

**NetApp Solution Deployment Guidelines**

**Thomson Reuters – cDOT Replication**

**Synopsis:** This document details the various types of data replication technologies that are used within NetApp clustered Data ONTAP (cDOT) at Thomson Reuters.

**Authors:** Michael Arndt (arndt@netapp.com)

**Contributors:** Mitchell Vallone, Ken Zola, Ian Daniel, Riley Johnson

**Document Version:** V4

**Date:** May 2014

**CONFIDENTIAL INFORMATION**

This document contains information proprietary to Thomson Reuters and may not be reproduced, disclosed or used in whole or part without express permission of Thomson Reuters.

© Thomson Reuters 2013

Contents

[1 Introduction 4](#_Toc387242065)

[1.1 Management Summary 4](#_Toc387242066)

[1.2 Assumptions 4](#_Toc387242067)

[1.3 Change History 4](#_Toc387242068)

[1.4 Distribution List 5](#_Toc387242069)

[1.5 Glossary 5](#_Toc387242070)

[2 Requirements and Limitations 6](#_Toc387242071)

[2.1 Requirements for Intercluster replication 6](#_Toc387242072)

[2.2 Cluster peers and Vserver peers 6](#_Toc387242073)

[2.3 Limitations related to replication 6](#_Toc387242074)

[3 Migration from 7 Mode to cDOT 7](#_Toc387242075)

[3.1 Migration using the CLI 7](#_Toc387242076)

[3.2 Migration with the 7MTT 8](#_Toc387242077)

[3.3 Using WFA for migrations 8](#_Toc387242078)

[3.4 Host based migrations 8](#_Toc387242079)

[4 SnapMirror and SnapVault in cDOT 9](#_Toc387242080)

[4.1 SnapMirror and SnapVault commands in cDOT 9](#_Toc387242081)

[4.2 Similarities to 7 Mode 9](#_Toc387242082)

[4.3 Differences from 7 Mode 9](#_Toc387242083)

[4.4 Deduplication and compression with SnapVault 10](#_Toc387242084)

[4.5 Naming conventions at TR 10](#_Toc387242085)

[5 Non-Disruptive Volume Migration in cDOT 11](#_Toc387242086)

[5.1 Moving volumes 11](#_Toc387242087)

[5.2 Moving LIFs 11](#_Toc387242088)

[6 Clustered ONTAP CLI examples 12](#_Toc387242089)

[6.1 Intercluster LIFs and peering 12](#_Toc387242090)

[6.1.1 Create Intercluster LIF with default route 12](#_Toc387242091)

[6.1.2 Create cDOT to cDOT cluster peer (on source or destination) 12](#_Toc387242092)

[6.1.3 Create cDOT to cDOT Vserver peer (on destination) 12](#_Toc387242093)

[6.1.4 Accept cDOT to cDOT Vserver peer (on source) 12](#_Toc387242094)

[6.1.5 Create 7 Mode to cDOT Vserver transition peer (on destination) 12](#_Toc387242095)

[6.1.6 Update cluster peer addresses if nodes are added to a cluster 12](#_Toc387242096)

[6.2 Create cDOT destination volume for all replication types (on destination) 13](#_Toc387242097)

[6.3 SnapMirror transition from 7 Mode to cDOT 13](#_Toc387242098)

[6.3.1 Update snapmirror.access on 7 Mode system 13](#_Toc387242099)

[6.3.2 Create SnapMirror relationship 13](#_Toc387242100)

[6.3.3 Update, quiesce, and break relationship (on destination) 13](#_Toc387242101)

[6.3.4 Mount a volume in the namespace 14](#_Toc387242102)

[6.3.5 Modify volume snapshot policy 14](#_Toc387242103)

[6.3.6 Modify volume QoS policy 14](#_Toc387242104)

[6.3.7 Enable snap autodelete 14](#_Toc387242105)

[6.3.8 Configure qtree quotas 14](#_Toc387242106)

[6.3.9 Enable deduplication on a volume and assign efficiency policy 14](#_Toc387242107)

