DeepSpace Storage

Quick Start Guide

Catalog Data Services and System Managed Storage for Ceph Environments

DeepSpace Storage
Nov 1, 2021

Table of Contents

Introduction

The purpose of this document is to provide a road map for the installation of the DeepSpace CDS-SMS ecosystem using a simplified configuration that will provide the user with an opportunity to test and evaluate a basic but fully functional system configuration.

The intended audience will have already read the CDS and SMS FAQ documents and developed an understanding of both the basic architecture and the features it will deliver to the end user. This document will attempt a simple explanation of all of the steps to go beyond a basic understanding of the DeepSpace ecosystem, and get the operator to the point of having a basic configuration operational in approximately one hour.

Before You Start

We'll assume you have the following prepared:

1. Access to the internet.
2. A server loaded with any version of Centos7 using the "Server with GUI" configuration.
3. The user will need root access or sudo privileges.
4. The server should have 32-64G of memory as it will co-host the RDBMS server, master server, and a client.
5. The server should have 2 empty file systems running native XFS as provided in the Centos distribution, and those file systems should be mounted directly under the system root, I.e., /test1 and /test2.
6. The system must have been previously configured for Ceph client storage using the librados API, this can be confirmed by running the following command:

```
# ceph -s
  cluster:
    id:     7288bfb6-f36c-42a1-a5c2-e5f63f5c5bf8
    health: HEALTH_OK
```

Lastly – You may find it helpful to print this guide and check off each step as completed. Missing any of these steps will result in a system that will require troubleshooting, which is not within the scope of this document.

System Preparation

After installing the "Server with GUI" configuration, you should perform a system update with yum:

```
# yum -y update
```

```
# vi /etc/selinux/config
```

Change enforcing to disabled:

```
# systemctl disable firewalld
```

Finally, make sure the master server is identified as “salt” in you name service, or with an entry in the /etc/hosts file on the same line as the master server’s regular hostname.

```
# grep salt /etc/hosts
192.168.0.5  alfalfa salt
```

Reboot the system for changes to take effect.

```
# reboot
```

Install and configure salt.

Start with the salt-master:

```
# curl -L https://bootstrap.saltstack.com -o install_salt.sh
```

```
# sh install_salt.sh -P -M
…
…
Complete!
 *  INFO: Running install_centos_stable_post()
 *  INFO: Running install_centos_check_services()
 *  INFO: Running install_centos_restart_daemons()
 *  INFO: Running daemons_running()
 *  INFO: Salt installed!
```

Next edit the /etc/salt/master, look for the line “interface:”, un-comment the line and edit the address to match your server ip.

```
[root@alfalfa scranage]# grep interface /etc/salt/master
# The address of the interface to bind to:
interface: 192.168.0.5
```

Test and make certain you can resolve the salt host name to the local server:

```
# ping salt
PING alfalfa (192.168.0.5) 56(84) bytes of data.
64 bytes from alfalfa (192.168.0.5): icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from alfalfa (192.168.0.5): icmp_seq=2 ttl=64 time=0.043 ms
```

Start and enable the salt-master service:

```
# systemctl start salt-master.service
```

```
# systemctl enable salt-master.service
```

Next install the salt-minion.

```
# sh install_salt.sh -P
…
…
*  INFO: Running install_centos_stable_post()
*  INFO: Running install_centos_check_services()
*  INFO: Running install_centos_restart_daemons()
*  INFO: Running daemons_running()
*  INFO: Salt installed!
```

```
# systemctl start salt-minion.service
```

```
# systemctl enable salt-minion.service
```

Finally, view and accept the key.

```
# salt-key -L
Accepted Keys:
Denied Keys:
Unaccepted Keys:
alfalfa
Rejected Keys:
```

```
# salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
alfalfa
Proceed? [n/Y] y
Key for minion alfalfa accepted.
```

```
# salt "*" test.ping
alfalfa:
  True
```

Installing the DeepSpace Catalog Data Service

The Catalog Data Service provides the catalog of user files (with version control), as well as the I/O subsystem for archive storage including Ceph, and Tape targets. Once the RDBMS is operational, it is time to install and test the CDS. During the testing process, you will gain an overview of how to store and retrieve files directly using the CDS command line interface.

