregate	<code>storage aggregate modify -aggregate <i>aggr_name</i> -cache-raid-group-size <i>size</i></code>
Change the maximum size of any other RAID groups	<code>storage aggregate modify -aggregate <i>aggr_name</i> -maxraidsz <i>size</i></code>

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 -maxraidsz 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
```

Related concepts

[Considerations for sizing RAID groups](#) on page 97

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.

Note: It is best to use the default RAID group size for array LUNs. The default RAID group size is adequate for most organizations. 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.

Minimum number of 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. A spare disk is also required to provide important information (a *core file*) to technical support in case of a controller disruption.

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. However, if any disk in those RAID groups fails, then no spare is available for any future disk failures, or for a core file, until the spare is replaced. For this reason, having more than one spare is recommended.

Related concepts

[Spare requirements for multi-disk carrier disks](#) on page 24

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.
- If SyncMirror is in use, the spare must be in the same pool as the disk it is replacing.
- 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 the failed disk is part of a mirrored aggregate and there are no hot spares available in the correct pool, Data ONTAP uses a spare from the other pool.
Using drives from the wrong pool is not optimal because you no longer have fault isolation for your SyncMirror configuration.

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 102

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 102

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 102

[About degraded mode](#) on page 101

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.

Changing the schedule for automatic RAID-level scrubs

You can change the start time and duration of the scheduled scrubs if you want your data to be scrubbed more often than the default RAID-level scrub schedule allows.

About this task

The default RAID-level scrub schedule depends the platform model:

- For All Flash FAS platforms, the default schedule starts a scrub on Sunday at 1:00 a.m. that runs for six hours.
- For all other platforms, the default schedule starts a scrub every day at 1:00 a.m. On Sundays, the scrub runs for 12 hours; all other days it runs for four hours.

Steps

1. Update the RAID-level scrub schedule:

```
storage raid-options modify -node nodename -name raid.scrub.schedule -
value duration[h/m]@weekday@start_time,[duration[h|
m]@weekday@start_time,...]
```

- `duration` is specified in either hours or minutes.
- `weekday` is the day of the week: **mon**, **tue**, **wed**, **thu**, **fri**, **sat**, or **sun**.

- `start_time` is the hour when the scrub should start, using 24-hour format.

The current automatic RAID-level scrub schedule is replaced by the schedule you specify.

2. Confirm your new schedule:

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

Example

The following command prevents any RAID-level scrubbing on node `sys1-a` from occurring on Mondays, and increases the time duration on Saturdays to eight hours:

```
storage raid-options modify -node sys1-a -name raid.scrub.schedule -
value 4h@tue@1,4h@wed@1,4h@thu@1,4h@fri@1,8h@sat@1,12h@sun@1
```

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.

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
- Plex resynchronization
- Synchronous mirror verification

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.reconstruct.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**.

Controlling the performance impact of plex resynchronization

Plex resynchronization ensures that both plexes of a mirrored aggregate are identical. You can control the performance impact of plex resynchronization by using the `raid.resync.perf_impact` option.

About this task

When plex resynchronization and RAID data reconstruction are running at the same time, Data ONTAP limits the combined resource utilization to the greatest 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 by both operations has a medium impact.

You should set this option to the same value for both nodes in an HA configuration.

Step

1. Enter the following command:

```
storage raid-options modify -node node_name raid.resync.perf_impact impact
```

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

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

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

The default impact is **medium**.

Controlling the performance impact of mirror verification

You use mirror verification to ensure that the two plexes of a synchronous mirrored aggregate are identical. You can control the speed of mirror verification, and its effect on system resources, by using the `raid.verify.perf_impact` option.

About this task

When mirror verification and RAID-level scrubbing are running at the same time, Data ONTAP limits the combined resource utilization to the greatest 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 of both operations has a medium impact.

For more information about synchronous mirroring, see the *Clustered Data ONTAP Data Protection Tape Backup and Recovery Guide*.

Step

1. Enter the following command:

```
storage raid-options modify -node node_name raid.verify.perf_impact  
impact
```

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

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

low means that the storage system uses very little of the system resources; this setting lightly affects storage system performance, but the mirror verification 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 mirrored or unmirrored.
- 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.

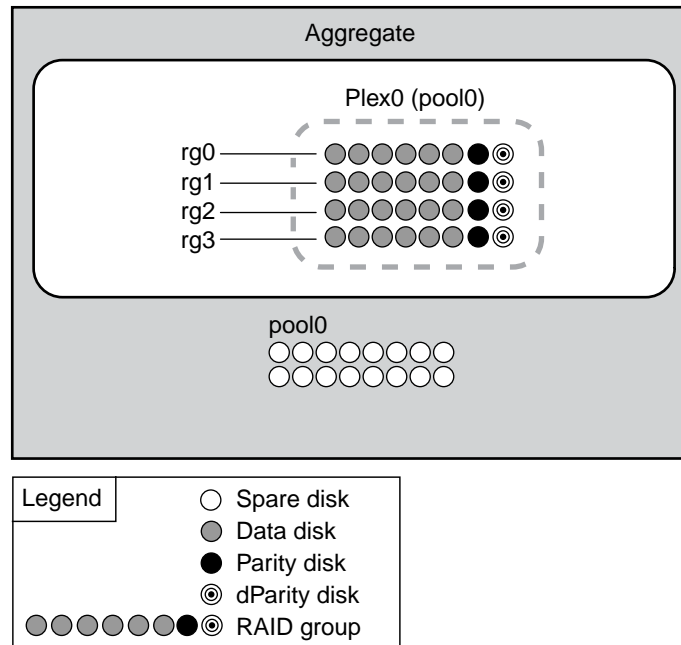
Related information

[*NetApp Technical Report 3437: Storage Subsystem Resiliency Guide*](#)

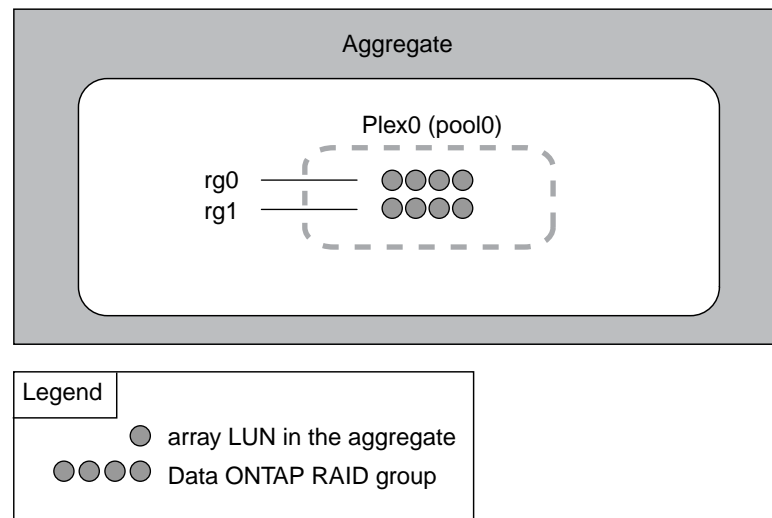
How unmirrored aggregates work

Unless you are using SyncMirror, all of your aggregates are unmirrored. Unmirrored aggregates have only one *plex* (copy of their data), which contains all of the RAID groups belonging to that aggregate.

The following diagram shows an unmirrored aggregate composed of disks, with its one plex. The aggregate has four RAID groups: rg0, rg1, rg2, and rg3. Each RAID group has 6 data disks, one parity disk, and one dparity (double parity) disk. All disks used by the aggregate come from the same pool, pool0.



The following diagram shows an unmirrored aggregate with array LUNs, with its one plex. It has two RAID groups, rg0 and rg1. All array LUNs used by the aggregate come from the same pool, pool0.



How mirrored aggregates work

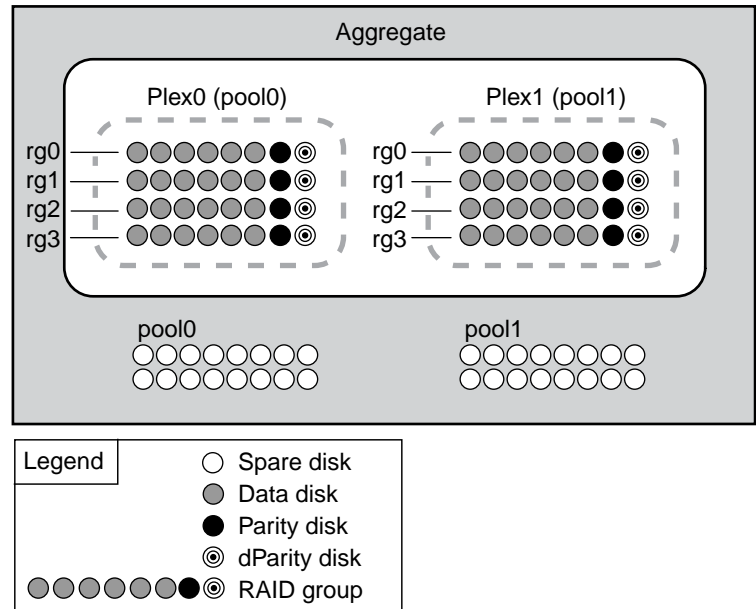
Mirrored aggregates have two *plexes* (copies of their data), which use the SyncMirror functionality to duplicate the data to provide redundancy.

When a mirrored aggregate is created (or when a second plex is added to an existing unmirrored aggregate), Data ONTAP copies the data in the original plex (plex0) to the new plex (plex1). The plexes are physically separated (each plex has its own RAID groups and its own pool), and the plexes are updated simultaneously. This provides added protection against data loss if more disks fail than the RAID level of the aggregate protects against or there is a loss of connectivity, because the unaffected plex continues to serve data while you fix the cause of the failure. After the plex that had a problem is fixed, the two plexes resynchronize and reestablish the mirror relationship.

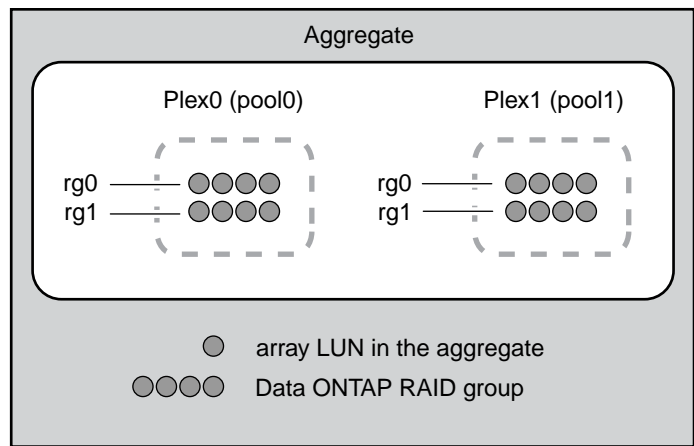
Note: The time for the two plexes to resynchronize can vary and depends on many variables such as aggregate size, system load, how much data has changed, and so on.

The disks and array LUNs on the system are divided into two pools: pool0 and pool1. Plex0 gets its storage from pool0 and plex1 gets its storage from pool1.

The following diagram shows an aggregate composed of disks with SyncMirror enabled and implemented. A second plex has been created for the aggregate, plex1. The data in plex1 is a copy of the data in plex0, and the RAID groups are also identical. The 32 spare disks are allocated to pool0 or pool1, 16 disks for each pool.



The following diagram shows an aggregate composed of array LUNs with SyncMirror enabled and implemented. A second plex has been created for the aggregate, plex1. Plex1 is a copy of plex0, and the RAID groups are also identical.



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 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 using physical SSDs](#) on page 143

How Flash Pool aggregates work

In general, Flash Pool aggregates are used and managed in a similar manner as standard aggregates. However, you need to understand how both the SSD and HDD RAID groups interact and affect the rest of the system.

The SSD RAID groups, also called the *SSD cache*, can be composed of physical SSDs or allocation units from SSD storage pools (but not both).

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. If the Flash Pool aggregate uses an SSD storage pool for its SSD cache, the cache RAID type can be changed only when the first SSD RAID groups are added, and the size of the SSD RAID groups are determined by the number of SSDs in the storage pool.

When you enable data compression manually for a volume in a Flash Pool aggregate, adaptive compression is enabled by default.

There is a platform-dependent maximum size for the SSD cache.

Related concepts

[Rules for mixing drive types in Flash Pool aggregates](#) on page 125

[Rules for mixing HDD types in aggregates](#) on page 124

[How the available Flash Pool cache capacity is calculated](#) on page 113

Related tasks

[Changing the RAID type of RAID groups in a Flash Pool aggregate](#) on page 150

Related information

[NetApp Hardware Universe](#)

[Clustered Data ONTAP 8.3 Logical Storage Management Guide](#)

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.

You cannot use the following types of storage objects in Flash Pool aggregates:

- SSDs that are partitioned for root-data partitioning
Partitioned HDDs can be used for the HDD RAID groups of a Flash Pool aggregate.
- Array LUNs

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.

You can use data compression on volumes associated with a Flash Pool aggregate. Compressed blocks are cached in the Flash Pool cache for both read and write operations.

Data in read-only volumes, such as SnapMirror or SnapVault destinations, is cached in the Flash Pool cache.

Flash Pool aggregates can be created from mirrored aggregates; however, the SSD configuration must be kept the same for both plexes.

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 do not support write caching.

Related information

[NetApp Hardware Universe](#)

[NetApp Technical Report 4070: Flash Pool Design and Implementation Guide](#)

Considerations for RAID type and spare management for Flash Pool cache

Because of the higher cost of SSDs, it is especially important to find the right balance between cost and protection against drive failure. Knowing how the performance and availability of Flash Pool aggregates are affected by SSDs becoming unavailable can help you determine the correct approach for your needs.

The failure or unavailability of an SSD affects its RAID group and aggregate the same way as for HDDs. When an SSD being used in a Flash Pool cache becomes unavailable, the RAID group that contains that SSD goes into degraded mode, and the cache performance for that Flash Pool aggregate is reduced until the RAID group can be reconstructed by copying the contents of the unavailable SSD to a spare SSD.

As with HDD RAID groups, if a cache RAID group experiences more concurrent SSD failures than its RAID level can correct for (3 for RAID-DP, 2 for RAID4), data integrity and availability can be compromised.

Therefore, for maximum data availability, you should use RAID-DP for both the HDD RAID groups and the cache. In addition, you should have a spare SSD available at all times so that reconstruction can begin immediately in the case of an SSD failure. The spare SSD can be shared between multiple Flash Pool aggregates, although all Flash Pool aggregates using that spare will be without a spare after an SSD fails until the spare can be replaced.

If using RAID-DP and providing a spare for the Flash Pool cache is prohibitively expensive, and if your RAID group size is not larger than the maximum RAID4 RAID group size, the next best alternative is to use RAID4 for the Flash Pool cache (you should still use RAID-DP for the HDD RAID groups), and provide a spare SSD. This is preferable to using RAID-DP with no spare, because double SSD failures are relatively rare, especially with a smaller RAID group size, and reconstruction takes less time than procuring and installing a spare.

Note: If you are using SSD storage pools, you cannot change the RAID type of the cache RAID groups after they have been added to an aggregate. If you believe that you might want to increase the size of your storage pools over time, you should consider the maximum size you will require when determining the correct RAID type to use for the SSD cache.

Comparison of Flash Pool aggregates and Flash Cache

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 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
Compressed blocks cached?	Yes	No
Supported with array LUNs?	No	Yes

How the available Flash Pool cache capacity is calculated

Flash Pool cache capacity cannot exceed a platform-dependent limit for the node or HA configuration. Knowing the available cache capacity enables you to determine how much cache capacity you can add to your Flash Pool aggregates before reaching the limit.

The current Flash Pool cache capacity is the sum of all of the Flash Pool caches (as reported by the `storage aggregate show` command). For nodes using SyncMirror, including MetroCluster configurations, only the Flash Pool cache of one plex counts toward the cache limit.

If Flash Cache modules are installed in a node, 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 nodes in an HA configuration, the cache size limit applies 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.

For nodes in a MetroCluster configuration, the cache size limit applies to the configuration as a whole, but it cannot be split arbitrarily between the nodes as it can for an HA configuration. This means that the limit for each node in the MetroCluster configuration is one quarter of the total cache size limit. If Flash Cache is present on any of the nodes, then the size of the largest Flash Cache module must be subtracted from the overall limit before the per-node limit is calculated.

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.

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 cache capacity for the HA pair would be 12 TB minus 3 TB, or 9 TB.

Calculation for a MetroCluster configuration

For a MetroCluster configuration composed of four storage controllers with a maximum cache capacity of 12 TB, the maximum Flash Pool cache capacity for each node would be 12 TB divided by four, or 3 TB.

Calculation for a MetroCluster configuration with Flash Cache modules

For a MetroCluster configuration composed of four storage controllers with a maximum cache capacity of 12 TB and 2 TB of Flash Cache installed on one node, the maximum Flash Pool cache capacity for each node would be 12 TB minus 2 TB divided by four, or 2.5 TB.

Related concepts

[How Flash Pool SSD partitioning increases cache allocation flexibility for Flash Pool aggregates](#) on page 116

[How Flash Pool aggregates work](#) on page 111

Related references

[Commands for managing aggregates](#) on page 161

Related information

[NetApp Hardware Universe](#)

How Flash Pool aggregate caching policies work

Caching policies are applied to volumes that reside in Flash Pool aggregates. You should understand how caching policies work before changing them.

In most cases, the default caching policy of **auto** is the best caching policy to use. The caching policy should be changed only if a different policy provides better performance for your workload. Configuring the wrong caching policy can severely degrade volume performance; the performance degradation could increase gradually over time. Use caution when modifying caching policies, and if you experience performance issues with a volume for which the caching policy has been changed, return the caching policy to **auto**.

Caching policies combine a read caching policy and a write caching policy. The policy name concatenates the names of the read caching policy and the write caching policy, separated by a hyphen. Underscores are used within a read or write caching policy name. For example, the **all_read_random_write-random_write** policy combines the “all_read_random_write” read caching

policy and the “random_write” write caching policy. If there is no hyphen in the policy name, the write caching policy is “none”, except for the **auto** policy.

Read caching policies optimize for future read performance by placing a copy of the data in the cache in addition to the stored data on HDDs. For read caching policies that insert data into the cache for write operations, the cache operates as a *write-through* cache.

Data inserted into the cache by using the write caching policy exists only in cache; there is no copy in HDDs. Flash Pool cache is RAID protected. Enabling write caching makes data from write operations available for reads from cache immediately, while deferring writing the data to HDDs until it ages out of the cache.

The following table provides a rough approximation of what types of data are inserted into cache for specific read and write caching policies.

Policy name	Insertions using read caching policy				Insertions using write caching policy	Privilege level
	Random reads	Sequential reads	Random writes	Sequential writes	Random overwrites	
auto	Yes	No	No	No	Yes	admin
none	No	No	No	No	No	admin
random_read	Yes	No	Yes	No	No	advanced
noread-random_write	No	No	No	No	Yes	advanced
meta	Metadata only	No	No	No	No	advanced
meta-random_write	Metadata only	No	No	No	Yes	advanced
random_read_write	Yes	No	Yes	No	No	advanced
random_read_write-random_write	Yes	No	Yes	No	Yes	advanced
all_read	Yes	Yes	No	No	No	advanced
all_read-random_write	Yes	Yes	No	No	Yes	advanced
all_read_random_write	Yes	Yes	Yes	No	No	advanced
all_read_random_write-random_write	Yes	Yes	Yes	No	Yes	advanced
all	Yes	Yes	Yes	Yes	No	advanced
all-random_write	Yes	Yes	Yes	Yes	Yes	advanced

Metadata is cached for all policies except **none**.

You can change the caching policy for a volume that resides on a Flash Pool aggregate by using the `-caching-policy` parameter with the `volume create` command. When you create a volume on a Flash Pool aggregate, by default, the **auto** caching policy is assigned to the volume.

If you move a volume from a Flash Pool aggregate to a single-tier aggregate, it loses its caching policy; if you later move it back to a Flash Pool aggregate, it is assigned the default caching policy of **auto**. If you move a volume between two Flash Pool aggregates, the caching policy is preserved.

Related tasks

[Determining and enabling volume write-caching eligibility for Flash Pool aggregates](#) on page 149

Related information

[NetApp Technical Report 4070: Flash Pool Design and Implementation Guide](#)

Modifying caching policies

You should modify the caching policy of a volume only if a different caching policy is expected to provide better performance. The volume must be on a Flash Pool aggregate.

About this task

In most cases, the default caching policy of **auto** is the best caching policy to use. The caching policy should be changed only if a different policy provides better performance for your workload. Configuring the wrong caching policy can severely degrade volume performance; the performance degradation could increase gradually over time. Use caution when modifying caching policies, and if you experience performance issues with a volume for which the caching policy has been changed, you should return the caching policy to **auto**.

Step

1. Modify the volume's caching policy:

