

Enterprise Storage

**SnapVault – CDOT 8.2**

**Synopsis:** This document details the snapvaulting operation within the cDOT Snapmirror command set.

**Segment:** D&E

**Authors:** David Ellis

**Contributors:** Ian Daniels

Joel Edstrom

David NG

**Document Version:** V1.0

**Date:** 12/3/2014

**Document Status**: First Revision- Distributed

Document Location: [Snapvault -CDOT 8.2](https://theshare.thomsonreuters.com/sites/AI/storwiki/_layouts/WordViewer.aspx?id=/sites/AI/storwiki/Shared%20Documents/Projects/ClusterMode/8040%20OR%20Docs/Snapvault%20TR%20Guide.doc&Source=https%3A%2F%2Ftheshare%2Ethomsonreuters%2Ecom%2Fsites%2FAI%2Fstorwiki%2FShared%2520Documents%2FForms%2FAllItems%2Easpx%3FRootFolder%3D%252Fsites%252FAI%252Fstorwiki%252FShared%2520Documents%252FProjects%252FClusterMode%252F8040%2520OR%2520Docs%26InitialTabId%3DRibbon%252EDocument%26VisibilityContext%3DWSSTabPersistence&DefaultItemOpen=1&DefaultItemOpen=1)

**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 2014

Contents

[1 Introduction 3](#_Toc405373546)

[1.1 Management Summary 3](#_Toc405373547)

[1.2 Assumptions 3](#_Toc405373548)

[1.3 Document References 3](#_Toc405373549)

[1.4 Change History 3](#_Toc405373550)

[1.5 Distribution List 3](#_Toc405373551)

[1.6 Glossary 4](#_Toc405373552)

[2 Prerequisites 5](#_Toc405373553)

[2.1 Intercluster replication 5](#_Toc405373554)

[2.1.1 Creating Intercluster LIF 5](#_Toc405373555)

[2.2 Cluster peers and vserver peers 5](#_Toc405373556)

[2.2.1 Verify Cluster peering is setup 5](#_Toc405373557)

[2.2.2 Create cluster peer relationship 5](#_Toc405373558)

[2.2.3 Vserver Requirements 5](#_Toc405373559)

[2.2.4 Peering the Vservers 5](#_Toc405373560)

[2.3 Volumes 5](#_Toc405373561)

[2.3.1 Creating DP Volume 6](#_Toc405373562)

[3 Enabling Snapshots - Source 7](#_Toc405373563)

[3.1 Snapshot Naming Convention 7](#_Toc405373564)

[3.2 Cron Schedules 7](#_Toc405373565)

[3.2.1 Cron Schedule naming convention 7](#_Toc405373566)

[3.2.2 Create Cron Schedule 7](#_Toc405373567)

[3.3 Snapshot policies 7](#_Toc405373568)

[3.3.1 Snapshot policy parameters 7](#_Toc405373569)

[3.3.2 Create snapshot policy 8](#_Toc405373570)

[3.3.3 Apply snapshot policy to volume 8](#_Toc405373571)

[4 Enabling Snapvault – Destination 9](#_Toc405373572)

[4.1.1 Cron Schedules 9](#_Toc405373573)

[4.1.2 Creating snapmirror policy 9](#_Toc405373574)

[4.1.3 Defining snapmirror policy rules 9](#_Toc405373575)

[4.1.4 Define snapmirror relationship 9](#_Toc405373576)

[4.1.5 Initialize snapmirror 10](#_Toc405373577)

[Snapmirror initialize –destination-path <destination vserver:destination\_path> 10](#_Toc405373578)

[4.1.6 Verify snapmirror relationship 10](#_Toc405373579)

[5 Creating R/O user accounts 11](#_Toc405373580)

[5.1 Create Role 11](#_Toc405373581)

[5.2 Define User 11](#_Toc405373582)

[5.3 Role/User Verification 11](#_Toc405373583)

# Introduction

## Management Summary

This document outlines creating snapvault relationships for ESX and Oracle volumes. Including, scheduling cron jobs, and creating snapshot policies both on the primary and destination sources.