Make sure /etc/hosts on all clients includes all client host names as well as the master.

Create deepspace.repo in /etc/yum.repos.d/deepspace.repo

Create the repo file using your preferred editor using the root account or sudo. It should contain the following:

```
[opencm]
name=Open Catalog Manager
baseurl=http://repo.deepspacestorage.com/repos/OpenCM/1/0
```

```
enabled=1
gpgcheck=0

[openui]
name=Open Storage Management User Interface
baseurl=http://repo.deepspacestorage.com/repos/OpenUI/1/0
enabled=1
gpgcheck=0

[opensmf]
name=Open System Monitoring Facility
baseurl=http://repo.deepspacestorage.com/repos/OpenSMF/1/0
enabled=1
gpgcheck=0

[opensms]
name=Open System Managed Storage
baseurl=http://repo.deepspacestorage.com/repos/OpenSMS/1/0
enabled=1
gpgcheck=0
```

1/2

Install the CDS using yum

```
# yum -y install DSCM DSCM-tools
```

1/3

NOTE: This installation will also install mt, mtx, and mariadb packages if not already installed.

```
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.den01.meanservers.net
 * centos-sclo-rh: mirrors.xmission.com
 * centos-sclo-sclo: mirror.cs.uwp.edu
 * epel: mirrors.syringanetworks.net
 * extras: mirror.compevo.com
 * rpmfusion-free-updates: mirror.math.princeton.edu
 * updates: mirror.cs.uwp.edu
Resolving Dependencies
--> Running transaction check
---> Package DSCM.x86_64 0:1.0.3-47_rb.el7 will be installed
---> Package DSCM-tools.x86_64 0:1.0.3-47_rb.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package          Arch       Version             Repository               Size
================================================================================

Installing:
```

2/3

```
 DSCM            x86_64     1.0.3-47_rb.el7  opencm          6.1 M                3/3
 DSCM-tools      x86_64     1.0.3-47_rb.el7  opencm          65 k

Transaction Summary

================================================================================

Install  2 Packages
Total download size: 6.2 M
Installed size: 36 M
Is this ok [y/d/N]: y

Downloading packages:

(1/2): DSCM-tools-1.0.3-47_rb.el7.x86_64.rpm                 |  65 kB  00:00:00
(2/2): DSCM-1.0.3-47_rb.el7.x86_64.rpm                       | 6.1 MB  00:00:01
--------------------------------------------------------------------------------

Total                                              3.1 MB/s | 6.2 MB  00:00:01

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction

Installing : DSCM-tools-1.0.3-47_rb.el7.x86_64                1/2

Installing : DSCM-1.0.3-47_rb.el7.x86_64  I                   2/2
Verifying  : DSCM-1.0.3-47_rb.el7.x86_64                      1/2
Verifying  : DSCM-tools-1.0.3-47_rb.el7.x86_64                2/2


Installed:

DSCM.x86_64 0:1.0.3-47_rb.el7 DSCM-tools.x86_64 0:1.0.3-47_rb.el7

Complete!
```

Install and configure the RDBMS

We will use the MariaDB 5.5 server that ships with Centos 7.x. If it is not already installed, do:

Install mariadb version 5.5 (NOTE: Skip if already installed)

```
# yum install -y mariadb mariadb-server

$ rpm -qa |grep maria
mariadb-libs-5.5.44-2.el7.x86_64
mariadb-5.5.44-2.el7.x86_64
mariadb-server-5.5.44-2.el7.x86_64
```

Either su to the root user, or run each of the following with ‘sudo’.

```
# systemctl start mariadb

# systemctl enable mariadb

# mysql_secure_installation
```

Set your root database user and accept all defaults.

Create the my.cnf file in /etc/my.cnf.d/my.cnf on the database server with your preferred editor.

```
# cat /etc/my.cnf.d/my.cnf
```

```
[mysqld]
open_files_limit=0
max_connections=1000
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
max_allowed_packet=128M
wait_timeout=31536000
query_cache_type=1
query_cache_size=64M
ignore_builtin_innodb
plugin_load=ha_innodb.so
innodb_buffer_pool_size=37G
innodb_log_buffer_size=64M
thread_cache_size=100
tmp_table_size=64M
max_heap_table_size=64M
innodb_file_per_table=1
datadir=/var/lib/mysql
init_file=/var/lib/mysql/init.sql
#slow_query_log=1
#long_query_time = 1
event_scheduler = 1
innodb_adaptive_hash_index=0
```

Note: Adjust this number based on system memory, it should be 50-70% of installed memory.

