



NetApp[®]
Go further, faster

Technical Report

Clouds: Running Multiple Oracle Customers with Solaris Zones on NetApp MultiStore

Jorge Costa, NetApp
February 2010 | TR-3818

TABLE OF CONTENTS

1	INTRODUCTION	3
2	LAYOUT DETAILS	3
3	IMPLEMENTATION	5
7.1	BUILDING THE VFILER UNIT	5
7.2	CREATING A GENERIC ZONE TEMPLATE	10
7.3	ADDING A NEW ZONE	22
7.4	CREATING A NEW DATABASE ENVIRONMENT	25
3.5	CONFIGURING SNAPMANAGER FOR ORACLE	30
3.6	MOVING THE VFILER UNIT BETWEEN CONTROLLERS	39
3.7	MOVING A ZONE BETWEEN SERVERS	40
8	CONCLUSION	42

1 INTRODUCTION

At present, there is a lot of interest in clouds, or shared infrastructures. With the rise of energy costs, the cost of floor space, and the lack of local IT support skills, it is evident that many IT departments are looking to consolidate and virtualize their servers, use remote cloud services for new applications, or simply host seasonal projects that would normally require acquiring new servers.

To address these issues, customers use different virtualizations solutions that are available on the market, like XEN hypervisors, AIX PowerVM, and Solaris™ Zones. Each has its strengths and limitations.

Although VMware is the market leader in terms of server virtualization, VMware still has some challenges with Oracle. As of today, Oracle still does not fully support its products on any x86 virtualized product other than its own Oracle® VM.

Oracle's position is quite different in regard to non-x86 platforms. Virtualization, or server partitioning, has been around in the RISC world for quite a while. As such, Oracle understands and supports Oracle in other virtualizations products such as Solaris Zones.

This document addresses the queries for using Solaris containers with NetApp® technologies such as SnapDrive® for UNIX® and SnapManager® for Oracle (SMO). It describes how to group a large number of customers (think clouds) running Oracle Databases on a large number of Solaris containers and perform backup and restore on them, easily and quickly, using NetApp Snapshot™. The SMO fast vol restore feature enables restoring TBs of data in just a few minutes.

When restoring a FlexVol® volume from a Snapshot copy, everything contained within the FlexVol volume is restored. This entails that multiple volumes be required to back up/restore multiple databases. However, Data ONTAP® limits the number of FlexVol volumes we can have per controller. This can pose a challenge if you do not have a large number of hosts and a large number of small database instances that fit within the limits.

To solve this problem, first reduce the number of FlexVol volumes by sharing as many objects (of the same type) within the same FlexVol volume. Then virtualize the servers to serve a large number of servers and databases from a small collection of volumes.

2 LAYOUT DETAILS

Figure 1 illustrates how FlexVol volumes, qtrees, and Solaris containers can be grouped with the security requirements of a multicustomer shared infrastructure. In the example below:

- Oracle files from different databases are shared in the same FlexVol volume as long they are from the same type.
- redologs from SID1 are stored along redologs from SID2.
- Oracle archive files from SID1 and SID2 are contained in the same FlexVol volume.

The only FlexVol volumes that are not shared are the FlexVol volumes containing the Oracle data files, allowing a fast restore if required.

- Each customer has a dedicated virtualized storage system (vFiler™ unit) and its own set of FlexVol volumes, allowing access only to the storage resources assigned to their company. Figure 1 shows an example of a `customer01` who has access to a series of qtrees and a few dedicated FlexVol volumes.

Figure 1 may seem a bit complex and difficult to understand at first glance. However, that will change as we work through this document and build the solution one step at a time.

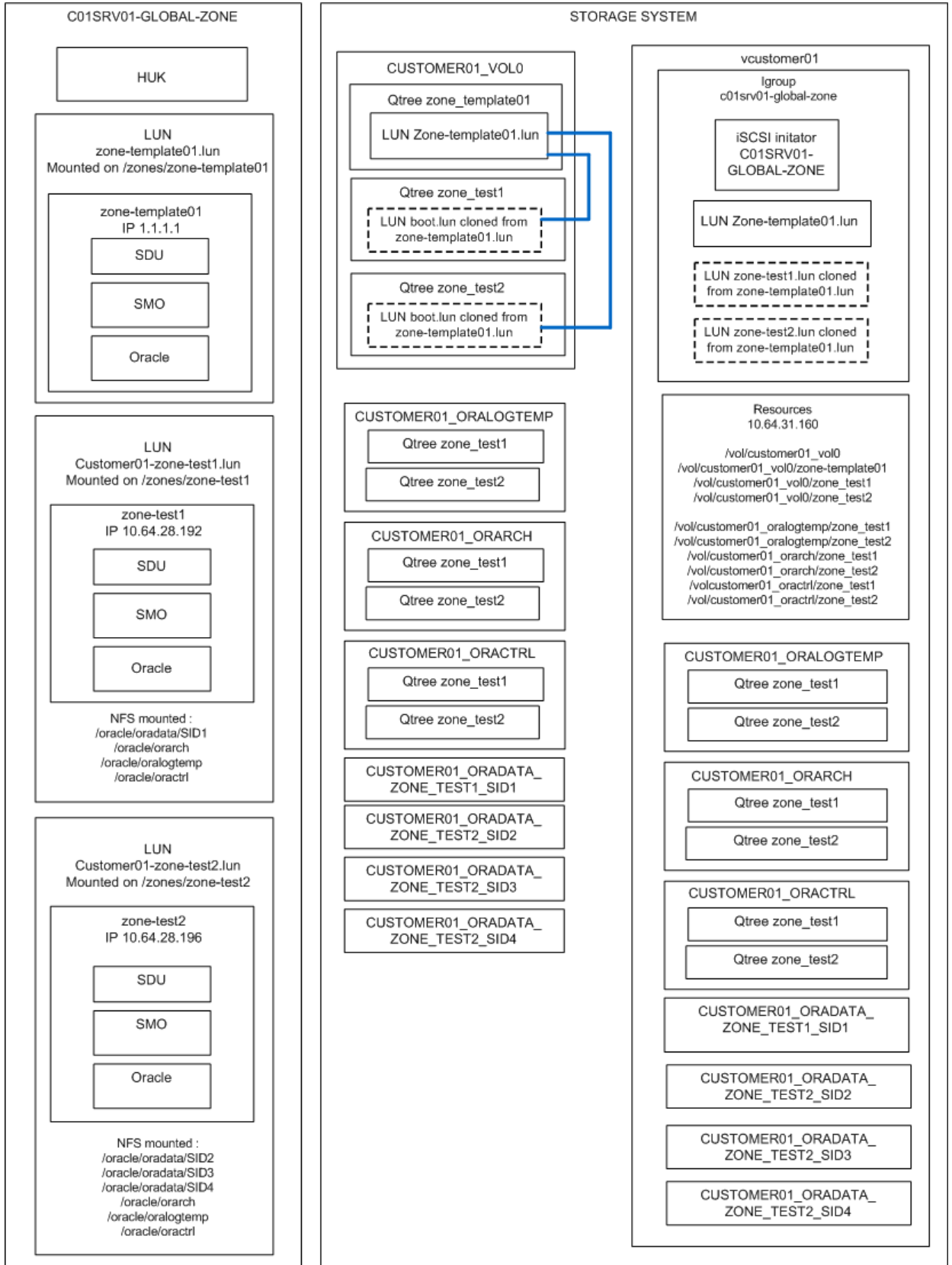


Figure 1) Layout overview.

3 IMPLEMENTATION

This document describes the step-by-step procedure of deploying a shared multicustomer deployment of servers, Solaris Zones, and databases across a single NetApp controller (or clustered pair).

You can use any supported configuration listed in the NetApp Interoperability Matrix. This example uses versions SDU 4.1.1, SMO 3.0.2, Data ONTAP 7.3.1, and Solaris 10U4.

In this example, we build an environment for a customer called `customer01`. The infrastructure consists of the following:

- One Solaris server with two zones each containing two Oracle Databases
- A group of dedicated FlexVol volumes on a shared NetApp controller
- A set of qtrees for each database
- A dedicated vFiler unit

The implementation of this solution is outlined by the following steps:

- 1 [Building the vFiler unit](#)
- 2 [Creating a generic zone template](#)
- 3 [Adding a new zone](#)
- 4 [Creating a new database environment](#)
- 5 [Configuring SnapManager for Oracle](#)
- 6 [Moving the vFiler unit between controllers](#)
- 7 [Moving a zone between servers](#)

7.1 BUILDING THE VFILER UNIT

Each customer has his/her own vFiler unit to prevent unauthorized access or access to other customers' data.

In this example, a vFiler unit named `vcustomer01` is created.

All vFiler units require a `vol0` volume; this can be stored in a FlexVol volume or in a qtree. This example uses a FlexVol volume named `customer01_vol0`.

To create the vFiler unit:

1. Log in to the NetApp storage system and create the FlexVol volumes for the Solaris Zone OS. This volume contains the O.S. boot LUNs for diverse containers `o10` for the `vcustomer01` vFiler customer:

```
filer> vol create customer01_vol0 aggr0 30g
filer> vol options customer01_vol0 guarantee none
filer> vol options customer01_vol0 nosnapdir on
filer> vol autosize customer01_vol0 on
filer> snap reserve customer01_vol0 0
filer> snap sched customer01_vol0 0 0 0
filer> sis on /vol/customer01_vol0
filer> sis start -s /vol/customer01_vol0
```

2. Create a vFiler unit named `vcustomer01`.

In this example, the virtual IP address for this vFiler unit is 10.64.31.160.

```
filer> vfiler create vcustomer01 -i 10.64.31.160 /vol/customer01_vol0
Mon Dec 7 14:07:20 GMT [vcustomer01@filer: useradmin.added.deleted:info]: The
role 'compliance' has been added.
Mon Dec 7 14:07:21 GMT [vcustomer01@filer: export.file.missing:warning]: Could
not open '/etc/exports' for reading.
Mon Dec 7 14:07:21 GMT [vcustomer01@filer: cmds.vfiler.info:notice]: vFiler unit
vcustomer01 initialized.
```

```

The etc configuration directory for vfiler "vcustomer01" is
/vol/customer01_vol0/etc.

Setting up vfiler vcustomer01
Configure vfiler IP address 10.64.31.160? [y]:
Interface to assign this address to {e0a, e0b, e0c, e0d}: e0a
Netmask to use: [255.255.255.0]:
    The administration host is given root access to the filer's
    /etc files for system administration. To allow /etc root access
    to all NFS clients enter RETURN below.
Please enter the name or IP address of the administration host:
Do you want to run DNS resolver? [n]:
Do you want to run NIS client? [n]:
Mon Dec  7 14:07:30 GMT [vcustomer01@filer: useradmin.added.deleted:info]: The
user 'root' has been added.
Default password for root on vfiler vcustomer01 is "".
New password:
Retype new password:
Mon Dec  7 14:07:34 GMT [vcustomer01@filer: passwd.changed:info]: passwd for user
'root' changed.
Do you want to setup CIFS? [y]: n
filer>

```

- From your Solaris server, log in through iSCSI to the vFiler unit, not to the storage system.
The server hostname used here is `c01srv01-global-zone`.

Where:

`c##` is customer01 and `srv##` is server01

```

root@c01srv01-global-zone:/# iscsiadm add discovery-address 10.64.31.160:3260
root@c01srv01-global-zone:/#

filer> vfiler run vcustomer01 iscsi initiator show

===== vcustomer01
Initiators connected:
  TSIH  TPGGroup  Initiator/ISID/IGroup
  1906   1000    iqn.1986-03.com.sun:01:fac868f0ffff.4ae7abc5 / 40:00:00:2a:00:00
/

```

- Create an igroup for the global zone of the Solaris server using the following convention:

`<##>-<srv##>-<global-zone>`

Where ## is the number of servers. `<##>-<srv##>-<global-zone>`

Each customer can have multiple servers:

```

filer> vfiler run vcustomer01 igroup create -i -t solaris c01srv01-global-zone
iqn.1986-03.com.sun:01:fac868f0ffff.4ae7abc5

```

Note: Solaris does not support the root file system of a zone on NFS. Hence, configure the iSCSI and create a LUN for each container.

- Install the NetApp Host Utilities Kit:

```

root@c01srv01-global-zone:/ # pkgadd -d ./NTAPSANTool.pkg

The following packages are available:
  1  NTAPSANTool      Solaris SAN Host Utilities for NetApp Storage.
                        (i386) 5.1

Select package(s) you wish to process (or 'all' to process

```

```

all packages). (default: all) [?,??,q]: all

Processing package instance <NTAPSANTool> from </software/NTAPSANTool.pkg>

Solaris SAN Host Utilities for NetApp Storage.(i386) 5.1
Copyright (c) 2005-2009 NetApp, Inc.
All Rights Reserved.
Using </> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.

The following files are already installed on the system and are being
used by another package:
  /usr <attribute change only>
  /usr/share <attribute change only>
  /usr/share/man <attribute change only>
  /usr/share/man/man1 <attribute change only>

Do you want to install these conflicting files [y,n,?,q] y
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <NTAPSANTool> [y,n,?] y

Installing Solaris SAN Host Utilities for NetApp Storage. as <NTAPSANTool>

## Installing part 1 of 1.
/opt/NTAP/SANToolkit/NOTICES.PDF
/opt/NTAP/SANToolkit/bin/basic_config
/opt/NTAP/SANToolkit/bin/brocade_info
/opt/NTAP/SANToolkit/bin/cisco_info
/opt/NTAP/SANToolkit/bin/controller_info
/opt/NTAP/SANToolkit/bin/filer_info
/opt/NTAP/SANToolkit/bin/mcdata_info
/opt/NTAP/SANToolkit/bin/mpxio_set
/opt/NTAP/SANToolkit/bin/qlogic_info
/opt/NTAP/SANToolkit/bin/san_version
/opt/NTAP/SANToolkit/bin/sanlun
/opt/NTAP/SANToolkit/bin/solaris_info
/opt/NTAP/SANToolkit/bin/ssd_config.pl
/opt/NTAP/SANToolkit/bin/vidpid.dat
/usr/share/man/man1/basic_config.1
/usr/share/man/man1/brocade_info.1
/usr/share/man/man1/cisco_info.1
/usr/share/man/man1/controller_info.1
/usr/share/man/man1/filer_info.1
/usr/share/man/man1/mcdata_info.1
/usr/share/man/man1/mpxio_set.1
/usr/share/man/man1/qlogic_info.1
/usr/share/man/man1/sanlun.1
/usr/share/man/man1/solaris_info.1
[ verifying class <none> ]
## Executing postinstall script.
Please add /opt/NTAP/SANToolkit/bin in the PATH environment variable
.
.

Installation of <NTAPSANTool> was successful.

