Technical Report

Infinite Volume Deployment and Implementation Guide

Sandra Moulton, NetApp
June 2013 | TR-4178

Abstract

This technical report focuses on NetApp® Infinite Volume implementations starting in clustered Data ONTAP® 8.2.

This report describes in detail how to implement and use Infinite Volume and provides information on best practices, operational considerations, and troubleshooting.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Current market trends for data storage indicate that file-based storage is the fastest-growing market segment in the storage industry. An explosion of unstructured content growth in the market for active deep archives and content repositories creates new challenges in the storage and retrieval of large volumes of data. NetApp's' solution for this problem is Infinite Volume, which provides a single large container that is optimized for scale and ease of management. Infinite Volume is built on NetApp's' proven clustered Data ONTAP solution.

Notes:

1. The same information applies to both FAS and V-Series systems, unless otherwise noted.
2. An Infinite Volume is a single large scalable file system that contains a collection of FlexVol® volumes called constituents.

   References to an Infinite Volume mean the logical container, not its individual constituents.
3. An Infinite Volume includes a namespace constituent and multiple data constituents. The namespace constituent contains the directory hierarchy and file names with pointer redirectors to the physical location of the data files. The data constituents contain the physical data in an Infinite Volume.

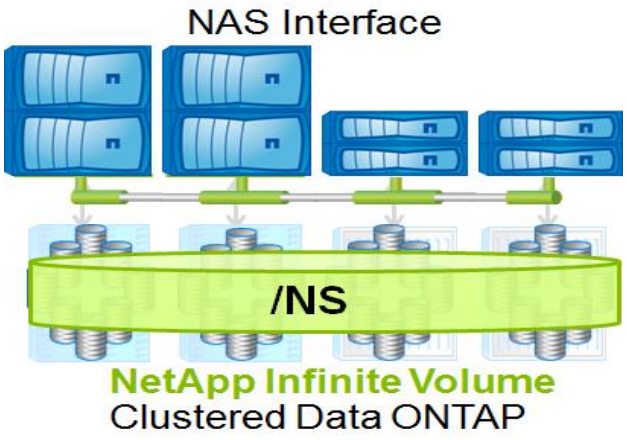
   References to either the namespace constituent or data constituents mean that specifically, not the entire Infinite Volume.
4. References to FlexVol volumes are specific to clustered Data ONTAP 8.2.
5. Whenever references are made to SVM they are referring to Storage Virtual Machine which is the logical container that Infinite Volume resides within.

2 Overview of NetApp Infinite Volume Overview

Part of the NetApp clustered Data ONTAP offering, Infinite Volume can store petabytes of data and billions of files in a single file system. Infinite Volume integrates with NetApp's proven technologies and products, such as deduplication, data compression, Snapshot™ copies, SnapMirror® replication, and nondisruptive operations. Infinite Volume is designed to address the needs of large unstructured repositories of primary data, which are also known as enterprise content repositories. Infinite Volume is focused on use cases that can be characterized by input/output (I/O) patterns in which data is written once and seldom changed. However, this data is used for normal business operations, and therefore content must be kept online for fast retrieval, rather than being moved to secondary storage.

Infinite Volume works by grouping storage resources across multiple aggregates and nodes into a single namespace and presenting a single file system to the client. It automatically distributes files across the nodes and aggregates within a cluster.

Figure 1) Infinite Volume high-level view.



2.1 Infinite Volume Components

An Infinite Volume is made of a group of constituents stitched together into a single volume. When an Infinite Volume is created, it automatically creates the following constituents distributed across nodes:

- Namespace constituent
- One or more namespace mirror constituents
- Data constituents

The namespace constituent contains directory and file names and pointer references to the physical location of the file in the Infinite Volume. It is also the junction path, which is the client-accessible namespace for the entire Infinite Volume. There is one namespace constituent per Infinite Volume, and by default, it is a maximum of 10TB.

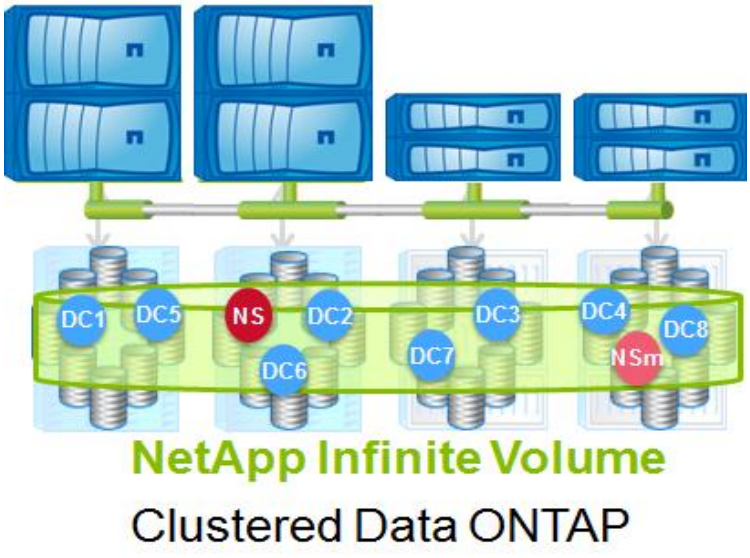
The namespace mirror constituent contains an asynchronous volume SnapMirror copy of the namespace constituent. It serves two main purposes: backup for the namespace constituent and enabling support for differential tape backup by using SnapDiff. There is one namespace mirror constituent for backup of the namespace constituent. It is replicated every 5 minutes, and is equal in size to the namespace constituent.

SnapDiff requires that each node contain either a namespace constituent or a namespace mirror constituent. Because one namespace constituent and one namespace mirror constituent are already created by default, adding SnapDiff involves creating additional namespace mirror constituents on each node that contains an Infinite Volume data constituent, but that doesn't contain a namespace constituent or a namespace mirror constituent. By default, the namespace constituent mirrors created for SnapDiff are replicated once a day, but this can be modified to a value larger than 1 hour. The namespace mirror constituents are equal in size to the namespace constituent.

The data constituents contain the data from files stored in the Infinite Volume. An entire file exists within a single data constituent. Data constituents are created on each node that has at least one aggregate assigned to the Infinite Volume. Upon Infinite Volume creation, equal amounts of usable data constituent space are created on each node that contains an aggregate assigned to the Storage Virtual Machine (SVM) for Infinite Volume. Data constituents can grow up to the maximum supported size for the model of system that contains it. For details, see NetApp Hardware Universe.

Figure 2 shows the underlying constituents of the Infinite Volume. The blue circles are data constituents, the red is the namespace constituent, and the pink is the namespace mirror constituent.

Figure 2) Infinite Volume constituent layout.



3 General Infinite Volume Features

Infinite Volume provides a single volume that can store large amounts of data within a single file system. Some of the main features include:

- Single large container
  - Single file system
  - Single junction path
  - Automatic load balancing of files during ingest
- Nondisruptive scalability; start small grow as needed
  - Up to 20PB
  - Up to 2 billion files
  - Scales across up to 10 nodes in a cluster
  - Automatic capacity balancing
- Provided by default with clustered Data ONTAP
  - Does not require any additional licenses
- Supports multi-tenancy
  - Allows multiple SVMs with FlexVol volumes and SVMs with Infinite Volume on the same unified scale-out storage cluster
- Supports storage efficiency
  - Thin provisioning
  - Deduplication
  - Data compression
- Offers data resiliency
  - NetApp RAID-DP®
  - Snapshot copies
  - Asynchronous volume SnapMirror
  - Differential tape backup (NetApp SnapDiff)

NetApp recommends that the number of Infinite Volumes in a cluster be less than or equal to the number of nodes in the cluster.

3.1 Infinite Volume Capacity Balancing

By default, Infinite Volume automatically balances capacity by distributing usable space equally across nodes. Further, when new files are written to an Infinite Volume, they are distributed evenly across data constituents mainly in a round-robin fashion, with preference for data constituents that have the most available space.

4 Infinite Volume Requirements

This section discusses the requirements and limitations of an Infinite Volume.

Table 1) Overview of Infinite Volume requirements.

Requirement	
Minimum Data ONTAP version	Clustered Data ONTAP 8.2
Junction path	One junction path per Infinite Volume. The junction path can be only one element; for example, `/acme`, not `/acme/bugs`. The NFS and CIFS clients can mount or map shares at or below this junction path.
License	Infinite Volume does not require any licenses.
Hardware	All 3000 and 6000 series FAS and V-Series systems that are supported with clustered Data ONTAP 8.2 can contain a combination of different models in the same Infinite Volume. For a list of platforms that support Infinite Volume, see NetApp Hardware Universe at support.netapp.com/knowledge/docs/hardware/NetApp/syscfg/index.shtml.
Storage Virtual Machine	Each Infinite Volume requires a dedicated SVM. A cluster can contain numerous SVMs with Infinite Volume and SVMs with FlexVol volumes that share resources.
Aggregate type supported	64-bit only
Maximum file size	16TB
Maximum number of files	2 billion
Maximum volume size	The maximum Infinite Volume size is 20PB on a 10-node FAS6290 cluster. This value is limited by the model of system and the size of the drives.
Maximum data constituent size	The maximum data constituent size is determined by the type of storage systems with aggregates assigned to the Vserver with Infinite Volume. For example, a FAS6290 has a maximum data constituent size of 100TB.
Maximum namespace constituent size	The maximum namespace constituent size is 10TB.
Maximum number of nodes in the cluster	Infinite Volume requires a minimum of 2 nodes and can span up to 10 nodes in a cluster; there is no limit on the number of nodes in a cluster.
Supported protocols	NFSv3, NFSv4.1, pNFS, SMB 1.0
Tape backup	NFS, SMB, or differential support (SnapDiff)

5 When to Use an Infinite Volume

Choosing when to use an Infinite Volume involves understanding the benefits as well as the recommended use cases. Infinite Volume is best suited for use cases that require a single large NAS volume with a single junction path that needs to grow beyond 100TB and store up to 2 billion files. It is best suited for unstructured data that is static in nature (primarily write once, with infrequent overwrites, appends, and deletes) and for data with an average file size of at least 100KB. It is not recommended for metadata-intensive workloads, transactional workloads that require low latency, or data that changes frequently.

Infinite Volume supports 2 billion files and has the same recommendations as FlexVol volumes for architecting high-file-count and deep-directory environments. For recommendations on how to best architect high-file-count and directory infrastructures, refer to TR-3537: High-File-Count Environment Best Practices.

Note: These are guidelines, not rules; your environment may have different requirements for specific use cases. NetApp highly recommends fully testing and understanding the performance impact of any new configuration before it is implemented in production.

6 Performance

Because Infinite Volume is part of Data ONTAP, it is tightly integrated with the NetApp WAFL® (Write Anywhere File Layout) file structure and is therefore optimized to perform with high efficiency. Infinite Volume is able to leverage the internal characteristics of Data ONTAP as well as the load-balancing capabilities of the Infinite Volume architecture.

However, the following factors can affect the performance of the Infinite Volume:

- The application and the type of dataset being used
- The type of access pattern—sequential or random access
- The size of the I/O
- The average file size
- Frequency of access on data files
- The number and type of storage systems the Infinite Volume spans
- The amount of load on the system
- Disk type ATA or SAS, and the RPM of the disk
- The number of disk spindles in the aggregates
- The number of constituents in the Infinite Volume
- Network speed

When considering adding Infinite Volume, be sure to use standard sizing and testing methods that would be used when considering the addition of applications to the storage system. It is important to understand how the new Infinite Volume and its workload will affect your systems and whether you have the bandwidth to accommodate an Infinite Volume and its use with acceptable impact on the applications currently running on your storage system.

6.1 I/O Performance of Infinite Volume

The Infinite Volume read and write performance is a function of the hardware platform that is being used, the size of the I/O, and the amount of load that is placed on the system. Infinite Volume is optimized for large files that are written once and may be read frequently from the same node.

The performance varies based on the factors previously mentioned and should be tested before implementing in production.

6.2 Flash Cache Cards and Flash Pool

The Infinite Volume constituents can benefit from NetApp Flash Cache™ or Flash Pool™ intelligent caching if its workload consists of large amounts of blocks that are read repeatedly. In that case, Flash Cache and Flash Pool can significantly reduce the number of reads to hard disks, thus improving the read performance. The amount of performance improvement from Flash Cache or Flash Pool depends on the number of shared blocks, the access rate, the active dataset size, and the data layout. Use the same tools that are used with standard FlexVol volumes to determine what the benefit will be with Infinite Volume constituents.

