Thomas Cameron, RHCE, RHCX, CNE, MCSE, MCT
Solutions Architect Team Lead
Lon Hohberger
Cluster Software Engineer
Dedicated to the Memory of
Jim Webb, Global Professional Services Consultant
Cluster Failover For Any Application

- This session will present the installation and configuration of a high availability cluster of virtual machines using Red Hat Enterprise Linux Advanced Platform (RHEL AP).
- Demonstrations will include:
  - setting up an iSCSI target (server)
  - setting up iSCSI initiators (clients)
  - Red Hat Global Filesystem and Red Hat Cluster Suite functionality included in RHEL AP
  - the Conga web-based cluster management tool
Cluster Failover For Any Application (cont.)

- This presentation will include:
  - setting up clustered logical volumes
  - deploying and clustering virtual machines
  - managing applications on that cluster
- This presentation is available online at http://people.redhat.com/tcameron
All hosts are in the fictional “redhat.lan” DNS domain.

Wintermute, Case and Molly are physical servers.

Case and Molly are iSCSI initiators (clients) and they connect to Wintermute, an iSCSI target (server)
Case and Molly make up the dom0-cluster.

Armitage and Lady3Jane are domU guests on Molly.

TheFinn and Hideo are domU guests on Case.

Armitage, Lady3Jane, TheFinn and Hideo make up the Apache cluster.
The apache-cluster is a cluster of web servers. These servers run on the four domU machines.
Setting up the dom0 cluster

- Ensure that the Clustering and Cluster Storage groups are installed
- `yum grouplist`
  
  Installed Groups: Cluster Storage Clustering
Setting up the dom0 cluster

- Install luci on a machine which will not be a part of the cluster. For this presentation, wintermute is not actually part of the cluster.
- `yum install luci`
- `luci_admin init` and set administrative password
- `chkconfig luci on`
- `service luci restart`
Setting up the dom0 cluster

[root@wintermute ~]# luci_admin init
Initializing the luci server

Creating the 'admin' user

Enter password:
Confirm password:

Please wait...
The admin password has been successfully set.
Generating SSL certificates...
The luci server has been successfully initialized

You must restart the luci server for changes to take effect.

Run "service luci restart" to do so

[root@wintermute ~]# service luci restart
Shutting down luci: [ OK ]
Starting luci: Generating https SSL certificates... done [ OK ]

Point your web browser to https://wintermute.redhat.lan:8084 to access luci

[root@wintermute ~]#
Setting up the dom0 cluster

- Go to the URL mentioned by when restarting the luci service
- In this case, https://wintermute.redhat.lan:8084
- Note that you'll have to add an exception in Firefox since it is a self-signed certificate
Luci Homebase

Welcome to Luci, admin.

Select an action from the list on the left.
Setting up the dom0 cluster

- Click on the Cluster tab, then on the “Create a new cluster” link
- In this example, the dom0 cluster is called “dom0-cluster.” Not imaginative, but effective.
- The two members are case and molly.
- Choose the “Download packages” radio button – the nodes will download the necessary packages from RHN
- Verify that “Enable Shared Storage Support” is checked
- Verify that “Reboot nodes before joining cluster” is checked to make sure that the systems join cleanly at boot time
- “Check if node passwords are identical” if they are
Create a new cluster

Cluster Name: don6-cluster

<table>
<thead>
<tr>
<th>Node Hostname</th>
<th>Root Password</th>
<th>Key ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>case.redhat.lan</td>
<td>**********</td>
<td></td>
</tr>
<tr>
<td>molly.redhat.lan</td>
<td>**********</td>
<td></td>
</tr>
</tbody>
</table>

Add a cluster node

- Download packages
- Use locally installed packages.
- Enable Shared Storage Support
- Reboot nodes before joining cluster
- Check if node passwords are identical.

View SSL cert fingerprints

Submit
Setting up the dom0 cluster

- Click on Submit to install and reboot
dom0-cluster

Please be patient - this cluster's configuration is being modified.

