



NETAPP TECHNICAL REPORT

Disaster Recovery Support for DataFabric Manager Data Using SnapDrive

Saravanan Manickam and Shridhar Bandekar, NetApp
May 2008 | TR-3655

ABSTRACT

This technical report explains how to set up the Disaster Recovery Support for DataFabric[®] Manager (DFM) Data feature without using Protection Manager. In Operations Manager for DataFabric Manager Server 3.7, disaster recovery support is available only on Windows[®] operating systems that use Protection Manager.

Customers who do not have a Protection Manager license or who have their DFM set up on Linux[®] or Solaris[™] should refer to this document for information about disaster recovery support for their DFM data. The solution described uses Operations Manager, SnapDrive[®], and SnapMirror[®] products.

This information is for users of NetApp[®] storage systems that use these products.

TABLE OF CONTENTS

1	INTRODUCTION	3
1.1	ABOUT SNAPDRIVE	3
1.2	ABOUT SNAPMIRROR.....	3
2	DISASTER RECOVERY SUPPORT FOR DFM DATA	4
2.1	DEPLOYMENT SCENARIO	4
2.2	HARDWARE AND SOFTWARE REQUIREMENTS	5
2.3	SETTING UP DISASTER RECOVERY SETUP USING SNAPDRIVE AND SNAPMIRROR	5
2.4	PERFORMING FAILOVER TO THE SECONDARY DFM SERVER FROM THE PRIMARY	9
2.5	PERFORMING FAILBACK TO THE PRIMARY DFM SERVER FROM THE SECONDARY	11
3	GLOSSARY	17
4	CONCLUSION	18

1 INTRODUCTION

Operations Manager for DataFabric Manager Server 3.7 supports a new feature, Disaster Recovery for DataFabric Manager Data. This feature, supported only on Windows, uses Protection Manager and SnapDrive for Windows.

This document is meant for:

- Customers who do not have a Protection Manager license but need disaster recovery support for their DFM data on Windows.
- Customers who have their DFM set up on a Linux or Solaris platform, with or without a Protection Manager license, and need disaster recovery support for their DFM data.

To understand this technical report, you should understand the concepts and working of Operations Manager, Data ONTAP[®], SnapDrive, and SnapMirror.

Note: The term “DFM data” means data that is stored on the server related to Operations Manager, Protection Manager, Performance Advisor, and Provisioning Manager.

1.1 ABOUT SNAPDRIVE

SnapDrive is a tool for storage provisioning and Snapshot[™] management on Solaris, Linux, and Windows systems connecting to NetApp FAS storage systems. Using this tool, administrators can easily access the following storage provisioning functionalities:

- Creating LUNs
- Connecting LUNs to the UNIX[®] or Windows system
- Creating a volume group and file system on the LUNs
- Expanding the volume groups by using simple commands on the system console
- Creating user data copies that are host-file-system-consistent
- Restoring data from the Snapshot copies.

Other NetApp software products, such as SnapManager[®] for Oracle[®] and SnapManager for SAP[®], use these functionalities to allow Snapshot management that is integrated into applications that use the storage.

1.2 ABOUT SNAPMIRROR

SnapMirror, available with Data ONTAP, is a method of mirroring volumes and qtrees. Using SnapMirror, you can enable and disable scheduled and manual data transfers, request information about transfers, initialize data transfer, start an update of a mirror, temporarily pause updates to a mirror, break mirror relationships, resynchronize broken mirrors, list destination information, release child mirrors, store volume images to tape, retrieve volume images from tape, and abort ongoing transfers.

SnapMirror can also be used to replicate volumes or qtrees. The processes and behaviors involved are slightly (and sometimes subtly) different between the various kinds of data mirroring.

2 DISASTER RECOVERY SUPPORT FOR DFM DATA

This section describes a deployment scenario of the Disaster Recovery for DataFabric Manager Data feature on a Solaris and Linux host using SnapDrive for UNIX. You might execute equivalent SnapDrive for Windows commands on a Windows DFM server. This section also lists the hardware and software required and the steps to perform disaster recovery of DFM data.

2.1 DEPLOYMENT SCENARIO

In the example shown in Figure 1, Linux host Alpha is connected to primary NetApp storage system Jack. The DataFabric Manager data resides on a LUN created on the primary storage system. The DFM data on primary storage system Jack is mirrored by a secondary storage system Jill for disaster recovery. The secondary storage system may be nearby location or at a distant location. The Linux Host Beta is configured as a standby DFM server.