```


BUILDING THE BASE SET OF FLEXVOL VOLUMES

Each customer has their own basic set of FlexVol volumes (`customer_vol0`, `oralogtemp`, `orarch`, `oractrl`).

On these FlexVol volumes, we will create a Solaris template for a zone containing a full copy of the OS (whole root zone). This template will be built on an iSCSI LUN (Solaris doesn't support zones on NFS), and for each new Solaris zone we will clone the LUN and map it to the global zone. With this method a customer can easily deploy up to 254 Solaris Zones on a single controller head with minimum resource consumption.

As an example, a single controller could host 64 vFiler units with 10 databases and Snapshot history for 3 weeks on primary disk. This is a total of 640 databases per controller.

Consider that the customer can have an identical setup on the second head and double that number of zones; however, since there will be multiple customers running services (other zones, databases) on the same controller head it is most likely that the controller resources (CPU, disk bandwidth, memory) will be exhausted before a customer reaches those numbers.

When this occurs, any new requests for new zones or databases should be redirected to a new controller head.

The first step is to log in to the NetApp storage and create FlexVol volumes for the Oracle files. `customer01_oralogtemp` will contain all the online redologs and temp tablespaces for all Oracle instances. We will not take Snapshot copies of this FlexVol volume.

```
filer> vol create customer01_oralogtemp aggr0 10g
filer> vol options customer01_oralogtemp guarantee none
filer> vol options customer01_oralogtemp nosnapdir on
filer> snap sched customer01_oralogtemp 0 0 0
```

`customer01_orarch` contains the Oracle archived redo logs:

```
filer> vol create customer01_orarch aggr0 30g
filer> vol options customer01_orarch guarantee none
filer> vol options customer01_orarch nosnapdir on
filer> snap sched customer01_orarch 0 0 0
filer> sis on /vol/customer01_orarch
filer> sis start -s /vol/customer01_orarch
```

`customer01_oractrl` contains the Oracle control files:

```
filer> vol create customer01_oractrl aggr0 1g
filer> vol options customer01_oractrl guarantee none
filer> vol options customer01_oractrl nosnapdir on
filer> snap sched customer01_oractrl 0 0 0
filer> sis on /vol/customer01_oractrl
filer> sis start -s /vol/customer01_oractrl
```

Note that a FlexVol volume for the Oracle data files is not created. A dedicated FlexVol volume for the Oracle data files is created later, as described in section 7.4.

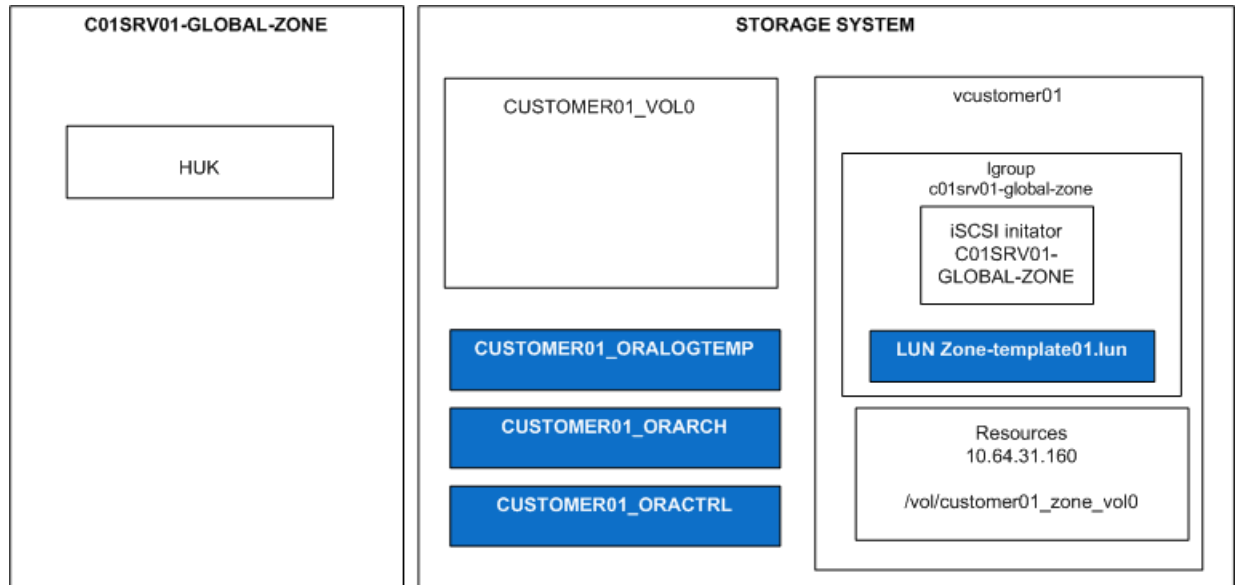


Figure 3) Building FlexVol volumes.

7.2 CREATING A GENERIC ZONE TEMPLATE

For each new Solaris Zone, create one qtree in each FlexVol volume. Before creating these zones to host applications, create a zone template. This template will be named `zone-template01`. Depending on the requirements of the applications there could be a different number of templates. For this example, `zone-template01` will have Oracle binaries, SnapDrive, and SnapManager for Oracle installed.

Notice that the naming on the Data ONTAP side is `zone_template01`, but on Solaris it is `zone-template01`. This difference in the naming convention is because Data ONTAP does not support "-" in the FlexVol volume name and Solaris does not support "-" as part of the hostname.

This section describes how to create a template for the zones. This template will be cloned every time a new zone is built. In this zone, install all utilities required by different environments.

Different templates can be built as desired. The `zone-template01` template created here includes users, groups, and kernel parameters for Oracle, the Oracle binaries, SnapDrive, and SnapManager for Oracle.

1. Create a qtree for the new zone inside the `customer01_vol0` FlexVol volume. This qtree will contain the boot LUN of our zone:

```
filer> vfiler run vcustomer01 qtree create /vol/customer01_vol0/zone_template01
```

2. Create a boot LUN for the template:

```
filer> vfiler run vcustomer01 lun create -t solaris -o noreserve -s 20g
/vol/customer01_vol0/zone_template01/zone_template01.lun
```

3. Map the LUN to the zone igroup:

```
filer> vfiler run vcustomer01 lun map
/vol/customer01_vol0/zone_template01/zone_template01.lun c01srv01_global_zone
```

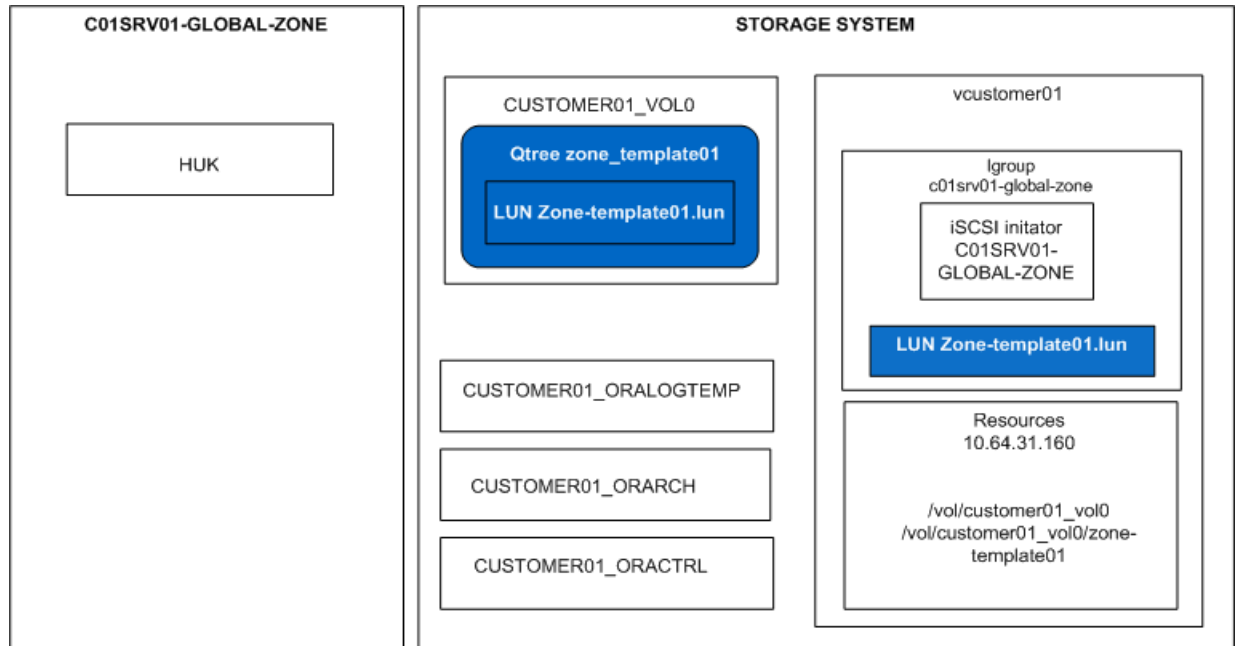


Figure 4) Creating a generic zone template (1).

4. Log in to the host and reload the iSCSI driver:

```
root@c01srv01-global-zone:/ # devfsadm -i iscsi
root@c01srv01-global-zone:/ # /opt/NTAP/SANToolkit/bin/sanlun lun show | grep
zone_template
vcustomer01: /vol/customer01_vol0/zone_template01/zone_template01.lun
/dev/rdisk/c2t60A980004335427467345371482F507Ad0s2 scsi_vhci0 iSCSI 20g
(21474836480) GOOD
```

5. Mount the new LUN:

```
root@c01srv01-global-zone:/ # mkdir /zones/zone_template01
root@c01srv01-global-zone:/ # newfs
/dev/rdisk/c2t60A980004335427467345371482F507Ad0s2
root@c01srv01-global-zone:/ # mount
/dev/dsk/c2t60A980004335427467345371482F507Ad0s2 /zones/zone_template01
root@c01srv01-global-zone:/ # chmod 700 /zones/zone_template01
```

6. Create a pool with the licensed number of CPUs for this zone:

```
root@c01srv01-global-zone:/ # svcadm enable svc:/system/pools:default
root@c01srv01-global-zone:/ # svcadm enable svc:/system/pools/dynamic:default

root@c01srv01-global-zone:/ # pooladm -e
root@c01srv01-global-zone:/ # pooladm -s /etc/pooladm.conf

root@c01srv01-global-zone:/ # cat >/tmp/pool.cfg <<EOF
create pset ps_pool1 ( uint pset.min = 1 ; uint pset.max = 2)
create pool pool1
associate pool pool1 ( pset ps_pool1 )
EOF

root@c01srv01-global-zone:/ # poolcfg -f /tmp/pool.cfg
root@c01srv01-global-zone:/ # pooladm -c
root@c01srv01-global-zone:/ # pooladm -n
```

7. Build the configuration file for the `zone-template01` template:

```

root@c01srv01-global-zone:/ # cat >/tmp/zone.cfg << EOF
create -b
set zonepath=/zones/zone-template01
set autoboot=false
set ip-type=shared
set pool=pool1
add net
set address=1.1.1.1
set physical=e1000g0
end
EOF
root@c01srv01-global-zone:/ # zonecfg -z zone-template01 -f /tmp/zone.cfg

```

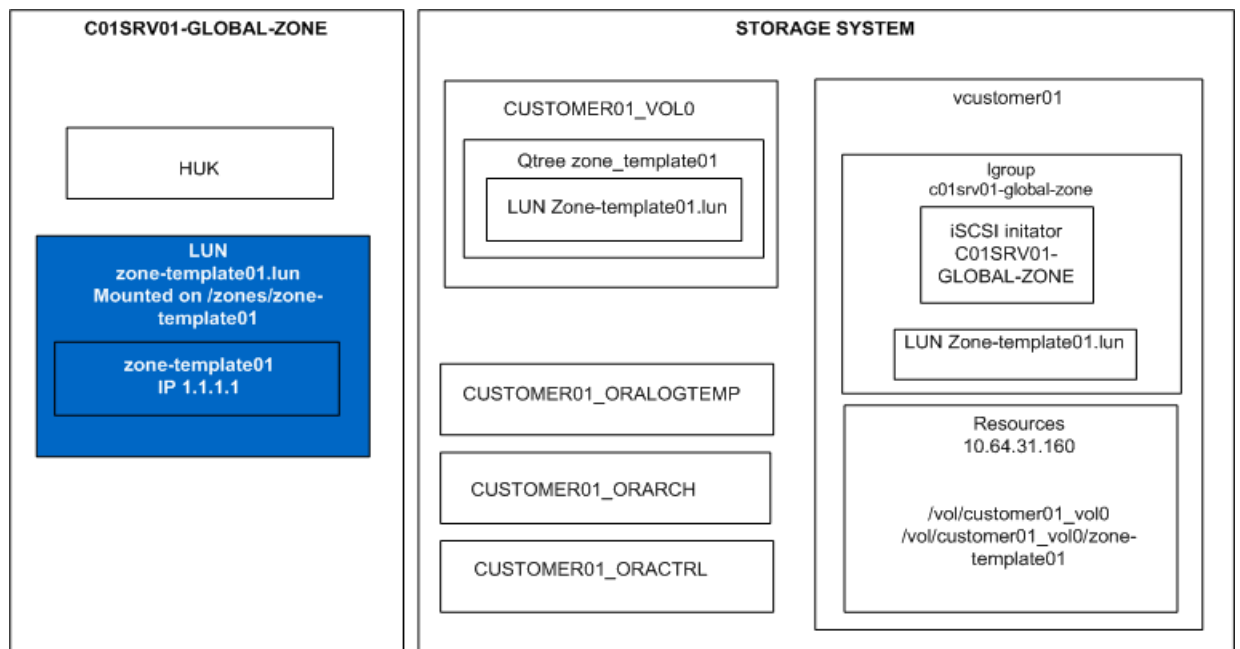


Figure 5) Creating a generic zone template (2).