[6.3.10 Disable filesys-size-fixed 15](#_Toc387242108)

[6.3.11 Remove TDP snapmirror relationships 15](#_Toc387242109)

[6.3.12 Remove vserver transition peer relationship 15](#_Toc387242110)

[6.4 Intercluster SnapMirror from cDOT to cDOT 16](#_Toc387242111)

[6.4.1 Create job schedule (on destination) 16](#_Toc387242112)

[6.4.2 Create SnapMirror relationship (on destination) 16](#_Toc387242113)

[6.4.3 List SnapMirror relationships (on source) 16](#_Toc387242114)

[6.4.4 Update, quiesce, and break relationship (on destination) 16](#_Toc387242115)

[6.5 Intercluster SnapVault from cDOT to cDOT 17](#_Toc387242116)

[6.5.1 Create job schedule and SnapShot policy (on source) 17](#_Toc387242117)

[6.5.2 Create job schedule and SnapMirror policy (on destination) 17](#_Toc387242118)

[6.5.3 Create SnapMirror XDP relationship (on destination) 17](#_Toc387242119)

[6.5.4 Update, quiesce, and break relationship (on destination) 17](#_Toc387242120)

[6.5.5 List SnapMirror relationships (on source) 17](#_Toc387242121)

[6.6 Deduplication and compression on secondary volumes 18](#_Toc387242122)

[6.6.1 Enabling deduplication on a secondary volume 18](#_Toc387242123)

[6.6.2 Enabling compression on a secondary volume 18](#_Toc387242124)

[6.7 Non-disruptive volume movement in cDOT 18](#_Toc387242125)

[6.7.1 Volume move and cutover all at once 18](#_Toc387242126)

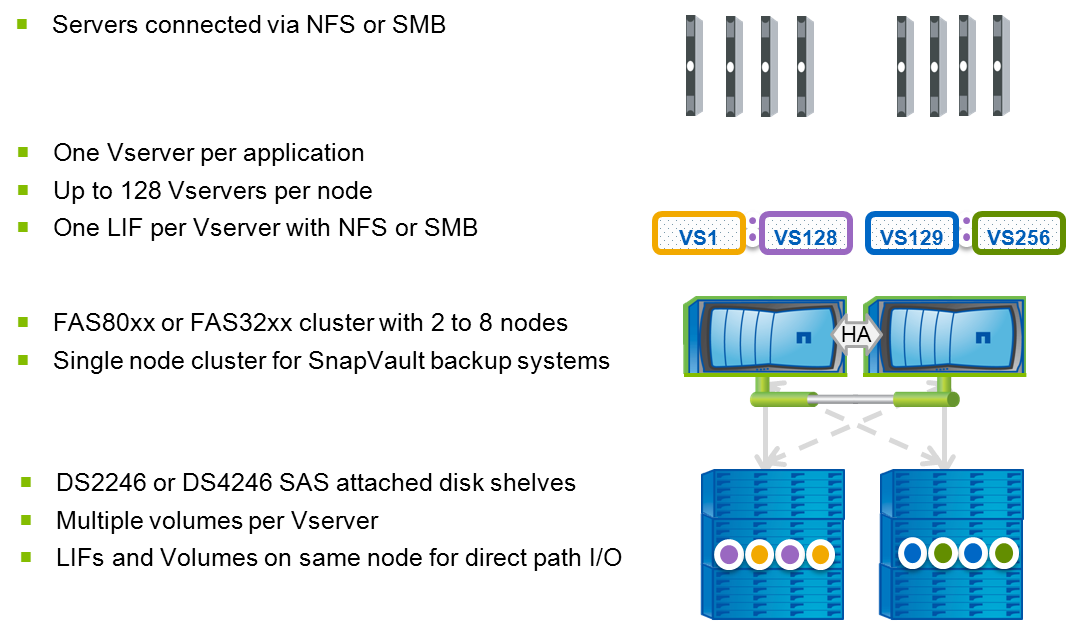
[6.7.2 Volume move with scheduled cutover 18](#_Toc387242127)

[6.7.3 LIF migration 18](#_Toc387242128)

# Introduction

## Management Summary

This document details the NetApp clustered Data ONTAP (cDOT) replication technologies used for various storage solutions within Thomson Reuters. The majority of the shared storage and dedicated cDOT solution deployments will be configured identically for a number of components, and those common configurations related to data replication will be described in this document. A typical shared storage cDOT architecture is depicted below.



