



Technical Report

Oracle Data Protection Handbook

Jim Lanson, Srinath Alapati, Darrin Chapman, NetApp
February 2010 | TR-3787 | Version 1.0.1

ABSTRACT

This document provides an overview of data protection methods, products, and sample scenarios for the Oracle® Database environment. This includes descriptions of commonly used Oracle and NetApp® data protection products. It does not cover all possible product combinations and scenarios but will be regularly updated as products evolve and new scenarios arise. We hope this document will enable field personnel (sales, systems engineers, and partners) to be able to intelligently discuss data protection strategies with Oracle customers and make appropriate recommendations.

TABLE OF CONTENTS

1	PURPOSE	4
2	AUDIENCE.....	4
3	OVERVIEW	4
4	ORACLE DATA PROTECTION PRODUCTS	4
4.1	RECOVERY MANAGER (RMAN).....	4
4.2	DATA GUARD	4
4.3	ACTIVE DATA GUARD	5
4.4	ORACLE FLASHBACK TECHNOLOGY	5
4.5	ORACLE REAL APPLICATION CLUSTERS	5
4.6	ORACLE SECURE BACKUP	5
5	NETAPP DATA PROTECTION PRODUCTS	6
5.1	OPEN SYSTEMS SNAPVAULT	6
5.2	SNAPVAULT	6
5.3	SNAPMIRROR	6
5.4	METROCLUSTER	6
5.5	NETAPP VTL.....	7
5.6	SNAPMANAGER FOR ORACLE	7
5.7	SNAPVALIDATOR FOR ORACLE	7
6	REAL-WORLD DATA PROTECTION SCENARIOS.....	8
7	SCENARIO 1: PUTTING DR DATA TO USE WITH QTREE SNAPMIRROR AND FLEXCLONE 9	
7.1	INSTALL AND CONFIGURE NETAPP STORAGE.....	10
7.2	INSTALL AND CONFIGURE DATABASE HOSTS	13
7.3	INSTALL AND CONFIGURE ORACLE	14
7.4	INSTALL AND CONFIGURE DATABASE HOT BACKUP SCRIPTS*.....	15
7.5	CONFIGURE SNAPMIRROR ON SOURCE AND DESTINATION.....	16
7.6	VERIFY PROPER BACKUP AND REPLICATION OPERATION.....	17
7.7	INSERT TEST RECORD INTO EXAMPLE DATABASE.....	18
7.8	PERFORM SNAPMIRROR UPDATE.....	19
7.9	CREATE FLEXCLONE VOLUME FROM SNAPMIRROR DESTINATION AND BRING INTO OPERATION.....	20
7.10	VERIFY EXISTENCE OF TEST RECORD	20
8	SCENARIO 2: PUTTING DR DATA TO USE WITH VOLUME SNAPMIRROR AND FLEXCLONE 23	
8.1	INSTALL AND CONFIGURE NETAPP STORAGE.....	24
8.2	INSTALL AND CONFIGURE DATABASE HOSTS	25

8.3	INSTALL AND CONFIGURE ORACLE	25
8.4	INSTALL AND CONFIGURE DATABASE HOT BACKUP SCRIPTS	25
8.5	CONFIGURE SNAPMIRROR ON SOURCE AND DESTINATION	26
8.6	VERIFY PROPER BACKUP AND REPLICATION OPERATION	27
8.7	INSERT TEST RECORD INTO EXAMPLE DATABASE	27
8.8	PERFORM SNAPMIRROR UPDATE	28
8.9	CREATE FLEXCLONE VOLUME ON THE SNAPMIRROR DESTINATION AND BRING INTO OPERATION	29
8.10	VERIFY EXISTENCE OF TEST RECORD	33
9	SCENARIO 3: MULTIUSE COPIES WITH SNAPMIRROR, SNAPMANAGER FOR ORACLE, AND FLEXCLONE	35
9.1	INSTALL AND CONFIGURE NETAPP STORAGE	36
9.2	INSTALL AND CONFIGURE DATABASE HOSTS	37
9.3	INSTALL AND CONFIGURE ORACLE	38
9.4	ADD HOSTS AND CONFIGURE APPROPRIATE RESOURCE POOLS UNDER PROTECTION MANAGER	38
9.5	INSTALL AND CONFIGURE SNAPDRIVE FOR UNIX	39
9.6	INSTALL AND CONFIGURE SNAPMANAGER FOR ORACLE 3.0 INCLUDING CREATION OF REPOSITORY	40
9.7	CREATE SMO PROFILE FOR MANAGING ORACLE DATABASE	42
9.8	INTEGRATION WITH PROTECTION MANAGER	43
9.9	VERIFY PROPER OPERATION	46
9.10	INSERT TEST RECORD INTO SAMPLE DATABASE	46
9.11	CREATE FLEXCLONE COPY FROM SNAPMIRROR DESTINATION AND BRING UP ALTERNATE SERVER ON CLONE	47
9.12	VERIFY TEST RECORD EXISTS IN DATABASE RUNNING ON CLONE	48
10	VERSION HISTORY	50

1 PURPOSE

The purpose of this document is to provide an overview of data protection methods, products, and sample scenarios for the Oracle Database environment. This will allow field personnel (sales, systems engineers, and partners) to be able to intelligently discuss data protection strategies with customers and make appropriate recommendations.

While the initial release of this document covers only one scenario and does not necessarily utilize all relevant products, this document will be regularly updated as products are added or updated and scenarios are added.

2 AUDIENCE

This document is targeted at those people responsible for discussing data protection with present or potential NetApp customers and for making recommendations regarding solutions to data protection challenges. This intended audience includes NetApp and partner sales representatives and systems engineers.

3 OVERVIEW

First you will find a brief description of the data protection products offered by Oracle, followed by a description of those comparable products offered by NetApp. It is important to remember that it is not always an either/or situation. There are many cases in which NetApp data protection products complement those offered by Oracle. Points of differentiation will be noted where applicable. The final section offers real-world customer scenarios in which one or more NetApp products are used to address some type of challenge or need.

4 ORACLE DATA PROTECTION PRODUCTS

4.1 RECOVERY MANAGER (RMAN)

Oracle Recovery Manager, a command-line and Enterprise Manager-based tool, is the method preferred by Oracle for efficiently backing up and recovering your Oracle Database. RMAN is designed to work intimately with the server, providing block-level corruption detection during backup and restore. RMAN optimizes performance and space consumption during backup with file multiplexing and backup set compression. It integrates with Oracle Secure Backup and third-party media management products for tape backup.

RMAN takes care of all underlying database procedures before and after backup or restore, freeing dependency on OS and SQL*Plus scripts. It provides a common interface for backup tasks across different host operating systems and offers features not available through user-managed methods, such as parallelization of backup/recovery data streams, a backup file retention policy, and a detailed history of all backups.

4.2 DATA GUARD

This provides the management, monitoring, and automation software infrastructure to create and maintain one or more synchronized standby databases to protect data from failures, disasters, errors, and corruptions.

Data Guard standby databases can be located at remote disaster recovery sites thousands of miles away from the production data center, or they may be located in the same city, on the same campus, or even in the same building. If the production database becomes unavailable because of a planned or an unplanned outage, Data Guard can switch any standby database to the production role, thus minimizing downtime and preventing data loss.

4.3 ACTIVE DATA GUARD

Oracle Active Data Guard, an option for Oracle Database 11g™ Enterprise Edition, enhances quality of service by off-loading resource-intensive workloads from your production database to one or more synchronized standby databases. Active Data Guard enables read-only access to a physical standby database for queries, real-time reporting, Web-based access, and so on *while continuously applying changes* received from the production database. Active Data Guard also eliminates the overhead of performing backups on production systems by enabling RMAN block-change tracking and fast incremental backups using a physical standby database.

4.4 ORACLE FLASHBACK TECHNOLOGY

Oracle Flashback technology provides a set of new features to view and rewind data back and forth in time. Flashback features offer the capability to query historical data, perform change analysis, and perform self-service repair to recover from logical corruptions while the database is online. With Oracle Flashback technology, you can indeed undo the past.

Oracle Database 9i™ introduced Flashback Query to provide a simple, powerful, and completely nondisruptive mechanism for recovering from human errors. It allows users to view the state of data at a point in time in the past without requiring any structural changes to the database.

Oracle Database 10g™ extended Flashback technology to provide fast and easy recovery at the database, table, row, and transaction level. Flashback technology revolutionizes recovery by operating just on the changed data. The time it takes to recover the error is now equal to the same amount of time it took to make the mistake. Oracle Database 10g Flashback technologies include Flashback Database, Flashback Table, Flashback Drop, Flashback Versions Query, and Flashback Transaction Query.

Flashback technology can just as easily be utilized for nonrepair purposes, such as historical auditing with Flashback Query and undoing test changes with Flashback Database. Oracle Database 11g introduces an innovative method to manage and query long-term historical data with Flashback Data Archive. This release also provides an easy, one-step transaction back-out operation with the new Flashback Transaction capability.

4.5 ORACLE REAL APPLICATION CLUSTERS

The Oracle Real Application Clusters (RAC) option supports the transparent deployment of a single database across a cluster of servers, providing fault tolerance from hardware failures or planned outages. Oracle RAC running on clusters can provide Oracle's highest level of capability in terms of availability, scalability, and low-cost computing. Oracle RAC supports mainstream business applications of all kinds. This includes OLTP, DSS, and Oracle's unique ability to effectively support mixed OLTP/DSS environments. This also includes popular packaged products such as SAP®, PeopleSoft, Siebel, and Oracle E*Business Suite, as well as custom applications.

Oracle RAC provides single-image installation and management. The DBA has a single point of control to install and manage a RAC cluster from the GUI interface or command line.

4.6 ORACLE SECURE BACKUP

Oracle Secure Backup (OSB) delivers unified data protection for heterogeneous file systems, network-attached storage (NAS), and the Oracle Database in complex distributed IT environments. You can centrally manage Oracle Secure Backup domains (environments) using a single console and common management interface across the spectrum of servers and NAS devices.

Beyond file system protection, Oracle Secure Backup has built-in Oracle Database integration, providing optimized backup/restore for the database. Oracle Secure Backup delivers centralized tape backup management for the entire IT environment:

- Oracle Database integration with Recovery Manager supporting Oracle Database 9i to Oracle Database 11g

- Optimized performance achieving approximately 25% to 40% faster backups than comparable media management utilities with up to 10% less CPU utilization
- File system data protection: UNIX®, Windows®, and Linux® servers
- NAS data protection leveraging the Network Data Management Protocol (NDMP)

With a highly scalable client/server architecture, Oracle Secure Backup leverages Secure Sockets Layer (SSL) technology for secure intradomain communication and two-way server authentication. Domain-wide data protection operations may be standardized through user-defined policies controlling backup/restore, security, and media. Policies and day-to-day activities are managed using the OSB Web tool, command line, or Oracle Enterprise Manager (EM) interface.

5 NETAPP DATA PROTECTION PRODUCTS

5.1 OPEN SYSTEMS SNAPVAULT

Open Systems SnapVault® (OSSV) leverages the block-level incremental backup technology found in SnapVault to protect Windows, Linux, UNIX, and VMware® systems running on mixed storage. OSSV increases performance and enables more frequent data protection by moving and creating backups from changed data blocks, not entire changed files. And since only changed blocks are stored, less capacity is needed.

For distributed organizations, this strategy of transmitting small amounts of data not only makes disk-based backup cost effective for remote offices; it also provides the option of backing up directly to a core data center for centralized management of data protection.

5.2 SNAPVAULT

With SnapVault, data protection occurs at the block level, copying only the data blocks that have changed since the last backup, not entire files. This enables backups to run more frequently and utilize less capacity because no redundant data must be moved or stored.

For distributed organizations this not only makes disk-based backup cost effective, it provides the option of backing up directly from remote facilities to a core data center, centralizing management and minimizing investment needs at the edge.

5.3 SNAPMIRROR

NetApp SnapMirror® software is a cost-effective, easy-to-use disaster recovery and data distribution solution. It replicates data at high speeds over LAN or WAN, providing you with high data availability and fast disaster recovery for your business-critical applications, including Microsoft® Exchange, Microsoft SQL Server®, and Oracle. By mirroring data to one or more NetApp storage systems and continually updating the mirrored data, your data is kept current, and you can use the mirrored data for multiple purposes. If your business is geographically dispersed and all locations need access to the same data set, such as training videos or CAD tools, you can use SnapMirror to distribute the same data to all locations. Employee productivity and efficiency are dramatically improved by allowing local access to current mirrored data.

5.4 METROCLUSTER

MetroCluster is a unique solution that combines array-based clustering with synchronous mirroring to deliver continuous availability and zero data loss at the lowest cost. As a self-contained solution at the array level, MetroCluster is able to transparently recover from failures so business-critical applications continue uninterrupted. It also eliminates repetitive change management activities to reduce the risk of human error and administrative overhead.

You can now benefit from new MetroCluster enhancements:

- Nondisruptive upgrades to minimize planned downtime

- Testing with VMware high availability to achieve continuous availability in a virtualized environment
- Integration with NetApp deduplication to gain additional storage efficiencies

Whether you have a single data center, a campus, or a metropolitan-wide environment, MetroCluster is a cost-effective solution that provides continuous data availability for your critical business environment.

5.5 NETAPP VTL

As data levels grow and backup windows shrink, traditional tape-based backup can't keep pace. To get ahead of business requirements for data protection, you need a solution that boosts performance and reliability without adding complexity. NetApp VTL can help.

Purpose built for data center needs, NetApp VTL with deduplication delivers an industry-leading combination of performance, scalability, and integration. NetApp VTL makes intelligent disk appear as tape, enabling:

- High-speed backup and restore
- Cost efficiency and scaling up to 10PB
- Faster, more efficient tape creation and disaster recovery
- Simplified management, implementation, and scaling

Because NetApp VTL is built on 100% NetApp technology and a common platform, you gain simplicity, investment protection, and enterprise support built on years of primary and secondary storage expertise.

5.6 SNAPMANAGER FOR ORACLE

Applications built on Oracle are essential for supporting your business, and they need to be available around the clock. Your administrators need tools that allow them to support your critical business processes based on Oracle more efficiently.

Automate and simplify your Oracle Database management with backup, recover, restore, and cloning features in our SnapManager[®] for Oracle (SMO). Leverage our technology stack to create near-instant and space-efficient Snapshot[™] copies and clones of your Oracle Databases. Integrate with native Oracle technologies and get complete automation of Oracle data management.

Use policies to simplify, standardize, and automate data protection. Increase backup frequency without impacting performance for higher data protection. Recover and restore a failed database to full production in minutes, regardless of database size.

Create complete database clones in seconds on primary storage or directly to your development and test environment. Use clones to engage in parallel QA, development, testing, and other processes and deploy applications faster than ever before.

5.7 SNAPVALIDATOR FOR ORACLE

Organizations rely on the integrity of Oracle data. With database servers and storage systems at opposite ends of the I/O path, data can be at risk at several points. Help protect your Oracle data with NetApp SnapValidator[®], which detects and prevents potential corruptions before they happen.

Avoid the cost of unplanned disruption and downtime associated with resolving data corruptions.

Get end-to-end protection with data that is checked from server to storage. Extend enterprise-class checksum capabilities to modular storage. Prevent unplanned downtime from data corruption and spend more time on other IT activities.

Add intelligence and awareness to your modular storage systems across multiple network protocols. Use our SnapValidator to comply with the Oracle HARD initiative.

6 REAL-WORLD DATA PROTECTION SCENARIOS

OVERVIEW

The following scenarios are based on actual data protection challenges encountered by Oracle customers and are provided as examples of how NetApp data protection products can be used either alone or in combination with other NetApp or Oracle data protection products. The solutions outlined are by no means intended as the only possible solution but rather as examples of how those challenges can be met and protection extended in an Oracle environment.

7 SCENARIO 1: PUTTING DR DATA TO USE WITH QTREE SNAPMIRROR AND FLEXCLONE

OVERVIEW

The scenario discussed in this section encompasses a disaster recovery setup, meaning data from the primary production site is replicated to a DR site. This DR site can be made immediately available in the event of a disaster such that users can be redirected to the DR data. During normal production operations, when the primary site is active, the DR site and therefore the DR infrastructure are idle and unused. In traditional deployments, two key challenges arise when customers want to utilize these idle DR resources. The first is that DR replication needs to be stopped in order to fully test DR copies or utilize the DR copies for database development, testing, and staging; the second challenge is that each test copy at the DR site takes up space, thus increasing the footprint.

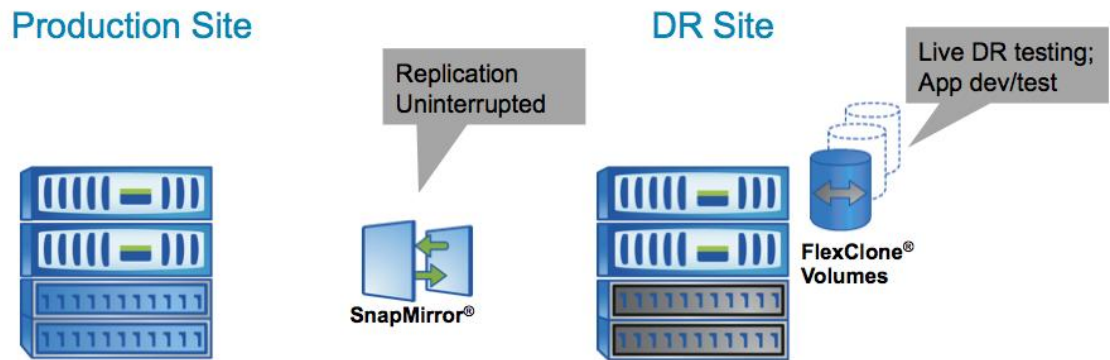


Figure 1) Typical SnapMirror disaster recovery implementation.