7 Scalability

An Infinite Volume can span up to 10 nodes in a cluster. The number of nodes contained in the Infinite Volume can increase its capacity and throughput capabilities. Although capacity can scale linearly, performance may not scale in a linear manner as the number of nodes increases. The amount of performance scalability is based on the type of access in the Infinite Volume. Reading hot data and writing new large files scales the best; small file access of random files, metadata, and directory access scales less.

It is important to make sure that neither your data constituents nor the Infinite Volume as a whole run out of space. As with FlexVol volumes, you can use NetApp OnCommand® Unified Manager to monitor your space usage and set thresholds for alerts at both the Infinite Volume level and the data constituent level. For details, see the section on "Resizing an Infinite Volume."

7.1 Increasing the Size of an Infinite Volume

One of the major benefits of an Infinite Volume is its ability to nondisruptively scale as needed. You can grow an Infinite Volume by simply resizing it, adding disks, adding aggregates, or adding nodes.

Note: The storage capacity of an Infinite Volume cannot be decreased.

8 Security

Infinite Volume uses a unique security style called the *unified security style,* which allows all users to view and set file permissions regardless of whether they are CIFS or NFS clients. Further, it allows the file permissions to include both Windows® and UNIX® users and groups, simplifying security management.

Note: Although the SVM root volume might have been created with a different security style, this does not affect the Infinite Volume.

Benefits of the unified security style include:

- When an NFS client sets UNIX permissions on a file that uses NTFS file permissions, the changes are merged into the NTFS file permissions.
- When a Windows client sets NTFS permissions on a file, the permissions can include UNIX principals.

For more information about unified security, see the File Access and Protocols Management Guide and TR-4067: Clustered Data ONTAP NFS Best Practice and Implemenation Guide.

9 Configuration and Operation

This section discusses how to configure an Infinite Volume and provides end-to-end examples of creating and resizing. Although the section discusses some basic things, it is assumed that the NetApp storage system is already installed and running and that you are familiar with basic NetApp administration.

9.1 Command Summary

The following sections describe the Infinite Volume commands. Most commands are the same as the FlexVol commands. Some commands can be used at the SVM level, some at the Infinite Volume level, and others can be used at the constituent level.

SVM for Infinite Volume Level Commands

These commands are used to create and manage a SVM for Infinite Volume. To view information at the Infinite Volume or data constituent level, see the commands in Tables 3 and 4.

Table 2) SVM for Infinite Volume level commands.

Command	Summary
vserver create –vserver <SVMname> -is-repository true	Creates a SVM that can contain an Infinite Volume. The option –is-repository true denotes the SVM as one that will contain an Infinite Volume; not using this option creates a SVM for FlexVol volumes.
vserver modify –vserver <SVMname> -aggr-list <aggrname>, <aggrname>, …	Assigns aggregates to be available for the SVM for Infinite Volume to use.
vserver show –vserver <SVMname>	Displays configuration information about the SVM for Infinite Volume, including the list of assigned aggregates, Snapshot policy, allowed protocols, and whether the SVM is one for FlexVol volumes or Infinite Volume.
volume efficiency policy create –vserver <SVMname> - policy <policyname> -schedule <cron job schedule> –duration <time interval> -enabled true -qos-policy <background | best-effort>	Creates a volume efficiency policy for a specific SVM. This policy can then be assigned to an Infinite Volume. The value for –schedule must correlate to an existing cron job schedule in the cluster. The value for –duration represents how many hours you want compression and deduplication to run before stopping. The option –qos-policy assigns a priority to the volume efficiency policy of either background or best-effort.

Infinite Volume Level Commands

These commands are used to create and manage an Infinite Volume. To view information at the data constituent level, see the commands in Table 4.

Table 3) Infinite Volume Level Commands.

Command	Summary
volume create –vserver <SVMname> -volume <InfiniteVolumename> -size <size in GB|TB|PB>	Creates an Infinite Volume.
volume create –vserver <SVMname> -volume <InfiniteVolumename> -size <size in GB|TB|PB> -namespace-aggregate <aggr_list>	Creates an Infinite Volume specifying a particular aggregate for the namespace constituent.

Command	Summary
volume create –vserver <SVMname> -volume <InfiniteVolumename> -size <size in GB|TB|PB> –data-aggr-list <aggr_list>	Creates an Infinite Volume specifying specific aggregates for the data constituents.
volume create –vserver <SVMname> -volume <InfiniteVolumename> -size <size in GB|TB|PB> -enable-snapdiff true	Creates an Infinite Volume with SnapDiff support enabled.
volume show –volume <InfiniteVolumename>	Displays high-level information about the Infinite Volume such as SVM, State, Type (RW or DP), Size, and Available Space.
volume show –vserver <SVMname> –volume <InfiniteVolumename>	Displays detailed information about the Infinite Volume.
volume modify –vserver <SVMname> –volume <InfiniteVolumename> -space-guarantee <none|volume>	Enables (none) or disables (volume) thin provisioning on the Infinite Volume's data constituents. The namespace constituent and namespace mirror constituents are always thick provisioned.
volume modify –vserver <SVMname> –volume <InfiniteVolumename> -size <size in GB|TB|PB>	Nondisruptively increases the size of the Infinite Volume.
volume efficiency on –vserver <SVMname> –volume <InfiniteVolumename>	Enables deduplication on the specified Infinite Volume.
volume efficiency modify –vserver <SVMname> –volume <InfiniteVolumename> –compression true	Enables postprocess compression of subsequently created data. This requires that deduplication first be enabled on the Infinite Volume.
volume efficiency modify –vserver <SVMname> –volume <InfiniteVolumename> – compression true –inline-compression true	Enables inline compression and postprocess of subsequently created data. This requires that deduplication first be enabled on the Infinite Volume. There is no way to enable inline compression without postprocess compression and deduplication.
volume efficiency modify –vserver <SVMname> –volume <InfiniteVolumename> -policy <policy_name>	Assigns or changes which compression/deduplication scheduling policy is assigned to the Infinite Volume.
volume efficiency show –vserver <SVMname> –volume <InfiniteVolumename>	Displays whether compression and deduplication are enabled and which efficiency policy is assigned to the Infinite Volume.
volume show –volume <InfiniteVolumename> -fields used, sis-space-saved, sis-space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent, compression-space-saved, compression-space-saved-percent	Shows aggregated space usage details for an Infinite Volume including space used, total space saved, percent total savings, deduplication and compression space, and percent savings.

Data Constituent Level Commands

These commands are used to view information at the data constituent level.

Table 4) Commands for compression and deduplication of existing data.

Command Data ONTAP 8.1	Summary
volume efficiency show –vserver <SVMname> -is-constituent true	Displays data constituent level information, including whether compression and deduplication are enabled, which efficiency policy is assigned, status, and progress details.
volume show –vserver <SVMname> -is-constituent true -fields used, sis-space-saved, sis-space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent, compression-space-saved, compression-space-saved-percent	Displays space used, total space saved, percent total savings, deduplication and compression space, and percent savings for each constituent in a specific Infinite Volume.

9.2 Interpreting Space Usage and Savings

The `volume show` command can show both space used and space saved per Infinite Volume or per constituent. When viewing space usage at the Infinite Volume level, use the `volume show` command with the `–volume <InfiniteVolumename>` option. The output is the sum of all space for all constituents, including the namespace constituent, namespace mirror constituents, and data constituents. When viewing space usage at the constituent level, use the `volume show` command, substituting `–is-constituent true` for the `–volume <InfiniteVolumename>` option. The data shown is for each individual constituent. Constituents are labeled as follows:

- Data constituents are identified by the volume name `<InfiniteVolumename>_1024_data####`
- Namespace constituent are identified as `<InfiniteVolumename>_ns`
- Namespace mirror constituents are identified as `<InfiniteVolumename>_mirror####`

Here is a summary of the fields that can be displayed with the `volume show` command.

Command

```
volume show -fields used, sis-space-saved, sis-space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent, compression-space-saved, compression-space-saved-percent
```

Table 5) Interpreting volume show savings values.

Field	Description
vserver	SVM name
volume	Infinite Volume or constituent name
used	Amount of physical space used on the volume
sis-space-saved	Number of bytes saved by both compression and deduplication
sis-space-saved-percent	Percentage space saved by both compression and deduplication (compression+deduplication bytes saved) / (used + compression bytes saved + deduplication bytes saved) * 100
dedupe-space-saved	Number of bytes saved by deduplication

Field	Description
dedupe-space-saved-percent	Percentage of space saved by deduplication dedupe bytes saved / (used + dedupe bytes saved) * 100
compression-space-saved	Number of bytes saved by compression
compression-space-saved-percent	Percentage of space saved by compression compression bytes saved / (used + compression bytes saved) * 100

To determine the logical space used, add the values “used + sis-space-saved.”

9.3 Quick Start

This section describes the steps necessary to create an Infinite Volume. It assumes you already have a cluster setup and aggregates of appropriate size on each of the nodes you want included in the Infinite Volume.

Table 6) Infinite Volume quick start.

Action	Command	Example
Create a SVM for Infinite Volume.	volume create –vserver <SVMname> -is-repository true -rootvolume <rootvolname> -aggregate <aggr that will contain the root volume> -ns-switch file -rootvolume-security-style mixed	vserver create -vserver vsIV1 -is-repository true -rootvolume root_iv1 -aggregate aggr1 -ns-switch file -rootvolume-security-style mixed
Assign aggregates to the Infinite Volume.	vserver modify - vserver <SVMname> –aggr-list <comma separate list of aggregates>	vserver modify –vserver vsIV1 -aggr-list agg1, agg2, agg3, aggr4
Create an Infinite Volume.	volume create –vserver <SVMname> –volume <InfiniteVolumename> -size < size in GB|TB|PB>	volume create -vserver vsIV1 -volume InfiniteVol1 -size 220TB
Create data LIFs for the SVM with Infinite Volume, allowing both NFS and CIFS.	Network interface create –vserver <SVMname> -lif <lifname> -role <type> -data-protocol nfs,cifs –home-node <nodename> -home-port <portnumber> -address <ipaddress> -netmask <netmask> -status admin up –failover-policy nextavail	network interface create -vserver vsIV1 -lif datalif1 -role data -data-protocol nfs,cifs -home-node fas6290c-svl01 -home-port e0a -address 10.1.1.60 -netmask 255.255.252.0 -status-admin up -failover-policy nextavail
Optional		
Configure NFS. Create a mount point and mount the Infinite Volume using /NS as the junction point through one of the data LIFs.	vserver nfs create -vserver <SVMname> -v3 enabled -v4.1 enabled -v4.1-pnfs enabled	vserver nfs create -vserver vsIV1 -v3 enabled -v4.1 enabled -v4.1-pnfs enabled