8. Install a copy of the operating system in the container.

This may take several minutes to complete.

```

root@c01srv01-global-zone:/ # zoneadm -z zone-template01 install

```

9. After the operating system is installed, boot it and connect to the console.

10. Perform the first boot configuration procedure. Configure the hostname, DNS, and so on.

```

root@c01srv01-global-zone:/ # zoneadm -z zone-template01 boot
root@c01srv01-global-zone:/ # zlogin -C zone-template01

[Connected to zone 'zone-template01' console]

SunOS Release 5.10 Version Generic_120012-14 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

```

```
Hostname: zone-template01

Select a Language

    0. English
    1. French
    2. German
    3. Italian
    4. Japanese
    5. Korean
    6. Simplified Chinese
    7. Spanish
    8. Swedish
    9. Traditional Chinese

Please make a choice (0 - 9), or press h or ? for help:
```

11. Enter relevant information as prompted.

The following message appears after the container is created:

```
System identification is completed.

rebooting system due to change(s) in /etc/default/init

[NOTICE: Zone rebooting]

SunOS Release 5.10 Version Generic_120012-14 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: zone-template01

zone-template01 console login:
```

When the container is up and running, you can install all the software required in your zone.

12. Install the Oracle binaries.

13. Create the Oracle user:

```
root@zone-template01:/ # /usr/sbin/groupadd oinstall
root@zone-template01:/ # /usr/sbin/groupadd dba
root@zone-template01:/ # useradd -d /export/home/oracle -g dba -G oinstall -m -s
/usr/bin/bash oracle
root@zone-template01:/ # mkdir /export/home/oracle
root@zone-template01:/ # chown oracle:dba /export/home/oracle
```

14. Create the .profile, .bash_profile, .bashrc:

```
root@zone-template01:/ # cat > /export/home/oracle/.profile <<EOF
# This is the default standard profile provided to a user.
# They are expected to edit it to meet their own needs.

MAIL=/usr/mail/${LOGNAME:?}
umask 022
TMP=/tmp
TMPDIR=$TMP
DISPLAY=localhost:0.0
export TMP TMPDIR DISPLAY
ORACLE_BASE=/oracle/
ORACLE_HOME=/oracle/app/oracle/product/10.2.0/db_1
```

```

PATH=$ORACLE_HOME/bin:$PATH
export ORACLE_BASE ORACLE_HOME ORACLE_SID PATH
EOF

root@zone-template01:/ # cp /export/home/oracle/.profile
/export/home/oracle/.bash_profile
root@zone-template01:/ # chown -r oracle.dba /export/home/oracle

```

15. Set the kernel parameters:

```

root@zone-template01:/ # projadd oracle
root@zone-template01:/ # echo "user.oracle:124::oracle::project.max-sem-
ids=(priv,256,deny);project.max-shm-memory=(priv,4294967296,deny)" >>
/etc/projects
root@zone-template01:/ # echo "oracle::::project=oracle" >> /etc/user_attr

```

16. Install the Oracle software:

```

root@zone-template01:/ # mkdir /oracle
root@zone-template01:/ # chown oracle:dba /oracle

oracle@zone-template01:~$ export SOFTWARE=/software/oracledvd/database
oracle@zone-template01:~$ export ORACLE_BASE=/oracle
oracle@zone-template01:~$ export
ORACLE_HOME=/oracle/app/oracle/product/10.2.0/db_1
oracle@zone-template01:~$ runInstaller -silent -responseFile
$SOFTWARE/response/enterprise.rsp ORACLE_HOME=$ORACLE_HOME ORACLE_HOME_NAME=db_1
n_configurationOption=3

Starting Oracle Universal Installer...

Checking installer requirements...

Checking operating system version: must be 5.10.    Actual 5.10
                                     Passed

Checking Temp space: must be greater than 250 MB.   Actual 6023 MB   Passed
Checking swap space: must be greater than 500 MB.   Actual 7064 MB   Passed

All installer requirements met.

Preparing to launch Oracle Universal Installer from /tmp/OraInstall2009-11-18_05-
20-43PM.
Please wait ...
Oracle Universal Installer, Version 10.2.0.1.0 Production
Copyright (C) 1999, 2005, Oracle. All rights reserved.

You can find a log of this install session at:
 /oracle/oraInventory/logs/installActions2009-11-18_05-20-43PM.log
.....
..... 100% Done.

Loading Product Information
.....
..... 100% Done.

Analyzing dependencies
.....
Starting execution of Prerequisites...
Total No of checks: 9

```

```

Performing check for CertifiedVersions
Checking operating system requirements ...
Expected result: One of 5.10
Actual Result: 5.10
Check complete. The overall result of this check is: Passed

Check complete: Passed
=====
Performing check for Packages
Checking operating system package requirements ...
Checking for SUNWbtool; found CCS tools bundled with SunOS(SUNWbtool). Passed
Checking for SUNWarc; found Lint Libraries (usr)(SUNWarc). Passed
Checking for SUNWhea; found SunOS Header Files(SUNWhea). Passed
Checking for SUNWlibm; found Math & Microtasking Library Headers & Lint Files
(Usr)(SUNWlibm). Passed
Checking for SUNWlibms; found Math & Microtasking Libraries (Usr)(SUNWlibms).
Passed
Checking for SUNWsprot; found Solaris Bundled tools(SUNWsprot). Passed
Checking for SUNWtoo; found Programming Tools(SUNWtoo). Passed
Checking for SUNWilof; found ISO-8859-1 (Latin-1) Optional Fonts(SUNWilof).
Passed
Checking for SUNWilcs; found X11 ISO8859-1 Codeset Support(SUNWilcs). Passed
Checking for SUNWil5cs; found X11 ISO8859-15 Codeset Support(SUNWil5cs).
Passed
Checking for SUNWxwfont; found X Window System platform required fonts(SUNWxwfont).
Passed
Check complete. The overall result of this check is: Passed

Check complete: Passed
=====
Performing check for TotalMemory
Checking physical memory requirements ...
Expected result: 922MB
Actual Result: 3836MB
Check complete. The overall result of this check is: Passed

Check complete: Passed
=====
Performing check for SwapSpace
Checking available swap space requirements ...
Expected result: 3836MB
Actual Result: 5901MB
Check complete. The overall result of this check is: Passed

Check complete: Passed
=====
Performing check for OracleBase
Validating ORACLE_BASE location (if set) ...
Check complete. The overall result of this check is: Passed

Check complete: Passed
=====
Performing check for OracleHomeSpace
Checking Oracle Home path for spaces...
Check complete. The overall result of this check is: Passed

Check complete: Passed

```

```

=====
Performing check for DetectAnyInvalidASMHome
Checking for proper system clean-up...
Check complete. The overall result of this check is: Passed

Check complete: Passed
=====
Performing check for CompatibilityChecks
Checking for Oracle Home incompatibilities ....
Actual Result: NEW_HOME
Check complete. The overall result of this check is: Passed

Check complete: Passed
=====
PrereqChecks complete

..... 100% Done.

-----
Summary
Global Settings
  Source: /software/oracle.downloads/database/stage/products.xml
  Oracle Home: /oracle/app/oracle/product/10.2.0/db_1 (db_1)
  Installation Type: Enterprise Edition
Product Languages
  English
Space Requirements
  / Required 1.78GB : Available 14.29GB
  /tmp/ Required 124MB (only as temporary space) : Available 5.75GB
New Installations (108 products)
  Oracle Database 10g 10.2.0.1.0
  Enterprise Edition Options 10.2.0.1.0
  Oracle Partitioning 10.2.0.1.0
  Oracle Spatial 10.2.0.1.0
  Oracle OLAP 10.2.0.1.0
  Oracle Enterprise Manager Console DB 10.2.0.1.0
  Oracle Net Services 10.2.0.1.0
  Oracle Database 10g 10.2.0.1.0
  Oracle Net Listener 10.2.0.1.0
  HAS Files for DB 10.2.0.1.0
  Oracle Internet Directory Client 10.2.0.1.0
  Oracle Call Interface (OCI) 10.2.0.1.0
  Oracle Programmer 10.2.0.1.0
  Oracle interMedia 10.2.0.1.0
  Enterprise Manager Agent Core 10.2.0.1.0
  Oracle JVM 10.2.0.1.0
  Database Configuration and Upgrade Assistants 10.2.0.1.0
  Oracle interMedia Locator 10.2.0.1.0
  Oracle XML Development Kit 10.2.0.1.0
  Oracle Text 10.2.0.1.0
  Oracle Database Utilities 10.2.0.1.0
  Generic Connectivity Common Files 10.2.0.1.0
  Oracle Advanced Security 10.2.0.1.0
  Enterprise Manager Repository Core 10.2.0.1.0
  PL/SQL 10.2.0.1.0
  Oracle Net 10.2.0.1.0
  Assistant Common Files 10.2.0.1.0
  Oracle Notification Service 10.1.0.3.0
  Enterprise Manager plugin Common Files 10.2.0.1.0 Beta
  Buildtools Common Files 10.2.0.1.0

```


Installation Common Files 10.2.0.1.0
Oracle LDAP administration 10.2.0.1.0
Oracle Java Client 10.2.0.1.0
Precompiler Common Files 10.2.0.1.0
Oracle Recovery Manager 10.2.0.1.0
SQL*Plus 10.2.0.1.0
iSQL*Plus 10.2.0.1.0
Enterprise Manager plugin Common Files 10.2.0.1.0
HAS Common Files 10.2.0.1.0
Oracle Clusterware RDBMS Files 10.2.0.1.0
Oracle Wallet Manager 10.2.0.1.0
Enterprise Manager Minimal Integration 10.2.0.1.0
Oracle Database User Interface 2.2.13.0.0
Secure Socket Layer 10.2.0.1.0
Required Support Files 10.2.0.1.0
Oracle Globalization Support 10.2.0.1.0
Database SQL Scripts 10.2.0.1.0
OLAP SQL Scripts 10.2.0.1.0
PL/SQL Embedded Gateway 10.2.0.1.0
Oracle Locale Builder 10.2.0.1.0
Character Set Migration Utility 10.2.0.1.0
LDAP Required Support Files 10.2.0.1.0
Oracle Help for the Web 1.1.10.0.0
Oracle JDBC Thin Driver for JDK 1.4 10.2.0.1.0
Oracle JDBC Thin Driver for JDK 1.2 10.2.0.1.0
Oracle interMedia Client Option 10.2.0.1.0
Oracle Required Support Files 32 bit 10.2.0.0.0
Oracle Code Editor 1.2.1.0.OI
Perl Interpreter 5.8.3.0.2
JDBC Common Files 10.2.0.1.0
Oracle Help For Java 4.2.6.1.0
Oracle Containers for Java 10.2.0.1.0
Database Workspace Manager 10.2.0.1.0
Oracle Core Required Support Files 10.2.0.1.0
Platform Required Support Files 10.2.0.1.0
Oracle interMedia Locator RDBMS Files 10.2.0.1.0
Oracle JDBC/OCI Instant Client 10.2.0.1.0
Oracle interMedia Annotator 10.2.0.1.0
SQLJ Runtime 10.2.0.1.0
Oracle interMedia Java Advanced Imaging 10.2.0.1.0
Oracle Database 10g interMedia Files 10.2.0.1.0
Oracle Data Mining RDBMS Files 10.2.0.1.0
Enterprise Manager Baseline 10.2.0.1.0
Oracle Ice Browser 5.2.3.6.0
Oracle UIX 2.1.22.0.0
XML Parser for Java 10.2.0.1.0
Precompiler Required Support Files 10.2.0.1.0
XML Parser for Oracle JVM 10.2.0.1.0
Oracle Database 10G 32 bit 10.2.0.1.0
Oracle Message Gateway Common Files 10.2.0.1.0
Oracle Starter Database 10.2.0.1.0
Sample Schema Data 10.2.0.1.0
Parser Generator Required Support Files 10.2.0.1.0
Agent Required Support Files 10.2.0.1.0
Oracle RAC Required Support Files-HAS 10.2.0.1.0
RDBMS Required Support Files 10.2.0.1.0
RDBMS Required Support Files for Instant Client 10.2.0.1.0
XDK Required Support Files 10.2.0.1.0
Oracle OLAP API 10.2.0.1.0
Oracle OLAP RDBMS Files 10.2.0.1.0
DBJAVA Required Support Files 10.2.0.1.0
SQL*Plus Required Support Files 10.2.0.1.0
Oracle JFC Extended Windowing Toolkit 4.2.33.0.0

```

Oracle Extended Windowing Toolkit 3.4.38.0.0
Oracle Display Fonts 9.0.2.0.0
Bali Share 1.1.18.0.0
Enterprise Manager Common Files 10.2.0.1.0
Enterprise Manager Agent DB 10.2.0.1.0
Oracle Net Required Support Files 10.2.0.1.0
Enterprise Manager Repository DB 10.2.0.1.0
SSL Required Support Files for InstantClient 10.2.0.1.0
regexp 2.1.9.0.0
Oracle Universal Installer 10.2.0.1.0
Oracle One-Off Patch Installer 10.2.0.1.0
Installer SDK Component 10.2.0.1.0
Java Runtime Environment 1.4.2.0.0
Sun JDK 1.4.2.0.8
Sun JDK extensions 10.1.2.0.0
-----

Installation in progress (Wed Nov 18 17:21:07 GMT 2009)
..... 12% Done.
..... 25% Done.
..... 38% Done.
..... 51% Done.
..... 64% Done.
..... 74% Done.
Install successful

Linking in progress (Wed Nov 18 17:24:59 GMT 2009)
.. 74% Done.
Link successful

Setup in progress (Wed Nov 18 17:27:33 GMT 2009)
..... 100% Done.
Setup successful

End of install phases.(Wed Nov 18 17:27:38 GMT 2009)
WARNING:A new inventory has been created in this session. However, it has not yet
been registered as the central inventory of this system.
To register the new inventory please run the script
'/oracle/oraInventory/orainstRoot.sh' with root privileges.
If you do not register the inventory, you may not be able to update or patch the
products you installed.

The following configuration scripts
/oracle/app/oracle/product/10.2.0/db_1/root.sh
need to be executed as root for configuring the system. If you skip the execution
of the configuration tools, the configuration will not be complete and the product
wont function properly. In order to get the product to function properly, you will
be required to execute the scripts and the configuration tools after exiting the
OUI.

The installation of Oracle Database 10g was successful.
Please check '/oracle/oraInventory/logs/silentInstall2009-11-18_05-20-43PM.log'
for more details.