## Assumptions

The person reading this document should be conversant with NetApp Snapmirror and SnapVault replication software. He/she should also have a grasp of the data protection scheme for each of technology stacks currently utilizing snapvault within the Thompson Reuters environment

## Document References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Document Name and URL** | **Version** | **Date** | **Author** |
|  | NetApp Solution Deployment Guidelines-Thomson Reuters – cDOT Replication | V4 | May 2014 | Michael Arndt |
|  | Snapvault Express Guide |  | May 2013 | NetApp |

## Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver** | **Date** | **Author** | **Key Changes** |
| 0.1 | 11/12/2014 | David Ellis | [First Draft] |
| 1.1 | 12/3/2014 | David Ellis | First Revision, Distributed |
|  |  |  |  |
|  |  |  |  |

## Distribution List

|  |  |
| --- | --- |
| **Name** | **Role** |
| Storage D&E Teams |  |
| Kevin Atkins | Storage OR Team Lead |

## 

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| cDOT | Clustered Data ONTAP |
| Node | One storage controller in a cDOT system |
| Cluster | A collection of one or more noes that for a cDOT system |
| Vserver | A logical storage virtual server also known as a Storage Virtual Machine,(SVM). Which contains LIFS, Volumes, and configuration information |
| 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. |
| 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. |

# Prerequisites

## Intercluster replication

Backup standalone clusters in the TR environment will exist outside of a primary data cluster. This will require an intercluster connection to allow for data transfer between nodes of separate clusters. A dedicated intercluster lif, created on each node within a cluster, is utilized for the data transfer.

### Creating Intercluster LIF

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

## Cluster peers and vserver peers

The cluster peer allows for coordination between clusters. A vserver peer is a feature that enables cluster administrators to set up peering applications such as SnapMirror relationships between vservers. One vserver can be peered with multiple Vservers within a cluster or across clusters.

### Verify Cluster peering is setup

*Cluster peer show*

### Create cluster peer relationship

Cluster peer create –per-addrs<remote\_ICL\_IP1,remote\_ICL\_IP2…>username,password

### Vserver Requirements

The language setting of the destination vserver matches the language setting of the vserver containing the source volume.

### Peering the Vservers

From the destination node:

*Vserver peer create –vserver <destination vserver> -peer-vserver <source vserver> -applications snapmirror –peer cluster <source node>*

*Vserver peer show –* should display pending mode until peer relationship is accepted from the source.

From the source node:

*Vserver peer accept –vserver < source vserver> -peer-vserver <destination vserver>*

## Volumes

For a volume to successfully become a part of a SnapMirrior relationship it must meet certain requirements.

* It must not be a 32 bit volume
* The destination volume must be of the type –DP, it is created with settings that reflect the best practices for destination volumes. Also listed are the adjustments that will need to be made to fit within TR standards.
* *Vol show –volume <vol name> instance*

|  |  |  |
| --- | --- | --- |
| Space Setting | Default Values | TR Values |
| Space-guarantee | Volume | None |
| Autosize | True | Deprecated cdot v8.2 |
| Autosize-mode | Grow\_shrink | Off |
| Autosize-grow-threshold-percent | 85% | n/a |
| Autosize-shrink-threshold-percent | 80% | n/a |
| Min-autosize | Initial volume size | n/a |
| Max-autosize | Max aggregate size | n/a |
| Snap reserve | 0 | 0 |
| Fractional reserve | 0 | 0 |

* Destination volume size must be equal to or larger the source volume.

### Creating DP Volume

*vol create <vol name> -aggr <aggr name> -size<{integer}[KB|GB|TB|PB]> -vserver<vserver name> -state online –type <DP> -policy <default> - autosize-mode <off> -space-guarantee <none> -snapshot-policy none*

# Enabling Snapshots - Source

There are a two components to taking snapshots locally on a cdot nodes, successful execution requires both a schedule and policy. Within the TR environment it has been determined that a singular cron schedule and policy will exist for each volume utilizing snapshots.

## Snapshot Naming Convention

Vol\_name.yyyy.mm.dd\_hhmm

## Cron Schedules

Cron schedules when applied to snapshot polices determine when incremental updates will occur. Like a Unix cron schedule, the job runs at a specified time.

### Cron Schedule naming convention

Each cron schedule utilized to trigger snapshots should be named so the volume and cron name parameter match.

Vol name = *cbxxxx\_bu\_app\_nosnap*

Cron name parameter = *cbxxx\_bu\_app\_nosnap*

### Create Cron Schedule

*Cron create –name<vol name> -month<cron\_month> -dayofweek<cron\_dayofweek> -day<cron\_dayofmonth> -hour<cron\_hour> -minute <cron\_minute>*

As an example the following cron schedule was created to take a snapshot at minute 00 of every hour.

*cron create cb0000\_de\_svtest\_7\_snap -hour all -minute 00*

## Snapshot policies

Snapshot policies for TR purposes will be specific to a vserver and volume. A snapshot policy includes at least one schedule and a maximum number of snapshot copies per schedule. When applied to a volume the policy specifies the schedule on which snapshot copies are taken and the maximum number of copies maintained.