However, with SnapMirror and FlexClone, users can actively utilize the DR resources for nonproduction purposes such as DR testing, database development, testing, and staging while the primary production site is active and without interrupting replication of critical data to the DR site. Also, the copies at the DR site used for testing do not consume any additional space. These space-efficient copies start consuming space only when data is changed or when new data is created. Thus, customers can instantly create many numbers of space-efficient copies without increasing the footprint.

The ability to simultaneously utilize the standby DR resources while making sure the critical data is replicated is very appealing to many customers. In addition to this, the fact that the copies at the DR sites take only seconds to create with no additional storage is truly an incredible value proposition.

PRODUCTS USED

NetApp SnapMirror (qtree), FlexClone, Snapshot copies

SUMMARY OF IMPLEMENTATION STEPS

Table 7-1) Implementation steps.

Step	Description 1
1	Install and configure NetApp storage, including provisioning of necessary storage and installation of proper licenses.
2	Install and configure database hosts.
3	Install and configure Oracle.
4	Install and configure and schedule database hot backup scripts.
5	Configure SnapMirror.
6	Verify proper Snapshot copies from hot backup and operation of SnapMirror.
7	Insert test record into sample database.
8	Wait until next hot backup, then perform a SnapMirror update from latest hot backup.
9	Create FlexClone copy from SnapMirror destination and bring up alternate server on clone.
10	Verify test record exists in database running on clone.

7.1 INSTALL AND CONFIGURE NETAPP STORAGE

Two FAS3070 storage systems were used for storage of Oracle Database 11g log and control files. See Figure 7-2.

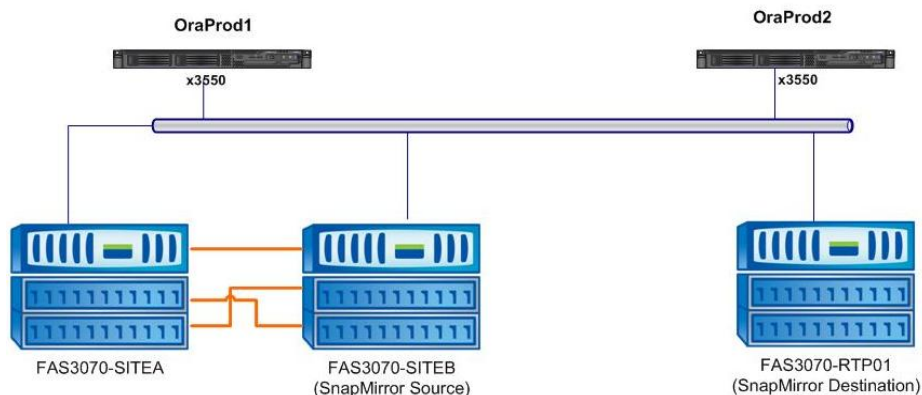


Figure 7-2) Test environment.

While the primary storage system for Oracle was FAS3070-SITEB, part of a NetApp MetroCluster system, MetroCluster functionality was not part of this scenario. It will be covered in a future scenario. FAS3070-RTP01 was used as the replication destination. Table 7-2 lists the specifications of the NetApp FAS systems used.

Table 7-2) Storage systems.

	FAS3070-SITEB	FAS3070-RTP01
Model	FAS3070	FAS3070
Operating System	Data ONTAP® 7.2.6.1	Data ONTAP 7.3.1
Licenses	NFS, SnapMirror	NFS, SnapMirror, FlexClone

SOURCE NETAPP STORAGE SYSTEM (FAS3070-SITEB)

Disk aggregates and volumes were created as shown in Figure 7-3. Two volumes, U03 and U04, were created to house the NFS storage for the Oracle environment.

Logical Disk Layout

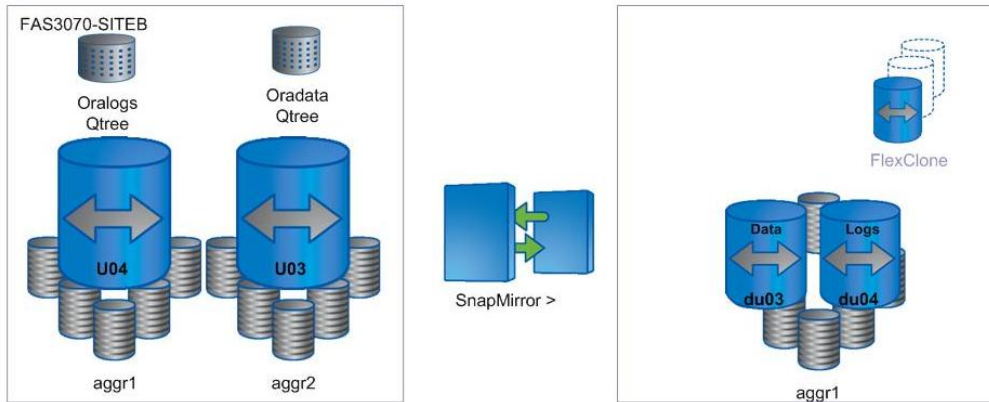


Figure 7-3) Disk layout.

The configuration of each volume is outlined in Figure 7-4.

```
FAS3070-SITEB> vol status u03
Volume State      Status      Options
u03 online        raid_dp, flex  nosnap=on, create_ucose=on,
                  mirrored      convert_ucose=on,
                  maxdirsiz=31457
Containing aggregate: 'aggr2'
FAS3070-SITEB> vol status u04
Volume State      Status      Options
u04 online        raid_dp, flex  nosnap=on, create_ucose=on,
                  mirrored      convert_ucose=on,
                  maxdirsiz=31457
Containing aggregate: 'aggr1'
FAS3070-SITEB>
```

Figure 7-4) Source volume status.

Within each volume one qtree was created for the log and control files (Oralogs) and another for the data files (Oradata) (Figure 7-5).

```
FAS3070-SITEB> qtree status
Volume  Tree      Style Oplocks  Status
-----
vol1    unix      enabled normal
vol0    unix      enabled normal
u04     unix      enabled normal
u04     Orlogs    unix      disabled normal
Cache SiteB    unix      enabled normal
lunvol   unix      enabled normal
cifsvol  unix      enabled normal
u03     unix      enabled normal
u03     Oradata    unix      enabled normal
FAS3070-SITEB>
```

Figure 7-5) Source qtree status.

Each qtree was added to the NFS exports file and exported for mounting by the Oracle server as shown in Figure 7-6.

```
FAS3070-SITEB> exportfs
/vol/u03          -sec=sys,rw,nosuid
/vol/u04/Oralogs  -sec=sys,rw
/vol/u04          -sec=sys,rw,nosuid
/vol/u03/Oradata  -sec=sys,rw
/vol/vol0/home    -sec=sys,rw,nosuid
/vol/vol0         -sec=sys,rw,anon=0,nosuid
/vol/vol1         -sec=sys,rw,nosuid
/vol/lunvol       -sec=sys,rw,nosuid
/vol/cifsvol      -sec=sys,rw,nosuid
/vol/Cache_SiteB -sec=sys,rw,nosuid
FAS3070-SITEB>
```

Figure 7-6) Source NFS exports.

CONFIGURE DESTINATION STORAGE

The destination FAS storage controller, FAS3070-RTP01, contains the aggregate (aggr1) that will host the SnapMirror replication destination, as shown in Figure 7-7.

```
fas3070-rtp01*> aggr status
      Aggr State      Status      Options
      aggr0 online     raid_dp, aggr  root
      aggr1 online     raid_dp, aggr
      aggr2 online     raid_dp, aggr
      DPRGrepo online  raid_dp, aggr
testdatasets_donotdelete online  raid_dp, aggr
      aggrLUN online   raid_dp, aggr  raidsize=6
      aggrFS online    raid_dp, aggr  raidsize=6
      aggraged online  raid_dp, aggr
fas3070-rtp01*>
```

Figure 7-7) Destination aggregate status.

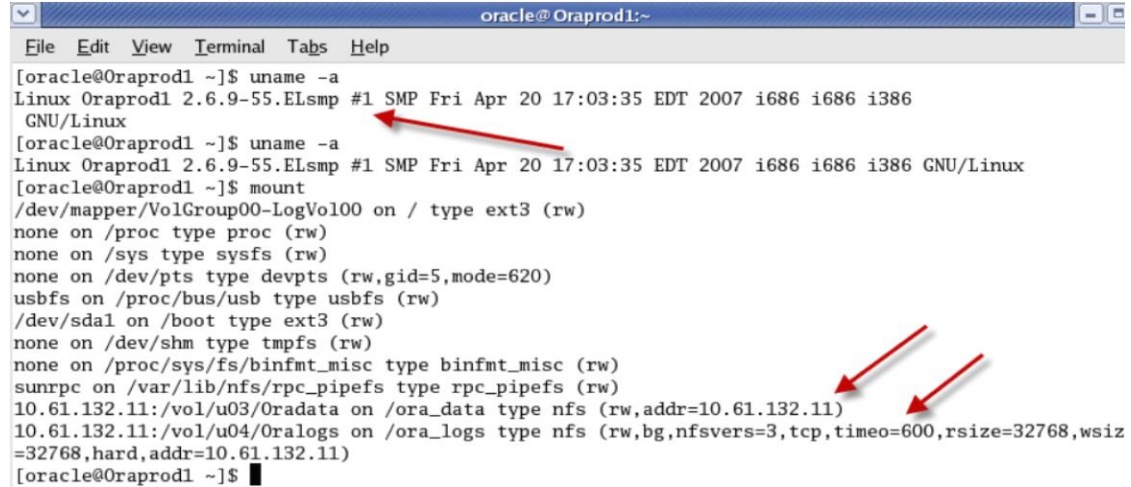
As shown in Figure 7-8, there are two destination volumes in this scenario. Volume du03 contains the replicated control and log files. Volume du04 contains the replicated data files themselves.

```
fas3070-rtp01*> vol status du03
      Volume State      Status      Options
      du03 online     raid_dp, flex
      Containing aggregate: 'aggr1'
fas3070-rtp01*> vol status du04
      Volume State      Status      Options
      du04 online     raid_dp, flex
      Containing aggregate: 'aggr1'
fas3070-rtp01*>
```

Figure 7-8) Destination volume status.

7.2 INSTALL AND CONFIGURE DATABASE HOSTS

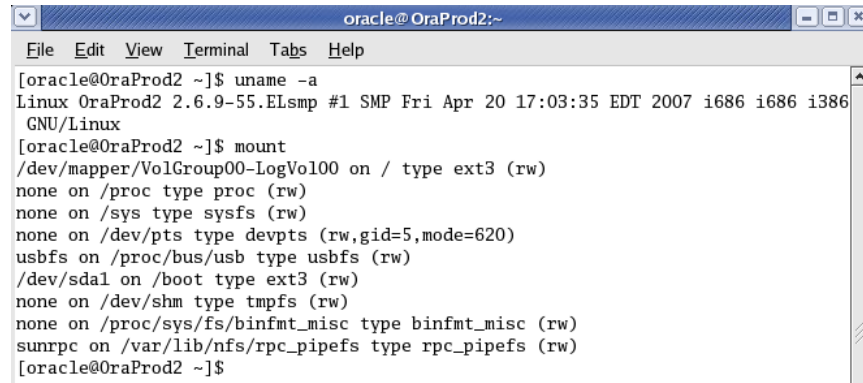
For the purposes of this test scenario, two IBM 3550 servers with 4GB of memory (Oraprod1 and Oraprod2) were used. Each used local direct-attached storage for the operating system and the Oracle binaries. Oraprod1 is the main production database server. It is running Red Hat Advanced server 4.0 (Figure 7-9).



```
oracle@Oraprod1:~  
File Edit View Terminal Tabs Help  
[oracle@Oraprod1 ~]$ uname -a  
Linux Oraprod1 2.6.9-55.ELsmp #1 SMP Fri Apr 20 17:03:35 EDT 2007 i686 i686 i386  
GNU/Linux  
[oracle@Oraprod1 ~]$ uname -a  
Linux Oraprod1 2.6.9-55.ELsmp #1 SMP Fri Apr 20 17:03:35 EDT 2007 i686 i686 i386 GNU/Linux  
[oracle@Oraprod1 ~]$ mount  
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)  
none on /proc type proc (rw)  
none on /sys type sysfs (rw)  
none on /dev/pts type devpts (rw,gid=5,mode=620)  
usbfs on /proc/bus/usb type usbfs (rw)  
/dev/sdal on /boot type ext3 (rw)  
none on /dev/shm type tmpfs (rw)  
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)  
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)  
10.61.132.11:/vol/u03/Oradata on /ora_data type nfs (rw,addr=10.61.132.11)  
10.61.132.11:/vol/u04/Oralogs on /ora_logs type nfs (rw,bg,nfsvers=3,tcp,timeo=600,rsiz  
=32768,hard,addr=10.61.132.11)  
[oracle@Oraprod1 ~]$
```

Figure 7-9) Main Oracle server (Oraprod1).

Oraprod2 (Figure 7-10) is the server that will be used to run the clone off the database replication destination.



```
oracle@OraProd2:~  
File Edit View Terminal Tabs Help  
[oracle@OraProd2 ~]$ uname -a  
Linux OraProd2 2.6.9-55.ELsmp #1 SMP Fri Apr 20 17:03:35 EDT 2007 i686 i686 i386  
GNU/Linux  
[oracle@OraProd2 ~]$ mount  
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)  
none on /proc type proc (rw)  
none on /sys type sysfs (rw)  
none on /dev/pts type devpts (rw,gid=5,mode=620)  
usbfs on /proc/bus/usb type usbfs (rw)  
/dev/sdal on /boot type ext3 (rw)  
none on /dev/shm type tmpfs (rw)  
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)  
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)  
[oracle@OraProd2 ~]$
```

Figure 7-10) Test Oracle server (Oraprod2).

7.3 INSTALL AND CONFIGURE ORACLE

Oracle Enterprise Edition was installed on both servers, Oraprod1 and Oraprod2, as shown in Figure 7-11.

```
SQL> select * from v$version;

BANNER
-----
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
PL/SQL Release 11.1.0.6.0 - Production
CORE      11.1.0.6.0      Production
TNS for Linux: Version 11.1.0.6.0 - Production
NLSRTL Version 11.1.0.6.0 - Production

SQL>
```

Figure 7-11) Oracle version information (Oraprod1 and Oraprod2).

ORAPROD1

The Oracle data files were placed on the NFS mounted volume called /Oradata. The log and control files were located on /Oralogs, also an NFS mounted volume (see Figure 7-12).

```
SQL> select name from v$controlfile;

NAME
-----
/ora_logs/orcl1/control01.ctl
/ora_logs/orcl1/control02.ctl
/ora_logs/orcl1/control03.ctl

SQL> select member from v$logfile;

MEMBER
-----
/ora_logs/orcl1/redo03.log
/ora_logs/orcl1/redo02.log
/ora_logs/orcl1/redo01.log

SQL>
```

```
SQL> select name from v$datafile;

NAME
-----
/ora_data/orcl1/system01.dbf
/ora_data/orcl1/sysaux01.dbf
/ora_data/orcl1/undotbs01.dbf
/ora_data/orcl1/users01.dbf
/ora_data/orcl1/example01.dbf
```

Figure 7-12) Database file locations.

For the purposes of verification of proper replication and cloning, the table scott.emp, part of the example database, was used (Figure 7-13).

```
SQL> desc scott.emp;

Name                               Null?    Type
-----
EMPNO                               NOT NULL NUMBER(4)
ENAME                               VARCHAR2(10)
JOB                                 VARCHAR2(9)
MGR                                 NUMBER(4)
HIREDATE                           DATE
SAL                                 NUMBER(7,2)
COMM                                NUMBER(7,2)
DEPTNO                             NUMBER(2)

SQL>
```

Figure 7-13) Employee table description.

For the purposes of later data comparison, Figure 7-14 shows the current contents of the table.

```
SQL> select ename from scott.emp;

ENAME
-----
SMITH
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS

ENAME
-----
JAMES
FORD
MILLER

14 rows selected.

SQL> █
```

Figure 7-14) Employee table (scott.emp) initial contents.

7.4 INSTALL AND CONFIGURE DATABASE HOT BACKUP SCRIPTS*

The following script (Figure 7-15) was set up in cron to run hourly and is used to generate hot backups on the Oracle production machine.

```
#!/bin.csh -f
# Place all of the critical tablespaces in
# hot backup mode.
$ORACLE_HOME/bin/sqlplus system/netapp1 @dobegin.sql

# rename and delete old snapshots
/usr/bin/ssh FAS3070-SITEB snap delete u03 oradata_03
/usr/bin/ssh FAS3070-SITEB snap rename u03 oradata_02 oradata_03
/usr/bin/ssh FAS3070-SITEB snap rename u03 oradata_01 oradata_02
/usr/bin/ssh FAS3070-SITEB snap rename u03 oradata_00 oradata_01

/usr/bin/ssh FAS3070-SITEB snap delete u04 oralogs_03
/usr/bin/ssh FAS3070-SITEB snap rename u04 oralogs_02 oralogs_03
/usr/bin/ssh FAS3070-SITEB snap rename u04 oralogs_01 oralogs_02
/usr/bin/ssh FAS3070-SITEB snap rename u04 oralogs_00 oralogs_01

#Take a new snapshot

/usr/bin/ssh FAS3070-SITEB snap create u03 oradata_00
/usr/bin/ssh FAS3070-SITEB snap create u04 oralogs_00

#Remove all affected tablespaces from
# hot backup mode
$ORACLE_HOME/bin/sqlplus system/netapp1 @doend.sql
..
```

Figure 7-15) Oracle hot backup script.

