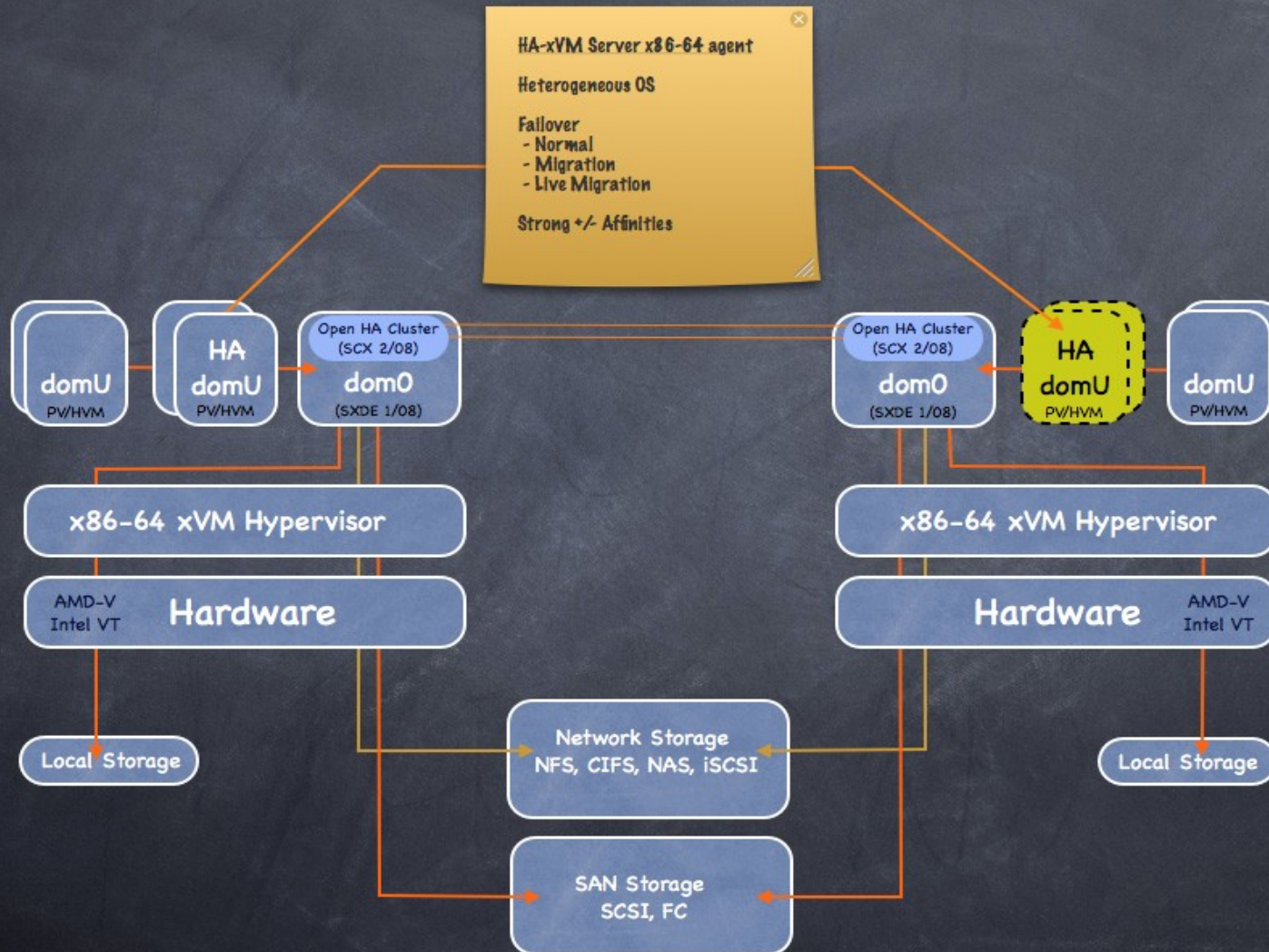


HA-xVM Server x86-64 agent cheat sheet

HA-xVM Server x86-64 Overview



HA-xVM Server x86-64 agent cheat sheet

Environment

The purpose of this cheat sheet is to demonstrate how Open HA Cluster (OHAC) can be used to mitigate against a physical server failure, where such a server would be running several xVM Server x86-64 guest domains. In particular we will deploy the following,

- Install paravirtualised (PV) OpenSolaris and CentOS guest domains.
- Use shared storage and network storage for guest domains.
- Manage the PV guest domains using the HA-xVM Server x86-64 agent.
- Failover guest domains in planned and unplanned scenarios.
- Observe the effect of a strong negative affinity.
- Clone the CentOS guest domain to create another guest domain.

For shared storage, we will deploy a cluster file system, physical DID device and ZFS volume.

For network storage, we will deploy a ZFS/NFS server from another cluster. Please note that we will not use the ZFS `sharenfs=on` property for our ZFS/NFS service, refer to http://blogs.sun.com/SC/entry/why_sunw_nfs_is_required as to why. If you decide NOT to use another cluster to host the ZFS/NFS service and instead want to use a node outside of OHAC then the ZFS `sharenfs=on` property can be used.

Within this cheat sheet the following clusters are used,

OHAC/xVM x86-64 servers – 3 nodes (podio1, podio2 & podio3), 8GB memory

SC 3.2/ZFS SPARC servers – 2 nodes (pmutt1 & pmutt2), 8GB memory

Caveats

Within this cheat sheet simple configurations are shown which do not represent either best practice or address performance issues, especially when configuring various disk storage configurations. Instead the cheat sheet will hopefully allow you to consider what is possible thereby allowing to explore whatever configuration suits your environment.

Known issues

Bug ID: 6579356 – stopper: `rpc.metad` is dumping core when adding disk to set

Bug ID: 6671904 – Can't NFS share a ZFS file system with SCX 2/08 using `HASStoragePlus`

HA-xVM Server x86-64 agent cheat sheet

Contents

1. Install SXDE 1/08 and SCX 2/08 on each OHAC/xVM node.
2. Boot Solaris xVM together with Open HA Cluster on each OHAC/xVM node only.
3. Install Solaris 10 08/07 and SC 3.2 on each SC 3.2/ZFS node.
4. Shared storage between OHAC/xVM nodes.
 - 4 (a). Install domu1 onto a highly available local file system.
 - 4 (b). Convert the highly available local file system to a cluster file system.
 - 4 (c). Install domu2 onto a physical DID partition.
 - 4 (d). Install domu3 onto a ZFS volume.
5. Configure each OHAC/xVM node for live migration.
 - 5 (a). Live migrate domu1 between all three nodes.
 - 5 (b). Live migrate domu2 between all three nodes.
6. Verify that each domu can reach milestone multi-user-server.
7. Install the HA-xVM Server x86-64 agent.
8. Setup a public key ssh authentication between Domain-0 and each domu.
 - 8 (a). Verify a public key ssh authentication between Domain-0 and each domu.
9. Setup the HA-xVM Server x86-64 agent administrative file system.
10. Place domain domu1 under the control of the HA-xVM Server x86-64 agent.
 - 10 (a). Verify that the OHAC resource is online.
 - 10 (b). Disable/enable the OHAC resource.
 - 10 (c). Switch the OHAC resource to the other node.
11. Fault injection test – Crash the node.
 - 11 (a). Fault injection test – Evacuate the node.
 - 11 (c). Fault injection test – destroy the domain.
12. Place domu2 under the control of the HA-xVM Server x86-64 agent.
13. Place domu3 under the control of the HA-xVM Server x86-64 agent.
14. All shared storage domains under HA-xVM Server x86-64 agent control.
15. Assign a strong negative affinity against domu3.
16. Switch domu1 to the node where domu3 is running.
 - 17 (a). Create a ZFS zpool and share using HA-NFS.
 - 17 (b). On each OHAC/xVM x86-64 node, mount the ZFS/NFS file systems.
18. Install domu4 on the NFS filesystem.
19. Setup a public key ssh authentication between Domain-0 and domu4.
 - 19 (a). Verify a public key ssh authentication between Domain-0 and domu4.
20. Place domain domu4 under the control of the HA-xVM Server x86-64 agent.
21. Live migrate domu4 to node3.
 - 22 (a). Clone domain domu4 to create domu4C.
 - 22 (b). Copy and create/amend the XML file for cloned domain domu4C.
23. Clone domu4's ZFS file systems for the cloned domain domu4C.
 - 23 (b). Mount the cloned domain's ZFS file system.
24. Manually start and configure domu4C.
25. Place domu4 under the control of the HA-xVM Server x86-64 agent.
26. Ensure all OHAC resources are online.
27. Turning debug on for the HA-xVM Server x86-64 agent.

HA-xVM Server x86-64 agent cheat sheet

1. Install SXDE 1/08 and SCX 2/08 on each OHAC/xVM node.

Refer to the following url for detailed documentation on how to install OHAC.