## Assumptions

It is assumed the person(s) reading this document are conversant with NetApp hardware and software. They will also be conversant with the Linux and Windows operating systems, NFS, and CIFS protocols, as well as NetApp SnapMirror and SnapVault replication software.

## Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver** | **Date** | **Author** | **Key Changes** |
| 1 | November 2013 | Michael Arndt | Initial Version |
| 2 | December 2013 | Michael Arndt | Added details about configuring deduplication and compression on the snapvault secondary. |
| 3 | January 2014 | Michael Arndt | Minor cleanup. Added example of updating cluster peer addresses when nodes are added. |
| 4 | May 2014 | Michael Arndt | Updates for 7 Mode to cDOT migrations. |

## Distribution List

|  |  |
| --- | --- |
| **Name** | **Role** |
| Brett Truhler | Customer |
| Stewart Bird | Customer |
| Mitchell Vallone | Reviewer |
| Ken Zola | Reviewer |
| Ian Daniel | Reviewer |

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| cDOT | clustered Data ONTAP |
| Node | One storage controller in a cDOT system. |
| Cluster | A collection of one or more nodes that form a cDOT system. |
| **Vserver** | A logical storage virtual server, also known as a Storage Virtual Machine (SVM), which contains LIFs, Volumes, and configuration information. |
| **Port** | A physical network interface on a cDOT system. |
| Cluster Port | The ports used for intracluster traffic only. By default, each node has two cluster ports on 10-GbE ports enabled for jumbo frames. |
| Data Port | The ports used for data traffic. These ports are accessed by NFS, CIFS, FC, FCoE, and iSCSI clients for data requests. Each node has a minimum of one data port. |
| Intercluster Port | The ports used for cross-cluster communication. An Intercluster LIF can also reside on a data port. |
| **LIF** | Logical Interface – a cDOT logical network interface with an IP address, assigned to a single Vserver. |
| Cluster LIF | The LIF that is used for intracluster traffic. Cluster LIFs can be configured only on cluster ports. |
| Data LIF | The LIF that is associated with a Vserver and is used for communicating with clients. Data LIFs can be configured only on data ports. |
| Intercluster LIF | The LIF that is used for cross-cluster communication, backup, and replication. Intercluster LIFs can be configured on data ports or intercluster ports. |
| **SnapMirror** | NetApp replication software that mirrors data from one NetApp storage system to another for the purposes of migration, disaster recovery, or disk to disk backup. Both SnapMirror and SnapVault technologies in cDOT operate under the *snapmirror* command set. |
| TDP | A TDP SnapMirror relationship is one that is being used to transition data from a 7 Mode storage system to a cDOT storage system during a data migration. |
| DP | A DP SnapMirror relationship is a Data Protection relationship, for the purposes of disaster recovery or data migration between Vservers or clusters. |
| XDP | A XDP SnapMirror relationship is used for SnapVault in cDOT, and allows for more snapshots to be retained on the destination than on the source. |

# Requirements and Limitations

## Requirements for Intercluster replication

All replication traffic between a 7 Mode system and a cDOT system, or between different cDOT clusters, will require the use of an Intercluster LIF. By convention in the TR environment, we will configure one Intercluster LIF per node, on the same port used for Vserver data LIFs. The configuration of Intercluster LIFs is covered in detail in the *Thomson Reuters - cDOT Base Configuration - Deployment Guidelines* document, but we briefly discuss the Intercluster LIF requirement here as well since it is a vital part of the configuration for systems that will perform replication.