* Scripts are only provided as examples and are not supported by NetApp.

As you can see, it preserves the last four Snapshot copies for both volumes on the production FAS system. The dobegin and doend.sql scripts (Figure 7-16) were used to put Oracle into a hot backup mode.

```
[oracle@Oraprod1 ~]$ more dobegin.sql
CONNECT system/netapp1
SET FEEDBACK off
SET PAGESIZE 0
SPOOL begin.sql
Select
    'Alter tablespace ' ||
    tablespace_name ||
    ' Begin Backup;'
FROM
    dba_tablespaces;
Spool off
EXIT
[oracle@Oraprod1 ~]$
```

```
[oracle@Oraprod1 ~]$ more doend.sql
CONNECT system/netapp1
SET FEEDBACK off
SET PAGESIZE 0
SPOOL end.sql
Select
    'Alter tablespace ' ||
    tablespace_name ||
    ' End Backup;'
FROM
    dba_tablespaces;
Spool off
EXIT
[oracle@Oraprod1 ~]$
```

Figure 7-16) Contents of dobegin and doend SQL scripts.

These scripts are provided as is for an example only and are not represented as the only way to perform database-consistent backups.

7.5 CONFIGURE SNAPMIRROR ON SOURCE AND DESTINATION

As previously mentioned, NetApp SnapMirror was licensed on both the source and destination FAS controllers. In addition, the SnapMirror options are set on the source and destination, as shown in Figure 7-17.

```
FAS3070-SITEB> options snapmirror
snapmirror.access      all
snapmirror.checkip.enable off
snapmirror.delayed_acks.enable on
snapmirror.enable      on
snapmirror.log.enable  on
snapmirror.vbn_log_enable off
FAS3070-SITEB>
```

```
fas3070-rtp01*> options snapmirror
snapmirror.access      *
snapmirror.checkip.enable off
snapmirror.delayed_acks.enable on
snapmirror.enable      on
snapmirror.log.enable  on
snapmirror.readahead    0
snapmirror.readahead_freq 0
snapmirror.vbn_log_enable off
fas3070-rtp01*>
```

Figure 7-17) SnapMirror options on source and destination.

To verify proper configuration of SnapMirror, first view the status on the primary FAS controller, FAS3070-SITEB (Figure 7-18).


```

FAS3070-SITEB> snapmirror status -l
Snapmirror is on.

Source:                FAS3070-SITEB: /vol/u03/Oradata
Destination:           fas3070-rtp01: /vol/du03/Oradata
Status:                Idle
Progress:              -
State:                Source
Lag:                  140:13:45
Mirror Timestamp:      Fri Jun 19 18:27:55 GMT 2009
Base Snapshot:         fas3070-rtp01(0118042010)_du03_Oradata-src.0
Current Transfer Type: -
Current Transfer Error: -
Contents:              -
Last Transfer Type:    -
Last Transfer Size:    2834200 KB
Last Transfer Duration: 00:00:49
Last Transfer From:    -

Source:                FAS3070-SITEB: /vol/u04/Oralogs
Destination:           fas3070-rtp01: /vol/du04/Oralogs
Status:                Idle
Progress:              -
State:                Source
Lag:                  140:12:56
Mirror Timestamp:      Fri Jun 19 18:28:44 GMT 2009
Base Snapshot:         fas3070-rtp01(0118042010)_du04_Oralogs-src.0
Current Transfer Type: -
Current Transfer Error: -
Contents:              -
Last Transfer Type:    -
Last Transfer Size:    2176040 KB
Last Transfer Duration: 00:00:42
Last Transfer From:    -
FAS3070-SITEB>

```

Figure 7-18) SnapMirror status on source (FAS3070-SITEB).

Now look at the status on the destination FAS controller, FAS3070-rtp01 (Figure 7-19).

```

fas3070-rtp01*> snapmirror status /vol/du03/Oradata /vol/du04/Oralogs
Snapmirror is on.

```

Source	Destination	State	Lag	Status
fas3070-siteb: /vol/u03/Oradata	fas3070-rtp01: /vol/du03/Oradata	Snapmirrored	20:26:30	Idle
fas3070-siteb: /vol/u04/Oralogs	fas3070-rtp01: /vol/du04/Oralogs	Snapmirrored	161:53:12	Idle

```

fas3070-rtp01*>

```

Figure 7-19) SnapMirror status on destination (FAS3070-RTP01).

7.6 VERIFY PROPER BACKUP AND REPLICATION OPERATION

Next, make sure that the dohotbackup script is running correctly on the production host, Oraprod1. This is done by looking at the Snapshot copies on the production FAS controller, as shown in Figure 7-20. You will also notice the Snapshot copy made (and locked) by SnapMirror.

```

FAS3070-SITEB> snap list u03
Volume u03
working...

  %/used    %/total    date            name
  -----
  0% ( 0%)  0% ( 0%)  Jul 09 07:12   oradata_00
  0% ( 0%)  0% ( 0%)  Jul 09 07:00   fas3070-rtp01(0118042010)_du03_Oradata-src
.17 (snapmirror)
  0% ( 0%)  0% ( 0%)  Jul 09 06:12   oradata_01
  1% ( 0%)  0% ( 0%)  Jul 09 05:12   oradata_02
  1% ( 0%)  0% ( 0%)  Jul 09 04:12   oradata_03
FAS3070-SITEB> date
Thu Jul 9 07:12:47 EDT 2009
FAS3070-SITEB> snap list u04
Volume u04
working...

  %/used    %/total    date            name
  -----
  0% ( 0%)  0% ( 0%)  Jul 09 07:12   oralogs_00
  0% ( 0%)  0% ( 0%)  Jul 09 07:00   fas3070-rtp01(0118042010)_du04_Oralogs-src
.16 (snapmirror)
  0% ( 0%)  0% ( 0%)  Jul 09 06:12   oralogs_01
  1% ( 0%)  0% ( 0%)  Jul 09 05:12   oralogs_02
  1% ( 0%)  0% ( 0%)  Jul 09 04:12   oralogs_03
  6% ( 5%)  1% ( 0%)  Jul 08 13:39   fas3070-rtp01(0118042010)_u04_dest.1 (snap
mirror)
FAS3070-SITEB>

```

Figure 7-20) Snap list on FAS3070-SITEB.

On the destination FAS controller, check the SnapMirror status of those destination volumes to verify proper operation.

```
fas3070-rtp01*> snapmirror status /vol/du03/Oradata /vol/du04/Oralogs
Snapmirror is on.
Source                               Destination                               State      Lag      Status
fas3070-siteb: /vol/u03/Oradata      fas3070-rtp01: /vol/du03/Oradata      Snapmirrored  00:03:35  Idle
fas3070-siteb: /vol/u04/Oralogs      fas3070-rtp01: /vol/du04/Oralogs      Snapmirrored  00:03:02  Idle
fas3070-rtp01*>
```

Figure 7-21) SnapMirror status on FAS3070-RTP01.

7.7 INSERT TEST RECORD INTO EXAMPLE DATABASE

In this next phase of testing, we are going to insert a record into one of the tables on the production Oracle server (scott.emp on Oraprod1) and make it available to a FlexClone copy of the destination. This will provide proper operation of replication and cloning. First we will record the initial contents of the employee table, as shown in Figure 7-22.

```
SQL> select empno, ename from scott.emp;

EMPNO ENAME
-----
7369 SMITH
7499 ALLEN
7521 WARD
7566 JONES
7654 MARTIN
7698 BLAKE
7782 CLARK
7788 SCOTT
7839 KING
7844 TURNER
7876 ADAMS

EMPNO ENAME
-----
7900 JAMES
7902 FORD
7934 MILLER
```

Figure 7-22) Initial contents of employee table.

A record was inserted into the production database, as shown in Figure 7-23.

```
SQL> insert into scott.emp values
2 (3406,'Lanson','TME',1234,'05-JUL-2002',1600,500,30);

1 row created.

SQL> select empno, ename from scott.emp;

EMPNO ENAME
-----
3406 Lanson
7369 SMITH
7499 ALLEN
7521 WARD
7566 JONES
7654 MARTIN
7698 BLAKE
7782 CLARK
7788 SCOTT
7839 KING
7844 TURNER

EMPNO ENAME
-----
7876 ADAMS
7900 JAMES
7902 FORD
7934 MILLER
```

Figure 7-23) Employee table contents after insertion of record.

Now let's wait for the next scheduled database-consistent Snapshot copy (Figure 7-24).

```

10.61.132.11 - PuTTY
FAS3070-SITEB> snap list u03
Volume u03
working...

  %/used      %/total    date      name
  -----
  0% ( 0%)    0% ( 0%)    Jul 09 08:12 oradata_00
  0% ( 0%)    0% ( 0%)    Jul 09 08:00 fas3070-rtp01(0118042010)_du03_Oradata-src.18 {snapmirror}
  0% ( 0%)    0% ( 0%)    Jul 09 07:12 oradata_01
  1% ( 0%)    0% ( 0%)    Jul 09 06:12 oradata_02
  1% ( 0%)    0% ( 0%)    Jul 09 05:12 oradata_03
FAS3070-SITEB> snap list u04
Volume u04
working...

  %/used      %/total    date      name
  -----
  0% ( 0%)    0% ( 0%)    Jul 09 08:12 oralog_00
  0% ( 0%)    0% ( 0%)    Jul 09 08:00 fas3070-rtp01(0118042010)_du04_Oralog-src.17 {snapmirror}
  0% ( 0%)    0% ( 0%)    Jul 09 07:12 oralog_01
  0% ( 0%)    0% ( 0%)    Jul 09 06:12 oralog_02
  1% ( 0%)    0% ( 0%)    Jul 09 05:12 oralog_03
  6% ( 6%)    1% ( 1%)    Jul 08 13:39 fas3070-rtp01(0118042010)_u04_dest.1 {snapmirror}
FAS3070-SITEB>

```

Figure 7-24) First scheduled Snapshot copy after record insertion.

7.8 PERFORM SNAPMIRROR UPDATE

Now that we have chosen the database-consistent Snapshot copy with which to create the clone, we can either wait for the next scheduled SnapMirror update or perform a manual update as indicated in Figure 7-25.

```

fas3070-rtp01*> date
Thu Jul 9 08:21:30 EDT 2009
fas3070-rtp01*> snapmirror update -s oradata_00 -S fas3070-siteb:/vol/u03/Oradata /vol/du03/Oradata
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.
fas3070-rtp01*> snapmirror update -s oralog_00 -S fas3070-siteb:/vol/u04/Oralog /vol/du04/Oralog
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.
fas3070-rtp01*>

```

Figure 7-25) SnapMirror update using database-consistent Snapshot copy.

Note now, in Figure 7-26, the name of the resulting Snapshot copy used by SnapMirror.

```

fas3070-rtp01*> snap list du03
Volume du03
working....

  %/used      %/total    date      name
  -----
  0% ( 0%)    0% ( 0%)    Jul 09 08:24 fas3070-rtp01(0118042010)_du03_Oradata-dst.44 {busy,snapmirror}
  0% ( 0%)    0% ( 0%)    Jul 09 08:00 hourly.0
  1% ( 1%)    0% ( 0%)    Jul 09 00:00 nightly.0
  3% ( 2%)    0% ( 0%)    Jul 08 20:00 hourly.1
  4% ( 1%)    0% ( 0%)    Jul 08 16:00 hourly.2
  4% ( 0%)    0% ( 0%)    Jul 08 14:44 clone u03 testcl.1
  4% ( 0%)    0% ( 0%)    Jul 08 14:27 fas3070-rtp01(0118042010)_du03_Oradata-dst.4 {busy,vclone}
fas3070-rtp01*> snap list du04
Volume du04
working....

  %/used      %/total    date      name
  -----
  0% ( 0%)    0% ( 0%)    Jul 09 08:25 fas3070-rtp01(0118042010)_du04_Oralog-dst.42 {busy,snapmirror}
  0% ( 0%)    0% ( 0%)    Jul 09 08:00 hourly.0
  2% ( 2%)    0% ( 0%)    Jul 09 00:00 nightly.0
  5% ( 3%)    0% ( 0%)    Jul 08 20:00 hourly.1
  6% ( 1%)    1% ( 0%)    Jul 08 16:00 hourly.2
fas3070-rtp01*>

```

Figure 7-26) SnapMirror update using database-consistent Snapshot copy.

Now it's time to create the clone.

7.9 CREATE FLEXCLONE VOLUME FROM SNAPMIRROR DESTINATION AND BRING INTO OPERATION

The following is a description of the clone command that will be used to clone the volume containing the Oracle Database.

```
vol clone create clone_name [-s {volume|file|none}] -b parent_name
[parent_snap]
```

clone_name is the name of the FlexClone volume that you want to create.

-s {volume | file | none} specifies the space guarantee setting for the new FlexClone volume. If no value is specified, the FlexClone volume is given the same space guarantee setting as its parent.

parent_name is the name of the FlexVol® volume that you intend to clone.

parent_snap is the name of the base Snapshot copy of the parent FlexVol volume. If no name is specified, Data ONTAP creates a base Snapshot copy with the name *clone_cl_name_prefix.id*, where *cl_name_prefix* contains the up to 16 characters of the name of the new FlexClone volume and *id* is a unique digit identifier (for example, 1, 2, and so on).

```
fas3070-rtp01*> vol clone create u03 clone -b du03 fas3070-rtp01(0118042010) du03 Oradata-dst.44
Thu Jul 9 08:30:55 EDT [waf1.qtree.qsmBreak.base:error]: Breaking snapmirrored qtree 1 in volume u03_clone: base
no longer exists. Use snapmirror resync or initialize to re-establish the snapmirror.
Thu Jul 9 08:30:55 EDT [waf1.volume.clone.created:info]: Volume clone u03_clone of volume du03 was created succes
Creation of clone volume 'u03_clone' has completed.
fas3070-rtp01*> vol clone create u04 clone -b du04 fas3070-rtp01(0118042010) du04 Orlogs-dst.42
Thu Jul 9 08:31:39 EDT [waf1.qtree.qsmBreak.base:error]: Breaking snapmirrored qtree 1 in volume u04_clone: base
no longer exists. Use snapmirror resync or initialize to re-establish the snapmirror.
Creation of clone volume 'u04_clone' has completed.
fas3070-rtp01*>
```

Figure 7-27) Creation of clones of the Oracle Database volumes.

You will see in the messages above a reference to the breaking of a SnapMirror relationship. This is for the clone, not the original volume/qtree. That replication relationship for the primary volume continues uninterrupted.

Now, on the other Oracle server, Oraprod2, let's check to make sure the new clone volumes are mounted. Note that if a clone is made of a volume containing an NFS exported qtree, it will automatically be added to the exports file.

```
[root@OraProd2 oracle]# mount
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sdal on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
10.61.132.70:/vol/u03_clone/Oradata on /ora_data type nfs (rw,addr=10.61.132.70)
10.61.132.70:/vol/u04_clone/Oralogs on /ora_logs type nfs (rw,addr=10.61.132.70)
[root@OraProd2 oracle]#
```

Figure 7-28) Mounted file systems on Oraprod2.

7.10 VERIFY EXISTENCE OF TEST RECORD

The next step is to start up Oracle from the newly cloned database. Then we can make sure the newly inserted record is there. Figure 7-29 shows that, on Oraprod2, after the instance is started, a select command is executed against the table to verify the existence of the inserted record.

```

SQL> startup
ORACLE instance started.

Total System Global Area 1707446272 bytes
Fixed Size                  1300380 bytes
Variable Size               1191184484 bytes
Database Buffers            503316480 bytes
Redo Buffers                 11644928 bytes
Database mounted.
Database opened.
SQL> select empno, ename from scott.emp;

  EMPNO ENAME
-----
  3406 Lanson
  7369 SMITH
  7499 ALLEN
  7521 WARD
  7566 JONES
  7654 MARTIN
  7698 BLAKE
  7782 CLARK
  7788 SCOTT
  7839 KING
  7844 TURNER

  EMPNO ENAME
-----
  7876 ADAMS
  7900 JAMES
  7902 FORD
  7934 MILLER

```

Figure 7-29) Startup and verification of test record.

The database started from the clone volumes can now be used for development, testing, and staging. Note that it is also possible to clone a FlexClone volume. Once the testing is complete, the FlexClone volume can either be destroyed or split. If the FlexClone volume is split, the volume will consume space. Therefore, make sure that you have sufficient space in the aggregate before splitting a clone.

Upon completion of database testing and destroying the clone, if you need to refresh the database from the production (primary) system repeat the steps as shown in Figures 7-25, 7-26 and 7-27.

SUMMARY AND BEST PRACTICES

A few key behaviors are different when creating FlexClone volumes on a qtree SnapMirror destination system.