<http://opensolaris.org/os/community/ha-clusters/ohac/Documentation/SCXdocs/SCX/>

2. Boot Solaris xVM together with Open HA Cluster on each OHAC/xVM node only.

On each OHAC/xVM node, amend /boot/grub/menu.lst to boot option 1 (Solaris xVM) and cap Domain-0's memory.

Refer to the following url for any relevant release notes,

<http://www.opensolaris.org/os/community/xen/docs/relnotes/>

```
#----- ADDED BY BOOTADM - DO NOT EDIT -----  
title Solaris xVM  
kernel$ /boot/$ISADIR/xen.gz dom0_mem=2048M  
module$ /platform/i86xpv/kernel/$ISADIR/unix /platform/i86xpv/kernel/$ISADIR/unix  
module$ /platform/i86pc/$ISADIR/boot_archive
```

Note: before booting each node disable cde-login.

```
bash-3.2# svcadm disable cde-login
```

3. Install Solaris 10 08/07 and SC 3.2 on each SC 3.2/ZFS node.

Refer to the following urls for download links to Solaris 10 and SC 3.2

S10 u4 - <http://www.sun.com/software/solaris/get.jsp>

SC 3.2 - <http://www.sun.com/download/products.xml?id=4581ab9e>

SC 3.2 installation docs - <http://docs.sun.com/app/docs/doc/819-2970>

HA-xVM Server x86-64 agent cheat sheet

4. Shared storage between OHAC/xVM nodes.

The following shared storage is available across our three cluster nodes.

```
bash-3.2# cat /etc/release
    Solaris Express Developer Edition 1/08 snv_79b X86
    Copyright 2008 Sun Microsystems, Inc. All Rights Reserved.
    Use is subject to license terms.
    Assembled 24 January 2008
bash-3.2# cat /etc/cluster/release
    Sun Cluster 3.2u1 for Solaris 11 i386
    Copyright 2008 Sun Microsystems, Inc. All Rights Reserved.
bash-3.2#
bash-3.2# cldevice list -v d5 d6 d7
DID Device      Full Device Path
-----
d5              podio2:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Ad0
d5              podio1:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Ad0
d5              podio3:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Ad0
d6              podio2:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Bd0
d6              podio1:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Bd0
d6              podio3:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Bd0
d7              podio2:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Cd0
d7              podio1:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Cd0
d7              podio3:/dev/rdisk/c0t600C0FF000000000008C2D1A3FC5600Cd0
bash-3.2#
```

We will deploy these DID devices as follows,

d5s0 – Highly available local file system to be used as domu1's (vzodio1a) system image. Afterwards we will convert the highly available local file system to a cluster file system.

d6s0 – Physical partition to be used as domu2's (vzodio1b) system image.

d7s0 – ZFS volume to be used as domu3's (vzodio1b) system image.

Within the examples that follow, we have created 20gb slices for d5s0, d6s0 and d7s0 and created a SVM metadvice state database on all nodes. Finally, please note the caveat on the first page of this cheat sheet.

```
bash-3.2# metadb -a -c3 -f c4t0d0s7
```

HA-xVM Server x86-64 agent cheat sheet

4 (a). Install domu1 onto a highly available local file system.

Note: Entries marked in blue need to be executed on all cluster nodes.

```
bash-3.2# uname -n
podio1
bash-3.2# metaset -s dg_d1 -a -h podio1 podio2 podio3
bash-3.2# metaset -s dg_d1 -t
bash-3.2# metaset -s dg_d1 -a /dev/did/rdsk/d5
bash-3.2# cat >> /etc/lvm/md.tab <<-END
dg_d1/d100      -m      dg_d1/d110
dg_d1/d110      1 1      /dev/did/rdsk/d5s0
END
bash-3.2# metainit -s dg_d1 -a
bash-3.2# newfs /dev/md/dg_d1/rdsk/d100
bash-3.2#
bash-3.2# mkdir -p /global/bootdisk
bash-3.2#
bash-3.2# cat >> /etc/vfstab <<-END
/dev/md/dg_d1/dsk/d100 /dev/md/dg_d1/rdsk/d100 /global/bootdisk ufs 3 no logging
END
bash-3.2# metaset -s dg_d1 -t
bash-3.2# mount /global/bootdisk
bash-3.2# virt-install -n domu1 --paravirt -r 1024 --nographics -f /global/bootdisk/root.img -s 16 -l /net/10.4.195.10/export/nv/x_iso/78/78.iso
```

```
Starting install...
Creating storage file... 100% |=====| 16 GB    00:00
Creating domain...                                0 B 00:10
v3.0.4-1-xvm chgset 'Sun Jan 06 11:43:59 2008 -0800 13229:fc49a937b8f7'
SunOS Release 5.11 Version snv_78 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Configuring /dev
Solaris Interactive Text (Console session)
Using install cd in /dev/dsk/c0d1p0
Using RPC Bootparams for network configuration information.
Attempting to configure interface xnf0...
Skipped interface xnf0
Reading ZFS config: done.
Setting up Java. Please wait...
Beginning system identification...
Searching for configuration file(s)...
Search complete.
Discovering additional network configuration...
```

<... snipped ...>

HA-xVM Server x86-64 agent cheat sheet

4 (b). Convert the highly available local file system to a cluster file system.

Note: Entries marked in blue need to be executed on all cluster nodes.

Once domu1 has been installed we need to stop the domain and change the highly available local file system to a cluster file system.

```
bash-3.2# uname -n
podio1
bash-3.2# virsh list --all
  Id Name                State
-----
   0 Domain-0            running
   2 domu1                blocked

bash-3.2# xm shutdown -w domu1
Domain domu1 terminated
All domains terminated
bash-3.2# virsh list --all
  Id Name                State
-----
   0 Domain-0            running
  - domu1                shut off

bash-3.2# umount /global/bootdisk
bash-3.2# vi /etc/vfstab
change
/dev/md/dg_d1/dsk/d100 /dev/md/dg_d1/rdisk/d100 /global/bootdisk ufs 3 no logging
to
/dev/md/dg_d1/dsk/d100 /dev/md/dg_d1/rdisk/d100 /global/bootdisk ufs 3 yes global,logging
bash-3.2#
bash-3.2# mount /global/bootdisk
bash-3.2# virsh start domu1
Domain domu1 started

bash-3.2# virsh list --all
  Id Name                State
-----
   0 Domain-0            running
   3 domu1                blocked

bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

4 (c). Install domu2 onto a physical DID partition.

Note, we cannot create a second SVM diskset for /dev/did/rdisk/d6 because of Bug ID: 6579356.

```
bash-3.2# uname -n
podio2
bash-3.2# virt-install -n domu2 --paravirt -r 1024 --nographics -f /dev/did/dsk/d6s0 -s 16 -l /net/10.4.195.10/export/nv/x_iso/78/78.iso
```

```
Starting install...
Creating domain...                               0 B 00:26
v3.0.4-1-xvm chgset 'Sun Jan 06 11:43:59 2008 -0800 13229:fc49a937b8f7'
SunOS Release 5.11 Version snv_78 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Configuring /dev
Solaris Interactive Text (Console session)
Using install cd in /dev/dsk/c0d1p0
Using RPC Bootparams for network configuration information.
Attempting to configure interface xnf0...
Skipped interface xnf0
Reading ZFS config: done.
Setting up Java. Please wait...
Beginning system identification...
Searching for configuration file(s)...
Search complete.
Discovering additional network configuration...

<... snipped ...>
```

HA-xVM Server x86-64 agent cheat sheet

4 (d). Install domu3 onto a ZFS volume.

```
bash-3.2# uname -n
podio3
bash-3.2# zpool create zfs_bootdisk /dev/did/dsk/d7s0
bash-3.2#
bash-3.2# virt-install -n domu3 --paravirt -r 1024 --nographics -f /zfs_bootdisk/root.img -s 16 -l /net/10.4.195.10/export/nv/x_iso/78/78.iso
```

```
Starting install...
Creating storage file... 100% |=====| 16 GB 00:00
Creating domain... 0 B 00:05
v3.0.4-1-xvm chgset 'Sun Jan 06 11:43:59 2008 -0800 13229:fc49a937b8f7'
SunOS Release 5.11 Version snv_78 64-bit
Copyright 1983-2007 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Configuring /dev
Solaris Interactive Text (Console session)
Using install cd in /dev/dsk/c0d1p0
Using RPC Bootparams for network configuration information.
Attempting to configure interface xnf0...
Skipped interface xnf0
Reading ZFS config: done.
Setting up Java. Please wait...
Beginning system identification...
Searching for configuration file(s)...
Search complete.
Discovering additional network configuration...

<... snipped ...>
```

HA-xVM Server x86-64 agent cheat sheet

5. Configure each OHAC/xVM node for live migration.

Perform this part on any cluster node.