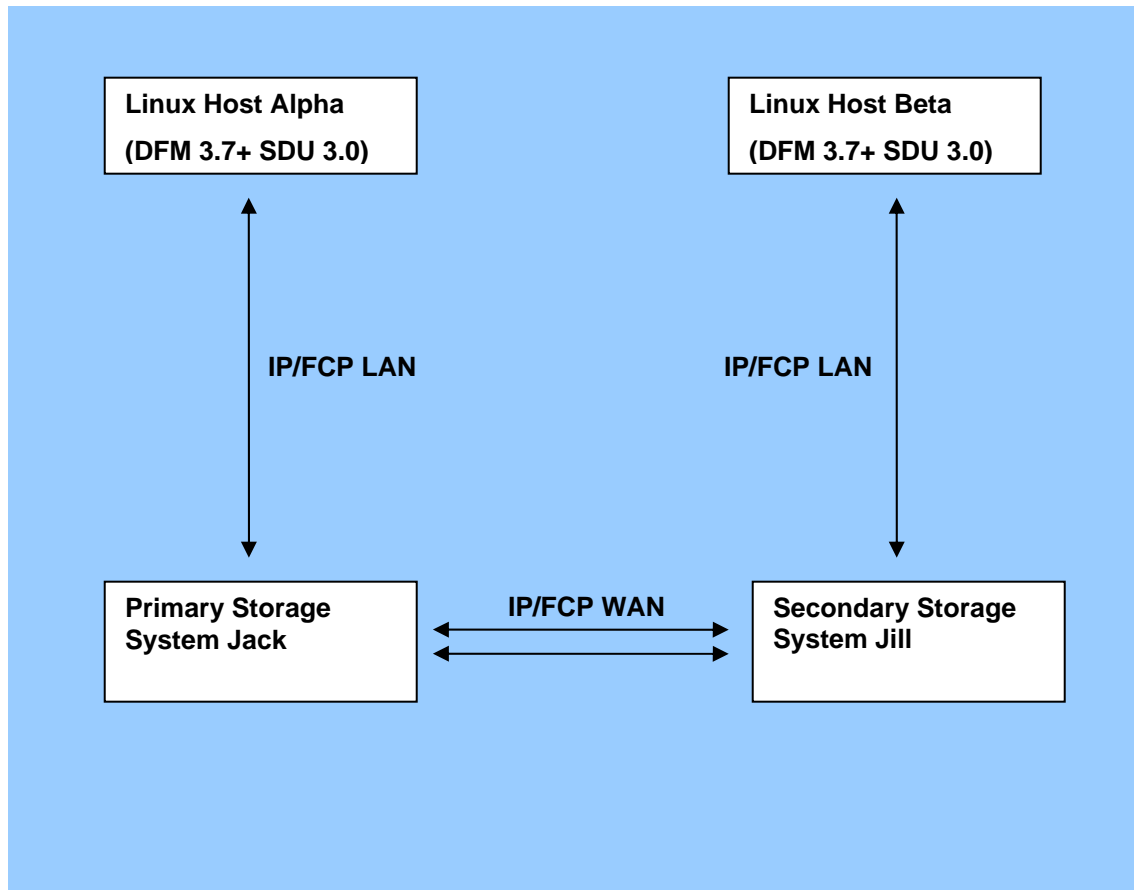


Figure 1

In summary, connectivity is required only between the following hosts :

- Connectivity between Alpha and Jack
- Connectivity between Beta and Jill
- Connectivity between Jack and Jill

2.2 HARDWARE AND SOFTWARE REQUIREMENTS

- Two Solaris, Linux or Windows servers with the same version and patches installed.
- Two NetApp or IBM storage systems with the same Data ONTAP version
- The DFM servers and storage systems must have IP (iSCSI) or Fibre Channel Protocol (FCP) connectivity based on the protocol you intend to use.
- SnapDrive for UNIX 3.0 or later is required for UNIX and Linux systems. Refer to the SnapDrive for UNIX compatibility matrix to learn about supported combinations:
- https://now.netapp.com/NOW/knowledge/docs/olio/guides/snapdrive_unix_compatibility
- SnapDrive for Windows 6.0 or later is required for Windows systems.
- Disaster recovery is supported only when DataFabric Manager data is stored on a NetApp LUN.
- The user must have SnapMirror and SnapRestore licenses on both the source and destination storage systems.
- The user must configure the Snapshot-based backup for DataFabric Manager.
- NetApp recommends a dedicated flexible volume for DataFabric Manager data because Volume SnapMirror is used to mirror the data.
- The storage systems involved in disaster recovery setup should be accessible via their DNS names. This is important in establishing the SnapMirror relationship.

