

In 60 Minutes: Build a Storage Cloud that is
Sustainable, Low Cost, and Secure

Contents

Demo and Virtual Machine Description.....	3
Virtual Machine Preparation.....	4
Install and Configure Virtual Machine HOL2925_MHVTL	4
Install and Configure Virtual Machine HOL2925_Solaris-HSM	6
Install and Configure Virtual Machine HOL2925_Solaris_Swift	7
Demo Steps	8
Rollback and Start The Three Virtual Machines (VMs).....	8
Configure HSM QFS Server (HOL2925_Solaris-HSM)	9
Configure Swift Storage Node(HOL2925_Solaris_Swift)	15
Verify Tape Archive Function on HSM QFS Server (HOL2925_Solaris-HSM)	23

Demo and Virtual Machine Description

There are three Virtual Machines powered off in Virtual Box on each laptop.

1. **HOL2925_MHVTL.** This VM is Linux and is running Virtual Tape Library software. It will be used by Oracle HSM to archive data to tape. In this case, virtual tape. Once this system is booted, there will be no additional activity or instructions from the server.
2. **HOL2925_Solaris-HSM.** This VM is running Solaris 11.2 and Oracle HSM 6.0. The StorageTek SAM-QFS product name was changed to Oracle HSM in the Spring of 2015. In this server you will run the Oracle HSM User Interface (GUI) from a Firefox browser, you will run cloud instructions from terminal windows and will create and monitor data in the Oracle HSM File System.
3. **HOL2925_Solaris_Swift.** This VM is running Solaris 11.2, Oracle HSM 6.0 client and OpenStack Swift Havana. In this server you will run Swift commands from the command line in terminal windows to activate a storage cloud environment for use by Oracle HSM as disk archive storage.

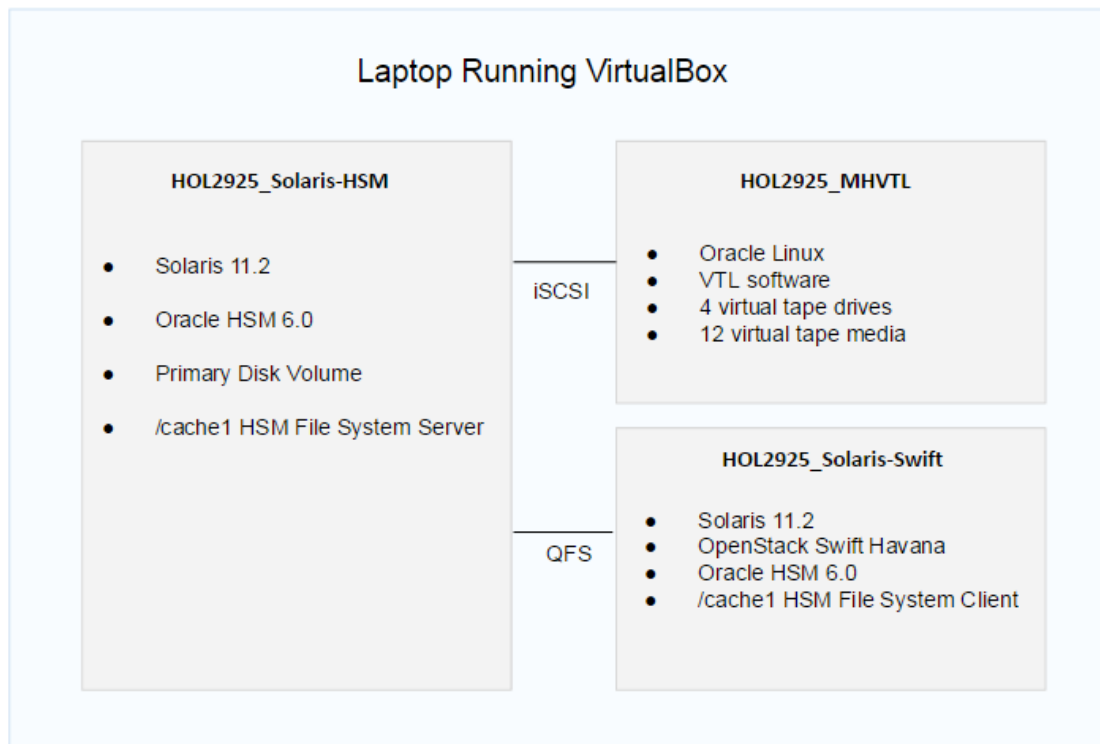


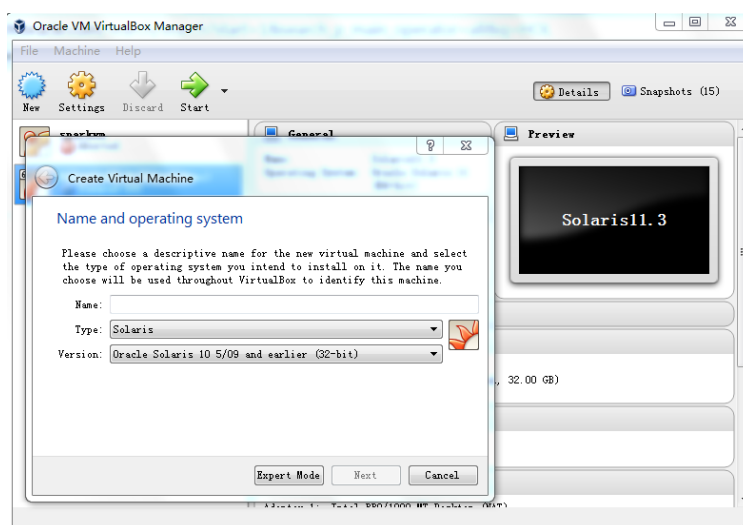
Figure 1: Virtual Box running three Virtual Machines and the virtual connectivity between them

Virtual Machine Preparation

Before we start preparing the virtual machines for this lab, we need to download and install Oracle VirtualBox.

- ✓ Oracle VirtualBox download page:
<https://www.virtualbox.org/wiki/Downloads>
- ✓ Oracle VirtualBox installation guide:
<https://www.virtualbox.org/manual/ch02.html>

With VirtualBox being successfully installed, you should be able to create the new virtual machines.