- Qtree SnapMirror does not maintain the same Snapshot copies of the volume on the source and destination systems. Because of this characteristic, a FlexClone volume created from a Snapshot copy on the qtree SnapMirror destination does not cause a lock on that Snapshot copy on the source volume. Deleting that Snapshot copy on the source volume has no impact on the replication or the destination volume. Therefore, the advantage of qtree SnapMirror is that a FlexClone volume can live for a long time on the SnapMirror destination system without space implications on the source system.
- SnapMirror updates require a common SnapMirror Snapshot copy. Therefore, do not delete SnapMirror Snapshot copies on either the source or destination system.
- If a Snapshot copy is not specified when creating a FlexClone volume on the qtree SnapMirror destination volume, the `vol clone` command creates a new Snapshot copy on that volume.
- If a FlexClone volume is created using the qtree SnapMirror baseline Snapshot copy, the qtree in the FlexClone volume will be writable.
- If a FlexClone volume is created on the qtree SnapMirror destination volume without specifying a backing Snapshot copy for the clone creation, a separate SnapMirror relationship appears in the `snapmirror status` command output. Following is an example that demonstrates this behavior. In the example below, there is a qtree SnapMirror relationship between `fas1:/vol/vol1/qt1` and `fas2:/vol/vol4/qt1`.

Below is the `snapmirror status` output for the relationship:

```
fas1:/vol/vol1/qt1 fas2:/vol/vol4/qt1 Snapmirrored 2689:49:43
Transferring (294 MB done)
```

A FlexClone volume called `c2` is created from the parent volume `vol4`.

```
fas2> vol clone create c2 -b vol4
```

A new relationship for the FlexClone volume appears in the `snapmirror status` command output:

```
fas1:/vol/vol1/qt1 fas2:/vol/c2/qt1 Snapmirrored 2689:51:31 Idle
```

```
fas1:/vol/vol1/qt1 fas2:/vol/vol4/qt1 Snapmirrored 2689:51:31
Transferring (1232 MB done)
```

Note that the new FlexClone SnapMirror relationship does not impact the qtree SnapMirror relationship belonging to the parent volume.

The qtree in the FlexClone volume `c2` is read-only whereas the FlexClone volume itself is writable.

In order to make the qtree in the FlexClone volume writable, quiesce and break operations must be performed on the FlexClone volume relationship. The original qtree SnapMirror relationship remains unaffected.

```
fas2> snapmirror quiesce /vol/c2/qt1
```

```
fas2> snapmirror break /vol/c2/qt1
```

```
snapmirror break: Destination /vol/c2/qt1 is now writable.
```

- FlexClone volumes are backed by the Snapshot copy from which they are created. The backing Snapshot copy is hard-locked (with a busy tag) and therefore cannot be deleted. Once the FlexClone volume is destroyed, the lock is removed.
- If the FlexClone volume is split, the FlexClone volume becomes a normal flexible volume and therefore requires full space allocation, depending on the space guarantees. When the FlexClone volume is split, all existing Snapshot copies of the FlexClone volume are deleted. For more information, refer to the “Storage Management Guide” on NOW™ (NetApp on the Web).

8 SCENARIO 2: PUTTING DR DATA TO USE WITH VOLUME SNAPMIRROR AND FLEXCLONE

OVERVIEW

The scenario discussed in this section encompasses a disaster recovery setup, meaning data from the primary production site is replicated to a DR site. This DR site can be made immediately available in the event of a disaster such that users can be redirected to the DR data. During normal production operations, when the primary site is active, the DR site and therefore the DR infrastructure are idle and unused. In traditional deployments, two key challenges arise when customers want to utilize these idle DR resources. The first is that DR replication needs to be stopped in order to fully test DR copies or utilize the DR copies for database development, testing, and staging; the second challenge is that each test copy at the DR site takes up space, thus increasing the footprint.

However, with SnapMirror and FlexClone, users can actively utilize the DR resources for nonproduction purposes such as DR testing, database development, testing, and staging while the primary production site is active and without interrupting replication of critical data to the DR site. Also, the copies at the DR site used for testing do not consume any additional space. These space-efficient copies start consuming space only when data is changed or when new data is created. Thus, customers can instantly create many numbers of space-efficient copies without increasing the footprint.

The ability to simultaneously utilize the standby DR resources while making sure the critical data is replicated is very appealing to many customers. In addition to this, the fact that the copies at the DR sites take only seconds to create with no additional storage is truly an incredible value proposition.

Unlike qtree SnapMirror, volume SnapMirror maintains the same number of Snapshot copies on the production and DR storage systems. If a user or an application creates a Snapshot copy on the production system, a SnapMirror update will propagate that Snapshot copy to the DR system. Likewise, if a user deletes a Snapshot copy on the production system, a SnapMirror update will attempt to delete the corresponding Snapshot copy on the DR system. In essence, volume SnapMirror attempts to keep the production and DR systems as mirror images. Therefore, when using volume SnapMirror with FlexClone on the destination, extra care needs to be taken. These details and best practices are discussed in this scenario.

PRODUCTS USED

NetApp SnapMirror (volume), FlexClone, Snapshot copies

SUMMARY OF IMPLEMENTATION STEPS

Table 8-1) Implementation steps.

Step	Description1
1	Install and configure NetApp storage, provision necessary storage, and install proper licenses.
2	Install and configure database hosts.
3	Install and configure Oracle.
4	Install and configure and schedule database hot backup scripts.
5	Configure volume SnapMirror.
6	Verify proper Snapshot copies from hot backup and operation of SnapMirror.
7	Insert test record into sample database.
8	Wait until next hot backup, then perform a SnapMirror update from latest hot backup.
9	Create FlexClone copy from SnapMirror destination and bring up alternate server on clone.
10	Verify test record exists in database running on clone.

8.1 INSTALL AND CONFIGURE NETAPP STORAGE

Two FAS3070 storage systems were used for storage of Oracle Database 11g log and control files. See Figure 7-1.

While the primary storage system for Oracle was FAS3070-SITEB, part of a NetApp MetroCluster system, MetroCluster functionality was not part of this scenario. It will be covered in a future scenario. FAS3070-RTP01 was used as the replication destination. Table 7-2 lists the specifications of the NetApp FAS systems used.

SOURCE NETAPP STORAGE SYSTEM (FAS3070-SITEB)

Disk aggregates and volumes were created as shown in Figure 8-1. Two volumes, U03 and U04, were created to house the NFS storage for the Oracle environment.

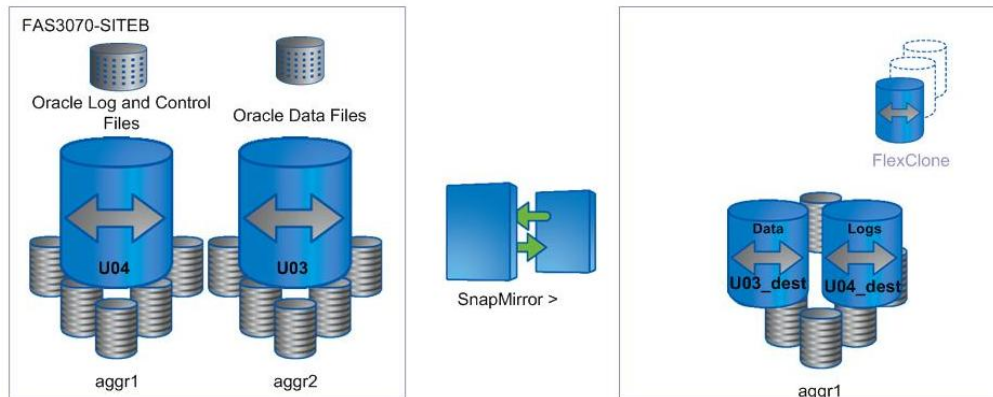


Figure 8-1) Disk layout.

Configuration of each volume is outlined in Figure 7-4.

Each volume was added to the NFS exports file and exported for mounting by the Oracle server as shown in Figure 8-2.

```
FAS3070-SITEB> exportfs
/vol/u03          -sec=sys,rw,anon=0
/vol/u04          -sec=sys,rw,anon=0
/vol/vol0/home    -sec=sys,rw
/vol/vol0         -sec=sys,rw,anon=0
/vol/vol1         -sec=sys,rw,nosuid
/vol/lunvol       -sec=sys,rw,nosuid
/vol/cifsvol      -sec=sys,rw,nosuid
/vol/Cache_SiteB -sec=sys,rw,nosuid
FAS3070-SITEB>
```

Figure 8-2) Source NFS exports.

CONFIGURE DESTINATION STORAGE

The destination FAS storage controller, FAS3070-RTP01, contains the aggregate (aggr1) that will host the SnapMirror replication destination, as shown in Figure 7-6.

As shown in Figure 8-3, there are two destination volumes in this scenario. Volume u03_dest contains the replicated control and log files. Volume u04_dest contains the replicated data files themselves.


```

fas3070-rt01> vol status u03_dest
      Volume State      Status      Options
      u03_dest online   raid_dp, flex  nosnap=on, snapmirrored=on,
                        snapmirrored  create_ucose=on,
                        read-only     convert_ucose=on,
                                       maxdirs=31457,
                                       fs_size_fixed=on

      Containing aggregate: 'aggr1'
fas3070-rt01> vol status u04_dest
      Volume State      Status      Options
      u04_dest online   raid_dp, flex  nosnap=on, snapmirrored=on,
                        snapmirrored  create_ucose=on,
                        read-only     convert_ucose=on,
                                       maxdirs=31457,
                                       fs_size_fixed=on

      Containing aggregate: 'aggr1'
fas3070-rt01*>

```

Figure 8-3) Destination volume status.

8.2 INSTALL AND CONFIGURE DATABASE HOSTS

For the purposes of this test scenario, two IBM 3550 servers with 4GB of memory (Oraprod1 and Oraprod2) were used. Each used local direct-attached storage for the operating system and the Oracle binaries. Oraprod1 is the main production database server. It is running Red Hat Advanced server 4.0 (Figure 8-4).

```

[oracle@Oraprod1 orcl1]$ uname -a
Linux Oraprod1 2.6.9-55.El.smp #1 SMP Fri Apr 20 17:03:35 EDT 2007 i686 i686 i386 GNU/Linux
[oracle@Oraprod1 orcl1]$ mount
/dev/mapper/VolGroup00-LogVol100 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sdal on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
10.61.132.11:/vol/u03 on /ora_data type nfs (rw,bg,nfsvers=3,tcp,timeo=600,rsz=32768,wsz=32768,hard,addr=10.61.132.11)
10.61.132.11:/vol/u04 on /ora_logs type nfs (rw,bg,nfsvers=3,tcp,timeo=600,rsz=32768,wsz=32768,hard,addr=10.61.132.11)

```

Figure 8-4) Main Oracle server (Oraprod1).

Oraprod2 (Figure 7-9) is the server that will be used to run the clone off the database replication.

8.3 INSTALL AND CONFIGURE ORACLE

Oracle Enterprise Edition was installed on both servers, Oraprod1 and Oraprod2, as previously shown in Figure 7-10.

ORAPROD1

The Oracle data files were placed on the NFS mounted volume called /Ora_data. The log and control files were located on /Ora_logs, also an NFS mounted volume (see Figure 7-11)

For purposes of verification of proper replication and cloning the table scott.emp, part of the example database, was used (Figure 7-12).

For purposes of later data comparison, Figure 7-13 shows the current contents of the table.

8.4 INSTALL AND CONFIGURE DATABASE HOT BACKUP SCRIPTS

The script shown in Figure 7-14 was set up in cron to run hourly and is used to generate hot backups on the Oracle production machine.

As you can see, it preserves the last four Snapshot copies for both volumes on the production FAS system. The dobegin.sql and doend.sql scripts (Figure 7-15) were used to put Oracle into a hot backup mode.

These scripts are provided as is for an example only and are not represented as the only way to perform database-consistent backups.

8.5 CONFIGURE SNAPMIRROR ON SOURCE AND DESTINATION

As previously mentioned, NetApp SnapMirror was licensed on both the source and destination FAS controllers. In addition, the SnapMirror options are set on the source and destination, as shown in Figure 7-16.

To verify proper configuration of SnapMirror, first view the status on the primary FAS controller, FAS3070-SITEB (Figure 8-5).

```
FAS3070-SITEB> snapmirror status -l
Snapmirror is on.

Source:                FAS3070-SITEB:u03
Destination:           fas3070-rtp01:u03_dest
Status:                Idle
Progress:              -
State:                 Source
Lag:                   00:34:07
Mirror Timestamp:      Thu Aug 13 08:30:20 EDT 2009
Base Snapshot:         fas3070-rtp01(0118042010)_u03_dest.44
Current Transfer Type: -
Current Transfer Error: -
Contents:              -
Last Transfer Type:    -
Last Transfer Size:    1766364 KB
Last Transfer Duration: 00:00:42
Last Transfer From:    -

Source:                FAS3070-SITEB:u04
Destination:           fas3070-rtp01:u04_dest
Status:                Idle
Progress:              -
State:                 Source
Lag:                   00:34:08
Mirror Timestamp:      Thu Aug 13 08:30:19 EDT 2009
Base Snapshot:         fas3070-rtp01(0118042010)_u04_dest.44
Current Transfer Type: -
Current Transfer Error: -
Contents:              -
Last Transfer Type:    -
Last Transfer Size:    1819624 KB
Last Transfer Duration: 00:00:42
Last Transfer From:    -
FAS3070-SITEB>
```

Figure 8-5) SnapMirror status on source (FAS3070-SITEB).

Now look at the status on the destination FAS controller, FAS3070-rtp01 (Figure 8-6).

```
fas3070-rtp01*> snapmirror status fas3070-siteb:u03
Snapmirror is on.
Source                Destination                State                Lag                Status
fas3070-siteb:u03     fas3070-rtp01:u03_dest  Snapmirrored        00:40:56          Idle
fas3070-rtp01*> snapmirror status fas3070-siteb:u04
Snapmirror is on.
Source                Destination                State                Lag                Status
fas3070-siteb:u04     fas3070-rtp01:u04_dest  Snapmirrored        00:41:04          Idle
fas3070-rtp01*>
```

Figure 8-6) SnapMirror status on destination (FAS3070-RTP01).

8.6 VERIFY PROPER BACKUP AND REPLICATION OPERATION

Next, make sure that the dohotbackup script is running correctly on the production host, Oraprod1. This is done by looking at the Snapshot copies on the production FAS controller, as shown in Figure 8-7. You will also notice the Snapshot copy made (and locked) by SnapMirror.

```
FAS3070-SITEB> snap list u03
Volume u03
working...

  %/used    %/total    date        name
  -----
  2% ( 2%)  0% ( 0%)   Aug 13 08:30 fas3070-rtp01{0118042010}_u03_dest.44 (snapmirror)
  2% ( 0%)  0% ( 0%)   Aug 13 08:19 oradata_00
  2% ( 0%)  0% ( 0%)   Aug 13 07:19 oradata_01
  2% ( 0%)  0% ( 0%)   Aug 13 06:19 oradata_02
  2% ( 0%)  0% ( 0%)   Aug 13 05:19 oradata_03
FAS3070-SITEB> date
Thu Aug 13 09:14:30 EDT 2009
FAS3070-SITEB> snap list u04
Volume u04
working...

  %/used    %/total    date        name
  -----
  1% ( 1%)  0% ( 0%)   Aug 13 08:30 fas3070-rtp01{0118042010}_u04_dest.44 (snapmirror)
  1% ( 0%)  0% ( 0%)   Aug 13 08:19 oralogs_00
  1% ( 0%)  0% ( 0%)   Aug 13 07:19 oralogs_01
  1% ( 0%)  0% ( 0%)   Aug 13 06:19 oralogs_02
  1% ( 0%)  0% ( 0%)   Aug 13 05:19 oralogs_03
FAS3070-SITEB>
```

Figure 8-7) Snap list on FAS3070-SITEB.

8.7 INSERT TEST RECORD INTO EXAMPLE DATABASE

In this next phase of testing, we are going to insert a record into one of the tables on the production Oracle server (scott.emp on Oraprod1) and make it available to a FlexClone copy of the destination. This will provide proper operation of replication and cloning. First we will record the initial contents of the employee table, as shown in Figure 8-8.


```
SQL> insert into scott.emp values (9696, 'Obama', 'Pres', 6666, '21-JAN-2009', 6000, 500, 30);

1 row created.

SQL> select empno, ename from scott.emp;

  EMPNO ENAME
  -----
    3406 Lanson
    7369 SMITH
    7499 ALLEN
    7521 WARD
    7566 JONES
    7654 MARTIN
    7698 BLAKE
    7782 CLARK
    7788 SCOTT
    7839 KING
    7844 TURNER

  EMPNO ENAME
  -----
    7876 ADAMS
    7900 JAMES
    7902 FORD
    7934 MILLER
    9696 Obama
```



16 rows selected.

Figure 8-8) Record insertion.