Action	Command	Example
Configure CIFS. Create a CIFS share. Map a network drive to \\<data lif ip address>\sharename; for example, \\10.1.1.60\InfiniteVol1	vserver services dns create -vserver <SVMname> -domains <domain.name> -name-servers <NIS ips> vserver cifs create -vserver <SVMname> -cifs-server <cifsservername> -domain <domain.name> vserver cifs share create -vserver <SVMname> -share-name <sharename> -path <junctionpath>	vserver services dns create -vserver vsIV1 -domains example.com -name-servers 10.1.1.100, 10.1.1.101 vserver cifs create -vserver vsIV1 -cifs-server CIFSIV1 -domain example.com vserver cifs share create -vserver vsIV1 -share-name InfiniteVol1 -path /NS
Enable deduplication	volume efficiency on -vserver <SVMname> -volume <InfiniteVolumename>	volume efficiency on -vserver vsIV1 -volume InfiniteVol1
Add compression	volume efficiency modify –vserver <SVMname> -volume <InfiniteVolumename> -compression true -inline-compression true	volume efficiency modify -vserver vsIV1 -volume InfiniteVol1 -compression true -inline-compression true
Create SnapMirror relationship: Assumes that SVM for Infinite Volume is already created (use the same steps as source)and that a cluster peer relationship exists. 1. Set up a SVM peer relationship. 2. Create a SnapMirror relationship. 3. Initialize SnapMirror. 4. Optionally you can mount the data protection Infinite Volume as read-only.	vserver peer create –vserver <source SVMname> -peer-vserver <destination SVMname> -applications snapmirror-peer-cluster <second cluster name> snapmirror create -destination-path <DPSVMname:DPvolumename>– sourcepath <SVMname:volumename> snapmirror initialize -destination-path <DPSVMname:DPvolumename> volume mount –vserver <DPSVMname> -volume <DPvolumename> -junction-path <junction-path>	vserver peer create –vserver vsIV1 -peer-vserver vsIV1DP -applications snapmirror-peer-cluster cluster2 snapmirror create -destination-path -vsIV1DP:InfiniteVol1DP –sourcepath vsIV1:InfiniteVol1 -type DP snapmirror initialize -destination-path vsIV1DP:InfiniteVol1DP volume mount –vserver vsIV1DP -volume InfiniteVol1DP -junction-path /nsDP
Enable SnapDiff - Increase Infinite Volume size by (namespace constituent size * (# nodes that contain data constituents – 1)) - Enable SnapDiff	volume modify -volume <InfiniteVolumename> -size +< size in GB|TB> volume modify -vserver <SVMname> -volume <InfiniteVolumename> –enable-snapdiff true	volume modify -volume InfiniteVol1DP -size +10TB volume modify -vserver vsIV1DP -volume InfiniteVol1DP –enable-snapdiff true

9.4 Creating an Infinite Volume

As mentioned earlier, when you create an Infinite Volume you must start with a clustered Data ONTAP 8.2 configuration and a SVM for Infinite Volume. NetApp recommends using the `vserver modify –aggr-list` command to assign aggregates to the SVM to control which systems the Infinite Volume will use. When you create the Infinite Volume, it automatically creates a namespace constituent, one or more namespace mirror constituents, and a number of data constituents. The size of Infinite Volume you create includes the space for the namespace constituent, the namespace mirror constituents, and the data constituents. NetApp recommends specifying a size when creating an Infinite Volume to avoid creating it at the minimum size.

The Infinite Volume algorithm for selecting which aggregates to use and the number and size of the data constituents is aimed at simplifying administration. It evaluates the available resources and automatically selects a configuration that provides optimal space balancing across nodes. Even though this is done automatically, it is useful for the administrator to understand how the algorithm works.

By default, the Infinite Volume creation process evaluates all available aggregates, either those that are explicitly assigned to the SVM, or all available if none are assigned. Each time the process creates a new constituent, it selects the aggregate that has the most free space available or that is the least overcommitted.

How the Namespace Constituent Is Created

The process starts by selecting the aggregate that has the most available space or that is the least overcommitted and creates a 10TB namespace constituent. You can specify which aggregate to use with volume create option -namespace-aggregate. The process creates only one namespace constituent per Infinite Volume.

NetApp recommends selecting the system that has the most memory, contains an aggregate with the fastest disks, and has the least amount of additional work for the namespace constituent. If there are numerous Infinite Volumes in your cluster, NetApp recommends no more than one namespace constituent per node.

How the Namespace Mirror Constituents are Created

After the Infinite Volume creation process creates the namespace constituent, it creates one or more namespace mirror constituents equal in size to the namespace constituent in the following way:

- Creates a 10TB namespace mirror for data protection on the aggregate that has the most available space or that is the least overcommitted on a node other than the one containing the namespace constituent. If the Infinite Volume spans more than two nodes, the namespace mirror constituent for data protection will not be created on the HA partner of the node containing the namespace constituent.
- Creates 10TB namespace mirror constituents for SnapDiff on the largest aggregates on each node that has aggregates assigned to the Infinite Volume, other than the ones that contain the namespace constituent or the namespace mirror constituent created for data protection.

How Data Constituents Are Created

The remaining size of the Infinite Volume is divided across all nodes that have aggregates assigned to the SVM with Infinite Volume and are used for data constituents. As with the namespace constituent and namespace mirror constituents each time the process creates a new constituent, it selects the aggregate that has the most free space or that is the least overcommitted. It creates constituents one node at a time.

- The first data constituent on the first node is created up to the maximum supported constituent size for the model of system on the aggregate that has the most free space or that is the least overcommitted.

- Subsequent data constituents on the first node are created in the same manner, up to the maximum supported constituent size for the model of system. Before each constituent is created, the aggregates on the node are evaluated to select the one that has the most free space or is the least overcommitted.
- This process continues until the first node contains its portion of data constituent space.
- The same process is followed for all nodes that have aggregates assigned to the SVM with Infinite Volume.

Note: If you use different models of systems that have different maximum constituent sizes in an Infinite Volume, the algorithm automatically creates data constituents across all nodes with a maximum size equal to the lower of the maximum supported data constituents size available across all nodes with aggregates assigned to the Infinite Volume.

For example, if you create an 800TB Infinite Volume that spans two FAS6290 and two FAS3270 systems, it would be configured as follows, assuming that the aggregates are large enough and that SnapDiff is not enabled:

FAS6290-node1: 10TB namespace constituent, 70TB data constituent1, 70TB data constituent2, 55TB data constituent3

FAS6290-node2: 70TB data constituent4, 70TB data constituent5, 55TB data constituent6

FAS3270-node3: 10TB namespace mirror constituent, 70TB data constituent7, 70TB data constituent8, 55TB data constituent9

FAS3270-node4: 70TB data constituent10, 70TB data constituent11, 55TB data constituent12

If you first created an Infinite Volume that spanned only nodes with the same maximum data constituent size and later added nodes that have a smaller maximum data constituent size, upon resizing the Infinite Volume that includes aggregates from the newer nodes, all resize operations adhere to the new lower maximum constituent size.

For example, if you have a four-node FAS6290 cluster with aggregates assigned to the Infinite Volume, the data constituents could be up to 100TB in size. If you later assign aggregates from a FAS3270 node, resize operations limit any constituents from growing beyond 70TB; data constituents already created beyond 70TB would continue to exist.

Options When Creating an Infinite Volume

You can control which aggregates are used for the namespace constituent and data constituents by using the `volume create` options `-namespace-aggregate` and `-data-aggr-list`. NetApp recommends selecting the node that has the most memory and the aggregate on that node with the fastest disks for the namespace constituent. If you are performing a lot of directory access, Flash Cache may be beneficial.

When creating an Infinite Volume, you can decide whether to enable SnapDiff. Enabling SnapDiff at Infinite Volume creation results in each node that contains a data constituent also containing a namespace constituent or namespace mirror constituent. For example, in a four-node cluster without SnapDiff, you would have one namespace constituent and one namespace mirror constituent. In the same four-node cluster with SnapDiff enabled, you would instead have one namespace constituent and three namespace mirror constituent.

In most situations the namespace constituent is 10TB, which allows the Infinite Volume to support 2 billion files. Exceptions include an Infinite Volume that is less than 80TB, or if `-max-namespace-constituent-size` is set. Setting `-max-namespace-constituent-size` limits the number of files and directories that can be written to the Infinite Volume. To determine the minimum size of namespace constituent you will need, use (max number of files desired)/2B * 10TB.

For example, for 1 billion files ((1 billion / 2 billion) * 10TB) you would need a total of 5TB. Remember that you need the same size for each of the namespace mirrors. If you set `max-namespace-constituent-size`, you can do a resize operation to increase the size up to the maximum of 10TB. However, you can never decrease the size.

You can control the maximum size of the data constituents that would be necessary if you are using different models of systems that have different maximum constituent sizes on the volume SnapMirror source and destination Infinite Volume.

For example, if you are using two FAS6290 that have a maximum constituent size of 100TB on the source and two FAS3270 systems that have a maximum constituent size of 70TB on the destination, you need to create the source Infinite Volume with `max-data-constituent-size` set to 70TB. To accomplish, this you would use the command `volume create –vserver <vservername> -volume <InfiniteVolumename> -size <GB|TB|PB> -max-data-constituent-size 70TB`. NetApp recommends determining whether this is necessary in the planning stage, before the initial creation of the Infinite Volume.

9.5 Resizing an Infinite Volume

You can resize an Infinite Volume if space is available in the aggregates assigned to the SVM. You can also add additional aggregates from existing nodes, or add aggregates from new nodes and then perform a resize operation. In this case the Infinite Volume is resized as follows.

- The total new data size of the Infinite Volume is equally divided across the nodes that have aggregates assigned to the SVM with Infinite Volume.
- The first resize operation involves adding one or more additional namespace mirror constituents, if aggregates from new nodes exist and SnapDiff is enabled.
- Next, for each node, the existing data constituents are resized up to the amount of space required, or the maximum data constituent size, whichever is less.
- If more space is required, each node creates additional data constituents. The first new data constituent is created on the aggregate that has the most available space. Subsequent data constituents are created in the same fashion for each node.

As in the previous example, if you have an 800TB Infinite Volume that spans two FAS6290 and two FAS3270 systems and you want to resize it to 1PB, it would be configured as follows, assuming that the aggregates are large enough to support and SnapDiff is not enabled.

FAS6290-node1: 10TB namespace constituent, 70TB data constituent1, 70TB data constituent2, 70TB data constituent3 (the 55TB data constituent3 was resized to 70TB), 35TB data constiutent13 (added)

FAS6290-node2: 70TB data constituent4, 70TB data constituent5, 70TB data constituent6 (the 55TB data constituent6 was resized to 70TB), 35TB data constiutent14 (added)

FAS3270-node3: 10TB namespace mirror constituent, 70TB data constituent7, 70TB data constituent8, 70TB data constituent9 (the 55TB data constituent9 was resized to 70TB), 35TB data constiutent15 (added)

FAS3270-node4: 70TB data constituent10, 70TB data constituent11, 70TB data constituent12 (the 55TB data constituent12 was resized to 70TB), 35TB data constiutent16 (added)

Like the Infinite Volume creation algorithm, the resize operation requires each node to have sufficient space to succeed. For example, if you have an Infinite Volume that spans four nodes and you are increasing the size by 400TB, each node must have 100TB free space for the resize to be successful.

NetApp highly recommends that free space be equal across all nodes that contain data constituents to allow equal space distribution and optimal balancing across nodes. If one node does not contain sufficient space, add disks to this node or move FlexVol volumes off this node onto another node.

Note: Always resize the SnapMirror destination of the Infinite Volume before resizing the source Infinite Volume.

9.6 Best Practice Recommendations for an Infinite Volume

NetApp recommends the following best practices:

- Limit the number of Infinite Volumes per cluster to no more than the number of nodes in your cluster.
- Use a junction path that is one element and specific to the specific Infinite Volume you are creating; for example use /videos or /logs. You can't use /videos/hires or /logs/2013. A directory is different from a junction path and can be created directly below the junction path or below other directories.
- Infinite Volume Size:
  - When creating an Infinite Volume, specify a size to avoid it being created at the minimum size.
  - Be sure to create the Infinite Volume with a size that includes the amount of usable space you require plus the space required for namespace constituent and one or more namespace mirror constituents. For example, if you want 200TB of space available for file data, be sure to create a 220TB Infinite Volume, assuming no SnapDiff or just two nodes.
- Aggregate specifics:
  - Assign aggregates from a maximum of 10 nodes to the SVM to control which systems will be used by the Infinite Volume by using the `vserver modify –aggr-list` command.
  - Assign aggregates from only as many nodes as you plan for the Infinite Volume to grow to need.
  - Do not assign aggregates from FAS2000 series systems.
  - Create aggregates that are at least as large as the largest supported data constituents plus 10% to reduce the potential of the aggregate running out of space and to enable the creation of the largest data constituents possible. For example, if the maximum data constituent size is 100 TB, aggregates should be near 110TB or 220TB, so that 100TB data constituents can fit on them.
- Namespace constituent specifics:
  - Select the system with the most memory that contains an aggregate with the fastest disks and the least amount of existing workload for the namespace constituent.