Copy or include the contents of init.sql and ds.cnf into @@DATADIR of MariaDB. If you are not sure what your mysql data directory is, run the following command using the mysql root password when prompted

```
# mysql -s -u root -p -e "SELECT @@datadir;"
Enter password:
@@datadir
/var/lib/mysql/
```

```
# cp /usr/local/ds/Librarian/sqls/init.sql /var/lib/mysql/
```

```
# systemctl restart mariadb
```

ds.cnf must now be customized for the DB location and copy to /etc/my.cnf.d directory.

With your preferred editor, open /etc/my.cnf.d/ds.cnf as root.

```
[deepspace]
database=deepspace
host=localhost
password=password1234
user=dscm

[dssmf]
database=smfdb
host=localhost
password=password1234
user=smf

[dsfs]
database=smsdb
host=localhost
password=password1234
user=sms
```

Create the DeepSpace user accounts in MariaDB

Create all of the database users identified in /etc/my.cnf.d/ds.cnf using the passwords provided in that file.

```
# dbuser_conf.pl

DB root password: **********

Connected to MySQL database at localhost successfully.
CREATE USER dscm@localhost
CREATE USER smf@localhost
CREATE USER sms@localhost
```

Create all DSCM Tables in the Database

```
# cd /usr/local/ds/Librarian/sqls
```

As root or sudo:

```
# ./create_DStables.sh
DB Root Password:
Host containing DB :dbserver3
create_DStables.sh: verify root access to DB server successful
create_DStables.sh: get access info
create_DStables.sh: verify dscm access to DB server at dbserver2 successful
create_DStables.sh: CREATE deepspace DATABASE
create_DStables.sh: Drop deepspace successful
create_DStables.sh: Create deepspace successful
create_DStables.sh: BUILD TABLES
create_DStables.sh: POOL successfully created
create_DStables.sh: ROTSPEC successfully created
create_DStables.sh: VREC/dVREC successfully created
create_DStables.sh: FREC/dFREC successfully created
create_DStables.sh: statistics successfully created
create_DStables.sh: event successfully created
create_DStables.sh: DEVREC successfully created
create_DStables.sh: VMS_ALLOC successfully created
create_DStables.sh: VMS_AVAIL successfully created
```

Now create all SMF tables in the database:

```
# cd /usr/local/ds/Librarian/sqls
```

```
# ./create_SMFtables.sh
DB Root Password:
Host containing DB: dbserver3
create_SMFtables.sh: verify root access to DB server successful
create_SMFtables.sh: get access info
create_SMFtables.sh: verify smf access to DB server at dbserver2 successful
create_SMFtables.sh: CREATE smfdb DATABASE
create_SMFtables.sh: Drop smfdb successful
create_SMFtables.sh: Create smfdb successful
create_SMFtables.sh: BUILD TABLES
create_SMFtables.sh: SMF successfully created
```

You are now done installing and configuring you RDBMS server. Please follow the next section to validate that you user accounts and tables are correctly configured.