2.3 SETTING UP DISASTER RECOVERY SETUP USING SNAPDRIVE AND SNAPMIRROR

Here is the process to set up disaster recovery for DFM. Detailed steps follow.

Primary DFM Server

- Install DataFabric Manager Server 3.7.
- Install either SnapDrive for UNIX or SnapDrive for Windows (see section 2.2).
- Create a flexible volume on both the primary and the secondary storage system with the same size and physical properties.
- Create FCP or iSCSI based storage (LUN) for DataFabric Manager data on the primary storage system by using SnapDrive.
- Run the `dfm datastore setup` command to migrate the data to a directory in the LUN.
- Configure a backup schedule for Snapshot-based backup.
- Run the `dfm backup diag` command to check the DFM data backup information.
- Set up a mirror relationship to mirror DFM data from the primary storage system to the secondary storage system.
- Configure a SnapMirror schedule on the secondary storage system

Secondary DFM Server:

- Install DataFabric Manager Server 3.7.
- Install either SnapDrive for UNIX or SnapDrive for Windows (see section 2.2).
- Configure the server as a standby system by disabling the DFM services. Use the `dfm service disable` command.

To Set Up Disaster Recovery on the Primary DFM Server (Alpha)

These steps assume that you have installed DataFabric Manager Server 3.7 and either SnapDrive for UNIX or SnapDrive for Windows on the server.

- 1 Check that the DFM services are up.
- 2 Create a flexible volume on Jack and one on Jill with the same size and physical properties for DFM data.

In the example, flexible volume `primary_volume` is created on Jack and `secondary_volume` is created on Jill.

- 3 Create a LUN on `primary_volume` and mount it on Alpha, as shown in the following example

```
[root@Alpha /]# snapdrive storage create -lun Jack: /vol/primary_volume/dfm_lun -lunsize
15g -fs /mnt/dfmserver
LUN Jack:/vol/primary_volume/dfm_lun ... created
mapping new lun(s) ... done
discovering new lun(s) ...done

LUN to device file mappings:
- Jack:/vol/primary_volume/dfm_lun => /dev/sdc

disk group dfmserver_SdDg created
host volume dfmserver_SdHv created
file system /mnt/dfmserver created

[root@Alpha /]#
```

In this example:

`dfm_lun` is the new LUN created on `primary_volume`.

`/mnt/dfmserver` is the mount point where the created LUN is mounted.

Note: On a Windows server, use the SnapDrive for Windows Disk Create wizard, or use the `sdcli disk create` command to create the LUN.

- 4 Move the DFM data to the LUN mounted at `/mnt/dfmserver` by using the `dfm datastore setup /mnt/dfmserver` command.

Note: For a Windows DFM server, move the DFM data to the LUN mounted at the appropriate drive.

- 5 Configure the Snapshot-based backup schedule on Alpha.

From the Operations Manager Web browser, go to Control Center > Setup > Database Backup and schedule Snapshot-based backup.

The backup can also be scheduled by using the `dfm backup schedule set` command.

Example: To create a Snapshot-based hourly backup schedule, use

```
$dfm backup schedule set -t snapshot -H 30.
```

- 6 Make sure that the DFM data is getting backed up according to the Snapshot schedule.

To verify, use the `snap list` command on the primary storage system, or use the `dfm backup list` command.

- 7 Run the `dfm backup diag` command to check the DFM data backup information, and make sure that Snapshot is a default backup type.
- 8 Log in to the secondary storage system (Jill) and create a mirror relationship from the primary storage system (Jack) to Jill. The secondary volume on Jill must be in restricted mode before establishing the mirror relationship. Observe the established mirror relationship on the primary and secondary storage systems.

```
Jill*> snapmirror initialize -S Jack:primary_volume Jill:secondary_volume
```

```
Transfer started.
```

```
Monitor progress with 'snapmirror status' or the snapmirror log.
```

```
Jill*> snapmirror status
```

```
Snapmirror is on.
```

Source	Destination	State	Lag	Status
Jack:primary_volume done)	Jill:secondary_volume	Uninitialized	-	Transferring (150 MB

```
Jill*> snapmirror status
```

```
Snapmirror is on.
```

Source	Destination	State	Lag	Status
Jack:primary_volume	Jill:secondary_volume	Snapmirrored	00:00:28	Idle

```
Jill*>
```

```
Jack> snapmirror status
```

```
Snapmirror is on.
```

Source	Destination	State	Lag	Status
Jack:primary_volume	Jill:secondary_volume	Source	00:01:10	Idle

```
Jack>
```

- 9 Create a SnapMirror schedule on Jill, so that the data in the primary volume of Jack is periodically mirrored to the secondary volume of Jill.