    If your cluster contains more than one Infinite Volume, use the `–namespace-aggregate` option with the `volume create` command to specify where the namespace constituent is created and will enable one namespace constituent per node. For example, if you have a four-node cluster and four Infinite Volumes, place each namespace constituent on a different node. The namespace constituent of Infinite Volume one could be on node1, the namespace constituent of Infinite Volume two could be on node2, and so on. This does not apply to the namespace mirror constituents.
  - Set `–max-namespace-constituent-size` only if you are sure that the Infinite Volume will never need to store 2 billion files. Not setting this option enables the aggregate to have sufficient space to accommodate any growth required if you ever need to support 2 billion files in the Infinite Volume.
- SnapMirror specifics:
  - When planning your Infinite Volume, first determine if there is a possibility of having models of systems that support smaller data constituents than the source. It is important to know this before creating the Infinite Volume to make sure that data constituents don't get created that are larger than the destination can support. If you need to limit the size of the source's data constituent, you can do so with the command `volume create –vserver <SVMname> -volume <InfiniteVolumename> -size <GB|TB|PB> -max-data-constituent-size <GB|TB|PB>`.
- Infinite Volume supports 2 billion files and has the same recommendations as FlexVol volumes for architecting high-file-count and deep directory environments..
- Use the same type of drives in the aggregates that you assign for data constituents.

9.7 End-to-End Infinite Volume Configuration Examples

This section gives examples of creating an Infinite Volume in clustered Data ONTAP 8.2. For details of each step, including additional options available, see the Logical Storage Management Guide.

The first example describes the process of creating a new Infinite Volume, including setting up and managing compression and deduplication on it.

The second example describes the process of creating a data protection mirror of the Infinite Volume on a second cluster. It also includes adding SnapDiff.

Note: The steps are spelled out in detail, so the process appears much longer than it would be in the real world.

Example One: Creating an Infinite Volume Called InfiniteVol1

This example creates a 200TB Infinite Volume called InfiniteVol1 in SVM vsIV1 spread across two nodes.

1. Create a SVM for Infinite Volume called vsIV1.

```
Cluster1::> vserver create -vserver vsIV1 -is-repository true -rootvolume
root_iv1 -aggregate aggr1 -ns-switch file -rootvolume-security-style mixed
[Job 76] Job succeeded:
Vserver creation completed
```

2. Assign aggregates to the SVM for Infinite Volume. Select aggregates from the nodes that you want to contain constituents for the Infinite Volume. NetApp recommends selecting aggregates with the same type of drives for the Infinite Volume. You can view your aggregates by using the aggr show command.

```
Cluster1::> aggr show
Aggregate            Availsize   Size            State     Nodes
-------------        ---------   ---------       --------  ------------
aggr0_node1            3.52TB      2.91TB        online    fas6290c-svl01
aggr0_ node2           3.52TB      2.91TB        online    fas6290c-svl02
aggr1                200.09TB    200.09TB        online    fas6290c-svl01
aggr2                110.09TB    110.09TB        online    fas6290c-svl01
aggr3                200.09TB    200.09TB        online    fas6290c-svl02
aggr4                110.09TB    110.09TB        online    fas6290c-svl02
   6 entries were displayed.

Cluster1::> vserver modify –vserver vsIV1 -aggr-list agg1, agg2, agg3,
aggr4
```

3. Create a 220TB Infinite Volume called InfiniteVol1.

```
Cluster1::> volume create -vserver vsIV1 -volume InfiniteVol1 -size 220TB
[Job 77] Job succeeded: Created Infinite Volume successfully.
[Job 78] Creating constituent "InfinteVol1_ns" on aggregate "aggr1"
[Job 78] Creating constituent "InfinteVol1_1024_data0001" on aggregate
"aggr2”
[Job 78] Creating constituent "InfinteVol1_1024_data0002" on aggregate
"aggr3”
[Job 78] Job succeeded: Created Infinite Volume successfully.
```

4. Create data LIFs for the Infinite Volume to allow both NFS and CIFS. Follow the best practices from the Data ONTAP Network Management Guide.

```
Cluster1::> network interface create -vserver vsIV1 -lif datalif1 -role
data -data-protocol nfs,cifs -home-node fas6290c-svl01 -home-port e0a -
address 10.1.1.60 -netmask 255.255.252.0 -status-admin up -failover-policy
nextavail
```

```
Cluster1::> network interface create -vserver vsIV1 -lif datalif2 -role
data -data-protocol nfs,cifs -home-node fas6290c-svl02 -home-port e0a -
address 10.1.1.61 -netmask 255.255.252.0 -status-admin up -failover-policy
nextavail
```

```
Cluster1::> network interface create -vserver vsIV1 -lif datalif3 -role
data -data-protocol nfs,cifs -home-node fas6290c-svl01 -home-port e0b -
address 10.1.1.62 -netmask 255.255.252.0 -status-admin up -failover-policy
nextavail
```

```
Cluster1::> network interface create -vserver vsIV1 -lif datalif4 -role
data -data-protocol nfs,cifs -home-node fas6290c-svl02 -home-port e0b -
address 10.1.1.63 -netmask 255.255.252.0 -status-admin up -failover-policy
nextavail
```

5. If required in your environment, enable NFS on the SVM with Infinite Volume.

```
Cluster1::> vserver nfs create -vserver vsIV1 -v3 enabled -v4.1 enabled -
v4.1-pnfs enabled
```

You can validate the configuration with the vserver nfs show command.

```
Cluster1::> vserver nfs show -vserver vsIV1 -fields v3,v4.1,v4.1-pnfs
vserver v3      v4.1    v4.1-pnfs
------- ------- ------- ---------
vsIV1   enabled enabled enabled
```

Check the security permissions using the volume show command and modify if necessary.

```
Cluster1::> volume show -vserver vsIV1 -volume InfiniteVol1 -fields unix-
permissions
vserver volume       unix-permissions
------- ------------ ----------------
vsIV1   InfiniteVol1 ---rwxrwxrwx
```

On the Linux® client, create a directory to use as a mount point and mount the Infinite Volume, using /ns as the junction point through one of the data LIFs.

6. If required in your environment, enable CIFS on the SVM with Infinite Volume.

```
Cluster1::> vserver services dns create -vserver vsIV1 -domains example.com
-name-servers 10.1.1.100, 10.1.1.101
```

```
Cluster1::> vserver cifs create -vserver vsIV1 -cifs-server CIFSIV1 -
domain example.com
```

Verify that the CIFS server is up and running with the vserver cifs show command.

```
Cluster1::> vserver cifs show -vserver vsIV1
```

```
Vserver: vsIV1
CIFS Server NetBIOS Name: CIFSIV1
NetBIOS Domain/Workgroup Name: EXAMPLE
Fully Qualified Domain Name: EXAMPLE.com
Default Site Used by LIFs Without Site Membership:
Authentication Style: domain
CIFS Server Administrative Status: up
```

Create a share on the Infinite Volume by using the vserver cifs share create command on the junction path of the Infinite Volume.

```
Cluster1::> vserver cifs share create -vserver vsIV1 -share-name
InfiniteVol1 -path /NS
```

Check the security permissions using the volume show command and modify if necessary.

```
Cluster1::> volume show -vserver vsIV1 -volume InfiniteVol1 -fields unix-
permissions
vserver volume       unix-permissions
------- ------------ ----------------
vsIV1   InfiniteVol1 ---rwxrwxrwx
```

On the Windows client, map a network drive to \\"data lif ip address"\ `sharename`; for example, \\`10.1.1.60\InfiniteVol1`.

7. (Optional) Enable deduplication on the Infinite Volume (`volume efficiency on`), followed by compression (`volume efficiency modify –compression true –inline-compression true`) (`-inline-compression` is required only if you want to use inline compression), and verify that it is turned on. The `volume efficiency show` command shows the compression and deduplication configuration for Infinite Volume.

```
Cluster1::> volume efficiency on -vserver vsIV1 -volume InfiniteVol1
[Job 6455] Job succeeded: Efficiency for volume "InfiniteVol1" of Vserver
"vsIV1" is enabled.
Already existing data could be processed by running "volume efficiency
start -vserver vsIV1 -volume InfiniteVol1 -scan-old-data true".

Cluster1::> volume efficiency modify -vserver vsIV1 -volume InfiniteVol1 -
compression true -inline-compression true
[Job 6456] Job succeeded: Modified efficiency settings for volume
"InfiniteVol1" in Vserver "vsIV1" successfully.
```

To validate the efficiency settings you can use the volume efficiency show command.

```
Cluster1::> volume efficiency show -vserver vsIV1 -volume InfiniteVol1 -
fields state,compression,inline-compression,schedule,policy

vserver volume      state   schedule policy compression inline-compression
------ ----------   ------- -------- ------ ----------- ------------------
vsIV1  InfiniteVol1 Enabled sun-sat@0  -       true          true
```

Assign a volume efficiency policy for the deduplication and compression operations. Start by making sure that there is an appropriate job schedule.

```
Cluster1::> job schedule cron show
Name                Description
----------------    ----------------------------------------------------
5min                @:00,:05,:10,:15,:20,:25,:30,:35,:40,:45,:50,:55
8hour               @2:15,10:15,18:15
daily               @0:10
hourly              @:05
weekly              Sun@0:15
5 entries were displayed.
```

Create a volume efficiency policy using one of these job schedules and assign it to the Infinite Volume. In this example we are using the schedule of daily.

```
Cluster1::> volume efficiency policy create -vserver vsIV1 -policy
daily2310 -schedule daily -enabled true -duration 8
```

Assign the volume efficiency policy to the Infinite Volume.

```
Cluster1::> volume efficiency modify -vserver vsIV1 -volume InfiniteVol1 -
policy daily2310
[Job 6458] Job succeeded: Modified efficiency settings for volume
"InfiniteVol1" in Vserver "vsIV1" successfully.
```

That's all there is to it.

Example Two: Adding a Data Protection Mirror for the Infinite Volume

This example adds a data protection mirror for the Infinite Volume. Although it is not necessary, the example includes adding SnapDiff to the destination volume. The example assumes that a cluster peer relationship already exists between the source and destination clusters.

1. Create a SVM for Infinite Volume called vsIV1DP.

```
Cluster2::> vserver create -vserver vsIV1DP -is-repository true -
rootvolume root_iv1dp -aggregate aggr1 -ns-switch file -rootvolume-
security-style mixed
[Job 149] Job succeeded:
Vserver creation completed
```

2. Identify the size of the source Infinite Volume and its constituents.

```
Cluster1::> volume show –volume InfiniteVol1 –fields size
vserver volume       size
------- ------------ -----
vsIV1   InfiniteVol1 220TB
```

3. Assign aggregates to the SVM for Infinite Volume. Select aggregates from the nodes that you want to contain constituents for the Infinite Volume. NetApp recommends selecting aggregates with the same type of drives for the Infinite Volume. You can view your aggregates by using the `aggr show` command.

```
Cluster2::> aggr show
Aggregate            Availsize   Size          State     Nodes
------------         ---------   -------       ------    ------------
aggr0_node3            3.52TB      2.91TB      online    fas6290c-svl03
aggr0_ node4           3.52TB      2.91TB      online    fas6290c-svl04
aggr1                200.09TB    200.09TB      online    fas6290c-svl03
aggr2                110.09TB    110.09TB      online    fas6290c-svl03
aggr3                200.09TB    200.09TB      online    fas6290c-svl04
aggr4                110.09TB    110.09TB      online    fas6290c-svl04
   6 entries were displayed.

Cluster2::> vserver modify –vserver vsIV1DP -aggr-list agg1, agg2, agg3,
aggr4
```

4. Create a SVM peer relationship between the source and destination cluster and SVM for Infinite Volume. Make sure that you already have a cluster peer relationship set up between the clusters.

```
Cluster2::> vserver peer create –vserver vsIV1DP –peer-vserver vsIV1 –
applications snapmirror –peer-cluster cluster1
Info: [Job 151] 'vserver peer create' job queued

Cluster1::> vserver peer accept -vserver vsIV1 -peer-vserver vsIV1DP
Info: [Job 150] 'vserver peer accept' job queued

Cluster2::> vserver peer show
            Peer        Peer
Vserver     Vserver     State
----------- ----------- ------------
vsIV1DP     vsIV1       peered
```

5. Create an Infinite Volume (in this example, called InfiniteVol1DP) equal to the source Infinite Volume size minus the size of the source namespace mirror constituent.