## Cluster peers and Vserver peers

The cluster peer feature allows two different cDOT clusters to coordinate replication relationships. The Vserver peer feature is an authorization infrastructure that enables a cluster administrator to set up peering applications such as SnapMirror relationships between Vservers. Only a cluster administrator can set up cluster peer and Vserver peer relationships. One Vserver can be peered with multiple Vservers within a cluster or across clusters.

## Limitations related to replication

The following limitations apply when performing replication in a cDOT environment:

* The time on the clusters that you want to connect using an intercluster network must be synchronized within 300 seconds (cluster peers can be in different time zones).
* At least one Intercluster LIF must be created on every node in the cluster.
* Every Intercluster LIF requires an IP address dedicated for Intercluster replication.
* Every Intercluster LIF on every node in a cluster must be able to connect to every Intercluster LIF on every node in a peer cluster.
* As of cDOT 8.2.1, there is a limit of 63 cluster peer relationships per cluster. There are no limits on the number of Vserver peer relationships.
* The volume language of the source volume must be the same as the volume language of the destination volume when performing SnapMirror or SnapVault replication in cDOT. A volume language of C.UTF-8 should be used on all volumes at TR.
* Volumes containing LUNs cannot be replicated from 7 Mode to cDOT at this time. Any 7 Mode LUNs moving to cDOT would need to be replicated via a client side copy operation.
* Replication between a 7 Mode system and a cDOT system can only be performed with volume SnapMirror, and only for the purposes of migration. SnapVault relationships between 7 Mode and cDOT are not supported.

# Migration from 7 Mode to cDOT

## Migration using the CLI

The process of migrating data from 7 Mode to cDOT is very similar to the process of performing a volume based migration between 7 Mode systems. Note that while vFiler configuration can be migrated between controllers in 7 Mode with DataMotion or the *vfiler migrate* CLI, there is no way to migrate a vFiler configuration in 7 Mode to a Vserver configuration in cDOT. With this in mind, a Vserver must be pre-configured on a cDOT system in order to perform a data migration from a 7 Mode system to the cDOT system. The following steps outline the process at a high level:

1. Create and configure a new Vserver on the cDOT target system.
2. Create *DP* volumes on the cDOT Vserver to match the volumes on the 7 Mode vFiler. Note that the volume language will automatically be configured to match the source volume language when the TDP SnapMirror baseline transfer is performed.
3. Perform baseline TDP SnapMirror data transfers from the 7 Mode controller or vFiler to the cDOT system. Once the SnapMirror baseline transfers have finished, the volumes can be mounted in the namespace and NFS export policies can be applied, to allow servers to test access to the new cDOT environment.
4. During the migration cutover window:
   1. Shutdown applications, then unmount NFS mountpoints and/or disconnect CIFS shares. You may also want to disable all protocols on the source vFiler to ensure that no further updates are made to the data during the cutover.
   2. Perform a final SnapMirror update. Then quiesce and break the SnapMirror relationships to make the cDOT volumes read/write.
   3. Offline the 7 Mode volumes to make them inaccessible, if desired.
   4. Perform any final configuration that is required on the cDOT Vserver, such as:
      1. Mount the volumes in the namespace.
      2. Assign volume snapshot schedules and volume QoS policies.
      3. Enable snap autodelete.
      4. Configure qtree quota rules and enable quotas for each volume as required.
      5. Configure deduplication as required.
      6. Turn the filesys-size-fixed off for the volumes.
      7. Perform SnapVault relationship setup in the new cDOT environment. This cannot be performed until after the migration cutover. Take care to create volumes on the cDOT destination system with a language that matches the source volume setting, which was automatically set the same as it was configured on the 7 Mode system. Typical volume languages in 7 Mode at TR are “*en\_US*” or “*C*”.
   5. Connect NFS clients and/or CIFS shares and start applications.
   6. Delete the TDP SnapMirror relationships and remove the Vserver transition peer relationship.