To ensure minimal data loss, be sure to create a SnapMirror schedule so that it runs immediately after the scheduled Snapshot-based DFM backup.

Note: Be cautious when you edit the `/etc/snapmirror.conf` file. If the file is non-empty, make sure that the new entries are appended instead of flushing and writing a new file.

```
Jill> rdfile /etc/snapmirror.conf
```

```
Gamma:volume1 Jill:volume2 - 55 * * *
```

```
Jill> wrfile /etc/snapmirror.conf
```

```
Gamma:volume1 Jill:volume2 - 55 * * *
Jack:primary_volume Jill:secondary_volume - 30 0,7-21 * 1,2,3,4,5
read: error reading standard input: Interrupted system call
```

```
Jill> rdfile /etc/snapmirror.conf
```

```
Gamma:volume1 Jill:volume2 - 55 * * *
Jack:primary_volume Jill:secondary_volume - 30 0,7-21 * 1,2,3,4,5
```

The entry "Jack:primary_volume Jill:secondary_volume - 30 0,7-21 * 1,2,3,4,5" is read as,

Jack:primary_volume <=> Source

Jill:secondary_volume <=> Destination

- <=> Empty arguments (kbs and restart are set to default)

30 <=> 30 minutes

0,7-21 <=> 0 hour (midnight), 7 hrs to 21 hrs (7 am to 9 pm)

* <=> Every day-of-month

1,2,3,4,5 <=> Monday through Friday

The entry Jack:primary_volume Jill:secondary_volume - 30 0,7-21 * 1,2,3,4,5 kicks off the mirror at half past every hour between 7:30 a.m. and 9:30 p.m. and again at midnight, Monday through Friday.

For more information on configuring /etc/snapmirror.conf, go to http://<filer_ip>/na_admin/man/man5/na_snapmirror.conf.5.html.

- Run the `snapdrive storage show -all` command on the primary DFM server and save the output. When you perform failover, you will need the disk group name (SdDg) and the host volume name (SdHv)

```
[root@Alpha /]# snapdrive storage show -all
dg: dfmserver_SdDg          dgtype lvm
hostvol: /dev/mapper/dfmserver_SdDg-dfmserver_SdHv    state: AVAIL
fs: /dev/mapper/dfmserver_SdDg-dfmserver_SdHv    mount point: /mnt/dfmserver (persistent) fstype
ext3

device filename    adapter path    size    proto    state    clone    lun path                backing
snapshot
-----
/dev/sdc            - P            15g    iscsi    online   No       fas270-184-201:/vol/primary_volume/dfm_lun
-
[root@Alpha /]#
```

In the example, the disk group name (SdDg) is `dfmserver_SdDg` and the host volume name (SdHv) is `dfmserver_SdHv`.

Note: This information is not required if SnapDrive for Windows is used.

Disaster recovery setup for DFM data is now complete.

2.4 PERFORMING FAILOVER TO THE SECONDARY DFM SERVER FROM THE PRIMARY

If one of the following events occurs, you need to perform failover to the secondary DataFabric Manager server from the primary:

- Disaster happens at the primary site
- The primary site is shut down for maintenance

Here is the process for failing over from the primary site to the secondary site. Detailed steps follow.

- Break the mirror relationship between the secondary and primary storage systems by using the `snapmirror break` command (unless the relationship was already broken during the disaster).
- Use SnapRestore® to restore the secondary volume with the latest consistent DFM Snapshot copy.
- Connect and mount the mirrored LUN present on the secondary volume from the secondary DFM server by using the SnapDrive storage connect CLI.
- Enable the services on the secondary DFM server.
- Configure DataFabric Manager to use the DFM data available at the new mount point by using the `dfm datastore setup -n/` command.
- Start the remaining DFM services on the secondary DFM server.