```
bash-3.2# /usr/cluster/bin/scha_cluster_get -O PRIVATELINK_HOSTNAME_NODE podio1
clusternode1-priv
bash-3.2# /usr/cluster/bin/scha_cluster_get -O PRIVATELINK_HOSTNAME_NODE podio2
clusternode2-priv
bash-3.2# /usr/cluster/bin/scha_cluster_get -O PRIVATELINK_HOSTNAME_NODE podio3
clusternode3-priv
bash-3.2#
```

Perform this part on the first node [podio1].

```
bash-3.2# uname -n
podio1
bash-3.2# svccfg -s xend setprop config/xend-relocation-address = "\"\"
bash-3.2# svccfg -s xend setprop config/xend-relocation-hosts-allow = "\"^clusternode2-priv$ ^clusternode3-priv\""
bash-3.2# svcadm refresh xend && svcadm restart xend
bash-3.2#
```

Perform this part on the second node [podio2].

```
bash-3.2# uname -n
podio2
bash-3.2# svccfg -s xend setprop config/xend-relocation-address = "\"\"
bash-3.2# svccfg -s xend setprop config/xend-relocation-hosts-allow = "\"^clusternode1-priv$ ^clusternode3-priv\""
bash-3.2# svcadm refresh xend && svcadm restart xend
bash-3.2#
```

Perform this part on the third node [podio3].

```
bash-3.2# uname -n
podio3
bash-3.2# svccfg -s xend setprop config/xend-relocation-address = "\"\"
bash-3.2# svccfg -s xend setprop config/xend-relocation-hosts-allow = "\"^clusternode1-priv$ ^clusternode2-priv\""
bash-3.2# svcadm refresh xend && svcadm restart xend
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

5 (a). Live migrate domu1 between all three nodes.

Perform this step on the cluster node where domu1 is running. Note only domu1 and domu2 can be live migrated.

```
bash-3.2# uname -n
```

```
podio1
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
3	domu1	blocked

```
bash-3.2# xm migrate --live domu1 clusternode2-priv
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
-	domu1	shut off

```
bash-3.2#
```

```
bash-3.2# uname -n
```

```
podio2
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
1	domu2	blocked
2	domu1	blocked

```
bash-3.2# xm migrate --live domu1 clusternode3-priv
```

```
bash-3.2#
```

```
bash-3.2# uname -n
```

```
podio3
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
1	domu3	blocked
2	domu1	blocked

```
bash-3.2# xm migrate --live domu1 clusternode1-priv
```

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

5 (b). Live migrate domu2 between all three nodes.

Perform this step on the cluster node where domu2 is running. Note only domu1 and domu2 can be live migrated.

```
bash-3.2# uname -n
```

```
podio2
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
1	domu2	blocked

```
bash-3.2# xm migrate --live domu2 clusternode3-priv
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
-	domu2	shut off

```
bash-3.2#
```

```
bash-3.2# uname -n
```

```
podio3
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
1	domu3	blocked
3	domu2	blocked

```
bash-3.2# xm migrate --live domu2 clusternode1-priv
```

```
bash-3.2#
```

```
bash-3.2# uname -n
```

```
podio1
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
4	domu1	blocked
5	domu2	blocked

```
bash-3.2# xm migrate --live domu2 clusternode2-priv
```

HA-xVM Server x86-64 agent cheat sheet

6. Verify that each domu can reach milestone multi-user-server.

Before we place the domus under the control of HA-xVM Server x86-64 agent, ensure that each domu can boot successfully to the milestone multi-user-server. This milestone state is required by the agent which probes the domu to determine it has successfully booted.

```
bash-3.2# rlogin vzodio1a -l root
Password:
Last login: Thu Mar 13 11:50:39 from vpn-129-150-118
Sun Microsystems Inc. SunOS 5.11 snv_78      October 2007
# svcs multi-user-server
STATE      STIME      FMRI
online     10:54:05  svc:/milestone/multi-user-server:default
# exit
Connection to vzodio1a closed.
bash-3.2#
bash-3.2# rlogin vzodio1b -l root
Password:
Last login: Thu Mar 13 12:13:10 from vpn-129-150-118
Sun Microsystems Inc. SunOS 5.11 snv_78      October 2007
# svcs multi-user-server
STATE      STIME      FMRI
online     12:11:47  svc:/milestone/multi-user-server:default
# exit
Connection to vzodio1b closed.
bash-3.2#
bash-3.2# rlogin vzodio1c -l root
Password:
Last login: Thu Mar 13 12:30:34 from vpn-129-150-118
Sun Microsystems Inc. SunOS 5.11 snv_78      October 2007
# svcs multi-user-server
STATE      STIME      FMRI
online     12:54:29  svc:/milestone/multi-user-server:default
# exit
Connection to vzodio1c closed.
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

7. Install the HA-xVM Server x86-64 agent.

Perform this step on all cluster nodes.

Save the SUNWscxvm.tar file you downloaded and then extract and pkgadd.

```
bash-3.2# tar -xf SUNWscxvm.tar
bash-3.2# pkgadd -d .
```

The following packages are available:

```
 1 SUNWscxvm      Open HA Cluster for xVM x86-64 Guest Domains
                   (i386) 3.2.0,REV=2008.03.19.08.02,debug
```

Select package(s) you wish to process (or 'all' to process all packages). (default: all) [?,??,q]:

Processing package instance <SUNWscxvm> from </var/tmp>

```
Open HA Cluster for xVM x86-64 Guest Domains(i386) 3.2.0,REV=2008.03.19.08.02,debug
Copyright 2008 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Using </opt> as the package base directory.
## Processing package information.
## Processing system information.
   4 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.
```

Installing Open HA Cluster for xVM x86-64 Guest Domains as <SUNWscxvm>

```
## Installing part 1 of 1.
/opt/SUNWscxvm/bin/control_xvm
/opt/SUNWscxvm/bin/functions
/opt/SUNWscxvm/bin/ppkssh
/opt/SUNWscxvm/etc/SUNW.xvm
/opt/SUNWscxvm/etc/config
/opt/SUNWscxvm/util/xvm_config
/opt/SUNWscxvm/util/xvm_register
/opt/cluster/lib/rgm/gdsdata/SUNW.xvm <symbolic link>
[ verifying class <none> ]
```

Installation of <SUNWscxvm> was successful.

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

8. Setup a public key ssh authentication between Domain-0 and each domu.

The HA-xVM Server x86-64 agent provides a probe for Solaris/Linux guest domains. This probe uses ssh to check if the required milestone or runlevel has been reached by the guest domain. However, in order for the probe to work you must setup a public key ssh authentication between Domain-0 and the guest domain so that the ssh session does not require a password.

As we will place domu1, domu2 and domu3 under the control of the agent you must repeat this step for each domu. Please note you are free to use your own group name and user name.

Perform this step for domu1 (vzodio1a), domu2 (vzodio1b) and domu3 (vzodio1c).

```
bash-3.2# rlogin vzodio1a -l root
Password:
Last login: Fri Mar 14 03:19:08 from vpn-129-150-117
Sun Microsystems Inc. SunOS 5.11 snv_78      October 2007
#
# svcadm enable ssh
# svcadm enable ftp
# groupadd -g 1000 fmgrou
# useradd -g 1000 -u 1000 -d /export/fmuser fmuser
# mkdir -p /export/fmuser
# chown fmuser:fmgrou /export/fmuser
# passwd fmuser
# su - fmuser
$ mkdir .ssh
$ chmod 700 .ssh
$
```

HA-xVM Server x86-64 agent cheat sheet

8. Setup a public key ssh authentication between Domain-0 and each domu (continued).

Perform this part on Domain-0 for each cluster node and repeat these steps for each domu.

```
bash-3.2# groupadd -g 1000 fmgroup
bash-3.2# useradd -g 1000 -u 1000 -d /export/fmuser fmuser
bash-3.2# mkdir -p /export/fmuser
bash-3.2# chown fmuser:fmgroup /export/fmuser
bash-3.2# passwd fmuser
bash-3.2# su - fmuser
$ ssh-keygen -t dsa                                (Note, do not enter a passphrase)
Generating public/private dsa key pair.
Enter file in which to save the key (/export/fmuser/.ssh/id_dsa):
Created directory '/export/fmuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /export/fmuser/.ssh/id_dsa.
Your public key has been saved in /export/fmuser/.ssh/id_dsa.pub.
The key fingerprint is:
57:70:9a:65:20:95:cb:d4:4a:63:72:ff:a3:ac:43:f4 fmuser@podio1
$
$ cd .ssh
$ cp id_dsa.pub authorized_keys
$
$ ftp vzodio1a                                    (Note, vzodio1a is domu1's hostname, repeat for domu2 and domu3)
Connected to vzodio1a.
220 vzodio1a FTP server ready.
Name (vzodio1a:root): fmuser
331 Password required for fmuser.
Password:
230 User fmuser logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd .ssh
250 CWD command successful.
ftp> append authorized_keys                        <--- Note we are appending
200 PORT command successful.
150 Opening BINARY mode data connection for authorized_keys.
226 Transfer complete.
local: authorized_keys remote: authorized_keys
603 bytes sent in 0.00028 seconds (2110.69 Kbytes/s)
ftp> bye
221-You have transferred 603 bytes in 1 files.
221-Total traffic for this session was 1029 bytes in 1 transfers.
221-Thank you for using the FTP service on vzodio1a.
221 Goodbye.
$
```

HA-xVM Server x86-64 agent cheat sheet

8. Setup a public key ssh authentication between Domain-0 and each domu (continued).

Perform this part on domu1, and repeat for domu2 and domu3.