## Migration with the 7MTT

The NetApp 7 Mode Transition Tool (7MTT) is another option for performing data migration from 7 Mode to cDOT. The 7MTT is a general purpose migration GUI, and may be useful for some migration scenarios at TR. Documentation on the 7MTT can be found on the NetApp Support site:

<http://support.netapp.com/documentation/productlibrary/index.html?productID=61584>

## Using WFA for migrations

For migration use cases that will be repeated many times, such as for vFilers hosting Oracle databases or VMware datastores, WFA workflows may be created to automate many of the Vserver creation and configuration tasks, as well as configuring the SnapMirror transition relationships.

For use cases that involve 7 Mode vFilers serving data via NFS to the same group of servers for all volumes, sample WFA migration workflows have been developed for TR. These workflows should cover many common 7 Mode use cases such as Oracle, VMware, and general purpose vFilers serving data over NFS. These workflows are as follows:

* *TR\_7Mode\_to\_cDOT\_Transition\_for\_NFS\_vFiler\_\_\_Setup\_Relationships*: This workflow creates a new Vserver with identically named volumes as the source vFiler, and initiates TDP SnapMirror relationships.
* *TR\_7Mode\_to\_cDOT\_Transition\_for\_NFS\_vFiler\_\_\_Mount\_or\_Update\_or\_Cutover*: This workflow has 3 different modes of operations. The *Mount* mode mounts the volumes in the namespace and applies NFS export policies, to allow for testing access to the Vserver. The *Update* mode simply updates the TDP SnapMirror relationships. The *Cutover* mode will do a full cutover operation, including a full configuration of the new Vserver and it’s volumes after the TDP SnapMirror relationships are broken off.
* *TR\_cDOT\_SnapVault\_Setup*: While this workflow is not specifically used for transition from 7 Mode to cDOT, it would be used immediately after a migration from 7 Mode to setup the SnapVault relationships for the new cDOT Vserver and it’s volumes.

## Host based migrations

If an application provides for non-disruptive migration of data from the server side, this method may be preferred over a storage based migration with a short outage window during the migration cutover. VMware environments are an example of this, as they support storage vMotion as a methodology for non-disruptively migrating virtual machines from one datastore to another.

# SnapMirror and SnapVault in cDOT

## SnapMirror and SnapVault commands in cDOT

While 7 mode provided a completely different command set for SnapMirror and SnapVault, both SnapMirror and SnapVault in cDOT use the *snapmirror* command set. In cDOT, a SnapMirror relationship of type *DP* is similar to a 7 Mode SnapMirror relationship, and a SnapMirror relationship of type *XDP* is similar to a 7 Mode SnapVault relationship.

The *clustered ONTAP CLI Examples* section of this document provides step by step instructions on how to configure SnapMirror and SnapVault relationships in a cDOT environment.

## Similarities to 7 Mode

SnapMirror and SnapVault replication in cDOT perform similar to the way they worked in 7 Mode in a variety of ways. The following list shows some of these similarities:

* SnapMirror provides volume based replication for the purposes of DR or data migration.
* SnapVault provides replication for the purposes of disk to disk backup.
* SnapMirror maintains volume efficiency savings from deduplication or compression.
* If deduplication is enabled on a SnapVault secondary volume, a deduplication job is run automatically at the end of each SnapVault update.
* SnapMirror and SnapVault relationships can be cascaded in certain scenarios.
* SnapMirror replicates all snapshots in the source volume to the destination volume.
* SnapVault allows for a different number of snapshots to be maintained on the source and destination volumes.

## Differences from 7 Mode

While many aspects of SnapMirror and SnapVault replication remain the same, there are some changes to be aware of:

* In cDOT, SnapMirror for data migration would only be used to move volumes between Vservers or between clusters, since the non-disruptive volume movement features of cDOT would be used to move volumes between nodes and aggregates in the same cluster.
* SnapVault in cDOT also maintains volume efficiency saving from deduplication or compression.
* There is no qtree based replication in cDOT.
* SnapVault in cDOT is performed using volume to volume replication, as opposed to using volume to qtree replication in 7 Mode. This means that SnapVault relationships are always 1 volume to 1 volume in cDOT.
* SnapVault uses policies and the concept of a *snapmirror-label* to determine which snapshots from the source volume should be replicated to the destination volume.