Testing The RDBMS Configuration

Check Status and Boot Configuration

Before starting, make sure the database server is running, and configured to restart on reboot.

```
# systemctl status mariadb
mariadb.service - MariaDB database server
Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
Active: active (running) since Sun 2021-06-20 09:40:47 MDT; 3 weeks 6 days ago
Process: 2978 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited,
status=0/SUCCESS)
Process: 2796 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited,
status=0/SUCCESS)
Main PID: 2977 (mysqld_safe)
Tasks: 25
CGroup: /system.slice/mariadb.service
        ├─2977 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
        └─3686 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql –plugin-
       dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/m...


Jun 20 09:40:44 alfalfa systemd[1]: Starting MariaDB database server...
Jun 20 09:40:44 alfalfa mariadb-prepare-db-dir[2796]: Database MariaDB is probably
initialized in /var/lib/mysql already, nothing is done.
Jun 20 09:40:44 alfalfa mariadb-prepare-db-dir[2796]: If this is not the case, make sure the
/var/lib/mysql is empty before running mariadb...db-dir.
Jun 20 09:40:44 alfalfa mysqld_safe[2977]: 210620 09:40:44 mysqld_safe Logging to
'/var/log/mariadb/mariadb.log'.
Jun 20 09:40:44 alfalfa mysqld_safe[2977]: 210620 09:40:44 mysqld_safe Starting mysqld
daemon with databases from /var/lib/mysql
Jun 20 09:40:47 alfalfa systemd[1]: Started MariaDB database server.


Hint: Some lines were ellipsized, use -l to show in full.
```

Note that the “Loaded:” line at the top indicates enabled, and the “Active” line indicates active (running), otherwise run “systemctl start” and “systemctl enable” as needed either as root, or with ‘sudo’.

Check DS User Accounts and Tables

Note: The following steps must be tested on the DS Master Server to make sure the correct machine identity is validated. In a multi-host environment, you would repeat on each client.

From the DeepSpace server, log into the dscm account, when prompted enter the password from the **dsfs.cnf** you created in the /etc/my.cnf.d directory under the [deepspace] heading. Then make sure the needed tables can be viewed.

```
$ mysql -u dscm -p deepspace
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

    Welcome to the MariaDB monitor.  Commands end with ; or \g.
    Your MariaDB connection id is 103169
    Server version: 5.5.68-MariaDB MariaDB Server

    Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

    Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Perform show tables for deepspace:

```
MariaDB [deepspace]> show tables;                                    1/2
    +------------------------+
    | Tables_in_deepspace |
    +------------------------+
    | DEV                    |
    | DEVREC                 |
    | FREC                   |
    | POOL                   |
    | POOL_GL                |
    | POOL_TBL               |
    | POOL_UL                |
    | ROTSPEC                |
    | ROTSPEC_LS             |
    | VMS_ALLOC              |
    | VMS_AVAIL              |                                         2/2
    | VREC                   |
    | dFREC                  |
    | dVREC                  |
    | statistics             |
    | vFREC                  |
    | vVREC                  |
    +------------------------+
    17 rows in set (0.00 sec)
```

Exit MariaDB deepspace table interface:

```
MariaDB [deepspace]> quit

    Bye
```

From the DeepSpace server, log into the smf account, when prompted enter the password from the **dsfs.cnf** you created in the /etc/my.cnf.d directory under the [dssmf] heading. Then make sure the needed tables can be viewed.

```
$ mysql -u smf -p smfdb
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

    Welcome to the MariaDB monitor.  Commands end with ; or \g.
    Your MariaDB connection id is 103193
    Server version: 5.5.68-MariaDB MariaDB Server

    Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

    Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Perform show tables for smfdb:

```
MariaDB [smfdb]> show tables;
    +--------------------+
    | Tables_in_smfdb |
    +--------------------+
    | SMF                |
    +--------------------+
    1 row in set (0.00 sec)
```

Exit MariaDB smfdb table interface:

```
MariaDB [smfdb]> quit

    Bye
```

From the DeepSpace server, log into the sms account, when prompted enter the password from the **dsfs.cnf** you created in the /etc/my.cnf.d directory under the [dssmf] heading. Then make sure the needed tables can be viewed.

```
# mysql -u sms -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2977
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Now create the smsdb database:

```
MariaDB [(none)]> create database smsdb;

    Query OK, 1 row affected (0.00 sec)
```

Exit MariaDB smsdb table interface:

```
MariaDB [(none)]> quit;
```

You are now ready to proceed with installing the DeepSpace Catalog Data Service.

Install DSCM

```
# cd /usr/local/ds
```

```
# ./ds install                                                    1/3
        DSCM environment file '/etc/dsenv'
The Environment file '/etc/dsenv' does not exist.