Creating node "case.redhat.lan" for cluster "dom0-cluster"
Currently unable to reach the roci agent on this node. The install state is not yet complete.
Install Reboot Configure Join

Stop waiting for this job to complete

Creating node "molly.redhat.lan" for cluster "dom0-cluster"
Currently unable to reach the roci agent on this node. The install state is not yet complete.
Install Reboot Configure Join

Stop waiting for this job to complete
Setting up the dom0 cluster

- When the systems come back up the cluster and the nodes should all show up in green
Choose a cluster to administer

Cluster Name: dom0-cluster

- Status: Quorate
- Total Cluster Votes: 2
- Minimum Required Quorum: 1

Nodes

- case.redhat.lan
- molly.redhat.lan

Services

- No Services Defined

The Conga Cluster and Storage Management System is Copyright © 2000–2008 Red Hat, Inc.
Distributed under the GNU GPL license.
Setting up the dom0 cluster

- Enable fence_xvmd (dom0 fencing) by adding `<fence_xvmd/>` to the `<cluster>` section of `/etc/cluster/cluster.conf`
- Reboot the systems and check that fence_xvmd is in the process table
Setting up the iSCSI Target

- RHEL 5 AP with
  - @ Clustering
  - @ Cluster Storage
  - scsi-target-utils
Setting up the iSCSI Target

- In this case, wintermute will be the target
- First, set up partition(s) to export
- /sbin/fdisk to create partitions
  - For this demonstration, three 9GB partitions were created, /dev/hda5, /dev/hda6 and /dev/hda7
Setting up the iSCSI Target

```
[root@wintermute ~]# fdisk -l

Disk /dev/hda: 40.0 GB, 40007761920 bytes
255 heads, 63 sectors/track, 4064 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Filesystem Boot Start  End   Blocks  Id  System
/dev/hda1   *       1    13 104391 83  Linux
/dev/hda2    14 1288 10241437+  83  Linux
/dev/hda3   1289 1415 1020127+  82  Linux swap / Solaris
/dev/hda4   1416 4864 27704392+  5  Extended
/dev/hda5   1416 2510 8795556  83  Linux
/dev/hda6   2511 3605 8795556  83  Linux
/dev/hda7   3606 4700 8795556  83  Linux
[root@wintermute ~]#
```
Setting up the iSCSI Target

- Verify that `scsi-target-utils` is installed
- Enable the `tgtd` service with your favorite service management tool
Setting up the iSCSI Target

```
[root@wintermute ~]# yum list scsi-target-utils
Loading "security" plugin
Loading "rhnplugin" plugin
rhel-i386-server-cluster- 100% |-----------------------------| 1.4 kB 00:00
rhel-i386-server-producti 100% |-----------------------------| 1.2 kB 00:00
rhel-i386-server-fastrack 100% |-----------------------------| 1.2 kB 00:00
rhel-i386-server-vt-5 100% |-----------------------------| 1.4 kB 00:00
rhel-i386-server-5 100% |-----------------------------| 1.4 kB 00:00
rhel-i386-server-cluster- 100% |-----------------------------| 1.4 kB 00:00
rhel-i386-server-suppleme 100% |-----------------------------| 1.2 kB 00:00
rhn-tools-rhel-i386-serve 100% |-----------------------------| 1.2 kB 00:00

Installed Packages
scsi-target-utils.i386 0.0-0.20070626snap.el5 installed
[root@wintermute ~]# chkconfig tgtd on
[root@wintermute ~]# service tgtd restart
Stopping SCSI target daemon:
Starting SCSI target daemon: [ OK ]
[root@wintermute ~]#]
```
Setting up the iSCSI Target

- Use `tgtadm` to define the iSCSI target qualified names
- `tgtadm --lld iscsi --op new --mode target \ --tid 1 -T iqn.2008-06.lan.redhat.disk1`
- `/usr/sbin/tgtadm` is the SCSI target admin tool
- `--lld iscsi` defines the low level driver as iSCSI
- `--op new` defines an operation to create a new definition
- `--mode target` defines that definition as a target
- `--tid 1` defines the target identifier as “1”
- `-T iqn.2008-06.lan.redhat.disk1` defines the iSCSI qualified name with the date (year and month), reversed hostname and the disk label
Setting up the iSCSI Target

- Use `tgtadm` to add the partitions created earlier as targets
- `tgtadm --lld iscsi --op new --mode \ logicalunit --tid 1 --lun=1 -b /dev/hda5`
- `--mode localunit` defines the new device as a LUN
- `--tid 1` matches to the target identifier previously defined
- `--lun=1` defines this as the first logical unit (LUN)
- `-b /dev/sdb2` associates that LUN to the block device `/dev/hda5`
- Repeat for each of the partitions created earlier
Setting up the iSCSI Target

- Use `tgtadm` to allow clients to attach to the targets
- `tgtadm --lld iscsi --op bind --mode target --tid 1 -I 172.31.100.3`
- Or
- `tgtadm --lld iscsi --op bind --mode target --tid 1 -I ALL`
  - `--op bind` defines a bind operation
  - `-I [addr]` or `-I ALL` allows a specific host or everyone to connect
Setting up the iSCSI Target