Install and Configure Virtual Machine HOL2925_MHVTL

For virtual machine HOL2925_MHVTL, we will use Oracle Linux 6 as the OS. Then install mhVTL in the virtual machine.

- ✓ Download Oracle VM Template for Oracle Linux, then import it to VirtualBox
We can use the Oracle VM Templates for Oracle Linux to quickly provision a new virtual machine running Oracle Linux 6. Here is the download link:
<http://www.oracle.com/technetwork/server-storage/vm/enterprise-linux-090498.html>
To import the VM template, please follow this document:
https://docs.oracle.com/cd/E26217_01/E26796/html/qs-import-vm.html
- ✓ Follow the following guide to install mhVTL in the virtual machine
<https://sites.google.com/site/linuxvtl2/>
- ✓ Configure a virtual tape library
 - ✓ Create the directory first

```
[root@vtl ~]# mkdir /etc/mhvtl/
```

- ✓ Then configure the main configuration file mhvtl.conf

```
[root@vtl mhvtl]# cat /etc/mhvtl/mhvtl.conf
# Home directory for config file(s)
MHVTL_CONFIG_PATH=/etc/mhvtl
# Default media capacity (500 M)
CAPACITY=500
# Set default verbosity [0|1|2|3]
VERBOSE=1
# Set kernel module debugging [0|1]
VTL_DEBUG=0
```

- ✓ The following device.conf file will configure one changer and four tape drives:

```
[root@vtl mhvtl]# cat /etc/mhvtl/device.conf
Library: 10 CHANNEL: 00 TARGET: 00 LUN: 00
Vendor identification: STK
Product identification: L700
Unit serial number: XYZZY_A
NAA: 10:22:33:44:ab:00:00:00
Home directory: /opt/mhvtl
Backoff: 400
# fifo: /var/tmp/mhvtl

Drive: 11 CHANNEL: 00 TARGET: 01 LUN: 00
Library ID: 10 Slot: 01
Vendor identification: IBM
Product identification: ULTRIUM-TD5
Unit serial number: XYZZY_A1
NAA: 10:22:33:44:ab:00:01:00
Compression: factor 1 enabled 1
Compression type: lzo
Backoff: 400
# fifo: /var/tmp/mhvtl

Drive: 12 CHANNEL: 00 TARGET: 02 LUN: 00
Library ID: 10 Slot: 02
Vendor identification: IBM
Product identification: ULTRIUM-TD5
Unit serial number: XYZZY_A2
NAA: 10:22:33:44:ab:00:02:00
Compression: factor 1 enabled 1
Compression type: lzo
Backoff: 400
# fifo: /var/tmp/mhvtl

Drive: 13 CHANNEL: 00 TARGET: 03 LUN: 00
Library ID: 10 Slot: 03
Vendor identification: IBM
Product identification: ULTRIUM-TD5
Unit serial number: XYZZY_A3
NAA: 10:22:33:44:ab:00:03:00
Compression: factor 1 enabled 1
Compression type: lzo
Backoff: 400
# fifo: /var/tmp/mhvtl

Drive: 14 CHANNEL: 00 TARGET: 04 LUN: 00
Library ID: 10 Slot: 04
Vendor identification: IBM
Product identification: ULTRIUM-TD5
Unit serial number: XYZZY_A4
NAA: 10:22:33:44:ab:00:04:00
Compression: factor 1 enabled 1
Compression type: lzo
Backoff: 400
# fifo: /var/tmp/mhvtl
```

- ✓ At least configure the content configuration file which holds all tapes:

```
[root@vtl mhvtl]# cat library_contents.10-ibm
Drive 1:
Drive 2:
Drive 3:
Drive 4:
Picker 1:
MAP 1:
MAP 2:
MAP 3:
MAP 4:
Slot 1: S00001L4
Slot 2: S00002L4
Slot 3: S00003L4
Slot 4: S00004L4
Slot 5: S00005L4
Slot 6: S00006L4
Slot 7: S00007L4
Slot 8: S00008L4
Slot 9: S00009L4
Slot 10: S00010L4
Slot 11: S00011L4
Slot 12: S00012L4
Slot 13: S00013L4
Slot 14: S00014L4
Slot 15: S00015L4
Slot 16: S00016L4
Slot 17: S00017L4
Slot 18: S00018L4
Slot 19: S00019L4
Slot 20: CLN020L4
```

- ✓ Verify the virtual tape library

```
[root@vtl mhvtl]# service mhvtl start
[root@vtl mhvtl]# lsscsi -g
[1:0:0:0] cd/dvd VBOX CD-ROM 1.0 /dev/sr0 /dev/sg0
[2:0:0:0] disk ATA VBOX HARDDISK 1.0 /dev/sda /dev/sg1
[32:1:0:0] mediumx STK SL150 0016 /dev/sch0 /dev/sg6
[32:1:0:1] tape HP Ultrium 4-SCSI A3K6 /dev/st0 /dev/sg2
[32:1:0:2] tape HP Ultrium 4-SCSI A3K6 /dev/st1 /dev/sg3
[32:1:0:3] tape HP Ultrium 4-SCSI A3K6 /dev/st2 /dev/sg4
[32:1:0:4] tape HP Ultrium 4-SCSI A3K6 /dev/st3 /dev/sg5
```

Install and Configure Virtual Machine HOL2925_Solaris-HSM

For virtual machine HOL2925_Solaris-HSM, we will use Oracle Solaris 11 as the OS. Then install Oracle HSM 6.0 in the virtual machine.