   In the case since the source Infinite Volume is 220TB and contains only one data constituent mirror that is 10TB in size, the destination Infinite Volume should be 210TB.

```
Cluster2::> volume create -vserver vsIV1DP -volume InfiniteVol1DP -size
210TB –type DP
[Job 153] Job succeeded: Created Infinite Volume successfully.
```

   This creates an Infinite Volume but does not create the constituents. You can verify with the volume show command as follows.

```
Cluster2::> volume show -is-constituent true
There are no entries matching your query.
```

6. Create a Snapmirror relationship between the source and destination Infinite Volumes.

```
Cluster2::> snapmirror create -source-path vsIV1:InfiniteVol1 -
destination-path vsIV1DP:InfiniteVol1DP -type DP
Operation succeeded: snapmirror create the relationship with destination
vsIV1DP:InfiniteVol1DP.
```

7. Initialize SnapMirror. This creates the constituents on the destination Infinite Volume.

```
Cluster2::> snapmirror initialize -destination-path vsIV1DP:InfiniteVol1DP
[Job 154] Job is queued: snapmirror initialize of destination
vsIV1DP:InfiniteVol1DP.
```

You can check the status of the SnapMirror initialization with the `snapmirror show` command.

```
Cluster2::> snapmirror show
                                                                    Progress
Source       Destination Mirror  Relationship   Total            Last
Path    Type Path        State        Status    Progress Healthy Updated
------- ---- --------- ------       -------   -------- ------- --------
vsIV1:InfiniteVol1
        DP   vsIV1DP:InfiniteVol1DP
                         Snapmirrored  Idle        -       true    -
```

The initialization created the constituents. You can verify with the volume show command.

```
Cluster2::> volume show–is-constituent true
Vserver  Volume   Aggregate    State   Type   Size  Available  Used%
-------- -------- ------------ ------- ----   ----- ---------- -----
vsIV1DP  InfiniteVol1DP_ns
                  aggr1        online  DP     10TB    9.50TB    5%
vsIV1DP  InfiniteVol1DP_1024_data0001
                  aggr2        online  DP     100TB  13.23TB   86%
vsIV1DP  InfiniteVol1DP_1024_data0002
                  aggr3        online  DP     100TB  23.28TB   76%
3 entries were displayed.
```

8. (Optional) Mount the data protection Infinite Volume as read-only.

```
Cluster2::> volume mount –vserver vsIV1DP –volume InfinteVol1DP –junction-
path /nsDP
```

9. (Optional) Enable SnapDiff on the SnapMirror destination.
   a) To enable SnapDiff on the SnapMirror destination you first need to increase the size of the destination Infinite Volume to accommodate the creation of one of more namespace mirror constituents. The new Infinite Volume size should reflect an increase equal to the number of nodes (minus one) times the size of the source namespace constituent.

To determine the size of the source namespace constituent use the `volume show` command with the `–is-constituent` option and look at the size of the namespace identified with the volume name ending in _ns.

```
Cluster1::> volume show -is-constituent true
Vserver Volume    Aggregate    State   Type Size  Available Used%
------  --------  ------------ ------- ---- ----- ---------- -----
vsIV1   InfiniteVol1_1024_data0001
                  aggr1        online   RW  100TB    13.23TB   86%
vsIV1   InfiniteVol1_1024_data0002
                  aggr2        online   RW  100TB    13.23TB   86%
vsIV1   InfiniteVol1_ns
                  aggr3        online   RW   10TB     9.50TB    5%
vsIV1   InfiniteVol1_ns_mirror0001
                  aggr4        online   DP   10TB     9.50TB    5%
4 entries were displayed.
```

b) Next increase the size of the destination Infinite Volume by the size of the source's namespace constituent multiplied by the number of nodes with data constituents. This calculation is based on SnapDiff requiring every node to contain either a namespace constituent or namespace mirror constituent. In this example it is 10TB since we only have two nodes and the first already contains the namespace constituent.

```
Cluster2::> volume modify -volume InfiniteVol1DP -size +10TB
Warning: Volume "vsIV1DP:InfiniteVol1DP" is a SnapMirror destination
volume. The File system Size for this volume is derived from its source
and cannot be changed. The specified size will be used as the Volume Size.
Do you want to continue? {y|n}: y
[Job 160] Job succeeded: Modified Infinite Volume successfully.
```

To verify the Infinite Volume increased in size you can use the volume show command.

```
labatt::> volume show -volume InfiniteVol1DP
Vserver   Volume       Aggregate State  Type   Size  Available  Used%
--------- ------------ --------- ------ ----   ----- ---------- -----
vsIV1DP   InfiniteVol1DP
                       -         online  DP    220TB  45.45TB    79%
```

c) Now you can enable SnapDiff on the destination Infinite Volume. This will create the namespace mirror constituent(s) that are required on each node that contains a data constituent other than the node that contains the namespace constituent.

```
Cluster2::> volume modify -vserver vsIV1DP -volume InfiniteVol1DP -enable-
snapdiff true
[Job 161] Job succeeded: Modified Infinite Volume successfully.
```

To verify the namespace mirror constituent(s) got created you can use the volume show command and look for constituents that end in _ns_mirror00##.

```
Cluster2::> volume show -is-constituent true
Vserver   Volume       Aggregate  State    Type   Size  Available Used%
--------- ------------ ---------- -------- ------ ----- --------- -----
vsIV1DP   InfiniteVol1DP_1024_data0001
                       aggr1_labatt2_SATA
                                  online    DP    100TB    13.23TB   86%
vsIV1DP   InfiniteVol1DP_1024_data0002
                       aggr1_labatt1_SATA
                                  online    DP    100TB    13.23TB   86%
vsIV1DP   InfiniteVol1DP_ns
                       aggr1_labatt2_SATA
                                  online    DP     10TB     9.50TB    5%
vsIV1DP   InfiniteVol1DP_ns_mirror0001
                       aggr1_labatt1_SATA
                                  online    DP     10TB     9.50TB    5%
4 entries were displayed.
```

That's all there is to it.

10 Upgrading and Reverting

Major and minor nondisruptive upgrades to clustered Data ONTAP are supported with both compression and deduplication. For details, see Nondisruptive upgrades in the "Other NetApp Features" section.

10.1 Upgrading to a Newer Version of Clustered Data ONTAP

Upgrading to a newer version of clustered Data ONTAP is a nondisruptive operation regardless of whether or not the systems contain FlexVol volumes or Infinite Volumes.

If your Infinite Volume did not contain a namespace mirror constituent, should add it after the upgrade. To do this you need to have an aggregate on a node, other than the node that contains the namespace constituent which contains at least as much space as the size of the namespace constituent, typically 10TB. If this is not the case, you will have to add disks or aggregates to meet this requirement. Then you can resize the Infinite Volume to add 10TB, using the command `volume modify –vserver <SVMname> -volume <InfiniteVolumename> -size +10TB`. This results in the automatic creation of the namespace mirror constituent.

If your Infinite Volume did contain a namespace mirror constituent that was created before clustered Data ONTAP 8.2, you must contact Customer Success Services for assistance in converting to the new format.

To see the status of the upgrade, issue the command `system node upgrade-revert show` in advanced mode. If there is a problem with the upgrade, you can look at the /mroot/etc/mlog/mgwd.log file and search for messages tagged as CR::RepositoryUpgrade. When the issue is resolved, you can reissue the command `system node upgrade-revert upgrade` in advanced mode to complete the upgrade processing. For more details on upgrading see the Upgrade and Revert/Downgrade Guide for Clustered Data ONTAP 8.2.

10.2 Reverting to a Newer Version of Clustered Data ONTAP

When you're considering reverting a system that contains an Infinite Volume to a version of clustered Data ONTAP earlier than 8.2, contact NetApp Customer Success Services for assistance.

11 Infinite Volume with Other NetApp Features

NetApp has several products that work with Infinite Volume. This section discusses how Infinite Volume works with other NetApp products and features. It gives details about support, limitations, and best practices.

11.1 Data Protection

Snapshot Copies

Infinite Volume supports crash-consistent snapshot copies at the Infinite Volume level, not at the individual data constituent level. Similarly to the way an administrator uses a FlexVol volume, an administrator can create, delete, and restore data files by using Snapshot on the Infinite Volume along with similar (hourly, nightly, weekly) Snapshot schedule management. A Snapshot copy of an Infinite Volume shows the files for the entire namespace at the Infinite Volume level. Snapshot copies can be used to recover the entire Infinite Volume to a point in time; they cannot be used to recover individual files or specific constituents. Infinite Volume does not support Snapshot rename or auto snapshot delete.

By default, the Snapshot schedule is set to default-1weekly. This policy sets up Snapshot copies on a schedule of creation every hour, retaining a maximum of six hourly, two daily, and one weekly copies. Older Snapshot copies are automatically deleted to free up space. You can modify the schedule, or you can add more schedules to the Snapshot policy, but the copies cannot be scheduled more frequently than hourly.

```
cluster1::> snapshot policy show default-1weekly
Vserver: vsIV1
                 Number of Is
Policy Name      Schedules Enabled  Comment
---------------- --------- -------  ----------------------------------------
default-1weekly      3       true    Default policy with 6 hourly, 2 daily & 1
weekly schedule.

Schedule    Count         Prefix       SnapMirror Label
--------    -----         -------      ----------------
hourly        6           hourly             -
daily         2           daily              -
weekly        1           weekly             -
```

The maximum number of Snapshot copies for an Infinite Volume is slightly fewer than the 255 supported on a FlexVol volume. You need to subtract 2 for each namespace mirror constituent and another 2 if you have a SnapMirror relationship between Infinite Volumes.

SnapRestore

SnapRestore® functionality is supported with Infinite Volume, and it works on the Infinite Volume as a whole, the same way as with FlexVol volumes. Single file Snapshot restore is not supported.

Volume SnapMirror

Volume SnapMirror allows you to replicate data to another location for disaster recovery purposes. Infinite Volumes support asynchronous volume SnapMirror at both the Infinite Volume level and at the namespace constituent level.

Namespace Constituent Level Volume SnapMirror

There are two types of volume SnapMirror copies of the namespace constituent; both are automatically created upon Infinite Volume creation and both are intracluster. The namespace mirror constituents do not require a SnapMirror license and cannot be managed with the SnapMirror commands.

The first type is the namespace mirror constituent that will be used for data protection of the namespace constituent. This namespace mirror constituent is automatically created on a node other than the one that contains the namespace constituent. If the Infinite Volume spans more than two nodes, the namespace mirror constituent is created on a node other than the HA partner of the node that contains the namespace constituent. This relationship is set up to replicate every 5 minutes, and this schedule cannot be modified. In the event of a failure of the source namespace constituent, contact NetApp Customer Success Services for assistance in failing over to the namespace mirror.

The second type of volume SnapMirror copy of the namespace constituent is created only if the Infinite Volume spans more than two nodes and SnapDiff is enabled. It is created upon SnapDiff enablement or on an Infinite Volume resize, if aggregates from new nodes are added and SnapDiff is enabled. It is set up on a daily replication schedule or as configured for SnapDiff.

Infinite Volume Level Volume SnapMirror

Volume SnapMirror copies of the entire Infinite Volume are supported from one cluster to another; you cannot replicate the Infinite Volume within the same cluster. Replication transfers data between the namespace constituent and data constituents on the source to the destination; namespace mirror constituents are not replicated as part of the Infinite Volume replication. This relationship can be set up to replicate hourly or less frequently; it can't be scheduled more frequently than once per hour.

Although you can have different models and a different number of systems on the destination, NetApp recommends keeping these items similar. Having different model of systems may require that you create the source with a –max-data-constituent-size smaller than would otherwise be created in order to accommodate successful transfers to the destination. This can limit the maximum size to which the Infinite Volume can grow. Also, if there are fewer nodes, the transfer time may be affected if the maximum transfer limit of 100 active transfers per node is reached.

Users have read-only access to the active file system on the destination Infinite Volume via the NFSv3 protocol. Users can also use CIFS and NFS to gain read-only access to Snapshot copies on the destination Infinite Volume. The active file system on the destination volume is an exported Snapshot copy of the active file system from the source volume. Replication and recovery are managed at the Infinite Volume level, not at the individual constituent level. If you need to fail over to the destination Infinite Volume, follow the same procedures as you would for a FlexVol volume. In the event of a failover, if SnapDiff is not enabled on the destination Infinite Volume, it does not contain a namespace mirror constituent. NetApp recommends creating the namespace mirror constituent after promoting the destination to read/write by resizing the Infinite Volume to +10TB which causes the namespace mirror constituent to be automatically created. This is not required if SnapDiff is enabled on the destination, because SnapDiff would have already created all of the namespace mirror constituents.