### Snapshot policy parameters

* -policy – snapshot policy name

This should be consistent with the volume name, cbxxx\_bu\_app\_nosnap

* -schedule1 – cron schedule which this policy should follow
* -prefix - This specifies what the snapshot will be named. For TR purposes, if the snapshot is to be vaulted, the prefix should be equal to the volume name of the snapvault destination.

*sv\_<ret>\_cbxxx\_bu\_app*

* Vserver – vserver where volume resides.
* Snapmirror Label – This label allows the vaulting system to identify what snapshots need to be backed up to secondary location. Currently all labels whould be set as *snapvault*

### Create snapshot policy

*Snapshot policy create –policy <name> -vserver <vserver> -enabled true -schedule1<cron schedule> -count1 <retention> -prefix1<destination volume name> -snapmirror-label1<snapvault>*

### Apply snapshot policy to volume

A policy ceases to be effective until it is applied to the appropriate volume, enabling snapshots to occur at the defined interval.

*vol modify –vserver <vserver name> -volume <volume name> -snapshot-policy <snapshot policy name>*

# Enabling Snapvault – Destination

Much like local snapshots on the source node, creating snapvault relationships on the destination node requires a cron schedule and a snapmirror policy. The snapmirror policy will be particular to a volume on a given vserver however; the cron jobs are able to be utilized for multiple polices as there is no prefix associated with snapmirror policies.

### Cron Schedules

Cron schedules function similarly across all nodes with respect to their schedules. When creating cron schedules to be used with snapmirror policies they should be named in the following manner, *XDP\_Time*.

The following command would create a schedule which could be utilized to trigger a snapmirror job at 9am daily. All snapmirror policies to be executed at the same time should use the same cron schedule. There is no limit to the number of times a cron schedule can be utilized.

*cron create –name XDP\_09 –hour 9*

### Creating snapmirror policy

The snapmirror policy specifies the configuration attribute for a snapmirror relationship. For vault relationships snapmirrors have rules that define which snapshot copies are protected. Policy names should equivalent to the snapvault destination volume name.

*snapmirror policy create –vserver <Vserver name> -policy <sm\_policy>*

### Defining snapmirror policy rules

A policy that will be associated with a Snapmirror vault relationship must have atleast one rule and at most ten.

Parameters:

* Policy - snapmirror policy name
* Snapmirror label – this should be consistent with the label on the source volume for TR purposes the label will always be snapvault until deemed otherwise necessary.
* Keep – specifies the maximum number of snapshot copies that are retained on the Snapmirror vault destination,also known as the defined retention period, 7,14,30,45 days.

*snapmirror policy add-rule –vserver <vserver> -policy<sm\_policy> -snapmirror-label<snapvault> -keep<7/14/30/45>*

### Define snapmirror relationship

*snapmirror create –source-path <source-vserver:source path> -destination-path <destination vserver:destination path> -policy <sm\_policy> -type <XDP> -schedule <XDP\_time>*

### Initialize snapmirror

### *Snapmirror initialize –destination-path <destination vserver:destination\_path>*

### Verify snapmirror relationship

Successful creation of a snapmirror relationship can be verified by the following command. The mirrored state will equal snapmirorred/transferring if relationship is functioning properly.

*Snapmirror show –destination-path <destination\_vserver:destination\_vol>*

# Creating R/O user accounts

It has been noted that in certain instances it will be necessary for individuals, outside of the Storage team, may require access to certain command directories, in this specific instance, the snap vault command directory. Granting a user read only access to the snapmirror command directory will allow defined user to view snapvault relationship information however, it will prohibit them from making any administrative changes.

In most circumstances user accounts should be given ssh access and utilize a predefined key. Existing roles should be expanded to include all necessary functionality rather creating multiple accounts for specific tasks. The following commands will grant a defined user in this instance, db\_snapvault, access to the snapmirror command directory.

## Create Role

A specific role should be created allowing read only access to the snapmirror command directory. These commands are executed against the destination controller.

*Security login role create –role db\_snapvault –cmddirname –snapmirror –access readonly –query “” –vserver<node mgmt vserver>*

## Define User

An existing user, or a new user can be assigned to utilize any previously created role. This example will assume no previous user is in existence.

*Security login create –username dv\_snapvault –application ssh –authmethod publickey <appropriate public key> -vserver <node mgmt vserver>*

## Role/User Verification

Verifiy user Accounts:

*Security login show –vserver<vserver> -username <uname>*

Verify Roles

*Security login role show <role>*