```
volume modify -volume volume_name -caching-policy policy_name
```

Example

The following example modifies the caching policy of a volume named “vol2” to the policy **none**:

```
volume modify -volume vol2 -caching-policy none
```

Related information

[Clustered Data ONTAP 8.3.2 man page: volume modify - Modify volume attributes](#)

How Flash Pool SSD partitioning increases cache allocation flexibility for Flash Pool aggregates

Flash Pool SSD partitioning, also known as *Advanced Drive Partitioning*, enables you to group SSDs together into an *SSD storage pool* that can be allocated to multiple Flash Pool aggregates. This amortizes the cost of the parity SSDs over more aggregates, increases SSD allocation flexibility, and maximizes SSD performance .

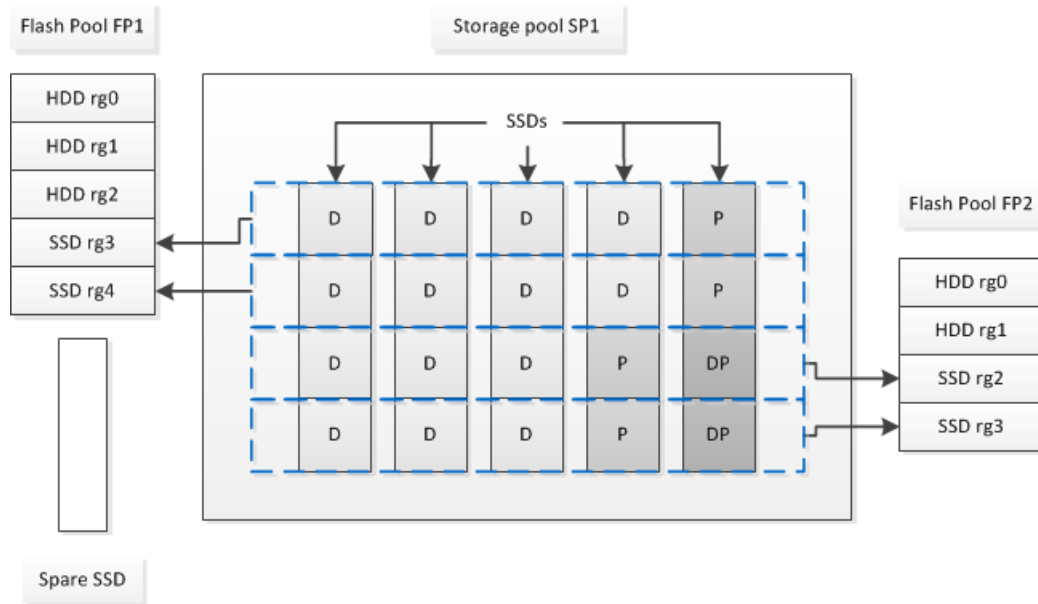
The storage pool is associated with an HA pair, and can be composed of SSDs owned by either node in the HA pair.

When you add an SSD to a storage pool, it becomes a *shared SSD*, and it is divided into 4 partitions.

Storage from an SSD storage pool is divided into *allocation units*, each of which represents 25% of the total storage capacity of the storage pool. Allocation units contain one partition from each SSD in the storage pool, and are added to a Flash Pool cache as a single RAID group. By default, for storage pools associated with an HA pair, two allocation units are assigned to each of the HA partners, but you can reassign the allocation units to the other HA partner if needed (allocation units must be owned by the node that owns the aggregate).

SSD storage pools do not have a RAID type. When an allocation unit is added to a Flash Pool aggregate, the appropriate number of partitions are designated to provide parity to that RAID group.

The following diagram shows one example of Flash Pool SSD partitioning. The SSD storage pool pictured is providing cache to two Flash Pool aggregates:



Storage pool SP1 is composed of 5 SSDs; in addition, there is one hot spare SSD available to replace any SSD that experiences a failure. Two of the storage pool's allocation units are allocated to Flash Pool FP1, and two are allocated to Flash Pool FP2. FP1 has a cache RAID type of RAID4, so the allocation units provided to FP1 contain only one partition designated for parity. FP2 has a cache RAID type of RAID-DP, so the allocation units provided to FP2 include a parity partition and a double-parity partition.

In this example, two allocation units are allocated to each Flash Pool aggregate; however, if one Flash Pool aggregate needed a larger cache, you could allocate three of the allocation units to that Flash Pool aggregate, and only one to the other.

Related concepts

[What a Flash Pool aggregate is](#) on page 111

Related tasks

[Creating a Flash Pool aggregate using SSD storage pools](#) on page 145

Considerations for when to use SSD storage pools

SSD storage pools provide many benefits, but they also introduce some restrictions that you should be aware of when deciding whether to use SSD storage pools or dedicated SSDs.

SSD storage pools make sense only when they are providing cache to two or more Flash Pool aggregates. SSD storage pools provide the following benefits:

- Increased storage utilization for SSDs used in Flash Pool aggregates
SSD storage pools reduce the overall percentage of SSDs needed for parity by enabling you to share parity SSDs between two or more Flash Pool aggregates.
- Ability to share spares between HA partners
Because the storage pool is effectively owned by the HA pair, one spare, owned by one of the HA partners, can function as a spare for the entire SSD storage pool if needed.
- Better utilization of SSD performance
The high performance provided by SSDs can support access by both controllers in an HA pair.

These advantages must be weighed against the costs of using SSD storage pools, which include the following items:

- Reduced fault isolation
The loss of a single SSD affects all RAID groups that include one of its partitions. In this situation, every Flash Pool aggregate that has cache allocated from the SSD storage pool that contains the affected SSD has one or more RAID groups in reconstruction.
- Reduced performance isolation
If the Flash Pool cache is not properly sized, there can be contention for the cache between the Flash Pool aggregates that are sharing it. This risk can be mitigated with proper cache sizing and QoS controls.
- Decreased management flexibility
When you add storage to a storage pool, you increase the size of all Flash Pool caches that include one or more allocation units from that storage pool; you cannot determine how the extra capacity is distributed.

How you use SSD storage pools

To enable SSDs to be shared by multiple Flash Pool aggregates, you place them in a *storage pool*. After you add an SSD to a storage pool, you can no longer manage it as a stand-alone entity—you must use the storage pool to assign or allocate the storage provided by the SSD.

You create storage pools for a specific HA pair. Then, you add allocation units from that storage pool to one or more Flash Pool aggregates owned by the same HA pair. Just as disks must be owned by the same node that owns an aggregate before they can be allocated to it, storage pools can provide storage only to Flash Pool aggregates owned by one of the nodes that owns the storage pool.

If you need to increase the amount of Flash Pool cache on your system, you can add more SSDs to a storage pool, up to the maximum RAID group size for the RAID type of the Flash Pool caches using the storage pool. When you add an SSD to an existing storage pool, you increase the size of the storage pool's allocation units, including any allocation units that are already allocated to a Flash Pool aggregate.

You should provide one or more spare SSDs for your storage pools, so that if an SSD in that storage pool becomes unavailable, Data ONTAP can use a spare SSD to reconstruct the partitions of the malfunctioning SSD. You do not need to reserve any allocation units as spare capacity; Data ONTAP can use only a full, unpartitioned SSD as a spare for SSDs in a storage pool.

After you add an SSD to a storage pool, you cannot remove it, just as you cannot remove disks from an aggregate. If you want to use the SSDs in a storage pool as discrete drives again, you must destroy

all Flash Pool aggregates to which the storage pool's allocation units have been allocated, and then destroy the storage pool.

Requirements and best practices for using SSD storage pools

There are some technologies that cannot be combined with Flash Pool aggregates that use SSD storage pools.

You cannot use the following technologies with Flash Pool aggregates that use SSD storage pools for their cache storage:

- MetroCluster
- SyncMirror

Mirrored aggregates can coexist with Flash Pool aggregates that use storage pools; however, Flash Pool aggregates cannot be mirrored.
- Physical SSDs

Flash Pool aggregates can use SSD storage pools or physical SSDs, but not both.

SSD storage pools must conform to the following rules:

- SSD storage pools can contain only SSDs; HDDs cannot be added to an SSD storage pool.
- SSD storage pools can contain between 3 and 28 SSDs.

If an SSD storage pool contains more SSDs than the maximum RAID4 RAID group size for SSDs, then it cannot be used for a Flash Pool aggregate whose cache has a RAID type of RAID4.
- All SSDs in an SSD storage pool must be owned by the same HA pair.
- You cannot use SSDs that have been partitioned for root-data partitioning in a storage pool.

If you provide storage from a single storage pool to two caches with different RAID types, and you expand the size of the storage pool beyond the maximum RAID group size for RAID4, the extra partitions in the RAID4 allocation units go unused. For this reason, it is a best practice to keep your cache RAID types homogenous for a storage pool.

You cannot change the RAID type of cache RAID groups allocated from a storage pool. You set the RAID type for the cache before adding the first allocation units, and you cannot change it later.

When you create a storage pool or add SSDs to an existing storage pool, you must use the same size SSDs. If a failure occurs and no spare of the correct size exists, Data ONTAP can use a larger SSD to replace the failed SSD. However, the larger SSD is right-sized to match the size of the other SSDs in the storage pool, resulting in lost SSD capacity.

You can use only one spare SSD for a storage pool. If the storage pool provides allocation units to Flash Pool aggregates owned by both nodes in the HA pair, then the spare SSD can be owned by either node. However, if the storage pool provides allocation units only to Flash Pool aggregates owned by one of the nodes in the HA pair, then the SSD spare must be owned by that same node.

Creating an SSD storage pool

You create SSD storage pools to provide SSD cache for two to four Flash Pool aggregates.

About this task

Storage pools do not support a diskcount parameter; you must supply a disk list when creating or adding disks to a storage pool.

Steps

1. Determine the names of the spare SSDs available to you:

```
storage aggregate show-spare-disks -disk-type SSD
```

The SSDs used in a storage pool can be owned by either node of an HA pair.

2. Create the storage pool:

```
storage pool create -storage-pool sp_name -disk-list disk1,disk2,...
```

3. Optional: Show the newly created storage pool:

```
storage pool show -storage-pool sp_name
```

Result

After the SSDs are placed into the storage pool, they no longer appear as spares on the cluster, even though the storage provided by the storage pool has not yet been allocated to any Flash Pool caches. The SSDs can no longer be added to a RAID group as a discrete drive; their storage can be provisioned only by using the allocation units of the storage pool to which they belong.

Related concepts

[How Flash Pool SSD partitioning increases cache allocation flexibility for Flash Pool aggregates](#) on page 116

[Considerations for adding SSDs to an existing storage pool versus creating a new one](#) on page 121

Related tasks

[Creating a Flash Pool aggregate using SSD storage pools](#) on page 145

[Adding SSDs to an SSD storage pool](#) on page 120

Related references

[Commands for managing SSD storage pools](#) on page 122

Adding SSDs to an SSD storage pool

When you add SSDs to an SSD storage pool, you increase the storage pool's physical and usable sizes and allocation unit size. The larger allocation unit size also affects allocation units that have already been allocated to Flash Pool aggregates.

Before you begin

You must have determined that this operation will not cause you to exceed the cache limit for your HA pair. Data ONTAP does not prevent you from exceeding the cache limit when you add SSDs to an SSD storage pool, and doing so can render the newly added storage capacity unavailable for use.

About this task

When you add SSDs to an existing SSD storage pool, the SSDs must be owned by one node or the other of the same HA pair that already owned the existing SSDs in the storage pool. You can add SSDs that are owned by either node of the HA pair.

Steps

1. Optional: View the current allocation unit size and available storage for the storage pool:

```
storage pool show -instance sp_name
```

2. Find available SSDs:

```
storage disk show -container-type spare -type SSD
```

3. Add the SSDs to the storage pool:


```
storage pool add -storage-pool sp_name -disk-list disk1,disk2...
```

The system displays which Flash Pool aggregates will have their size increased by this operation and by how much, and prompts you to confirm the operation.

Related concepts

[How Flash Pool SSD partitioning increases cache allocation flexibility for Flash Pool aggregates](#) on page 116

[Considerations for adding SSDs to an existing storage pool versus creating a new one](#) on page 121

[How the available Flash Pool cache capacity is calculated](#) on page 113

Related tasks

[Determining the impact to cache size of adding SSDs to an SSD storage pool](#) on page 121

[Creating an SSD storage pool](#) on page 119

Related information

[NetApp Hardware Universe](#)

Considerations for adding SSDs to an existing storage pool versus creating a new one

You can increase the size of your SSD cache in two ways—by adding SSDs to an existing SSD storage pool or by creating a new SSD storage pool. The best method for you depends on your configuration and plans for the storage.

The choice between creating a new storage pool or adding storage capacity to an existing one is similar to deciding whether to create a new RAID group or add storage to an existing one:

- If you are adding a large number of SSDs, creating a new storage pool provides more flexibility because you can allocate the new storage pool differently from the existing one.
- If you are adding only a few SSDs, and increasing the RAID group size of your existing Flash Pool caches is not an issue, then adding SSDs to the existing storage pool keeps your spare and parity costs lower, and automatically allocates the new storage.

If your storage pool is providing allocation units to Flash Pool aggregates whose caches have different RAID types, and you expand the size of the storage pool beyond the maximum RAID4 RAID group size, the newly added partitions in the RAID4 allocation units are unused.

Determining the impact to cache size of adding SSDs to an SSD storage pool

If adding SSDs to a storage pool causes your platform model's cache limit to be exceeded, Data ONTAP does not allocate the newly added capacity to any Flash Pool aggregates. This can result in some or all of the newly added capacity being unavailable for use.

About this task

When you add SSDs to an SSD storage pool that has allocation units already allocated to Flash Pool aggregates, you increase the cache size of each of those aggregates and the total cache on the system. If none of the storage pool's allocation units have been allocated, adding SSDs to that storage pool does not affect the SSD cache size until one or more allocation units are allocated to a cache.

Steps

1. Determine the usable size of the SSDs you are adding to the storage pool:

```
storage disk show disk_name -fields usable-size
```

2. Determine how many allocation units remain unallocated for the storage pool:

```
storage pool show-available-capacity sp_name
```

All unallocated allocation units in the storage pool are displayed.

3. Calculate the amount of cache that will be added by applying the following formula:

$(4 - \text{number of unallocated allocation units}) \times 25\% \times \text{usable size} \times \text{number of SSDs}$

Commands for managing SSD storage pools

Data ONTAP provides the `storage pool` command for managing SSD storage pools.

If you want to...	Use this command...
Display how much storage a storage pool is providing to which aggregates	<code>storage pool show-aggregate</code>
Display how much cache would be added to the overall cache capacity for both RAID types (allocation unit data size)	<code>storage pool show -instance</code>
Display the disks in a storage pool	<code>storage pool show-disks</code>
Display the unallocated allocation units for a storage pool	<code>storage pool show-available-capacity</code>
Change the ownership of one or more allocation units of a storage pool from one HA partner to the other	<code>storage pool reassign</code>

Related information

[Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#)

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.

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.

Related tasks

[Assigning aggregates to SVMs](#) on page 159

Related information

[Clustered Data ONTAP 8.3 System Administration Guide](#)

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 with a usable size between 90% and 105% 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 disk selection preview.
You can use the `-simulate` option to identify which disks Data ONTAP selects automatically. If you are happy with the disks selected, you can proceed with automatic disk selection. Otherwise, you can use one of the previous methods to ensure that the correct disks are selected.
- Through an explicit disk list.
You can list the names of specific disks you want to use. However, it is a best practice to use the `diskcount` option rather than disk lists; this allows Data ONTAP to make the best disk selection for your configuration.

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 for 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.

If your node uses root-data partitioning, the same mixing rules apply to the partitioned HDDs as to HDDs that are not partitioned. Partitioned HDDs can be mixed with HDDs that are not partitioned in the same aggregate if they are otherwise compatible, but not in the same RAID group. Physical (unpartitioned) HDDs that are added to an existing RAID group that contains partitioned HDDs become partitioned.

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 accept mixed HDD types as long as that option is set to **off**.

Related concepts

How Data ONTAP reports disk types on page 9

Related information

NetApp Technical Report 3437: Storage Subsystem 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 standard 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 with 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.
- **Snapshot Reserve Unusable**
The amount of space originally allocated for aggregate Snapshot reserve that is unavailable for aggregate Snapshot copies because it is being used by volumes associated with the aggregate. Can occur only for aggregates with a non-zero aggregate Snapshot reserve.
- **Total Used**
The sum of all space used or reserved in the aggregate by volumes, metadata, or Snapshot copies.
- **Total Physical Used**
The amount of space being used for data now (rather than being reserved for future use). Includes space used by aggregate 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 is 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%
Total Physical Used	34.82KB	0%

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 the `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 <code>volume show-space</code> command output can exceed the value in this row.	<ul style="list-style-type: none"> 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.	<p>Changing the type of guarantee for the volume to none. This row will go to 0.</p> <p>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.</p>
Flexible Volume Metadata	The total amount of space used in the aggregate by the volume's metadata files.	No direct method to control.
Delayed Frees	<p>Blocks that Data ONTAP used for performance and cannot be immediately freed.</p> <p>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.</p> <p>For SnapMirror destinations, this row has a value of 0 and is not displayed.</p>	No direct method to control.
File Operation Metadata	<p>The total amount of space reserved for file operation metadata.</p> <p>After space is used for file operation metadata, it is not returned as free space to the aggregate, but it is reused by subsequent file operations.</p>	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.

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 should 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.

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.

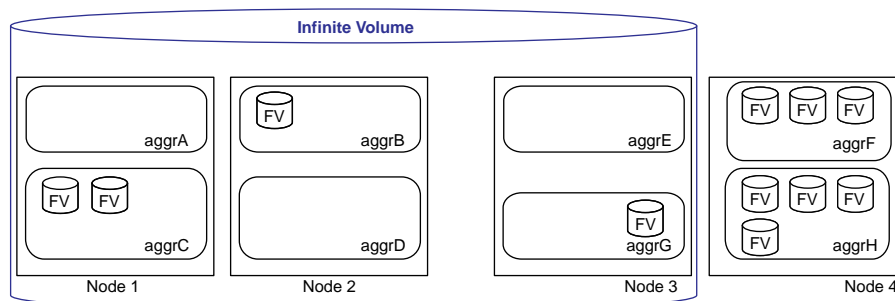
How FlexVol volumes and Infinite Volumes share aggregates

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.

The 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 add-aggregates` command.

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 using unpartitioned drives

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.
- You should have determined the correct RAID group size given the number of drives or array LUNs you are using to create the aggregate and your plans for expanding the aggregate in the future.
- If you have multiple drive types in your node (heterogeneous storage), you should understand how you can ensure that the correct drive type is selected.

About this task

This procedure should not be used to create an aggregate composed of root or data partitions.

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:

- They must begin with either a letter or an underscore (_).
- They can contain only letters, digits, and underscores.
- They can contain 250 or fewer characters.

Steps

1. Display a list of available spares:

```
storage aggregate show-spare-disks -original-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 `-node` parameter or specify drives that are owned by that node.

The following list describes some of the options you can specify:

- 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)
- The number of disks or array LUNs to be added, or a list of specific drives or array LUNs that are to be included in the aggregate
For optimal disk selection, use the `diskcount` parameter rather than a disk list. This allows Data ONTAP to make an optimal disk selection for your configuration.
- 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
3. Monitor zeroing progress and verify the RAID groups and drives of your new aggregate:
- ```
storage aggregate show-status aggr_name
```

### Example

You want to create a storage configuration for a single-node cluster and have 4 shelves of SAS disks totaling 96 drives, with 3 drives used for the root aggregate. You decide to allocate the remaining drives in the following manner:

- 2 spares
- 91 drives used for data in one aggregate:
  - 4 RAID groups of 18 drives each
  - 1 RAID group of 19 drives

To accomplish this goal, you use the following steps:

1. Create the first RAID group of 19 disks:
 

```
storage aggregate create -aggregate n01sas01 -node cl-01 -diskcount 19 -maxraidsize 19 -disktype SAS
```
2. Change the aggregate's maximum RAID group size to 18:
 

```
storage aggregate modify -aggregate n01sas01 -maxraidsize 18
```
3. Add the remaining 54 disks as 4 RAID groups of 18 disks each:
 

```
storage aggregate add-disks -aggregate n01sas01 -node cl-01 -diskcount 54 -disktype SAS
```

Later, if you added a 24-drive shelf to the node and wanted to add the storage to this aggregate, you could again modify the RAID group size to 24 so that the entire shelf could be accommodated in a single RAID group, provided that a RAID group size of 24 is acceptable for your installation.

### Related concepts

[What aggregates are](#) on page 108

[Considerations for sizing RAID groups](#) on page 97

[Understanding how Data ONTAP works with heterogeneous storage](#) on page 123

### Related tasks

[Creating an aggregate using root-data partitioning](#) on page 136

[Creating a Flash Pool aggregate using physical SSDs](#) on page 143

## Creating an aggregate using root-data partitioning

Typically, root aggregates are created in the factory, and data aggregates are created during initial system setup. However, if you need to create a new data aggregate using partitioned drives, there are some differences in the procedure from creating an aggregate using physical, unpartitioned drives.

### Before you begin

You should know what drives or partitions will be used in the new aggregate.

You should have determined the correct RAID group size given the number of drives or partitions you are using to create the aggregate and your plans for expanding the aggregate in the future.

### About this task

Drives and partitions are owned by a specific node; when you create an aggregate, all of the partitions you use for the new aggregate must be owned by the same node, which becomes the home node for the new aggregate.

When you provision a data aggregate from partitions, it is best to use the *diskcount* method of specifying the number of partitions to be used in the aggregate. Data ONTAP chooses partitions so that each node has at least one disk with both partitions as spare, which makes that disk available for a core dump if necessary. With this configuration, if the node experiences a controller disruption, valuable information about the problem (the core file) can be made available to technical support.

Aggregate names must conform to the following requirements:

- They must begin with either a letter or an underscore (\_).
- They can contain only letters, digits, and underscores.
- They can contain 250 or fewer characters.

### Steps

1. View the list of spare data partitions:

```
storage aggregate show-spare-disks -original-owner node_name
```

The list of disks with at least one spare partition is displayed. Data partitions are shown under Local Data Usable. You must use data partitions when you create a data aggregate.

2. Determine how many partitions you want to use in the aggregate.

Remember to leave an appropriate number of spare data partitions. Data ONTAP will not use a root partition as a spare for a data aggregate.

3. Simulate the creation of the aggregate:

```
storage aggregate create -aggregate aggr_name -node node_name -diskcount
number_of_partitions -simulate true
```

This enables you to see the result of the aggregate creation without actually provisioning any storage. If any warnings are displayed from the simulated command, you can adjust the command and repeat the simulation.

4. Create the aggregate:

```
storage aggregate create -aggregate aggr_name -node node_name -diskcount
number_of_partitions
```

5. Check zeroing status and verify the composition of the aggregate:



```
storage aggregate show-status aggr_name
```

#### Related concepts

[Understanding root-data partitioning](#) on page 25

[Considerations for sizing RAID groups](#) on page 97

#### Related tasks

[Creating an aggregate using unpartitioned drives](#) on page 134

[Correcting misaligned spare partitions](#) on page 142

## Increasing the size of an aggregate that uses unpartitioned drives

You can add disks or array LUNs to an aggregate so that it can provide more storage to its associated volumes.

#### Before you begin

- You must understand the requirement to add disks or array LUNs owned by the same system and pool
- For aggregates composed of disks, you must understand the following:
  - Benefits of keeping your RAID groups homogeneous for disk size and speed
  - Which types of disks can be used together
  - Checksum rules when disks of more than one checksum type are in use
  - How to ensure that the correct disks are added to the aggregate (the disk addition operation cannot be undone)
  - How to add disks to aggregates from heterogeneous storage
  - Minimum number of disks to add for best performance
  - Number of hot spares you need to provide for protection against disk failures
  - Requirements for adding disks from multi-disk carrier disk shelves
  - Requirement to add storage to both plexes of a mirrored aggregate at the same time to ensure that the plexes are the same size and contain the same disk types
  - If you are adding cache to a Flash Pool aggregate, the cache limit for your system model and how much cache you are adding towards the limit

#### About this task

This procedure should not be used for aggregates composed of root or data partitions.

Following these best practices when you add storage to an aggregate optimizes aggregate performance:

- Add a complete RAID group at one time.  
The new RAID group does not have to be exactly the same size as the existing RAID groups, but it should not be less than one half the size of the existing RAID groups.
- If any small RAID groups exist already, you can bring them up to the size of the other RAID groups, as long as you add at least as many data drives as are already in the RAID group.

- Avoid adding a small number of drives to an existing RAID group.  
Doing so results in the added disks being the target for a disproportionate percentage of new data, causing the new disks to become a performance bottleneck.

### Steps

1. Verify that appropriate spare disks or array LUNs are available for you to add:

```
storage aggregate show-spare-disks -original-owner node_name
```

For disks, make sure that enough of the spares listed are of the correct type, size, speed, and checksum type for the target RAID group in the aggregate to which you are adding the disks.

2. Add the disks or array LUNs:

```
storage aggregate add-disks -aggregate aggr_name [-raidgroup new] disks
```

If you are adding disks with a different checksum than the aggregate, as when creating a Flash Pool aggregate, or if you are adding disks to a mixed checksum aggregate, you must use the `-checksumstyle` parameter.

If you are adding disks to a Flash Pool aggregate, you must use the `-disktype` parameter to specify the disk type.

You can use the `-disksize` parameter to specify a size of the disks to add. Only disks with approximately the specified size are selected for addition to the aggregate.

If you specify the `-raidgroup` parameter with a value of **new**, the storage is added to a new RAID group; this is generally the best way to add storage to an aggregate. You can also specify an existing RAID group with the `-raidgroup` parameter. If you are adding SSDs to the SSD cache of a Flash Pool aggregate, you do not need to specify the RAID group name; the SSD RAID group is selected by default based on the type of the disks you are adding.

`disks` specifies the disks to be added in one of the following ways:

- `-diskcount`, usually further qualified by disk type, disk size, or checksum type
- `-disklist disk1 [disk2...]`

If possible, you should use the `diskcount` option rather than selecting specific disks. Doing so allows Data ONTAP to optimize the disk selection for your configuration.

If you are adding disks to a mirrored aggregate and you are specifying disk names, you must also use the `-mirror-disklist` parameter.

### Related concepts

[How to control disk selection from heterogeneous storage](#) on page 123

[Considerations for sizing RAID groups](#) on page 97

[What a Flash Pool aggregate is](#) on page 111

[Rules for mixing drive types in Flash Pool aggregates](#) on page 125

[Rules for mixing HDD types in aggregates](#) on page 124

[How the available Flash Pool cache capacity is calculated](#) on page 113

[What happens when you add storage to an aggregate](#) on page 143

### Related tasks

[Increasing the size of an aggregate that uses root-data partitioning](#) on page 139

### Related information

[NetApp Technical Report 4070: Flash Pool Design and Implementation Guide](#)

## Increasing the size of an aggregate that uses root-data partitioning

When you add storage to an existing aggregate that is using partitioned drives, you should be aware of whether you are adding a partitioned drive or an unpartitioned drive, and the tradeoffs of mixing those types of drives in the same RAID group versus creating a new RAID group.

### Before you begin

- You should understand whether you are adding partitions or unpartitioned drives to the aggregate.
- You should know what the RAID group size is for the aggregate you are adding the storage to.
- When you add unpartitioned drives to an existing aggregate, the size of the existing RAID groups is inherited by the new RAID group, which can affect the number of parity disks required.