```

17. Run the Oracle root scripts:

```

root@zone-template01:/ # cd /oracle/app/oracle/product/10.2.0/db_1
root@zone-template01:/ # ./root.sh

```

18. Apply the latest patch sets to the Oracle 10.2 Database software:

```

oracle@zone-template01:~$ ftp updates.oracle.com

```

```

Connected to bigip-updates.oracle.com.
220 FTP server ready.
Name (updates.oracle.com:xyz): xyz@xxx
331 Username OK, please send password.
Password:
230-
230- Welcome to the Oracle Patch Download FTP Server
230-
230- For detailed help, use command "quote site help".
230
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd 6810189
250 Changed directory OK.
ftp> ls
200 PORT command OK.
150 Opening data connection for file listing.
total 1
p6810189_10204_AIX5L.zip
p6810189_10204_HPUX-IA64.zip
p6810189_10204_Linux-x86-64.zip
p6810189_10204_Linux-x86.zip
p6810189_10204_MSWIN-x86-64.zip
p6810189_10204_Solaris-64.zip
p6810189_10204_Win32.zip
226 Listing complete. Data connection has been closed.
ftp> get p6810189_10204_Solaris-64.zip
ftp> bye

oracle@zone-template01:~$ unzip p6810189_10204_Linux-x86-64.zip
oracle@zone-template01:~$ cd Disk1
oracle@zone-template01:~$ export SOFTWARE=`pwd`
oracle@zone-template01:~$ runInstaller -silent -responseFile
$SOFTWARE/response/patchset.rsp ORACLE_HOME=$ORACLE_HOME ORACLE_HOME_NAME=db_1

```

19. Run the Oracle root scripts again:

```

oracle@zone-template01:~$ su -
root@zone-template01:/ # cd /u01/app/oracle/product/10.2.0/db_1
root@zone-template01:/ # ./root.sh

```

20. Install SnapDrive for UNIX:

```

root@zone-template01:/ # ls
NTAPsnapdrive.pkg  snapdrive.admin  uninstall
install            snapdrive.ask

root@zone-template01:/ # ./install
Installing NTAPsnapdrive now ...
NTAPsnapdrive install completed successfully.
snapdrive Installation complete.  Log is in /tmp/snapdrive_install_log.19600.

root@zone-template01:/ # snapdrive config set root vcustomer01

```

21. Install SnapManager for Oracle:

```

root@zone-template01:/ # ./netapp.smo.sunos-x86-3.0.2.bin
Preparing to install...
Extracting the JRE from the installer archive...
Unpacking the JRE...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

```

```

InstallAnywhere will guide you through the installation of SnapManager for
Oracle, version 3.0.2

Enter Operating System User Name and Group Name

Please enter the operating system user name and group name that should be used
to run SnapManager for Oracle commands. Press <ENTER> to accept the default
values.

    User Name (DEFAULT: oracle):
    User Group (DEFAULT: dba):

Server Startup Type
-----

Whenever the host reboots, should the SnapManager for Oracle server start up
automatically?

i  ->1- Yes
    2- No
    3- (Return to Previous Step)

Configuration Summary
-----

Please review the following settings before continuing:

Installation directory (fixed location): /opt/NTAPsmo

User Id: oracle
User Group: dba
Restart SnapManager for Oracle server whenever the host reboots: Yes

SnapManager for Oracle Configured Successfully

```

22. To start the server component of SnapManager for Oracle, enter the following command logged in as root:

```
smo_server start
```

23. Once the server is started, you can launch the SnapManager for Oracle GUI do one of the following:

- To start the GUI on this host, enter the command **smogui**.
- To start the GUI from any host, enter the following URL in a web browser:
<https://zone-template01:27214>

The web browser automatically downloads and run the GUI.

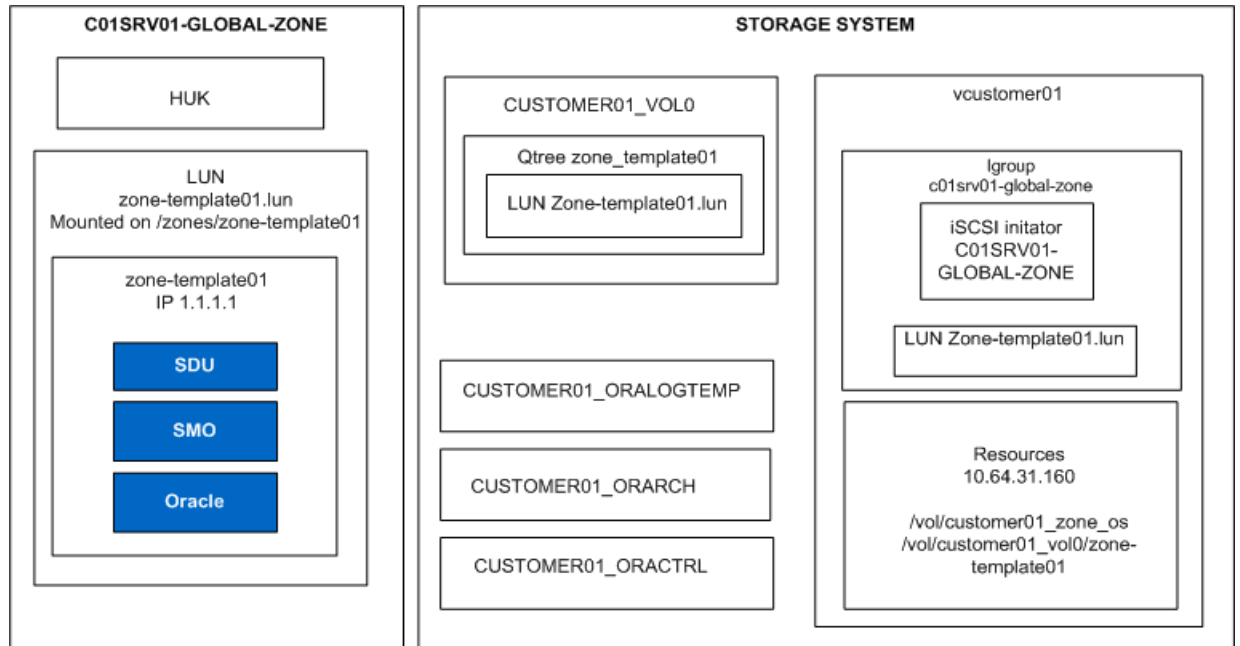


Figure 6) Creating a generic zone template (3).

24. After installing the software, reconfigure the template to remove the hostname, DNS, and so on.

```

root@zone-template01:/ # sys-unconfig
WARNING

This program will unconfigure your system. It will cause it
to revert to a "blank" system - it will not have a name or know
about other systems or networks.

This program will also halt the system.

Do you want to continue (y/n) ? y
svc.startd: The system is coming down. Please wait.
svc.startd: 81 system services are now being stopped.
svc.startd: The system is down.

[NOTICE: Zone halted]

```

25. Take a Snapshot copy of the file system containing the template. It will be used for cloning our boot LUNs:

```

root@c01srv01-global-zone:/ # zoneadm -z zone-template01 detach
root@c01srv01-global-zone:/ # umount /zones/zone-template01/

vcustomer01> snap create customer01_vol0 goldcopy

```

7.3 ADDING A NEW ZONE

This section describes how to create a new zone called `zone-test1` for a new customer named `customer01`.

Suppose each customer can have multiple Solaris servers with multiple zones but the zone names are unique among all its servers. This allows us to move a zone easily across servers; for example, `zone1` will only exist at any point in time in a single server.

To add a new zone:

1. Create a `qtree` for the new zone inside the `customer01_vol0` FlexVol volume, which contains the boot LUN of the zone:

```
filer> vfiler run vcustomer01 qtree create /vol/customer01_vol0/zone_test1
```

Note: Create a unique name for each zone.

2. Clone the boot LUN from the template:

```
filer> vfiler run vcustomer01 lun clone create
/vol/customer01_vol0/zone_test1/boot.lun -o noreserve -b
/vol/customer01_vol0/zone_template01/zone_template01.lun goldcopy
```

3. Add the boot LUN to the `igroup`:

```
filer> vfiler run vcustomer01 lun map /vol/customer01_vol0/zone_test1/boot.lun
c01srv01-global-zone
```

4. Log in to the host and reload the iSCSI driver:

```
root@c01srv01-global-zone:~# devfsadm -C

root@c01srv01-global-zone:~# /opt/NTAP/SANToolkit/bin/sanlun lun show | grep
zone_test1
    vcustomer01: /vol/customer01_vol0/zone_test1/boot.lun
/dev/rdisk/c2t60A98000433542746734537361596A51d0s2 scsi_vhci0 iSCSI      20g
(21474836480)      GOOD
vcustomer01: /vol/customer01_vol0/customer01_zone_test1/boot.lun
/dev/rdisk/c2t60A9800043354274673454584674677Ad0s2 scsi_vhci0 iSCSI      20g
(21474836480)      GOOD
root@c01srv01-global-zone:~#
root@c01srv01-global-zone:~#
```

5. Mount the new LUN:

```
root@c01srv01-global-zone:~# mkdir /zones/zone_test1
root@c01srv01-global-zone:~# mount
/dev/dsk/c2t60A9800043354274673454584674677Ad0s2 /zones/zone_test1
root@c01srv01-global-zone:~# chmod 700 /zones/zone_test1
```

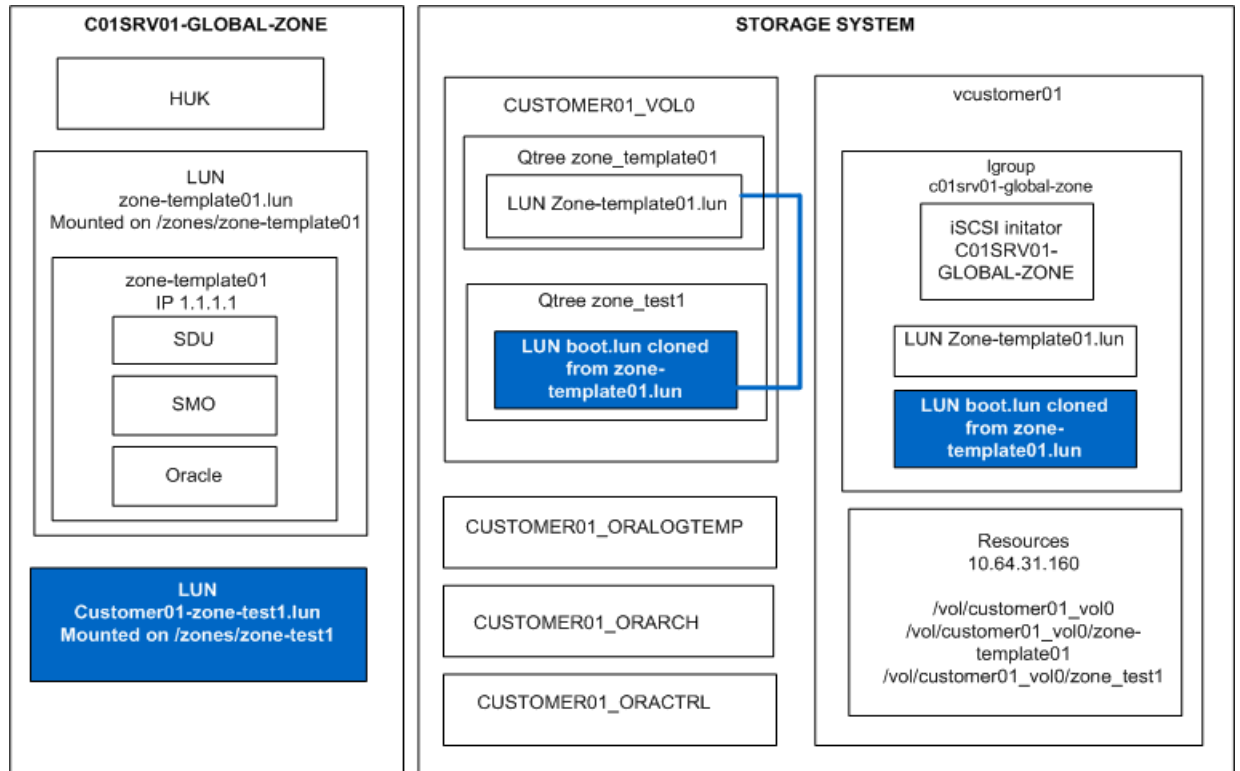


Figure 7) Adding a new zone (1).