[root@wintermute ~]# tgtadm --lld iscsi --op new --mode target --tid 1 -T iqn.2008-06.lan.redhat.disk1
[root@wintermute ~]# tgtadm --lld iscsi --op new --mode target --tid 2 -T iqn.2008-06.lan.redhat.disk2
[root@wintermute ~]# tgtadm --lld iscsi --op new --mode target --tid 3 -T iqn.2008-06.lan.redhat.disk3
[root@wintermute ~]# tgtadm --lld iscsi --op new --mode logicalunit --tid 1 -lun 1 -b /dev/hda5
[root@wintermute ~]# tgtadm --lld iscsi --op new --mode logicalunit --tid 2 -lun 2 -b /dev/hda6
[root@wintermute ~]# tgtadm --lld iscsi --op new --mode logicalunit --tid 3 -lun 3 -b /dev/hda7
[root@wintermute ~]# tgtadm --lld iscsi --op bind --mode target --tid 1 -I ALL
[root@wintermute ~]# tgtadm --lld iscsi --op bind --mode target --tid 2 -I ALL
[root@wintermute ~]# tgtadm --lld iscsi --op bind --mode target --tid 3 -I ALL
[root@wintermute ~]#
Setting up the iSCSI Target

- Use tgtadm to verify that you are exporting all targets correctly:
- `tgtadm --lld iscsi --op show --mode target`
Setting up the iSCSI Target

```
[root@wintermute ~]# tgtadm --lld iscsi --op show --mode target
Target 1: ign.2008-06.lan.redhat.disk1
    System information:
        Driver: iscsi
        Status: running
    I_T nexus information:
    LUN information:
        LUN: 0
            Type: controller
            SCSI ID: deadbeef1:0
            SCSI SN: beaf10
            Size: 0
            Backing store: No backing store
        LUN: 1
            Type: disk
            SCSI ID: deadbeef1:1
            SCSI SN: beaf11
            Size: 8G
            Backing store: /dev/hda5
    Account information:
    ACL information:
        ALL
Target 2: ign.2008-06.lan.redhat.disk2
    System information:
```
Setting up the iSCSI Target

- To make it persistent, either create a custom startup script or just add the lines to `/etc/rc.d/rc.local`
Setting up the iSCSI Target

```bash
[root@wintermute ~]# cat /etc/rc.d/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

tgtadm --lld iscsi --op delete --mode target --tid=1
tgtadm --lld iscsi --op delete --mode target --tid=2
tgtadm --lld iscsi --op delete --mode target --tid=3
tgtadm --lld iscsi --op new --mode target --tid 1 -T iqn.2008-06.lan.redhat.disk1
tgtadm --lld iscsi --op new --mode target --tid 2 -T iqn.2008-06.lan.redhat.disk2
tgtadm --lld iscsi --op new --mode target --tid 3 -T iqn.2008-06.lan.redhat.disk3
tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 1 -b /dev/hda5
tgtadm --lld iscsi --op new --mode logicalunit --tid 2 --lun 2 -b /dev/hda6
tgtadm --lld iscsi --op new --mode logicalunit --tid 3 --lun 3 -b /dev/hda7
tgtadm --lld iscsi --op bind --mode target --tid 1 -I ALL
tgtadm --lld iscsi --op bind --mode target --tid 2 -I ALL
tgtadm --lld iscsi --op bind --mode target --tid 3 -I ALL
tgtadm --lld iscsi --op show --mode target | grep Target

touch /var/lock/subsys/local
[root@wintermute ~]#
```
Setting up the iSCSI Initiator

- Use the `iscsiadm` utility to determine what the target is exporting:

```
iscsiadm -m discovery -t sendtargets -p [host]:3260
```

- `-m discovery` tells iscsiadm to discover new information
- `-t sendtargets` says that information should be target definitions
- `-p [host]:3260` is the host and optionally the port number to discover
Setting up the iSCSI Initiator

```
[root@molly ~]# iscsiadm -m discovery -t sendtargets -p 172.31.100.9
172.31.100.9:3260,1 iqn.2008-06.lan.redhat.disk1
172.31.100.9:3260,1 iqn.2008-06.lan.redhat.disk2
172.31.100.9:3260,1 iqn.2008-06.lan.redhat.disk3
[root@molly ~]#
```
Setting up the iSCSI Initiator