### About this task

When you add storage to an existing aggregate, you can let Data ONTAP choose the RAID group to add the storage to, or you can designate the target RAID group for the added storage, including creating a new RAID group.

If an unpartitioned drive is added to a RAID group composed of partitioned drives, the new drive is partitioned, leaving an unused spare partition. If you do not want the new drive to be partitioned, you can add it to a RAID group that contains only unpartitioned (physical) drives. However, partitioning a drive might be preferable to creating a new, small RAID group.

Following these best practices when you add storage to an aggregate optimizes aggregate performance:

- Add a complete RAID group at one time.  
The new RAID group does not have to be exactly the same size as the existing RAID groups, but it should not be less than one half the size of the existing RAID groups.
- If any small RAID groups exist already, you can bring them up to the size of the other RAID groups, as long as you add at least as many data drives as are already in the RAID group.
- Avoid adding a small number of drives to an existing RAID group.  
Doing so results in the added drives being the target for a disproportionate percentage of new data, causing the new drives to become a performance bottleneck.

When you provision partitions, you must ensure that you do not leave the node without a drive with both partitions as spare. If you do, and the node experiences a controller disruption, valuable information about the problem (the core file) might not be available to provide to technical support.

### Steps

1. Show the available spare storage on the system that owns the aggregate:

```
storage aggregate show-spare-disks -original-owner node_name
```

You can use the `-is-disk-shared` parameter to show only partitioned drives or only unpartitioned drives.

2. Show the current RAID groups for the aggregate:

```
storage aggregate show-status aggr_name
```

3. Simulate adding the storage to the aggregate:

```
storage aggregate add-disks -aggregate aggr_name -diskcount
number_of_disks_or_partitions -simulate true
```

This enables you to see the result of the storage addition without actually provisioning any storage. If any warnings are displayed from the simulated command, you can adjust the command and repeat the simulation.

4. Add the storage to the aggregate:

```
storage aggregate add-disks -aggregate aggr_name -diskcount
number_of_disks_or_partitions
```

You can use the `-raidgroup` parameter to designate the RAID group to add the storage to, or to create a new RAID group, by using a value of `new`.

If you are adding partitions to the aggregate, you must use a disk that shows available capacity for the required partition type. For example, if you are adding partitions to a data aggregate (and using a disk list), the disk names you use must show available capacity in the `Local Data Usable` column.

5. Verify that the storage was added successfully:

```
storage aggregate show-status -aggregate aggr_name
```

6. Ensure that the node still has at least one drive with both the root partition and the data partition as spare:

```
storage aggregate show-spare-disks -original-owner node_name
```

If the node does not have a drive with both partitions as spare and it experiences a controller disruption, then valuable information about the problem (the core file) might not be available to provide to technical support.

### Example: Adding partitioned drives to an aggregate

The following example shows that the `c11-s2` node has multiple spare partitions available:

```
c11-s2::> storage aggregate show-spare-disks -original-owner c11-s2 -is-disk-shared true

Original Owner: c11-s2
Pool0
 Shared HDD Spares
```

| Disk   | Type | RPM  | Checksum | Local Data Usable | Local Root Usable | Physical Size | Status |
|--------|------|------|----------|-------------------|-------------------|---------------|--------|
| 1.0.1  | BSAS | 7200 | block    | 753.8GB           | 73.89GB           | 828.0GB       | zeroed |
| 1.0.2  | BSAS | 7200 | block    | 753.8GB           | 0B                | 828.0GB       | zeroed |
| 1.0.3  | BSAS | 7200 | block    | 753.8GB           | 0B                | 828.0GB       | zeroed |
| 1.0.4  | BSAS | 7200 | block    | 753.8GB           | 0B                | 828.0GB       | zeroed |
| 1.0.8  | BSAS | 7200 | block    | 753.8GB           | 0B                | 828.0GB       | zeroed |
| 1.0.9  | BSAS | 7200 | block    | 753.8GB           | 0B                | 828.0GB       | zeroed |
| 1.0.10 | BSAS | 7200 | block    | 0B                | 73.89GB           | 828.0GB       | zeroed |

2 entries were displayed.

The following example shows that the `data_1` aggregate is composed of a single RAID group of five partitions:

```
c11-s2::> storage aggregate show-status -aggregate data_1

Owner Node: c11-s2
Aggregate: data_1 (online, raid_dp) (block checksums)
Plex: /data_1/plex0 (online, normal, active, pool0)
RAID Group /data_1/plex0/rg0 (normal, block checksums)
```

| Position | Disk   | Pool | Type | RPM  | Usable Size | Physical Size | Status   |
|----------|--------|------|------|------|-------------|---------------|----------|
| shared   | 1.0.10 | 0    | BSAS | 7200 | 753.8GB     | 828.0GB       | (normal) |
| shared   | 1.0.5  | 0    | BSAS | 7200 | 753.8GB     | 828.0GB       | (normal) |
| shared   | 1.0.6  | 0    | BSAS | 7200 | 753.8GB     | 828.0GB       | (normal) |
| shared   | 1.0.11 | 0    | BSAS | 7200 | 753.8GB     | 828.0GB       | (normal) |
| shared   | 1.0.0  | 0    | BSAS | 7200 | 753.8GB     | 828.0GB       | (normal) |

5 entries were displayed.

The following example shows which partitions would be added to the aggregate:

```
c11-s2::> storage aggregate add-disks data_1 -diskcount 5 -simulate true
```

Addition of disks would succeed for aggregate "data\_1" on node "c11-s2". The following disks would be used to add to the aggregate: 1.0.2, 1.0.3, 1.0.4, 1.0.8, 1.0.9.

The following example adds five spare data partitions to the aggregate:

```
c11-s2::> storage aggregate add-disks data_1 -diskcount 5
```

The following example shows that the data partitions were successfully added to the aggregate:

```
c11-s2::> storage aggregate show-status -aggregate data_1
```

Owner Node: c11-s2  
Aggregate: data\_1 (online, raid\_dp) (block checksums)  
Plex: /data\_1/plex0 (online, normal, active, pool0)  
RAID Group /data\_1/plex0/rg0 (normal, block checksums)

| Position | Disk   | Pool | Type | RPM  | Usable<br>Size | Physical<br>Size | Status   |
|----------|--------|------|------|------|----------------|------------------|----------|
| shared   | 1.0.10 | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.5  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.6  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.11 | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.0  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.2  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.3  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.4  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.8  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |
| shared   | 1.0.9  | 0    | BSAS | 7200 | 753.8GB        | 828.0GB          | (normal) |

10 entries were displayed.

The following example verifies that an entire disk, disk 1.0.1, remains available as a spare:

```
c11-s2::> storage aggregate show-spare-disks -original-owner c11-s2 -is-disk-shared true
```

Original Owner: c11-s2  
Pool0  
Shared HDD Spares

| Disk   | Type | RPM  | Checksum | Local<br>Data<br>Usable | Local<br>Root<br>Usable | Physical<br>Size | Status |
|--------|------|------|----------|-------------------------|-------------------------|------------------|--------|
| 1.0.1  | BSAS | 7200 | block    | 753.8GB                 | 73.89GB                 | 828.0GB          | zeroed |
| 1.0.10 | BSAS | 7200 | block    | 0B                      | 73.89GB                 | 828.0GB          | zeroed |

2 entries were displayed.

## Related concepts

[How to control disk selection from heterogeneous storage](#) on page 123

[What a Flash Pool aggregate is](#) on page 111

[Considerations for sizing RAID groups](#) on page 97

[Rules for mixing drive types in Flash Pool aggregates](#) on page 125

[Rules for mixing HDD types in aggregates](#) on page 124

[How the available Flash Pool cache capacity is calculated](#) on page 113

[What happens when you add storage to an aggregate](#) on page 143

## Related tasks

[Correcting misaligned spare partitions](#) on page 142

[Increasing the size of an aggregate that uses unpartitioned drives](#) on page 137

## Related information

[NetApp Technical Report 4070: Flash Pool Design and Implementation Guide](#)

## Correcting misaligned spare partitions

When you add partitioned disks to an aggregate, you must leave a disk with both the root and data partition available as spare for every node. If you do not and your node experiences a disruption, Data ONTAP might not be able to create a core file.

### Before you begin

You must have both a spare data partition and a spare root partition on the same type of disk owned by the same node.

### Steps

1. Display the spare partitions for the node:

```
storage aggregate show-spare-disks -original-owner node_name
```

Note which disk has a spare data partition (spare\_data) and which disk has a spare root partition (spare\_root). The spare partition will show a non-zero value under the Local Data Usable or Local Root Usable column.

2. Replace the disk with a spare data partition with the disk with the spare root partition:

```
storage disk replace -disk spare_data -replacement spare_root -action start
```

You can copy the data in either direction; however, copying the root partition takes less time to complete.

3. Monitor the progress of the disk replacement:

```
storage aggregate show-status -aggregate aggr_name
```

4. After the replacement operation is complete, display the spares again to confirm that you have a full spare disk:

```
storage aggregate show-spare-disks -original-owner node_name
```

You should see a spare disk with usable space under both Local Data Usable and Local Root Usable.

### Example

You display your spare partitions for node c1-01 and see that your spare partitions are not aligned:

```
c1::> storage aggregate show-spare-disks -original-owner c1-01

Original Owner: c1-01
Pool0
Shared HDD Spares
```

| Disk   | Type | RPM  | Checksum | Local<br>Data<br>Usable | Local<br>Root<br>Usable | Physical<br>Size |
|--------|------|------|----------|-------------------------|-------------------------|------------------|
| 1.0.1  | BSAS | 7200 | block    | 753.8GB                 | 0B                      | 828.0GB          |
| 1.0.10 | BSAS | 7200 | block    | 0B                      | 73.89GB                 | 828.0GB          |

You start the disk replacement job:

```
c1::> storage disk replace -disk 1.0.1 -replacement 1.0.10 -action start
```

While you are waiting for the replacement operation to finish, you display the progress of the operation:

```
c1::> storage aggregate show-status -aggregate aggr0_1
```

Owner Node: c1-01  
Aggregate: aggr0\_1 (online, raid\_dp) (block checksums)  
Plex: /aggr0\_1/plex0 (online, normal, active, pool0)  
RAID Group /aggr0\_1/plex0/rg0 (normal, block checksums)

| Position | Disk   | Pool | Type | RPM  | Usable Size | Physical Size | Status                        |
|----------|--------|------|------|------|-------------|---------------|-------------------------------|
| shared   | 1.0.1  | 0    | BSAS | 7200 | 73.89GB     | 828.0GB       | (replacing, copy in progress) |
| shared   | 1.0.10 | 0    | BSAS | 7200 | 73.89GB     | 828.0GB       | (copy 63% completed)          |
| shared   | 1.0.0  | 0    | BSAS | 7200 | 73.89GB     | 828.0GB       | (normal)                      |
| shared   | 1.0.11 | 0    | BSAS | 7200 | 73.89GB     | 828.0GB       | (normal)                      |
| shared   | 1.0.6  | 0    | BSAS | 7200 | 73.89GB     | 828.0GB       | (normal)                      |
| shared   | 1.0.5  | 0    | BSAS | 7200 | 73.89GB     | 828.0GB       | (normal)                      |

After the replacement operation is complete, you confirm that you have a full spare disk:

```
ie2220::> storage aggregate show-spare-disks -original-owner c1-01
```

Original Owner: c1-01  
Pool0  
Shared HDD Spares

| Disk  | Type | RPM  | Checksum | Local Data Usable | Local Root Usable | Physical Size |
|-------|------|------|----------|-------------------|-------------------|---------------|
| 1.0.1 | BSAS | 7200 | block    | 753.8GB           | 73.89GB           | 828.0GB       |

## 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.

### Related tasks

[Increasing the size of an aggregate that uses unpartitioned drives](#) on page 137

[Increasing the size of an aggregate that uses root-data partitioning](#) on page 139

## Creating a Flash Pool aggregate using physical SSDs

You create a Flash Pool aggregate by enabling the feature on an existing 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 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, but larger RAID groups require RAID-DP.
- You must have determined the RAID level you want to use for the SSD cache.
- You must have determined the maximum cache size for your system and determined that adding SSD cache to your aggregate will not cause you to exceed it.
- 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.

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.

### Steps

1. Mark the aggregate as eligible to become a Flash Pool aggregate:

```
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` command.

You can specify the SSDs by ID or by using the `diskcount` and `disktype` parameters.

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` parameter.

If you want the cache RAID group size to be different from the default for the RAID type you are using, you should change it now, by using the `-cache-raid-group-size` parameter.

### Related concepts

[What a Flash Pool aggregate is](#) on page 111

[Requirements for using Flash Pool aggregates](#) on page 112



[Considerations for RAID type and spare management for Flash Pool cache](#) on page 112

[How the available Flash Pool cache capacity is calculated](#) on page 113

#### Related tasks

[Creating a Flash Pool aggregate using SSD storage pools](#) on page 145

[Determining Flash Pool candidacy and optimal cache size](#) on page 147

[Determining and enabling volume write-caching eligibility for Flash Pool aggregates](#) on page 149

[Creating an aggregate using unpartitioned drives](#) on page 134

#### Related information

[NetApp Technical Report 4070: Flash Pool Design and Implementation Guide](#)

## Creating a Flash Pool aggregate using SSD storage pools

You create a Flash Pool aggregate with SSD storage pools by enabling the feature on an existing aggregate composed of HDD RAID groups, and then adding one or more SSD storage pool allocation units to that aggregate.

#### Before you begin

- You must have identified a valid 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 created an SSD storage pool to provide the SSD cache to this Flash Pool aggregate.  
Any allocation unit from the storage pool that you want to use must be owned by the same node that owns the Flash Pool aggregate.
- You must have determined how much cache you want to add to the aggregate.  
You add cache to the aggregate by allocation units. You can increase the size of the allocation units later by adding SSDs to the storage pool if there is room.
- You must have determined the RAID type you want to use for the SSD cache.  
After you add a cache to the aggregate from SSD storage pools, you cannot change the RAID type of the cache RAID groups.
- You must have determined the maximum cache size for your system and determined that adding SSD cache to your aggregate will not cause you to exceed it.  
You can see the amount of cache that will be added to the total cache size by using the `storage pool show` command.
- You must have familiarized yourself with the configuration requirements for Flash Pool aggregates.

#### About this task

If you want the RAID type of the cache to differ from that of the HDD RAID groups, you must specify the cache RAID type when you add the SSD capacity. After you add the SSD capacity to the aggregate, you can no longer change the RAID type of the cache.

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.

**Steps**

1. Mark the aggregate as eligible to become a Flash Pool aggregate:

```
storage aggregate modify -aggregate aggr_name -hybrid-enabled true
```

If this step does not succeed, determine write-caching eligibility for the target aggregate.

2. Show the available SSD storage pool allocation units:

```
storage pool show-available-capacity
```

3. Add the SSD capacity to the aggregate:

```
storage aggregate add aggr_name -storage-pool sp_name -allocation-units
number_of_units
```

If you want the RAID type of the cache to be different from that of the HDD RAID groups, you must change it when you enter this command by using the `raidtype` parameter.

You do not need to specify a new RAID group; Data ONTAP automatically puts the SSD cache into separate RAID groups from the HDD RAID groups.

You cannot set the RAID group size of the cache; it is determined by the number of SSDs in the storage pool.

The cache is added to the aggregate and the aggregate is now a Flash Pool aggregate. Each allocation unit added to the aggregate becomes its own RAID group.

4. Optional: Confirm the presence and size of the SSD cache:

```
storage aggregate show aggr_name
```

The size of the cache is listed under `Total Hybrid Cache Size`.

**Related concepts**

[How Flash Pool SSD partitioning increases cache allocation flexibility for Flash Pool aggregates](#) on page 116

[What a Flash Pool aggregate is](#) on page 111

[Requirements for using Flash Pool aggregates](#) on page 112

[Considerations for RAID type and spare management for Flash Pool cache](#) on page 112

[How the available Flash Pool cache capacity is calculated](#) on page 113

**Related tasks**

[Creating a Flash Pool aggregate using physical SSDs](#) on page 143

[Creating an aggregate using unpartitioned drives](#) on page 134

[Determining Flash Pool candidacy and optimal cache size](#) on page 147

[Determining and enabling volume write-caching eligibility for Flash Pool aggregates](#) on page 149

**Related information**

[NetApp Technical Report 4070: Flash Pool Design and Implementation Guide](#)

## Determining Flash Pool candidacy and optimal cache size

Before converting an existing aggregate to a Flash Pool aggregate, you can determine whether the aggregate is I/O bound, and what would be the best Flash Pool cache size for your workload and budget. You can also check whether the cache of an existing Flash Pool aggregate is sized correctly.

### Before you begin

You should know approximately when the aggregate you are analyzing experiences its peak load.

### Steps

1. Enter advanced mode:

```
set advanced
```

2. If you need to determine whether an existing aggregate would be a good candidate for conversion to a Flash Pool aggregate, determine how busy the disks in the aggregate are during a period of peak load, and how that is affecting latency:

```
statistics show-periodic -object disk:raid_group -instance
raid_group_name -counter disk_busy|user_read_latency -interval 1 -
iterations 60
```

You can decide whether reducing latency by adding Flash Pool cache makes sense for this aggregate.

### Example

The following command shows the statistics for the first RAID group of the aggregate “aggr1”:

```
statistics show-periodic -object disk:raid_group -instance /aggr1/
plex0/rg0 -counter disk_busy|user_read_latency -interval 1 -iterations
60
```

3. Start AWA:

```
system node run -node node_name waf1 awa start aggr_name
```

AWA begins collecting workload data for the volumes associated with the specified aggregate.

4. Exit advanced mode:

```
set admin
```

5. Allow AWA to run until one or more intervals of peak load have occurred.

AWA analyzes data for up to one rolling week in duration. Running AWA for more than one week will report only on data collected from the most recent week. Cache size estimates are based on the highest loads seen during the data collection period. You do not need to ensure that the load is high for the entire data collection period.

AWA collects workload statistics for the volumes associated with the specified aggregate.

6. Enter advanced mode:

```
set advanced
```

7. Display the workload analysis:

```
system node run -node node_name waf1 awa print
```

You can use the `-t` option to show information about the volumes in the aggregate that are the best candidates for being on a Flash Pool aggregate.

AWA displays the workload statistics and optimal Flash Pool cache size.

## 8. Stop AWA:

```
system node run -node node_name waf1 awa stop
```

All workload data is flushed and is no longer available for analysis.

## 9. Exit advanced mode:

```
set admin
```

**Example**

In the following example, AWA was run on aggregate “aggr1”. Here is the output of the `awa print` command after AWA had been running for about 3 days (442 10-minute intervals):

```
FP AWA Stats

 Host lada66a Memory 93788 MB
 ONTAP Version NetApp Release
R8_3_1x_awa_2809198_1504220853: Tue Apr 21 18:35:12 PDT 2015

Basic Information
 Aggregate lada66a_aggr1
 Current-time Thu Apr 23 16:42:17 PDT 2015
 Start-time Thu Apr 23 08:03:51 PDT 2015
 Total runtime (sec) 31103
 Interval length (sec) 600
 Total intervals 54
 In-core Intervals 1024

Summary of the past 20 intervals
 max

 Read Throughput (MB/s): 181.772
 Write Throughput (MB/s): 550.611
 Cacheable Read (%): 12
 Cacheable Write (%): 30
 Max Projected Cache Size (GiB): 787.077

Summary Cache Hit Rate vs. Cache Size
Referenced Cache Size (GiB): 787.077
Referenced Interval: ID 53 starting at Thu Apr 23 16:33:07 PDT 2015
 Size 20% 40% 60% 80%
100%
 Read Hit (%) 9 20 28 32
35
 Write Hit (%) 17 21 23 25 30
```

The results provide the following pieces of information:

- **Read Throughput and Write Throughput**  
The throughput measurements can help you identify an aggregate that is receiving a higher amount of traffic. Note that these numbers do not indicate whether that aggregate is I/O bound.
- **Max Projected Cache Size**  
The size at which the SSD cache would hold every eligible data block that was requested from disk during the AWA run. Note that this does not guarantee a hit for all future I/O operations, because they might request data that is not in the cache. However, if the workload during the AWA run was a typical one, and if your budget allows for it, this would be an ideal size for your Flash Pool cache.
- **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 related to the performance increase you would see by converting the aggregate to a Flash Pool aggregate, but not an exact prediction.

- **Summary Cache Hit Rate vs. Cache Size**  
This table can help you predict the performance impact of decreasing the size of the SSD cache from Max Projected Cache Size. These values are highly impacted by your workload. Depending on whether data that was aged out of the cache was ever accessed again, the impact of decreasing the size of the cache might be large or almost nonexistent. You can use this table to find the right balance between cost and performance for your workload and budget.

**Related tasks**

[Determining and enabling volume write-caching eligibility for Flash Pool aggregates](#) on page 149

**Related information**

[NetApp Technical Report 4070: Flash Pool Design and Implementation Guide](#)

# Determining and enabling volume write-caching eligibility for Flash Pool aggregates

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 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.

**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:<br><code>storage aggregate modify aggr_name -hybrid-enabled false</code><br><br>You have completed this task. All of the volumes in the aggregate are eligible for write caching. |

| If...                                                                                                             | Then...                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data ONTAP displays an error message telling you that the aggregate cannot be converted to a Flash Pool aggregate | <p>Determine which volumes are not eligible:</p> <pre><b>volume show -volume * -fields hybrid-cache-write-caching-ineligibility-reason -aggregate aggr_name</b></pre> <p>Each volume in the aggregate is listed, along with its reason for ineligibility if it is ineligible. Eligible volumes display a hyphen ("-").</p> |

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 <code>-force-hybrid-enabled</code> 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.
```

#### Related tasks

[Determining Flash Pool candidacy and optimal cache size](#) on page 147

## 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 if you need to reduce the parity overhead for your SSD RAID groups.

#### Before you begin

- You should understand the considerations for RAID type and spare management for Flash Pool cache.
- If you are changing the RAID type of the SSD cache from RAID-DP to RAID4, you should ensure that doing so will not cause you to exceed your cache size limit.

#### About this task

You can change the RAID type of the SSD cache for a Flash Pool aggregate using SSD storage pools only when you add the first storage pool allocation units to the aggregate.

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. In this case, the RAID type is also displayed for each RAID group.

All HDD RAID groups in a Flash Pool aggregate must have the same RAID type, which should be RAID-DP.

### 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 `storage aggregate show-status` 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 using physical SSDs named “test” initially have a RAID type of RAID-DP. The following command changes the RAID type of the SSD cache to RAID4, and converts the Flash Pool aggregate to the mixed RAID type:

```
storage aggregate modify -aggregate test -raidtype raid4 -disktype SSD
```

The output from the `storage aggregate show-status` command shows that the aggregate has a mixed RAID type, the HDD RAID groups have a RAID type of RAID-DP, and the SSD cache has a RAID type of RAID4.

```
storage aggregate show-status test

Aggregate test (online, mixed_raid_type, hybrid) (block checksums)
Plex /test/plex0 (online, normal, active, pool0)
 RAID Group /test/plex0/rg0 (normal, block checksums, raid-dp)

 Position Disk Pool Type RPM Usable Physical
 ----- Disk -----
 dparity 1.2.3 0 BSAS 7200 827.7GB 828.0GB (normal)
 parity 1.2.4 0 BSAS 7200 827.7GB 828.0GB (normal)
 data 1.2.5 0 BSAS 7200 827.7GB 828.0GB (normal)
 data 1.2.6 0 BSAS 7200 827.7GB 828.0GB (normal)
 data 1.2.8 0 BSAS 7200 827.7GB 828.0GB (normal)

 RAID Group /test/plex0/rg1 (normal, block checksums, raid4)

 Position Disk Pool Type RPM Usable Physical
 ----- Disk -----
 parity 1.3.3 0 SSD - 82.59GB 82.81GB (normal)
 data 1.4.0 0 SSD - 82.59GB 82.81GB (normal)
 data 1.4.1 0 SSD - 82.59GB 82.81GB (normal)
 data 1.4.2 0 SSD - 82.59GB 82.81GB (normal)
```

### Related concepts

[Considerations for RAID type and spare management for Flash Pool cache](#) on page 112

### Related information

[NetApp Hardware Universe](#)

## Determining drive and RAID group information for an aggregate

Some aggregate administration tasks require that you know what types of drives compose the aggregate, their size, checksum, and status, whether they are shared with other aggregates, and the size and composition of the RAID groups.

### Step

1. Show the drives for the aggregate, by RAID group:

```
storage aggregate show-status aggr_name
```

The drives are displayed for each RAID group in the aggregate.

You can see the RAID type of the drive (data, parity, dparity) in the `Position` column. If the `Position` column displays `shared`, then the drive is shared: if it is an HDD, it is a partitioned disk; if it is an SSD, it is part of a storage pool.

#### Example: a Flash Pool aggregate using an SSD storage pool and data partitions

```
cluster1::> storage aggregate show-status nodeA_fp_1

Owner Node: cluster1-a
Aggregate: nodeA_fp_1 (online, mixed_raid_type, hybrid) (block checksums)
Plex: /nodeA_fp_1/plex0 (online, normal, active, pool0)
RAID Group /nodeA_fp_1/plex0/rg0 (normal, block checksums, raid_dp)

 Position Disk Pool Type RPM Usable Physical

shared 2.0.1 0 SAS 10000 472.9GB 547.1GB (normal)
shared 2.0.3 0 SAS 10000 472.9GB 547.1GB (normal)
shared 2.0.5 0 SAS 10000 472.9GB 547.1GB (normal)
shared 2.0.7 0 SAS 10000 472.9GB 547.1GB (normal)
shared 2.0.9 0 SAS 10000 472.9GB 547.1GB (normal)
shared 2.0.11 0 SAS 10000 472.9GB 547.1GB (normal)

RAID Group /nodeA_flashpool_1/plex0/rg1 (normal, block checksums, raid4) (Storage
Pool: SmallSP)

 Position Disk Pool Type RPM Usable Physical

shared 2.0.13 0 SSD - 186.2GB 745.2GB (normal)
shared 2.0.12 0 SSD - 186.2GB 745.2GB (normal)

8 entries were displayed.
```

### Related concepts

[Understanding root-data partitioning](#) on page 25

[How Flash Pool SSD partitioning increases cache allocation flexibility for Flash Pool aggregates](#) on page 116

## 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.

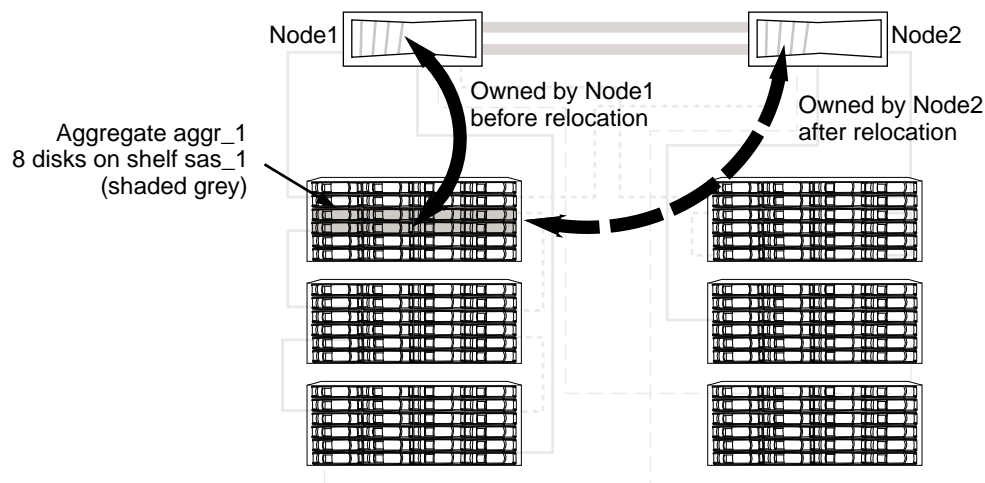


## How aggregate relocation works

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.

[Clustered Data ONTAP 8.3 Infinite Volumes Management Guide](#)

### Related tasks

[Relocating ownership of aggregates used by Infinite Volumes](#) on page 132

## How root-data partitioning affects aggregate relocation

If you have a platform model that uses root-data partitioning, also called *shared disks*, aggregate relocation processing occurs just as with physical (nonshared) disks.

The container disk ownership changes to the destination node during aggregate relocation only if the operation transfers ownership of all partitions on that physical disk to the destination node. This ownership change occurs only with permanent aggregate relocation operations.

Ownership changes that occur during negotiated storage failover takeover or giveback events are temporary.

## 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 fails 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 the following:
  - Takeover
  - Giveback
  - Shutdown
  - Another aggregate relocation operation
  - Disk ownership changes
  - Aggregate or volume configuration operations
  - Storage controller replacement
  - Data ONTAP upgrade
  - Data ONTAP revert
- If you have a MetroCluster configuration, you should not initiate aggregate relocation while disaster recovery operations (*switchover*, *healing*, or *switchback*) are in progress.
- If you have a MetroCluster configuration and initiate aggregate relocation on a switched-over aggregate, the operation might fail because it exceeds the DR partner's volume limit count.
- 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.

*Clustered Data ONTAP 8.3 Infinite Volumes Management Guide*

- Before initiating the aggregate relocation, you should 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 Node3 form an HA pair and Node2 and Node4 form an HA pair.

```
cluster::> 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 be moved only within the HA pair.

```
cluster::> 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:

```
cluster::> storage aggregate relocation show -node node1
Source Aggregate Destination Relocation Status

node1
 aggr_1 node3 In progress, module: waf1
 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 132

## 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.  | <code>storage aggregate relocation start</code> |
| Monitor the aggregate relocation process | <code>storage aggregate relocation show</code>  |

#### Related information

[Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#)

## 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                                                                                                                                                                                                            |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-node nodename</code>                 | Specifies the name of the node that currently owns the aggregate                                                                                                                                                   |
| <code>-destination nodename</code>          | Specifies the destination node where aggregates are to be relocated                                                                                                                                                |
| <code>-aggregate-list aggregate name</code> | Specifies the list of aggregate names to be relocated from source node to destination node (This parameter accepts wildcards)                                                                                      |
| <code>-override-vetoes true/false</code>    | Specifies whether to override any veto checks during the relocation operation<br>Use of this option can potentially lead to longer client outage, or aggregates and volumes not coming online after the operation. |

| Parameter                                                   | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-relocate-to-higher-version <i>true/false</i></code>  | <p>Specifies whether the aggregates are to be relocated to a node that is running a higher version of Data ONTAP than the source node</p> <ul style="list-style-type: none"> <li>You cannot perform an aggregate relocation from a node running Data ONTAP 8.2 to a node running Data ONTAP 8.3 or higher using the <code>-relocate-to-higher-version <i>true</i></code> parameter.<br/>You must first upgrade the source node to Data ONTAP 8.2.1 or higher before you can perform an aggregate relocation operation using this parameter.<br/>Similarly, you must upgrade to Data ONTAP 8.2.1 or higher before you can upgrade to Data ONTAP 8.3.</li> <li>Although you can perform aggregate relocation between nodes running different minor versions of Data ONTAP (for example, 8.2.1 to 8.2.2, or 8.2.2 to 8.2.1), you cannot perform an aggregate relocation operation from a higher major version to a lower major version (for example, 8.3 to 8.2.2).</li> </ul> |
| <code>-override-destination-checks <i>true/false</i></code> | <p>Specifies if the aggregate relocation operation should override the check performed on the destination node</p> <p>Use of this option can potentially lead to longer client outage, or aggregates and volumes not coming online after the operation.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## 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, but hard vetoes cannot be, even if forced.

You can review the EMS details for any giveback vetoes by using the following command:

```
event log show -node * -event gb*
```

You can review the EMS details for aggregate relocation by using the following command:

```
event log show -node * -event arl*
```

The following tables summarize the soft and hard vetoes, along with recommended workarounds:

**Veto checks during aggregate relocation**

| <b>Vetoing subsystem module</b> | <b>Workaround</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vol Move                        | <p>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.</p> <p>Wait for the volume move to complete.</p> <p>If this veto is overridden, cutover resumes 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 <code>volume move trigger-cutover</code> command.</p> |
| Backup                          | <p>Relocation of an aggregate is vetoed if a dump or restore job is in progress on a volume hosted by the aggregate.</p> <p>Wait until the dump or restore operation in progress is complete.</p> <p>If this veto is overridden, the backup or restore operation is aborted and must be restarted by the backup application.</p>                                                                                                                                                                                                    |
| Lock manager                    | <p>To resolve the issue, gracefully shut down the CIFS applications that have open files, or move those volumes to a different aggregate.</p> <p>Overriding this veto results in loss of CIFS lock state, causing disruption and data loss.</p>                                                                                                                                                                                                                                                                                     |
| Lock Manager NDO                | <p>Wait until the locks are mirrored.</p> <p>This veto cannot be overridden; doing so disrupts Microsoft Hyper-V virtual machines.</p>                                                                                                                                                                                                                                                                                                                                                                                              |
| RAID                            | <p>Check the EMS messages to determine the cause of the veto:</p> <p>If disk add or disk ownership reassignment operations are in progress, wait until they complete.</p> <p>If the veto is due to a mirror resync, a mirror verify, or offline disks, the veto can be overridden and the operation restarts after giveback.</p>                                                                                                                                                                                                    |

**Destination checks during aggregate relocation**

| <b>Vetoing subsystem module</b> | <b>Workaround</b>                                                                                                                                                                                                                                                                                         |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Disk Inventory                  | <p>Relocation of an aggregate fails if the destination node is unable to see one or more disks belonging to the aggregate.</p> <p>Check storage for loose cables and verify that the destination can access disks belonging to the aggregate being relocated.</p> <p>This check cannot be overridden.</p> |
| WAFL                            | <p>Relocation of an aggregate fails if the relocation would cause the destination to exceed its limits for maximum volume count or maximum volume size.</p> <p>This check cannot be overridden.</p>                                                                                                       |

| Vetoing subsystem module | Workaround                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Lock Manager NDO         | <p>Relocation of an aggregate fails if:</p> <ul style="list-style-type: none"> <li>The destination does not have sufficient lock manager resources to reconstruct locks for the relocating aggregate.</li> <li>The destination node is reconstructing locks.</li> </ul> <p>Retry aggregate relocation after a few minutes.<br/>This check cannot be overridden.</p>                                                                                                           |
| Lock Manager             | <p>Permanent relocation of an aggregate fails if the destination does not have sufficient lock manager resources to reconstruct locks for the relocating aggregate.</p> <p>Retry aggregate relocation after a few minutes.<br/>This check cannot be overridden.</p>                                                                                                                                                                                                           |
| RAID                     | <p>Check the EMS messages to determine the cause of the failure:</p> <ul style="list-style-type: none"> <li>If the failure is due to an aggregate name or UUID conflict, troubleshoot and resolve the issue. This check cannot be overridden.</li> </ul> <p>Relocating an aggregate fails 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.</p> |

## 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 in your cluster.

### Steps

1. Check the list of aggregates already assigned to the SVM:

```
vserver show -fields aggr-list
```

The aggregates currently assigned to the SVM are displayed. If there are no aggregates assigned, “-” is displayed.

2. Add or remove assigned aggregates, depending on your requirements:

| If you want to...            | Use this command...                    |
|------------------------------|----------------------------------------|
| Assign additional aggregates | <code>vserver add-aggregates</code>    |
| Unassign aggregates          | <code>vserver remove-aggregates</code> |

The listed aggregates are assigned to or removed from the SVM. If the SVM already has volumes that use an aggregate that is not assigned to the SVM, a warning message is displayed, but the command is completed successfully. Any aggregates that were already assigned to the SVM and that were not named in the command are unaffected.

### Example

In the following example, the aggregates `aggr1` and `aggr2` are assigned to SVM `svm1`:

```
vserver add-aggregates -vserver svm1 -aggregates aggr1,aggr2
```

## 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 volume-guaranteed volumes 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 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 a volume with a guarantee type of **volume**. 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.
- 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).



## 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 | <code>volume show -aggregate aggregate_name</code>                   |
| Has Infinite Volumes           | <code>volume show -is-constituent * -aggregate aggregate_name</code> |

All volumes (and, if you have Infinite Volumes, constituents) that reside on the specified aggregate are displayed.

## Determining whether a Flash Pool aggregate is using an SSD storage pool

You manage Flash Pool aggregates differently when they use SSD storage pools to provide their cache than when they use discrete SSDs.

### Step

1. Display the aggregate's drives by RAID group:

```
storage aggregate show-status aggr_name
```

If the aggregate is using one or more SSD storage pools, the value for the `Position` column for the SSD RAID groups is displayed as `Shared`, and the name of the storage pool is displayed next to the RAID group name.

## Commands for managing aggregates

You use the `storage aggregate` command to manage your aggregates.

| If you want to...                                           | Use this command...                                                                                |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Display the size of the cache for all Flash Pool aggregates | <code>storage aggregate show -fields hybrid-cache-size-total -hybrid-cache-size-total &gt;0</code> |
| Display disk information and status for an aggregate        | <code>storage aggregate show-status</code>                                                         |
| Display spare disks by node                                 | <code>storage aggregate show-spare-disks</code>                                                    |

| If you want to...                                   | Use this command...                                      |
|-----------------------------------------------------|----------------------------------------------------------|
| Display the root aggregates in the cluster          | <code>storage aggregate show -has-mroot true</code>      |
| Display basic information and status for aggregates | <code>storage aggregate show</code>                      |
| Display the type of storage used in an aggregate    | <code>storage aggregate show -fields storage-type</code> |
| Bring an aggregate online                           | <code>storage aggregate online</code>                    |
| Delete an aggregate                                 | <code>storage aggregate delete</code>                    |
| Put an aggregate into the restricted state          | <code>storage aggregate restrict</code>                  |
| Rename an aggregate                                 | <code>storage aggregate rename</code>                    |
| Take an aggregate offline                           | <code>storage aggregate offline</code>                   |
| Change the RAID type for an aggregate               | <code>storage aggregate modify -raidtype</code>          |

**Related information**

*[Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#)*

## 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](#)
- [RAID group limits](#)
- [FlexClone file and FlexClone LUN limits](#)

Storage limits for Cloud ONTAP are documented in the *Cloud ONTAP Release Notes*.

### Aggregate limits

| Storage object        | Limit                                   | Native storage                     | Storage arrays  | Data ONTAP Edge |
|-----------------------|-----------------------------------------|------------------------------------|-----------------|-----------------|
| Aggregates            | Maximum per node <sup>1</sup>           | 100                                | 100             | 60              |
|                       | Maximum size <sup>2</sup>               | Model-dependent                    | Model-dependent | 16 TB           |
|                       | Minimum size <sup>3</sup>               | RAID-DP: 5 disks<br>RAID4: 3 disks | Model-dependent | 1 disk          |
| Aggregates (mirrored) | Maximum suggested per node <sup>4</sup> | 64                                 | 64              | N/A             |
| RAID groups           | Maximum per aggregate                   | 150                                | 150             | 60              |

#### Notes:

1. In an HA configuration, this limit applies to each node individually, so the overall limit for the pair is doubled.
2. See the [Hardware Universe](#).
3. For root aggregates using physical disks, the minimum size is 3 disks for RAID-DP and 2 disks for RAID4.  
See the [Hardware Universe](#) for the minimum aggregate size for storage arrays.
4. You can create more than 64 mirrored aggregates on a node, but doing so could cause plex synchronization problems after certain types of failures.

### RAID group limits

For maximum and default RAID group sizes, see the [Hardware Universe](#).

| Limit            | Native storage | Storage arrays | Data ONTAP Edge |
|------------------|----------------|----------------|-----------------|
| Maximum per node | 400            | 400            | 60              |

| Limit                 | Native storage | Storage arrays | Data ONTAP Edge |
|-----------------------|----------------|----------------|-----------------|
| Maximum per aggregate | 150            | 150            | 60              |

#### FlexClone file and FlexClone LUN limits

| Limit                                        | Native storage | Storage arrays | Data ONTAP Edge |
|----------------------------------------------|----------------|----------------|-----------------|
| Maximum per file or LUN <sup>1</sup>         | 32,767         | 32,767         | 32,767          |
| Maximum total shared data per FlexVol volume | 640 TB         | 640 TB         | 100 TB          |

#### Note:

1. If you try to create more than 32,767 clones, Data ONTAP automatically creates a new physical copy of the parent file or LUN.  
This limit might be lower for FlexVol volumes that use deduplication.

#### Related information

*[Find the Release Notes for your version of Cloud ONTAP](#)*

## Copyright information

---

Copyright © 1994–2017 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

**RESTRICTED RIGHTS LEGEND:** Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark information

---

Active IQ, AltaVault, Arch Design, ASUP, AutoSupport, Campaign Express, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Element, Fitness, Flash Accel, Flash Cache, Flash Pool, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, Fueled by SolidFire, GetSuccessful, Helix Design, LockVault, Manage ONTAP, MetroCluster, MultiStore, NetApp, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANscreen, SANshare, SANtricity, SecureShare, Simplicity, Simulate ONTAP, Snap Creator, SnapCenter, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, SolidFire, SolidFire Helix, StorageGRID, SyncMirror, Tech OnTap, Unbound Cloud, and WAFL and other names are trademarks or registered trademarks of NetApp, Inc., in the United States, and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web.

<http://www.netapp.com/us/legal/netapptmlist.aspx>

## How to send comments about documentation and receive update notifications

---

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

[doccomments@netapp.com](mailto:doccomments@netapp.com)

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

# Index

## A

### ACP

- how you use to increase storage availability for SAS-connected disk shelves [21](#)

### active-passive configurations

- setting up with root-data partitioning [32](#)

### adding

- key management servers [75](#)

### Advanced Drive Partitioning

- storage pools [116](#)
- See also* root-data partitioning

### aggregate ownership

- relocation of [152](#)

### aggregate relocation

- benefits of [153](#)
- commands for [156](#)
- effect on root-data partitioning [154](#)
- effect on shared disks [154](#)
- how it works [153](#)
- Infinite Volumes [132](#)
- monitoring progress of [157](#)
- overriding a veto of [157](#)

### aggregate show-space command

- how to determine aggregate space usage by using [126](#)

### aggregates

- adding unpartitioned drives or array LUNs to [137](#)
- assigning to SVMs [159](#)
- associated with Infinite Volumes [130](#)
- changing size of RAID groups for [98](#)
- commands for displaying space usage information [43](#)
- commands for managing [161](#)
- configuration requirements for multi-disk carrier shelves [24](#)
- considerations for sizing RAID groups [97](#)
- considerations for using disks from multi-disk carriers in [25](#)
- creating Flash Pool [143](#)
- creating Flash Pool using SSD storage pools [145](#)
- creating SSD storage pool for Flash Pool [119](#)
- creating using root-data partitioning [136](#)
- creating using shared HDDs [136](#)
- creating using unpartitioned drives [134](#)
- description and characteristics of [108](#)
- determination of checksum type of array LUN [125](#)
- determining candidacy and optimal cache size for Flash Pool [147](#)
- determining drive information for [152](#)
- determining RAID group information for [152](#)
- determining which volumes reside on [161](#)
- disk speeds supported by Data ONTAP [11](#)
- effect of SVM on selection [122](#)
- Flash Pool, defined [111](#)
- Flash Pool, determining volume write-caching eligibility for [149](#)
- Flash Pool, determining whether using a storage pool [161](#)

### Flash Pool, how they work [111](#)

- how associated with SVMs with Infinite Volume [130](#)
  - how cache capacity is calculated for Flash Pool [113](#)
  - how drive checksum types affect management [12](#)
  - how root-data partitioning affects storage management [26](#)
  - how storage classes use for Infinite Volumes [130](#)
  - how storage pools increase cache allocation flexibility for Flash Pool [116](#)
  - how to determine space usage in [126](#)
  - how you use storage pools with Flash Pool [118](#)
  - increasing size of, when they use root-data partitioning [139](#)
  - increasing the size of with unpartitioned drives [137](#)
  - introduction to managing [134](#)
  - maximum and minimum size of [163](#)
  - maximum per node [163](#)
  - maximum size, methods of calculating [11](#)
  - methods of creating space in [160](#)
  - mirrored, explained [109](#)
  - mirrored, maximum per node [163](#)
  - ownership change [154](#)
  - relocating ownership of [132](#)
  - relocation of [153](#), [154](#)
  - replacing disks used in [36](#)
  - requirements and best practices for using storage pools with Flash Pool [119](#)
  - requirements for Infinite Volumes [129](#)
  - requirements for storage classes [130](#)
  - requirements for using Flash Pool [112](#)
  - root, which drives are partitioned for use in [27](#)
  - rules about mixing storage types in [125](#)
  - rules for mixing HDD types in [124](#)
  - rules for storage array families [125](#)
  - sharing between FlexVol volumes and Infinite Volumes [129](#)
  - tips for creating and backing up, for sensitive data [18](#)
  - unmirrored, explained [108](#)
  - ways to use disks with mixed speeds in [123](#)
  - what happens when adding storage to [143](#)
- ### All-Flash FAS models
- management differences from FAS models [21](#)
- ### All-Flash FAS platforms
- how root-data partitioning works for [25](#)
- ### Alternate Control Path
- ACP [21](#)
- ### array LUN ownership
- how it works [54](#)
- ### array LUNs
- adding to aggregates [137](#)
  - considerations for sizing RAID groups for [97](#)
  - how to control selection from heterogeneous storage [123](#)
  - reasons to assign ownership of [45](#), [54](#)
  - verifying back-end configuration for [59](#)
  - See also* systems running Data ONTAP that can use
- ### assigning
- aggregates [130](#)
- ### ATA drives



- how Data ONTAP reports disk types [9](#)
- authentication keys
  - changing [80](#)
  - deleting [83](#)
  - how Storage Encryption uses [66](#)
  - removing management server that stores [77](#)
  - retrieving [82](#)
- autoassignment
  - automatic ownership assignment [46](#)
- Automated Workflow Analyzer
  - AWA [147](#)
- automatic ownership assignment
  - configuring for disks [50](#)
  - guidelines for disks [48](#)
  - how it works for disks [46](#)
  - when it is invoked [47](#)
  - which policy to use for disk [47](#)
- automatic RAID-level scrubs
  - changing the schedule for [103](#)
- availability
  - how Data ONTAP uses RAID to ensure data [92](#)
- AWA
  - determining Flash Pool optimal cache size by using [147](#)
- AZCS type checksums
  - configuration rules [12](#)
  - effect on aggregate management [12](#)
  - effect on spare management [12](#)

**B**

- back-end configurations
  - verifying [59](#)
- BCS type checksums
  - configuration rules [12](#)
  - effect on aggregate management [12](#)
  - effect on spare management [12](#)
- benefits
  - of Storage Encryption [67](#)
- best practices
  - for using storage pools [119](#)
- block checksum type
  - changing for array LUNs [63](#)
- BSAS drives
  - how Data ONTAP reports disk types [9](#)

## C

- cache capacity
  - how calculated for Flash Pool aggregates [113](#)
- cache size
  - determining impact to, when adding SSDs to storage pool [121](#)
  - determining optimal for Flash Pool aggregates [147](#)
- cache storage
  - requirements and best practices for using storage pools for Flash Pool aggregate [119](#)
- caches
  - comparison of Flash Pool and Flash Cache [113](#)
- caching policies
  - Flash Pool aggregates, how they work [114](#)
  - modifying [116](#)

- capacity
  - how it is allocated in new Infinite Volumes [131](#)
  - methods of calculating aggregate and system [11](#)
- carriers
  - determining when to remove multi-disk [24](#)
  - how Data ONTAP avoids RAID impact when removing multi-disk [23](#)
  - spare requirements for multi-disk [24](#)
- certificates
  - installing replacement SSL [84](#)
  - installing SSL, on cluster [71](#)
  - preventing SSL expiration issues [83](#)
  - removing old SSL [83](#)
  - SSL requirements [71](#)
- changing
  - authentication keys [80](#)
  - RAID group size [98](#)
  - RAID type for Flash Pool cache [150](#)
- checksums
  - checking the type [62](#)
  - configuration rules [12](#)
  - rules for aggregates [125](#)
  - type, changing for array LUNs [63](#)
  - type, effect on aggregate and spare management [12](#)
- commands
  - aggregate management, list of [161](#)
  - disk management, list of [42](#)
  - for displaying aggregate space usage information [43](#)
  - for displaying FlexVol volume space usage information [43](#)
  - for displaying information about storage shelves [43](#)
  - how you use wildcard character with disk ownership [51](#)
  - SSD storage pool management, list of [122](#)
  - storage aggregate [42](#), [43](#), [161](#)
  - storage disk [42](#)
  - storage pool [122](#)
  - storage pool management, list of [122](#)
  - storage shelves, displaying information about [43](#)
  - volume show-footprint [43](#)
  - volume show-space [43](#)
  - volume snapshot [43](#)
- comments
  - how to send feedback about documentation [167](#)
- composition
  - changing array LUN [64](#)
- configuring
  - automatic ownership assignment of disks [50](#)
- connection types
  - how disks can be combined for SAS [11](#)
  - supported storage [10](#)
- constituents
  - determining which ones reside on an aggregate [161](#)
- continuous media scrubbing
  - how Data ONTAP uses, to prevent media errors [20](#)
  - impact on system performance [20](#)
  - reasons it should not replace scheduled RAID-level disk scrubs [21](#)
- core files
  - spare disk requirement for [99](#)
- creating
  - aggregates using unpartitioned drives [134](#)
  - Flash Pool aggregates [143](#)

Flash Pool aggregates using SSD storage pools [145](#)  
 current owner  
 disk ownership type, defined [45](#)

## D

data  
   how Data ONTAP uses RAID to protect and ensure availability [92](#)  
   tips for creating and backing up aggregates containing sensitive [18](#)  
   using sanitization to remove from disks [40](#)  
 data at rest  
   introduction to securing with Storage Encryption [66](#)  
 data disks  
   removing [39](#)  
 data integrity  
   how RAID-level disk scrubs verify [103](#)  
 Data ONTAP disk types  
   comparison with industry standard [9](#)  
 Data ONTAP Edge  
   how disk ownership works [47](#)  
 Data ONTAP-v  
   how disk ownership works [47](#)  
 data protection  
   in case of disk loss or theft [67](#)  
   through emergency shredding [68](#)  
   when moving disks to end-of-life [68](#)  
   when returning disks to vendors [68](#)  
 data reconstruction  
   controlling performance impact of RAID [105](#)  
 data shredding  
   performing emergency, on disks using Storage Encryption [88](#)  
 degraded mode [101](#)  
 deleting  
   authentication keys [83](#)  
 destroying data on disks  
   using Storage Encryption [87](#)  
 disk  
   performance monitors [18](#)  
 disk connection types  
   how disks can be combined for SAS [11](#)  
 disk operations  
   with SEDs [67](#)  
 disk ownership  
   application to array LUNs [45, 55, 56](#)  
   application to disks [45, 55](#)  
   assigning array LUNs [58](#)  
   configuring automatic assignment [50](#)  
   how it works [54](#)  
   how it works for Data ONTAP Edge [47](#)  
   how it works for Data ONTAP-v [47](#)  
   ownership  
     removing array LUN ownership [65](#)  
     removing array LUN ownership [65](#)  
 disk ownership commands  
   how you use wildcard character with [51](#)  
 Disk Qualification Package  
   when you need to update [35](#)  
 disk remove -w  
   removing an array LUN [65](#)  
 disk sanitization

  introduction to how it works [16](#)  
   process described [16](#)  
   when it cannot be performed [17](#)  
 disk scrubbing  
   reasons continuous media scrubbing should not replace scheduled RAID-level [21](#)  
 disk shelves  
   aggregate configuration requirements for multi-disk carrier [24](#)  
   commands for displaying information about [43](#)  
   configuration requirements for multi-disk carrier [24](#)  
   how you use ACP to increase storage availability for SAS-connected [21](#)  
   requirements for using multi-disk carrier [23](#)  
 disk types  
   how to control selection from heterogeneous storage [123](#)  
 disks  
   adding to a node [34](#)  
   adding to aggregates [137](#)  
   assigning ownership for [48](#)  
   commands for managing [42](#)  
   considerations for removing from storage systems [37](#)  
   considerations for using, from multi-disk carriers in aggregates [25](#)  
   data, converting to spare [36](#)  
   determining information about, for aggregates [152](#)  
   displaying information about Storage Encryption [79](#)  
   evacuation process, about [23](#)  
   guidelines for assigning ownership [48](#)  
   how automatic ownership assignment works [46](#)  
   how available for Data ONTAP use [45, 55](#)  
   how Data ONTAP handles failed, with available hot spares [102](#)  
   how Data ONTAP handles failed, with no available hot spare [102](#)  
   how Data ONTAP reduces failures using Rapid RAID Recovery [18](#)  
   how Data ONTAP reports types [9](#)  
   how low spare warnings can help you manage spare [101](#)  
   how RAID-level scrubs verify data integrity [103](#)  
   how root-data partitioning affects storage management [26](#)  
   how shared HDDs work [25](#)  
   how they can be combined for SAS connection type [11](#)  
   how to control selection from heterogeneous storage [123](#)  
   introduction to how DATA ONTAP works with heterogeneous storage [123](#)  
   introduction to managing ownership for [45](#)  
   loop IDs for FC-AL connected, about [15](#)  
   managing using Data ONTAP [9](#)  
   matching spares defined [100](#)  
   minimum required hot spare [99](#)  
   performing emergency data shredding on Storage Encryption [88](#)  
   RAID drive types, defined [15, 96](#)  
   RAID protection levels for [92](#)  
   reasons to assign ownership of [45, 54](#)  
   removing data [39](#)  
   removing failed [37](#)

- removing hot spares [38](#)
  - removing ownership from [49](#)
  - replacing in aggregate [36](#)
  - requirements for using partitioned [28](#)
  - rules for mixing HDD types in aggregates [124](#)
  - sanitization process described [16](#)
  - sanitization, what happens if interrupted [17](#)
  - sanitizing [87](#)
  - setting state to end-of-life [88](#)
  - spare requirements for multi-disk carrier [24](#)
  - spare, appropriate [100](#)
  - speeds supported by Data ONTAP [11](#)
  - SSD and HDD capability differences [22](#)
  - stopping sanitization [41](#)
  - types of ownership for [45](#)
  - unpartitioned, adding to aggregates [137](#)
  - using sanitization to remove data from [40](#)
  - ways to mix speed of, in aggregates [123](#)
  - what happens when Data ONTAP takes them offline [18](#)
  - when automatic ownership assignment is invoked [47](#)
  - when sanitization cannot be performed [17](#)
  - when they can be put into maintenance center [20](#)
  - when you need to update the Disk Qualification Package for [35](#)
  - which autoassignment policy to use for [47](#)
  - disks, self-encrypting
    - considerations for running in FIPS compliance mode [73](#)
    - introduction to running in FIPS compliance mode [73](#)
    - taking out of FIPS compliance mode [74](#)
  - displaying
    - key management server information [77](#)
    - key management server status [76](#)
    - Storage Encryption disk information [79](#)
  - documentation
    - how to receive automatic notification of changes to [167](#)
    - how to send feedback about [167](#)
  - DQP
    - Disk Qualification Package [35](#)
  - DR home owner
    - disk ownership type, defined [45](#)
  - drive errors
    - how the maintenance center helps prevent [19](#)
  - drive types
    - RAID, defined [15](#), [96](#)
  - drives
    - considerations for sizing RAID groups for [97](#)
    - disks [9](#)
    - how Data ONTAP handles failed, with available hot spares [102](#)
    - how Data ONTAP handles failed, with no available hot spares [102](#)
    - how low spare warnings can help you manage spare [101](#)
    - name formats [12](#)
    - pre-cluster name formats [13](#)
    - rules for mixing types in Flash Pool aggregates [125](#)
    - which are partitioned for root-data partitioning [27](#)
- ## E
- emergency data shredding
    - data protection through [68](#)
    - performing on disks using Storage Encryption [88](#)
  - end-of-life
    - setting disk state to [88](#)
  - entry-level platforms
    - how root-data partitioning works for [25](#)
    - understanding root-data partitioning for [25](#)
  - errors
    - how Data ONTAP uses media scrubbing to prevent media [20](#)
    - how the maintenance center helps prevent drive [19](#)
  - evacuation process
    - for disks, about [23](#)
  - existing storage pools
    - considerations for adding SSDs to [121](#)
  - external key management servers
    - defined [66](#)
    - displaying information about [77](#)
- ## F
- failed disks
    - removing [37](#)
  - family
    - defined [125](#)
  - FAS models
    - management differences from All-Flash FAS models [21](#)
  - FC storage connection type
    - how disks can be combined for [11](#)
    - support for [10](#)
  - FCAL drives
    - how Data ONTAP reports disk types [9](#)
  - feedback
    - how to send comments about documentation [167](#)
  - Fibre Channel
    - FC [10](#)
  - FIPS compliance mode
    - considerations for running self-encrypting disks in [73](#)
    - introduction to running SEDs in [73](#)
    - meaning for self-encrypting disks [73](#)
    - putting self-encrypting disks into [74](#)
    - taking self-encrypting disks out of [74](#)
  - Flash Cache
    - compared with Flash Pool aggregates [113](#)
  - Flash Pool aggregates
    - caching policies, how they work [114](#)
    - changing RAID type [150](#)
    - compared with Flash Cache [113](#)
    - creating [143](#)
    - creating SSD storage pool for [119](#)
    - creating using SSD storage pools [145](#)