Do you want to create it?[y/n]y

Library Directory[ /usr/local/lib ]:
Binary Directory[ /usr/local/bin ]:
Server Machine[ groot ]:
Program Number[ 667777770 ]:
Client Name[ HOST ]:
ISODATES[ on ]:
Library Directory: /usr/local/lib
Binary Directory: /usr/local/bin
Server Machine: groot
Program Number: 667777770
Client Name: HOST
ISODATES: on

Values OK?[y/n]y

Master install

Installing from '/usr/local/ds'

Installing: DSlink
Installing: RCOM
Installing: ds_env
Installing: dsreport
Installing: lock_drive
Installing: adnadd
Installing: adnlist
Installing: adnmod
Installing: adndel
Installing: ismaster
Installing: mntlist
Installing: dsreinit
Installing: isgrent
Installing: dslog
Installing: dslog_change
Installing: dslog_test
Installing: dslog_exit
Installing: dsopmon
Installing: DSV
```

```
Installing: ds_menus
Installing: NUTIL
Installing: DSnet
Installing: ds_menus
Installing: stkmount
Installing: stkunmount
Installing: posn_stack
Installing: stc_prolog
Installing: add_vols.pl
Installing: get_media.pl
Installing: robo_mount.pl
Installing: robo_unmount.pl
Installing: test_library.pl
Installing: scan_changer
Installing: mtx
Installing: /usr/local/lib/DSCM
Installing: Librarian
Installing: Librarian/Adn
Installing: Librarian/Alog
Installing: Librarian/App
Installing: Librarian/const
Installing: Librarian/dev.default
Installing: Librarian/fmts
Installing: Librarian/jdir
Installing: Librarian/loc
Installing: Librarian/ops
Installing: Librarian/site_exits
Installing: Librarian/tapecap
Installing: Librarian/tname
Installing: Librarian/trusted_hosts
Installing: Librarian/rlvms_config
Installing: Librarian/Tdb
The new tapecap file contains tapecap entries for the currently supported list of devices.

SDISK.1600, adding to tapecap...
SDISK.6250, adding to tapecap...
SDISK.CART, adding to tapecap...
CART, adding to tapecap...
9X40, adding to tapecap...
SDLT, adding to tapecap...
DLT, adding to tapecap...
LTO, adding to tapecap…
RADOS, adding to tapecap...
DISK, adding to tapecap…

DSCM ok

Do you want to install the programmer's library?[y/n]y
```

```
Programmer library directory [ /usr/lib ]:                                    3/3
Installing: librltape.a
Programmer include directory [ /usr/include/ds ]:

Directory /usr/include/ds does not exist, do you wish to create it? [y/n]y
Installing: reel.h
Installing: reel_defs.h
Installing: tstamp.h
Installing: spec.h
Installing: frec.h
Installing: vrec.h
Installing: vcur.h
Installing: pool.h
Installing: tcap.h
Installing: adn.h
Installing: rotspec.h
Installing: reel_struct.h
Installing: df.h
Installing: reel_err.h
Programmer example directory [ /usr/local/lib/DSCM_examples ]:
Installing: ansi_read.c
Installing: ansi_write.c
Installing: Makefile
*****************************************************************
Depending on your system you may have to run ranlib on libdstape.a
*****************************************************************

Installation complete
```

It is vital the RPC service be enabled and started!

```
# systemctl enable rpcbind
```

```
# systemctl start rpcbind
```

Finish by starting ds, and checking it:

```
# systemctl start dsd.service
```

```
# systemctl enable dsd.service
```

Add /usr/local/bin to your path. If you are a bash shell user for example, edit the .bashrc file in your home directory and add:

```
export PATH=$PATH:/usr/local/bin
```

Then:

```
. .bashrc
```

```
# ds

Checking status of DSCM servers:
Checking dslog Network Log Daemon server status:
        dslog server is up.
Checking DS Mount Request Daemon server status:
        DS server is up.
Checking DSnet Network Daemon server status:
        DSnet server is up.
```

Adding Ceph Devices

Ceph devices are created as virtual drives, and the objects are written to the cluster as logical volumes that share the virtual drive mount points. Logical volumes have a maximum size of 125MB, this is a maximum imposed by LibRADOS usage rules. You need not worry about volume sizes as we span volumes within the CDS implementation.

So, we'll start by creating the drives in the CDS:

```
# build_drives.sh -r 20 -v
cleanup Adn files
cleanup RADOS Dev
cleanup RADOS site_exits
setup RADOS Adns
setup RADOS Devs
setup RADOS site_exits
```

```
# systemctl restart dsd.service
```

You now have 20 drives. The next step is to create a ceph pool, then we can make ceph scratch volumes. The pool name can be anything, but a pool should never be shared between different catalog servers, consider using the master host name as part of the pool name for clarity, and give it root ownership – this will take place in the CDS setup, but accommodate it now in the cluster pool name.

