Open HA Cluster on OpenSolaris™

Example configuration to run a build system and development cluster environment on a single system

Combining technologies to work

Thorsten Frueauf, 09/03/2009

This white paper describes how to configure a build and development environment for Open HA Cluster on a physical system running OpenSolaris, using technologies like VirtualBox, Weak Membership, Crossbow, Clearview, IPsec and COMSTAR.

Table of Contents

1 Introduction

For developers it is often convenient to have all tools necessary for their work in one place, ideally on a laptop for maximum mobility.

For system administrators, it is often critical to have a test system on which to try out things and learn about new features. Of course the system needs to be low cost and transportable to anywhere they need to be.

HA Clusters are often perceived as complex to setup and resource hungry in terms of hardware requirements.

This white paper explains how to setup a single x86 based system (like a laptop) with OpenSolaris, configuring a build environment for Open HA Cluster and using VirtualBox to setup a two node cluster.

OpenSolaris technologies like Crossbow (to create virtual networking adapters), COMSTAR (to setup non-shared storage as iSCSI targets and using them as iSCSI initiators), ZFS (to mirror the iSCSI targets), Clearview (the new architecture for IPMP), and IPsec (to secure the cluster private interconnect traffic) are used for the host system and VirtualBox guests to configure Open HA Cluster. The image packaging system (IPS) is being used to deploy the build packages into the guests. Open HA Cluster technologies like weak membership (to not require an extra quorum device) and the integration into OpenSolaris technologies are leveraged to setup three typical FOSS applications: HA MySQL, HA Tomcat and scalable Apache webserver.

The instructions can be used as a step-by-step guide to setup any x86 based system that is capable to run OpenSolaris. In order to try out if your system works, simply boot the OpenSolaris live CD-ROM and confirm with the Device Driver Utility (DDU) that all required components are able to run. A hardware compatibility list can be found at http://www.sun.com/bigadmin/hcl/.

2 Host Configuration

The example host system used throughout this white paper is a Toshiba Tecra® M10 Laptop with the following hardware specifications:

- 4 GB main memory
- Intel® Core™2 Duo P8400@2.26Ghz
- 160 GB SATA hard disk
- 1 physical network nic (1000 Mbit) – `e1000g0`
- 1 wireless network nic (54 Mbit) – `iwh0`

The system should at least have a minimum of 3GB of main memory in order to host two VirtualBox OpenSolaris guest systems.

2.1 BIOS Configuration

The Toshiba Tecra M10 has been updated to the BIOS version 2.0. By default, the option to use the CPU virtualization capabilities is disabled.
This option needs to be enabled in order to use 64bit guests with VirtualBox:

BIOS screen SYSTEM SETUP (1/3) → OTHERS
Set “Virtualization Technology” to “Enabled”.

2.2 OpenSolaris Configuration

In this example OpenSolaris 2009.06 build 111 has been installed on the laptop.

For generic information on how to install OpenSolaris 2009.06, see the official guide at http://dlc.sun.com/osol/docs/content/2009.06/getstart/index.html.

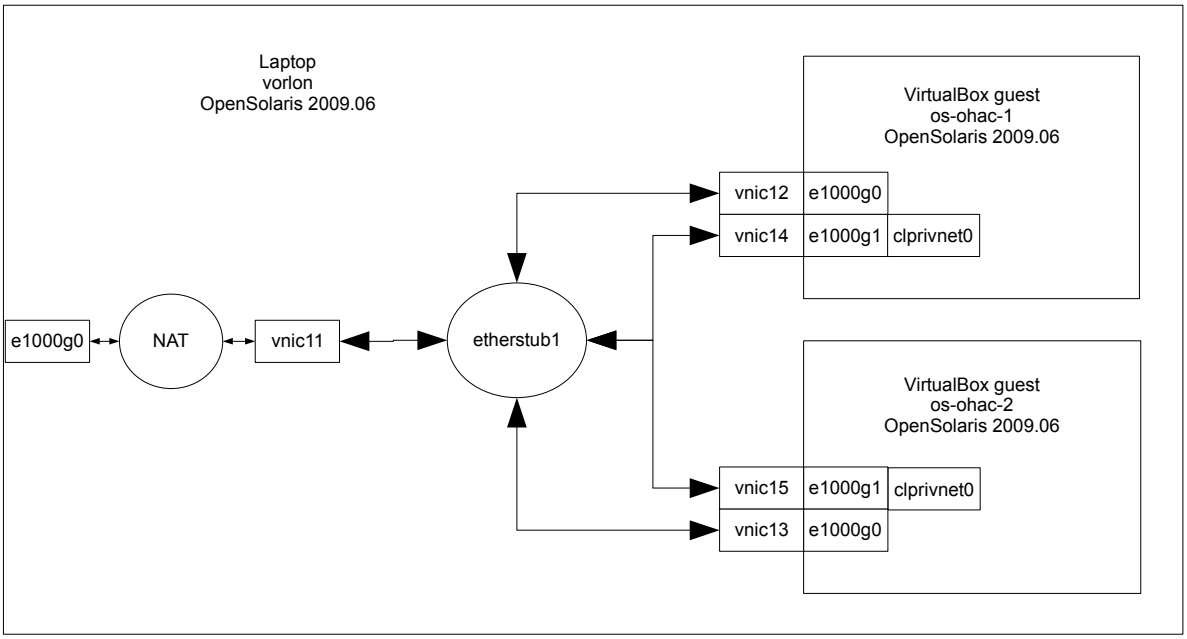
The following configuration choices will be used as an example:

- Hostname: vorlon
- User: ohacdemo

2.2.1 Network Configuration

By default OpenSolaris enables the Network Auto-Magic (NWAM) service.

Since NWAM is currently designed to use only one active NIC at a time (and actively unconfigures all other existing NICs), the following steps are required to disable NWAM and setup a static networking configuration. The diagram shows an overview of the target network setup:



The following IP addresses will be used:

IP Address	Alias	Comment
10.0.2.100	vorlon-int	vnic11
10.0.2.101	os-ohac-1	vnic12
10.0.2.102	os-ohac-2	vnic13
10.0.2.110	os-ohac-lh1	
10.0.2.111	os-ohac-lh2	

Disable the NWAM service:

```
vorlon# svcadm disable nwam
```

Create the virtual network:

```
vorlon# dladm create-etherstub etherstub1
```

```
vorlon# dladm create-vnic -l etherstub1 vnic11
vorlon# dladm create-vnic -l etherstub1 vnic12
vorlon# dladm create-vnic -l etherstub1 vnic13
vorlon# dladm create-vnic -l etherstub1 vnic14
vorlon# dladm create-vnic -l etherstub1 vnic15
```

Add the IP addresses and aliases to /etc/inet/hosts:

```
vorlon# vi /etc/inet/hosts
::1 vorlon vorlon.local localhost loghost
127.0.0.1  vorlon.local localhost loghost
#
# Internal network for VirtualBox
10.0.2.100      vorlon-int
10.0.2.101      os-ohac-1
10.0.2.102      os-ohac-2
10.0.2.110      os-ohac-lh1
10.0.2.111      os-ohac-lh2
```

Add the default netmasks for the used subnets to /etc/inet/netmasks:

```
vorlon# vi /etc/inet/netmasks
10.0.1.0   255.255.255.0
10.0.2.0   255.255.255.0
```

Configure the internal host IP used to access the network to the VirtualBox guest:

```
vorlon# vi /etc/hostname.vnic11
vorlon-int
```

Always plumb the vnics used by the VirtualBox guests when booting:

```
vorlon# # touch /etc/hostname.vnic12
/etc/hostname.vnic13 /etc/hostname.vnic14
/etc/hostname.vnic15
```

If you want the VirtualBox guests to be able to reach the external network connected to either e1000g0 or iwh0, then setup ipfilter to perform Network Address Translation (NAT) for the internal virtual network:

```
vorlon# vi /etc/ipf/ipf.conf
pass in all
pass out all

# vi /etc/ipf/ipnat.conf
map e1000g0 10.0.2.0/24 -> 0/32 portmap tcp/udp auto
map e1000g0 10.0.2.0/24 -> 0/32
map iwh0 10.0.2.0/24 -> 0/32 portmap tcp/udp auto
map iwh0 10.0.2.0/24 -> 0/32
```

If you want to make e.g. the tomcat URL configured later in section 4.7 accessible from outside of the hosts external network, add the following line to /etc/ipf/ipnat.conf:

```
rdr e1000g0 0.0.0.0/0 port 8080 -> 10.0.2.110 port 8080
tcp
```

Configure the public network on e1000g0 depending on your individual setup.