## Deduplication and compression with SnapVault

In a cDOT environment, data efficiency savings due to deduplication and compression are maintained when replicating data via SnapMirror or SnapVault. With this in mind, the following guidelines should be followed at TR when implementing deduplication or compression in conjunction with SnapVault:

* Deduplication should be enabled on SnapVault secondary volumes only if it is not enabled on the primary volume. If deduplication is enabled on the primary volume, we do not need to enable deduplication on the secondary volume.
* Compression should be enabled on SnapVault secondary volumes only if it is not enabled on the primary volume. If compression is enabled on the primary volume, we do not need to enable compression on the secondary volume.
* In general, only use deduplication and/or compression on a SnapVault secondary volume if it is providing significant (>10%) space savings on the volume. If significant space savings are not be obtained, it may not be worth the CPU overhead to run deduplication and/or compression jobs on a SnapVault secondary.
* If deduplication and compression are enabled on the SnapVault secondary, the jobs run immediately after every SnapVault update. This schedule does not need to be configured, and there is no way to configure an alternate schedule.

## Naming conventions at TR

The following standards and naming conventions will be followed at TR for the sake of consistency:

* The job schedules will be named "*daily*" and then the hour on which they will be scheduled for. For example, "*daily1*" for a schedule that takes place at 1am every day.
* The snapmirror-label will always be "*snapvault*” when configuring volume snapshot policies and snapmirror policy rules for volumes that are backed up with SnapVault.
* The job schedules will be configured at the cluster Vserver level, but all other policies (such as volume snapshot policies and snapmirror policies) are configured on the data Vserver level.
* The SnapMirror policies that are used for SnapVault relationships will be named "*XDP<retention>*". For example, "*XDP07*", "*XDP14*", "*XDP45*".

# Non-Disruptive Volume Migration in cDOT

## ****Moving volumes****

A cDOT system is capable of moving volumes from one aggregate in a cluster to any other aggregate in the same cluster, without any disruption to the servers that are accessing this data. The following factors should be considered when using the non-disruptive volume movement technology on a cDOT system:

* A non-disruptive volume move cannot be used to move a volume from one Vserver to a different Vserver, or from one cluster to a different cluster. If this is required, SnapMirror must be used and the cutover will be disruptive.
* The volume move functionality is non-disruptive for all protocols. By default, a volume move will perform the final cutover with less than 45 seconds of I/O delay, although this is a configurable setting.
* A volume move can be configured to cutover as soon as possible after the baseline transfer is complete, or it can be configured to only perform the cutover when initiated by an administrator during a change window.
* If a volume move is being done with a manual cutover, the *Move Phase* of the volume move operation will change to *cutover\_hard\_deferred* when it is ready for the administrator to trigger the final cutover.
* The LIF migration technology that is typically used in conjunction with a volume move is non-disruptive for NFS and SMB versions 2.0 and higher. LIF migration is disruptive for SMB version 1.0.

The *clustered ONTAP CLI Examples* section of this document provides step by step instructions on how to perform non-disruptive volume movement and LIF migration operations on a cDOT system.

## ****Moving LIFs****

In the NFS and CIFS environment at TR, we will typically configure a Vserver for each application, and the Vserver will have one or more volumes along with a single LIF used for data access. This approach simplifies administration during volume move activity, as we know that all volumes in a Vserver and the LIF assigned to the Vserver must be moved to the same node in the cluster in order to maintain I/O on a direct path. Therefore, the recommendations around the use of volume move technology in cDOT for TR are as follows:

* When moving a volume to an aggregate on a different node, move all volumes from that Vserver to the same aggregate.
* Once all the volumes have been moved to the new node, also move the LIF for the Vserver to the same node.
* As described in the *Thomson Reuters - cDOT Base Configuration - Deployment Guidelines,* each LIF should be part of a valid failover group in order to make sure that the LIF can only be moved to a valid physical port when it is moved from one node to another.