- Build a zone file. Set the path and the correct IP address:

```

root@c01srv01-global-zone:/ # cat >/tmp/zone.cfg << EOF
create -b
set zonepath=/zones/zone_test1
set autoboot=false
set pool=pool1
set ip-type=shared
add net
set address=10.64.28.192
set physical=e1000g0
end
EOF
root@c01srv01-global-zone:/ # zonecfg -z zone_test1 -f /tmp/zone.cfg

```

Note that we skip the install bit, because we have a fully installed clone LUN:

```

root@c01srv01-global-zone:/ # zoneadm -z zone_test1 attach
root@c01srv01-global-zone:/ # zoneadm -z zone_test1 boot

```

- Log in and configure the OS, hostname, DNS, and so on:

```

root@c01srv01-global-zone:/ # zlogin -C zone_test1
[Connected to zone 'zone_test1' console]

SunOS Release 5.10 Version Generic_120012-14 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: zone_test1

```

```

Select a Language

0. English
1. French
2. German
3. Italian
4. Japanese
5. Korean
6. Simplified Chinese
7. Spanish
8. Swedish
9. Traditional Chinese

Please make a choice (0 - 9), or press h or ? for help:
.

```

After the zone is added, the following message appears:

```

System identification is completed.
rebooting system due to change(s) in /etc/default/init

[NOTICE: Zone rebooting]

```

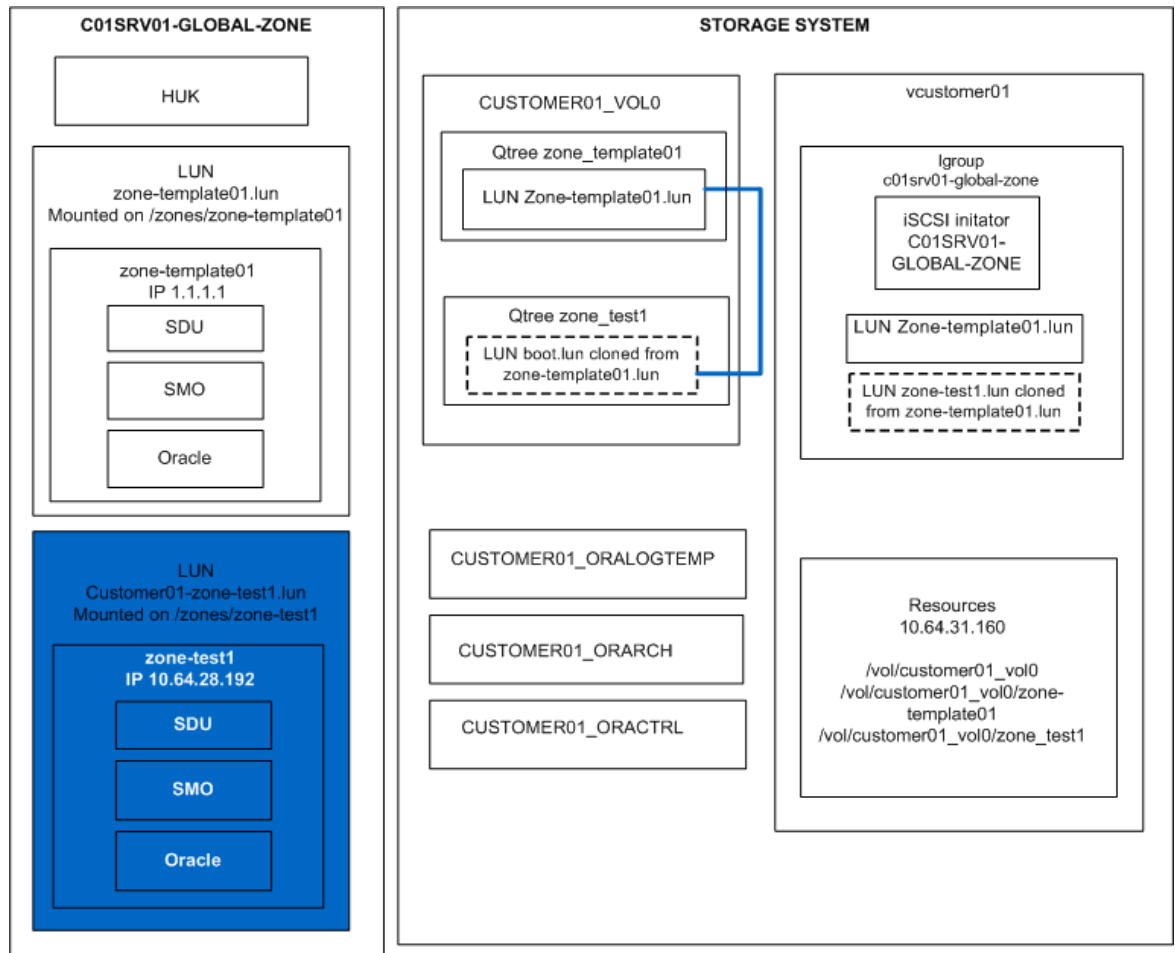


Figure 8 Adding a new zone (2).

7.4 CREATING A NEW DATABASE ENVIRONMENT

1. Create new qtrees for the new zone:

```
filer> qtree create /vol/customer01_oralogtemp/zone_test1
filer> qtree create /vol/customer01_orarch/zone_test1
filer> qtree create /vol/customer01_oractrl/zone_test1
```

2. Build a dedicated FlexVol volume to contain the Oracle data files.

This enables restoring the FlexVol volumes quickly and immediately.

Suppose the SID is named SID1; the FlexVol volume will be named:

```
oradata_<customer#>_<zone_name>_SID1
```

A customer can have the same SID number running on different zones:

```
vol create /vol/customer01_oradata_zone_test1_SID1 aggr0 10g
vol options customer01_oradata_zone_test1_SID1 guarantee none
vol options customer01_oradata_zone_test1_SID1 nosnapdir on
vol options customer01_oradata_zone_test1_SID1 nosnap on
snap sched customer01_oradata_zone_test1_SID1 0 0 0
```

3. Add the qtrees and FlexVol volumes to the vcustomer01 vFiler unit:

```
vfiler add vcustomer01 /vol/customer01_oralogtemp
vfiler add vcustomer01 /vol/customer01_orarch
vfiler add vcustomer01 /vol/customer01_oractrl
vfiler add vcustomer01 /vol/customer01_oradata_zone_test1_SID1
```

4. Export the file systems Read Write to the zone IP address.

The IP address of the zone is 10.64.28.192.

```
vfiler run vcustomer01 exportfs -p rw=10.64.28.192,anon=0
/vol/customer01_oralogtemp/zone_test1
vfiler run vcustomer01 exportfs -p rw=10.64.28.192,anon=0
/vol/customer01_orarch/zone_test1
vfiler run vcustomer01 exportfs -p rw=10.64.28.192,anon=0
/vol/customer01_oractrl/zone_test1
vfiler run vcustomer01 exportfs -p rw=10.64.28.192,anon=0
/vol/customer01_oradata_zone_test1_SID1
vfiler run vcustomer01 exportfs -a
```

5. Log in to the zone and create the Oracle directories. Then, edit the `/etc/vfstab` file and perform `mount -a` to mount the volumes.

```
root@zone-test1:/ # mkdir -p /oracle/oradata/SID1
root@zone-test1:/ # mkdir -p /oracle/oralogtemp
root@zone-test1:/ # mkdir -p /oracle/orarch
root@zone-test1:/ # mkdir -p /oracle/oractrl
root@zone-test1:/ # echo >> /etc/vfstab
root@zone-test1:/ # echo 'vcustomer01:/vol/customer01_oradata_zone_test1_SID1 -
/oracle/oradata/SID1 nfs - yes
rw,bg,hard,rsize=32768,wsiz=32768,vers=3,nointr,proto=tcp,suid,forcedirectio' >>
/etc/vfstab
root@zone-test1:/ # echo 'vcustomer01:/vol/customer01_oralogtemp/zone_test1 -
/oracle/oralogtemp nfs - yes
rw,bg,hard,rsize=32768,wsiz=32768,vers=3,nointr,proto=tcp,suid,forcedirectio' >>
/etc/vfstab
root@zone-test1:/ # echo 'vcustomer01:/vol/customer01_orarch/zone_test1 -
/oracle/orarch nfs - yes
```

```

rw,bg,hard,rsz=32768,wsz=32768,vers=3,nointr,proto=tcp,suid,forcedirectio' >>
/etc/vfstab
root@zone-test1:/ # echo 'vcustomer01:/vol/customer01_oractrl/zone_test1 -
/oracle/oractrl nfs - yes
rw,bg,hard,rsz=32768,wsz=32768,vers=3,nointr,proto=tcp,suid,forcedirectio' >>
/etc/vfstab
root@zone-test1:/ # mount -a

```

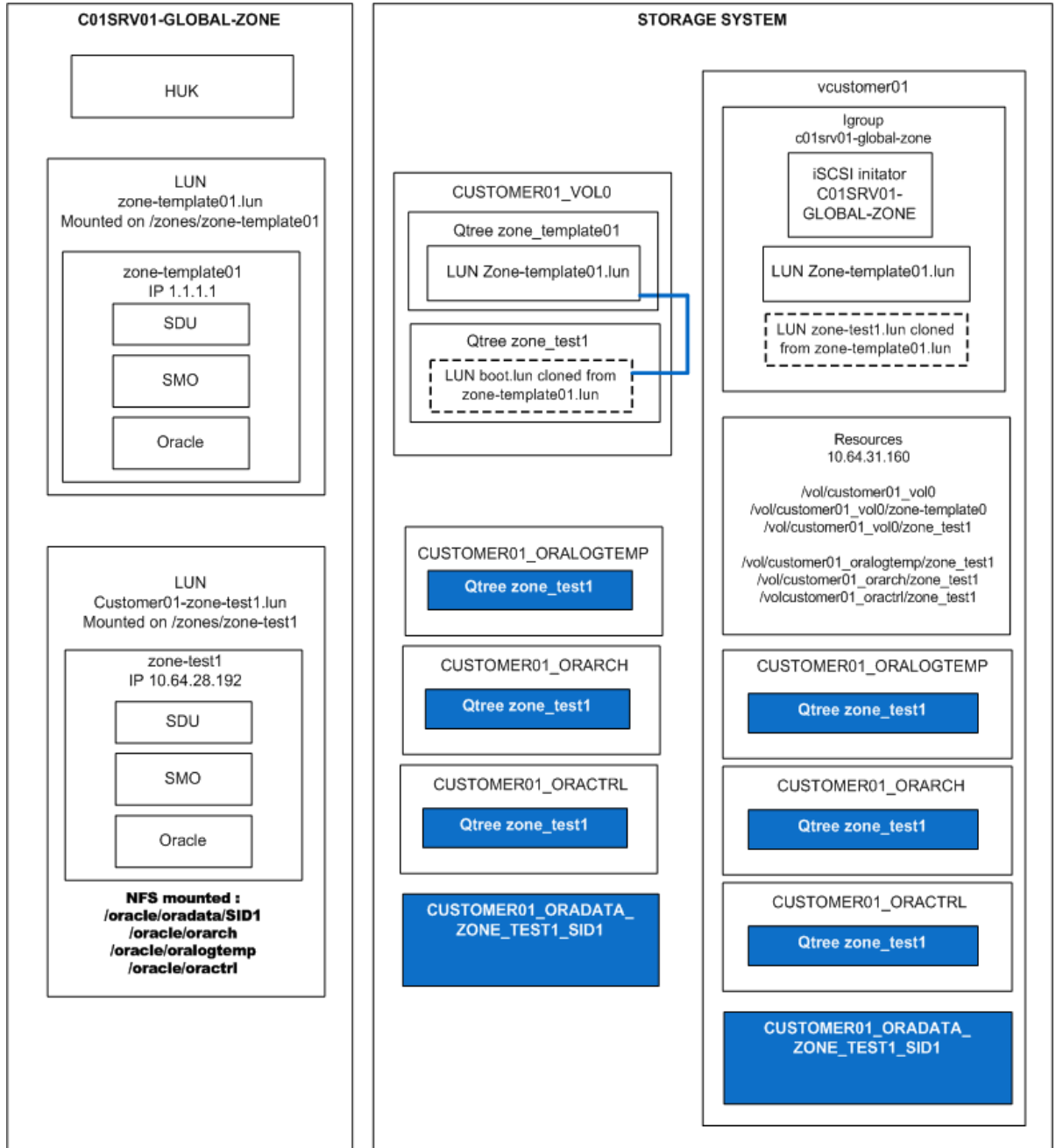


Figure 9) Creating a database environment.