Setting up a SnapMirror relationship between a FlexVol volume and an Infinite Volume is not supported.

Table 7) Supported SnapMirror configurations with Infinite Volume.

SnapMirror Relationship	Supported or Not
Infinite Volume1 in clusterA to Infinite Volume2 in clusterB	Supported. Infinite Volume supports replication of an Infinite Volume as a whole to a destination cluster (intercluster only).
Infinite Volume1 in clusterA to Infinite Volume2 in clusterB and Infinite Volume3 in clusterB to Infinite Volume4 in clusterA	Supported. Infinite Volume supports bidirectional data exchange between two sites
Infinite Volume1 in clusterA to Infinite Volume2 in clusterB and also Infinite Volume1 in clusterA to Infinite Volume3 in clusterC	Supported. Infinite Volume supports multiple-mirror fanout deployments.
Infinite Volume1 in clusterA to Infinite Volume2 in clusterB and Infinite Volume2 in clusterB to Infinite Volume3 in clusterC	Not supported. Infinite Volume does not support SnapMirror cascading.
Data constituent to data constituent	Not supported. Infinite Volume replication is managed at the Infinite Volume level as a single relationship. You can't set up individual relationships between data constituents.
FlexVol volume to Infinite Volume	Not supported.
Infinite Volume to FlexVol volume	Not supported.

As with FlexVol volumes, in the event of a resize operation, perform the resize on the destination before the source to enable the successful continuation of updates.

Load Sharing Mirrors

Infinite Volume does not support load sharing mirrors.

SnapVault

SnapVault® functionality is not supported with Infinite Volume.

SnapDiff

SnapDiff is an application programming interface that interfaces with third-party applications to support differential tape back of an Infinite Volume. The applications can retrieve from SnapDiff a list of new, deleted, and changed files since the last tape backup and use the list to back up only those files, which can result in significantly faster backups to tape.

SnapDiff can be enabled on an Infinite Volume at any time, including Infinite Volume creation. Enabling SnapDiff requires sufficient space for a namespace mirror constituent on each node with an aggregate assigned to the SVM for Infinite Volume that does not already contain a namespace constituent or namespace mirror constituent.

Although you can back up from the SnapMirror destination, restoration from SnapDiff is allowed only to the source Infinite Volume.

For more information on SnapDiff support for Infinite Volume, refer to the Data ONTAP 8.2 Data Protection Tape Backup and Recovery Guide.

HA Pair Controller Configuration

HA pair controller configurations are supported with Infinite Volume. Both nodes in an HA pair should have the same version of Data ONTAP, except during an upgrade, when one node might be running temporarily an earlier version.

NetApp recommends that no active postprocess compression and deduplication operations take place during the planned takeover or giveback.

1. Perform the planned takeover or giveback during a time when compression and deduplication operations are not scheduled to run.
2. Determine whether any compression and deduplication operations are active and stop them until the planned takeover or giveback is complete.

There is no way to see the status or progress at the Infinite Volume level. You can use the `volume efficiency show –is-constituent true` command to see whether the status or progress of compression and deduplication on individual data constituents is Active or Idle. If compression and deduplication is running, then the status is Active.

On a system with compression and/or deduplication enabled, the output of the `volume efficiency show –is-constituent true` command is similar to the following.

```
cluster::> volume efficiency show -is-constituent true
Vserver     Volume                  State       Status Progress Policy
----------  ----------------------  --------    ------ -------- ----------
vs0         repo_vol_1024_data0001  Enabled     Active 42756 KB Scanned -
vs0         repo_vol_1024_data0002  Enabled     Active 27712 KB Scanned -
vs0         repo_vol_1024_data0003  Enabled     Active 37890 KB Searched -

...
```

You can use the `volume efficiency stop` command to abort the active compression/deduplication operations on the volume and the `volume efficiency start` command to restart it. The `volume efficiency stop` command creates a checkpoint and, when takeover or giveback completes, you can restart the deduplication/compression operations from the latest checkpoint with the `volume efficiency start` command. For additional information about active-active controller configurations, refer to TR-3450: Active-Active Controller Configuration Overview and Best Practice Guidelines.

11.2 Other NetApp Features

Flash Cache Cards

Flash Cache™ cards can be used to reduce the number of random disk reads of blocks that are read repeatedly. Read performance from Infinite Volume constituents can benefit from the Flash Cache if the workload consists of a lot of blocks that are read repeatedly. The performance improvement of Infinite Volumes with Flash Cache depends on the duplication rate, the access rate, the active dataset size, and the data layout. Use the same tools as with standard FlexVol volumes to determine what the benefit will be with Infinite Volume constituents.

Deduplication savings on the data constituents also enable efficient use of Flash Cache, by retaining the deduplication savings on Flash Cache that exist on disk. In that way, if you have 32k duplicate blocks on disk, after you run deduplication only one block will be used on disk; and if it is randomly accessed, only one block will be used in Flash Cache as well. This can significantly increase the amount of data that can be stored in Flash Cache.

For additional information about the Flash Cache card, refer to TR-3832: Flash Cache Best Practice Guide.

Flash Pool

Flash Pool™ intelligent caching allows flash technology in the form of solid-state disks (SSDs) and traditional hard disk drives (HDDs) to be combined to form a single Data ONTAP aggregate. When SSD and HDD technologies are combined in a Data ONTAP aggregate, the NetApp storage system takes advantage of the latency and throughput benefits of the SSD while maintaining the mass storage capacity of HDD.

The constituents of an Infinite Volume can benefit from Flash Pool if its workload consists of high numbers of blocks that are read repeatedly, by reducing the number of HDD reads, thus improving performance. The amount of performance improvement with Flash Pool depends on the number of shared blocks, the access rate, the active dataset size, and the data layout. Infinite Volume is not targeted for workloads with a large number of frequent small overwrites of data, and it does not typically benefit from the overwrite cache capabilities of Flash Pool. Use the same tools as are used with standard FlexVol volumes to determine what the benefit will be with Infinite Volume constituents.

Deduplication savings on the data constituents also make efficient use of Flash Pool, by retaining the deduplication savings on SSD that exist on HDD. In that way, if you have 32k duplicate blocks on disk, after you run deduplication, only one block is used on disk (HDD), and if any requests for duplicate blocks are randomly requested, only one block is used in the Flash Pool aggregate (SSD) as well. This can significantly increase the amount of data that can be stored in the Flash Pool aggregate. For more information on Flash Pool, refer to TR-4070: NetApp Flash Pool Design and Implementation Guide.

Deduplication

Infinite Volumes support deduplication and volume efficiency priority options. Deduplication is configured at the Infinite Volume level but savings are within a data constituent. Deduplication does not run on the namespace constituent or namespace mirrors.

Data Compression

Infinite Volumes support data compression and volume efficiency priority options. Compression is configured at the Infinite Volume level and is performed the same way on a data constituent as on a FlexVol volume. Compression does not run on the namespace constituent or namespace mirrors.

Thin Provisioning

Infinite Volume supports thin provisioning for data constituents, by allowing the administrator to set a volume guarantee of None for thin provisioning or Volume for thick provisioning.

Setting a volume guarantee of None offers better space utilization of the storage resources within the cluster. It does this by allowing space within the aggregate to be consumed as needed, as well as allowing you to create volumes larger than the amount of physical space in the system. Autosnapshot delete and autogrow are not supported with Infinite Volume.

Setting a volume guarantee of Volume preallocates blocks from the aggregate to the data constituent, preventing sharing of disk space between constituents or FlexVol volumes traditionally resulting in lower space utilization. For more information on thin provisioning, refer to TR-3965: Thin Provisioning Deployment and Implementation Guide.

FlexCache

Infinite Volume does not support FlexCache® intelligent caching.

FlexClone Volumes and Files

Infinite Volume does not support FlexClone® volumes or files.

32-Bit Aggregate Support

Infinite Volume is only supported with 64-bit aggregates. If you want to use an aggregate that is currently a 32-bit aggregate, you must upgrade the aggregate to a 64-bit aggregate. For information on upgrading an aggregate, refer to TR-3978: In-Place Expansion of 32-Bit Aggregates to 64-Bit Overview and Best Practices.

64-Bit Aggregate Support

Infinite Volume is supported with 64-bit aggregates. For information on 64-bit support, refer to TR-3978: In-Place Expansion of 32-Bit Aggregates to 64-Bit Overview and Best Practices.

Aggregate Relocate

Infinite Volume supports aggregate relocate, and it works the same way as relocating ownership of an aggregate that contains FlexVol volumes, with one exception: if you are relocating ownership of an aggregate that contains the namespace constituent or a namespace mirror constituent, and you are using SnapDiff. SnapDiff requires each node to contain either a namespace constituent or a namespace mirror constituent. To avoid an aggregate relocate failing if you are using SnapDiff, follow these procedures:

1) Make sure that all nodes that contain data constituents have a namespace constituent or namespace mirror constituent. If the aggregate that you want to relocate contains only data constituents, and the node to which you are relocating ownership doesn't already contain a namespace constituent or namespace mirror constituent, you must first have the Infinite Volume create a namespace mirror constituent on the new node. In this case, you should add an aggregate from the node to which you are planning to relocate ownership of the aggregate to the aggr-list, and then resize the Infinite Volume by the size of the namespace constituent.

   Example: 4-node cluster, Infinite Volume contains the following:

   Node1: aggr1 → namespace constituent, DC1
   aggr2 → DC4, DC7
   Node2: aggr3 → namespace mirror1, DC2
   aggr4→ DC5, DC8
   Node3: aggr5 → namespace mirror2, DC3
   aggr6 → DC6, DC9

   Not in Infinite Volume:

   Node4 contains aggr7 but doesn't contain any Infinite Volume constituents

   To relocate ownership of aggr6 from node3 to its HA partner node4, you would first make sure that node4 has at least as much space available on an aggregate as the size of the namespace constituent, in this example aggr7. Then you would add aggr7 to the SVM's aggr-list and resize the Infinite Volume by using the following commands. (Set the +size to be at least the size of the namespace constituent.)

```
vserver modify –vserver <vserver_name> –aggr-list aggr1, aggr2, aggr3,
aggr4, aggr5, aggr6, aggr7

volume modify –vserver <vserver_name> -volume <volume_name> –size +10TB
```

This automatically creates a namespace mirror constituent on node4, thus allowing a successful aggregate relocate of aggr6 to node4.

2) Relocate ownership of all aggregates that contain Infinite Volume data constituents only before relocating ownership of the aggregates with the namespace constituent or namespace mirror constituent. After all of the aggregates that contain no namespace constituent or namespace mirror constituent have been relocated, then you can relocate ownership of the aggregate that contains the namespace constituent or namespace mirror constituent. There is no need to move an aggregate that contains only a namespace mirror constituent and no data constituents.

   Example. 4-node cluster, Infinite Volume contains the following:

   Node1: aggr1 → namespace constituent, DC1
   aggr2 → DC5, DC9
   Node2: aggr3 → namespace mirror1, DC2
   aggr4→ DC6, DC10
   Node3: aggr5 → namespace mirror2, DC3
   aggr6 → DC7, DC11
   Node4: aggr7 → namespace mirror2, DC4
   aggr8 → DC8, DC12

   If you want to relocate ownership of aggr5 and aggr6 from node3 to its HA partner node4, you would have to relocate ownership of aggr6 first before relocating ownership of aggr7. After the aggregate relocates complete, consider contacting Customer Success Services for assistance in removing the extra namespace mirror constituent on node4.

3) If you want to relocate ownership of the aggregate that contains the namespace constituent or namespace mirror constituent, and you want to keep ownership of the other aggregates that contain data constituents on the source system, you should contact Customer Success Services for assistance in creating a namespace mirror constituent on the source node before attempting the aggregate relocate. In the previous example above, would be the case if you wanted to relocate ownership of agg5 from node3 to node4 but you wanted to keep ownership of aggr6 with node3.

   Example. 4 node cluster, Infinite Volume contains the following:

   Node1: aggr1 → namespace constituent, DC1
   aggr2 → DC5, DC9
   Node2: aggr3 → namespace mirror1, DC2
   aggr4→ DC6, DC10
   Node3: aggr5 → namespace mirror2, DC3
   aggr6 → DC7, DC11
   Node4: aggr7 → namespace mirror2, DC4
   aggr8 → DC8, DC12

   If you want to relocate ownership of aggr5 to node4 but you want node3 to retain ownership of agg6 this procedure would fail. You should contact Customer Success Services for assistance in creating a namespace mirror constituent on node3 so the aggregate relocate can succeed.