Now let's wait for the next scheduled database-consistent Snapshot copy.

```

FAS3070-SITEB> snap list u03
Volume u03
working...

  %/used      %/total    date           name
  -----
  0% ( 0%)    0% ( 0%)    Aug 13 16:19   oradata_00
  0% ( 0%)    0% ( 0%)    Aug 13 15:30   fas3070-rtp01(0118042010)_u03_dest.51 (snapmirror)
  1% ( 0%)    0% ( 0%)    Aug 13 15:19   oradata_01
  1% ( 0%)    0% ( 0%)    Aug 13 14:19   oradata_02
  1% ( 0%)    0% ( 0%)    Aug 13 13:19   oradata_03
FAS3070-SITEB> snap list u04
Volume u04
working...

  %/used      %/total    date           name
  -----
  0% ( 0%)    0% ( 0%)    Aug 13 16:19   oralsogs_00
  0% ( 0%)    0% ( 0%)    Aug 13 15:30   fas3070-rtp01(0118042010)_u04_dest.51 (snapmirror)
  0% ( 0%)    0% ( 0%)    Aug 13 15:19   oralsogs_01
  0% ( 0%)    0% ( 0%)    Aug 13 14:19   oralsogs_02
  0% ( 0%)    0% ( 0%)    Aug 13 13:19   oralsogs_03
FAS3070-SITEB>

```

Figure 8-9) First scheduled Snapshot copy after record insertion.

8.8 PERFORM SNAPMIRROR UPDATE

Now we can either wait for the next scheduled SnapMirror update or perform a manual update as shown in Figure 8-10. This will determine that the most recent database-consistent Snapshot copies, oradata_00 and oralsogs_00, have been replicated.

```

fas3070-rtp01*> snapmirror update u03_dest
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.
fas3070-rtp01*> snapmirror update u04_dest
Transfer started.
Monitor progress with 'snapmirror status' or the snapmirror log.
fas3070-rtp01*>

```

Figure 8-10) SnapMirror update using database-consistent Snapshot copy.

Note now, in Figure 8-11, the name of the resulting Snapshot copy used by SnapMirror. With volume SnapMirror, note that Oracle hot backups show up as the same name on both production and DR systems.

```

  %/used      %/total    date           name
  -----
  0% ( 0%)    0% ( 0%)    Aug 13 17:04   fas3070-rtp01(0118042010)_u03_dest.53
  0% ( 0%)    0% ( 0%)    Aug 13 16:30   fas3070-rtp01(0118042010)_u03_dest.52
  0% ( 0%)    0% ( 0%)    Aug 13 16:19   oradata_00
  1% ( 0%)    0% ( 0%)    Aug 13 15:19   oradata_01
  1% ( 0%)    0% ( 0%)    Aug 13 14:19   oradata_02
  1% ( 0%)    0% ( 0%)    Aug 13 13:19   oradata_03
fas3070-rtp01*> snap list u04_dest
Volume u04_dest
working...

  %/used      %/total    date           name
  -----
  0% ( 0%)    0% ( 0%)    Aug 13 17:04   fas3070-rtp01(0118042010)_u04_dest.53
  0% ( 0%)    0% ( 0%)    Aug 13 16:30   fas3070-rtp01(0118042010)_u04_dest.52
  0% ( 0%)    0% ( 0%)    Aug 13 16:19   oralsogs_00
  0% ( 0%)    0% ( 0%)    Aug 13 15:19   oralsogs_01
  0% ( 0%)    0% ( 0%)    Aug 13 14:19   oralsogs_02
  0% ( 0%)    0% ( 0%)    Aug 13 13:19   oralsogs_03
fas3070-rtp01*>

```

Figure 8-11) SnapMirror update using database-consistent Snapshot copy.

We can see now in 8-12A and 8-12B that the next SnapMirror update has been completed. The same hot backups exist on both production and DR systems.

```
FAS3070-SITEB> snap list u03
Volume u03
working...

  %/used    %/total    date          name
  -----
  0% ( 0%)  0% ( 0%)    Aug 20 14:23  fas3070-rtp01(0118042010)_u03_dest.151 (snapmirror)
  0% ( 0%)  0% ( 0%)    Aug 20 14:20  oradata_00
  0% ( 0%)  0% ( 0%)    Aug 20 13:20  oradata_01
  1% ( 0%)  0% ( 0%)    Aug 20 12:20  oradata_02
  1% ( 0%)  0% ( 0%)    Aug 20 11:20  oradata_03
FAS3070-SITEB> snap list u04
Volume u04
working...

  %/used    %/total    date          name
  -----
  0% ( 0%)  0% ( 0%)    Aug 20 14:23  fas3070-rtp01(0118042010)_u04_dest.149 (snapmirror)
  0% ( 0%)  0% ( 0%)    Aug 20 14:20  oralogs_00
  0% ( 0%)  0% ( 0%)    Aug 20 13:20  oralogs_01
  0% ( 0%)  0% ( 0%)    Aug 20 12:20  oralogs_02
  0% ( 0%)  0% ( 0%)    Aug 20 11:20  oralogs_03
FAS3070-SITEB> █
```

Source Before clone

Figure 8-12a) Snapshot copies on source volumes.

```
fas3070-rtp01*> snap list u03_dest
Volume u03_dest
working...

  %/used    %/total    date          name
  -----
  0% ( 0%)  0% ( 0%)    Aug 20 14:23  fas3070-rtp01(0118042010)_u03_dest.151
  0% ( 0%)  0% ( 0%)    Aug 20 14:20  oradata_00
  0% ( 0%)  0% ( 0%)    Aug 20 13:30  fas3070-rtp01(0118042010)_u03_dest.150
  0% ( 0%)  0% ( 0%)    Aug 20 13:20  oradata_01
  1% ( 0%)  0% ( 0%)    Aug 20 12:20  oradata_02
  1% ( 0%)  0% ( 0%)    Aug 20 11:20  oradata_03
fas3070-rtp01*> snap list u04_dest
Volume u04_dest
working...

  %/used    %/total    date          name
  -----
  0% ( 0%)  0% ( 0%)    Aug 20 14:23  fas3070-rtp01(0118042010)_u04_dest.149
  0% ( 0%)  0% ( 0%)    Aug 20 14:20  oralogs_00
  0% ( 0%)  0% ( 0%)    Aug 20 13:30  fas3070-rtp01(0118042010)_u04_dest.148
  0% ( 0%)  0% ( 0%)    Aug 20 13:20  oralogs_01
  0% ( 0%)  0% ( 0%)    Aug 20 12:20  oralogs_02
  0% ( 0%)  0% ( 0%)    Aug 20 11:20  oralogs_03
fas3070-rtp01*> █
```

Destination before clone

Figure 8-12b) Snapshot copies on destination volumes.

Now it's time to create the clone.

8.9 CREATE FLEXCLONE VOLUME ON THE SNAPMIRROR DESTINATION AND BRING INTO OPERATION

The following is a description of the clone command that is used to clone the volume containing the Oracle Database.

```
vol clone create clone_name [-s {volume|file|none}] -b parent_name
[parent_snap]
```

clone_name is the name of the FlexClone volume that you want to create.

-s {volume | file | none} specifies the space guarantee setting for the new FlexClone volume. If no value is specified, the FlexClone volume is given the same space guarantee setting as its parent.

parent_name is the name of the FlexVol volume that you intend to clone.

parent_snap is the name of the base Snapshot copy of the parent FlexVol volume. If no name is specified, Data ONTAP creates a base Snapshot copy with the name *clone_cl_name_prefix.id*, where *cl_name_prefix* contains the up to 16 characters of the name of the new FlexClone volume and *id* is a unique digit identifier (for example, 1, 2, and so on).

Figure 8-13 shows the commands used to create the cloned volumes.

```
fas3070-rtp01*> vol clone create u03_clone -b u03_dest oradata_00
Mon Aug 17 08:29:16 EDT [ems.engine.inputSuppress:info]: Event 'waf
es since Thu Aug 13 17:18:17 EDT 2009.
Creation of clone volume 'u03_clone' has completed.
fas3070-rtp01*> vol clone create u04_clone -b u04_dest oralog_00
Creation of clone volume 'u04_clone' has completed.
fas3070-rtp01*>
```

Figure 8-13) Creation of clones of the Oracle Database volumes.

Next, check the status (Figure 8-14) to make sure the cloned volumes are online and ready.

```
fas3070-rtp01*> vol status u03_clone
Volume State Status Options
u03_clone online raid_dp, flex nosnap=on, maxdirsize=83804
Clone, backed by volume 'u03_dest', snapshot 'fas3070-rtp01(0118042010)_u03_dest.53'
Containing aggregate: 'aggr1'
fas3070-rtp01*> vol status u04_clone
Volume State Status Options
u04_clone online raid_dp, flex nosnap=on, maxdirsize=83804
Clone, backed by volume 'u04_dest', snapshot 'fas3070-rtp01(0118042010)_u04_dest.53'
Containing aggregate: 'aggr1'
fas3070-rtp01*>
```

Figure 8-14) Status of new clones.

A BRIEF INTERLUDE

Before we continue with the verification, let's discuss the impact, both immediate and potential, that the cloning process has on our SnapMirror relationship. You can see in figure 8-15a that the source shows no indication (yet) of the clone just created, but the destination shown in figure 8-15b indicates that the *oradata_00* and *oratalogs_00* Snapshot copies are now “locked” by the clone, preventing deletion until the clone is destroyed. Figure 8-15a Snapshot copies are now locked by the clones.

```
FAS3070-SITEB> snap list u03
Volume u03
working...

%used %/total date name
-----
0% ( 0%) 0% ( 0%) Aug 20 14:23 fas3070-rtp01(0118042010)_u03_dest.151 (snapmirror)
0% ( 0%) 0% ( 0%) Aug 20 14:20 oradata_00
1% ( 0%) 0% ( 0%) Aug 20 13:20 oradata_01
1% ( 0%) 0% ( 0%) Aug 20 12:20 oradata_02
1% ( 0%) 0% ( 0%) Aug 20 11:20 oradata_03
FAS3070-SITEB> snap list u04
Volume u04
working...

%used %/total date name
-----
0% ( 0%) 0% ( 0%) Aug 20 14:23 fas3070-rtp01(0118042010)_u04_dest.149 (snapmirror)
0% ( 0%) 0% ( 0%) Aug 20 14:20 oralog_00
0% ( 0%) 0% ( 0%) Aug 20 13:20 oralog_01
0% ( 0%) 0% ( 0%) Aug 20 12:20 oralog_02
0% ( 0%) 0% ( 0%) Aug 20 11:20 oralog_03
FAS3070-SITEB>
```

Figure 8-15a) Snapshot copies are now locked by the clones.

```

fas3070-rtp01*> snap list u03_dest
Volume u03_dest
working...

```

%/used	%/total	date	name
0% (0%)	0% (0%)	Aug 20 14:23	fas3070-rtp01(0118042010)_u03_dest.151
0% (0%)	0% (0%)	Aug 20 14:20	oradata_00 (busy,snapmirror,vclone)
0% (0%)	0% (0%)	Aug 20 13:30	fas3070-rtp01(0118042010)_u03_dest.150
0% (0%)	0% (0%)	Aug 20 13:20	oradata_01
1% (0%)	0% (0%)	Aug 20 12:20	oradata_02
1% (0%)	0% (0%)	Aug 20 11:20	oradata_03

```

fas3070-rtp01*> snap list u04_dest
Volume u04_dest
working...

```

%/used	%/total	date	name
0% (0%)	0% (0%)	Aug 20 14:23	fas3070-rtp01(0118042010)_u04_dest.149
0% (0%)	0% (0%)	Aug 20 14:20	orlogs_00 (busy,snapmirror,vclone)
0% (0%)	0% (0%)	Aug 20 13:30	fas3070-rtp01(0118042010)_u04_dest.148
0% (0%)	0% (0%)	Aug 20 13:20	orlogs_01
0% (0%)	0% (0%)	Aug 20 12:20	orlogs_02
0% (0%)	0% (0%)	Aug 20 11:20	orlogs_03

```

fas3070-rtp01*>

```

Destination after clone

Figure 8-15b) Snapshot copies are now locked by the clones.

Now let's take a look at what happens after the *next* SnapMirror update.

```

FAS3070-SITEB> snap list u03
Volume u03
working...

```

%/used	%/total	date	name
0% (0%)	0% (0%)	Aug 20 14:30	fas3070-rtp01(0118042010)_u03_dest.152 (snapmirror)
0% (0%)	0% (0%)	Aug 20 14:20	oradata_00 (snapmirror)
1% (0%)	0% (0%)	Aug 20 13:20	oradata_01
1% (0%)	0% (0%)	Aug 20 12:20	oradata_02
1% (0%)	0% (0%)	Aug 20 11:20	oradata_03

```

FAS3070-SITEB> snap list u04
Volume u04
working...

```

%/used	%/total	date	name
0% (0%)	0% (0%)	Aug 20 14:30	fas3070-rtp01(0118042010)_u04_dest.150 (snapmirror)
0% (0%)	0% (0%)	Aug 20 14:20	orlogs_00 (snapmirror)
0% (0%)	0% (0%)	Aug 20 13:20	orlogs_01
0% (0%)	0% (0%)	Aug 20 12:20	orlogs_02
0% (0%)	0% (0%)	Aug 20 11:20	orlogs_03

Source after next SnapMirror update

Figure 8-16a) Snapshot copies are now locked on source by the clones.

```
fas3070-rtp01*> snap list u03_dest
Volume u03_dest
working...
```

%/used	%/total	date	name
0% (0%)	0% (0%)	Aug 20 14:30	fas3070-rtp01(0118042010)_u03_dest.152
0% (0%)	0% (0%)	Aug 20 14:23	fas3070-rtp01(0118042010)_u03_dest.151
0% (0%)	0% (0%)	Aug 20 14:20	oradata_00 (busy, snapmirror, vclone)
1% (0%)	0% (0%)	Aug 20 13:20	oradata_01
1% (0%)	0% (0%)	Aug 20 12:20	oradata_02
1% (0%)	0% (0%)	Aug 20 11:20	oradata_03

```
fas3070-rtp01*> snap list u04_dest
Volume u04_dest
working...
```

%/used	%/total	date	name
0% (0%)	0% (0%)	Aug 20 14:30	fas3070-rtp01(0118042010)_u04_dest.150
0% (0%)	0% (0%)	Aug 20 14:23	fas3070-rtp01(0118042010)_u04_dest.149
0% (0%)	0% (0%)	Aug 20 14:20	orlogs_00 (busy, snapmirror, vclone)
0% (0%)	0% (0%)	Aug 20 13:20	orlogs_01
0% (0%)	0% (0%)	Aug 20 12:20	orlogs_02
0% (0%)	0% (0%)	Aug 20 11:20	orlogs_03

```
fas3070-rtp01*>
```

Destination after next SnapMirror update

Figure 8-16b) Snapshot copies are now locked on destination by the clones.

So, what are these screenshots indicating?

First of all, when clones are created, the Snapshot copies they are based on, in this case, oradata_00 and orlogs_00, are now hard-locked as shown by the “busy, snapmirror” status. This protects them from deletion. They will remain locked until the clones are either split or destroyed. After the next SnapMirror update, it can be seen in Figure 8-16a that the Snapshot copies are now shown as soft-locked on the source, as shown by “snapmirror” status (not busy). As part of the SnapMirror update process, a “soft” lock is propagated from the destination to the source volume. This prevents Data ONTAP from deleting these soft-locked Snapshot copies. However, since these are not “hard” locked, a user or a script has the ability to delete the Snapshot copies on the source system. This behavior is intentional.

If one of these soft-locked Snapshot copies is deleted, a following volume SnapMirror update will attempt to delete the corresponding Snapshot copy on the destination but fails to delete it because there is a clone dependent on that Snapshot and it is hard-locked. Therefore, SnapMirror updates for that volume will stop, and the risk of data loss is increased should a failover to the destination become necessary. SnapMirror updates can be resumed by destroying or splitting the clone. However, there are ways to prevent this scenario to begin with.

A review of the Oracle hot backup script in Figure 14 shows that, when it runs, it first deletes the oldest database-consistent backup. If a clone of this Snapshot copy (oradata_03 or orlogs_03) was created at the destination, then subsequent SnapMirror updates will fail. As previously stated, this situation will remain until the clone is split or destroyed. Using this script means that if the clone was created from the most frequent database-consistent Snapshot copy (oradata_00 or orlogs_00), then, based on hourly backups, SnapMirror updates will begin failing in about four hours. The question then arises as to what can be done to avoid this situation.

It really depends on how long the clones are needed.

If the clones on the destination are not needed for more than four hours, nothing needs to be done. The clone on the destination would have been deleted within four hours and, therefore, SnapMirror updates would continue as usual.

If they are needed for more than four hours, one of the following could be done:

1. Modify the script to retain more hot backups so the one chosen for the clone will not be deleted for a longer period of time. The script can be modified not to delete a Snapshot copy if there is a lock next to it; for example: oradata_00 (SnapMirror).

2. Modify the script to increase the time between backups.
3. Run a special “out of schedule” hot backup specifically for the clone.

Now, let’s get back to our verification test.

On the other Oracle server, Oraprod2, let’s check to make sure the new clone volumes are mounted. Note that if a clone is made of a volume containing an NFS exported volume, it will automatically be added to the exports file.

```
root@Oraprod2 ora_data1# mount
/dev/mapper/VolGroup00-LogVol100 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sdal on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
0.61.132.70:/vol/u03_clone on /ora_data type nfs (rw,addr=10.61.132.70)
0.61.132.70:/vol/u04_clone on /ora_logs type nfs (rw,addr=10.61.132.70)
root@Oraprod2 ora_data1#
```

Figure 468-16) Mounted file systems on Oraprod2.