```
# su - fmuser
Sun Microsystems Inc. SunOS 5.11 snv_78      October 2007
$ cd .ssh
$ chmod 700 authorized_keys
$ ls -l
total 2
-rwx----- 1 fmuser  fmgrou  603 Mar 14 05:08 authorized_keys
$
```

Perform this part as root on Domain-0 for each cluster node.

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1a
The authenticity of host 'vzodio1a (10.11.188.33)' can't be established.
RSA key fingerprint is 73:96:e1:cb:1a:d2:d1:95:16:44:3f:1d:28:f7:6a:ec.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vzodio1a,10.11.188.33' (RSA) to the list of known hosts.
Last login: Sat Mar 15 06:19:21 2008 from podio1
Sun Microsystems Inc. SunOS 5.11      snv_78  October 2007
$
$ uname -n
vzodio1a
$ exit
Connection to vzodio1a closed.
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1a
Last login: Sat Mar 15 06:21:49 2008 from podio1
Sun Microsystems Inc. SunOS 5.11      snv_78  October 2007
$ exit
Connection to vzodio1a closed.
bash-3.2#
```

Note, repeat the above (ssh -l fmuser ...) as root on each Domain-0 cluster node.

HA-xVM Server x86-64 agent cheat sheet

8 (a). Verify a public key ssh authentication between Domain-0 and each domu.

Perform this part as root on each Domain-0 for each cluster node. This step must succeed before you proceed further.

```
bash-3.2# uname -n
podio1
bash-3.2# for i in a b c; do ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1$i 'uname -n'; done
vzodio1a
vzodio1b
vzodio1c
bash-3.2#
```

```
bash-3.2# uname -n
podio2
bash-3.2# for i in a b c ; do ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1$i 'uname -n'; done
vzodio1a
vzodio1b
vzodio1c
bash-3.2#
```

```
bash-3.2# uname -n
podio3
bash-3.2# for i in a b c ; do ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1$i 'uname -n'; done
vzodio1a
vzodio1b
vzodio1c
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

9. Setup the HA-xVM Server x86-64 agent administrative file system.

In order for the HA-xVM Server x86-64 agent to perform administrative tasks when starting a domain or performing a migration or live migration of a domain we require an accessible file system which the HA-xVM Server x86-64 agent can read/write information to from any node within the cluster. This can be a clustered file system or NFS.

Ideally we would create a specific cluster file system for the HA-xVM Server x86-64 agent administrative file system, however currently Bug ID: 6579356 prevents us from creating a second diskset on Nevada. As such we will use the cluster file system that domu1 uses, i.e. /global/bootdisk.

So for the purpose of this cheat sheet /global/bootdisk/haxvm-admin will be used.

The following information will be stored (read/written/deleted) in /global/bootdisk/haxvm-admin.

- noop flag used by the STOP and START methods whenever a migration or live migration is being done.
- XML dump of the domain after a successful start, which is used to define the domain on a cluster node.

Check that /global/bootdisk is currently globally mounted.

```
bash-3.2# mount -p | grep /global/bootdisk
/dev/md/dg_d1/dsk/d100 - /global/bootdisk ufs - no rw,intr,largefiles,nologging,noquota,xattr,nodfratime,onerror=panic,global
bash-3.2#
bash-3.2# mkdir /global/bootdisk/haxvm-admin
bash-3.2#
bash-3.2# ls -l /global/bootdisk
total 11012850
drwxr-xr-x  2 root  root      512 Mar 16 11:43 haxvm-admin
-rwxr-xr-x  1 root  root    17179869184 Mar 16 07:44 root.img
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

10. Place domain domu1 under the control of the HA-xVM Server x86-64 agent.

Perform this step on the cluster node where domu1 is currently running.

```
bash-3.2# uname -n
podio1
bash-3.2# virsh list --all
 Id Name                State
-----
  0 Domain-0            running
  3 domu1               blocked

bash-3.2# clrt register SUNW.gds
bash-3.2# clrt register SUNW.HASStoragePlus
bash-3.2# clrg create RG1
bash-3.2# clrs create -g RG1 -t SUNW.HASStoragePlus -p FilesystemMountPoints=/global/bootdisk domu1-has
bash-3.2# clrg online -M RG1
```

Before proceeding, ensure that domu1 has reached multi-user-server milestone and that the RG is running on the node where the domain is running, If it is not you MUST switch the RG to that node, i.e.

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1a 'svcs multi-user-server'
STATE      STIME      FMRI
online     5:03:33   svc:/milestone/multi-user-server:default
bash-3.2#
bash-3.2# clrg switch -n podio1 RG1
bash-3.2# cd /opt/SUNWscxvm/util
bash-3.2#
bash-3.2# vi xvm_config                                (Amend xvm_config with the following)
```

```
RS=domu1-rs
RG=RG1
HAS_RS=domu1-has
DOMAIN=domu1
PLUGIN_PROBE="/opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzodio1a:multi-user-server:online"
FAILOVER="migrate -live"
ADMIN=/global/bootdisk/haxvm-admin
```

Test the PLUGIN_PROBE, note that 27 is passed by the HA-xVM Server x86-64 agent and represents 90% of PROBE_TIMEOUT (Default=30).

```
bash-3.2# /opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzodio1a:multi-user-server 27
vzodio1a multi-user-server is online
```

```
bash-3.2# ./xvm_register
```

HA-xVM Server x86-64 agent cheat sheet

10 (a). Verify that the OHAC resource is online.

Perform this step on the cluster node where the domain domu1 is running.

```
bash-3.2# uname -n
podio1
bash-3.2# clrs status -g RG1 +
```

=== Cluster Resources ===

Resource Name	Node Name	State	Status Message
-----	-----	-----	-----
domu1-has	podio1	Online	Online
	podio2	Offline	Offline
	podio3	Offline	Offline
domu1-rs	podio1	Online	Online
	podio2	Offline	Offline
	podio3	Offline	Offline

```
bash-3.2#
```

Note, that placing the domain under OHAC control did not require the domain to be shutdown.

```
bash-3.2# virsh list --all
 Id Name                State
-----
  0 Domain-0             running
  4 domu1                 blocked
```

```
bash-3.2#
```

Note that while the domain is managed by OHAC you must only issue OHAC commands against the domain. If you need to issue xVM commands against the domain you must place the domain outside of OHAC control. The only exception to this are xVM display commands.

HA-xVM Server x86-64 agent cheat sheet

10 (b). Disable/enable the OHAC resource.

Perform this step on the cluster node where the domain is running.

```
bash-3.2# uname -n
podio1
bash-3.2# clrs disable domu1-rs
bash-3.2#
bash-3.2# virsh list --all
  Id Name                State
-----
  0 Domain-0             running

bash-3.2#
bash-3.2# clrs enable domu1-rs
bash-3.2#
bash-3.2# virsh list --all
  Id Name                State
-----
  0 Domain-0             running
  6 domu1                blocked

bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

10 (c). Switch the OHAC resource to the other node.

Note, that when OHAC resource domu1-rs was registered (Page 20), FAILOVER="migrate -live" was specified. As such when the RG is switched between nodes the domain domu1 will be live migrated.

```
bash-3.2# uname -n
```

```
podio1
```

```
bash-3.2# virsh list --all
```

```
Id Name                State
-----
 0 Domain-0             running
 6 domu1                blocked
```

```
bash-3.2# clrg status RG1
```

<--- Cluster status BEFORE switch

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline

```
bash-3.2# clr status domu1-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
domu1-rs	podio1	Online	Online - Service is online.
	podio2	Offline	Offline
	podio3	Offline	Offline

```
bash-3.2#
```

```
bash-3.2# clrg switch -n podio2 RG1
```

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

10 (c). Switch the OHAC resource to the other node (continued).