- Set permissions on the mount points:

```

root@zone-test1:/ # chown oracle:dba /oracle/oradata
root@zone-test1:/ # chown oracle:dba /oracle/oradata/SID1

```

```

root@zone-test1:/ # chown oracle:dba /oracle/oralogtemp
root@zone-test1:/ # chown oracle:dba /oracle/orarch
root@zone-test1:/ # chown oracle:dba /oracle/oractrl
root@zone-test1:/ # chmod 777 /oracle/oradata/SID1

root@zone-test1:/ # mkdir /oracle/oralogtemp/SID1
root@zone-test1:/ # mkdir /oracle/orarch/SID1
root@zone-test1:/ # mkdir /oracle/oractrl/SID1

root@zone-test1:/ # chown oracle:dba /oracle/oralogtemp/SID1
root@zone-test1:/ # chown oracle:dba /oracle/orarch/SID1
root@zone-test1:/ # chown oracle:dba /oracle/oractrl/SID1

```

7. Log in as oracle user:

```

root@zone-test1:/ # su - oracle

```

8. Create a dbca template.

This template is used to place the Oracle Database objects in the correct locations.

```

oracle@zone-test1:~$ cat > $ORACLE_HOME/assistants/dbca/templates/SolarisV1.3.dbc
<<EOF
<?xml version = '1.0'?>
<DatabaseTemplate name="SolarisZoneV1.3" description="" version="10.2.0.0.0">
  <CommonAttributes>
    <option name="ISEARCH" value="false"/>
    <option name="OMS" value="false"/>
    <option name="JSERVER" value="true"/>
    <option name="SPATIAL" value="true"/>
    <option name="ODM" value="true">
      <tablespace id="SYSAUX"/> </option>
    <option name="IMEDIA" value="true"/>
    <option name="XDB_PROTOCOLS" value="true">
      <tablespace id="SYSAUX"/>
    </option>
    <option name="ORACLE_TEXT" value="true">
      <tablespace id="SYSAUX"/>
    </option>
    <option name="SAMPLE_SCHEMA" value="false"/>
    <option name="CWMLITE" value="true">
      <tablespace id="SYSAUX"/>
    </option>
    <option name="EM_REPOSITORY" value="true">
      <tablespace id="SYSAUX"/>
    </option>
  </CommonAttributes>
  <Variables/>
  <CustomScripts Execute="false"/>
  <InitParamAttributes>
    <InitParams>
      <initParam name="pga_aggregate_target" value="229" unit="MB"/>
      <initParam name="log_archive_dest_1"
value=" 'LOCATION={ORACLE_BASE}/orarch/{DB_NAME}' " />
      <initParam name="processes" value="150" />
      <initParam name="db_create_online_log_dest_1"
value="{ORACLE_BASE}/oralogtemp/{DB_NAME}" />
      <initParam name="control_files"
value=" (&quot;{ORACLE_BASE}/oractrl/{DB_NAME}/control01.ctl&quot;;
&quot;{ORACLE_BASE}/oractrl/{DB_NAME}/control02.ctl&quot;)" />
      <initParam name="sga_target" value="689" unit="MB"/>
      <initParam name="compatible" value="10.2.0.1.0"/>

```

```

    <initParam name="background_dump_dest"
value="{ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/bdump"/>
    <initParam name="job_queue_processes" value="10"/>
    <initParam name="user_dump_dest"
value="{ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/udump"/>
    <initParam name="dispatchers" value="(PROTOCOL=TCP) (SERVICE={SID}XDB)"/>
    <initParam name="audit_file_dest"
value="{ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/adump"/>
    <initParam name="open_cursors" value="300"/>
    <initParam name="db_block_size" value="8" unit="KB"/>
    <initParam name="undo_tablespace" value="UNDOTBS1"/>
    <initParam name="log_archive_format" value="%t_%s_%r.dbf"/>
    <initParam name="core_dump_dest"
value="{ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/cdump"/>
    <initParam name="remote_login_passwordfile" value="EXCLUSIVE"/>
    <initParam name="undo_management" value="AUTO"/>
    <initParam name="db_file_multiblock_read_count" value="16"/>
  </InitParams>
  <MiscParams>
    <databaseType>MULTIPURPOSE</databaseType>
    <maxUserConn>20</maxUserConn>
    <percentageMemTOSGA>25</percentageMemTOSGA>
    <customSGA>>false</customSGA>
    <archiveLogMode>>true</archiveLogMode>

<initParamFileName>{ORACLE_BASE}/admin/{DB_UNIQUE_NAME}/pfile/init.ora</initParamF
ileName>
  </MiscParams>
  <SPfile useSPfile="true">{ORACLE_HOME}/dbs/spfile{SID}.ora</SPfile>
</InitParamAttributes>
<StorageAttributes>
  <DataFiles>

<Location>{ORACLE_HOME}/assistants/dbca/templates/Seed_Database.dfb</Location>
  <SourceDBName>seeddata</SourceDBName>
  <Name id="1" Tablespace="SYSTEM" Contents="PERMANENT" Size="470"
autoextend="true"
blocksize="8192">{ORACLE_BASE}/oradata/{DB_NAME}/system01.dbf</Name>
  <Name id="2" Tablespace="UNDOTBS1" Contents="UNDO" Size="25"
autoextend="true"
blocksize="8192">{ORACLE_BASE}/oradata/{DB_NAME}/undotbs01.dbf</Name>
  <Name id="3" Tablespace="SYSAUX" Contents="PERMANENT" Size="230"
autoextend="true"
blocksize="8192">{ORACLE_BASE}/oradata/{DB_NAME}/sysaux01.dbf</Name>
  <Name id="4" Tablespace="USERS" Contents="PERMANENT" Size="5"
autoextend="true"
blocksize="8192">{ORACLE_BASE}/oradata/{DB_NAME}/users01.dbf</Name>
  </DataFiles>
  <TempFiles>
    <Name id="1" Tablespace="TEMP" Contents="TEMPORARY"
Size="20">{ORACLE_BASE}/oralogtemp/{DB_NAME}/temp01.dbf</Name>
  </TempFiles>
  <ControlfileAttributes id="Controlfile">
    <maxDatafiles>100</maxDatafiles>
    <maxLogfiles>16</maxLogfiles>
    <maxLogMembers>3</maxLogMembers>
    <maxLogHistory>1</maxLogHistory>
    <maxInstances>8</maxInstances>
    <image name="control01.ctl" filepath="{ORACLE_BASE}/oractrl/{DB_NAME}/"/>
    <image name="control02.ctl" filepath="{ORACLE_BASE}/oractrl/{DB_NAME}/"/>
  </ControlfileAttributes>
  <RedoLogGroupAttributes id="1">
    <reuse>>false</reuse>
    <fileSize unit="KB">51200</fileSize>
    <Thread>1</Thread>

```

```

        <member ordinal="0" memberName="redo01.log"
filepath="{ORACLE_BASE}/oralogtemp/{DB_NAME}/"/>
        </RedoLogGroupAttributes>
        <RedoLogGroupAttributes id="2">
        <reuse>>false</reuse>
        <fileSize unit="KB">51200</fileSize>
        <Thread>1</Thread>
        <member ordinal="0" memberName="redo02.log"
filepath="{ORACLE_BASE}/oralogtemp/{DB_NAME}/"/>
        </RedoLogGroupAttributes>
        <RedoLogGroupAttributes id="3">
        <reuse>>false</reuse>
        <fileSize unit="KB">51200</fileSize>
        <Thread>1</Thread>
        <member ordinal="0" memberName="redo03.log"
filepath="{ORACLE_BASE}/oralogtemp/{DB_NAME}/"/>
        </RedoLogGroupAttributes>
        </StorageAttributes>
</DatabaseTemplate>

EOF

```

- Build the Oracle instance using dbca:

```

oracle@zone-test1:~$ dbca -silent -createDatabase -templateName SolarisV1.3.dbc
-gdbName SID1 -sid SID1 -syspassword netapp -systempassword netapp -characterSet
WE8ISO8859P15

```

- Repeat steps 1–10 to create the zones and database instances required.

3.5 CONFIGURING SNAPMANAGER FOR ORACLE

This example describes how to configure SnapManager for Oracle.

Suppose there are two zones— **zone-test1** and **zone-test2** (see Figure 10) with a total of four Oracle instances between them. **SID1** on zone_test1 is the **SMODB repository**, and the remaining Oracle instances will be backed up by SMO.

```
ssh root@zone-test1 ps -ef | grep pmon
oracle 2871 1755 0 12:45:14 ?          0:08 ora_pmon_SID1
ssh root@zone-test2 ps -ef | grep pmon
oracle 4312 3303 0 13:05:30 ?          0:01 ora_pmon_SID2
oracle 4538 3303 0 13:12:26 ?          0:01 ora_pmon_SID3
oracle 4774 3303 0 13:23:54 ?          0:00 ora_pmon_SID4
```

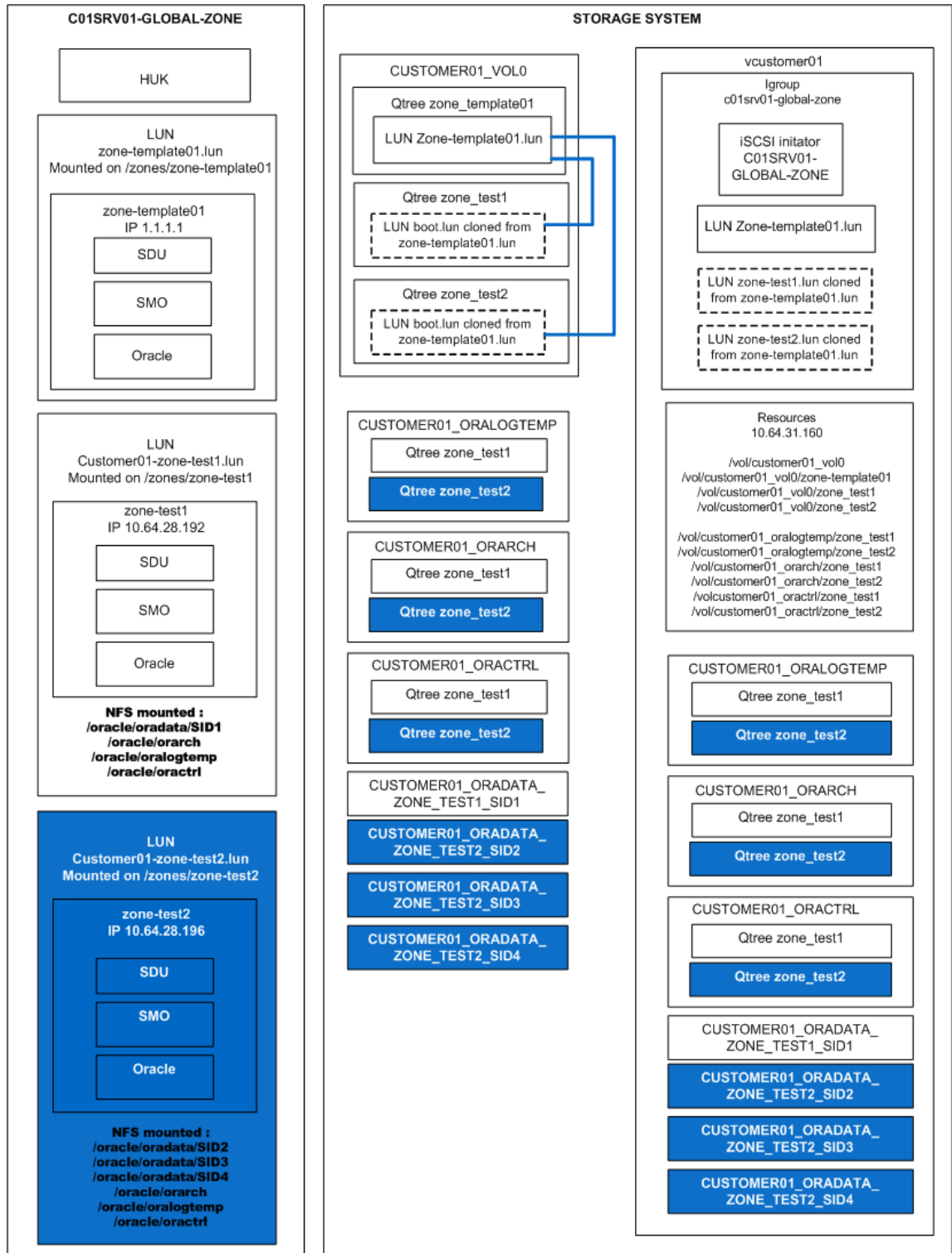


Figure 10) Configuring SMO.

To configure SnapManager for Oracle:

1. Log in to the storage system and enable `vfiler.vol_clone_zapi_allow`. This option enables building FlexClone[®] volumes from within the vFiler unit:

```
Filer> option vfiler.vol_clone_zapi_allow on
```

2. Log in to zone-test1 and edit the tnsnames.ora file as follows:

```
oracle@zone-test1:~$ vi
/oracle/app/oracle/product/10.2.0/db_1/network/admin/tnsnames.ora

SID1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.64.28.192)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = SID1)
    )
  )
)

SID2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.64.28.196)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = SID2)
    )
  )
)

SID3 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.64.28.196)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = SID3)
    )
  )
)

SID3 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.64.28.196)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = SID3)
    )
  )
)

SID4 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.64.28.196)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = SID4)
    )
  )
)
```

3. Copy this file to zone-test2 and start the listener on both zones:

```
oracle@zone-test1:~$ lsrnctl start
oracle@zone-test2:~$ lsrnctl start
```

4. Log in to zone-test1 again and create an Oracle user named smoadmin:

```
oracle@zone-test1:~$ export ORACLE_SID=SID1
```



```
oracle@zone-test1:~$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.1.0 - Production on Wed Nov 4 14:37:20 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> create user smoadmin identified by netapp;
User created.
SQL> grant resource, connect, create session to smoadmin;
Grant succeeded.
```

SETTING UP RMAN

1. Create the RMAN catalog:

```
SQL> create tablespace RMAN datafile '/oracle/oradata/SID1/rman01.dbf' size 10M
autoextend on;
Tablespace created.

SQL> create user rman identified by netapp default tablespace RMAN quota unlimited
on RMAN;
User created.

SQL> grant connect, resource, create session, recovery_catalog_owner to rman;
Grant succeeded.

oracle@zone-test1:~$ rman catalog=rman/netapp@SID1
Recovery Manager: Release 10.2.0.1.0 - Production on Wed Nov 4 15:09:41 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.
connected to recovery catalog database
RMAN> create catalog tablespace "RMAN";
recovery catalog created
RMAN> quit
Recovery Manager complete.
```

2. With the RMAN catalog in place, register the databases in RMAN. Log in to zone_test2 and enter:

```
oracle@zone-test2:~$ export ORACLE_SID=SID2
oracle@zone-test2:~$ rman catalog rman/netapp@SID1 target /

Recovery Manager: Release 10.2.0.1.0 - Production on Wed Nov 4 15:13:19 2009
Copyright (c) 1982, 2005, Oracle. All rights reserved.

connected to target database: SID2 (DBID=404903879)
connected to recovery catalog database

RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
RMAN> quit
Recovery Manager complete.
```