8.10 VERIFY EXISTENCE OF TEST RECORD

The next step is to start up Oracle from the newly cloned database. Then we can check and make sure the newly inserted record is there. On Oraprod2, you can see in Figure 8-17 that, after the instance is started, a select command is executed against the table to verify the existence of the inserted record.

```
ORACLE instance started.

Total System Global Area 1707446272 bytes
Fixed Size 1300380 bytes
Variable Size 1207961700 bytes
Database Buffers 486539264 bytes
Redo Buffers 11644928 bytes
Database mounted.
Database opened.
SQL> select empno, ename from scott.emp;
```

EMPNO	ENAME
3406	Lanson
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER

7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER
9696	Obama




Figure 8-17) Startup and verification of test record.

The database started from the clone volumes can now be used for development, testing, and staging. Note that it is also possible to clone a FlexClone volume. Once the testing is complete, the FlexClone volume can either be destroyed or split. If the FlexClone volume is split, the volume will consume space. Therefore, make sure that you have sufficient space in the aggregate before splitting a clone.

Upon completion of database testing and removal of the cloned volume, if you need to refresh the database from the production (primary) system, repeat steps as shown in Figures 8-13, 8-14, 8-15.

SUMMARY AND BEST PRACTICES

Volume SnapMirror maintains the same number of Snapshot copies on both source (production) and destination (DR) systems. In other words, the production and DR volumes look like mirror images. Therefore, any creation or deleting of Snapshot copies on the source (production) system will cause SnapMirror to create and delete corresponding Snapshot copies on the destination (DR) system. If you or a script unknowingly deletes a Snapshot copy on the source system, a subsequent SnapMirror update attempts to delete that Snapshot copy on the destination system. However, if that Snapshot copy is “locked” on the destination system, such as when a FlexClone volume is created using that specific Snapshot copy on the destination system, SnapMirror is unable to delete the Snapshot copy and subsequent updates will fail until the FlexClone volume is destroyed or split.

The following best practices will prevent scenarios such as failed updates due to accidental Snapshot deletions. Use these guidelines when creating FlexClone volumes on a volume SnapMirror destination system:

1. When creating a FlexClone volume on the destination system, do not use a SnapMirror Snapshot copy. Even though a SnapMirror update will propagate a lock to the source Snapshot copy after that lock exists on the destination volume, and therefore will not delete the Snapshot copy, it is still a good idea not to use the SnapMirror Snapshot copy to create a FlexClone volume on the destination system. This is because if a SnapMirror update was in progress during the clone creation, one could run into a situation in which that Snapshot copy is deleted on the source by SnapMirror while the FlexClone volume is being created.
2. Use a user- (or script- or application-) created Snapshot copy to create FlexClone volumes on the destination system. Make sure not to delete or roll the Snapshot copy that has a corresponding FlexClone volume on the destination. As mentioned earlier, SnapMirror will propagate and display a lock next to the Snapshot copy to the source system if there is a FlexClone volume on the destination. If using a script, the script must be able to check for the existence of locks in order to avoid deleting them.
3. Do not use hourly, daily, or monthly Snapshot copies. When these Snapshot copies expire, SnapMirror updates will fail.
4. Exercise caution when using SnapDrive® rolling Snapshot copies. If a SnapDrive rolling Snapshot copy is to be used for FlexClone volume creation, first rename the Snapshot copy, then perform a SnapMirror update, and then create the FlexClone volume on the destination using the renamed Snapshot copy.
5. If auto_delete is enabled for blocks environment, it can potentially delete a Snapshot copy on the source system that has a corresponding FlexClone volume on the destination. This will cause SnapMirror updates to fail.
6. If SnapMirror updates fail due to a deletion of the Snapshot copy on the source and that Snapshot copy is locked on the destination because of a FlexClone volume, destroying or splitting the FlexClone volume will release the lock and SnapMirror updates can be resumed.
7. If you intend to keep the FlexClone volume on a volume SnapMirror destination system for a long period of time (such as months), since that Snapshot copy is locked on the source system you need to plan for sufficient storage arising due to data growth on the source volume until that FlexClone volume is no longer needed.

9 SCENARIO 3: MULTIUSE COPIES WITH SNAPMIRROR, SNAPMANAGER FOR ORACLE, AND FLEXCLONE

OVERVIEW

The scenario discussed in this section encompasses an Oracle backup policy with a disaster recovery setup, meaning data from the primary production site is backed up on a regular basis and is replicated to a DR site. This DR site can be made immediately available in the event of a disaster such that users can be redirected to the DR data.

During normal production operations when the primary site is active, the DR site and therefore the DR infrastructure are idle and unused. In traditional deployments, two key challenges arise when customers want to utilize these idle DR resources. The first is that DR replication needs to be stopped in order to fully test DR copies or utilize the DR copies for database development, testing, and staging; and the second challenge is that each test copy at the DR site takes up space, thus increasing the footprint.

However, with SnapMirror and FlexClone, users can actively utilize the DR resources for nonproduction purposes such as DR testing, database development, testing, and staging while the primary production site is active and without interrupting replication of critical data to the DR site. Also, the copies at the DR site used for testing do not consume any additional space. These space-efficient copies start consuming space only when data is changed or when new data is created. Thus, customers can instantly create many numbers of space-efficient copies without increasing the footprint.

In this scenario, SnapManager for Oracle 3.0, along with NetApp Protection Manager, was used to facilitate not only database-consistent backups but also the mirroring necessary for disaster recovery. It also enables easy creation of space-efficient, nondisruptive clones. The ability to simultaneously utilize the standby DR resources while making sure the critical data is replicated is very appealing to many customers. In addition to this, the fact that the copies at the DR sites take only seconds to create with no additional storage is truly an incredible value proposition.

PRODUCTS USED

NetApp SnapMirror (volume), FlexClone, Snapshot copies, SnapManager for Oracle, SnapDrive for UNIX, Protection Manager

PREIMPLEMENTATION

- Make sure there is an available installation of both NetApp Operations Manager and Protection Manager. If there is not, install both in accordance with the requirements of each product and the guidelines for installation and configuration contained in the documentation available on the NetApp NOW site.
- Make sure there is an available Oracle instance to contain the SnapManager for Oracle repository. This repository cannot be on the same server as the managed databases.
- Have all logins and permissions set up in order to make configuration changes to the above products.

SUMMARY OF IMPLEMENTATION STEPS

Table 9-1) Implementation steps.

Step	Description 1
1	Install and configure NetApp storage, including provisioning of necessary storage and installation of proper licenses.
2	Install and configure database hosts.
3	Install and configure Oracle.
4	Add hosts and configure appropriate resource groups under Protection Manager.

5	Install and configure SnapDrive for UNIX.
6	Install and configure SnapManager for Oracle 3.0, including creation of repository.
7	Create SMO profile for managing Oracle Database.
8	Integrate with Protection Manager for DR mirroring.
9	Insert test record into sample database.
10	Wait until next hot backup, then perform a SnapMirror update from the latest hot backup.
11	Create FlexClone copy from SnapMirror destination and bring up alternate server on clone.
12	Verify test record exists in database running on clone.

9.1 INSTALL AND CONFIGURE NETAPP STORAGE

Two FAS3070 storage systems were used for storage of Oracle Database 11g log and control files. See Figure 7-1.

While the primary storage system for Oracle was FAS3070-SITEB, part of a NetApp MetroCluster system, MetroCluster functionality was not part of this scenario. It will be covered in a future scenario. FAS3070-RTP01 was used as the replication destination. Table 7-2 lists the specifications of the NetApp FAS systems used.

SOURCE NETAPP STORAGE SYSTEM (FAS3070-SITEB)

Disk aggregates and volumes were created as shown in Figure 8-1. Three volumes, U02, U03, and U04, were created to house the NFS storage for the Oracle environment.

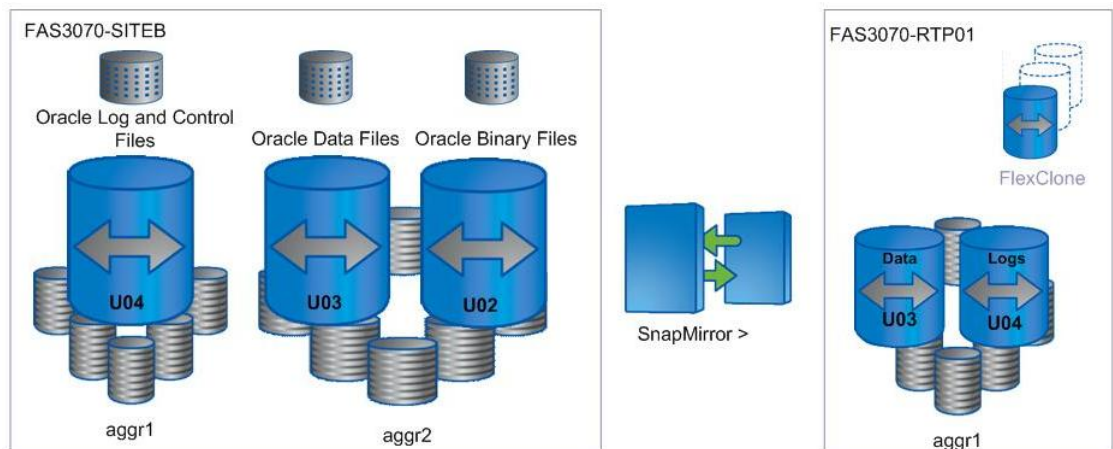


Figure 8-1) Disk layout.

The configuration of each volume is outlined in Figure 7-4.

Each volume was added to the NFS exports file and exported for mounting by the Oracle server as shown in Figure 8-5.

```
FAS3070-SITEB# exportfs
/vol/u02 -sec=sys,rw,anon=0
/vol/u03 -sec=sys,rw,anon=0
/vol/u04 -sec=sys,rw,anon=0
/vol/vol10/home -sec=sys,rw
/vol/vol10 -sec=sys,rw,anon=0
/vol/vol11 -sec=sys,rw,nosuid
/vol/vol12 -sec=sys,rw,nosuid
/vol/lunvol -sec=sys,rw,nosuid
/vol/cifsvol -sec=sys,rw,nosuid
/vol/Cache_SiteB -sec=sys,rw,nosuid
FAS3070-SITEB#
```

Figure 8-2) Source NFS exports.

CONFIGURE DESTINATION STORAGE

The destination FAS storage controller, FAS3070-RTP01, contains the aggregate (aggr1) that will host the SnapMirror replication destination, as shown in Figure 7-6.

As shown in Figure 9-3, there are two destination volumes in this scenario. Volume u03 contains the replicated control and log files. Volume u04 contains the replicated data files themselves.

```
Fas3070-rtp01> vol status du03
Volume State      Status      Options
du03 online      raid dp, flex  nosnap=on, snapmirrored=on,
                        snapmirrored  create_ucose=on,
                        read-only     convert_ucose=on,
                        maxdirsize=31457,
                        fs_size_fixed=on

Containing aggregate: 'aggr1'
Fas3070-rtp01> vol status du04
Volume State      Status      Options
du04 online      raid dp, flex  nosnap=on, snapmirrored=on,
                        snapmirrored  create_ucose=on,
                        read-only     convert_ucose=on,
                        maxdirsize=31457,
                        fs_size_fixed=on

Containing aggregate: 'aggr1'
```

Figure 9-3) Destination volume status.

9.2 INSTALL AND CONFIGURE DATABASE HOSTS

For purposes of this test scenario, two IBM 3550 servers with 4GB of memory (Oraprod1 and Oraprod2) were used. Both used local direct-attached storage for the operating system. Oraprod1 had its binaries stored on an NFS mounted volume in addition to its data and log files. Oraprod2 had its binaries on direct-attached storage. Oraprod1 is the main production database server. It is running Red Hat Advanced server 4.0 (Figure 9-4).

```
[oracle@Oraprod1 ~]$ uname -a
Linux Oraprod1 2.6.9-55.ELsmp #1 SMP Fri Apr 20 17:03:35 EDT 2007 i686 i686 i386 GNU/Linux
[oracle@Oraprod1 ~]$ mount
/dev/mapper/VolGroup00-LogVol100 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sdal on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
10.61.132.11:/vol/u02 on /home/oracle/app type nfs (rw,bg,nfsvers=3,tcp,noac,timeo=600,rsz=32768,wsz=32768,hard,addr=10.61.132.11)
10.61.132.11:/vol/u03 on /ora_data type nfs (rw,bg,nfsvers=3,tcp,noac,timeo=600,rsz=32768,wsz=32768,hard,addr=10.61.132.11)
10.61.132.11:/vol/u04 on /ora_logs type nfs (rw,bg,nfsvers=3,tcp,noac,timeo=600,rsz=32768,wsz=32768,hard,addr=10.61.132.11)
```

Figure 9-4) Main Oracle server (Oraprod1).

Oraprod2 (Figure 7-9) is the server that will be used to run the clone from the database replication.

9.3 INSTALL AND CONFIGURE ORACLE

Oracle Enterprise Edition was installed on both servers, Oraprod1 and Oraprod2, as previously shown in Figure 7-10.

ORAPROD1

The Oracle data files were placed on the NFS mounted volume called /Ora_data. The log and control files were located on /Ora_logs, also an NFS mounted volume (see Figure 7-11).

For purposes of verification of proper replication and cloning, the table scott.emp, part of the example database, was used (Figure 7-12).

For purposes of later data comparison, Figure 7-13 shows the current contents of the table.

9.4 ADD HOSTS AND CONFIGURE APPROPRIATE RESOURCE POOLS UNDER PROTECTION MANAGER

9.4.1 Operations Manager

First make sure the hosts have been discovered by the local Operations Manager. If not they need to be added manually. Refer to the Operations Manager documentation for specific procedures. In Figure 9-5, below, a special group called Oracle has been created just for simplicity to show only resources relevant to this scenario.

The screenshot shows the NetApp Operations Manager web interface. The top navigation bar includes 'Control Center', 'Backup', and 'Disaster Recovery'. The left sidebar shows a tree view of groups: Global, DPRGrepo, MetroCluster, Oracle, and perf_char01. The main content area is titled 'Appliances, All' and shows a table of appliances under the 'Oracle' group. The table has columns for Type, Status, Appliance, Model, Serial Number, and System ID. Three appliances are listed: fcs3070-rt01, FAS3070-SITEA, and FAS3070-SITEB. Red arrows point to the first two rows. The bottom of the interface shows a 'Current Group' section with options like 'Add To Group...' and 'Refresh Group Monitors'.

Type	Status	Appliance	Model	Serial Number	System ID
		fcs3070-rt01.tmlab.netapp.corp.com	FAS3070	1073294	0118042010
		FAS3070-SITEA	FAS3070	3087296	0118052265
		FAS3070-SITEB	FAS3070	3087297	0118051930

Figure 9-5) Main Oracle server (Oraprod1).

9.4.2 Protection Manager

Now the resource pools have to be created within Protection Manager to host the disaster recovery destination. See Figure 9-6. For instructions on creating resource pools, refer to the NetApp Protection Manager Administration Guide available on the NetApp NOW site.

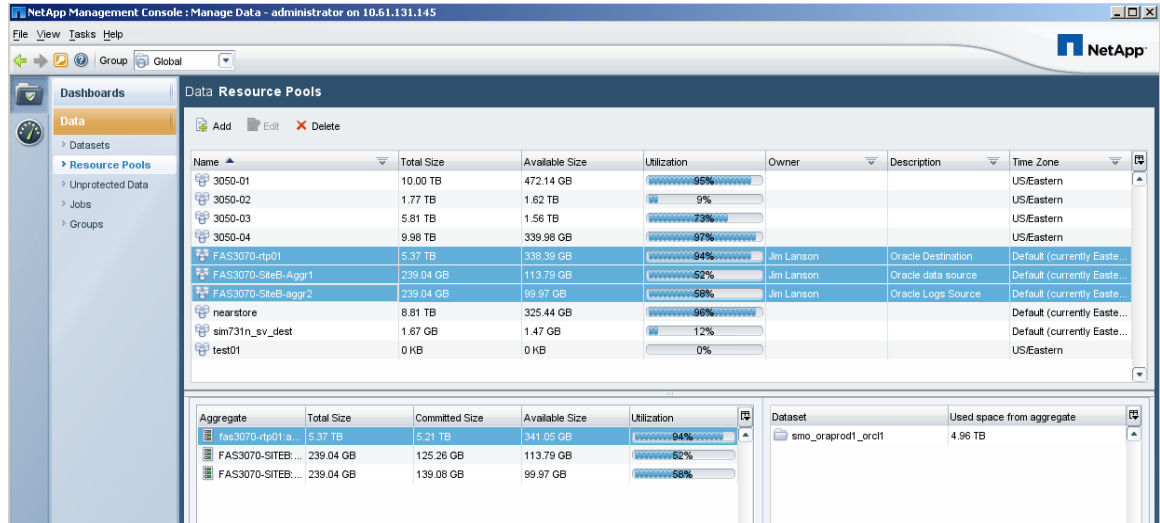


Figure 9-6) Protection Manager resource pools.