```
bash-3.2# uname -n
```

```
podio1
```

```
bash-3.2# virsh list --all
```

```
Id Name                State
-----
 0 Domain-0             running
```

```
bash-3.2# clrg status RG1
```

<--- Cluster status AFTER switch

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Offline
	podio2	No	Online
	podio3	No	Offline

```
bash-3.2# clrs status domu1-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
domu1-rs	podio1	Offline	Offline
	podio2	Online	Online - Service is online.
	podio3	Offline	Offline

```
bash-3.2#
```

```
bash-3.2# uname -n
```

```
podio2
```

```
bash-3.2# virsh list --all
```

```
Id Name                State
-----
 0 Domain-0             running
 3 domu2                 blocked
 4 domu1                 blocked
```

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

10 (c). Switch the OHAC resource to the other node (continued).

Please note that within the syslog the following messages were output.

```
Mar 16 01:54:13 podio1 SC[SUNWscxvm.stop]:RG1:domu1-rs: [ID 224972 daemon.notice] Domain domu1 is being live migrated to podio2
Mar 16 01:54:27 podio1 genunix: [ID 408114 kern.info] /xpvd/xnb@4,0 (xnbo0) offline
Mar 16 01:54:27 podio1 mac: [ID 736570 kern.info] NOTICE: vnic900 unregistered
Mar 16 01:54:27 podio1 SC[SUNWscxvm.stop]:RG1:domu1-rs: [ID 510634 daemon.notice] Domain domu1 successfully live migrated to podio2
```

Furthermore, once the domain is under OHAC control you can switch the RG from any cluster node, i.e. note that RG1 is running on node podio2 when we issue the switch from node podio1.

```
bash-3.2# uname -n
podio1
bash-3.2# clrg status RG1
```

=== Cluster Resource Groups ===

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Offline
	podio2	No	Online
	podio3	No	Offline

```
bash-3.2# clrg switch -n podio1 RG1
bash-3.2# clrg status RG1
```

=== Cluster Resource Groups ===

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline

```
bash-3.2# virsh list --all
Id Name          State
-----
 0 Domain-0      running
 7 domu1         blocked
```

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

11 (a). Fault injection test – Crash the node.

Crash the node that is currently hosting domain domu1.

```
bash-3.2# clrg status RG1
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline

```
bash-3.2# uname -n
```

```
podio1
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
7	domu1	blocked

```
bash-3.2# uadmin 1 1
```

The cluster will detect that node1 (podio1) has failed and will attempt to rehost RG1 onto node2 (podio2). After a minute or so the cluster will have successfully restarted the domain on node2 (podio2).

```
bash-3.2# clrg status RG1
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Offline
	podio2	No	Online
	podio3	No	Offline

```
bash-3.2# uname -n
```

```
podio2
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
3	domu2	blocked
5	domu1	blocked

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

11 (b). Fault injection test – Evacuate the node.

Evacuate the node that is currently hosting domain domu1. Note that cluster commands can be issued from any node within the cluster.

```
bash-3.2# clrg status RG1
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Offline
	podio2	No	Online
	podio3	No	Offline

```
bash-3.2# uname -n
```

```
podio2
```

```
bash-3.2# virsh list --all (Note that domu2 will not be affected by the evacuate as it is not yet under HA-xVM Server x86-64 agent control)
```

Id	Name	State
0	Domain-0	running
3	domu2	blocked
5	domu1	blocked

```
bash-3.2# clrg evacuate -n podio2
```

```
bash-3.2# clrg status RG1
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline

```
bash-3.2#
```

It's important to also understand that the HA-xVM Server x86-64 agent is designed to perform a normal failover if the node is being evacuated, which simply means that the domain will be gracefully stopped and started on another node determined by the Resource Group Manager (RGM). If you wanted to live migrate the domain, a switch of the RG should be performed.

Also, please note the following syslog messages were output.

```
Mar 16 05:45:30 podio2 SC[SUNWscxvm.stop]:RG1:domu1-rs: [ID 517403 daemon.notice] Node is being evacuated, normal failover will be performed
```

HA-xVM Server x86-64 agent cheat sheet

11 (c). Fault injection test – destroy the domain.

```
bash-3.2# clrg status RG1
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline

```
bash-3.2# uname -n
```

```
podio1
```

```
bash-3.2# uptime
```

```
5:51am up 9 min(s), 2 users, load average: 0.02, 0.13, 0.11
```

```
bash-3.2# virsh list --all
```

Id Name	State
0 Domain-0	running
1 domu1	blocked

```
bash-3.2# virsh destroy domu1
```

```
Domain domu1 destroyed
```

```
bash-3.2#
```

After a short time (Default Thorough_probe_interval=60 seconds), the agent probe will detect that the domain has failed and will restart it. Note that domu1-rs will not appear “Online” until the multi-user-server milestone state is reached.

```
bash-3.2# clrs status domu1-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
domu1-rs	podio1	Online	Online
	podio2	Offline	Offline
	podio3	Offline	Offline

```
bash-3.2# virsh list --all
```

Id Name	State
0 Domain-0	running
2 domu1	blocked

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

12. Place domu2 under the control of the HA-xVM Server x86-64 agent.

Perform this step on the cluster node where the domu2 is currently running.

Note, we cannot create a HAS resource for domu2 as we were unable to create a second SVM diskset for /dev/did/rdisk/d6 because of Bug ID: 6579356.

```
bash-3.2# uname -n
podio2
bash-3.2# virsh list --all
 Id Name                State
-----
  0 Domain-0             running
  3 domu2                 blocked
```

```
bash-3.2# clrg create RG2
bash-3.2# clrg online -M RG2
```

Before proceeding, ensure that domu2 has reached multi-user-server milestone and that the RG is running on the node where the domain is running, If it is not you MUST switch the RG to that node, i.e.

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1b 'svcs multi-user-server'
STATE      STIME      FMRI
online     5:03:33   svc:/milestone/multi-user-server:default
bash-3.2#
bash-3.2# clrg switch -n podio2 RG2
bash-3.2# cd /opt/SUNWscxvm/util
bash-3.2# vi xvm_config
```

(Amend xvm_config with the following)

```
RS=domu2-rs
RG=RG2
HAS_RS=
DOMAIN=domu2
PLUGIN_PROBE="/opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzodio1b:multi-user-server:online"
FAILOVER="migrate -live"
ADMIN=/global/bootdisk/haxvm-admin
```

Test the PLUGIN_PROBE, note that 27 is passed by the HA-xVM Server x86-64 agent and represents 90% of PROBE_TIMEOUT (Default=30).

```
bash-3.2# /opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzodio1b:multi-user-server:online 27
vzodio1a multi-user-server is online
```

```
bash-3.2# ./xvm_register
```

HA-xVM Server x86-64 agent cheat sheet

13. Place domu3 under the control of the HA-xVM Server x86-64 agent.

Perform this step on the cluster node where the domu3 is currently running. As domu3 uses a zpool, we must stop domu3 before placing the zpool under OHAC control. Note also that we cannot live migrate domu3 as the zpool is only accessible on one node at a time.

Before proceeding, ensure that domu3 has reached multi-user-server milestone

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio1c 'svcs multi-user-server'
STATE          STIME    FMRI
online         5:03:33 svc:/milestone/multi-user-server:default
bash-3.2#
bash-3.2# uname -n
podio3
bash-3.2# virsh list --all
 Id Name                State
-----
 0 Domain-0             running
 1 domu3                blocked

bash-3.2# xm shutdown -w domu3
Domain domu3 terminated
All domains terminated
bash-3.2#
bash-3.2# clrg create RG3
bash-3.2# clrs create -g RG3 -t SUNW.HASStoragePlus -p zpools=zfs_bootdisk domu3-has
bash-3.2# clrg online -M RG3
```

As we stopped domu3 earlier, when domu3-rs is created it will start domu3. Nevertheless you MUST still switch the RG to the node where domu3 was running, i.e. defined.

```
bash-3.2# clrg switch -n podio3 RG3
bash-3.2# cd /opt/SUNWscxvm/util
bash-3.2# vi xvm_config
```

(Amend xvm_config with the following)

```
RS=domu3-rs
RG=RG3
HAS_RS=domu3-has
DOMAIN=domu3
PLUGIN_PROBE="/opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzodio1c:multi-user-server:online"
FAILOVER="normal"
ADMIN=/global/bootdisk/haxvm-admin
```

```
bash-3.2# ./xvm_register
```

HA-xVM Server x86-64 agent cheat sheet

14. All shared storage domains under HA-xVM Server x86-64 agent control.

At this stage we have now successfully placed domu1, domu2 and domu3 under the control of the HA-xVM Server x86-64 agent.

```
bash-3.2# clrg status +
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline
RG2	podio1	No	Offline
	podio2	No	Online
	podio3	No	Offline
RG3	podio1	No	Offline
	podio2	No	Offline
	podio3	No	Online

```
bash-3.2#
```

The administrative file system for the HA-xVM Server x86-64 agent contains dumped XML format definitions for each domain, allowing the domains to move between cluster nodes without having to define each domain. Note the xml file is (re)created after every successful start of the domain.