The following example assumes a static IP configuration:

```
vorlon# vi /etc/hostname.e1000g0
10.0.1.42

vorlon# vi /etc/defaultrouter
10.0.1.1

vorlon# vi /etc/resolv.conf
nameserver 10.0.1.1

vorlon# vi /etc/nsswitch.conf
     => add dns to the hosts keyword:
hosts:      files dns
```

Enable the static networking configuration:

```
vorlon# svcadm enable svc:/network/physical:default
```

Enable the service for ipfilter:

```
vorlon# svcadm enable svc:/network/ipfilter:default
```

Enable IPv4 forwarding:

```
vorlon# routeadm -u -e ipv4-forwarding
```

2.2.2 Filesystem Configuration

Create some additional file systems for:

- crash dumps created for the host system (/var/crash)
- downloads of various files (/data)
- building Open HA Cluster source (/build)
- local IPS repositories (/ipsrepo)
- VirtualBox Images (/VirtualBox-Images)

```
vorlon# zfs create -o mountpoint=/var/crash -o
compression=on rpool/crash
vorlon# mkdir /var/crash/vorlon
vorlon# zfs create -o mountpoint=/data rpool/data
```

```
vorlon# zfs create -o mountpoint=/build -o compression=on
rpool/build
vorlon# zfs create -o mountpoint=/ipsrepo rpool/ipsrepo
vorlon# zfs create -o mountpoint=/VirtualBox-Images
rpool/vbox-images
vorlon# chown ohacdemo:staff /data /build /ipsrepo
/VirtualBox-Images
```

2.3 Setup Open HA Cluster build environment

The current version of the instructions on how to build the Open HA Cluster framework source gate on OpenSolaris can be found at:

- http://wiki.genunix.org/wiki/index.php/Compiling_OHAC

and for the agent source gate at:

- http://wiki.genunix.org/wiki/index.php/Compiling_OHAC_Agents

The following describes the steps used as an example.

Install some additional packages from the OpenSolaris IPS repository:

```
vorlon#  pkg publisher
PUBLISHER                             TYPE     STATUS
URI
opensolaris.org          (preferred)  origin   online
http://pkg.opensolaris.org/release/
vorlon# pkg install SUNWperl584core@5.8.4-0.111
vorlon# pkg install SUNWperl584usr@5.6.1-0.111
vorlon# pkg install SUNWtcat@6.0.18,5.11-0.111
vorlon# pkg install SUNWjavadb@10.4.2.1,5.11-0.111
vorlon# pkg install SUNWjdmk-base@0.5.11,5.11-0.111
vorlon# pkg install SUNWcacaort@0.5.11-0.111
vorlon# pkg install SUNWonbld@0.5.11-0.111
vorlon# pkg install SUNWant@1.7.1-0.111
vorlon# pkg install sunstudioexpress
vorlon# cd /opt
vorlon# ln -s SunStudioExpress /opt/SUNWspro
```

Build and install OpenDMK:

```
 - Download the following to /data/Colorado/opendmk
   https://opendmk.dev.java.net/download/opendmk-1.0-b02-
src-dual-01-Oct-2007_19-17-46.zip
   https://opendmk.dev.java.net/download/opendmk-1.0-b02-
binary-plug-01-Oct-2007_19-17-46.jar

 - unzip opendmk source archive:
ohacdemo@vorlon$ cd /build
ohacdemo@vorlon$ unzip /data/Colorado/opendmk/opendmk-
```

```
1.0-b02-src-dual-01-Oct-2007_19-17-46.zip
ohacdemo@vorlon$ cd OpenDMK-src

  # make sure the following command can open a X display.
  # If running from remote, use e.g. "ssh -g -X
<hostname>"
ohacdemo@vorlon$ java -jar
/data/Colorado/opendmk/opendmk-1.0-b02-binary-plug-01-
Oct-2007_19-17-46.jar
        => accept license agreement
        => select install directory
                => /build/OpenDMK-src

 - Build the OpenDMK source:
ohacdemo@vorlon$ /usr/bin/ant buildall

 - Copy the required files to /usr/share/lib/jdmk

vorlon# cd /build/OpenDMK-src/dist
vorlon# cp lib/jdmktk.jar /usr/share/lib/jdmk
vorlon# find etc/mibgen/mib_core.txt | cpio -pdu
/usr/share/lib/jdmk
```

Install four JATO related packages from the Sun Java Web Console 3.0.2 release:

```
    - Download the Sun Java Web Console 3.0.2 for
Solaris/x86 archive from https://cds.sun.com/is-
bin/INTERSHOP.enfinity/WFS/CDS-CDS_SMI-
Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=WC-
302-G-F@CDS-CDS_SMI

    Copy webconsole3.0.2-solx86.tar.gz to
/data/Colorado/SunJavaWebconsole/

  - Unpack the archive:
ohacdemo@vorlon$ mkdir /var/tmp/webconsole
ohacdemo@vorlon$ cd /var/tmp/webconsole
ohacdemo@vorlon$ gzcat
/data/Colorado/SunJavaWebconsole/webconsole3.0.2-
solx86.tar.gz | tar xf -

  - Install the following packages:

vorlon# yes | pkgadd -d . SUNWjato SUNWmctag SUNWmcon
vorlon# yes | pkgadd -d SunOS-5.11-i386 SUNWmcos
```

Download the following archives and install them:

```
ohacdemo@vorlon$ cd /data/Colorado
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/on/downloads/b111/on-src.tar.bz2
```

```
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/20080925/on-
filelist
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/colorado/2009052
0/colorado-src-20090520.tar.bz2
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/colorado/2009052
0/coloradods-src-20090520.tar.bz2
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/colorado/2009052
0/ohac-tools-20090520.i386.tar.bz2
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/20080925/ohac-
closed-bins-20080925.i386.tar.bz2
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/20080925/ohac-ext-
pkgs-20080925.i386.tar.bz2
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/20080925/ohac-ref-
proto-20080925.i386.tar.bz2
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/20080925/ohacds-
closed-bins-20080925.i386.tar.bz2
ohacdemo@vorlon$ wget -c
http://dlc.sun.com/osol/ohac/downloads/20080925/ohacds-
ext-pkgs-20080611.i386.tar.bz2

vorlon# cd /opt
vorlon# bzcat /data/Colorado/ohac-tools-
20090520.i386.tar.bz2 | tar xf -
vorlon# bzcat /data/Colorado/ohac-closed-bins-
20080925.i386.tar.bz2 | tar xf -
vorlon# bzcat /data/Colorado/ohac-ext-pkgs-
20080925.i386.tar.bz2 | tar xf -
vorlon# bzcat /data/Colorado/ohac-ref-proto-
20080925.i386.tar.bz2 | tar xf -
vorlon# bzcat /data/Colorado/ohacds-closed-bins-
20080925.i386.tar.bz2 | tar xf -
vorlon# bzcat /data/Colorado/ohacds-ext-pkgs-
20080611.i386.tar.bz2 | tar xf -

ohacdemo@vorlon$ mkdir /var/tmp/colorado
ohacdemo@vorlon$ cd /var/tmp/colorado
ohacdemo@vorlon$ bzcat /data/Colorado/on-src.tar.bz2 |
tar xpf -
vorlon# cat /data/Colorado/on-filelist | cpio -pdum /
```

Extract the cluster agent and framework source from the archives, build them and submit the IPS packages to a local repository:

```
ohacdemo@vorlon$ cd /build
```

```
ohacdemo@vorlon$ bzcat /data/Colorado/colorado-src-
20090520.tar.bz2 | tar xf -
ohacdemo@vorlon$ bzcat /data/Colorado/coloradods-src-
20090520.tar.bz2 | tar xf -

ohacdemo@vorlon$ /opt/scbld/bin/depotctl start -d
/ipsrepo/repo-1 -p 7376
Creating repository directory /ipsrepo/repo-1
Starting depot server on port 7376 using dir
/ipsrepo/repo-1
nohup: appending output to `nohup.out'
Started correctly
ohacdemo@vorlon$ cd /build/colorado
ohacdemo@vorlon$ /opt/scbld/bin/nbuild -Dp
IPSREPO_URL=http://localhost:7376/
ohacdemo@vorlon$ cd /build/coloradods
ohacdemo@vorlon$ /opt/scbld/bin/nbuild -Dp
IPSREPO_URL=http://localhost:7376/
```

Building the Open HA Cluster framework (/build/colorado) takes approximately 45 minutes on the system used in this example. The nbuild command will send an email to the user ohacdemo with a summary, after the build has finished.

You can monitor the log files within /build/colorado+5.11+i386/log/log.<timestamp>/ or /build/coloradods+5.11+i386/log/log.<timestamp>/ like log.txt to see the progress.

2.4 Install VirtualBox

Download VirtualBox from http://www.virtualbox.org/wiki/Downloads – select the archive for Solaris and OpenSolaris host on x86/amd64. Consult the VirtualBox User Guide for the complete installation instructions.