To Fail Over from the Primary to the Secondary Site

- 1 Break the mirror relationship between the secondary storage system (Jill) and the primary storage system (Jack) by using the `snapmirror break` command (unless the relationship has already broken during the disaster).

```
Jill*> snapmirror status
Snapmirror is on.

Source                Destination           State   Lag    Status
Jack:primary_volume   Jill:secondary_volume Snapmirrored 00:04:17 Idle

Jill*> snapmirror break secondary_volume
snapmirror break: Destination secondary_volume is now writable.

Volume size is being retained for potential snapmirror resync. If you would like to grow the volume and do not expect to resync, set vol option fs_size_fixed to off.

Jill*> snapmirror status
Snapmirror is on.

Source                Destination           State   Lag    Status
Jack:primary_volume   Jill:secondary_volume Broken-off 00:05:23 Idle
Jill*>
```

- 2 Use SnapRestore to restore the secondary volume with the latest consistent DFM Snapshot copy.

```
Jill*> snap list secondary_volume
```

Volume secondary_volume

working...

<i>%/used</i>	<i>%/total</i>	<i>date</i>	<i>name</i>
0% (0%)	0% (0%)	Mar 19 16:30	Jill (0084196620)_secondary_volume.2
1% (1%)	0% (0%)	Mar 19 16:26	dfm_backup_2008-03-18_22-00-01.sndb
2% (1%)	0% (0%)	Mar 19 16:21	Jill (0084196620)_secondary_volume.1
9% (3%)	0% (0%)	Mar 19 15:49	dfm_backup_2008-03-18_21-23-02.sndb

Jill*> snap restore -s dfm_backup_2008-03-18_22-00-01.sndb secondary_volume

WARNING! This will revert the volume to a previous snapshot.

All modifications to the volume after the snapshot will be
irrevocably lost.

Volume secondary_volume will be made restricted briefly before coming back online.

Are you sure you want to do this? y

You have selected volume secondary_volume, snapshot dfm_backup_2008-03-18_22-00-01.sndb

Proceed with revert? y

Volume secondary_volume: revert successful.

Jill*>

- 3 Connect and mount the mirrored LUN present on the secondary volume from the secondary DFM server by using the `snapdrive storage connect` command.

```
[root@Beta /]# snapdrive storage connect -lun Jill:/vol/secondary_volume/dfm_lun -fs /mnt/dfmserver_secondary -hostvol dfmserver_SdDg/dfmserver_SdHv
```

```
mapping lun(s) ... done
```

```
discovering lun(s) ... done
```

```
LUN Jill:/vol/secondary_volume/dfm_lun connected
```

```
- device filename(s): /dev/sdc
```

```
Importing dfmserver_SdDg
```

```
Connected fs /mnt/dfmserver_secondary
```

```
[root@Beta /]#
```

In this example:

`/mnt/dfmserver_secondary` is the new mount point where the LUN will be connected.

`dfmserver_SdDg` is the disk group name.

`dfmserver_SdHv` is the host volume name

Note: On a Windows server, use the SnapDrive for Windows Disk Connect wizard, or use the `sdcli disk connect` command to connect to the mirrored LUN.

- 4 Enable the services on the secondary DFM server (Beta) by using the `dfm service enable` command.
- 5 Configure DataFabric Manager to use the DFM data available at the new mount point `/mnt/dfmserver_secondary` by using the `dfm datastore setup -n` command

```
[root@Beta /]# dfm datastore setup -n /mnt/dfmserver_secondary
Stopping all services and jobs...
Changing database configuration settings...
Changed dbDir to /mnt/dfmserver_secondary/data.
Updated dbLogDir to /mnt/dfmserver_secondary/data.
Starting sql service...
Changed perfArchiveDir to /mnt/dfmserver_secondary/perfdata.
Changed scriptDir to /mnt/dfmserver_secondary/script-plugins.
Changed reportsArchiveDir to /mnt/dfmserver_secondary/reports.
Changed pluginsDir to /mnt/dfmserver_secondary/plugins.
Changed cmsDir to /mnt/dfmserver_secondary/web/cms.
Changed softwareDir to /mnt/dfmserver_secondary/web/software.
Changed databaseBackupDir to /mnt/dfmserver_secondary/data.
NOTE: You must move all archive database backups from '/mnt/dfmserver_new/data/' to new
directory.
No database upgrade is necessary.
Redefining SQL Views ...
Completed data setup.
[root@Beta /]#
```

- 6 Start the remaining DFM service on Secondary using 'dfm service start' command

2.5 PERFORMING FAILBACK TO THE PRIMARY DFM SERVER FROM THE SECONDARY

For failback operation, consider two cases. Case A addresses the failback steps for a maintenance shutdown scenario. Case B addresses the failback steps for a disaster in which the primary storage system is destroyed.

Case A: You have done failover to the secondary DFM server because of primary site maintenance. You do active management on managed hosts from the secondary DFM server.