```
bash-3.2# ls -l /global/bootdisk/haxvm-admin
```

```
total 6
-rw-r--r-- 1 root root 638 Mar 16 12:08 domu1-rs.xml
-rw-r--r-- 1 root root 629 Mar 16 12:13 domu2-rs.xml
-rw-r--r-- 1 root root 635 Mar 16 12:16 domu3-rs.xml
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

15. Assign a strong negative affinity against domu3.

Perform this step on any cluster x86-64 node.

Note that currently all domains are running on separate nodes within the cluster. We will now assign a strong negative affinity against the RG that manages domu3. This simply means that once the strong negative affinity is in place domu1 and domu2 will always “push” domu3 away thereby freeing system resources for domu1 and domu2. If there is only one node left within the cluster then domu1 and domu2 will remain active while domu3 will be shutdown, until another cluster node comes online.

```
bash-3.2# clrs status domu1-rs domu2-rs domu3-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
domu1-rs	podio1	Online	Online - Service is online.
	podio2	Offline	Offline
	podio3	Offline	Offline
domu2-rs	podio1	Offline	Offline
	podio2	Online	Online - Service is online.
	podio3	Offline	Offline
domu3-rs	podio1	Offline	Offline
	podio2	Offline	Offline
	podio3	Online	Online - Service is online.

```
bash-3.2#
```

```
bash-3.2# clrg set -p RG_Affinities---RG1,--RG2 RG3
```

HA-xVM Server x86-64 agent cheat sheet

16. Switch domu1 to the node where domu3 is ruuning.

Perform this step on any cluster x86-64 node.

Within Step 15 we setup a strong negative affinity so that domu1 and domu2 become our favoured domains and domu3 gives way whenever domu1 or domu2 resides on the same node. Here, we'll switch the RG that contains domu1 and observe what happens to domu3.

```
bash-3.2# clrg status RG1 RG2 RG3
```

<--- Cluster status BEFORE switch

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline
RG2	podio1	No	Offline
	podio2	No	Online
	podio3	No	Offline
RG3	podio1	No	Offline
	podio2	No	Offline
	podio3	No	Online

```
bash-3.2#
```

```
bash-3.2# clrg switch -n podio3 RG1
```

```
(C258827) NOTICE: Resource group(s) {RG3 on node podio3} were either switched to different nodes or brought offline as a result of their strong resource group affinities for resource group RG1.
```

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

16. Switch our domu1 to the node where domu3 is running (continued).

Note that RG1 is now running on podio3. RG3 could not move to podio2 as RG2 is running there, instead RG3 could only move to RG1. Finally also note that we did not have to define domu3 on podio1, the HA-xVM Server x86-64 agent did that for us using it's administrative file system /global/bootdisk/haxvm-admin.

```
bash-3.2# cllrg status RG1 RG2 RG3
```

<--- Cluster status AFTER switch

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG1	podio1	No	Offline
	podio2	No	Offline
	podio3	No	Online
RG2	podio1	No	Offline
	podio2	No	Online
	podio3	No	Offline
RG3	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

17 (a). Create a ZFS zpool and share using HA-NFS.

As noted at the start of this cheat sheet, for network storage we will deploy a ZFS/NFS server from outside of the OHAC/xVM cluster. Instead we will use another cluster to host the ZFS/NFS service. Please note that we will not use the ZFS sharenfs=on property for our ZFS/NFS service, refer to http://blogs.sun.com/SC/entry/why_sunw_nfs_is_required as to why. If you decide NOT to use another cluster to host the ZFS/NFS service the ZFS sharenfs=on property can be used.

Perform this step on one SC 3.2/ZFS node.

Note, that this cheat sheet shows simple configurations and as is not intended to represent best practice or address any performance issues by using the disks in the way we are below. Instead if you have more storage then of course feel free to organise the ZFS files differently. The purpose is simply to show the mechanism of using ZFS/NFS with a cluster and with PV guest domains.

```
bash-3.00# uname -n
pmutt1
bash-3.00# zpool create bootdisk c4t600C0FF000000000008AEB6EAC43CB00d0
bash-3.00# zfs create bootdisk/domu4
bash-3.00#
bash-3.00# mkdir /bootdisk/SUNW.nfs
bash-3.00# echo 'share -F nfs -o rw -d "HA-NFS" /bootdisk/domu4' > /bootdisk/SUNW.nfs/dfstab.zfs-nfs
bash-3.00# clrt register SUNW.HASStoragePlus
bash-3.00# clrt register SUNW.nfs
bash-3.00# clrg create -p Pathprefix=/bootdisk zfs-rg
bash-3.00# clrslh create -g zfs-rg -h mutt-1 zfs-lh
bash-3.00# clrs create -g zfs-rg -t SUNW.HASStoragePlus -p zpools=bootdisk zfs-has
bash-3.00# clrg online -M zfs-rg
bash-3.00# clrs create -g zfs-rg -t SUNW.nfs -y Resource_dependencies=zfs-has zfs-nfs
bash-3.00# clrs status -g zfs-rg +
```

=== Cluster Resources ===

Resource Name	Node Name	State	Status Message
zfs-lh	pmutt1	Online	Online - LogicalHostname online.
	pmutt2	Offline	Offline
zfs-has	pmutt1	Online	Online
	pmutt2	Offline	Offline
zfs-nfs	pmutt1	Online	Online - Service is online.
	pmutt2	Offline	Offline

HA-xVM Server x86-64 agent cheat sheet

17 (b). On each OHAC/xVM x86-64 node, mount the ZFS/NFS file systems.

Perform this step on each cluster node.

Note, mutt-1 shown here represents the logical hostname that is hosted by the SC 3.2/ZFS cluster.

```
bash-3.2# mkdir -p /bootdisk/domu4
bash-3.2#
bash-3.2# cat >> /etc/vfstab <<-END
mutt-1:/bootdisk/domu4 - /bootdisk/domu4 nfs 3 yes rw,soft
END
bash-3.2#
```

Ensure, NFS client and it's dependent SMF services are enabled.

```
bash-3.2# svcs -a | grep nfs
disabled      1:56:53 svc:/network/nfs/server:default
disabled      1:58:54 svc:/network/nfs/rquota:default
online        2:31:41 svc:/network/nfs/mapid:default
online        2:32:08 svc:/network/nfs/cbd:default
online        2:32:57 svc:/network/nfs/status:default
online        2:32:57 svc:/network/nfs/nlockmgr:default
online        2:32:58 svc:/network/nfs/client:default
bash-3.2#
```

```
bash-3.2# df -k /bootdisk/domu4
Filesystem      kbytes  used  avail capacity Mounted on
mutt-1:/bootdisk/domu4
                 84978954 5548284 79430670    7%  /bootdisk/domu4
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

18. Install domu4 on the NFS filesystem.

```
bash-3.2# uname -n
podio1
bash-3.2# virt-install -n domu4 --paravirt -r 1024 --nographics -f /bootdisk/domu4/root.img -s 16 -l nfs:10.4.195.10:/export/linux/centos50-64
```

```
Starting install...
Creating storage file... 100% |=====| 16 GB 00:00
Creating domain... 0 B 00:00
Bootdata ok (command line is method=nfs:10.4.195.10:/export/linux/centos50-64)
Linux version 2.6.18-8.el5xen (mockbuild@builder6.centos.org) (gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #1 SMP Thu Mar 15 19:56:43 EDT 2007
BIOS-provided physical RAM map:
Xen: 0000000000000000 - 0000000040800000 (usable)
No mptable found.
Built 1 zonelists. Total pages: 264192
Kernel command line: method=nfs:10.4.195.10:/export/linux/centos50-64
Initializing CPU#0
PID hash table entries: 4096 (order: 12, 32768 bytes)
Xen reported: 2393.638 MHz processor.
Console: colour dummy device 80x25
Dentry cache hash table entries: 262144 (order: 9, 2097152 bytes)
Inode-cache hash table entries: 131072 (order: 8, 1048576 bytes)
Software IO TLB disabled
Memory: 1014272k/1056768k available (2321k kernel code, 33572k reserved, 1312k data, 168k init)
Calibrating delay using timer specific routine.. 5986.39 BogoMIPS (lpj=11972796)
Security Framework v1.0.0 initialized
SELinux: Initializing.

<... snipped ...>
```

HA-xVM Server x86-64 agent cheat sheet

19. Setup a public key ssh authentication between Domain-0 and domu4.

The HA-xVM Server x86-64 agent for guest domains agent provides an optional default probe for Solaris/Linux guest domains. This probe uses ssh to check if the required milestone or runlevel has been reached by the guest domain. However, in order for the probe to work you must setup a public key ssh authentication between Domain-0 and the guest domain so that the ssh session does not require a password.

Please note you are free to use your own group name and user name.