- Use the `iscsadm` utility to log into the target
  - `iscsiadm -m node -T \iqn.2008-06.lan.redhat.disk1 \-p 172.31.100.9 -l`
- `-l` logs the initiator into the target
Setting up the iSCSI Initiator

```
[root@molly ~]# iscsiadm -m node -T iqn.2008-06.lan.redhat.disk1 -p 172.31.100.9 -l
Logging in to [iface: default, target: iqn.2008-06.lan.redhat.disk1, portal: 172.31.100.9,3260]
Login to [iface: default, target: iqn.2008-06.lan.redhat.disk1, portal: 172.31.100.9,3260]: successful
[root@molly ~]# iscsiadm -m node -T iqn.2008-06.lan.redhat.disk2 -p 172.31.100.9 -l
Logging in to [iface: default, target: iqn.2008-06.lan.redhat.disk2, portal: 172.31.100.9,3260]
Login to [iface: default, target: iqn.2008-06.lan.redhat.disk2, portal: 172.31.100.9,3260]: successful
[root@molly ~]# iscsiadm -m node -T iqn.2008-06.lan.redhat.disk3 -p 172.31.100.9 -l
Logging in to [iface: default, target: iqn.2008-06.lan.redhat.disk3, portal: 172.31.100.9,3260]
Login to [iface: default, target: iqn.2008-06.lan.redhat.disk3, portal: 172.31.100.9,3260]: successful
[root@molly ~]#
```
Setting up the iSCSI Initiator

- Verify that the LUNs are available to the initiator
- `fdisk -l`
Setting up the iSCSI Initiator

```
[root@molly ~]# fdisk -l

Disk /dev/sda: 100.0 GB, 10003024288 bytes
255 heads, 63 sectors/track, 12161 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

  Device Boot Start   End   Blocks   Id  System
/dev/sda1 *       1     13   104391   83  Linux
/dev/sda2     14   2563  20482875  83  Linux
/dev/sda3   2564  2694  1052257+ 82  Linux swap / Solaris

Disk /dev/sdb: 9606 MB, 9006649344 bytes
64 heads, 32 sectors/track, 8589 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdb doesn't contain a valid partition table

Disk /dev/sdc: 9606 MB, 9006649344 bytes
64 heads, 32 sectors/track, 8589 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdc doesn't contain a valid partition table

Disk /dev/sdd: 9606 MB, 9006649344 bytes
64 heads, 32 sectors/track, 8589 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdd doesn't contain a valid partition table

[root@molly ~]#
```
Setting up clustered logical volumes

- Turn on clustered logical volume management with the `/usr/sbin/lvmconf` utility:
  - `lvmconf --enable-cluster`
  - Changes `/etc/lvm/lvm.conf` so that the `locking_type` entry is changed to “3” (built-in cluster-wide locking)
- Ensure that `clvmd` is running and will survive a reboot
- Create partitions using `fdisk`
- Create physical volumes using (for instance) on one initiator:
  - `pvcreate /dev/sdb1 /dev/sdc1 /dev/sdd1`
Setting up clustered logical volumes

```
[root@case ~]# lvmconf --enable-cluster
[root@case ~]# chkconfig clvmd on
[root@case ~]# service clvmd restart
Stopping clvm: [FAILED]
Starting clvmd: [ OK ]
Activating VGs: [ OK ]

[root@case ~]# pvcreate /dev/sdb1 /dev/sdc1 /dev/sdd1
  Physical volume "/dev/sdb1" successfully created
  Physical volume "/dev/sdc1" successfully created
  Physical volume "/dev/sdd1" successfully created
[root@case ~]#
```
Setting up clustered logical volumes

- Run `pvdisplay /dev/sdb1 /dev/sdc1 /dev/scd1` on the other initiator(s) in the cluster to verify that they see the physical volumes.
Setting up clustered logical volumes

```
[root@molly ~]# pvdisplay /dev/sdb1 /dev/sdc1 /dev/sdd1
"/dev/sdb1" is a new physical volume of "8.39 GB"
--- NEW Physical volume ---
PV Name     /dev/sdb1
VG Name     
PV Size     8.39 GB
Allocatable NO
PE Size (KByte) 0
Total PE   0
Free PE    0
Allocated PE 0
PV UUID    R7Kq0S-H5B2-tpZC-Ydr3-NnaX-mOyY-wpZUsT

"/dev/sdc1" is a new physical volume of "8.39 GB"
--- NEW Physical volume ---
PV Name     /dev/sdc1
VG Name     
PV Size     8.39 GB
Allocatable NO
PE Size (KByte) 0
Total PE   0
Free PE    0
Allocated PE 0
PV UUID    2I87Pd-ueT9-bZWU-x9Kl-545d-pti0-DHC01k
```
Setting up clustered logical volumes