After the maintenance, you bring the old primary DFM server and primary storage system back and you want to fail back. In this case, there would be incremental data at the secondary site (secondary storage system), which you need to sync up with the primary. To achieve this failback operation, you follow this procedure. Detailed steps follow.

- Make sure that Snapshot-based backup is running on the secondary DFM server.
- Resync the DFM data from the secondary to the primary storage system via a reverse mirror relationship.

- Configure the mirror schedules by using `/etc/snapmirror.conf`, as in step 9 of section 2.3, to resync the data from secondary to primary storage system via a reverse mirror relationship.
- When the primary server is ready, break the reverse mirror relationship.
- Run the `snapdrive storage show -all` command on the secondary server and save the output. When performing failback, you need the disk group name (SdDg) and the host volume name (SdHv).
- Enable the DFM services on the primary.
- Disable the DFM services on the secondary.
- Restore the primary volume of the primary storage system with the latest consistent DFM Snapshot copy.
- Connect to the newly mirrored primary volume or LUN by using the `snapdrive storage connect` command from the primary server. Unmap or disconnect the existing LUN on the primary server, unless you bring up a new server.
- Configure DataFabric Manager to use the new mirrored data available at the primary storage system.
- Reverse the mirror relationship (forward mirror) so that data gets mirrored from the primary storage system's primary volume to the secondary storage system's secondary volume.

Case B: You have done failover to the secondary DFM server because of a disaster at the primary site. You do active management on managed hosts from the secondary DFM server.

After the disaster, you bring up a new primary DFM server and primary storage system and you want to fail back. In this case, you need to transfer the data from the secondary site (secondary storage system) to the primary. To achieve this failback operation, you follow this procedure. Detailed steps follow.

- Make sure that Snapshot-based backup is running on the secondary DFM server.
- Use the `snapmirror initialize` command to create mirror relationship from the secondary storage system to the new primary storage system.
- Configure the mirror schedules by using `/etc/snapmirror.conf`, as in step 9 of section 2.3, to mirror the data from the secondary to the new primary storage system.
- To use the new primary site, follow the steps in section 2.4. Here, the only difference is that you are failing over from the secondary storage system to the new primary storage system.

To Fail Back to the Primary DFM Server from the Secondary

- 1 Make sure that Snapshot-based backup is running on the secondary DFM server by using the `dfm backup schedule get` command.
- 2 Resync the DFM data from the secondary to the primary storage system via a reverse mirror relationship.

```

Jack> snapmirror status
Snapmirror is on.
Source                Destination            State    Lag    Status
Jack:primary_volume  Jill:secondary_volume Source    01:24:52 Idle
Jack>

Jill*> snapmirror status

```

Snapmirror is on.

<i>Source</i>	<i>Destination</i>	<i>State</i>	<i>Lag</i>	<i>Status</i>
<i>Jack:primary_volume</i>	<i>Jill:secondary_volume</i>	<i>Uninitialized</i>	<i>01:24:13</i>	<i>Idle</i>

Jill>*

Jack> snapmirror resync -S Jill:secondary_volume Jack:primary_volume

The resync base snapshot will be: dfm_backup_2008-03-18_22-00-01.sndb

These newer snapshots will be deleted from the destination:

hourly.0

Jill (0084196620)_secondary_volume.2

Are you sure you want to resync the volume? y

Volume primary_volume will be briefly unavailable before coming back online.

Revert to resync base snapshot was successful.

Transfer started.

Monitor progress with 'snapmirror status' or the snapmirror log.

Jack> snapmirror status

Snapmirror is on.

<i>Source</i>	<i>Destination</i>	<i>State</i>	<i>Lag</i>	<i>Status</i>
<i>Jill:secondary_volume</i>	<i>Jack:primary_volume</i>	<i>Snapmirrored</i>	<i>00:01:29</i>	<i>Idle</i>

Jack>

Jill> snapmirror status*

Snapmirror is on.

<i>Source</i>	<i>Destination</i>	<i>State</i>	<i>Lag</i>	<i>Status</i>
<i>Jack: primary_volume</i>	<i>Jill: secondary_volume</i>	<i>Uninitialized</i>	<i>01:26:34</i>	<i>Idle</i>
<i>Jill: secondary_volume</i>	<i>Jack: primary_volume</i>	<i>Source</i>	<i>00:01:23</i>	<i>Idle</i>

Jill>*

- 3 Configure the mirror schedules by using /etc/snapmirror.conf, as in step 9 of section 2.3, to resync the data from the secondary to the primary storage system via a reverse mirror relationship.
- 4 When the primary server is ready, break the reverse mirror relationship. Later, you need to make the primary DFM server up with the new data available at Jack

Jack> snapmirror break primary_volume

snapmirror break: Destination primary_volume is now writable.