- ✓ Download Oracle VM Template for Oracle Linux, then import it to VirtualBox
We can use the Oracle Solaris 11 VM Templates to quickly provision a new virtual machine running Oracle Solaris 11. Here is the download link:
<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/vm-templates-2245495.html>
To import the VM template, please follow this document:
https://docs.oracle.com/cd/E26217_01/E26796/html/qs-import-vm.html
- ✓ Following the HSM installation document to install the HSM 6.0 on the Virtual Machine:
http://docs.oracle.com/cd/E60433_01/en/E42062/html/installpkgs.htm#CIHDDBBH
- ✓ Following the HSM document to configure a Oracle HSM Archiving File Systems:

http://docs.oracle.com/cd/E60433_01/en/E42062/html/configfsbasics.htm#BABEEACE

Note: this HSM archiving file system should use the virtual tape library we've just created in the HOL2925_MHVTL

Install and Configure Virtual Machine HOL2925_Solaris_Swift

For virtual machine HOL2925_Solaris-Swift, we will use Oracle Solaris 11 as the OS. Then install Openstack Swift in the virtual machine.

- ✓ Download Oracle VM Template for Oracle Linux, then import it to VirtualBox
We can use the Oracle Solaris 11 VM Templates to quickly provision a new virtual machine running Oracle Solaris 11. Here is the download link:

<http://www.oracle.com/technetwork/server-storage/solaris11/downloads/vm-templates-2245495.html>

To import the VM template, please follow this document:

https://docs.oracle.com/cd/E26217_01/E26796/html/qs-import-vm.html

- ✓ Use Solaris IPS to install Openstack Swift

```
root@solSwift:~# pkg install swift swiftclient
```

After all the three virtual machines are installed, please create a VirtualBox snapshot for each of the virtual machines, and name the snapshots as HOL_ReadyToGo. During the HOL, you can roll back to the original state at any time when you meet unfixable problem.

Demo Steps

Rollback and Start The Three Virtual Machines (VMs)

Rollback and boot the three VMs in the following order

- HOL2925_MHVTL
- HOL2925_Solaris-HSM
- HOL2925_Solaris_Swift

Run the following steps for each of the above VMs

1. For each VM execute the following steps to rollback and boot the VM to the snapshot that is labeled “HOL_ReadyToGo”
 - a. Rollback the image
 - i. Click the VM name in the left column
 - ii. Click on ‘Snapshots’ in the top right
 - iii. Click to Highlight HOL_ReadyToGo
 - iv. Click the icon that represents “Restore selected snapshot of the virtual machine”
 1. Unclick the ‘Create a snapshot of the current machine state’ box
 2. Click Restore
 - b. Boot the machine
 - i. Click the green Start arrow at the top right
 - ii. Click return to select all defaults through the boot process
 - iii. Wait for the machine to boot completely

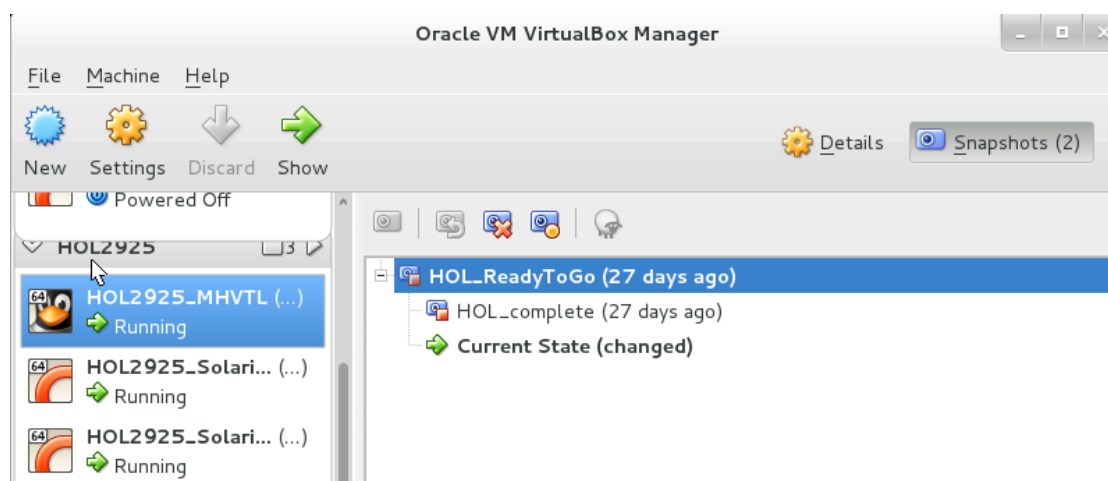


Figure 2: Virtual Box Manager started from the laptop

Configure HSM QFS Server (HOL2925_Solaris-HSM)

In the Solaris HSM system, a HSM file system (/cache1) will be created. Later on the OpenStack Swift storage will be linked to /cache1, so everything being written to the Swift storage will be automatically archived into the tapes.

COMMAND

- Switch to the HOL2925_Solaris-HSM Virtual Machine
 - Click on the Virtual Box icon at the bottom of your screen
 - All VM windows and the VM Manager will be displayed at the bottom.
 - Click on the HOL2925_Solaris-HSM window
- Log into the HOL2925_Solaris-HSM system with user 'root'; password 'l1admin'
(That is lower case l as in lucky, number 1,admin)

RESULT

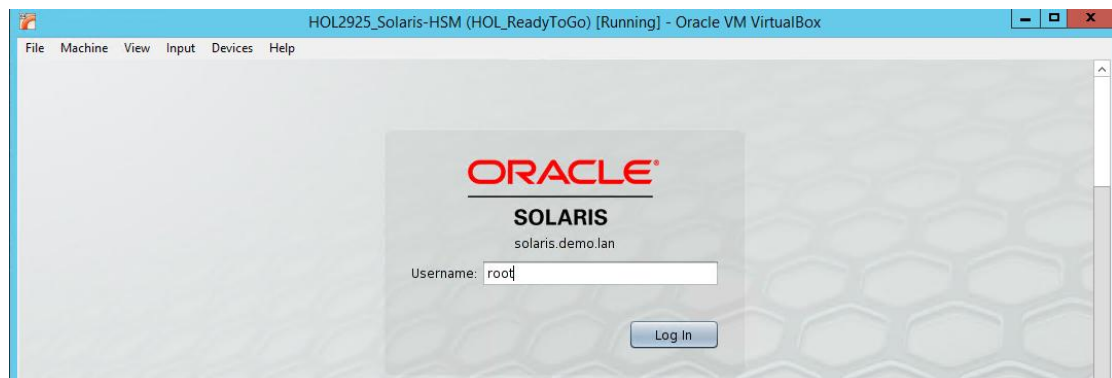


Figure 3: Oracle HSM Login Screen. Login using **root** and **l1admin**

COMMAND:

- Open a terminal in the HOL2925_Solaris-HSM

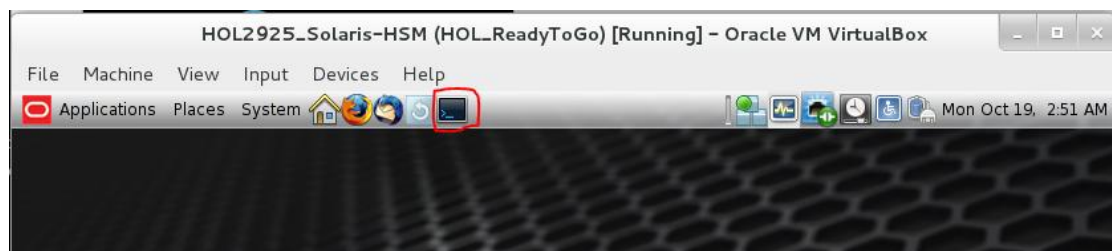


Figure 4: Open terminal

- Rebuild SAM-QFS file system named “cache1

```
root@solaris:~# umount -f /cache1
root@solaris:~# sammkfs -a 64 -S cache1
Warning: Creating a new file system prevents use with 4.6 or
earlier releases.
Use the -P option on sammkfs to create a 4.6 compatible file
system.
```

Building 'cache1' will destroy the contents of devices:

```
    /dev/dsk/c7t2d0s0
```

```
    /dev/dsk/c7t3d0s0
```

Do you wish to continue? [y/N]y

```
total data kilobytes      = 503744
```

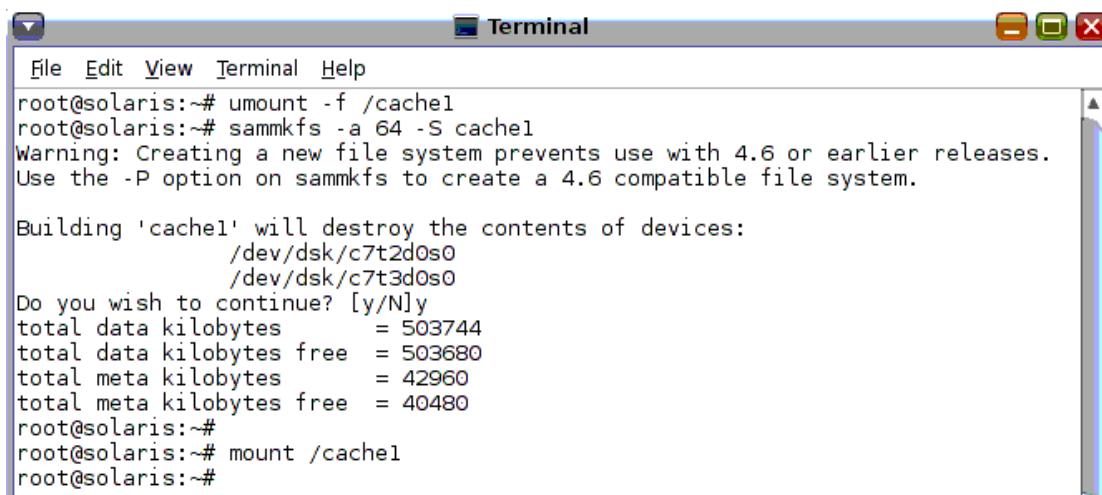
```
total data kilobytes free = 503680
```

```
total meta kilobytes     = 42960
```

```
total meta kilobytes free = 40480
```

```
root@solaris:~# mount /cache1
```

RESULT



```
Terminal
File Edit View Terminal Help
root@solaris:~# umount -f /cache1
root@solaris:~# sammkfs -a 64 -S cache1
Warning: Creating a new file system prevents use with 4.6 or earlier releases.
Use the -P option on sammkfs to create a 4.6 compatible file system.

Building 'cache1' will destroy the contents of devices:
    /dev/dsk/c7t2d0s0
    /dev/dsk/c7t3d0s0
Do you wish to continue? [y/N]y
total data kilobytes      = 503744
total data kilobytes free = 503680
total meta kilobytes     = 42960
total meta kilobytes free = 40480
root@solaris:~#
root@solaris:~# mount /cache1
root@solaris:~#
```

Figure 5: Rebuild SAM file system results

COMMAND

- Start the Firefox Web Browser by clicking on the Firefox icon on the top left of the HOL2925_Solaris-HSM VM

- Login to the <https://127.0.0.1:6789> using user root; password l1admin



Figure 6: Web Console login screen

RESULT



Figure 7: Web Console

COMMAND:

Click on SAM-QFS Manger

RESULT

The screenshot shows the SAM-QFS Manager interface for the host 'solaris.demo.lan'. The main window displays a table of file systems. The 'cache1' file system is highlighted with a red box. The table columns are File System Name, Mount Point, Description, NFS Shared, and Disk Usage (Total).

File System Name	Mount Point	Description	NFS Shared	Disk Usage (Total)
<zfs>	/	zfs		60% (7.26 GB)
<zfs>	/var	zfs		30% (4.11 GB)
<zfs>	/var/share	zfs		0% (2.84 GB)
<zfs>	/export	zfs		0% (2.84 GB)
<zfs>	/export/home	zfs		0% (2.84 GB)
<zfs>	/export/home/sam	zfs		0% (2.84 GB)
<zfs>	/rpool	zfs		0% (2.84 GB)
cache1	/cache1	qfs - server archiving		3% (503.74 MB)

Figure 8: SAM-QFS Manager used to configure the SAM-QFS file system and archive policies

You can see the **cache1** file system has already been created. Click on the file system name (samfs1) and you can see the devices (LUNs) that were used for the data and the metadata for this file system. Click 'File Systems' on the left to return to the list of file systems.