- Create volume groups using `/usr/sbin/vgcreate`
  - `vgcreate XenVol /dev/sdb1 /dev/sdc1 /dev/sdd1`
- Determine how many free physical extents available using `/usr/sbin/vgdisplay`
Setting up clustered logical volumes

```
[root@case ~]# vgdisplay
   --- Volume group ---
   VG Name          XenVol
   System ID        lvm2
   Format           lvm2
   Metadata Areas   3
   Metadata Sequence No 1
   VG Access        read/write
   VG Status        resizable
   Clustered        yes
   Shared           no
   MAX LV           0
   Cur LV           0
   Open LV          0
   Max PV           0
   Cur PV           3
   Act PV           3
   VG Size          25.16 GB
   PE Size          4.00 MB
   Total PE         6441
   Alloc PE / Size  0 / 0
   Free PE / Size   6441 / 25.16 GB
   VG UUID          K7wokx-l0mg-r7MU-quPG-XWlr-dhCp-pa02am
```
Setting up clustered logical volumes

- Create logical volume using `/usr/sbin/lvcreate`
  - `lvcreate -n XenLV -l 6441 XenVol`
Setting up clustered logical volumes

[root@case ~]# lvcreate -n XenLV -l 6441 XenVol
Logical volume "XenLV" created
[root@case ~]#
Setting up clustered logical volumes

- Verify that the other node can “see” the logical volume
  - `/usr/sbin/lvdisplay`
Setting up clustered logical volumes

```
[root@molly ~]# lvdisplay
--- Logical volume ---
LV Name /dev/XenVol/XenLV
VG Name XenVol
LV UUID Mbdw6e-PUsh-Bul6-CYpZ-2rQ1-Mb6n-e03PgP
LV Write Access read/write
LV Status available
# open 0
LV Size 25.16 GB
Current LE 6441
Segments 3
Allocation inherit
Read ahead sectors auto
- currently set to 256
Block device 253:0
```

Setting up clustered logical volumes

- Set up the shared GFS version 2 filesystem using `/sbin/mkfs.gfs2`
  - `mkfs.gfs2 -j 3 -p lock_dlm -t \dom0-cluster:gfs /dev/XenVol/XenLV`
  - `mkfs.gfs2` creates the gfs2 filesystem
  - `-j 3` defines the number of journals – in this case, one for each node plus one extra in case the cluster is expanded
  - `-p lock_dlm` defines the locking protocol. In this case, distributed lock manager
  - `-t dom0-cluster:gfs` specifies the lock table name, which is the cluster name and the shared filesystem name
  - `/dev/XenVol/XenLV` is the device to be formatted
Setting up clustered logical volumes

[root@case ~]# mkfs.gfs2 -j 3 -p lock_dlm -t dom0-cluster:gfs /dev/XenVol/XenLV
This will destroy any data on /dev/XenVol/XenLV.
Are you sure you want to proceed? [y/n] y

Device: /dev/XenVol/XenLV
Blocksize: 4096
Device Size: 25.16 GB (6595584 blocks)
Filesystem Size: 25.16 GB (6595583 blocks)
Journals: 3
Resource Groups: 101
Locking Protocol: "lock_dlm"
Lock Table: "dom0-cluster:gfs"

[root@case ~]#
Setting up clustered logical volumes

- Mount the newly created GFS filesystem. In this case on /var/lib/xen/images for both dom0 machines
  - `mount /dev/XenVol/XenLV /var/lib/xen/images`
- You can check /var/log/messages to see that the shared storage was mounted correctly
Setting up clustered logical volumes

```
[root@case ~]# mount /dev/XenVol/XenLV /var/lib/xen/images/
[root@case ~]# tail /var/log/messages
Jun 2 18:29:35 case clvmd: Cluster LVM daemon operation was successful.
Jun 2 20:01:50 case kernel: dlm: Using TCP connection to connect to daemon...
Jun 2 20:01:51 case clvmd: Cluster LVM daemon operation was successful.
Jun 2 20:18:45 case kernel: GFS 0.1.23-5.el4: 
Jun 2 20:18:45 case kernel: Trying to join cluster "case".
Jun 2 20:18:45 case kernel: Joined cluster. Now mounting FS...
Jun 3 00:19:14 molly kernel: Joined cluster. Now mounting FS...
```

```
[root@molly ~]# mount /dev/XenVol/XenLV /var/lib/xen/images/
[root@molly ~]# tail /var/log/messages
Jun 2 20:18:45 molly kernel: Joined cluster. Now mounting FS...
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=0: Trying to acquire journal lock...
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=0: Looking at journal...
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=0: Done
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=1: Trying to acquire journal lock...
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=1: Looking at journal...
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=1: Done
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=2: Trying to acquire journal lock...
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=2: Looking at journal...
Jun 3 00:19:14 molly kernel: GFS: fsid=dom0-cluster:gfs.0: jid=2: Done
```

```
[root@case ~]#
[root@molly ~]#```
Setting up clustered logical volumes

- Add the mount point to /etc/fstab on both hosts
Setting up clustered logical volumes