9.5 INSTALL AND CONFIGURE SNAPDRIVE FOR UNIX

One of the requirements for operating SnapManager for Oracle is the installation of SnapDrive for UNIX. Once this package is installed and operational, verify proper operation by performing the steps in Figure 9-7:

Configure SnapDrive and specify which OS user will be used to access the NetApp storage system used by the target databases.

```
[oracle@Oraprod1 bin]$ snapdrive config set root my_netapp_storage_system1
```

Verify that the above configuration succeeded.

```
[root@Oraprod1 bin]# ./snapdrive config list
username      appliance name  appliance type
-----
root          FAS3070-SITEB  StorageSystem
sd-oraprod1   10.61.132.11   StorageSystem
sd-admin      10.61.131.145  DFM
[root@Oraprod1 bin]# ./snapdrive storage show -all
NFS device: FAS3070-SITEB:/vol/u02      mount point: /home/oracle/app (non-persistent)
NFS device: FAS3070-SITEB:/vol/u03      mount point: /ora_data (non-persistent)
NFS device: FAS3070-SITEB:/vol/u04      mount point: /ora_logs (non-persistent)
[root@Oraprod1 bin]#
```

Figure 9-7) SnapDrive status.

9.6 INSTALL AND CONFIGURE SNAPMANAGER FOR ORACLE 3.0 INCLUDING CREATION OF REPOSITORY

SnapManager for Oracle uses a repository database to hold all data relative to database profiles and other configuration information. As stated earlier, this database cannot reside on the same Oracle server as the database it is protecting.

9.6.1 SnapManager for Oracle Repository

On the Oracle machine containing the database that will house the repository, create a new tablespace for the SnapManager repository:

```
SQL> create tablespace "smo" datafile /u01/oradata/smrepo/datafile/smo01.dbf'
size 100M autoextend on maxsize 1000M;
```

```
SQL> select name from v$datafile;

NAME
-----
C:\ORACLEXE\ORADATA\XE\SYSTEM.DBF
C:\ORACLEXE\ORADATA\XE\UNDO.DBF
C:\ORACLEXE\ORADATA\XE\SYSAUX.DBF
C:\ORACLEXE\ORADATA\XE\USERS.DBF
C:\ORACLEXE\ORADATA\XE\SMO01.DBF
SQL>
```

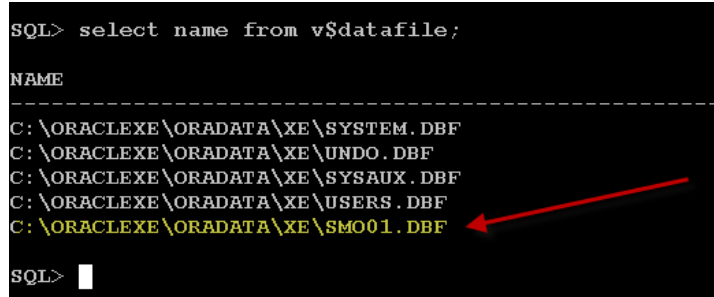


Figure 9-8) SMO repository data file.

SnapManager requires a minimum 4K block size for the tablespace into which it is installed. Check the block size for the “smo” tablespace using:

```
SQL> select tablespace_name, block_size from dba_tablespaces;

TABLESPACE_NAME          BLOCK_SIZE
-----
SYSTEM                   8192
UNDO                      8192
SYSAUX                   8192
TEMP                     8192
USERS                    8192
SMO                      8192
```

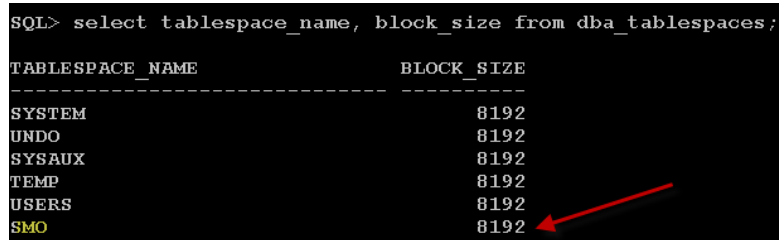


Figure 9-9) SMO block size verification.

Now the user name needed by SMO for communicating with the repository needs to be created.

```
SQL> create user smoadmin identified by adminpw1 temporary tablespace temp default
tablespace smo quota unlimited on smo;
```

```
SQL> grant connect, resource to smoadmin;
```

9.6.2 Installation of SnapManager for Oracle 3.0

After making sure that all requirements are met for the installation of SnapManager for Oracle, complete the installation using the SnapManager for Oracle 3.0 Installation and Administration Guide found on the NOW site.

After completing the installation, at the UNIX shell prompt enter the following command to start the SMO server:

```
[oracle@Oraprod1 bin]$ smo_server start
```

There should be a response indicating the server has started.

To verify proper installation enter the following:


```
[oracle@Oraprod1 bin]$ smo system verify -verbose
[ INFO] SMO-13505: SnapDrive environment verification passed.
[ INFO] SMO-13048: System Verify Operation Status: SUCCESS
[ INFO] SMO-13049: Elapsed Time: 0:00:01.324
Operation Id [N46d309c2f4a681e60fbacdb15b6e3d0e] succeeded.
[oracle@Oraprod1 bin]$
```

Figure 9-10) SMO verification.

Now enter the following command to invoke the SnapManager for Oracle GUI.

```
[oracle@Oraprod1 bin]$ Smogui &
```

A window similar to that in Figure 9-11 should appear.

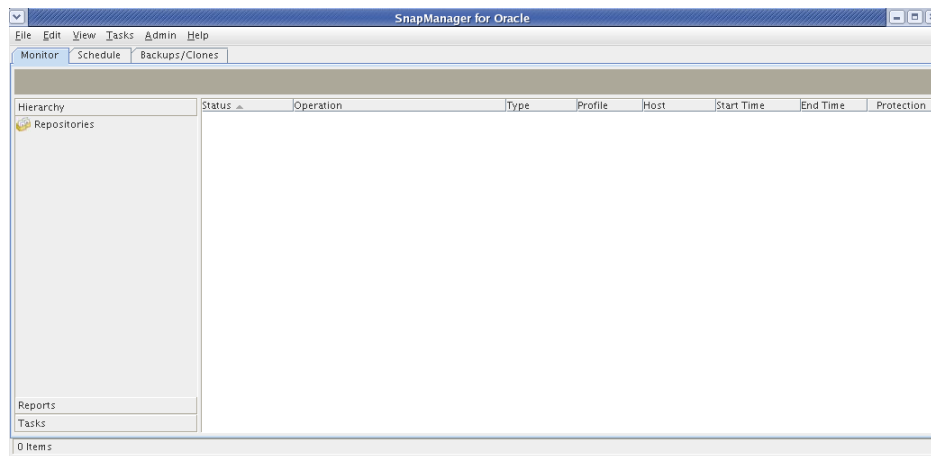


Figure 9-11) SnapManager for Oracle 3.0 user interface.

While the SMO graphical user interface (GUI) is shown in the figures below, the administrator does have the option of using the command line for the SMO actions performed.

The last step is to configure the repository either through the GUI in Figure 9-12 or using the command lines as shown below:

```
[oracle@Oraprod1 bin]$smo credential set -repository -host 10.61.25.22 -dbname XE -
port 1524 -login -username smoadmin
```

```
[oracle@Oraprod1 bin]$smo repository create -repository -dbname XE - host 10.61.25.22
-port 1524 -login -username smoadmin
```

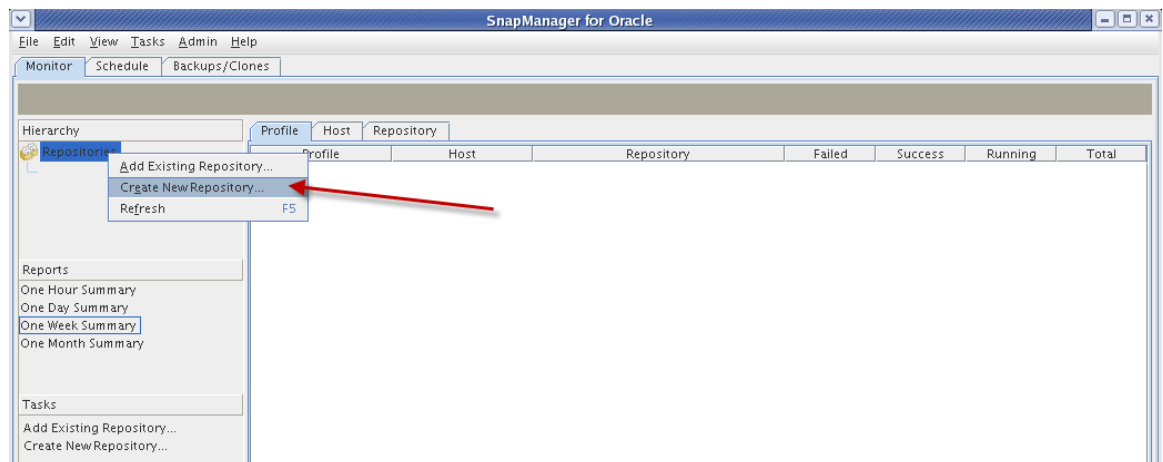


Figure 9-12) SnapManager for Oracle 3.0 user interface.

9.7 CREATE SMO PROFILE FOR MANAGING ORACLE DATABASE

Every database protected by SMO must have a profile. This profile contains credentials and other information pertinent to protection. The credentials in the profile must be of an Oracle user with a SYSDBA role. For the purposes of this scenario the user name is “oracle” and is created by:

```
SQL> create user oracle identified by netappl
```

```
SQL> grant sysdba to oracle
```

While SnapManager for Oracle integration with Oracle Recovery Manager (RMAN) is possible, it is not shown in this scenario. For further details refer to the SnapManager for Oracle Installation Guide located on the NOW site.

To create the profile for the database orcl1, either use the command below or start the wizard as shown in Figure 9-13.

```
[oracle@Oraprod1 bin]$smo profile create -profile Oracle1 - profile-password netappl  
-repository -dbname XE -login -username smoadmin -host 10.61.25.22 -port 1524 -  
database -dbname orcl1 - login -username oracle -password netappl -host 10.61.132.8 -  
port 1521 -sid orcl1 -osaccount oracle -osgroup dba -retain 100 -verbose
```

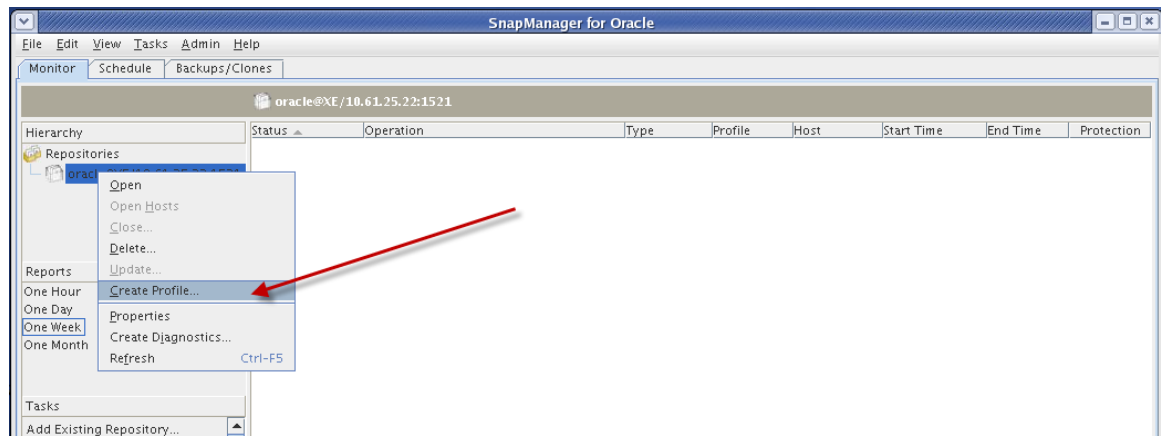


Figure 9-13) SnapManager for Oracle 3.0 profile creation.

After creation of the profile is complete you should see something similar to Figure 9-14.

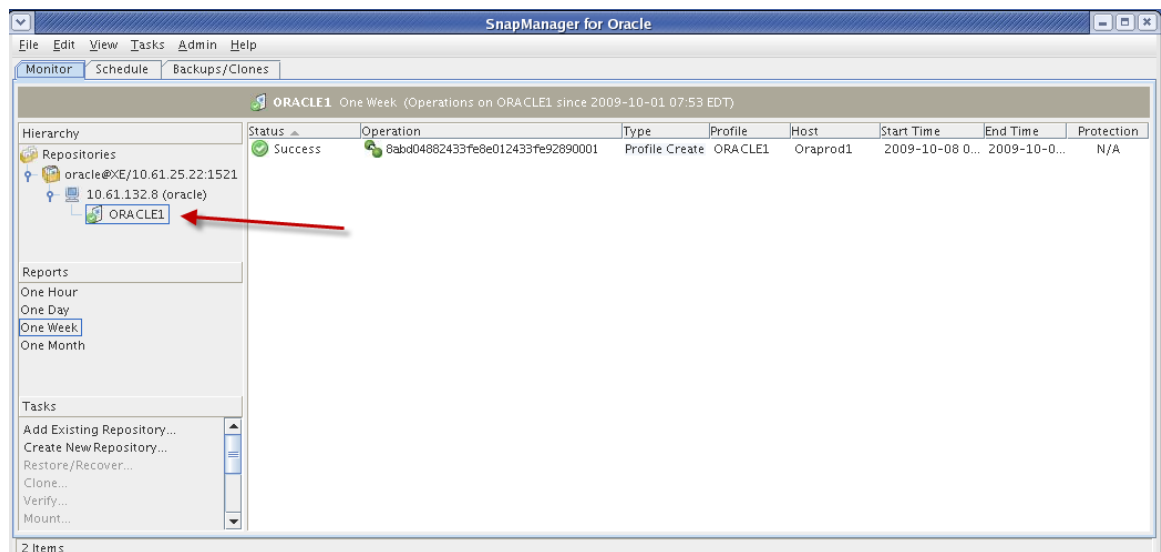


Figure 9-14) SnapManager for Oracle 3.0 profile created.

At this point you could schedule and/or execute backups. However, in order to provide further protection via mirroring from a DR perspective, integration with NetApp Protection Manager is necessary.

9.8 INTEGRATION WITH PROTECTION MANAGER

9.8.1 Operations Manager Setup

Log into the Operations Manager server and perform the following:

Create a new user on the server called sd-admin and record the password you give it.

Execute the following commands to set up the users, groups, and privileges necessary for SnapManager and SnapDrive to communicate with Operations Manager and Protection Manager:

```
OpsmgrHost>dfm user add sd-admin
OpsmgrHost>dfm role create sd-admin-role
OpsmgrHost>dfm role add sd-admin-role DFM.Core.AccessCheck
OpsmgrHost>dfm role add sd-admin-role DFM.Database.Write
OpsmgrHost>dfm user role set sd-admin sd-admin-role
OpsmgrHost>dfm user role add sd-admin GlobalDataProtection GlobalRestore
OpsmgrHost>dfm dfm host set FAS3070-SITEB hostLogin=root hostPassword=`netappl`
OpsmgrHost>dfm host role create -h FAS3070-SITEB -c "api-* , login-*" sd-admin-role
OpsmgrHost>dfm host usergroup create -h FAS3070-SITEB -r sd-admin-role sd-admin-group
OpsmgrHost>dfm host user create -h FAS3070-SITEB -r sd-admin-role -p <password> -g sd-admin-group sd_oraprod1
OpsmgrHost> dfm role add sd-admin-role SD.Config.Read Global
OpsmgrHost> dfm role add sd-admin-role SD.Config.Write Global
OpsmgrHost> dfm role add sd-admin-role SD.Config.Delete Global
OpsmgrHost> dfm role add sd-admin-role SD.Storage.Read Global
OpsmgrHost> dfm role add sd-admin-role SD.Storage.Write Global
OpsmgrHost> dfm role add sd-admin-role SD.Storage.Delete Global
OpsmgrHost> dfm role add sd-admin-role SD.Snapshot.Read Global
OpsmgrHost> dfm role add sd-admin-role SD.Snapshot.Write Global
OpsmgrHost> dfm role add sd-admin-role SD.Snapshot.Delete Global
OpsmgrHost> dfm role add sd-admin-role SD.Snapshot.Restore Global
OpsmgrHost> dfm role add sd-admin-role SD.Snapshot.Clone Global
OpsmgrHost> dfm user add -r sd-admin-role "tardb_host1\root"
```

On the database server enter the following command to register the credentials between SnapDrive and Operations Manager:

```
[root@oraprod1 snapdrive] # snapdrive config set -dfm sd-admin 10.61.131.145
```

Reply with the password previously assigned.:

```
[root@oraprod1 snapdrive] # snapdrive config set sd-oraprod1 FAS3070-SITEB
```

9.8.2 Protection Manager Setup

Log into SnapManager for Oracle and perform the following steps:

Right-click the desired profile as shown in Figure 9-15 and select Properties.

Within the Properties window, click on the Policies tab and check the Protection Manager Protection Policy box.

Now click on the drop-down next to the check box (Figure 9-15) and select the desired policy. In this case we have selected “DR Mirror.”