COMMAND

- Click on "Archive Media" on the left
- Click on "Tape Libraries" on the left to review the tape based archive devices

RESULTS

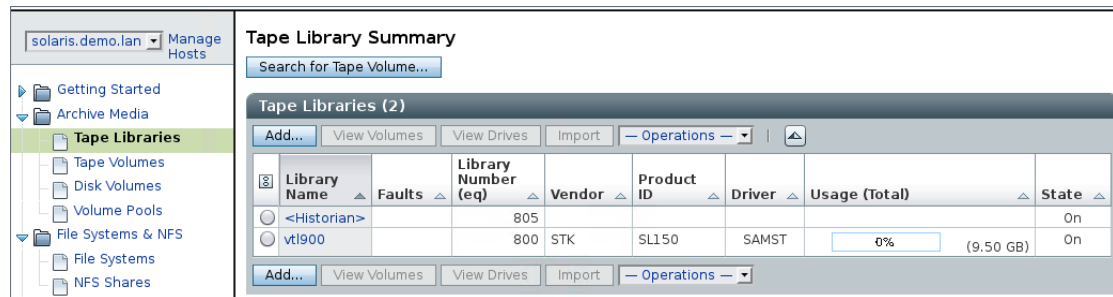


Figure 9: Tape Library Summary available for an archive copy

COMMAND

- Click on “Tape volumes” on the left to display the virtual media available for tape archive copies

RESULT

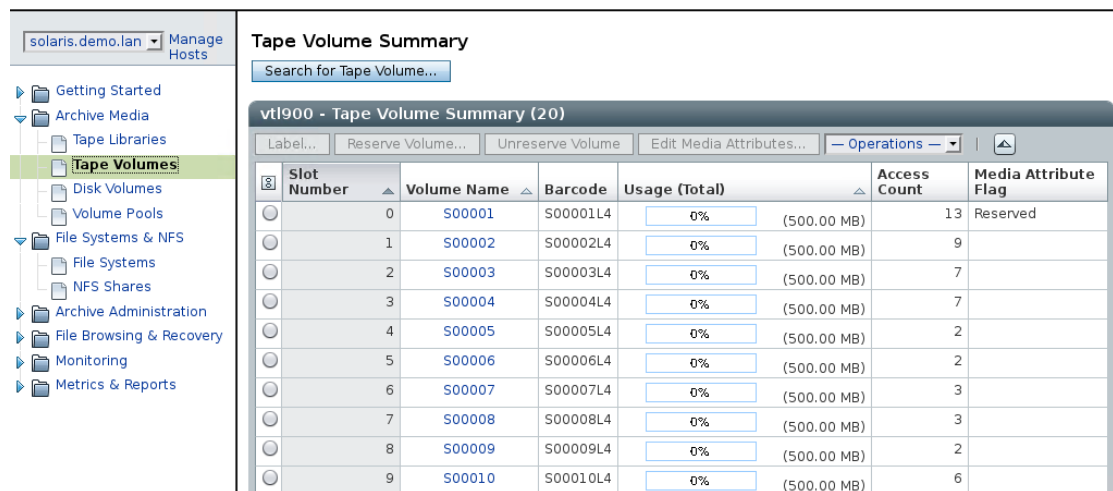
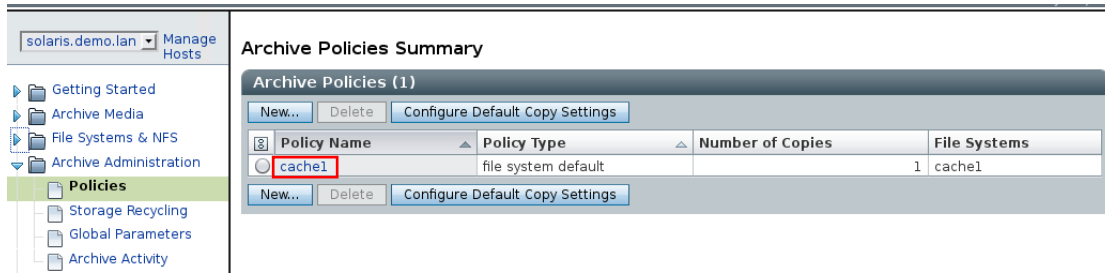


Figure 10: Tape Volumes available for tape archive copies

COMMAND

- Click on “Archive Administration” on the left
- Click on “Policy” on the left to review the Archive Policy Summary
- Click “cache1”



RESULT



Figure 11: File system cache1 archive policy

Now we can see all the files written into cache1 file system will be archived into tapes.

Here we need to modify the “Archive Age”. In this HOL, we need the files being archived as soon as possible, so we can set the archive age to 1 Minutes. To activate the change, click “Save”

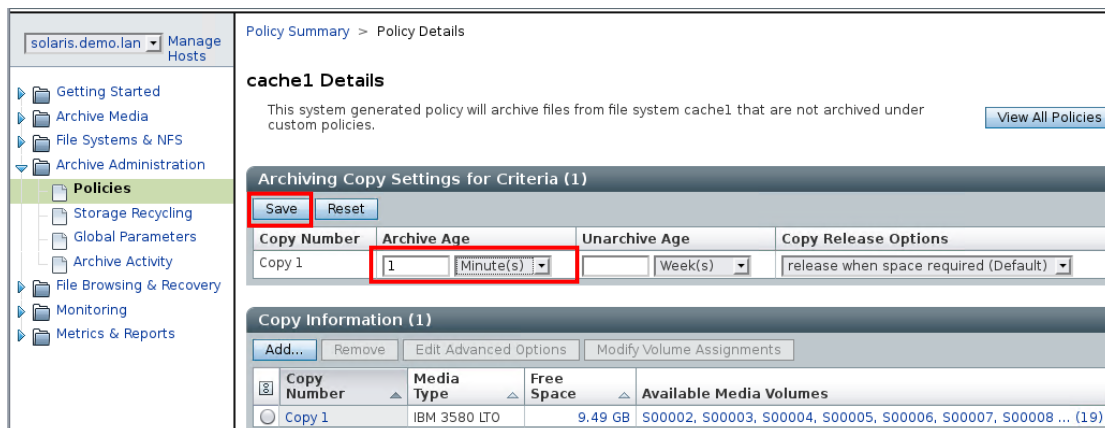


Figure 12: File system cache1 archive Age Setting

Configure Swift Storage Node(HOL2925_Solaris_Swift)