CREATING AN SMO REPOSITORY ON THE SID1 ORACLE INSTANCE

Log in to zone-test1 and do the following:

```
oracle@zone-test1:~$ smo repository create -repository -port 1521 -dbname SID1 -host
zone-test1 -login -username smoadmin
Enter password for database connection smoadmin@zone-test1:1521/SID1: *****
[ INFO] SMO-20019: Set password for repository "smoadmin@SID1/zone-test1:1521" in user
credentials for "oracle".
[ INFO] SMO-09202: Creating new schema as smoadmin on jdbc:oracle:thin:@//zone-
test1:1521/SID1.
[ INFO] SMO-09205: Schema generation complete.
[ INFO] SMO-09209: Performing repository version INSERT.
[ INFO] SMO-09210: Repository created with version: 64
[ INFO] SMO-13048: Repository Create Operation Status: SUCCESS
[ INFO] SMO-13049: Elapsed Time: 0:00:10.549
oracle@zone-test1:~$
```

CREATING AN SMO PROFILE FOR THE SID2 DATABASE

```
oracle@zone-test1:~$ smo profile create -profile SID2 -repository -dbname SID1 -host
zone-test1 -port 1521 -login -username smoadmin -database -dbname SID2 -host zone-test2
-osaccount oracle -osgroup dba -rman -login -username rman -password netapp -tnsname
SID1 -verbose
Enter password for profile SID2: *****
[ INFO] SMO-20020: Set password for profile "SID2" in user credentials for "oracle".
Enter password for user oracle@zone-test2: *****
[ INFO] SMO-13046: Operation GUID 8ac09c4424bfce370124bfce44b1000a starting on Profile
SID2
[ INFO] SMO-13505: SnapDrive environment verification passed.
[ WARN] SMO-13521: Minimum supported version for instance type "rdbms" not found.
[ INFO] SMO-13507: JDBC verification for "OS authenticated: SID2/zone-test2" passed.
[ WARN] SMO-13521: Minimum supported version for instance type "rdbms" not found.
[ INFO] SMO-13506: SQLPlus verification for database SID "SID2" passed. Environment:
[ORACLE_HOME=/oracle/app/oracle/product/10.2.0/db_1]
[ INFO] SMO-13509: RMAN verification using catalog "SID1" passed.
[ INFO] SMO-07431: Saving starting state of the database: SID2(OPEN).
[ INFO] SMO-07431: Saving starting state of the database: SID2(OPEN).
[ INFO] SD-00016: Discovering storage resources for /oracle/oradata/SID2.
[ INFO] SD-00017: Finished storage discovery for /oracle/oradata/SID2.
[ INFO] SD-00016: Discovering storage resources for /oracle/oralogtemp.
[ INFO] SD-00017: Finished storage discovery for /oracle/oralogtemp.
[ INFO] SD-00016: Discovering storage resources for /oracle/oralogtemp.
[ INFO] SD-00017: Finished storage discovery for /oracle/oralogtemp.
[ INFO] SD-00016: Discovering storage resources for /oracle/oractrl.
[ INFO] SD-00017: Finished storage discovery for /oracle/oractrl.
[ INFO] SMO-05070: Database profile SID2 is eligible for fast restore.
[ INFO] SMO-07433: Returning the database to its initial state: SID2(OPEN).
[ INFO] SMO-13048: Profile Create Operation Status: SUCCESS
[ INFO] SMO-13049: Elapsed Time: 0:01:15.122
Operation Id [8ac09c4424bfce370124bfce44b1000a] succeeded.
```

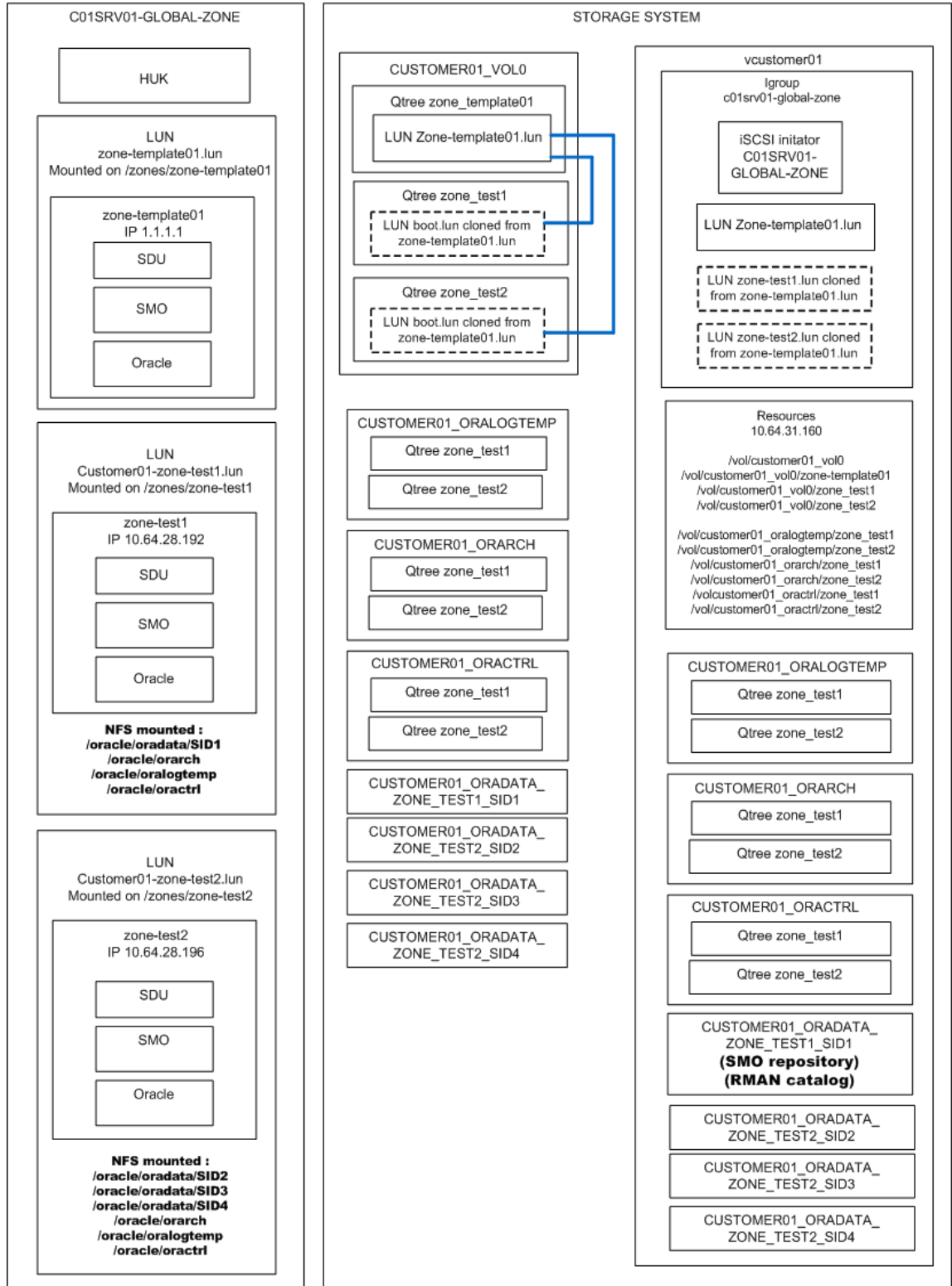


Figure 11) Creating an SMO profile.

TAKING A BACKUP

```
oracle@zone-test1:~$ smo backup create -profile SID2 -auto -full -verbose
Enter password for user oracle@zone-test2: *****
[ INFO] SMO-13046: Operation GUID 8ac09c4425bd0e8f0125bd0ea1900001 starting on Profile
SID2
[ INFO] SMO-02007: Auto-generated label "F_A_20091223194100GMT" for backup.
[ INFO] SMO-07431: Saving starting state of the database: SID2(OPEN).
[ INFO] SMO-07431: Saving starting state of the database: SID2(OPEN).
[ INFO] SMO-07127: Locked database for SnapManager operations - created lock file
"/oracle/app/oracle/product/10.2.0/db_1/dbs/.sm_lock_SID2" on host zone-test2.
[ INFO] ORACLE-20024: Spooling control file for database SID2 to trace as
/var/tmp/SM_23097.
[ INFO] ORACLE-20024: Spooling control file for database SID2 to trace as
/var/tmp/SM_23098.
[ INFO] ORACLE-20019: Placing database SID2 into online backup mode.
[ INFO] SD-00016: Discovering storage resources for /oracle/oradata/SID2.
[ INFO] SD-00017: Finished storage discovery for /oracle/oradata/SID2.
[ INFO] SD-00001: Beginning snapshot with name
smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0 for [/oracle/oradata/SID2].
[ INFO] SD-00002: Created snapshot with name
smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0 for [/oracle/oradata/SID2].
[ INFO] SD-00022: Querying for snapshot
vcustomer01:/vol/customer01_oradata_zone_test2_SID2:smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f
0125bd0ea1900001_0.
[ INFO] SD-00023: Finished querying for snapshot
vcustomer01:/vol/customer01_oradata_zone_test2_SID2:smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f
0125bd0ea1900001_0.
[ INFO] ORACLE-20020: Taking database SID2 out of online backup mode.
[ INFO] ORACLE-20025: Forcing archival of all online redo logs for database SID2.
[ INFO] ORACLE-20023: Backing up control file for database SID2 to
/oracle/oractrl/SID2/SMOBakCtl_1261597324292_0.
[ INFO] ORACLE-20023: Backing up control file for database SID2 to
/oracle/oractrl/SID2/SMOBakCtl_1261597326114_1.
[ INFO] SD-00016: Discovering storage resources for /oracle/oractrl.
[ INFO] SD-00017: Finished storage discovery for /oracle/oractrl.
[ INFO] SD-00016: Discovering storage resources for /oracle/orarch.
[ INFO] SD-00017: Finished storage discovery for /oracle/orarch.
[ INFO] SD-00001: Beginning snapshot with name
smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_0 for [/oracle/oractrl,
/oracle/orarch].
[ INFO] SD-00002: Created snapshot with name
smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_0 for [/oracle/oractrl,
/oracle/orarch].
[ INFO] SD-00022: Querying for snapshot
vcustomer01:/vol/customer01_orarch:smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_
0.
[ INFO] SD-00023: Finished querying for snapshot
vcustomer01:/vol/customer01_orarch:smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_
0.
[ INFO] SD-00022: Querying for snapshot
vcustomer01:/vol/customer01_oractrl:smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_
_0.
[ INFO] SD-00023: Finished querying for snapshot
vcustomer01:/vol/customer01_oractrl:smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_
_0.
[ INFO] SMO-07108: Deleting backup control file
"/oracle/oractrl/SID2/SMOBakCtl_1261597324292_0".
[ INFO] SMO-07108: Deleting backup control file
"/oracle/oractrl/SID2/SMOBakCtl_1261597326114_1".
[ INFO] SD-00025: Beginning to connect filesystem(s) [/oracle/oradata/SID2] from
snapshot smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0.
[ INFO] SD-00016: Discovering storage resources for /opt/NTAPsmo/mnt/-oracle-oradata-
SID2-20091223194255867_0.
[ INFO] SD-00017: Finished storage discovery for /opt/NTAPsmo/mnt/-oracle-oradata-SID2-
20091223194255867_0.
```

```

[ INFO] SD-00026: Finished connecting filesystem(s) [/oracle/oradata/SID2] from snapshot
smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0.
[ INFO] SD-00025: Beginning to connect filesystem(s) [/oracle/oractrl, /oracle/orarch]
from snapshot smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_0.
[ INFO] SD-00016: Discovering storage resources for /opt/NTAPsmo/mnt/-oracle-oractrl-
20091223194255904_0.
[ INFO] SD-00017: Finished storage discovery for /opt/NTAPsmo/mnt/-oracle-oractrl-
20091223194255904_0.
[ INFO] SD-00016: Discovering storage resources for /opt/NTAPsmo/mnt/-oracle-orarch-
20091223194255854_0.
[ INFO] SD-00017: Finished storage discovery for /opt/NTAPsmo/mnt/-oracle-orarch-
20091223194255854_0.
[ INFO] SD-00026: Finished connecting filesystem(s) [/oracle/oractrl, /oracle/orarch]
from snapshot smo_sid2_sid2_f_h_2_8ac09c4425bd0e8f0125bd0ea1900001_0.
[ INFO] SMO-07109: Cataloguing all files in backup set with RMAN,
TAG=SMO_F_A_20091223_1261597421735, RMAN=rman@SID1/catalog.
[ INFO] ORACLE-00115: Cataloguing file set [/opt/NTAPsmo/mnt/-oracle-oradata-SID2-
20091223194255867_0/system01.dbf, /opt/NTAPsmo/mnt/-oracle-oractrl-
20091223194255904_0/SID2/SMOBakCtl_1261597326114_1, /opt/NTAPsmo/mnt/-oracle-oradata-
SID2-20091223194255867_0/users01.dbf, /opt/NTAPsmo/mnt/-oracle-oradata-SID2-
20091223194255867_0/undotbs01.dbf, /opt/NTAPsmo/mnt/-oracle-oradata-SID2-
20091223194255867_0/sysaux01.dbf, /opt/NTAPsmo/mnt/-oracle-oractrl-
20091223194255904_0/SID2/SMOBakCtl_1261597324292_0] with RMAN.
[ INFO] SD-00031: Beginning to disconnect filesystem(s) [/opt/NTAPsmo/mnt/-oracle-
oradata-SID2-20091223194255867_0, /opt/NTAPsmo/mnt/-oracle-oractrl-20091223194255904_0,
/opt/NTAPsmo/mnt/-oracle-orarch-20091223194255854_0].
[ INFO] SD-00032: Finished disconnecting filesystem(s) [/opt/NTAPsmo/mnt/-oracle-
oradata-SID2-20091223194255867_0, /opt/NTAPsmo/mnt/-oracle-oractrl-20091223194255904_0,
/opt/NTAPsmo/mnt/-oracle-orarch-20091223194255854_0].
[ INFO] SMO-07131: Unlocked database for SnapManager operations - removed lock file
"/oracle/app/oracle/product/10.2.0/db_1/dbs/.sm_lock_SID2" on host zone-test2.
[ INFO] SMO-07433: Returning the database to its initial state: SID2(OPEN).
[ INFO] SMO-13048: Backup Create Operation Status: SUCCESS
[ INFO] SMO-13049: Elapsed Time: 0:03:23.197
Operation Id [8ac09c4425bd0e8f0125bd0ea1900001] succeeded.