Volume size is being retained for potential snapmirror resync. If you would like to grow the volume and do not expect to resync, set vol option fs_size_fixed to off.

Jack> snapmirror status

Snapmirror is on.

```

Source                Destination            State      Lag      Status
Jill:secondary_volume Jack:primary_volume  Broken-off 00:08:46 Idle
Jack>

Jill*> snapmirror status
Snapmirror is on.

Source                Destination            State      Lag      Status
Jack:primary_volume   Jill:secondary_volume Uninitialized 01:34:30 Idle
Jill:secondary_volume Jack:primary_volume   Source     00:09:19 Idle
Jill*>

```

- 5 Run the `snapdrive storage show -all` command on the secondary server and save the output. When you perform fail back, you will need the disk group name (SdDg) and the host volume name (SdHv).

```

[root@Beta /]# snapdrive storage show -all
dg: dfmserver_SdDg      dgtype lvm
hostvol: /dev/mapper/dfmserver_SdDg-dfmserver_SdHv   state: AVAIL
fs: /dev/mapper/dfmserver_SdDg-dfmserver_SdHv   mount point: /mnt/dfmserver_secondary
(persistent) fstype ext3

device filename      adapter path  size  proto  state  clone  lun path              backing
snapshot
-----
/dev/sdc              - P          15g  iscsi  online No    fas270-184-
173:/vol/secondary_volume/dfm_lun  -
[root@Beta /]#

```

In the example, the disk group name (SdDg) is `dfmserver_SdDg` and the host volume name (SdHv) is `dfmserver_SdHv`.

Note: If SnapDrive for Windows is used, this information is not required.

- 6 Enable the DFM services on the primary DFM server, Alpha.
- 7 Disable the DFM services on the secondary DFM server, Beta.
- 8 Restore the primary volume on the primary storage system Jack with the latest consistent DFM Snapshot copy.

```

Jack> snap list primary_volume
Volume primary_volume
working...

%/used  %/total  date      name
-----
0% ( 0%) 0% ( 0%) Mar 19 12:25 Jack(0084193458)_primary_volume.1
2% ( 1%) 0% ( 0%) Mar 19 12:21 dfm_backup_2008-03-19_23-25-03.sndb

```

```

3% ( 1%) 0% ( 0%) Mar 19 12:16 dfm_backup_2008-03-19_23-20-21.sndb
8% ( 5%) 0% ( 0%) Mar 19 10:56 dfm_backup_2008-03-18_22-00-01.sndb
9% ( 1%) 0% ( 0%) Mar 19 10:51 Jill(0084196620)_secondary_volume.1
10% ( 1%) 0% ( 0%) Mar 19 10:46 dfm_backup_2008-03-18_21-49-58.sndb

```

```

Jack> snap restore -s dfm_backup_2008-03-19_23-25-03.sndb primary_volume
WARNING! This will revert the volume to a previous snapshot.
All modifications to the volume after the snapshot will be
irrevocably lost.
Volume primary_volume will be made restricted briefly before coming back online.
Are you sure you want to do this? y

You have selected volume primary_volume, snapshot dfm_backup_2008-03-19_23-25-03.sndb
Proceed with revert? y
Volume primary_volume: revert successful.
Jack>

```

- 9 Connect to the newly mirrored dfm_lun on the primary volume of the primary storage system Jack by using the `snapdrive storage connect` command from the primary server. Unmap or disconnect the existing LUN on the primary, unless you bring up a new DFM server.

```

[root@Alpha /]# snapdrive storage show -all
dg: dfmserver_SdDg          dgtype lvm
hostvol: /dev/mapper/dfmserver_SdDg-dfmserver_SdHv   state: AVAIL
fs: /dev/mapper/dfmserver_SdDg-dfmserver_SdHv   mount point: /mnt/dfmserver (persistent) fstype
ext3

```

device	filename	adapter	path	size	proto	state	clone	lun	path	backing	snapshot
/dev/sdc		-	P	15g	iscsi	online	No		Jack:/vol/primary_volume/dfm_lun	-	