In this white paper version 2.2.4 has been used.

```
vorlon# pkgadd -G -d VirtualBoxKern-2.2.4-SunOS-
r47978.pkg
vorlon# pkgadd -G -d VirtualBox-2.2.4-SunOS-r47978.pkg
```

2.5 Install rdesktop

VirtualBox offers to start the guest using the VRDP protocol in order to access the guest console. rdesktop is a VRDP client that allows you to access the VRDP server, which VirtualBox starts for the guest.

```
vorlon# pkg install SUNWrdesktop
```

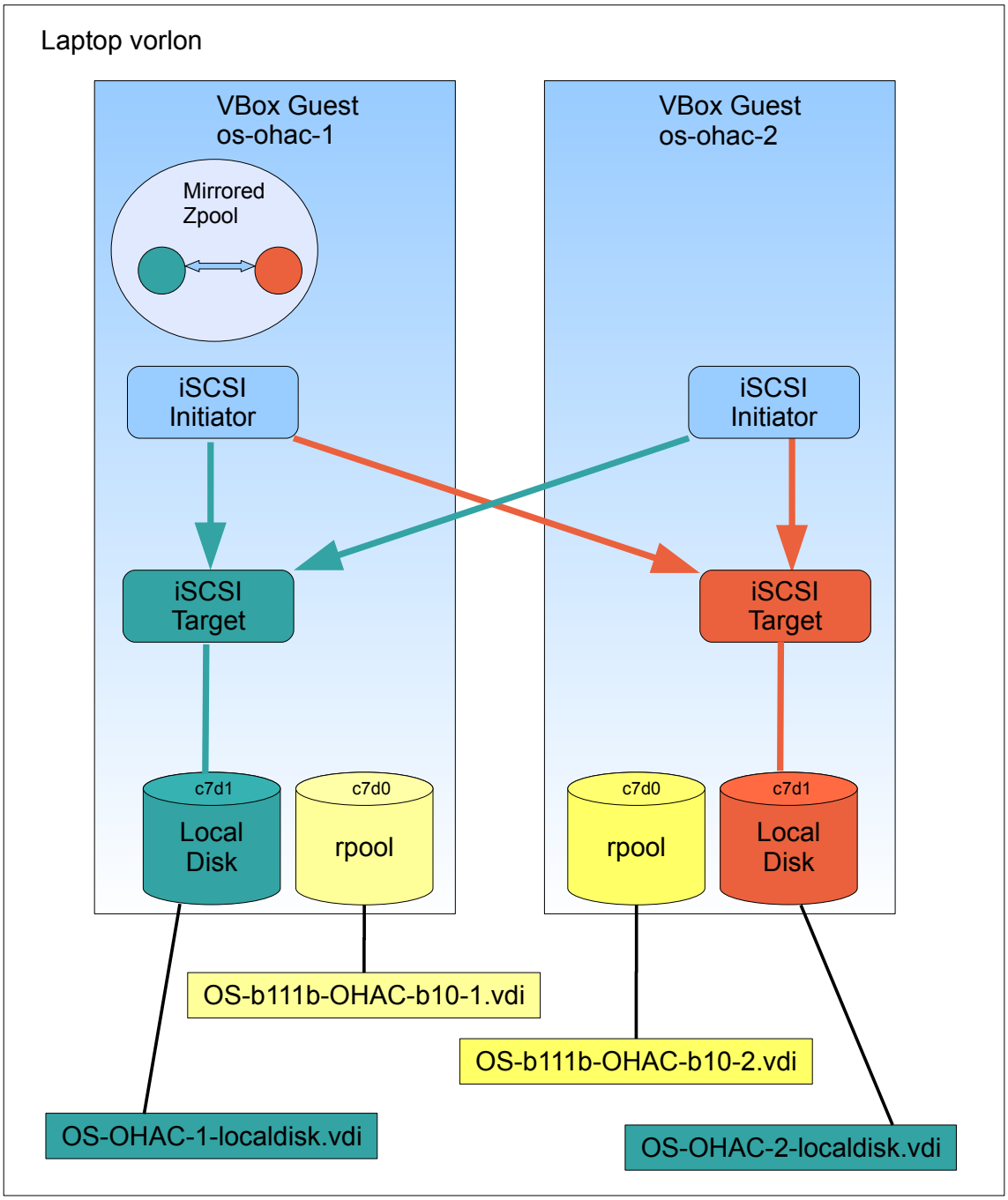
2.6 Download OpenSolaris ISO image

You can download the ISO image from
http://www.opensolaris.com/get/index.jsp.
The following example will assume it to be available as
/data/isos/OpenSolaris/2009.06/b111b2-x86/osol-0906-111b2-x86.iso.

3 VirtualBox Configuration

3.1 VirtualBox Guest Configuration

The following diagram describes the desired disk configuration:



3.1.1 Virtual Disk Configuration

First create the boot disks for the two guests, size 30 GB (= 30720 MB, dynamically expanding image):

- os-ohac-1 will use OS-b111b-OHAC-b10-1.vdi
- os-ohac-2 will use OS-b111b-OHAC-b10-2.vdi

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage createhd
--filename /VirtualBox-Images/OS-b111b-OHAC-b10-1.vdi
--size 30720 --format VDI --variant Standard --remember
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%.
..100%
Disk image created. UUID: 641be421-a838-4ac2-9ace-
083aa1775f99
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage createhd
--filename /VirtualBox-Images/OS-b111b-OHAC-b10-2.vdi
--size 30720 --format VDI --variant Standard --remember
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%.
..100%
Disk image created. UUID: 34a938d9-9e65-4253-887a-
2948d126deef
```

Then create the local disks to be used later for the COMSTAR/iSCSI configuration, size 30GB (=30720 MB, dynamically expanding image):

- os-ohac-1 will use OS-OHAC-1-localdisk.vdi
- os-ohac-2 will use OS-OHAC-2-localdisk.vdi

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage createhd
--filename /VirtualBox-Images/OS-OHAC-1-localdisk.vdi
--size 30720 --format VDI --variant Standard --remember
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%.
..100%
Disk image created. UUID: cfa3cef0-316e-4004-a0d6-
1160e96677ce
```

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage createhd
--filename /VirtualBox-Images/OS-OHAC-2-localdisk.vdi
--size 30720 --format VDI --variant Standard --remember
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%.
..100%
Disk image created. UUID: 89fd09cc-57bf-47f9-b9d9-
0dd3ca732d8b
```

3.1.2 Virtual Machine Configuration

Determine the MAC addresses used by the vnics configured in section 2.2.1:

```
ohacdemo@vorlon$ dladm show-vnic
LINK         OVER         SPEED  MACADDRESS
MACADDRTYPE         VID
vnic11       etherstub1   0      2:8:20:fa:bf:c
random              0
vnic12       etherstub1   0      2:8:20:d5:47:9d
random              0
vnic13       etherstub1   0      2:8:20:e2:99:94
random              0
vnic14       etherstub1   0      2:8:20:0:aa:4d
random               0
vnic15       etherstub1   0      2:8:20:f2:98:ad
random               0
```

The following shows which vnic is used by which VirtualBox guest:

VirtualBox Guest Name	VNIC used	MAC address
OS-b111b-OHAC-b10-1	vnic12	020820D5479D
	vnic14	02082000AA4D
OS-b111b-OHAC-b10-2	vnic13	020820E29994
	vnic15	020820F298AD

It is critical that the MAC address configured with the VirtualBox guest exactly matches with the MAC address configured for the corresponding vnic, otherwise network communication will not work.

Configure the virtual machines:

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage createvm
--name OS-b111b-OHAC-b10-1 -ostype OpenSolaris_64 –
register
```

```
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.
Virtual machine 'OS-b111b-OHAC-b10-1' is created and
registered.
UUID: 44b912d0-5e3d-4063-9db4-47b3f5575701
Settings file:
'/export/home/ohacdemo/.VirtualBox/Machines/OS-b111b-
OHAC-b10-1/OS-b111b-OHAC-b10-1.xml'
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage modifyvm OS-
b111b-OHAC-b10-1 --memory 1024 -hda /VirtualBox-
Images/OS-b111b-OHAC-b10-1.vdi -hdb /VirtualBox-
Images/OS-OHAC-1-localdisk.vdi --boot1 disk --boot2 dvd
--–dvd /data/isos/OpenSolaris/2009.06/b111b2-x86/osol-
0906-111b2-x86.iso --nic1 bridged --nictype1 82540EM
--cableconnected1 on --bridgeadapter1 vnic12
--macaddress1 020820D5479D --nic2 bridged --nictype2
82540EM --cableconnected2 on --bridgeadapter2 vnic14
--macaddress2 02082000AA4D  --audio solaudio
--audiocontroller ac97 --vrdp on --vrdpport 3390
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.
```

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage createvm
--name OS-b111b-OHAC-b10-2 -ostype OpenSolaris_64
--register
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.

Virtual machine 'OS-b111b-OHAC-b10-2' is created and
registered.
UUID: ce23d951-832b-4d50-9707-495c7ce0d30b
Settings file:
'/export/home/ohacdemo/.VirtualBox/Machines/OS-b111b-
OHAC-b10-2/OS-b111b-OHAC-b10-2.xml'
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage modifyvm OS-
b111b-OHAC-b10-2 --memory 1024 -hda /VirtualBox-
Images/OS-b111b-OHAC-b10-2.vdi -hdb /VirtualBox-
Images/OS-OHAC-2-localdisk.vdi --boot1 disk --boot2 dvd
--dvd /data/isos/OpenSolaris/2009.06/b111b2-x86/osol-
0906-111b2-x86.iso --nic1 bridged --nictype1 82540EM
--cableconnected1 on --bridgeadapter1 vnic13
--macaddress1 020820E29994 --nic2 bridged --nictype2
82540EM --cableconnected2 on --bridgeadapter2 vnic15
--macaddress2 020820F298AD --audio solaudio
--audiocontroller ac97 --vrdp on --vrdpport 3391
VirtualBox Command Line Management Interface Version
2.2.2
```

```
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.
```

3.2 VirtualBox Guest OpenSolaris Configuration

Both VirtualBox guest systems need to get installed with OpenSolaris 2009.06.

For generic information on how to install OpenSolaris 2009.06 see the official guide at http://dlc.sun.com/osol/docs/content/2009.06/getstart/index.html.

In section 3.1.2 the corresponding ISO image has been configured for the guests.

3.2.1 First Guest Installation (OS-b111b-OHAC-b10-1)

Start the virtual machine while on a desktop session on the host:

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage startvm OS-
b111b-OHAC-b10-1
```