Perform this step for domu4 (vzodio2a).

```
bash-3.2# bash-3.2# ssh vzodio2a
root@vzodio2a's password:
Last login: Tue Mar 18 05:17:27 2008 from 10.11.188.21
[root@vzodio2a ~]# uname -or
2.6.18-8.el5xen GNU/Linux
[root@vzodio2a ~]#
[root@vzodio2a ~]# groupadd -g 1000 fmgrou
[root@vzodio2a ~]# useradd -g 1000 -u 1000 -d /export/fmuser fmuser
[root@vzodio2a ~]# mkdir -p /export/fmuser
[root@vzodio2a ~]# chown fmuser:fmgrou /export/fmuser
[root@vzodio2a ~]# passwd fmuser
[root@vzodio2a ~]# su - fmuser
-bash-3.1$ mkdir .ssh
-bash-3.1$ chmod 700 .ssh
-bash-3.1$
```

HA-xVM Server x86-64 agent cheat sheet

19. Setup a public key ssh authentication between Domain-0 and domu4 (continued).

Perform this part on Domain-0 for each cluster node. Note this step assumes you have already setup the fmuser user and it's private/public keys, if not refer to slide 15.

```
bash-3.2# uname -n
podio1
bash-3.2#
bash-3.2# su - fmuser
$ cd .ssh
$ scp authorized_keys fmuser@vzodio2a:/export/fmuser
authorized_keys      100% |*****|
   00:00
$
```

Perform this part on domu4 (vzodio2a)

```
bash-3.2# ssh -l fmuser vzodio2a
fmuser@vzodio2a's password:
Last login: Tue Mar 18 04:30:05 2008 from 10.11.188.22
-bash-3.1$ ls -l
total 8
-rw-r--r-- 1 fmuser fmgrou 603 Mar 18 05:30 authorized_keys
-bash-3.1$
-bash-3.1$ cat authorized_keys >> .ssh/authorized_keys
-bash-3.1$
-bash-3.1$ cd .ssh
-bash-3.1$ chmod 700 authorized_keys
-bash-3.1$ ls -l
total 8
-rwx----- 1 fmuser fmgrou 2412 Mar 18 05:31 authorized_keys
-bash-3.1$
```

Note repeat the above for each cluster node, in our case node2 (podio2) and node3 (podio3).

HA-xVM Server x86-64 agent cheat sheet

19 (a). Verify a public key ssh authentication between Domain-0 and domu4.

Perform this part as root on Domain-0 of each cluster node.

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2a
The authenticity of host 'vzodio2a (10.11.188.36)' can't be established.
RSA key fingerprint is 7f:17:3c:c0:64:fc:e2:8b:3a:fd:60:80:44:26:bc:6a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vzodio2a,10.11.188.36' (RSA) to the list of known hosts.
Last login: Tue Mar 18 05:42:06 2008 from 10.11.188.22
-bash-3.1$
-bash-3.1$ uname -n
vzodio2a
-bash-3.1$ exit
logout
Connection to vzodio2a closed.
bash-3.2#
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2a 'uname -n'
vzodio2a
bash-3.2#
```

Note, repeat the above (ssh -l fmuser ...) as root on each Domain-0 cluster node.

Before we place domu4 under the control of the HA-xVM Server x86-64 agent, we must ensure that domu4 can boot successfully to runlevel 3. This runlevel is required as the agent probes domu4 to determine it has successfully booted to runlevel 3. You must also ensure that ssh works from Domain-0 on each cluster node to domu4 (vzodio2a). This part must succeed before you continue.

```
bash-3.2# uname -n
podio1
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2a '/sbin/runlevel'
bash-3.2#

bash-3.2# uname -n
podio2
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2a '/sbin/runlevel'
bash-3.2#

bash-3.2# uname -n
podio3
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2a '/sbin/runlevel'
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

20. Place domain domu4 under the control of the HA-xVM Server x86-64 agent.

Perform this step on the cluster node where domu4 is currently running.

```
bash-3.2# uname -n
podio1
bash-3.2# virsh list --all
 Id Name                State
-----
  0 Domain-0            running
  2 domu4                blocked
  3 domu1                blocked
```

```
bash-3.2# clrg create RG4
bash-3.2# clrg online -M RG4
```

Before proceeding, ensure that domu4 has reached runlevel 3 and that the RG is running on the node where the domain is running, If it is not you MUST switch the RG to that node, i.e.

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzdio2a '/sbin/runlevel'
N 3
bash-3.2#
bash-3.2# clrg switch -n podio1 RG4
bash-3.2# cd /opt/SUNWscxvm/util
bash-3.2#
bash-3.2# vi xvm_config
```

(Amend xvm_config with the following)

```
RS=domu4-rs
RG=RG4
HAS_RS=
DOMAIN=domu4
PLUGIN_PROBE="/opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzdio2a:runlevel:3"
FAILOVER="migrate --live"
ADMIN=/global/bootdisk/haxvm-admin
```

Test the PLUGIN_PROBE, note that 27 is passed by the HA-xVM Server x86-64 agent and represents 90% of PROBE_TIMEOUT (Default=30).

```
bash-3.2# /opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzdio2a:runlevel:3 27
vzdio2a runlevel is 3
bash-3.2#
bash-3.2# ./xvm_register
```

HA-xVM Server x86-64 agent cheat sheet

21. Live migrate domu4 to node3.

```
bash-3.2# clrg status RG4
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
RG4	podio1	No	Online
	podio2	No	Offline
	podio3	No	Offline

```
bash-3.2# clrs status domu4-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
domu4-rs	podio1	Online	Online
	podio2	Offline	Offline
	podio3	Offline	Offline

```
bash-3.2#
```

```
bash-3.2# clrg switch -n podio3 RG4
```

```
bash-3.2#
```

```
bash-3.2# clrs status domu4-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
domu4-rs	podio1	Offline	Offline
	podio2	Offline	Offline
	podio3	Online	Online

```
bash-3.2#
```

Note the following messages will appear on node1's (podio1) syslog.

```
Mar 18 06:57:35 podio1 SC[SUNWscxvm.stop]:RG4:domu4-rs: [ID 224972 daemon.notice] Domain domu4 is being live migrated to podio3
```

```
Mar 18 06:57:49 podio1 SC[SUNWscxvm.stop]:RG4:domu4-rs: [ID 510634 daemon.notice] Domain domu4 successfully live migrated to podio3
```

HA-xVM Server x86-64 agent cheat sheet

22 (a). Clone domain domu4 to create domu4C.

First we need to place the domain outside of OHAC control.

```
bash-3.2# clrs status domu4-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
-----	-----	----	-----
domu4-rs	podio1	Offline	Offline
	podio2	Offline	Offline
	podio3	Online	Online - Service is online.

```
bash-3.2# clrs disable domu4-rs
```

```
bash-3.2#
```

```
bash-3.2# clrs status domu4-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
-----	-----	----	-----
domu4-rs	podio1	Offline	Offline
	podio2	Offline	Offline
	podio3	Offline	Offline

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

22 (b). Copy and create/amend the XML file for cloned domain domu4C.

Within this step we will copy domu4's xml file while also changing the domain name, removing the uuid entry and changing the disk location.

Ref: <http://opensolaris.org/os/community/xen/docs/cloning-domains/>

```
bash-3.2# ls -l /global/bootdisk/haxvm-admin
total 8
-rw-r--r-- 1 root root 638 Mar 18 05:59 domu1-rs.xml
-rw-r--r-- 1 root root 629 Mar 18 06:25 domu2-rs.xml
-rw-r--r-- 1 root root 635 Mar 18 06:26 domu3-rs.xml
-rw-r--r-- 1 root root 682 Mar 18 07:04 domu4-rs.xml
bash-3.2#
bash-3.2# for i in domu4C
do
cp /global/bootdisk/haxvm-admin/domu4-rs.xml /var/tmp/$i
sed -e 's/<name>domu4<\/name>/<name>"$i"<\/name>/' /var/tmp/$i > /var/tmp/$i.1
sed /uuid/d /var/tmp/$i.1 >/var/tmp/$i.2
sed -e 's/\/bootdisk\/domu4\/root.img\/bootdisk\/"$i"\/root.img/' /var/tmp/$i.2 > /global/bootdisk/haxvm-admin/${i}-rs.xml
rm /var/tmp/$i /var/tmp/$i.1 /var/tmp/$i.2
done
bash-3.2#
bash-3.2# bash-3.2# ls -l
total 10
-rw-r--r-- 1 root root 638 Mar 18 05:59 domu1-rs.xml
-rw-r--r-- 1 root root 629 Mar 18 06:25 domu2-rs.xml
-rw-r--r-- 1 root root 635 Mar 18 06:26 domu3-rs.xml
-rw-r--r-- 1 root root 682 Mar 18 07:04 domu4-rs.xml
-rw-r--r-- 1 root root 636 Mar 18 07:17 domu4C-rs.xml
bash-3.2#
```

Note that while we have created an XML file for domu4C the disk location does not yet exist.