# Clustered ONTAP CLI examples

## Intercluster LIFs and peering

### Create Intercluster LIF with default route

network interface create -vserver <node\_vserver> -lif <node>\_icl\_lif -role intercluster -home-node <node> -home-port a0a-<vlan> -address <ip> -netmask <netmask>

network routing-groups route create -vserver <node\_vserver> -routing-group i<network>/<mask> -destination 0.0.0.0/0 -gateway <gateway>

network interface show

### Create cDOT to cDOT cluster peer (on source or destination)

cluster peer create -peer-addrs <node1\_icl\_lif>,<node2\_icl\_lif> -username admin

cluster peer show

cluster peer health show

### Create cDOT to cDOT Vserver peer (on destination)

vserver peer create -vserver <dst\_vsname> -peer-cluster <src\_cluster> -peer-vserver <src\_vsname> -applications snapmirror

vserver peer show

### Accept cDOT to cDOT Vserver peer (on source)

vserver peer show –peer-state pending

vserver peer accept -vserver <src\_vsname> -peer-vserver <dst\_vsname>

vserver peer show

### Create 7 Mode to cDOT Vserver transition peer (on destination)

vserver peer transition create -local-vserver <vsname> -src-filer-name <7\_mode\_filer>

vserver peer transition show

### Update cluster peer addresses if nodes are added to a cluster

cluster peer modify –cluster <remote\_cluster> -per-addrs <ip addresses for all remote cluster intercluster LIFS>

cluster peer show

## Create cDOT destination volume for all replication types (on destination)

vol create -vserver <vsname> -volume <volname> -aggregate <aggrname> -size <size> -security-style <style> -space-guarantee <guarantee> -snapshot-policy <policy> -type DP –language C.UTF-8

volume show

## ****SnapMirror transition from 7 Mode to cDOT****

### Update snapmirror.access on 7 Mode system

options snapmirror.acess host=<node1\_icl\_lif>,<node2\_icl\_lif>

OR

options snapmirror.access all

### Create SnapMirror relationship

snapmirror create -source-path <7\_mode-filer>:<srcvol> -destination-path <vsname>:<volname> -type TDP

snapmirror initialize -destination-path <vsname>:<volname>

snapmirror show –type TDP

### Update, quiesce, and break relationship (on destination)

snapmirror update -destination-path <dst\_vsname>:<dst\_volname>

OR

snapmirror update –destination-path <dst\_vsname>:\*

snapmirror quiesce -destination-path <dst\_vsname>:<dst\_volname>

OR

snapmirror quiesce -destination-path <dst\_vsname>:\*

snapmirror break -destination-path <dst\_vsname>:<dst\_volname>

OR

snapmirror break -destination-path <dst\_vsname>:\*

snapmirror show –type TDP

### Mount a volume in the namespace

volume mount -vserver <vsname> -volume <volname> -junction-path /<volname>

volume show -vserver <vsname> -fields junction-pathModify volume snapshot policy

volume modify -vserver <vsname> -volume <volname> -snapshot-policy <snapshot-policy>

volume snapshot policy show -vserver <vsname>

volume show -vserver <vsname> -fields snapshot-policy

### Modify volume QoS policy

volume modify -vserver <vsname> -volume <volname> -qos-policy-group <qos\_policy\_group>

volume show -vserver <vsname> -fields qos-policy-group

### Enable snap autodelete

volume modify -vserver <vsname> -volume <volname> -space-mgmt-try-first snap\_delete

volume snapshot autodelete modify -vserver <vsname> -volume <volname> -enabled true

volume snapshot autodelete show -vserver <vsname>

### Configure qtree quotas

volume quota policy rule create -vserver <vsname> -policy-name default -volume <volname> -type tree -target <qtname> -disk-limit <XXXg>

volume quota on -vserver <vsname> -volume <volname>

volume quota policy rule show -vserver <vsname>

volume quota show -vserver <vsname>

### Enable deduplication on a volume and assign efficiency policy

volume efficiency on -vserver <vsname> -volume <volname>

volume efficiency modify -vserver <vsname> -volume <volname> -policy <policyname>