This will start the console for OS-b111b-OHAC-b10-1 within the VirtualBox GUI. Perform the following steps:

- Select the keyboard layout
- Select the desktop language
- Once the desktop is running, start the installer by double clicking on the “Install OpenSolaris” icon on the desktop
- Within the Installer:
  - click “Next” after reading the Release Notes
  - Select the first disk from the left for Installation
    - select “Use the whole disk”
    - click “Next”
  - Select the region, location, time zone, data and time
    - click “Next”
  - Select the default language
    - click “Next”
  - Configure Users
    - provide the root password
    - Create a user account – it will get assigned the administrator role. This example uses
      - Name: OHAC Admin
      - Log-in name: ohacdemo
      - password: ohacdemo
    - Enter a unique computer name. Do not use the default. This example will use:
      - Computer name: os-ohac-1
    - click “Next”
  - Read the summary page, if everything is OK, click “Install”
  - After the installation completes, click “Reboot”

The next step is to configure the static networking for os-ohac-1. After the reboot, login as user ohacdemo and perform the following steps in a terminal window:

```
ohacdemo@os-ohac-1:~$ pfexec svcadm disable nwam
ohacdemo@os-ohac-1:~$ pfexec vi /etc/inet/hosts
::1 localhost loghost
127.0.0.1 localhost loghost
#
# Internal network for VirtualBox
10.0.2.100      vorlon-int
10.0.2.101      os-ohac-1 os-ohac-1.local
10.0.2.102      os-ohac-2
10.0.2.110      os-ohac-lh1
10.0.2.111      os-ohac-lh2

ohacdemo@os-ohac-1:~$ pfexec vi /etc/inet/netmasks
10.0.2.0   255.255.255.0

ohacdemo@os-ohac-1:~$ pfexec vi /etc/hostname.e1000g0
os-ohac-1

ohacdemo@os-ohac-1:~$ pfexec vi /etc/defaultrouter
vorlon-int

ohacdemo@os-ohac-1:~$ pfexec svcadm enable
svc:/network/physical:default
```

In case you have the host system connected to external networking, configure a nameservice such as DNS:

```
ohacdemo@os-ohac-1:~$ pfexec vi /etc/resolv.conf
nameserver <nameserver-ip>

ohacdemo@os-ohac-1:~$ pfexec vi /etc/nsswitch.conf
     => add dns to the hosts keyword:
hosts:      files dns
```

In case you want to capture crash dumps for the node:

```
ohacdemo@os-ohac-1:~$ pfexec zfs create -o
mountpoint=/var/crash -o compression=on rpool/crash
ohacdemo@os-ohac-1:~$ pfexec mkdir /var/crash/os-ohac-1
ohacdemo@os-ohac-1:~$ pfexec chmod 700 /var/crash/os-
ohac-1
ohacdemo@os-ohac-1:~$ pfexec dumpadm -y
```

In case you want the guest system to provide the text console during boot instead of the graphical view:

```
ohacdemo@os-ohac-1:~$ pfexec cp -p
/rpool/boot/grub/menu.lst /rpool/boot/grub/menu.lst.orig
ohacdemo@os-ohac-1:~$ pfexec vi /rpool/boot/grub/menu.lst
#splashimage /boot/grub/splash.xpm.gz
#background 215ECA
timeout 30
default 0
#---------- ADDED BY BOOTADM - DO NOT EDIT ----------
title OpenSolaris 2009.06
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/opensolaris
#splashimage /boot/solaris.xpm
#foreground d25f00
#background 115d93
#kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-
BOOTFS,console=graphics
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-
BOOTFS,console=text
module$ /platform/i86pc/$ISADIR/boot_archive
#--------------------END BOOTADM--------------------
```

In case you want the guest system to not run the graphical login, in order to conserve some main memory:

```
ohacdemo@os-ohac-1:~$ pfexec svcadm disable
svc:/application/graphical-login/gdm:default
```

Shutdown the guest:

```
ohacdemo@os-ohac-1:~$ pfexec init 5
```

Remove the OpenSolaris ISO image from future use:

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage modifyvm OS-
b111b-OHAC-b10-1 --dvd none
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.
```

3.2.2 Second Guest Installation (OS-b111b-OHAC-b10-2)

Start the virtual machine while on a desktop session on the host:

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage startvm OS-
b111b-OHAC-b10-2
```

This will start the console for OS-b111b-OHAC-b10-2 within the VirtualBox GUI. Perform the following steps:

- Select the keyboard layout
- Select the desktop language
- Once the desktop is running, start the installer by double clicking on the "Install OpenSolaris" icon on the desktop
- Within the Installer:
  - click "Next" after reading the Release Notes
  - Select the first disk from the left for Installation
    - select "Use the whole disk"
    - click "Next"
  - Select the region, location, time zone, data and time
    - click "Next"
  - Select the default language
    - click "Next"
  - Configure Users
    - provide the root password
    - Create a user account – it will get assigned the administrator role. This example uses
      - Name: OHAC Admin
      - Log-in name: ohacdemo
      - password: ohacdemo
    - Enter a unique computer name. Do not use the default. This example will use:
      - Computer name: os-ohac-2
    - click "Next"
  - Read the summary page, if everything is OK, click "Install"
  - After the installation completes, click "Reboot"

The next step is to configure the static networking for os-ohac-2. After the reboot, login as user ohacdemo and perform the following steps in a terminal window:

```
ohacdemo@os-ohac-2:~$ pfexec svcadm disable nwam
ohacdemo@os-ohac-2:~$ pfexec vi /etc/inet/hosts
::1 localhost loghost
127.0.0.1 localhost loghost
#
# Internal network for VirtualBox
10.0.2.100      vorlon-int
10.0.2.101      os-ohac-1
10.0.2.102      os-ohac-2 os-ohac-2.local
10.0.2.110      os-ohac-lh1
10.0.2.111      os-ohac-lh2

ohacdemo@os-ohac-2:~$ pfexec vi /etc/inet/netmasks
10.0.2.0   255.255.255.0

ohacdemo@os-ohac-2:~$ pfexec vi /etc/hostname.e1000g0
os-ohac-2

ohacdemo@os-ohac-2:~$ pfexec vi /etc/defaultrouter
vorlon-int
```

```
ohacdemo@os-ohac-2:~$ pfexec svcadm enable
svc:/network/physical:default
```

In case you have the host system connected to external networking, configure the nameservice like DNS:

```
ohacdemo@os-ohac-2:~$ pfexec vi /etc/resolv.conf
nameserver <nameserver-ip>

ohacdemo@os-ohac-2:~$ pfexec vi /etc/nsswitch.conf
     => add dns to the hosts keyword:
hosts:      files dns
```

In case you want to capture crash dumps for the node:

```
ohacdemo@os-ohac-2:~$ pfexec zfs create -o
mountpoint=/var/crash -o compression=on rpool/crash
ohacdemo@os-ohac-2:~$ pfexec mkdir /var/crash/os-ohac-2
ohacdemo@os-ohac-2:~$ pfexec chmod 700 /var/crash/os-
ohac-2
ohacdemo@os-ohac-2:~$ pfexec dumpadm -y
```

In case you want the guest system to provide the text console instead of the graphical view:

```
ohacdemo@os-ohac-2:~$ pfexec cp -p
/rpool/boot/grub/menu.lst /rpool/boot/grub/menu.lst.orig
ohacdemo@os-ohac-2:~$ pfexec vi /rpool/boot/grub/menu.lst
#splashimage /boot/grub/splash.xpm.gz
#background 215ECA
timeout 30
default 0
#---------- ADDED BY BOOTADM - DO NOT EDIT ----------
title OpenSolaris 2009.06
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/opensolaris
#splashimage /boot/solaris.xpm
#foreground d25f00
#background 115d93
#kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-
BOOTFS,console=graphics
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-
BOOTFS,console=text
module$ /platform/i86pc/$ISADIR/boot_archive
#--------------------END BOOTADM-------------------
```

In case you want the guest system to not run the graphical login, logout from the gnome session and login through the text console as user ohacdemo:

```
ohacdemo@os-ohac-2:~$ pfexec svcadm disable
```

```
svc:/application/graphical-login/gdm:default
```

Shutdown the guest:

```
ohacdemo@os-ohac-2:~$ pfexec init 5
```

Remove the OpenSolaris ISO image from future use:

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage modifyvm OS-
b111b-OHAC-b10-2 --dvd none
VirtualBox Command Line Management Interface Version
2.2.2
(C) 2005-2009 Sun Microsystems, Inc.
All rights reserved.
```

3.3 Getting Crash dumps from OpenSolaris guests

Sometimes it is necessary for debugging purposes to create a crash dump of an OpenSolaris guest, either because it is hung or there is no other way to interact with it, or because a specific state of the system is of interest for further analysis.

3.3.1 Booting OpenSolaris with kernel debugger enabled

The first step is to boot the OpenSolaris guest with the kernel debugger enabled. This step can be used for a one-time kernel debugger boot:

```
o when the grub line comes up, hit 'e'
o go to the splashimage line and hit 'd' to delete it
o go to the foreground line and hit 'd' to delete it
o go to the background line and hit 'd' to delete it
o go to the kernel$ line and hit 'e' to EDIT it
    o hit backspace/delete to remove ",console=graphics"
    o add “ -k” to the line
        o the line should now look like
          kernel$ /platform/i86pc/kernel $ISADIR/unix -B
$ZFS-BOOTFS -k
    o hit return to enter changes and go back
o hit 'b' to boot
```

If you want to always boot with the kernel debugger enabled, the above change needs to be made to the /rpool/boot/grub/menu.lst file to the corresponding entry. Example, add the following:

```
ohacdemo@vorlon$ pfexec vi /rpool/boot/grub/menu.lst
title os-ohac-200906 kernel debugger
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/os-ohac-200906
```

```
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-
BOOTFS,console=text -k
module$ /platform/i86pc/$ISADIR/boot_archive
```

3.3.2 How to break into the kernel debugger

On a physical x86 system, the default key combination to break into the kernel debugger is the `F1-a`. This does not work when OpenSolaris is running as a VirtualBox guest. You can either change the default key abort sequence using the `kbd(1)` command, or use the following in order to send F1-a to a VirtualBox guest:

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage controlvm
<solarisVMname> keyboardputscancode 3b 1e 9e bb
```

3.3.3 Forcing a crash dump

Once you have entered the kernel debugger prompt, the following will cause a crash dump to be written to the dump device:

```
> $<systemdump
```

See `dumpadm(1M)` for details on how to configure a dump device and savecore directory.