```
[root@case ~]# cat /etc/fstab
LABEL=/       /                     ext3  defaults  1   1
LABEL=/boot1  /boot                 ext3  defaults  1   2
tmpfs         /dev/shm               tmpfs defaults  0   0
devpts        /dev/pts               devpts gid=5,mode=620 0   0
sysfs         /sys                    sysfs defaults  0   0
proc          /proc                   proc  defaults  0   0
LABEL=SWAP-sda3 swap                 swap  defaults  0   0
/dev/mapper/XenVol-XenLV /var/lib/xen/images gfs2  rw,noatime  0   0
[root@case ~]#
```
Installing the domU machines

- Use your favorite installer (/usr/sbin/virt-manager if you're a Windows™ admin or /usr/sbin/virt-install if you're a real sysadmin)
- `virt-install -n armitage -r 384 \ -f /var/lib/xen/images/armitage.disk \ -s 3 --nonsparse -m 00:16:3e:30:9c:e3 \ --nographics --os-type linux --os-variant \ rhel5 --paravirt --location \ http://wintermute.redhat.lan/rhel5 \`
Installing the domU machines

-x "ip 172.31.100.10 netmask 255.255.255.0 \
gateway 172.31.100.1 dns 172.31.100.9 \
noipv6 text \\
ks=http://wintermute.redhat.lan/ks.cfg"

• Make sure that the Clustering and Cluster Storage groups are part of your installation
Installing the domU machines

```
[root@case ~]# virt-install -n armitage -r 384 -f /var/lib/xen/images/armitage.disk -s 3 --nonsparse -m 00:16:3e:30:9c:e3 --nographics --os-type linux --os-variant rhel5 --paravirt --location http://winternetuate.redhat.lan/rhel5 -x "ip 172.31.100.10 netmask 255.255.255.0 gateway 172.31.100.1 dns 172.31.100.9 noipv6 text ks=http://winternetuate.redhat.lan/ks.cfg"

Starting install...
Retrieving file .treeinfo 100% |---------------------------| 436 B 00:00
Retrieving file vmlinuz.. 100% |---------------------------| 2.1 MB 00:00
Retrieving file initrd.im 100% |---------------------------| 5.4 MB 00:00
Creating storage file... 14% |==| 454 MB 00:07 ETA |
```
Installing the domU machines

• Verify that all hosts are up
[tcameron@case ~]$ ssh root@armitage
The authenticity of host 'armitage (172.31.100.18)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'armitage,172.31.100.18' (RSA) to the list of known hosts.
root@armitage's password:
Last login: Thu Jun 5 16:35:26 2008
[root@armitage ~]

[root@hideo ~]$ ssh -Y root@hideo
The authenticity of host 'hideo (172.31.100.12)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hideo,172.31.100.12' (RSA) to the list of known hosts.
root@hideo's password:
[root@hideo ~]

[root@lady3jane ~]$ ssh -Y root@lady3jane
The authenticity of host 'lady3jane (172.31.100.11)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'lady3jane,172.31.100.11' (RSA) to the list of known hosts.
root@lady3jane's password:
[root@lady3jane ~]

[root@thefinn ~]$ ssh -Y root@thefinn
The authenticity of host 'thefinn (172.31.100.13)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'thefinn,172.31.100.13' (RSA) to the list of known hosts.
root@thefinn's password:
[root@thefinn ~]
Installing the domU machines

- Make sure that ricci is set to start at bootup
- Make sure ricci is running
Installing the domU machines

[root@lady3jane ~]# chkconfig ricci --list
ricci  0:off 1:off 2:on  3:on  4:on  5:on  6:off
[root@lady3jane ~]# service ricci start
Starting oddjobd: [ OK ]
generating SSL certificates... done
Starting ricci: [ OK ]
[root@lady3jane ~]#
Define the domUs as cluster resources

- Since the domU cluster will be a clustered service, disable the per-dom0 xendomains service
  - `chkconfig xendomains off`
  - `service xendomains stop`
- Move the config files for each domU from `/etc/xen` to `/var/lib/xen/images`
Create the domU Cluster