```

PERFORMING A RESTORE

```

oracle@zone-test1:~$ smo backup restore -profile SID2 -label F_A_20091223194100GMT -
complete -recover -alllogs -force -fast -verbose
Enter password for user oracle@zone-test2: *****
[ INFO] SMO-13046: Operation GUID 8ac09c4425bd43e20125bd43f05d0001 starting on Profile
SID2
[ INFO] SMO-07431: Saving starting state of the database: SID2(OPEN).
[ INFO] SMO-07431: Saving starting state of the database: SID2(OPEN).
[ INFO] SMO-07127: Locked database for SnapManager operations - created lock file
"/oracle/app/oracle/product/10.2.0/db_1/dbs/.sm_lock_SID2" on host zone-test2.
[ INFO] ORACLE-20000: Changing state for database instance SID2 from OPEN to MOUNTED.
[ INFO] ORACLE-20009: Attempting to reconnect to instance SID2 after shutdown/startup.
[ INFO] ORACLE-20011: Reconnect to instance SID2 successful.
[ INFO] SMO-07200: Beginning restore of database "SID2".
[ INFO] SD-00022: Querying for snapshot
vcustomer01:/vol/customer01_oradata_zone_test2_SID2:smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f
0125bd0ea1900001_0.
[ INFO] SD-00023: Finished querying for snapshot
vcustomer01:/vol/customer01_oradata_zone_test2_SID2:smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f
0125bd0ea1900001_0.
[ INFO] SD-00016: Discovering storage resources for /oracle/oralogtemp.
[ INFO] SD-00017: Finished storage discovery for /oracle/oralogtemp.
[ INFO] SD-00016: Discovering storage resources for /oracle/orarch.
[ INFO] SD-00017: Finished storage discovery for /oracle/orarch.
[ INFO] SD-00016: Discovering storage resources for /oracle/oralogtemp.
[ INFO] SD-00017: Finished storage discovery for /oracle/oralogtemp.
[ INFO] SD-00016: Discovering storage resources for /oracle/oractrl.
[ INFO] SD-00017: Finished storage discovery for /oracle/oractrl.

```

```

[ INFO] SD-00052: Beginning preview of volume restore of
[vcustomer01:/vol/customer01_oradata_zone_test2_SID2] (with host-side resources
[/oracle/oradata/SID2]) from snapshots
[[smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0]]
[ INFO] SD-00053: Finished preview of volume restore of
[vcustomer01:/vol/customer01_oradata_zone_test2_SID2] (with host-side resources
[/oracle/oradata/SID2]) from snapshots
[[smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0]]
[ INFO] SD-00052: Beginning preview of volume restore of
[vcustomer01:/vol/customer01_oradata_zone_test2_SID2] (with host-side resources
[/oracle/oradata/SID2]) from snapshots
[[smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0]]
[ INFO] SD-00053: Finished preview of volume restore of
[vcustomer01:/vol/customer01_oradata_zone_test2_SID2] (with host-side resources
[/oracle/oradata/SID2]) from snapshots
[[smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0]]
[ INFO] SD-00054: Beginning volume restore of
[vcustomer01:/vol/customer01_oradata_zone_test2_SID2] (with host-side resources
[/oracle/oradata/SID2]) from snapshots
[[smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0]]
[ INFO] SD-10030: Waiting for SnapDrive job (running)
[ INFO] SD-10030: Waiting for SnapDrive job (completed)
[ INFO] SD-00055: Finished volume restore of
[vcustomer01:/vol/customer01_oradata_zone_test2_SID2] (with host-side resources
[/oracle/oradata/SID2]) from snapshots
[[smo_sid2_sid2_f_h_1_8ac09c4425bd0e8f0125bd0ea1900001_0]]
[ INFO] ORACLE-30008: Beginning recovery process for database SID2.
[ INFO] ORACLE-30010: Database recovery point objective: latest committed transaction
for the incarnation
[ INFO] ORACLE-30017: Recovery point objective of latest committed transaction for the
incarnation reached after applying archived/online redo logs
[/oracle/oralogtemp/SID2/redo01.log].
[ INFO] ORACLE-20000: Changing state for database instance SID2 from MOUNTED to OPEN.
[ INFO] ORACLE-20032: Opening database SID2 with NORESETLOGS option.
[ INFO] SMO-07131: Unlocked database for SnapManager operations - removed lock file
"/oracle/app/oracle/product/10.2.0/db_1/dbs/.sm_lock_SID2" on host zone-test2.
[ INFO] SMO-07433: Returning the database to its initial state: SID2(OPEN).
[ INFO] SMO-13048: Backup Restore Operation Status: SUCCESS
[ INFO] SMO-13049: Elapsed Time: 0:02:35.773

```

Restore Plan:

Review:

```

The following files have been restored completely via: fast restore
/oracle/oradata/SID2/sysaux01.dbf
/oracle/oradata/SID2/system01.dbf
/oracle/oradata/SID2/undotbs01.dbf
/oracle/oradata/SID2/users01.dbf

```

Analysis:

```

All files have been restored via optimal restore mechanisms.

```

Operation Id [8ac09c4425bd43e20125bd43f05d0001] succeeded.

3.6 MOVING THE VFILER UNIT BETWEEN CONTROLLERS

MultiStore® offers the ability to move a whole vFiler unit with all its resources online between NetApp controllers. This feature is excellent for load balancing and performing planned maintenance on one controller or a controller pair. A vFiler unit can be moved to any controller as long it has enough disk space to copy the data and network connectivity to the source controller. This can be likened to a type of VMotion for storage resources.

1. Before moving the vFiler unit, create the FlexVol volumes allocated to the vFiler unit on the destination. This is applicable for the FlexClone volumes as well.
2. Issue the `vfiler migrate start` command from the destination storage to initiate the transfer.

In this example, the destination storage is named `remotefile`.

```
remotefiler> vfiler migrate start vcustomer01@filer
filer's Administrative login: root
filer's Administrative password:
Configuring SnapMirror to transfer vfiler vcustomer01's data from remote filer
filer.
Starting snapmirror initialize commands. It
could take a very long time when the source or
destination filers are involved in many
simultaneous transfers. The console will not be
available until all initialize commands are
started successfully. Please use the
"snapmirror status" command on the source
filer to monitor the progress.

Volume 'customer01_zone_os' is now restricted.
Volume 'customer01_oradata_zone_test1_SID1' is now restricted.
Volume 'customer01_oradata_zone_test2_SID2' is now restricted.
Volume 'customer01_oradata_zone_test2_SID3' is now restricted.
Volume 'customer01_oradata_zone_test2_SID4' is now restricted.
Volume 'customer01_oralogtemp' is now restricted.
Volume 'customer01_orarch' is now restricted.
Volume 'customer01_oractrl' is now restricted.
After migrate, configure vfiler IP address 10.64.31.160? [y]: y
Interface to assign this address to {e0a, e0b, e0c, e0d}: e0a
Netmask to use: [255.255.255.0]:
Data transfer for vfiler storage units initiated.
remotefiler*>
```

3. Check the status of the SnapMirror® copy:

```
remotefile*> vfiler migrate status vcustomer01@filer
Storage Path          Status
customer01_zone_os    snapmirrored
customer01_oradata_zone_test1_SID1  snapmirrored
customer01_oradata_zone_test2_SID2  snapmirrored
customer01_oradata_zone_test2_SID3  snapmirrored
customer01_oradata_zone_test2_SID4  snapmirrored
customer01_oralogtemp  snapmirrored
customer01_orarch     snapmirrored
customer01_oractrl    snapmirrored
```

4. Complete the migration by moving the vFiler unit to the destination controller:

```
remotefiler*> vfiler migrate complete vcustomer01@filer
Stopping remote vfiler...
Updating vfiler storage units...
Starting snapmirror update commands. It
```

```
could take a very long time when the source or
destination filers are involved in many
simultaneous transfers. The console will not be
available until all update commands are
started successfully. Please use the
"snapmirror status" command on the source
filer to monitor the progress.
```

```
Waiting for "customer01_oractrl" to become stable.
Vfiler vcustomer01 migrated.
```

By issuing a `df` command on the host, you will see a few NFS messages indicating that the command is completed:

```
oracle@zone-test1:~$ df
/                (/                ):47741628 blocks  3475537 files
/dev             (/dev            ):47741628 blocks  3475537 files
/proc           (proc           ):      0 blocks   16188 files
/system/contract (ctfs           ):      0 blocks  2147483554 files
/etc/mnttab     (mnttab        ):      0 blocks      0 files
/system/object  (objfs         ):      0 blocks  2147483447 files
/etc/svc/volatile (swap         ): 8933848 blocks  552508 files
/lib/libc.so.1  (/usr/lib/libc/libc_hwcap1.so.1):47741628 blocks  3475537 files
/dev/fd         (fd            ):      0 blocks      0 files
/tmp            (swap         ): 8933848 blocks  552508 files
/var/run        (swap         ): 8933848 blocks  552508 files
NFS server vcustomer01 not responding still trying
NFS server vcustomer01 ok
/oracle/oralogtemp (vcustomer01:/vol/customer01_oralogtemp/zone_test1):15539536
blocks  311148 files
/oracle/orarch     (vcustomer01:/vol/customer01_orarch/zone_test1):49808560 blocks
933751 files
/oracle/oractrl    (vcustomer01:/vol/customer01_oractrl/zone_test1): 1634000
blocks  31004 files
/oracle/oradata/SID1(vcustomer01:/vol/customer01_oradata_zone_test1_SID1):15213528
blocks  311175 files
```

3.7 MOVING A ZONE BETWEEN SERVERS

In addition to having the MultiStore vFiler unit migration capabilities, Solaris Zones can be easily moved between Solaris servers. This is a relatively simple operation because it only requires an update to the zone boot LUN mapping and shutdown/start the zone.

In this example, `zone_test2` from `c01srv01-global-zone` is moved to `c01srv02-global-zone` by performing the following steps:

1. Stop the zone as follows:

```
root@c01srv01-global-zone:~# zoneadm -z zone_test2 halt
root@c01srv01-global-zone:~# zoneadm -z zone_test2 detach
root@c01srv01-global-zone:~# umount /zones/zone_test2
```

2. Create an igroup for the new Solaris server as follows:

```
vcustomer01@filer*> igroup create -i -t solaris c01srv02-global-zone iqn.1986-
03.com.sun:01:fac868f0ffff.2fe7def1
```

3. Check the zoning as follows:

```
filer*> vfiler run vcustomer01 lun show -m
===== vcustomer01
LUN path                               Mapped to          LUN ID  Protocol
-----
/vol/customer01_zone_os/zone_template01/zone_template01.lun c01srv01-global-zone
0          iSCSI
```



```
/vol/customer01_zone_os/zone_test1/boot.lun c01srv01-global-zone 1
iSCSI
/vol/customer01_zone_os/zone_test2/boot.lun c01srv01-global-zone 2
iSCSI
```

4. Unmap the LUN from c01srv01-global-zone as follows:

```
filer*> vfiler run vcustomer01 lun unmap
/vol/customer01_zone_os/zone_test2/boot.lun c01srv01-global-zone
===== vcustomer01
sugar*> Wed Dec 23 20:41:28 GMT [vcustomer01@sugar: lun.map.unmap:info]: LUN
/vol/customer01_zone_os/zone_test2/boot.lun unmapped from initiator group
c01srv01-global-zone
```

5. Map the LUN to the c01srv02-global-zone server as follows:

```
sugar*> vfiler run vcustomer01 lun map /vol/customer01_zone_os/zone_test2/boot.lun
c01srv02-global-zone

===== vcustomer01
sugar*> Wed Dec 23 20:41:38 GMT [vcustomer01@sugar: lun.map:info]: LUN
/vol/customer01_zone_os/zone_test2/boot.lun was mapped to initiator group
c01srv02-global-zone=0
```

6. Log in to the host and scan the LUNS again:

```
root@c01srv02-global-zone:/# devfsadm -i iscsi
```

7. Mount the LUN:

```
root@c01srv02-global-zone:/# /opt/NTAP/SANToolkit/bin/sanlun lun show
controller:          lun-pathname          device
filename            adapter  protocol          lun size          lun
state
vcustomer01:  /vol/customer01_zone_os/zone_test2/boot.lun
/dev/rdisk/c2t60A98000433542746734547046745575d0s2  scsi_vhci0  iSCSI          30g
(32212254720)      GOOD

root@c01srv02-global-zone:/# mount
/dev/dsk/c2t60A98000433542746734547046745575d0s2 /zones/zone_test2
```

8. Copy the zone xml configuration file from the c01srv01-global-zone:

```
root@c01srv02-global-zone:/etc/zones# scp c01srv01-global-
zone:/etc/zones/zone_test2.xml /etc/zones/zone_test2.xml
zone_test2.xml          100% |*****| 358
00:00
```

9. Start the zone:

```
root@c01srv02-global-zone:/etc/zones# zoneadm -z zone_test2 attach
root@c01srv02-global-zone:/etc/zones# -z zone_test2 boot
```

8 CONCLUSION

This document shows that the combination of Solaris Zones, NFS, and NetApp MultiStore capabilities is a powerful framework to leverage Oracle Databases for shared multicustomer platforms. It describes how NetApp Snapshot can be leveraged to backup and restore clouds running Oracle Databases on several Solaris containers.

This report is not intended to be a definitive implementation or solutions guide. Many factors related to specific customer environments are not addressed in this document. Contact NetApp support to speak with one of our Oracle solutions experts for any deployment requirement. Please forward any errors, omissions, differences, new discoveries, or comments about this paper to jorge.costa@netapp.com.

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