As seen from above output, the mapping to the LUN exist. Execute 'snapdrive storage disconnect' cli To unmap it. The cli need not be executed if no LUN mapping exist.

```

[root@Alpha /]# snapdrive storage disconnect -fs /mnt/dfmserver
disconnect file system /mnt/dfmserver
- fs /mnt/dfmserver ... disconnected
- hostvol dfmserver_SdDg/dfmserver_SdHv ... disconnected
- dg dfmserver_SdDg ... disconnected

```

```
- LUN Jack:/vol/primary_volume/dfm_lun ... disconnected
```

```
0001-669 Warning:
```

```
Please save information provided by this command.
```

```
You will need it to re-connect disconnected filespecs.
```

```
[root@Alpha /]# snapdrive storage connect -lun Jack:/vol/primary_volume/dfm_lun -fs /mnt/dfmserver -  
hostvol dfmserver_SdDg/dfmserver_SdHv
```

```
mapping lun(s) ... done
```

```
discovering lun(s) ... done
```

```
LUN Jack:/vol/primary_volume/dfm_lun connected
```

```
- device filename(s): /dev/sdc
```

```
Importing dfmserver_SdDg
```

```
Connected fs /mnt/dfmserver
```

```
[root@Alpha /]#
```

Note: On a Windows server, use the SnapDrive for Windows Disk Connect wizard or the `sdcli disk connect` command to connect to the mirrored LUN.

- 10 Configure DataFabric Manager to use the newly mirrored DFM data available at the mounted file system `/mnt/dfmserver`.

```
[root@Alpha /]# dfm datastore setup -n /mnt/dfmserver
```

```
Stopping all services and jobs...
```

```
Changing database configuration settings ...
```

```
Updated dbLogDir to /mnt/dfmserver/data.
```

```
Starting sql service...
```

```
Changed perfArchiveDir to /mnt/dfmserver/perfdata.
```

```
Changed scriptDir to /mnt/dfmserver/script-plugins.
```

```
Changed reportsArchiveDir to /mnt/dfmserver/reports.
```

```
Changed pluginsDir to /mnt/dfmserver/plugins.
```

```
Changed cmsDir to /mnt/dfmserver/web/cms.
```

```
Changed softwareDir to /mnt/dfmserver/web/software.
```

```
No database upgrade is necessary.
```

```
Redefining SQL Views ...
```

```
Completed data setup.
```

```
[root@Alpha /]#
```

Start the DFM services by using the `dfm service start` command.

- 11 Reverse the mirror relationship (set forward mirror) so that data is mirrored from the primary storage system's primary volume to the secondary storage system's secondary volume.

```
Jill*> snapmirror status
Snapmirror is on.

Source                Destination           State   Lag    Status
Jack:primary_volume   Jill:secondary_volume Uninitialized 02:15:42 Idle
Jill:secondary_volume Jack:primary_volume   Source    00:50:31 Idle

Jill*> snapmirror resync -S Jack:primary_volume Jill:secondary_volume
The resync base snapshot will be: dfm_backup_2008-03-19_23-25-03.sndb
These newer snapshots will be deleted from the destination:
  Jack(0084193458)_primary_volume.1
Are you sure you want to resync the volume? y
Volume secondary_volume will be briefly unavailable before coming back online.
Revert to resync base snapshot was successful.
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.

Jill*> snapmirror status
Snapmirror is on.

Source                Destination           State   Lag    Status
Jack:primary_volume   Jill:secondary_volume Snapmirrored 00:00:02 Idle

Jack> snapmirror status
Snapmirror is on.

Source                Destination           State   Lag    Status
Jack:primary_volume   Jill:secondary_volume Source    00:00:32 Idle

Jack>
```

3 GLOSSARY

- DFM: Data Fabric Manager
- DR: Disaster Recovery
- FCP: Fibre Channel Protocol
- iSCSI: Internet Small Computer System Interface
- LUN: Logical Unit Number
- SDU: SnapDrive for UNIX
- SDW: SnapDrive for Windows

4 CONCLUSION

Customers with DFM installed on Solaris and Linux can do a disaster recovery of their DFM data. In addition, customers who do not have Protection Manager licensed on their Windows DFM servers can also make use of the information in this document.

© 2008 NetApp. All rights reserved. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, DataFabric, Data ONTAP, SnapDrive, SnapManager, SnapMirror, SnapRestore, and Snapshot are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. Windows is a registered trademark of Microsoft Corporation. Oracle is a registered trademark of Oracle Corporation. SAP is a registered trademark of SAP AG. Solaris is a trademark of Sun Microsystems, Inc. UNIX is a registered trademark of The Open Group. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.