Note: NetApp does not recommend moving an aggregate that contains a namespace or namespace mirror constituent to one that already contains one of these. Doing so would result in a lower level of high availability, because the source and destination mirror would both reside on the same node. If you need to move the namespace or namespace mirror constituent to avoid this situation, contact Customer Success Services for assistance.

DUMP

Infinite Volume does not support backup to tape via NDMP. Backup to tape must be done via a CIFS or NFS share and can use NetApp SnapDiff for differential tape backup.

LUNs

Infinite Volume does not support LUNs.

Nondisruptive Upgrades

Both major and minor nondisruptive upgrades are supported with Infinite Volume.

For details and requirements for performing a nondisruptive upgrade on your system, refer to Upgrade Advisor in the AutoSupport™ tool if you have AutoSupport enabled; otherwise, refer to the release notes for the version of Data ONTAP to which you are upgrading.

DataMotion for Volumes

DataMotion™ software for volumes supports nondisruptive vol move operations within an Infinite Volume. It can be performed only at the constituent level and requires support from Customer Success Services.

12 Troubleshooting

This section discusses basic troubleshooting methods and common considerations when working with compression and deduplication.

12.1 Infinite Volume Shows Sufficient Space But File Write Fails

When looking at the Infinite Volume space, the user sees the combined space of the namespace constituent, one or more namespace mirror constituents, and the data constituents. A file gets written in its entirety to a single data constituent. Even if the Infinite Volume shows available space, there are a couple of rare conditions that could cause a file write to fail with an insufficient space message.

One reason could be that you are writing a new file and the data constituent or it's aggregate does not have sufficient space for the new file.

A second reason could be that you are appending to an existing file. If the data constituent that contains the file is full or nearly full, it may not have sufficient space to update the file. If other data constituents have sufficient space, a workaround for this situation could be to save the appended version of the file with a new filename and then delete the original file.

12.2 Can't Change Space Guarantee (Disable Thin Provisioning)

Setting a space guarantee of *Volume* configures data constituents to thick provisioned. Thick provisioning involves preallocating space from the aggregate for the full size of the constituents, regardless of the amount of data in the constituent. For example, assume that a constituent of 100TB is created in an aggregate that has 200TB of available space. With a volume space guarantee of *Volume*, the aggregate would immediately show only 100TB of available space, regardless of whether any data has been written to the constituent. If 20TB was written to the constituent, the aggregate would continue to show 100TB of available space.

Setting a space guarantee of *None* configures data constituents to be thin provisioned. Thin provisioning does not preallocate any space in the aggregate. Space is used in the aggregate only as needed based on the data written to the constituent. For example, assume that a constituent of 100TB is created in an aggregate that has 200TB of available space. With a volume space guarantee of *None*, the aggregate

would continue to show 200TB of available space. If 20TB was written to the constituent, the aggregate would show 180TB of available space.

When attempting to change the space guarantee of an Infinite Volume from *None* to *Volume,* it first checks whether there is sufficient space on all the aggregates that contain data constituents. If one or more aggregates that contain data constituents has insufficient space, the space guarantee change fails for all data constituents. To resolve this issue, identify which aggregates have insufficient space and add disks or move volumes off the aggregate to make sufficient space.

Note: Infinite Volume does not support decreasing the volume size.

Example:

To see the space guarantee on the Infinite Volume, use the `volume show` command.

```
cluster::> volume show -vserver vserverIV -volume InfVol -fields space-
guarantee

vserver   volume space-guarantee

--------- ------ ---------------

vserverIV InfVol none
```

To find out if it's possible to change the guarantee from *None* to *Volume*, follow these steps.

1. Look at the size of each data constituent and use the `volume show` command to find the aggregate that contains it.

```
cluster::> volume show -vserver vserverIV -is-constituent true –fields
aggregate,size,used

Vserver   Volume                      Aggregate   Size    Used

-------   ------------------------    ----------  ----    ---------

vserverIV InfVol_1024_data0001        aggr1       40TB    9.80TB

vserverIV InfVol_1024_data0002        aggr4       40TB    9.80TB

vserverIV InfVol_1024_data0003        aggr9       40TB    9.83TB

vserverIV InfVol_1024_data0004        aggr2       40TB    9.88TB

vserverIV InfVol_1024_data0005        aggr5       40TB    9.88TB

vserverIV InfVol_ns                   aggr3       10TB    1.50TB

vserverIV InfVol_ns_mirror0001        aggr6       10TB    1.50TB

7 entries were displayed.
```

This example shows five 40TB thin provisioned data constituents labeled InfVol_1024_data0001-5 on five different aggregates, aggr1, aggr4, aggr9, aggr2, aggr5.

Note: It's not necessary to look at the namespace constituent or namespace mirror constituents because these are always thick provisioned.

2. Look at the available space on each aggregate that contains a data constituent to see how much available space they have.

```
aggr show

cluster::> aggr show aggr1,aggr4,aggr9,aggr2,aggr5 –fields size,availsize

Aggregate  Size     Availsize
---------  -------  -----
aggr1      10.09TB  9.80TB
aggr2      10.09TB  9.88TB
aggr4      10.09TB  9.80TB
aggr5      10.09TB  9.88TB
aggr9       1.46TB  1.23TB
5 entries were displayed.
```

To determine whether there is sufficient space in the aggregates to change the space guarantee to *None,* you can use a table similar to the following.

Table 8) How to determine the amount of free space required to change the space guarantee to None.

Aggregate	Data constituent size	Data constituent space used	Required free space in aggregate (data constituent size – data constituent space used)	Aggregate size	Aggregate space available for data constituents (aggregate size * 0.9)	Sufficient available space (aggregate space available for data constituents – required free space in aggregate)
agg1	40TB	9.80TB	30.2TB	10.09TB	9.08TB	No
agg2	40TB	9.80TB	30.2TB	10.09TB	9.08TB	No
agg4	40TB	9.83TB	30.17TB	10.09TB	9.08TB	No
agg5	40TB	9.88TB	30.12TB	10.09TB	9.08TB	No
aggr9	40TB	9.88TB	30.12TB	10.09TB	9.08TB	No

First you need to calculate how much additional space setting the space guarantee to *Volume* would take up in the aggregates that contain data constituents. As shown in Table 8, this is done by subtracting the size of the data constituent from the amount of space used by the data constituent.

Next, you need to calculate how much space is available for data constituents within the aggregate. To avoid aggregates running into problems with the aggregate filling up, and to enable the maintainance of optimal performance, Infinite Volume uses a maximum of 90% of the space in the aggregate. Therefore, to calculate how much space is available for data constituents you multiply the aggregate size by 0.9.

If the amount of required free space in the aggregate is less than or equal to the aggregate space available for data constituents, then the space guarantee change to *Volume* is successful. If not, you need to add capacity to the aggregates or move volumes to other aggregates to make space available.

In this example, because these aggregates are not at maximum capacity, it would be possible to add disks to be able to change the space guarantee to *Volume*.

12.3 Lower-Than-Expected Space Savings

If you do not see the expected savings when using deduplication or compression, consider the following possibilities.

Snap Reserve Space Is Exceeded

Once the snapshot reserve space is full, Snapshot copies begin to use space in the active file system. The `df –S` command reports savings by dividing "saved" by "used + saved" reported in the active file system only. The savings value does not take into account space used by Snapshot copies, regardless of whether they include savings. This means that if Snapshot copies start taking space from the active file system, this space is counted as used space in the active file system and therefore decreases the value reported as %saved in the `volume show` output. Consider the following example:

1. Create a new volume with the snapshot reserve set to 0, populate it with data, and take a Snapshot copy.

```
Cluster1::> volume show -volume InfVol -fields percent-snapshot-space
vserver     volume      percent-snapshot-space
-------     ----------  ----------------------
vsone       InfVol      0%

Cluster1::> snap show –vserver vsone -volume InfVol
                                                  ----Blocks---
vserver  Volume  Snapshot                State   Size  Total%  Used%
-------  ------- ---------------------- -----   ----- ------  -----
vsone    InfVol   daily.2013-03-15_0010 valid   220KB   0%     3%


Cluster1::> volume show –volume InfVol -fields used, sis-space-saved, sis-
space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent,
compression-space-saved, compression-space-saved-percent

Filesystem    used      total-saved  %total-saved  deduplicated %deduplicated
compressed    %compressed     Vserver
/vol/InfVol/  185584732      0             0%            0            0%
0             0%              vsone
Cluster1::> df -h test
File system           total  used   avail capacity Mounted on         Vserver
/vol/InfVol/          500GB  176GB  323GB   35%   /vol/InfVol/          vsone
/vol/InfVol/.snapshot   0KB 5884KB 5318GB  ---%   /vol/InfVol/.snapshot vsone
```

2. Run compression against the existing data with the option to compress data that is already locked in a Snapshot copy. (This must be run in advanced mode.)

```
Cluster1::> set advanced
Warning: These advanced commands are potentially dangerous; use them only
when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y

Cluster1::*> volume efficiency start –volume InfVol – scan-old-data true –
compression true -shared-blocks true
```

3. After compression completes, use `volume show` to see the amount of Snapshot copy space used and the savings results.

```
Cluster1::> volume show –volume InfVol -fields used, sis-space-saved, sis-
space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent,
compression-space-saved, compression-space-saved-percent
File system   used      total-saved  %total-saved  deduplicated
%deduplicated  compressed    %compressed   Vserver
/vol/InfVol/  264527140 93001804          26%           0
0%             93001804           26%      vsone

Cluster1::> df -h InfVol
File system           total    used  avail  capacity  Mounted on
/vol/InfVol/          500GB   252GB  247GB    50%     /vol/InfVol/
/vol/InfVol/.snapshot   0KB   163GB 5243GB   ---%     /vol/InfVol/.snapshot
```

4. Change the snapshot reserve space so that it is large enough to contain the full size of the Snapshot copy. Use `volume show` to see the savings results.

```
Cluster1::> volume show -volume InfVol -fields snapshot-space-used
vserver           volume      snapshot-space-used
-------           ----------  -------------------
vsone             InfVol      40%

Cluster1::> volume show –volume InfVol -fields used, sis-space-saved, sis-
space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent,
compression-space-saved, compression-space-saved-percent
File system     used       total-saved    %total-saved    deduplicated
%deduplicated    compressed    %compressed      Vserver
/vol/InfVol/    92598792    93001804          50%              0
0%               93001804          50%          vsone

Cluster1::> df -h InfVol
File system            total   used   avail  capacity   Mounted on
/vol/ InfVol/          300GB    88GB  111GB     44%     /vol/test/
/vol/ InfVol/.snapshot 200GB   163GB   36GB     82%     /vol/test/.snapshot
```

Savings are calculated by dividing “used space” by “used + savings:”

Snapshot size: 163*1024*1024 = 170917888 Used: 264527140 Saved: 93001804

saved / (used + saved)) = Savings
93001804 / (264527140+93001804) = 26%

To calculate the actual savings in the active file system (space not including Snapshot copies), subtract the amount of space used by the Snapshot copies that exceeds the snapshot reserve space from the value listed as “used” in the calculation.

Active File system Space: 93609252 [used - snapshot size (264527140 – 170917888)]
saved / (used + saved) = Savings
93001804 / (93609252+93001804) = 50%

This 50% savings represents the actual savings realized when the Snapshot copies that contain the original uncompressed blocks are deleted or expire.

Snapshot Copies

Snapshot copies lock blocks in place by design, so that they are available for data recovery. This locking mechanism does not allow blocks that are freed by deduplication or rewritten with compression to be returned to the free pool until the locks expire or are deleted. If you are using deduplication or compression with Snapshot copies, consider the following:

- If possible, run compression and deduplication to completion before taking a Snapshot copy.
- Use the `snap show` command to see what Snapshot copies exist and the `snap delete` command to remove them. Alternatively, wait for the Snapshot copies to expire and the space savings to appear.
- By default, the initial compression of existing data skips blocks that are locked in Snapshot copies. If the existing data on disk contains a lot of data that is locked by Snapshot copies, running compression with the `-scan-old-data` without the `-snapshot-blocks true` option may not show much savings.