```
[root@case ~]# ls /var/lib/xen/images/
armitage  hideo  lady3jane  thefinn
armitage.disk  hideo.disk  lady3jane.disk  thefinn.disk
[root@case ~]#
```
Define the domUs as cluster resources

- Define each dom0 as a failover domain
  - prefer_case
  - prefer_molly
- Choose the dom0 cluster, go to “Failover Domains,” then “Add a Failover Domain,” then create the domains
- Set prioritization so that the preferred machine has a lower number
## dom0-cluster

**Failover Domain Form**

<table>
<thead>
<tr>
<th>Failover Domain Name</th>
<th>prefer_case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prioritized</td>
<td>✓</td>
</tr>
<tr>
<td>Restrict failover to this domain's members</td>
<td>□</td>
</tr>
<tr>
<td>Do not fail back services in this domain</td>
<td>□</td>
</tr>
</tbody>
</table>

**Failover domain membership**

<table>
<thead>
<tr>
<th>Node</th>
<th>Member</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>case.redhat.lan</td>
<td>✓</td>
<td>1</td>
</tr>
<tr>
<td>molly.redhat.lan</td>
<td>✓</td>
<td>10</td>
</tr>
</tbody>
</table>

Submit
dom0-cluster

Failover Domain Name: prefer_molly

This Failover Domain is Ordered
This Failover Domain is Unrestricted

Members of this Failover Domain

- case.redhat.lan
- molly.redhat.lan

Services employing this Failover Domain:
- No Services Defined

Failover Domain Name: prefer_case

This Failover Domain is Ordered
This Failover Domain is Unrestricted

Members of this Failover Domain

- case.redhat.lan
- molly.redhat.lan

Services employing this Failover Domain:
- No Services Defined
Create the domU Cluster

- Enable live migration by setting the following three settings in `/etc/xen/xend-config.sxp`:
  - (xend-relocation-server yes)
  - (xend-relocation-port 8002)
  - (xend-relocation-address '')
- The '' in xend-relocation-address is two single quotes, not a double quote
- Be aware of the security implications of having this wide open, you can also specify a space separated list of regular expressions for hostnames or ip addresses
- Restart the xend service
Create the domU Cluster

• Test live migration.
• From case, issue the command:
  • `xm migrate armitage molly`
Create the domU Cluster

```
[root@case ~]# virsh list
Id Name     State
---------   ------
         0 Domain-0 running
          3 lady3jane blocked
           5 armitage blocked

[root@case ~]# xm migrate armitage molly
[root@case ~]# virsh list
Id Name     State
---------   ------
         0 Domain-0 running
          3 lady3jane blocked
```

Create the domU Cluster

```
[root@molly ~]# virsh list
Id  Name       State
----+-----------+---------
  0  Domain-0  running
  3  thefinn  blocked
  4  hideo    blocked
  6  armitage  blocked