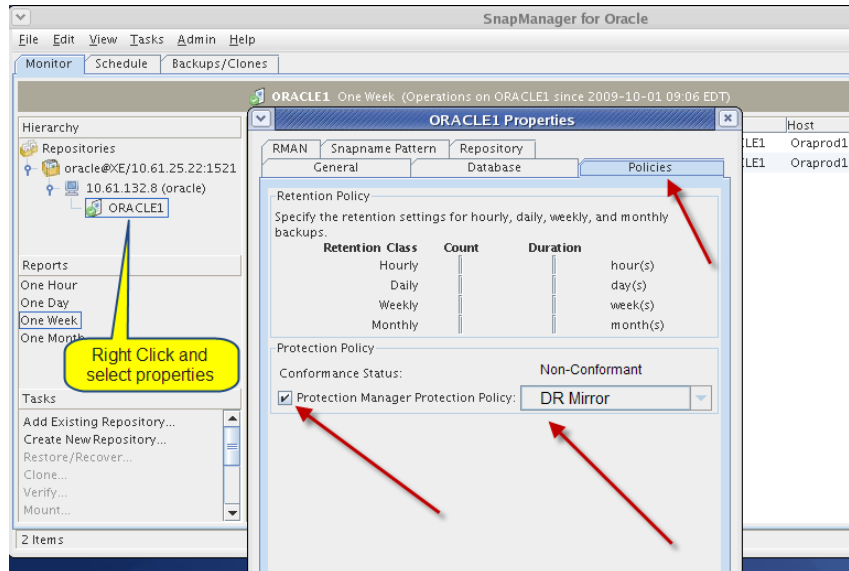


Figure 9-15) SnapManager for Oracle 3.0 Protection Policy selection.

SnapManager for Oracle will automatically create a data set for the database and register it within Protection Manager.

Now all that is left is to assign a resource pool to the data set within Protection Manager.

In Protection Manager, select the data set created, right-click, and select Edit (Figure 9-16).

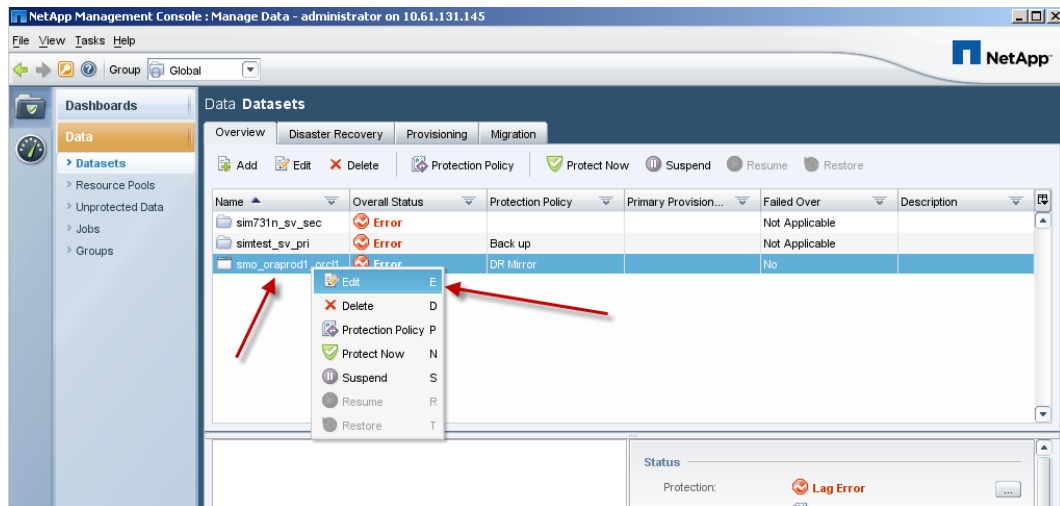


Figure 9-16) Protection Manager resource assignment (Step A).

Now assign the DR Mirror destination to the appropriate resource pool as shown in Figures 9-17 and 9-18.

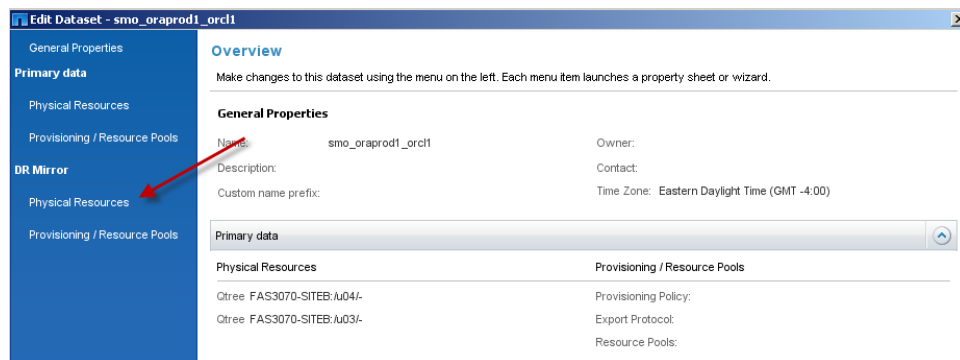


Figure 9-17) Protection Manager resource assignment (Step B).

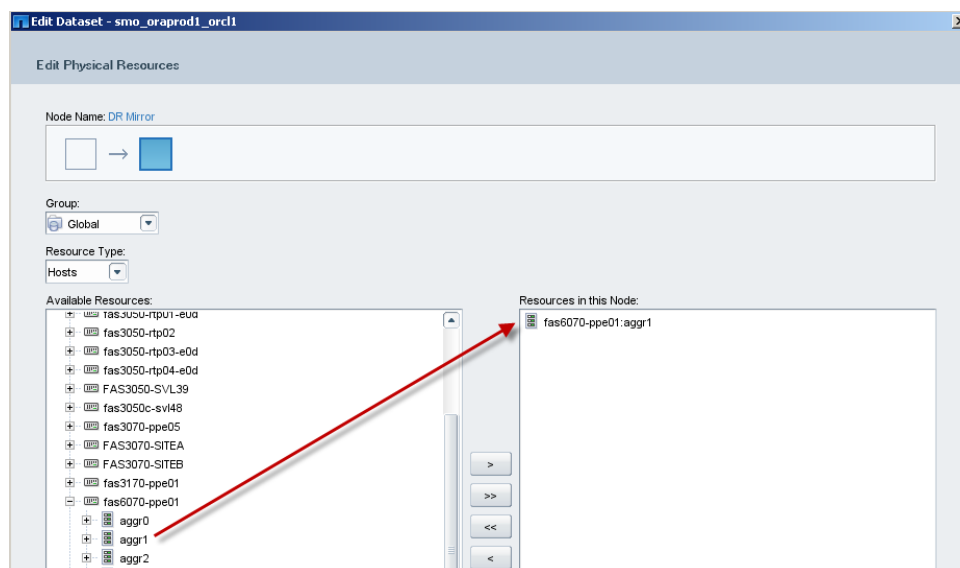


Figure 9-18) Protection Manager resource assignment (Step C).

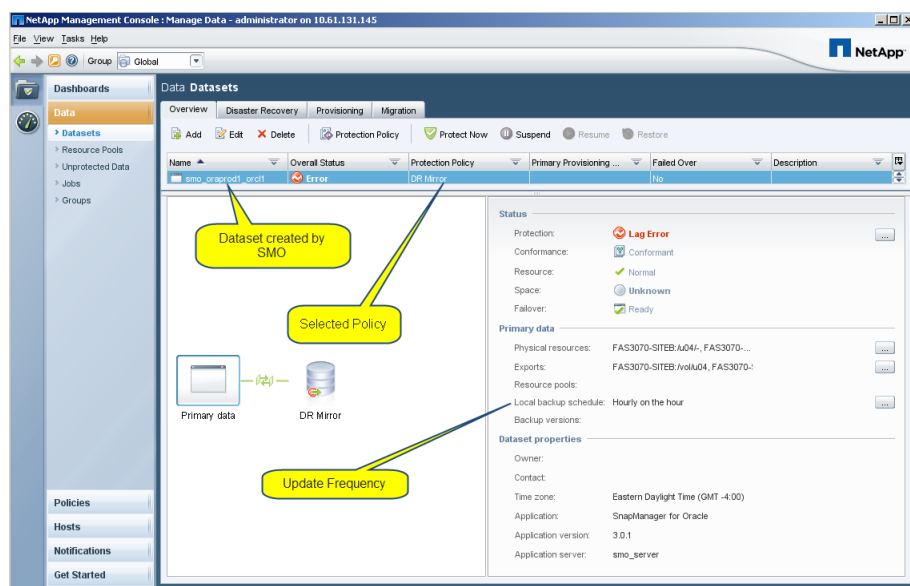


Figure 9-19) Protection Manager data set configuration.

As shown in Figure 9-19, there is now a data set within Protection Manager configured to update every hour under the “DR Mirror” policy.

9.9 VERIFY PROPER OPERATION

Just to make sure everything is operating normally, using Protection Manager click on the Jobs menu under Data (Figure 9-20), then click on the “Dataset” column heading and scroll down to the data set name, in this case “smo_oraprod1_orcl1.”

Job Type	Dataset	vFiler unit	Start	Job Status	End
Local backup confirmati...	smo_oraprod1_orcl1		12 Oct 10:00	✓ Succeeded	12 Oct 10:00
Mirror	smo_oraprod1_orcl1		12 Oct 09:30	✓ Succeeded	12 Oct 09:31
Local backup confirmati...	smo_oraprod1_orcl1		12 Oct 09:00	✓ Succeeded	12 Oct 09:00
Mirror	smo_oraprod1_orcl1		12 Oct 08:30	✓ Succeeded	12 Oct 08:31
Local backup confirmati...	smo_oraprod1_orcl1		12 Oct 08:00	✓ Succeeded	12 Oct 08:00
Mirror	smo_oraprod1_orcl1		12 Oct 07:30	✓ Succeeded	12 Oct 07:31
Local backup confirmati...	smo_oraprod1_orcl1		12 Oct 07:00	✓ Succeeded	12 Oct 07:00
Mirror	smo_oraprod1_orcl1		12 Oct 06:30	✓ Succeeded	12 Oct 06:31
Local backup confirmati...	smo_oraprod1_orcl1		12 Oct 06:00	✓ Succeeded	12 Oct 06:00
Mirror	smo_oraprod1_orcl1		12 Oct 05:30	✓ Succeeded	12 Oct 05:31
Local backup confirmati...	smo_oraprod1_orcl1		12 Oct 05:00	✓ Succeeded	12 Oct 05:00

Figure 9-20) DR Mirror job status.

9.10 INSERT TEST RECORD INTO SAMPLE DATABASE

To test for a successful backup, mirror, and cloning, let’s insert another record into the test table (Figure 9-21). On the production Oracle machine, Oraprod1, perform the following:

```
SQL> insert into scott.emp values (0123, 'Nosnal', 'VP', 6665, '22-JAN-2009', 5000, 500, 30);
```

1 row created.

```
SQL> select ename, empno from scott.emp;
```

ENAME	EMPNO
Nosnal	123
Lanson	3406
SMITH	7369
ALLEN	7499
WARD	7521
JONES	7566
MARTIN	7654
BLAKE	7698
CLARK	7782
SCOTT	7788
KING	7839

ENAME	EMPNO
TURNER	7844
ADAMS	7876
JAMES	7900
FORD	7902
MILLER	7934
Obama	9696

17 rows selected.

Figure 9-20) Test record insertion.

9.11 CREATE FLEXCLONE COPY FROM SNAPMIRROR DESTINATION AND BRING UP ALTERNATE SERVER ON CLONE

Figures 9-21 and 9-22 show the Snapshot copy status on the destination and source machines, respectively.

```
fas6070-ppe01*> snap list du03
Volume du03
working...

  %/used      %/total    date          name
  -----
  0% ( 0%)    0% ( 0%)    Oct 12 15:31  fas6070-ppe01(0118042010)_du03.123
  0% ( 0%)    0% ( 0%)    Oct 12 14:31  fas6070-ppe01(0118042010)_du03.122
  1% ( 0%)    0% ( 0%)    Oct 12 14:01  smo_oracle1_orcl1_f_h_1_8abd04882449d3c9012449d3ce0d0001_0
fas6070-ppe01*> snap list du04
Volume du04
working...

  %/used      %/total    date          name
  -----
  0% ( 0%)    0% ( 0%)    Oct 12 15:31  fas6070-ppe01(0118042010)_du04.123
  0% ( 0%)    0% ( 0%)    Oct 12 14:31  fas6070-ppe01(0118042010)_du04.122
  0% ( 0%)    0% ( 0%)    Oct 12 14:02  smo_oracle1_orcl1_f_h_2_8abd04882449d3c9012449d3ce0d0001_0
fas6070-ppe01*>
```

Figure 9-21) Snapshot summary on destination.

```
FAS3070-SITEB*> snap list u03
Volume u03
working...

  %/used      %/total    date          name
  -----
  0% ( 0%)    0% ( 0%)    Oct 12 15:31  fas6070-ppe01(0118042010)_du03.123 (snapmirror)
  0% ( 0%)    0% ( 0%)    Oct 12 14:01  smo_oracle1_orcl1_f_h_1_8abd04882449d3c9012449d3ce0d0001_0
FAS3070-SITEB*> snap list u04
Volume u04
working...

  %/used      %/total    date          name
  -----
  0% ( 0%)    0% ( 0%)    Oct 12 15:31  fas6070-ppe01(0118042010)_du04.123 (snapmirror)
  0% ( 0%)    0% ( 0%)    Oct 12 14:02  smo_oracle1_orcl1_f_h_2_8abd04882449d3c9012449d3ce0d0001_0
FAS3070-SITEB*>
```

Figure 9-22) Snapshot summary on source.

Now let's create the clones as shown in Figure 9-23. One clone was created for the data volume and one for the logs volume.

```
fas6070-ppe01*> snap list du03
Volume du03
working...

  %/used      %/total    date          name
  -----
  0% ( 0%)    0% ( 0%)    Oct 12 15:31  fas6070-ppe01(0118042010)_du03.123
  0% ( 0%)    0% ( 0%)    Oct 12 14:31  fas6070-ppe01(0118042010)_du03.122
  1% ( 0%)    0% ( 0%)    Oct 12 14:01  smo_oracle1_orcl1_f_h_1_8abd04882449d3c9012449d3ce0d0001_0
fas6070-ppe01*> vol clone create u03_clone -b du03 fas6070-ppe01(0118042010)_du03.123
Mon Oct 12 15:50:28 EDT [waf1.volume.clone.created:info]: Volume clone u03_clone of volume du03 was created successfully.
Creation of clone volume 'u03_clone' has completed.
fas6070-ppe01*> snap list du04
Volume du04
working...

  %/used      %/total    date          name
  -----
  0% ( 0%)    0% ( 0%)    Oct 12 15:31  fas6070-ppe01(0118042010)_du04.123
  0% ( 0%)    0% ( 0%)    Oct 12 14:31  fas6070-ppe01(0118042010)_du04.122
  0% ( 0%)    0% ( 0%)    Oct 12 14:02  smo_oracle1_orcl1_f_h_2_8abd04882449d3c9012449d3ce0d0001_0
fas6070-ppe01*> vol clone create u04_clone -b du04 fas6070-ppe01(0118042010)_du04.123
Creation of clone volume 'u04_clone' has completed.
fas6070-ppe01*>
```

Figure 9-23) Clone creation.

Now that the clones have been created, we can mount them on a different Oracle server (OraProd2) and bring up the database running on the clones.

```

oracle@OraProd2:~$ uname -a
Linux OraProd2 2.6.9-55.ELsmp #1 SMP Fri Apr 20 17:03:35 EDT 2007 i686 i686 i386 GNU/Linux
oracle@OraProd2:~$ mount
/dev/mapper/VolGroup00-LogVol100 on / type ext3 (rw)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
10.61.132.70:/vol/u03_clone on /ora_data type nfs (rw,addr=10.61.132.70)
10.61.132.70:/vol/u04_clone on /ora_logs type nfs (rw,addr=10.61.132.70)
oracle@OraProd2:~$

```

Figure 9-24) Clones mounted on OraProd2 server.

9.12 VERIFY TEST RECORD EXISTS IN DATABASE RUNNING ON CLONE

The last step is to log into Oracle and verify the existence of the record previously inserted on OraProd1, the production machine.

```

SQL> insert into scott.emp values (0123, 'Nosnal', 'VP', 6665, '22-JAN-2009', 5000, 500, 30);

1 row created.

SQL> select ename, empno from scott.emp;

ENAME          EMPNO
-----
Nosnal          123
Lanson          3406
SMITH           7369
ALLEN           7499
WARD            7521
JONES           7566
MARTIN          7654
BLAKE           7698
CLARK           7782
SCOTT           7788
KING            7839

ENAME          EMPNO
-----
TURNER          7844
ADAMS           7876
JAMES           7900
FORD            7902
MILLER          7934
Obama           9696

17 rows selected.

SQL>

```

Figure 9-25) DR Mirror job status.

The database started from the clone volumes can now be used for development, testing, and staging. Note that it is also possible to clone a FlexClone volume. Once the testing is complete, the FlexClone volume can either be destroyed or split. If the FlexClone volume is split, the volume will consume space. Therefore, make sure that you have sufficient space in the aggregate before splitting a clone.

10 VERSION HISTORY

Version 1.0	November 2009 Original document covering scenarios 1, 2, and 3
Version 1.0.1	February 2010 Minor change from use of rsh to ssh

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 must be used solely in connection with the NetApp products discussed in this document.

© Copyright 2010 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexClone, FlexVol, NOW, SnapDrive, SnapManager, SnapMirror, Snapshot, SnapValidator, and SnapVault are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Microsoft, SQL Server, and Windows are registered trademarks of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds. Oracle is a registered trademark and Oracle 11g and 9i are trademarks of Oracle Corporation. SAP is a registered trademark of SAP AG. VMware is a registered trademark of VMware, 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.