For more information about using compression and deduplication with Snapshot copies, see TR-3966: Data Compression and Deduplication Deployment and Implementation Guide for clustered Data ONTAP.

There May Not Be Enough Space for Deduplication to Run

The aggregate should have 3% of the total data's worth of free space for all deduplicated FlexVol volumes or data constituents, and each data constituent should have 4% of the total data's worth of free space. For details about the overhead associated with the deduplication metadata files, see TR-3966: Data Compression and Deduplication Deployment and Implementation Guide for clustered Data ONTAP.

The Number of Data Constituents May Be Affecting the Deduplication Savings

Deduplication works by removing duplicate blocks, storing only unique blocks within a data constituent; deduplication does not work on the namespace constituent or namespace mirror constituents. Deduplication is managed at the Infinite Volume level; however, it works at the data constituent level. Deduplication removes only duplicate blocks within the same data constituent. Duplicate blocks in different data constituents are not freed by deduplication. To achieve the best savings, make sure that the Infinite Volume creates the minimum required number of constituents, rather than creating more smaller constituents. You can do this by making sure that the aggregates assigned to the Vserver for Infinite Volume have sufficient available space to allow for the largest supported constituent size.

There May Be Little Duplicate or Compressible Data in the Volume

Run the Space Savings Estimation Tool (SSET) against the dataset to get an idea of the amount of duplicate and compressible data in the dataset. Although the compression savings estimate should be accurate to within 5%, the deduplication estimate may not be as accurate. This is because the SSET tool assumes that all data is in the same data constituent, which for an Infinite Volume will not be the case.

12.4 Latency Increases For a Couple of Minutes Every Hour

Infinite Volume latency can be affected while Snapshot copies are being created. The amount of impact varies, depending on how busy the system is and how many nodes are being used by the Infinite Volume. By default, the Snapshot schedule is default-1weekly. If necessary, you can change the default to have Snapshot copies taken when the system is less busy or when the Infinite Volume is less sensitive to latency. To see the Snapshot policy, use the command `volume show -vserver <SVMname> -volume <InfiniteVolumename> -fields snapshot-policy`. To see the available Snapshot policies, use the command `snap policy show;` or use `snap policy create` to create a new policy. To reduce the potential impact, configure the Infinite Volume to span only the number of nodes required.

12.5 SnapMirror Initialization or Updates Failing

Infinite Volume uses volume SnapMirror between clusters to provide disaster recovery for the Infinite Volume. The destination must be capable of creating a namespace constituent of the same size as well as the same number and size of data constituents as the source. The following conditions could cause the SnapMirror initialization or update to fail:

- The destination is on systems that don't support the size of the data constituent on the source. If you plan to use a model system that supports a smaller data constituent size on the destination, you must use the –max-data-constituent-size parameter on the source Infinite Volume.
- You resized the source Infinite Volume without first resizing the destination Infinite Volume.
- The destination Infinite Volume doesn't have sufficient space for the increased constituent size or additional constituents that will be created the first time an update occurs after the source size has been increased. Resizing the destination Infinite Volume does not actually increase the size of the constituents or add any additional constituents until after the next update occurs. If insufficient space exists to support the additional space, even though the resize appears to have succeeded, the SnapMirror update fails.
- Suppose that you have an Infinite Volume that spans at least three nodes; you have created identically sized Infinite Volumes on source and destination; and you have enabled SnapDiff on the destination only. Enabling SnapDiff requires each node in the Infinite Volume to have either a namespace constituent or a namespace mirror constituent. The source has one namespace constituent and one namespace mirror constituent. Because the Infinite Volume spans more than two nodes and SnapDiff is enabled, the destination Infinite Volume requires more namespace mirror constituents than the source. Since the size of the Infinite Volume is the sum of data constituents, the namespace constituent, and all namespace constituent mirrors, you must create the destination larger than the source.

  Assuming that SnapDiff is not enabled on the source, there will be just one namespace mirror constituent on the source. If SnapDiff is enabled on the destination, the number of namespace mirror constituents on the destination is the number of nodes – 1 minus 1, because that node contains the namespace constituent and therefore does not require a namespace mirror constituent.

  To figure out the proper size for the destination Infinite Volume, take the size of the source Infinite Volume + ("namespace constituent size" * "number of nodes -2"). You subtract two because the source Infinite Volume size already includes the namespace constituent and one namespace mirror constituent. For example, assume that you have an Infinite Volume that spans four nodes, the source Infinite Volume is 500TB, and the namespace constituent is 10TB. The destination Infinite Volume should be 520TB—500TB source (which includes the namespace constituent and one namespace mirror constituent) plus two additional namespace mirror constituents.

12.6 Attempt to Enable SnapDiff Failing

SnapDiff requires a namespace mirror constituent on each node that contains a data constituent other than the ones that already have a namespace constituent or namespace mirror constituent. If insufficient space exists on the aggregates assigned to the SVM for each of these nodes, enabling SnapDiff fails.

12.7 Slower-Than-Expected Performance

The information in this section assumes that basic proof-of-concept testing has been performed before running in production to understand what performance to expect from the Infinite Volume on the NetApp system. If unexpectedly slow performance is an issue, NetApp highly recommends contacting NetApp Customer Success Services for expert troubleshooting. In many cases, there are other factors such as misconfigured applications or conflicting policies that can be easily fixed to regain acceptable performance. See the section on, "Where to Get More Help," for contact information and data collection guidance.

Unexpectedly Slow Write Performance Caused by Adding Compression or Deduplication

If write performance appears to be degraded, check the NetApp system resources (CPU, memory, and I/O) to determine that they are not saturated. If resources are saturated, you can consider stopping some operations such as compression and deduplication to see if performance resumes. Stopping compression and deduplication operations generates a checkpoint, and these operations can be resumed when the system is less busy.

Also, write performance varies based on different platforms. Therefore it is not typically a good approach to compare write performance results across different NetApp platforms. Write performance can also be affected by using slower disk drives; comparing SAS drives to SATA drives can give different results.

If slow write performance continues to be an issue, NetApp highly recommends contacting NetApp Customer Success Services for expert troubleshooting. In many cases there are other factors such as misconfigured applications or conflicting policies that can be easily fixed to regain acceptable performance. See the "Where to Go for More Help," section for contact information and data collecting guidance.

Unexpectedly Slow Read Performance

Factors such as deduplication and compression savings, file size, frequency of reading a file, file location in the Infinite Volume, and type of access can affect read performance. To offset this effect, NetApp offers intelligent caching and pNFS. Intelligent caching provides the ability to cache frequently accessed blocks, which can result in a performance boost for random read requests. Intelligent caching applies to Flash Cache and Flash Pools. See the "Feature Interoperability" section for information about Infinite Volume with Flash Cache cards and Flash Pool. Using pNFS can also help improve performance of files that are read repeatedly, because access is sent directly to the node that contains the file data being requested. Sequential reads of large files accessed repeatedly via pNFS typically have the least performance impact.

12.8 Space Not Being Used as Expected

It is important to understand how the Infinite Volume allocates space to predict how different creation and resize operations operate. This section describes some common examples that may lead to unexpected behavior if you aren't aware of how constituents are created and aggregates are selected.

Infinite Volume Creation Failing

The Infinite Volume automatically creates space equally across all nodes with aggregates assigned to the SVM list; by default, this is all aggregates in the cluster. If one or more of the nodes that have aggregates assigned to the SVM do not have sufficient space for their portion of the Infinite Volume, the creation fails. For example, if your cluster has eight nodes, you do not specify an aggr-list and you try to create an 820TB Infinite Volume, each node in the cluster would require 100TB worth of available space, and two nodes would require an additional 10TB each for the namespace constituent and namespace mirror constituent (assuming that SnapDiff is not enabled). To avoid this situation, assign aggregates only from nodes with sufficient space to the SVM for Infinite Volume.

Resize Failing

Infinite Volume configures capacity equally balanced across all nodes in an Infinite Volume for all resize operations. This means that the resize operation divides the size of the increased space equally among all nodes that have an aggregate assigned to the Infinite Volume. If one or more of these nodes has insufficient space, the resize operation fails, even if other nodes in the Infinite Volume have extra space. For example, if you have an Infinite Volume that contains aggregates from four nodes and you increase the size by 200TB, each node must have 50TB of free space for the resize to be successful.

Added a Node But Infinite Volume Space Did Not Increase

Simply adding aggregates from new nodes to the aggr-list of the SVM with Infinite Volume does not increase the size of the Infinite Volume; you must also perform a resize operation. If you add an aggregate from a new node to the SVM list and do a resize, the usable capacity may not change if you are increasing the Infinite Volume by only 10TB and are using SnapDiff. That's because the first thing the resize operation does is to create a namespace mirror constituent on the new node that is equal in size to the namespace constituent, which is typically 10TB. If that is the size of the resize operation, then no additional space is created in the data constituents.

Added More Aggregates from Existing Nodes But Infinite Volume Did Not Use These Upon a Resize

Simply adding aggregates to the aggr-list of the SVM with Infinite Volume does not guarantee that a resize operation on the Infinite Volume uses these aggregates. Resize operations first increase the size of the existing constituents up to the maximum space possible before creating additional data constituents. If the size increase can be met without adding more constituents, then the new aggregates are not used. If the Infinite Volume resize does require new data constituents to be created, it selects aggregates on each node that have the most available space or are the least overcommitted. If the new aggregates added to the aggr-list do not meet this requirement as well as another aggregate on the node, it may not be used.

12.9 Location of logs and Error Messages

The location of the Infinite volume log files is:

```
/mroot/etc/log/mlog
```

12.10 Where to Get More Help

For additional support, contact one of the following.

- Your local account team
- Systems engineer
- Account manager
- NetApp Customer Success Services
- NetApp Support site
- 888 4 NETAPP (United States and Canada)
- 00 800 44 NETAPP (EMEA)
- +800 800 80 800 (Asia/Pacific)

Useful Information to Gather Before Contacting NetApp Customer Success Services

The following compression and deduplication commands and logs provide useful information for troubleshooting the root cause of compression and deduplication issues. This information is very useful when working with NetApp Customer Success Services.

- node show
- vserver show –vserver <SVMname> -fields aggr-list
- aggr show
- volume show –vserver <SVMname> -is-constituent *
- snapshot show –vserver <SVMname> -is-constituent *
- snapmirror show –vserver <SVMname> -is-constituent *
- event log show
- All mgwd logs located in /mroot/etc/log/
- /mroot/etc/log/messages
- Copy of the EMS logs for the time when the issue is seen

Additional References

- TR-4037: Introduction to NetApp Infinite Volume
  http://www.netapp.com/us/media/tr-4037.pdf
- Data ONTAP 8.2 Logical Storage Management Guide
  https://library.netapp.com/ecm/ecm_get_file/ECMP1196906
- Clustered Data ONTAP 8.2 Data Protection Guide
  https://library.netapp.com/ecm/ecm_get_file/ECMP1196819
- Clustered Data ONTAP 8.2 Documentation
  http://support.netapp.com/documentation/docweb/index.html?productID=61651
- Data ONTAP 8.2 File Access and Protocols Management Guide
  https://library.netapp.com/ecm/ecm_get_file/ECMP1196891
- TR-4067: Clustered Data ONTAP NFS Best Practice and Implemenation Guide
  http://www.netapp.com/us/media/tr-4067.pdf
- TR-4070: NetApp Flash Pool Design and Implementation Guide
- http://www.netapp.com/us/media/tr-4070.pdf
- TR-3978: In-Place Expansion of 32-Bit Aggregates to 64-Bit Overview and Best Practices
  http://www.netapp.com/us/media/tr-3978.pdf
- TR-3965: NetApp Thin Provisioning Deployment and Implementation Guide
  http://www.netapp.com/us/media/tr-3965.pdf
- TR-3450: High-Availability Pair Controller Configuration Overview and Best Practices
  http://www.netapp.com/us/media/tr-3450.pdf
- TR-4015: SnapMirror Configuration and Best Practices Guide for Clustered Data ONTAP 8.2
  http://www.netapp.com/us/media/tr-4015.pdf

Version History

Version	Date	Document Version History
Version 1.0	June 2013	Initial release for Infinite Volume with clustered Data ONTAP 8.2.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

NetApp
www.netapp.com