After the system has rebooted, either the `svc:/system/dumpadm:default` service will automatically save the crash dump into the configured savecore directory, or you need to manually run `savecore(1M)`, if the dumpadm service is disabled.

If you want to save a crash dump of the live running OpenSolaris system without breaking into the kernel debugger or requiring a reboot, run within that system:

```
# savecore -L
```

If you want to force a crash dump before rebooting the system, run within that system:

```
# reboot -d
```

3.3.4 Crash dump analysis with Solaris CAT

While it is possible to perform analysis of crash dumps using mdb(1), the Solaris Crash Analysis Tool (CAT) comes with additional commands and macros, which are useful to get a quick overview of the crash cause.

Solaris CAT is available through http://blogs.sun.com/solariscat/, which contains the download link to the most current version.

After installation of the corresponding SUNWscat package you can read the documentation at file:///opt/SUNWscat/docs/index.html.

4 Open HA Cluster Configuration

4.1 Open HA Cluster Installation

Start both nodes. In case you don't want the console window open all time, start the VirtualBox guests by using the VRDP protocol.

The following ports got configured for the guests:

OS-b111b-OHAC-b10-1	3390
OS-b111b-OHAC-b10-2	3391

```
ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage startvm OS-
b111b-OHAC-b10-1 --type vrdp

ohacdemo@vorlon$ /opt/VirtualBox/VBoxManage startvm OS-
b111b-OHAC-b10-2 --type vrdp
```

The console can be reached via the the rdesktop application.

Console for os-ohac-1:

```
ohacdemo@vorlon$ rdesktop localhost:3390
```

Console for os-ohac-2:

```
ohacdemo@vorlon$ rdesktop localhost:3391
```

4.1.1 First node cluster installation (os-ohac-1)

In order to have a clean environment to fall back to, first create a new boot environment (BE), in which the Open HA Cluster packages get installed:

```
ohacdemo@os-ohac-1:~$ pfexec beadm create os-ohac-200906
ohacdemo@os-ohac-1:~$ pfexec beadm activate os-ohac-
200906
```

Verify if the entry in /rpool/boot/grub/menu.lst looks correct:

```
ohacdemo@os-ohac-1:~$ cat /rpool/boot/grub/menu.lst
```

Boot the newly created and activated BE:

```
ohacdemo@os-ohac-1:~$ pfexec init 6
```

If you want to install the self compiled packages, as explained in section 2.3, configure the publisher for your local IPS repository that contains the self compiled cluster packages:

```
ohacdemo@os-ohac-1:~$ pfexec pkg set-publisher -O
http://vorlon-int:7376/ ohacbuild
```

Instead, if you want to install the official Open HA Cluster 2009.06 release, you need to register and obtain the necessary SSL keys at https://pkg.sun.com/register, then follow the instructions. Configure the publisher like:

```
ohacdemo@os-ohac-1:~$ pfexec pkg set-publisher -k
/var/pkg/ssl/Open_HA_Cluster_2009.06.key.pem -c
/var/pkg/ssl/Open_HA_Cluster_2009.06.certificate.pem -O
https://pkg.sun.com/opensolaris/ha-cluster/ ha-cluster
```

Install the cluster packages:

```
ohacdemo@os-ohac-1:~$ pfexec pkg install ha-cluster-full
```

Add /usr/cluster/bin to $PATH and /usr/cluster/man to $MANPATH within $HOME/.profile for the main user ohacdemo and user root.

4.1.2 First node cluster configuration (os-ohac-1)

Allow RPC communication for external systems:

```
ohacdemo@os-ohac-1:~$ pfexec svccfg -s
svc:/network/rpc/bind setprop config/local_only = false
ohacdemo@os-ohac-1:~$ pfexec svcadm refresh
svc:/network/rpc/bind
```

Install the first cluster node:

- the cluster name is set to os-ohac-demo
- the lofi option is used for global devices
- the nodes os-ohac-1 and os-ohac-2 are part of the cluster
- the default IP subnet of 172.16.0.0 is getting used for the cluster interconnect. If you share the interconnect from multiple clusters on the same public IP subnet, you need to make sure to configure a unique IP subnet for each cluster.
- e1000g1 is the network interface used for the cluster interconnect, which is attached to the switch etherstub1

```
ohacdemo@os-ohac-1:~$ pfexec /usr/cluster/bin/scinstall \
    -i \
    -C os-ohac-demo \
    -F \
    -G lofi \
    -T node=os-ohac-1,node=os-ohac-2,authtype=sys \
```

```
    -w
netaddr=172.16.0.0,netmask=255.255.240.0,maxnodes=64,maxp
rivatenets=10,numvirtualclusters=12 \
    -A trtype=dlpi,name=e1000g1\
    -B type=switch,name=etherstub1 \
    -m endpoint=:e1000g1,endpoint=etherstub1
```

Reboot the node:

```
ohacdemo@os-ohac-1:~$ pfexec init 6
```

4.1.3 Second node cluster installation (os-ohac-2)

In order to have a clean environment to fall back, first create a new boot environment (BE), in which the Open HA Cluster packages get installed:

```
ohacdemo@os-ohac-2:~$ pfexec beadm create os-ohac-200906
ohacdemo@os-ohac-2:~$ pfexec beadm activate os-ohac-
200906
```

Verify if the entry in /rpool/boot/grub/menu.lst looks correct:

```
ohacdemo@os-ohac-2:~$ cat /rpool/boot/grub/menu.lst
```

Boot the newly created and activated BE:

```
ohacdemo@os-ohac-2:~$ pfexec init 6
```

If you want to install the self compiled packages, like explained in section 2.3, configure the publisher for your local IPS repository that contains the self compiled cluster packages:

```
ohacdemo@os-ohac-2:~$ pfexec pkg set-publisher -O
http://vorlon-int:7376/ ohacbuild
```

Instead, if you want to install the official Open HA Cluster 2009.06 release, you need to register and obtain the necessary SSL keys at https://pkg.sun.com/register, then follow the instructions. Configure the publisher like:

```
ohacdemo@os-ohac-2:~$ pfexec pkg set-publisher -k
/var/pkg/ssl/Open_HA_Cluster_2009.06.key.pem -c
/var/pkg/ssl/Open_HA_Cluster_2009.06.certificate.pem -O
https://pkg.sun.com/opensolaris/ha-cluster/ ha-cluster
```

Install the cluster packages:

```
ohacdemo@os-ohac-2:~$ pfexec pkg install ha-cluster-full
```

Add /usr/cluster/bin to $PATH and /usr/cluster/man to $MANPATH within $HOME/.profile for the main user and user root.

4.1.4 Second node cluster configuration (os-ohac-2)

Allow RPC communication for external systems:

```
ohacdemo@os-ohac-2:~$ pfexec svccfg -s
svc:/network/rpc/bind setprop config/local_only = false
ohacdemo@os-ohac-2:~$ pfexec svcadm refresh
svc:/network/rpc/bind
```

Add the second node to the cluster:

- the cluster name to join is os-ohac-demo
- the sponsoring node is os-ohac-1
- the lofi option is used for global devices
- e1000g1 is the network interface used for the cluster interconnect, which is attached to the switch etherstub1

```
ohacdemo@os-ohac-2:~$ pfexec /usr/cluster/bin/scinstall \
    -i \
    -C os-ohac-demo \
    -N os-ohac-1 \
    -G lofi \
    -A trtype=dlpi,name=e1000g1 \
    -m endpoint=:e1000g1,endpoint=etherstub1
```

Reboot the node:

```
ohacdemo@os-ohac-2:~$ pfexec init 6
```

4.2 Weak Membership Configuration

Weak membership allows a two-node cluster to be configured without requiring a quorum device.

When weak membership is enabled, in the event of a network partition caused by total failure of the cluster interconnect, which will lead to split brain, each cluster node will try to contact the configured ping target. In this example the ping target is set to 10.0.2.100. If the node can reach the ping target successfully, it will form its own single-node cluster.

Note that if both nodes can reach the ping target, both nodes will form their own separate single-node clusters, which will lead to both nodes taking over the services that are configured to run on them.