volume efficiency show –instance

### Disable filesys-size-fixed

volume modify -vserver <vsname> -volume <volname> -filesys-size-fixed false

volume show -vserver <vsname> -fields filesys-size-fixed

### Remove TDP snapmirror relationships

snapmirror delete <vsname>:\*

snapmirror show –vserver <vsname>

### Remove vserver transition peer relationship

vserver peer transition delete -local-vserver <vsname> -src-filer-name <source>

vserver peer transition show

## ****Intercluster SnapMirror from cDOT to cDOT****

### Create job schedule (on destination)

job schedule cron create -name <schedule> -minute <min> -hour <hour>

job schedule show

### Create SnapMirror relationship (on destination)

snapmirror create -source-path <src\_vsname>:<srv\_volname> -destination-path <dst\_vsname>:<dst\_volname> -type DP -schedule <schedule>

snapmirror initialize -destination-path <dst\_vsname>:<dst\_volname>

snapmirror show –type DP

### List SnapMirror relationships (on source)

snapmirror list-destinations –type DP

### Update, quiesce, and break relationship (on destination)

snapmirror update -destination-path <dst\_vsname>:<dst\_volname>

snapmirror quiesce -destination-path <dst\_vsname>:<dst\_volname>

snapmirror break -destination-path <dst\_vsname>:<dst\_volname>

snapmirror show –type DP

snapmirror show -fields lag-time –type DP

## ****Intercluster SnapVault from cDOT to cDOT****

### Create job schedule and SnapShot policy (on source)

job schedule cron create -name <schedule> -minute <min> -hour <hour>

volume snapshot policy create -vserver <src\_vsname> -policy <snapshot\_policy> -enabled true -schedule1 <schedule> -count1 <#> -snapmirror-label1 <label> -prefix1 <prefix>

volume modify –vserver <src\_vsname> -volume <src\_volname> -snapshot-policy <snapshot\_policy>

job schedule show

volume snapshot policy show

### Create job schedule and SnapMirror policy (on destination)

job schedule cron create -name <schedule> -minute <min> -hour <hour>

snapmirror policy create -vserver <dst\_vsname> -policy XDP<policy>

snapmirror policy add-rule -vserver <dst\_vsname> -policy XDP<policy> -snapmirror-label <label> -keep <#>

job schedule show

### Create SnapMirror XDP relationship (on destination)

snapmirror create -source-path <src\_vsname>:<srv\_volname> -destination-path <dst\_vsname>:<dst\_volname> -type XDP -schedule <schedule> -polcy XDP<policy>

snapmirror initialize -destination-path <dst\_vsname>:<dst\_volname>

snapmirror show –type XDP

snapmirror show -fields lag-time –type XDP

### Update, quiesce, and break relationship (on destination)

snapmirror update -destination-path <dst\_vsname>:<dst\_volname>

snapmirror quiesce -destination-path <dst\_vsname>:<dst\_volname>

snapmirror break -destination-path <dst\_vsname>:<dst\_volname>

snapmirror show –type XDP

snapmirror show -fields lag-time –type XDP

### List SnapMirror relationships (on source)

snapmirror list-destinations –type XDP

## ****Deduplication and compression on secondary volumes****

### Enabling deduplication on a secondary volume

volume efficiency on -vserver <vsname> -volume <volname>

volume efficiency show

### Enabling compression on a secondary volume

volume efficiency modify -vserver <vsname> -volume <volname> -compression true

volume efficiency show

## ****Non-disruptive volume movement in cDOT****

### Volume move and cutover all at once

volume move start -vserver <vsname> -volume <volname> -destination-aggregate <aggr>

volume move show

volume move show -instance

### Volume move with scheduled cutover

volume move start -vserver <vsname> -volume <volname> -destination-aggregate <aggr> -cutover-action wait

volume move trigger-cutover -vserver <vsname> -volume <volname>

volume move show

volume move show -instance

### LIF migration

network interface migrate -vserver <vsname> -lif <lifname> -dest-node <node>

network interface modify -vserver <vsname> -lif <lifname> -home-node <node>

network interface show