COMMAND:

- Switch to the HOL2925_Solaris_Swift Virtual Machine
 - Click on the Virtual Box icon at the bottom of your screen
 - All VM windows and the VM Manager will be displayed at the bottom.
 - Click on the HOL2925_Solaris_Swift window
- Log into the HOL2925_Solaris_Swift system with user 'root'; password 'l1admin'
(That is lower case l as in lucky, number 1,admin)

RESULT



Figure 13: Solaris Swift Login Screen. Login using **root** and **l1admin**

COMMAND:

- Open a terminal in the HOL2925_Solaris_Swift

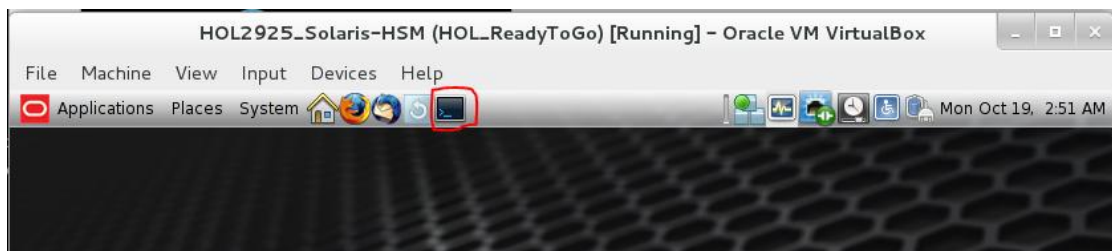


Figure 14: Open terminal

- Use the following command to mount the HSM cache1 client
The cache1 has already been configured as the HSM client file system:

```

root@solSwift:~# cat /etc/opt/SUNWsamfs/mcf
# mcf
#
# Generated by fsmgmt Mon Jul 28 00:04:09 2014
#
# A backup copy of the previously existing file can be
# found in the /etc/opt/SUNWsamfs/cfg_backups directory
#
# Equipment      Eq      Eq      Family   Device   Additional
# Identifier     Ord     Type    Set      State    Parameters
# -----      ---     ---     - - - - - - - - - -
#
# /etc/opt/SUNWsamfs/mcf
#
# SAM-QFS Master Configuration File for the Sam-in-the-Box simulator
#
#-----
#
cache1           10      ma      cache1 -   shared
nodev           11      mm      cache1 -
/dev/dsk/c1t2d0s0  12      md      cache1 -

```

```

root@solSwift:~# cat /etc/opt/SUNWsamfs/hosts.cache1
#
# Host file for family set 'cache1'
#
# Host              Host IP          Server   Not   Server
# Name              Addresses        Priority Used   Host
# -----          - - - - - - - - - -
solaris.demo.lan   10.0.0.10       1        -     server
solSwift.demo.lan 10.0.0.200      -         -

```

Mount the cache1 file system on the HOL2925_Solaris_Swift


```

root@solSwift:~# mkdir /srv/node/cache1
root@solSwift:~# cat /etc/vfstab
...
cache1 - /srv/node/cache1 samfs - yes shared
root@solSwift:~# mount cache1
root@solSwift:~# ls /srv/node/cache1/
lost+found
root@solSwift:~# chown -R swift:swift /srv/node/cache1
root@solSwift:~# chmod 755 /srv/node/cache1

```

COMMAND:

- Setup the Swift function on the HOL2925_Solaris_Swift
 - The Openstack Swift package has already been installed on the server

```

root@solSwift:~# pkg info swift swiftclient
      Name: cloud/openstack/swift
      Summary: OpenStack Swift
      Description: The OpenStack Object Store project, known as Swift, offers cloud
                  storage software so that you can store and retrieve data in
                  virtual containers
      Category: System/Administration and Configuration
(org.opensolaris.category.2008)
      System/Enterprise Management
(org.opensolaris.category.2008)
      System/Virtualization (org.opensolaris.category.2008)
      Web Services/Application and Web Servers
(org.opensolaris.category.2008)
      State: Installed
      Publisher: solaris
      Version: 1.10.0 (Havana 2013.2.3)
Build Release: 5.11
      Branch: 0.175.2.0.0.42.1
Packaging Date: June 23, 2014 01:03:38 AM
      Size: 2.27 MB
      FMRI:
pkg://solaris/cloud/openstack/swift@1.10.0,5.11-0.175.2.0.0.42.1:20140623T
010338Z

```

```

Name: library/python/swiftclient
Summary: Python and command-line clients for the OpenStack Swift API
Description: A python client for the OpenStack Object Storage API. There's
a
Python API (the 'swiftclient' module), and a command-line
script
('swift').
Category: Development/Python (org.opensolaris.category.2008)
System/Administration and Configuration
(org.opensolaris.category.2008)
System/Enterprise Management
(org.opensolaris.category.2008)
State: Installed
Publisher: solaris
Version: 2.1.0
Build Release: 5.11
Branch: 0.175.2.0.0.42.1
Packaging Date: June 23, 2014 01:45:56 AM
Size: 21.42 kB
FMRI:
pkg://solaris/library/python/swiftclient@2.1.0,5.11-0.175.2.0.0.42.1:20140
623T014556Z

```

- Create Rings

A ring represents a mapping between the names of entities stored on disk and their physical location. There are separate rings for accounts, containers, and one object ring per storage policy.

- Build the account ring

```

root@solSwift:~# cd /etc/swift/
root@solSwift:/etc/swift# swift-ring-builder account.builder create
6 1 1
root@solSwift:/etc/swift# swift-ring-builder account.builder add
r1z1-127.0.0.1:6002/cache1 100
Device d0r1z1-127.0.0.1:6002R127.0.0.1:6002/cache1_"" with 100.0
weight got id 0
root@solSwift:/etc/swift# swift-ring-builder account.builder
rebalance
Reassigned 64 (100.00%) partitions. Balance is now 0.00.

```

Note:

(1) The first command is used to configure the ring.