    - defined [111](#)
    - determining candidacy and optimal cache size for [147](#)
    - determining whether using a storage pool [161](#)
    - how cache capacity is calculated for [113](#)
    - how storage pools increase cache allocation flexibility for [116](#)

- how they work [111](#)
  - how you use storage pools with [118](#)
  - RAID type, considerations for [112](#)
  - requirements and best practices for using storage pools with [119](#)
  - requirements for using [112](#)
  - rules for mixing drive types in [125](#)
  - spare management, considerations for [112](#)
  - volume write-caching eligibility, determining [149](#)
- Flash Pool SSD partitioning
  - how it increases cache allocation flexibility for Flash Pool aggregates [116](#)
- FlexClone files
  - FlexClone LUNs
    - maximum per file or LUN [164](#)
    - maximum shared data per volume [164](#)
    - maximum size of volume [164](#)
  - maximum per file or LUN [164](#)
  - maximum shared data per volume [164](#)
  - maximum size of volume [164](#)
- FlexVol volumes
  - aggregate sharing with Infinite Volumes [129](#)
  - commands for displaying space usage information [43](#)
  - creating aggregates using unpartitioned drives [134](#)
  - determining which ones reside on an aggregate [161](#)
  - effect of SVM on aggregate selection [122](#)
- formats
  - drive name [12](#)
  - pre-cluster drive name [13](#)
- FSAS drives
  - how Data ONTAP reports disk types [9](#)

## G

- groups
  - RAID, how they work [96](#)
- guidelines
  - assigning disk ownership [48](#)

## H

- hard disk drives
  - HDDs [21, 22](#)
- HDD RAID groups
  - sizing considerations for [97](#)
- HDDs
  - assigning ownership for shared [49](#)
  - capability differences with SSDs [22](#)
  - creating aggregates using shared [136](#)
  - increasing size of aggregates that use shared [139](#)
  - rules for mixing types in aggregates [124](#)
  - shared, how they work [25](#)
  - speeds supported by Data ONTAP [11](#)
  - standard layouts for shared [27](#)
- heterogeneous storage
  - how to control disk selection from [123](#)
  - introduction to how Data ONTAP works with [123](#)
- high-performance aggregates
  - Flash Pool, defined [111](#)
- home owner
  - disk ownership type, defined [45](#)

- hot spares
  - appropriate [100](#)
  - defined [99](#)
  - how Data ONTAP handles failed disks with available [102](#)
  - how Data ONTAP handles failed disks with no available [102](#)
  - matching, defined [100](#)
  - minimum needed [99](#)
  - removing [38](#)
  - what disks can be used as [100](#)
- hybrid aggregates
  - Flash Pool aggregates [111](#)

## I

- IDs
  - about loop, for FC-AL connected disks [15](#)
- increasing
  - aggregate size using unpartitioned drives [137](#)
- Infinite Volumes
  - aggregate relocation [132](#)
  - aggregate requirements [129](#)
  - associated aggregates [130](#)
  - capacity allocation [131](#)
  - creating aggregates using unpartitioned drives [134](#)
  - determining which constituents reside on an aggregate [161](#)
  - how storage classes use aggregates for [130](#)
  - how to determine space usage for constituents [127](#)
  - relocating aggregates [132](#)
  - space allocation [131](#)
- InfiniteVol
  - Infinite Volumes [130](#)
- information
  - how to send feedback about improving documentation [167](#)
- initialization
  - performing on node to configure root-data partitioning [29](#)
- installing
  - replacement SSL certificates [84](#)
  - SSL certificates on cluster [71](#)

## K

- Key Management Interoperability Protocol
  - using for communication with key management servers [66](#)
- key management servers
  - adding [75](#)
  - displaying information about [77](#)
  - displaying status [76](#)
  - external, defined [66](#)
  - introduction to using SSL for secure communication [70](#)
  - precautions taken against unreachability during boot process [78](#)
  - removing [77](#)
  - verifying links [76](#)
- keys
  - changing authentication [80](#)

- how Storage Encryption uses authentication [66](#)
  - retrieving authentication [82](#)
- KMIP
  - Key Management Interoperability Protocol [66](#)
- L**
- layouts
  - standard shared HDD [27](#)
- levels
  - RAID protection, for disks [92](#)
- licenses
  - for using array LUNs [54](#)
  - installing for array LUN use [54](#)
  - V\_StorageAttach [54](#)
- limitations
  - Storage Encryption [68](#)
- limits
  - aggregate storage [163](#)
  - FlexClone file and LUN storage [163](#)
  - RAID group storage and size [163](#)
  - volume storage [163](#)
- loop IDs
  - about FC-AL connected disk [15](#)
- loops
  - configuring automatic ownership assignment for [50](#)
- low spare warnings
  - how they can help you manage spare drives [101](#)
- LUNs (array)
  - assigning ownership of [58](#)
  - changing checksum type [63](#)
  - changing ownership assignment [60](#)
  - changing size or composition [64](#)
  - checking the checksum type of [62](#)
  - Data ONTAP owning [56](#)
  - Data ONTAP RAID groups with [99](#)
  - examples of when Data ONTAP can use [57](#)
  - how available for Data ONTAP use [45](#), [55](#)
  - managing through Data ONTAP [53](#)
  - names
    - format of [61](#)
  - prerequisites to changing composition [63](#)
  - prerequisites to changing size [63](#)
  - RAID protection for [93](#)
  - reasons to assign ownership of [45](#), [54](#)
  - reasons you might assign to a system [56](#)
  - requirements before removing a system running Data ONTAP from service [65](#)
  - rules about mixing storage types in aggregates [125](#)
  - setting them up in Data ONTAP [53](#)
  - systems running Data ONTAP that can use [53](#)
  - verifying back-end configuration for [59](#)
- LUNs, array [53](#)
  - See also* LUNs (array)
- M**
- maintenance center
  - how it helps prevent drive errors [19](#)
  - when disks go into [20](#)
- managing
  - Storage Encryption [75](#)
- manual RAID-level scrubs
  - how to run [104](#)
- matching spare disks
  - defined [100](#)
- media errors
  - how Data ONTAP uses media scrubbing to prevent [20](#)
- media scrubbing
  - how Data ONTAP uses, to prevent media errors [20](#)
  - impact on system performance [20](#)
  - reasons it should not replace scheduled RAID-level disk scrubs [21](#)
- mirror verification
  - controlling performance impact [106](#)
- mirrored aggregates
  - explained [109](#)
- modifying
  - caching policies [116](#)
- MSATA drives
  - how Data ONTAP reports disk types [9](#)
- MSIDs
  - rekeying SEDs to [84](#)
- multi-disk carrier shelves
  - aggregate configuration requirements for [24](#)
  - configuration requirements for [24](#)
  - in aggregates, considerations for using disks from [25](#)
  - requirements for using [23](#)
- multi-disk carriers
  - determining when to remove [24](#)
  - how Data ONTAP handles when removing [23](#)
  - spare requirements for [24](#)
- N**
- name format
  - array LUNs [60](#)
- names
  - formats for drive [12](#)
  - formats for pre-cluster drive [13](#)
- new storage pools
  - considerations for adding SSDs to [121](#)
- NL-SAS drives
  - how Data ONTAP reports disk types [9](#)
- node behavior
  - All-Flash FAS [21](#)
- nodes
  - adding disks to [34](#)
  - how to control disk selection from heterogeneous storage on [123](#)
  - initializing for root-data partitioning [29](#)
- O**
- offline
  - what happens when Data ONTAP takes disks [18](#)
- original owner
  - disk ownership type, defined [45](#)
- owner
  - disk ownership type, defined [45](#)
- ownership
  - assigning array LUNs [58](#)
  - assigning for disks [48](#)

- assigning for shared HDDs [49](#)
- automatically assigning to a stack or shelf [50](#)
- disk, types of [45](#)
- guidelines for assigning disk [48](#)
- how it works for disks and array LUNs [54](#)
- introduction to managing, for disks [45](#)
- reasons to assign disk and array LUN [45](#), [54](#)
- removing from disks [49](#)

- ownership assignment

- when it is invoked for disks [47](#)

- ownership commands

- how you use wildcard character with disk [51](#)

## P

- partitioning

- how Flash Pool SSD increases cache allocation

- flexibility for [116](#)

- root-data, assigning ownership for HDDs partitioned for [49](#)

- root-data, creating aggregates using [136](#)

- root-data, how it works [25](#)

- root-data, how storage management is affected by [26](#)

- root-data, increasing the size of an aggregate using [139](#)

- root-data, requirements for using [28](#)

- root-data, standard layouts for [27](#)

- root-data, understanding [25](#)

- root-data, which drives are partitioned for [27](#)

- setting up active-passive configuration on node using root-data [32](#)

- partitions

- correcting misaligned spare [142](#)

- performance

- controlling impact of RAID data reconstruction [105](#)

- impact of media scrubbing on system [20](#)

- persistent reservations

- releasing all [65](#)

- physical secure IDs

- PSIDs [89](#)

- platforms

- understanding root-data partitioning for entry-level [25](#)

- plex resynchronization

- controlling performance impact of [106](#)

- plexes

- mirrored aggregate, explained [109](#)

- policies

- autoassignment, which to use for disks [47](#)

- modifying caching [116](#)

- pools

- considerations for when to use SSD storage [118](#)

- requirements and best practices for using storage [119](#)

- protection

- how Data ONTAP uses to RAID for data [92](#)

- RAID levels for disks [92](#)

- PSIDs

- how the factory resets SEDs with [89](#)

- SEDs that include the functionality [90](#)

- using to reset SED to factory original settings [90](#)

## R

- RAID

- avoiding impact to, when replacing multi-disk

- carriers [23](#)

- data reconstruction, controlling performance impact [105](#)

- drive types defined [15](#), [96](#)

- how Data ONTAP to protect data and data

- availability [92](#)

- how disk scrubs verify data integrity [103](#)

- operations, controlling performance impact [104](#)

- protection levels for disks [92](#)

- protection with SyncMirror and [93](#)

- scrub, controlling performance impact [105](#)

- scrubs, changing the automatic schedule for [103](#)

- type, changing for Flash Pool cache [150](#)

- type, determining for Flash Pool cache [112](#)

- RAID groups

- changing size of [98](#)

- definition [96](#)

- determining information about, for aggregates [152](#)

- how they work [96](#)

- maximum per aggregate [163](#)

- maximum per node [163](#)

- naming convention [97](#)

- sizing considerations for [97](#)

- what happens when adding storage to aggregates in [143](#)

- with array LUNs, considerations [99](#)

- RAID protection

- for array LUNs [93](#)

- RAID-DP

- described [92](#)

- RAID-level scrubs

- changing automatic schedule for [103](#)

- how to run manual [104](#)

- reasons media scrubbing should not replace

- scheduled [21](#)

- raid.timeout option

- considerations for changing [103](#)

- RAID0

- how Data ONTAP uses for array LUNs [93](#)

- use by Data ONTAP [93](#)

- RAID4

- described [93](#)

- Rapid RAID Recovery

- how Data ONTAP reduces disk failures using [18](#)

- rekeying

- SEDs to MSID [84](#)

- relocating aggregates

- Infinite Volumes [132](#)

- relocation

- aggregate ownership [152](#), [154](#)

- of aggregates [152–154](#)

- removing

- data disks [39](#)

- data, using disk sanitization [40](#)

- failed disks [37](#)

- hot spare disks [38](#)

- key management servers [77](#)

- multi-disk carriers, determining when it is safe [24](#)

- old SSL certificates [83](#)



- replacing
  - disks in aggregates [36](#)
- requirements
  - Flash Pool aggregate use [112](#)
  - for using root-data partitioning [28](#)
  - for using storage pools [119](#)
  - Infinite Volumes, aggregate [129](#)
- resetting
  - SEDs to factory original settings [90](#)
- resynchronization
  - controlling performance impact of plex [106](#)
- retrieving
  - authentication keys [82](#)
- returning SEDs to service
  - when authentication keys are no longer available [85](#)
- root data partitioning
  - initializing node to configure [29](#)
- root-data partitioning
  - assigning ownership for HDDs using [49](#)
  - correcting misaligned spare partitions for [142](#)
  - creating aggregates using [136](#)
  - effect on aggregate relocation [154](#)
  - how it works [25](#)
  - how storage management is affected by [26](#)
  - increasing the size of an aggregate using [139](#)
  - requirements for using [28](#)
  - setting up active-passive configuration on node using [32](#)
  - standard layouts for [27](#)
  - understanding [25](#)
  - which drives are partitioned for [27](#)
- rules
  - for mixing drive types in Flash Pool aggregates [125](#)
  - for mixing HDD types in aggregates [124](#)

## S

- sanitization
  - disk process described [16](#)
  - disk, introduction to how it works [16](#)
  - stopping disk [41](#)
  - tips for creating and backing up aggregates containing sensitive data [18](#)
  - using to remove data from disks [40](#)
  - what happens if interrupted [17](#)
  - when it cannot be performed [17](#)
- sanitizing
  - disks [87](#)
- SAS
  - storage connection type, support for [10](#)
- SAS drives
  - how Data ONTAP reports disk types [9](#)
- SAS-connected shelves
  - how disks can be combined for [11](#)
  - how you use ACP to increase storage availability for [21](#)
- SATA drives
  - how Data ONTAP reports disk types [9](#)
- scrubbing
  - how Data ONTAP uses media, to prevent media errors [20](#)
  - impact of media, on system performance [20](#)

- media, reasons it should not replace scheduled RAID-level disk scrubs [21](#)
- scrubs
  - changing automatic schedule for RAID-level [103](#)
  - controlling performance impact of RAID [105](#)
  - how to run manual RAID-level [104](#)
  - RAID-level, how they verify data integrity [103](#)
- secure communication
  - introduction to using SSL for [70](#)
- SEDs
  - considerations for running in FIPS compliance mode [73](#)
  - disk operations with [67](#)
  - how Storage Encryption works with [66](#)
  - how the factory resets with PSID [89](#)
  - putting into FIPS compliance mode [74](#)
  - resetting to factory original settings [90](#)
  - returning to service when authentication keys are no longer available [85](#)
  - returning to unprotected mode [84](#)
  - taking out of FIPS compliance mode [74](#)
  - that have PSID functionality [90](#)
- self-encrypting disks
  - considerations for running in FIPS compliance mode [73](#)
  - introduction to running in FIPS compliance mode [73](#)
  - meaning of FIPS compliance mode [73](#)
  - putting into FIPS compliance mode [74](#)
  - returning to service when authentication keys are no longer available [85](#)
  - SEDs [66](#), [67](#)
  - taking out of FIPS compliance mode [74](#)
- serial-attached SCSI
  - SAS [10](#)
- setting up
  - array LUNs [53](#)
  - Storage Encryption [69](#)
- setup wizards
  - running the Storage Encryption [72](#)
- shared HDDs
  - assigning ownership for [49](#)
  - creating aggregates using [136](#)
  - how they work [25](#)
  - increasing size of aggregates that use [139](#)
  - understanding [25](#)
- shared layouts
  - standard HDD [27](#)
- shared SSDs
  - storage pools [116](#)
- shelves
  - aggregate configuration requirements for multi-disk carrier [24](#)
  - commands for displaying information about [43](#)
  - configuration requirements for multi-disk carrier [24](#)
  - configuring automatic ownership assignment for [50](#)
  - how you use ACP to increase storage availability for SAS-connected [21](#)
  - requirements for using multi-disk carrier [23](#)
- shredding
  - performing emergency data, on disks using Storage Encryption [88](#)
- size
  - changing array LUN size [63](#)

- sizes
  - changing array LUN [64](#)
- sizing
  - RAID groups, considerations for [97](#)
- Snapshot reserve
  - commands for displaying size of [43](#)
- solid-state disks
  - SSDs [22](#)
- space
  - commands for displaying usage information [43](#)
  - how it is allocated in new Infinite Volumes [131](#)
  - methods of creating in an aggregate [160](#)
- space usage
  - how to determine and control volume, in aggregates [127](#)
  - how to determine in an aggregate [126](#)
- spare array LUNs
  - changing array LUN assignment [60](#)
  - changing ownership assignment [60](#)
  - checking the type [62](#)
  - disk ownership [60](#)
- spare disks
  - appropriate [100](#)
  - defined [99](#)
  - how checksum types affect management [12](#)
  - how Data ONTAP handles failed disks with available [102](#)
  - how Data ONTAP handles failed disks with no available [102](#)
  - how low spare warnings can help you manage [101](#)
  - managing for Flash Pool cache [112](#)
  - matching, defined [100](#)
  - minimum needed [99](#)
  - removing [38](#)
  - removing ownership from [49](#)
  - requirements for multi-disk carriers [24](#)
  - what disks can be used as [100](#)
- spare partitions
  - correcting misaligned [142](#)
- speeds
  - disk, supported by Data ONTAP [11](#)
  - ways to mix disk, in aggregates [123](#)
- SSD storage pools
  - commands for managing [122](#)
  - creating Flash Pool aggregates using [145](#)
  - storage pools [116](#)
- SSDs
  - adding to storage pools [120](#)
  - capability differences with HDDs [22](#)
  - changing size of RAID groups for [98](#)
  - considerations for adding to existing storage pool versus new one [121](#)
  - considerations for when to use storage pools [118](#)
  - determining impact to cache size of adding to storage pool [121](#)
  - how Data ONTAP manages wear life [22](#)
  - how Data ONTAP reports disk types [9](#)
  - how used in Flash Pool aggregates [111](#)
  - introduction to using [21](#)
  - shared
    - storage pools [116](#)
  - sizing considerations for RAID groups [97](#)
  - storage pools, creating [119](#)
  - storage pools, determining when used by a Flash Pool aggregate [161](#)
- SSL certificates
  - installing on cluster [71](#)
  - installing replacement [84](#)
  - preventing expiration issues [83](#)
  - removing old [83](#)
  - requirements [71](#)
- SSL connections
  - introduction to using for secure key management communication [70](#)
- stacks
  - configuring automatic ownership assignment for [50](#)
- standard layouts
  - shared HDD [27](#)
- state of disks
  - setting to end-of-life [88](#)
- stopping
  - disk sanitization [41](#)
- storage
  - adding to SSD storage pools [120](#)
  - how root-data partitioning affects management of [26](#)
  - how to control disk selection from heterogeneous [123](#)
  - what happens when adding to an aggregate [143](#)
- storage aggregate commands
  - for displaying space information [43](#)
  - for managing aggregates [161](#)
  - for managing disks [42](#)
- storage aggregate relocation start command
  - key parameters of [156](#)
- storage arrays
  - rules about mixing in aggregates [125](#)
- storage connection types
  - supported [10](#)
- storage disk commands
  - for managing disks [42](#)
- Storage Encryption
  - benefits [67](#)
  - destroying data using [87](#)
  - displaying disk information [79](#)
  - explained [66](#)
  - how it works [66](#)
  - information to collect before configuring [69](#)
  - installing replacement SSL certificates [84](#)
  - installing SSL certificates for [71](#)
  - introduction to securing data at rest with [66](#)
  - introduction to using SSL for secure key management communication [70](#)
  - limitations [68](#)
  - managing [75](#)
  - overview [66](#)
  - performing emergency data shredding [88](#)
  - preventing SSL certificate expiration issues [83](#)
  - purpose of external key management server [66](#)
  - removing old SSL certificates [83](#)
  - running the setup wizard [72](#)
  - sanitizing disks using [87](#)
  - setting disk state to end-of-life [88](#)
  - setting up [69](#)
  - SSL certificates requirements [71](#)
- storage limits
  - aggregate [163](#)



- FlexClone file and LUN [163](#)
- RAID group [163](#)
- volume [163](#)
- storage performance
  - introduction to using SSDs to increase [21](#)
  - performance
    - introduction to using SSDs to increase storage [21](#)
- storage pools
  - adding SSDs to [120](#)
  - advantages of SSD [118](#)
  - commands for managing [122](#)
  - considerations for adding SSDs to new versus existing [121](#)
  - considerations for when to use SSD [118](#)
  - creating [119](#)
  - creating Flash Pool aggregates using SSD [145](#)
  - determining impact to cache size of adding SSDs to [121](#)
  - determining when used by a Flash Pool aggregate [161](#)
  - disadvantages of SSD [118](#)
  - how they increase cache allocation for Flash Pool aggregates [116](#)
  - how you use [118](#)
  - requirements and best practices for [119](#)
- storage shelves
  - commands for displaying information about [43](#)
  - requirements for using multi-disk carrier [23](#)
- storage systems
  - considerations for removing disks from [37](#)
- suggestions
  - how to send feedback about documentation [167](#)
- SVMs
  - assigning aggregates to [159](#)
  - effect on aggregate selection [122](#)
- SVMs with Infinite Volume
  - aggregate requirements [129](#)
  - assigning aggregates [130](#)
  - delegation [130](#)
- SyncMirror
  - protection with RAID and [93](#)
- system capacity
  - methods of calculating [11](#)
- system performance
  - impact of media scrubbing on [20](#)

## T

- terminology
  - family [125](#)
- third-party storage
  - verifying back-end configuration [59](#)
- timeouts
  - RAID option, considerations for changing [103](#)
- tips
  - for creating and backing up aggregates, for sensitive data [18](#)
- topologies
  - supported storage connection type [10](#)
- Twitter
  - how to receive automatic notification of documentation changes [167](#)

## U

- unmirrored aggregates
  - explained [108](#)
- unpartitioned drives
  - using to create aggregates [134](#)
- unprotected mode
  - returning SEDs to [84](#)
- unreachable key management servers
  - precautions taken in the event of during boot process [78](#)
- used space
  - how to determine and control in aggregates, by volume [127](#)
  - how to determine in aggregate [126](#)

## V

- V-Series functionality
  - name change to FlexArray Virtualization [53](#)
- V-Series systems
  - See* LUNs (array)
- verifying
  - back-end configuration [59](#)
  - key management server links [76](#)
- vetoos
  - of an aggregate relocation [157](#)
  - overriding [157](#)
- VMDK drives
  - how Data ONTAP reports disk types [9](#)
- volume command
  - for displaying space information [43](#)
- volume show-footprint command
  - understanding output [127](#)
- volumes
  - creating aggregates for FlexVol, using unpartitioned drives [134](#)
  - creating aggregates for Infinite, using unpartitioned drives [134](#)
  - determining which ones reside on an aggregate [161](#)
  - determining write-caching eligibility for Flash Pool aggregates [149](#)
  - how to determine space usage of, in aggregates [127](#)
  - sharing of aggregates [129](#)
- Vservers
  - See* SVMs

## W

- WAFL external cache
  - compared with Flash Pool aggregates [113](#)
- wear life
  - how Data ONTAP manages SSD wear [22](#)
- wildcard characters
  - how you use with disk ownership commands [51](#)
- wizards
  - running the Storage Encryption setup [72](#)
- write caching
  - determining Flash Pool aggregate eligibility [149](#)
  - determining FlexVol volume eligibility [149](#)

## **Z**

zoned checksum type

changing for array LUNs [63](#)