```
# rados mkpool root/alfalfap

successfully created pool root/alfalfap
```

This step isn't absolutely required, but without it you see warnings back from ceph status.

```
# ceph osd pool application enable root/alfalfap custom-application

enabled application 'custom-application' on pool 'root/alfalfap'
```

And now create a corresponding pool in the CDS

```
# dspcreate uacc=ANY gacc=ANY root/alfalfap
```

Success is indicated with a silent return.

Now create a number of scratch volumes:

```
# scrpsubmit --vid=000000000 --pool=root/alfalfap --nvol=0100000
mvdb_inserted 100000 rows
Total time – 4.69669
```

The volume list can now be displayed with the dsreport command:

```
# dsreport --vlist
…..
…..
000099989   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099990   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099991   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099992   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099993   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099994   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099995   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099996   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099997   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099998   scr  RADOS   1048    NO  onsite      >root/alfalfap
000099999   scr  RADOS   1048    NO  onsite      >root/alfalfap

Command: dsreport --vlist --user=root --full=yes
```

Testing The CDS

First, create a volumeset to write to.

```
# dsvcreate ftrack=yes initialize=yes rformat=vs:262144:262144 loc=onsite cap=1048 1/2
type=RADOS vmode=777 format=DSCM group=root vexpire=L pool=root/alfalfap root/testvs
Allocated Volume: 000000003 2/2
```

Then access it in write mode.

```
# dsvaccess write=yes testvs

Device Reserved
```

And then write a file.

```
# dsvwrite if=/etc/dsenv fid=dsenv

Requesting volume 000000003...Awaiting Mount...complete
```

Observe that the file is now in the catalog.

```
# dsreport --volset=testvs:G0000:V00 --vsflist

Volumeset File List: 20210720 15:07:42

Fid                 SC   Blocks   Fexpire   Expires        Fcomment
---                 --   ------   -------   -------        --------
dsenv:G0:V0         1    1        S         EXPIRED        (D)
End of Volumeset


Command: dsreport --vsflist --volset=testvs:G0000:V00 –full=yes
```

Next read it back.

```
# dsvread fid=dsenv of=foobar
```

```
# diff ./foobar /etc/dsenv
```

The files should show no differences. Lastly, free the volumeset so it is available to other users.

```
# dsfree
```

Then you can scratch it if you like.

```
# dsvscratch force=y testvs:G0000:V00

Scratch Complete
```

You are now ready to install SMS and the dashboard (GUI).

Installing SMS

Systems Managed Storage as currently released only supports XFS file systems. Install the XFS fanotify version.

```
# yum clean expire-cache                                                    1/2
# yum install -y DSFS-xfs_fa
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.den01.meanservers.net
 * centos-sclo-rh: mirror.grid.uchicago.edu
 * centos-sclo-sclo: mirrors.radwebhosting.com
 * epel: mirrors.xmission.com
 * extras: mirrors.radwebhosting.com
 * rpmfusion-free-updates: mirror.math.princeton.edu
 * updates: mirrors.radwebhosting.com
Resolving Dependencies
--> Running transaction check
---> Package DSFS-xfs_fa.x86_64 0:1.0.3-2k_dbg_3_47.el7 will be installed
--> Finished Dependency Resolution
```

```
2/2
Dependencies Resolved

================================================================================
Package       Arch    Version           Repository      Size
================================================================================
Installing:
 DSFS-xfs_fa   x86_64  1.0.3-2k_dbg_3_47.el7 opensms      2.0 M