[root@molly ~]
```
Define the domUs as cluster resources

- Define the domUs as resources for the dom0 to which they are assigned
- Choose the cluster, then “Services,” then “Add a Virtual Service.”
  - Name of domU
  - Path to config
  - Automatically start
  - Failover domain
  - Recovery policy
  - Migration type
- Do this for all domUs
Create a Virtual Machine Service

Virtual machine name: armitage
Path to VM configuration files: /var/lib/xen/images
Automatically start this service: [ ]
Run exclusive: [ ]
Follower Domain: prefer_case
Recovery policy: Relocate
Migration type: Live

Create Virtual Machine Service
Define the domUs as cluster resources

- Test that Conga will migrate a domU to a different dom0
dom0-cluster

Service Name armitage (virtual service)
Status: Running on case.redhat-lan
Autostart is enabled for this service

Failover Domain Association: prefer_case

Service Name lady3jane (virtual service)
Status: Running on case.redhat-lan
Autostart is enabled for this service

Failover Domain Association: prefer_case

Service Name hideo (virtual service)
Status: Running on molly.redhat-lan
Autostart is enabled for this service

Failover Domain Association: prefer_molly

Service Name thefin (virtual service)
Status: Running on molly.redhat-lan
Autostart is enabled for this service

Failover Domain Association: prefer_molly
dom0-cluster

Service Name lady3jane (virtual service)
Status: Running on molly.redhat.lan
Autostart is enabled for this service
Failover Domain Association: prefer_molly

Service Name armitage (virtual service)
Status: Running on molly.redhat.lan
Autostart is enabled for this service
Failover Domain Association: prefer_molly

Service Name hideo (virtual service)
Status: Running on molly.redhat.lan
Autostart is enabled for this service
Failover Domain Association: prefer_molly

Service Name thefinn (virtual service)
Status: Running on molly.redhat.lan
Autostart is enabled for this service
Failover Domain Association: prefer_molly
Set up the Apache cluster

• Using similar steps as the dom0 cluster, set up a cluster of domUs
• Set up a prioritized failover domain which sets the service to prefer armitage, then hideo, then lady3jane, then thefinn
Create a new cluster

<table>
<thead>
<tr>
<th>Cluster Name</th>
<th>Node Hostname</th>
<th>Root Password</th>
<th>Key ID</th>
<th>Enable Shared Storage Support</th>
<th>Reboot nodes before joining cluster</th>
<th>Check if node passwords are identical</th>
</tr>
</thead>
<tbody>
<tr>
<td>apache-cluster</td>
<td>armitage.redhat.lan</td>
<td>**********</td>
<td>![Key ID]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>lady3.jane.redhat.lan</td>
<td>**********</td>
<td>![Key ID]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hides.redhat.lan</td>
<td>**********</td>
<td>![Key ID]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>thefinn.redhat.lan</td>
<td>**********</td>
<td>![Key ID]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add a cluster node:

- Download packages
- Use locally installed packages.

View SSL cert fingerprints

Submit
### Choose a cluster to administer

**Cluster Name: dom0-cluster**

- **Status:** Quorate
- **Total Cluster Votes:** 2
- **Minimum Required Quorum:** 1

**Nodes**
- ciao.redhat.lan
- molly.redhat.lan

**Services**
- armitage
- ladyjane
- iodeo
- thelinn

### Cluster Name: apache-cluster

- **Status:** Quorate
- **Total Cluster Votes:** 4
- **Minimum Required Quorum:** 3

**Nodes**
- armitage.redhat.lan
- hidee.redhat.lan
- ladyjane.redhat.lan
- thelinn.redhat.lan

**Services**
- No Services Defined
Add a Failover Domain

**Failover Domain Name**
prefer_armitage

- **Prioritized**: Yes
- **Restrict failover to this domain's members**: No
- **Do not fail back services in this domain**: No

**Failover domain membership**

<table>
<thead>
<tr>
<th>Node</th>
<th>Member</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>lady3jane.redhat.lan</td>
<td>✔️</td>
<td>20</td>
</tr>
<tr>
<td>hidden.redhat.lan</td>
<td>✔️</td>
<td>10</td>
</tr>
<tr>
<td>armitage.redhat.lan</td>
<td>✔️</td>
<td>1</td>
</tr>
<tr>
<td>thefin.redhat.lan</td>
<td>✔️</td>
<td>30</td>
</tr>
</tbody>
</table>

Submit
Set up the Apache cluster

- Define the following resources in apache-cluster
  - A script resource
    - `/etc/rc.d/init.d/httpd`
  - An ip address resource
    - `172.31.100.20`
apache-cluster

Add a Resource

IP Address Resource Configuration

IP address 172.31.100.20
Monitor link
Submit
apache-cluster

Add a Resource

Script Resource Configuration

Name: webserver
Full path to script file: /etc/rc.d/init.d/http

Submit
Set up the Apache cluster

- Assign those resources to a service
  - web-cluster

- The script resource should be a child of the ip address resource, as the ip address needs to be up before Apache will bind to it
apache-cluster

Service Name web-cluster

Status: Running on armitage.redhat.lan
AutoStart is enabled for this service

Failover Domain Association: prefer_armitage
Set up the Apache cluster

- To test, you can do something simple like
  - `hostname > /var/www/html/index.html`
Test the Apache cluster

• Fail the web-cluster service to another domU
apache-cluster

Service Name: **web-cluster**

**Status:** Running on armitage.redhat.lan
AutoStart is enabled for this service

**Failover Domain Association:** prefer_armitage
Test the Apache cluster

- Or you can “kick the plug” out of the domU
  - `virsh destroy hideo`
  - `echo s > /proc/sysrq-trigger ; echo b > /proc/sysrq-trigger`
Questions?
This presentation is available online at http://people.redhat.com/tcameron

- You can contact Thomas Cameron at tcameron@redhat.com
- You can contact Lon Hohberger at lhh@redhat.com

- Thank you very much for attending!
HEADLINE | LIBERATION SANS BOLD | Size 28

• Bullet 1 | Liberation Sans | Size 24
• Bullet 2
• Bullet 3
• Bullet 4