The 6 parameter specifies that 2 to the power of 6 partitions be created.

The next parameter specifies the number of replica copies, which is 1 when Swift is used with Oracle HSM. Oracle HSM will manage additional copies of data.

The final parameter indicates not to move a partition more than once in an hour. This setting is not used for objects with Oracle HSM as we avoid rebalancing Swift nodes.

- (2) The second command is to add file system to the ring. In a standard Swift implementation, there is a separate network for replication and the address would be specified. In this example, we are using the same network for replication so the IP address is the same. The final parameter (100) is the weight which indicates how much data this file system will hold relative to the other file systems. We are distributing this evenly so the weight will be 100 for all file systems.
- (3) The final step is to create the ring file by running the rebalance command

- Build the container ring

```
root@solSwift:/etc/swift# swift-ring-builder container.builder
create 6 1 1
root@solSwift:/etc/swift# swift-ring-builder container.builder add
r1z1-127.0.0.1:6001/cache1 100
Device d0r1z1-127.0.0.1:6002R127.0.0.1:6001/cache1_"" with 100.0
weight got id 0
root@solSwift:/etc/swift# swift-ring-builder container.builder
rebalance
Reassigned 64 (100.00%) partitions. Balance is now 0.00.
```

- Build the object ring

```
root@solSwift:/etc/swift# swift-ring-builder object.builder create 7
1 1
root@solSwift:/etc/swift# swift-ring-builder object.builder add
r1z1-127.0.0.1:6000/cache1 100
Device d0r1z1-127.0.0.1:6002R127.0.0.1:6000/cache1_"" with 100.0
weight got id 0
root@solSwift:/etc/swift# swift-ring-builder object.builder
rebalance
Reassigned 128 (100.00%) partitions. Balance is now 0.00.
```

- Verify

```

root@solSwift:/etc/swift# chown -R swift:swift /etc/swift/*
root@solSwift:/etc/swift# /usr/bin/swift-ring-builder object.builder
object.builder, build version 1
128 partitions, 1.000000 replicas, 1 regions, 1 zones, 1 devices, 0.00
balance
The minimum number of hours before a partition can be reassigned is
1
Devices:

```

id	region	zone	ip address	port	replication
0	1	1	127.0.0.1	6002	127.0.0.1
6002	cache1	100.00	128	0.00	

- Verify Configuration Files on HOL2925_Solaris_Swift window

The swift configuration files are in the /etc/swift directory. For simplicity this example uses the default values for all configuration files except for the proxy server.

The proxy server needs to have authentication setup. This example disables keystone authentication and uses the default simple tmpauth authentication service using the default test account.

Check the file /etc/swift/proxy-server.conf.

```

root@solSwift:/etc/swift# vi /etc/swift/proxy-server.conf
[DEFAULT]
# bind_ip = 0.0.0.0
bind_port = 8080
# bind_timeout = 30
# backlog = 4096
# swift_dir = /etc/swift
# user = swift

[pipeline:main]
#pipeline = catch_errors healthcheck proxy-logging cache bulk slo
ratelimit tempauth authtoken keystoneauth container-quotas
account-quotas proxy-logging proxy-server
pipeline = catch_errors healthcheck proxy-logging cache bulk slo
ratelimit tempauth container-quotas account-quotas proxy-logging
proxv-server

```

- Start Swift Services

```
root@solSwift:/etc/swift# svcadm enable memcached
root@solSwift:/etc/swift# svcs memcached
STATE          STIME      FMRI
online         0:01:36   svc:/application/database/memcached:default
```

```
root@solSwift:/etc/swift# SWIFT_SVC="swift-proxy-server
swift-account-server swift-container-server swift-object-server
swift-container-updater swift-object-updater swift-account-reaper
swift-object-expirer"
root@solSwift:/etc/swift# svcadm enable $SWIFT_SVC
root@solSwift:/etc/swift# svcs | grep swift
online 19:25:39 svc:/application/openstack/swift/swift-container-updater:default
online 19:25:39 svc:/application/openstack/swift/swift-account-server:default
online 19:25:39 svc:/application/openstack/swift/swift-object-updater:default
online 19:25:39 svc:/application/openstack/swift/swift-container-server:default
online 19:25:39 svc:/application/openstack/swift/swift-account-reaper:default
online 19:25:40 svc:/application/openstack/swift/swift-object-server:default
online 19:25:40 svc:/application/openstack/swift/swift-proxy-server:default
online 19:25:40 svc:/application/openstack/swift/swift-object-expirer:default
```

COMMAND:

- Verify Swift Function
 - su to swift user, check the environment setting

```
root@solSwift:~# su - swift
swift@solSwift.demo.lan:~$ cat .profile
export ST_AUTH=http://localhost:8080/auth/v1.0
export ST_USER=test:tester
export ST_KEY=testing
```

- List swift container, should be empty at this point

```
swift@solSwift.demo.lan:~$ swift list
```

- Create your first container, then list

```
swift@solSwift.demo.lan:~$ swift post container01
swift@solSwift.demo.lan:~$ swift list
container01
```

- check swift status

```
swift@solSwift.demo.lan:~$ swift stat

Account: AUTH_test
Containers: 1
Objects: 0
Bytes: 0
Accept-Ranges: bytes
X-Timestamp: 1429692573.70117
X-Trans-Id: txbc874459dc5842ebe99cc-0055dd8a1d
Content-Type: text/plain; charset=utf-8
```

- create an test file, and upload it to the container

```
swift@solSwift.demo.lan:~$ cd /var/tmp
swift@solSwift.demo.lan:/var/tmp$ cat test.txt
Hello World!
From OOW 2105
swift@solSwift.demo.lan:/var/tmp$ swift upload container01 ./test.txt
test.txt
```

- list the container01, see if the file is uploaded

```
swift@solSwift.demo.lan:~$ swift list container01
test.txt
swift@solSwift.demo.lan:/var/tmp$ swift stat container01
test.txt

Account: AUTH_test
Container: container01
Object: test.txt
Content Type: text/plain
Content Length: 28
Last Modified: Tue, 22 Sep 2015 08:25:39 GMT
ETag: 8d62fc13346406da8ae722089bceb038
Meta Mtime: 1442907331.056608
Accept-Ranges: bytes
X-Timestamp: 1442910339.22104
X-Trans-Id: txc0d13ba65aa963c1988ba-005601109e
```