Transaction Summary
================================================================================
Install 1 Package
Total download size: 2.0 M
Installed size: 9.7 M
Downloading packages:
DSFS-xfs_fa-1.0.3-2k_dbg_3_47.el7.x86_64.rpm
| 2.0 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : DSFS-xfs_fa-1.0.3-2k_dbg_3_47.el7.x86_64 1/1
Created symlink from /etc/systemd/system/multi-user.target.wants/sms-hsm.target to
        /usr/lib/systemd/system/sms-hsm.target.
Verifying  : DSFS-xfs_fa-1.0.3-2k_dbg_3_47.el7.x86_64 1/1

Installed:
  DSFS-xfs_fa.x86_64 0:1.0.3-2k_dbg_3_47.el7

Complete!
```

*** IMPORTANT ***

If this is a new installation, execute the following script

```
# /usr/local/opensms/init/sms_db_init.sh
sms_db_init.sh: Building base SMS tables if necessary
sms_db_init.sh: Base SMS tables successfully created
SMS base DB table setup complete
```

Install the GUI

Now install the Dashboard rpm as root or using sudo.

```
# yum install dashboard -y                                                    1/2
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.den01.meanservers.net
 * centos-sclo-rh: mirror.den01.meanservers.net
 * centos-sclo-sclo: mirrors.unifiedlayer.com
 * epel: mirrors.xmission.com
 * extras: mirror.sjc02.svwh.net
 * rpmfusion-free-updates: mirror.math.princeton.edu
 * updates: mirror.facebook.net
Resolving Dependencies
--> Running transaction check
---> Package dashboard.x86_64 0:0.1.0-0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package       Arch      Version    Repository            Size
================================================================================
Installing:
 dashboard     x86_64    0.1.0-0    openui                36 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 36 M
Installed size: 40 M
Downloading packages:
dashboard-0.1.0-0.x86_64.rpm                                | 36 MB  00:00:09
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : dashboard-0.1.0-0.x86_64          1/1
Created symlink from /etc/systemd/system/multi-user.target.wants/salt-master.service to
/usr/lib/systemd/system/salt-master.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/salt-minion.service to
/usr/lib/systemd/system/salt-minion.service.
--- Starting Dashboard ---
Created symlink from /etc/systemd/system/multi-user.target.wants/dashboard.service to
/usr/lib/systemd/system/dashboard.service.
Verifying  : dashboard-0.1.0-0.x86_64          1/1
```

```
Installed:                                                                    2/2
  dashboard.x86_64 0:0.1.0-0

Complete!
```

```
# systemctl status dashboard
dashboard.service - UI Dashboard and Admin Management for DeepSpace
Loaded: loaded (/usr/lib/systemd/system/dashboard.service; enabled; vendor preset: disabled)
Active: active (running) since Wed 2021-08-11 10:56:09 MDT; 11s ago
Main PID: 19440 (dashboard.jar)
   Tasks: 42
  CGroup: /system.slice/dashboard.service
          ├─19440 /bin/bash /usr/local/bin/dashboard.jar
          ├─19457 /usr/bin/java -Dsun.misc.URLClassPath.disableJarChecking=true -jar
          └─/usr/local/bin/dashboard.jar

Aug 11 10:56:09 alfalfa systemd[1]: Started UI Dashboard and Admin Management for
DeepSpace.
Aug 11 10:56:10 alfalfa dashboard.jar[19440]: .   ____          _            __ _ _
Aug 11 10:56:10 alfalfa dashboard.jar[19440]: /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
Aug 11 10:56:10 alfalfa dashboard.jar[19440]: ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
Aug 11 10:56:10 alfalfa dashboard.jar[19440]: \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
Aug 11 10:56:10 alfalfa dashboard.jar[19440]: '  |____| .__|_| |_|_| |_\__, | / / / /
Aug 11 10:56:10 alfalfa dashboard.jar[19440]: =========|_|==============|___/=/_/_/_/
Aug 11 10:56:10 alfalfa dashboard.jar[19440]: :: Spring Boot ::        (v2.0.0.RELEASE)

Hint: Some lines were ellipsized, use -l to show in full.
```

You are now ready to open the dashboard by using the https://localhost, you will get a certificate error, and need to accept the risk to continue. You may log in as root or enable operator privileges for other users by adding existing usernames to /usr/local/lib/DSCM/Librarian/ops. Editing the 'ops' file requires root or sudo privileges. Each username must be on its own line.