Special steps need to get followed in order to avoid data loss, before allowing the nodes to join one cluster again. Details are explained in the documentation

at http://opensolaris.org/os/community/ha-clusters/ohac/Documentation/OHACdocs/.

It can get enabled by the following commands:

```
ohacdemo@os-ohac-1:~$ pfexec clq set -p
multiple_partitions=true -p ping_targets=10.0.2.100
membership
This action might result in data corruption  or loss
Are you sure you want to enable multiple partitions in
the cluster to be operational (y/n) [n]?y
ohacdemo@os-ohac-1:~$ pfexec clq reset
ohacdemo@os-ohac-1:~$ clq status

Cluster Quorum ===

--- Quorum Votes Summary from (latest node
reconfiguration) ---

            Needed   Present   Possible
            ------   -------   --------
            1        2         2


--- Quorum Votes by Node (current status) ---

Node Name       Present       Possible       Status
---------       -------       --------       ------
os-ohac-1       1             1              Online
os-ohac-2       1             1              Online


--- Global Quorum Health Check (current status) ---

Node Name    Health Check Type    Entities      Status
---------    -----------------    --------      ------
os-ohac-1    Ping Targets         10.0.2.100    Ok

os-ohac-2    Ping Targets         10.0.2.100    Ok
```

As an alternative, you can configure a quorum device (either a quorum disk or a quorum server), in order to use strong membership. The procedure is explained at http://docs.sun.com/app/docs/doc/820-4677/cihecfab?l=en&a=view.

For the laptop configuration it would be possible to configure the quorum server on the host vorlon.

4.3 IPsec Configuration for the Cluster Interconnect

If the cluster interconnect is configured to use network interfaces on the public network, IPsec can be configured in order to protect the private TCP/IP traffic by

encrypting the IP packets. Note that the cluster heartbeat packets are send on the DLPI level lower than IP, which means they are not getting encrypted.

The following steps configure IPsec by using the Internet Key Exchange (IKE) method.

Prepare /etc/inet/ipsecinit.conf on both nodes:

```
both-nodes# cd /etc/inet
both-nodes# cp ipsecinit.sample ipsecinit.conf

os-ohac-1# ifconfig e1000g1
e1000g1:
flags=1008843<UP,BROADCAST,RUNNING,MULTICAST,PRIVATE,IPv4
> mtu 1500 index 4
        inet 172.16.0.129 netmask ffffff80 broadcast
172.16.0.255
        ether 2:8:20:0:aa:4d
os-ohac-1# ifconfig clprivnet0
clprivnet0:
flags=1009843<UP,BROADCAST,RUNNING,MULTICAST,MULTI_BCAST,
PRIVATE,IPv4> mtu 1500 index 5
        inet 172.16.4.1 netmask fffffe00 broadcast
172.16.5.255
        ether 0:0:0:0:0:1

os-ohac-2# ifconfig e1000g1
e1000g1:
flags=1008843<UP,BROADCAST,RUNNING,MULTICAST,PRIVATE,IPv4
> mtu 1500 index 4
        inet 172.16.0.130 netmask ffffff80 broadcast
172.16.0.255
        ether 2:8:20:f2:98:ad

os-ohac-2# ifconfig clprivnet0
clprivnet0:
flags=1009843<UP,BROADCAST,RUNNING,MULTICAST,MULTI_BCAST,
PRIVATE,IPv4> mtu 1500 index 5
        inet 172.16.4.2 netmask fffffe00 broadcast
172.16.5.255
        ether 0:0:0:0:0:2

os-ohac-1# vi ipsecinit.conf
{laddr 172.16.0.129 raddr 172.16.0.130} ipsec {auth_algs
any encr_algs any sa shared}
{laddr 172.16.4.1 raddr 172.16.4.2} ipsec {auth_algs any
encr_algs any sa shared}

os-ohac-2# vi ipsecinit.conf
{laddr 172.16.0.130 raddr 172.16.0.129} ipsec {auth_algs
any encr_algs any sa shared}
{laddr 172.16.4.2 raddr 172.16.4.1} ipsec {auth_algs any
encr_algs any sa shared}
```

Prepare /etc/inet/ike/config on both nodes:

```
both-nodes# cd /etc/inet/ike
both-nodes# cp config.sample config

os-ohac-1# vi config
{ label "clusternode1-priv-physical1-clusternode2-priv-
physical1"
  local_addr 172.16.0.129
  remote_addr 172.16.0.130
  p1_xform
  { auth_method preshared oakley_group 5 auth_alg md5
encr_alg 3des}
  p2_pfs 5
  p2_idletime_secs 30
}
{ label "clusternode1-priv-privnet0-clusternode2-priv-
privnet0"
  local_addr 172.16.4.1
  remote_addr 172.16.4.2
  p1_xform
  { auth_method preshared oakley_group 5 auth_alg md5
encr_alg 3des}
  p2_pfs 5
  p2_idletime_secs 30
}

os-ohac-2# vi config
{ label "clusternode2-priv-physical1-clusternode1-priv-
physical1"
  local_addr 172.16.0.130
  remote_addr 172.16.0.129
  p1_xform
  { auth_method preshared oakley_group 5 auth_alg md5
encr_alg 3des}
  p2_pfs 5
  p2_idletime_secs 30
}
{ label "clusternode2-priv-privnet0-clusternode1-priv-
privnet0"
  local_addr 172.16.4.2
  remote_addr 172.16.4.1
  p1_xform
  { auth_method preshared oakley_group 5 auth_alg md5
encr_alg 3des}
  p2_pfs 5
  p2_idletime_secs 30
}

both-nodes# /usr/lib/inet/in.iked -c -f
/etc/inet/ike/config
```

```
in.iked: Configuration file /etc/inet/ike/config
syntactically checks out.
```

Setup entries for pre-shared keys in /etc/inet/secret/ike.preshared on both nodes:

```
both-nodes# cd /etc/inet/secret

os-ohac-1# pktool genkey keystore=file outkey=ikekey
keytype=3des keylen=192 print=y
        Key Value
="329b7f792c5854dfd654674adf9220c45851dc61291c893b"

os-ohac-1# vi ike.preshared
{ localidtype IP
     localid 172.16.0.129
     remoteidtype IP
     remoteid 172.16.0.130
     key 329b7f792c5854dfd654674adf9220c45851dc61291c893b
}
{ localidtype IP
     localid 172.16.4.1
     remoteidtype IP
     remoteid 172.16.4.2
     key 329b7f792c5854dfd654674adf9220c45851dc61291c893b
}

os-ohac-2# vi ike.preshared
{ localidtype IP
     localid 172.16.0.130
     remoteidtype IP
     remoteid 172.16.0.129
     key 329b7f792c5854dfd654674adf9220c45851dc61291c893b
}
{ localidtype IP
     localid 172.16.4.2
     remoteidtype IP
     remoteid 172.16.4.1
     key 329b7f792c5854dfd654674adf9220c45851dc61291c893b
}

both-nodes# svcadm enable svc:/network/ipsec/ike:default
both-nodes# svcadm restart
svc:/network/ipsec/policy:default
```

4.4 COMSTAR iSCSI Configuraton

Open HA Cluster allows a shared nothing configuration to be setup by using the combination of COMSTAR/iSCSI and HA ZFS. If this is configured in combination with weak membership, it is important to **only** use the IP addresses configured on the clprivnet0 interface. This ensures that communication between

the nodes is only allowed when both nodes are part of the same cluster. Do **not** allow iSCSI traffic directly over the network interfaces used for the private interconnect (like e1000g1 in this example).

Install the iSCSI packages and reboot the nodes. This will then also import the corresponding SMF services:

```
both-nodes# pkg install SUNWiscsi SUNWiscsit SUNWstmf
both-nodes# init 6
```

Create one partition for the internal disk (assuming /dev/rdsk/c7d1 with 30GB size) and create a big slice for usage on both nodes:

```
both-nodes# fdisk c7d1p0
No fdisk table exists. The default partition for the disk
is:

  a 100% "SOLARIS System" partition

Type "y" to accept the default partition,  otherwise type
"n" to edit the partition table.
y

both-nodes# format c7d1
format> par
partition> pr
partition> 0
Enter partition id tag[unassigned]:
Enter partition permission flags[wm]:
Enter new starting cyl[0]: 2
Enter partition size[0b, 0c, 2e, 0.00mb, 0.00gb]: 3912c
partition> label
Ready to label disk, continue? y
partition> q
format> q
```

Configure the iSCSi target on both nodes:

```
both-nodes# svcadm disable
svc:/network/iscsi_initiator:default
both-nodes# svcadm enable stmf
both-nodes# svcadm enable target

os-ohac-1# itadm create-target
Target iqn.1986-03.com.sun:02:08adb6be-250f-49f9-cded-
885115ac5da9 successfully created

os-ohac-1# sbdadm create-lu /dev/rdsk/c7d1s0

Created the following LU:
```

```
              GUID                    DATA SIZE
SOURCE
--------------------------------  -------------------
----------------
600144f0944e440000004a1afe9b0001      32177229824
/dev/rdsk/c7d1s0

os-ohac-1# stmfadm add-view
600144f0944e440000004a1afe9b0001

os-ohac-2# itadm create-target
Target iqn.1986-03.com.sun:02:e946c38c-651f-ef9e-852a-
bb8615f4fe4a successfully created

os-ohac-2# sbdadm create-lu /dev/rdsk/c7d1s0

Created the following LU:

              GUID                    DATA SIZE
SOURCE
--------------------------------  -------------------
----------------
600144f02932490000004a1aff4c0001      32177229824
/dev/rdsk/c7d1s0

os-ohac-2# stmfadm add-view
600144f02932490000004a1aff4c0001
```