- Test downloads function. First go to another path (for example: /tmp). Then download the file from the swift container.

```

swift@solSwift.demo.lan:~$ cd /tmp
swift@solSwift.demo.lan:/tmp$ ls
dbus-0Ts5Nvj1Y1  gdm-auth-cookies-neaygg  pulse-5ffZcRlb48Na
dbus-Xh8rYZU796  hspferdata_root          ssh-XXXXCDaWPh
swift@solSwift.demo.lan:/tmp$ swift download container01 test.txt
test.txt [auth 0.022s, headers 0.122s, total 0.122s, 0.000 MB/s]
swift@solSwift.demo.lan:/tmp$ ls
dbus-0Ts5Nvj1Y1  hspferdata_root  test.txt
dbus-Xh8rYZU796  pulse-5ffZcRlb48Na
gdm-auth-cookies-neaygg  ssh-XXXXCDaWPh
swift@solSwift.demo.lan:/tmp$ cat test.txt
Hello World!
From OOW 2105

```

Verify Tape Archive Function on HSM QFS Server (HOL2925_Solaris-HSM)

COMMAND:

- Switch to the HOL2925_Solaris-HSM, then open the HSM management in the firefox. Login as root, password l1admin



- Click “File Browsing & Recovery”, “File Browser”
Input “/cache1” in the “Current Directory”, then click “Apply”

This page displays the file entries of a file directory. You can view the details of a file, or perform file operations if the file resides in a SAM-QFS file system. Use the mount point helper drop down menu to quickly navigate to the mount point of a file system.

Current Directory: — Select Mount P

Filter Criteria: No Filter Maximum Entries:

File/Directory Name	User	Size	Last Modified	Online Status	Archive Copies
Up One Level					
.archive	86	4.09 kB	October 20, 2015 7:10:40 PM	Online	
.domain	root	0 B	October 20, 2015 7:10:40 PM	Online	
.fuid	root	0 B	October 20, 2015 7:10:40 PM	Online	
.inodes	86	786.43 kB	October 20, 2015 7:10:40 PM	Online	
.stage	86	4.09 kB	October 20, 2015 7:10:45 PM	Online	
accounts	86	4.09 kB	October 21, 2015 2:30:19 AM	Online	
containers	86	4.09 kB	October 21, 2015 2:30:19 AM	Online	
lost+found	86	16.38 kB	October 20, 2015 7:10:20 PM	Online	
objects	86	4.09 kB	October 21, 2015 2:30:53 AM	Online	
tmp	86	4.09 kB	October 21, 2015 2:30:53 AM	Online	

- Click “objects”, “55”, “ee2”, “6e756f26ad8e76f8d531f37c460eaae2”, “1445423453.25503.data”

This page displays the file entries of a file directory. You can view the details of a file, or perform file operations if the file resides in a SAM-QFS file system. Use the mount point helper drop down menu to quickly navigate to the mount point of a file system.

Current Directory: — Select Mount Point

Filter Criteria: No Filter Maximum Entries:

File/Directory Name	User	Size	Last Modified	Online Status	Archive Copies
Up One Level					
1445423453.25503.data	86	28 B	October 21, 2015 2:30:53 AM	Online	

Actually, this is the file we just uploaded into the Swift

```

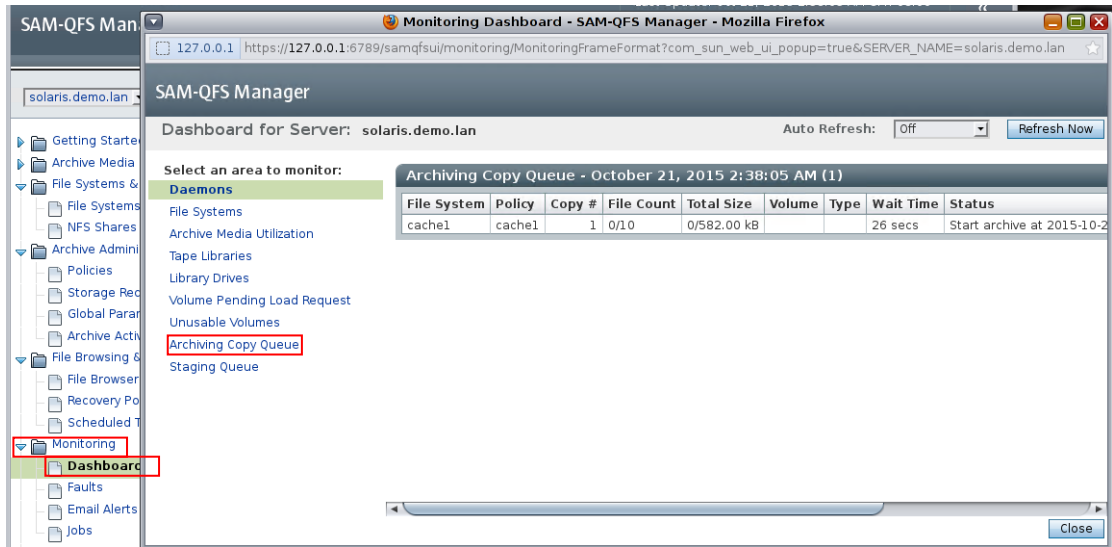
root@solaris:~# cat /cache1/objects/55/ee2/6e756f26ad8e76f8d531f37c460eaae2/1445423453.25503.data
Hello World!
From OOW 2105

root@solaris:~#

```

- According to our archive policy setting, all the files being written into the cache1 HSM file system will be automatically archived to the tape in 1 minute.
Go to “Monitoring”, “Dashboard”, “Archiving Copy Queue” to monitor the Archive

operation.



- After the archive is done, go back to “File Browser”, find /cache1/objects/55/ee2/6e756f26ad8e76f8d531f37c460eaae2. You will see that the 1445423453.25503.data file has already been archive.