HA-xVM Server x86-64 agent cheat sheet

23 (a). Clone domu4's ZFS file systems for the cloned domain domu4C.

Perform this step on the SC 3.2/ZFS server currently hosting the zpool bootdisk.

```
bash-3.00# uname -n
pmutt1
bash-3.00# zfs snapshot bootdisk/domu4@clone
bash-3.00# zfs clone bootdisk/domu4@clone bootdisk/domu4C
bash-3.00#
bash-3.00# zfs list -o name,mounted,available,referenced,origin,sharenfs
NAME                MOUNTED  AVAIL  REFER  ORIGIN                SHARENFS
bootdisk            yes      75.3G  40K    -                      off
bootdisk/domu4      yes      75.3G  5.93G  -                      off
bootdisk/domu4@clone -        -      5.93G  -                      -
bootdisk/domu4C     yes      75.3G  5.93G  bootdisk/domu4@clone  off
bash-3.00#
bash-3.00# cat >> /bootdisk/SUNW.nfs/dfstab.zfs-nfs <<-END
share -F nfs -o rw -d "HA-NFS" /bootdisk/domu4C
END
bash-3.00#
```

<--- Note

Note: The zfs-nfs resource dynamically shares the new shares defined above at the next Thorough_Probe_Interval property (by default 120 seconds).

```
bash-3.00# uname -n
pmutt1
bash-3.00# dfshares
RESOURCE                SERVER ACCESS  TRANSPORT
pmutt1:/bootdisk/domu4  pmutt1 -      -
pmutt1:/bootdisk/domu4C pmutt1 -      -
bash-3.00#
```

HA-xVM Server x86-64 agent cheat sheet

23 (b). Mount the cloned domain's ZFS file system.

Note, repeat this step on all OHAX/xVM cluster nodes.

```
bash-3.2# uname -n
podio1
bash-3.2# mkdir -p /bootdisk/domu4C
bash-3.2#
bash-3.2# cat >> /etc/vfstab <<-END
mutt-1:/bootdisk/domu4C - /bootdisk/domu4C nfs 3 yes rw,soft
END
bash-3.2#
```

Note, ensure NFS client is enabled. See page 36.

```
bash-3.2# mount /bootdisk/domu4C
bash-3.2#
bash-3.2# mount -p | grep bootdisk/domu4
mutt-1:/bootdisk/domu4 - /bootdisk/domu4 nfs - no rw,soft,xattr
mutt-1:/bootdisk/domu4C - /bootdisk/domu4C nfs - no rw,soft,xattr
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

24. Manually start and configure domu4C.

Perform this step on the node where you defined the cloned guest domain.

```
bash-3.2# uname -n
podio3
bash-3.2# cd /global/bootdisk/haxvm-admin
bash-3.2# ls -l
total 10
-rw-r--r--  1 root    root      638 Mar 18 05:59 domu1-rs.xml
-rw-r--r--  1 root    root      629 Mar 18 06:25 domu2-rs.xml
-rw-r--r--  1 root    root      635 Mar 18 06:26 domu3-rs.xml
-rw-r--r--  1 root    root      682 Mar 18 07:04 domu4-rs.xml
-rw-r--r--  1 root    root      636 Mar 18 07:17 domu4C-rs.xml
bash-3.2#
bash-3.2# virsh define ./domu4C-rs.xml
Domain domu4C defined from ./domu4C-rs.xml
```

```
bash-3.2# virsh list --all
 Id Name                State
-----
  0 Domain-0             running
  9 domu3                blocked
 - domu4C              shut off
```

```
bash-3.2# virsh start domu4C
Domain domu4C started
```

```
bash-3.2# virsh console domu4C
```

<...snipped...>

Amend IPADDR to reflect your new ip address for domu4C

```
[root@vzodio2a ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Amend HOSTNAME to reflect your new hostname for domu4C

```
[root@vzodio2b ~]# vi /etc/sysconfig/network
```

```
[root@vzodio2a ~]# reboot
```

HA-xVM Server x86-64 agent cheat sheet

24. Manually start and configure domu4C (continued).

Note that domu4C has now been changed to have a different IP address and hostname (vzodio2b).

```
bash-3.2# virsh console domu4C
```

<... snipped ...>

```
CentOS release 5 (Final)
Kernel 2.6.18-8.el5xen on an x86_64
```

```
vzodio2b login: root
```

```
Password:
```

```
Last login: Tue Mar 18 07:49:05 on xvc0
```

```
[root@vzodio2b ~]# ~.
```

```
bash-3.2# uname -n
```

```
podio3
```

```
bash-3.2# virsh list --all
```

Id	Name	State
0	Domain-0	running
9	domu3	blocked
14	domu4C	blocked

```
bash-3.2#
```

Repeat this part as root on each Domain-0 cluster node.

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2b '/sbin/runlevel'
```

```
The authenticity of host 'vzodio2b (10.11.188.37)' can't be established.
```

```
RSA key fingerprint is 7f:17:3c:c0:64:fc:e2:8b:3a:fd:60:80:44:26:bc:6a.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'vzodio2b,10.11.188.37' (RSA) to the list of known hosts.
```

```
N 3
```

```
bash-3.2#
```

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2b '/sbin/runlevel'
```

```
N 3
```

```
bash-3.2#
```

Note, repeat the above (ssh -l fmuser ...) as root on each Domain-0 cluster node.

HA-xVM Server x86-64 agent cheat sheet

25. Place domu4C under the control of the HA-xVM Server x86-64 agent.

Perform this step on the cluster node where domu4 is currently running.

```
bash-3.2# uname -n
podio3
bash-3.2# virsh list --all
 Id Name                State
-----
  0 Domain-0            running
  9 domu3                blocked
 14 domu4C              blocked
```

```
bash-3.2# clrg create RG5
bash-3.2# clrg online -M RG5
```

Before proceeding, ensure that domu4C has reached runlevel 3 and that the RG is running on the node where the domain is running. If it is not you MUST switch the RG to that node, i.e.

```
bash-3.2# ssh -l fmuser -i /export/fmuser/.ssh/id_dsa vzodio2b '/sbin/runlevel'
N 3
bash-3.2#
bash-3.2# clrg switch -n podio3 RG5
bash-3.2# cd /opt/SUNWscxvm/util
bash-3.2#
bash-3.2# vi xvm_config
```

(Amend xvm_config with the following)

```
RS=domu4C-rs
RG=RG5
HAS_RS=
DOMAIN=domu4C
PLUGIN_PROBE="/opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzodio2b:runlevel:3"
FAILOVER="migrate --live"
ADMIN=/global/bootdisk/haxvm-admin
```

Test the PLUGIN_PROBE, note that 27 is passed by the HA-xVM Server x64-64 agent and represents 90% of PROBE_TIMEOUT (Default=30).

```
bash-3.2# /opt/SUNWscxvm/bin/ppkssh -P fmuser:/export/fmuser/.ssh/id_dsa:vzodio2b:runlevel:3 27
vzodio2b runlevel is 3
bash-3.2#
bash-3.2# ./xvm_register
```

HA-xVM Server x86-64 agent cheat sheet

26. Ensure all OHAC resources are online.

```
bash-3.2# clrs status domu1-rs domu2-rs domu3-rs domu4-rs domu4C-rs
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
domu1-rs	podio1	Online	Online - Service is online.
	podio2	Offline	Offline
	podio3	Offline	Offline
domu2-rs	podio1	Offline	Offline
	podio2	Online	Online - Service is online.
	podio3	Offline	Offline
domu3-rs	podio1	Offline	Offline
	podio2	Offline	Offline
	podio3	Online	Online - Service is online.
domu4-rs	podio1	Offline	Offline
	podio2	Offline	Offline
	podio3	Online	Online - Service is online.
domu4C-rs	podio1	Offline	Offline
	podio2	Offline	Offline
	podio3	Online	Online - Service is online.

```
bash-3.2#
```

HA-xVM Server x86-64 agent cheat sheet

27. Turning debug on for the HA-xVM Server x86-64 agent.

Perform this step on all cluster nodes.

```
bash-3.2# vi /etc/syslog.conf
```

```
change  
*.err;kern.debug;daemon.notice;mail.crit /var/adm/messages  
to  
*.err;kern.debug;daemon.debug;mail.crit /var/adm/messages
```

```
bash-3.2# svcadm refresh system-log && svcadm restart system-log
```

```
bash-3.2#
```

```
bash-3.2# vi /opt/SUNWscxvm/etc/config
```

```
change  
DEBUG=  
to  
DEBUG=ALL
```

Debug messages should now appear in the syslog.

```
bash-3.2# tail -f /var/adm/messages
```

Reverse the steps above to turn off debug.