Configure iSCSI initiator on both nodes:

```
both-nodes# iscsiadm modify discovery -s enable

both-nodes# iscsiadm add static-config iqn.1986-
03.com.sun:02:08adb6be-250f-49f9-cded-
885115ac5da9,172.16.4.1

both-nodes# iscsiadm add static-config iqn.1986-
03.com.sun:02:e946c38c-651f-ef9e-852a-
bb8615f4fe4a,172.16.4.2
```

Make the devices known to the cluster on both nodes:

```
both-nodes# cldev refresh
both-nodes# cldev populate
```

Disable fencing for all devices from one node:

```
os-ohac-1# cluster set -p global_fencing=nofencing-
noscrub
```

4.5 HA ZFS Configuration

Create the mirrored zpool on one node:

```
ohacdemo@os-ohac-1:~$ cldev list -v
DID Device          Full Device Path
----------          ----------------
d1                  os-ohac-1:/dev/rdsk/c7d0
d2                  os-ohac-1:/dev/rdsk/c7d1
d3                  os-ohac-1:/dev/rdsk/c8t0d0
d4                  os-ohac-2:/dev/rdsk/c7d0
d5                  os-ohac-2:/dev/rdsk/c7d1
d6                  os-ohac-2:/dev/rdsk/c8t0d0
d7                  os-ohac-
1:/dev/rdsk/c0t600144F02932490000004A1AFF4C0001d0
d7                  os-ohac-
2:/dev/rdsk/c0t600144F02932490000004A1AFF4C0001d0
d8                  os-ohac-
1:/dev/rdsk/c0t600144F0944E440000004A1AFE9B0001d0
d8                  os-ohac-
2:/dev/rdsk/c0t600144F0944E440000004A1AFE9B0001d0

ohacdemo@os-ohac-1:~$ pfexec zpool create services
mirror /dev/did/dsk/d7s2 /dev/did/dsk/d8s2

ohacdemo@os-ohac-1:~$ pfexec zpool export services
```

Register the SUNW.gds and SUNW.HAStoragePlus resource type and create resource group service-rg, resource services-pool-rs for the zpool and resource services-lh-rs for the logical host on one node:

```
ohacdemo@os-ohac-1:~$ pfexec clrg create services-rg
ohacdemo@os-ohac-1:~$ pfexec clrt register
SUNW.HAStoragePlus
ohacdemo@os-ohac-1:~$ pfexec clrt register SUNW.gds
ohacdemo@os-ohac-1:~$ pfexec clrs create -g services-rg
-t HAStoragePlus -p Zpools=services services-pool-rs
ohacdemo@os-ohac-1:~$ pfexec clrslh create -g services-rg
-h os-ohac-lh1 services-lh-rs
ohacdemo@os-ohac-1:~$ pfexec clrg online -eM services-rg
```

4.6 HA MySQL Configuration

Install the MySQL 5.1 packages on both nodes:

```
both-nodes# pkg install SUNWmysql51
```

Configure MySQL on the node where the services-rg resource group is online:

```
ohacdemo@os-ohac-1:~$ clrg status services-rg

=== Cluster Resource Groups ===

Group Name        Node Name      Suspended      Status
----------        ---------      ---------      ------
services-rg       os-ohac-1      No             Online
                  os-ohac-2      No             Offline

os-ohac-1# zfs create services/mysql
os-ohac-1# mkdir -p /services/mysql/logs
os-ohac-1# mkdir -p /services/mysql/innodb
os-ohac-1# cp /etc/mysql/5.1/my.cnf
/services/mysql/my.cnf
os-ohac-1# vi /services/mysql/my.cnf
--- /etc/mysql/5.1/my.cnf    2009-05-27
15:03:50.591318099 +0200
+++ /services/mysql/my.cnf   2009-05-27
15:41:53.354358171 +0200
@@ -18,14 +18,14 @@
 [client]
 #password = your_password
 port      = 3306
-socket          = /tmp/mysql.sock
+socket          = /tmp/os-ohac-lh1.sock

 # Here follows entries for some specific programs

 # The MySQL server
 [mysqld]
 port      = 3306
-socket          = /tmp/mysql.sock
+socket          = /tmp/os-ohac-lh1.sock
 skip-locking
 key_buffer = 16K
 max_allowed_packet = 1M
@@ -45,6 +45,8 @@
 #skip-networking
 server-id = 1

+bind-address=os-ohac-lh1
+
 # Uncomment the following if you want to log updates
 #log-bin=mysql-bin

@@ -52,19 +54,19 @@
 #binlog_format=mixed

 # Uncomment the following if you are using InnoDB tables
-#innodb_data_home_dir = /var/mysql/5.1/data/
-#innodb_data_file_path = ibdata1:10M:autoextend
-#innodb_log_group_home_dir = /var/mysql/5.1/data/
-#innodb_log_arch_dir = /var/mysql/5.1/data/
```

```
+innodb_data_home_dir = /services/mysql/innodb
+innodb_data_file_path = ibdata1:10M:autoextend
+innodb_log_group_home_dir = /services/mysql/innodb
+#innodb_log_arch_dir = /services/mysql/innodb
 # You can set .._buffer_pool_size up to 50 - 80 %
 # of RAM but beware of setting memory usage too high
-#innodb_buffer_pool_size = 16M
-#innodb_additional_mem_pool_size = 2M
+innodb_buffer_pool_size = 16M
+innodb_additional_mem_pool_size = 2M
 # Set .._log_file_size to 25 % of buffer pool size
-#innodb_log_file_size = 5M
-#innodb_log_buffer_size = 8M
-#innodb_flush_log_at_trx_commit = 1
-#innodb_lock_wait_timeout = 50
+innodb_log_file_size = 5M
+innodb_log_buffer_size = 8M
+innodb_flush_log_at_trx_commit = 1
+innodb_lock_wait_timeout = 50

 [mysqldump]
 quick

both-nodes# cd /etc/mysql/5.1
both-nodes# mv my.cnf my.cnf.orig
both-nodes# ln -s /services/mysql/my.cnf .

os-ohac-1# /usr/mysql/bin/mysql_install_db
--datadir=/services/mysql
Installing MySQL system tables...
090527 15:41:56 [Warning] option 'thread_stack': unsigned
value 65536 adjusted to 131072
090527 15:41:56 [Warning] option 'thread_stack': unsigned
value 65536 adjusted to 131072
OK
Filling help tables...
090527 15:41:57 [Warning] option 'thread_stack': unsigned
value 65536 adjusted to 131072
090527 15:41:57 [Warning] option 'thread_stack': unsigned
value 65536 adjusted to 131072
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your
system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root
USER !
To do so, start the server, then issue the following
commands:

/usr/mysql/5.1/bin/mysqladmin -u root password 'new-
password'
```

```
/usr/mysql/5.1/bin/mysqladmin -u root -h os-ohac-1
password 'new-password'

Alternatively you can run:
/usr/mysql/5.1/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default.  This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr/mysql/5.1 ; /usr/mysql/5.1/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd /usr/mysql/5.1/mysql-test ; perl mysql-test-run.pl

Please report any problems with the
/usr/mysql/5.1/bin/mysqlbug script!

The latest information about MySQL is available at
http://www.mysql.com/
Support MySQL by buying support/licenses from
http://shop.mysql.com/

os-ohac-1# chown -R mysql:mysql /services/mysql
```

Manually test the MySQL configuration:

```
os-ohac-1# /usr/mysql/bin/mysqld --defaults-
file=/services/mysql/my.cnf --basedir=/usr/mysql
--datadir=/services/mysql --user=mysql --pid-
file=/services/mysql/mysqld.pid &
os-ohac-1# /usr/mysql/bin/mysql -S /tmp/os-ohac-lh1.sock
-uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.30 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> exit;
Bye
```

Configure the MySQL admin password for the admin user:

```
os-ohac-1# /usr/mysql/bin/mysqladmin -S /tmp/os-ohac-
lh1.sock  password 'mysqladmin'
```

Allow access to the database for both cluster nodes for user root:

```
os-ohac-1# /usr/mysql/bin/mysql -S /tmp/os-ohac-lh1.sock
-uroot -p'mysqladmin'
Your MySQL connection id is 3
Server version: 5.1.30 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> use mysql;
Database changed
mysql> GRANT ALL ON *.* TO 'root'@'os-ohac-1' IDENTIFIED
BY 'mysqladmin';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL ON *.* TO 'root'@'os-ohac-2' IDENTIFIED
BY 'mysqladmin';
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE user SET Grant_priv='Y' WHERE User='root'
AND Host='os-ohac-1';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE user SET Grant_priv='Y' WHERE User='root'
AND Host='os-ohac-2';
Query OK, 0 rows affected (0.01 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> exit;
Bye
```

Create and setup the HA MySQL resource configuration file:

```
os-ohac-1# mkdir /services/mysql/cluster-config
os-ohac-1# cd /services/mysql/cluster-config
os-ohac-1# cp /opt/SUNWscmys/util/ha_mysql_config .
os-ohac-1# cp /opt/SUNWscmys/util/mysql_config .

os-ohac-1# vi ha_mysql_config
RS=mysql-rs
RG=services-rg
PORT=
LH=services-lh-rs
HAS_RS=services-pool-rs

ZONE=
ZONE_BT=
PROJECT=

BASEDIR=/usr/mysql
DATADIR=/services/mysql
```

```
MYSQLUSER=mysql
MYSQLHOST=os-ohac-lh1
FMUSER=fmuser
FMPASS=fmuser
LOGDIR=/services/mysql/logs
CHECK=YES

os-ohac-1# vi mysql_config
MYSQL_BASE=/usr/mysql
MYSQL_USER=root
MYSQL_PASSWD=mysqladmin
MYSQL_HOST=os-ohac-lh1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCK=/tmp/os-ohac-lh1.sock
MYSQL_NIC_HOSTNAME="os-ohac-1 os-ohac-2"
MYSQL_DATADIR=/services/mysql

os-ohac-1# /opt/SUNWscmys/util/mysql_register -f
/services/mysql/cluster-config/mysql_config
sourcing /services/mysql/cluster-config/mysql_config and
create a working copy under
/opt/SUNWscmys/util/mysql_config.work

MySQL version 5 detected on 5.11


Check if the MySQL server is running and accepting
connections


Add faulmonitor user (fmuser) with password (fmuser) with
Process-,Select-, Reload- and Shutdown-privileges to user
table for mysql database for host os-ohac-1


Add SUPER privilege for fmuser@os-ohac-1


Add faulmonitor user (fmuser) with password (fmuser) with
Process-,Select-, Reload- and Shutdown-privileges to user
table for mysql database for host os-ohac-2


Add SUPER privilege for fmuser@os-ohac-2


Create test-database sc3_test_database


Grant all privileges to sc3_test_database for
faultmonitor-user fmuser for host os-ohac-1
```

```
Grant all privileges to sc3_test_database for
faultmonitor-user fmuser for host os-ohac-2


Flush all privileges


Mysql configuration for HA is done

os-ohac-1# kill -TERM `cat /services/mysql/mysqld.pid`

os-ohac-1# /opt/SUNWscmys/util/ha_mysql_register -f
/services/mysql/cluster-config/ha_mysql_config
sourcing /services/mysql/cluster-config/ha_mysql_config
and create a working copy under
/opt/SUNWscmys/util/ha_mysql_config.work
Registration of resource mysql-rs succeeded.
 remove the working copy
/opt/SUNWscmys/util/ha_mysql_config.work
os-ohac-1# clrs enable mysql-rs
```

Verify that the services-rg works on both nodes:

```
os-ohac-1# clrs status mysql-rs

=== Cluster Resources ===

Resource Name       Node Name      State        Status
Message
-------------       ---------      -----
--------------
mysql-rs            os-ohac-1      Online       Online -
Service is online.
                    os-ohac-2      Offline      Offline

os-ohac-1# clrg switch -n os-ohac-2 services-rg
os-ohac-1# clrs status mysql-rs

=== Cluster Resources ===

Resource Name       Node Name      State        Status
Message
-------------       ---------      -----
--------------
mysql-rs            os-ohac-1      Offline      Offline
                    os-ohac-2      Online       Online
```

4.7 HA Tomcat Configuration

Install the tomcat package on both nodes:

```
both-nodes# pkg install SUNWtcat
```

Configure Tomcat on the node where the services-rg resource group is online:

```
ohacdemo@os-ohac-1:~$ clrg status services-rg

=== Cluster Resource Groups ===

Group Name        Node Name      Suspended      Status
----------        ---------      ---------      ------
services-rg       os-ohac-1      No             Online
                  os-ohac-2      No             Offline

os-ohac-1# zfs create services/tomcat
os-ohac-1# vi /services/tomcat/env.ksh
#!/bin/ksh
CATALINA_HOME=/usr/tomcat6
CATALINA_BASE=/services/tomcat
JAVA_HOME=/usr/java
export CATALINA_HOME CATALINA_BASE JAVA_HOME

os-ohac-1# chown webservd:webservd
/services/tomcat/env.ksh

os-ohac-1# cd /var/tomcat6
os-ohac-1# tar cpf - . | ( cd /services/tomcat ; tar xpf
-)

os-ohac-1# cd /services/tomcat
os-ohac-1# mkdir cluster-config
os-ohac-1# chown webservd:webservd cluster-config
os-ohac-1# cd cluster-config
os-ohac-1# cp /opt/SUNWsctomcat/util/sctomcat_config .
os-ohac-1# cp /opt/SUNWsctomcat/bin/pfile .
os-ohac-1# chown webservd:webservd pfile
os-ohac-1# vi pfile
EnvScript=/services/tomcat/env.ksh
User=webservd
Basepath=/usr/tomcat6
Host=os-ohac-lh1
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20
os-ohac-1# vi sctomcat_config
RS=tomcat-rs
RG=services-rg
PORT=8080
```

```
LH=services-lh-rs
NETWORK=true
SCALABLE=false
PFILE=/services/tomcat/cluster-config/pfile
HAS_RS=services-pool-rs
ZONE=
ZONE_BT=
PROJECT=

os-ohac-1# /opt/SUNWsctomcat/util/sctomcat_register -f
/services/tomcat/cluster-config/sctomcat_config
sourcing /services/tomcat/cluster-config/sctomcat_config
and create a working copy under
/opt/SUNWsctomcat/util/sctomcat_config.work
Registration of resource tomcat-rs succeeded.
 remove the working copy
/opt/SUNWsctomcat/util/sctomcat_config.work

os-ohac-1# clrs enable tomcat-rs
```

Verify that the services-rg works on both nodes:

```
os-ohac-1# clrs status tomcat-rs

=== Cluster Resources ===

Resource Name       Node Name      State        Status
Message
-------------       ---------      -----
--------------
tomcat-rs           os-ohac-1      Online       Online -
Service is online.
                    os-ohac-2      Offline      Offline

os-ohac-1# clrg switch -n os-ohac-2 services-rg
os-ohac-1# clrs status tomcat-rs

=== Cluster Resources ===

Resource Name       Node Name      State        Status
Message
-------------       ---------      -----
--------------
tomcat-rs           os-ohac-1      Offline      Offline
                    os-ohac-2      Online       Online
```

Start firefox on vorlon and verify the tomcat page at http://os-ohac-lh1:8080/.

4.8 Scalable Apache Configuration

Create failover resource group for the shared address:

```
os-ohac-1# clrg create shared-ip-rg
os-ohac-1# clrssa create -g shared-ip-rg -h os-ohac-lh2
shared-ip-rs
os-ohac-1# clrg online -eM shared-ip-rg
```

Prepare the apache configuration file:

```
both-nodes# cd /etc/apache2/2.2
both-nodes# cp -p httpd.conf httpd.conf.orig
both-nodes# vi httpd.conf
     => change the ServerName entry like:
--- httpd.conf.orig     2009-05-19 18:29:05.182650000
+0200
+++ httpd.conf  2009-05-26 15:45:48.559087652 +0200
@@ -103,7 +103,8 @@
 #
 # If your host doesn't have a registered DNS name, enter
its IP address here.
 #
-ServerName 127.0.0.1
+#ServerName 127.0.0.1
+ServerName 10.0.2.111

 #
 # DocumentRoot: The directory out of which you will
serve your
```

The default httpd.conf file uses /var/apache/2.2/htdocs as DocumentRoot.

Configure the scalable resource group for apache:

```
os-ohac-1# clrt register SUNW.apache
os-ohac-1# clrg create -p Maximum_primaries=2 -p
Desired_primaries=2 -p RG_dependencies=shared-ip-rg
apache-rg
os-ohac-1# clrs create -g apache-rg -t SUNW.apache -p
Bin_dir=/usr/apache2/2.2/bin -p
Resource_dependencies=shared-ip-rs -p Scalable=True -p
Port_list=80/tcp apache-rs
os-ohac-1# clrg online -eM apache-rg
```

A References

1. HA Clusters Community page:
   http://opensolaris.org/os/community/ha-clusters/
2. Colorado Project page:
   http://opensolaris.org/os/project/colorado/
3. Open HA Cluster 2009.06 Release Notes:
   http://opensolaris.org/os/community/ha-clusters/ohac/Documentation/OHACdocs/relnotes/
4. Compiling Open HA Cluster core framework on OpenSolaris:
   http://wiki.genunix.org/wiki/index.php/Compiling_OHAC
5. Compiling Open HA Cluster data services on OpenSolaris:
   http://wiki.genunix.org/wiki/index.php/Compiling_OHAC_Agents
6. VirtualBox Download Page:
   http://www.virtualbox.org/wiki/Downloads
7. Solaris Cluster documentation:
   http://docs.sun.com/app/docs/prod/sun.cluster32#hic
8. Solaris Cluster Blog:
   http://blogs.sun.com/SC
9. OpenSolaris 2009.06 documentation:
   http://dlc.sun.com/osol/docs/content/2009.06/getstart/
10. Solaris OS Hardware Compatibility Lists:
    http://www.sun.com/bigadmin/hcl/
11. Toshiba OpenSolaris Laptops:
    http://www.opensolaris.com/toshibanotebook/index.html