

Linuxha.net
Step-by-step Guide:
Using a Xen VM as a Linuxha.net Application

Introduction

The purpose of this document is to describe suggested procedures for creating a Xen virtual machine as a Linuxha.net application – meaning that on physical hardware failure it will automatically fail-over to the other node in the cluster – all without access to storage that is visible to both nodes [i.e. no need for iSCSI or Fibre Channel based storage].

Linuxha.net is quite generic software that should work on most modern Linux distributions; however the table below describes the set-up using in this particular case.

CPU	64-bit Intel Quad Core
Operating System	Centos 5.3
DRBD version	08/03/01
Linuxha.net version	01/04/06
Application version	N/A
Node names	Centos5s1 (primary); centos5s2 (secondary)
Application physical volume (partition)	/dev/hda
Application virtual IP address	192.168.1.77

The following conventions are used throughout this document:

# ls	Single-line command entered as root user.
# useradd \ > --home-dir /home2/xyz \ > --gid abc \ > xyz	Multi-line command entered as root user.
\$ ls	Single-line command entered as non-root user
\$ cut \ > --delimiter=":" \ > --fields=1,5 \ / etc/passwd	Multi-line command entered as non-root user.
127.0.0.1 localhost 192.168.1.32 fc5s1 192.168.1.33 fc5s2 192.168.100.32 fc5s1b 192.168.100.33 fc5s2b	Command output or file contents

Existing Cluster Configuration

The documentation describes the addition of a new application to an existing cluster; documentation covering the build of clusters is readily available elsewhere on the linuxha.net web site.

Prior to implementation of a new application to provide clustering for Xen virtual machine the cluster configuration looked as follows:

```
# clstat
Cluster: centos5 - UP

      Node      Status
centos5s1      UP
centos5s2      UP

Application    Node      State   Started  Monitor  Stale  Fail-over?
  apache      centos5s2  STARTED 0:00:01  Running    0      Yes
  mysql        N/A        DOWN    N/A      N/A      N/A      Yes
  oracle-xe    N/A        DOWN    N/A      N/A      N/A      Yes
  samba        centos5s2  STARTED 0:00:00  Running    0      Yes
```

Initial Storage Configuration

To create a Xen instance a new volume group was created on a 10Gb partition using commands such as the following on both nodes:

```
# pvcreate /dev/hda
# vgcreate virtvg /dev/hda
```

An 8Gb file system was created and mounted as /virt on "centos5s1" using the commands:

```
# lvcreate -L 8g -n lv01 /dev/virtvg
# mkfs -t ext3 /dev/virtvg/lv01
# mkdir /virt
# mount /dev/virtvg/lv01 /virt
```

Initial Xen Software Image Readiness

I had a separate machine mount the an ISO image of Centos as follows:

```
# cd /isos
# mkdir centos5
# mount -o loop CentOS-5.2-x86_64-bin-DVD.iso centos5
```

Now the /isos file system was exported supporting cross-mounts to ensure the loop-back file system could be mounted, so in /etc/exports

```
isos 192.168.0.0/16(rw,crossmnt,no_root_squash)
```

Initial Xen Imaging

The "Virtual Machine Manager" GUI was started, and the Xen domain-0 was connected to. Then a new virtual machine was created - in this instance named "apache" - though of course it does not matter.

On the "Locating Installation Media" dialogue just an entry was given for "Install Media URL" - pointing to the loop-back ISO image just mounted:

```
nfs://192.168.1.104/isos/centos5
```

The image was then created on a file system rather than a disk partition – giving “/virt” as the directory to use – this ensures the Xen machine image gets built on the storage just allocated to “virtvg”.

For this example the configuration of the virtual machine was straightforward – it was allocated just a single virtual disk and a single bridged network interface. Once the image had been installed the virtual machine was shut-down:

```
# xm shutdown apache
```

At this point the configuration for the virtual machine existed as “/etc/xen/apache” and was defined as follows:

```
name = "apache"
uuid = "24bc09da-8dae-061d-0ce3-0dbcb1cb90f3"
maxmem = 512
memory = 512
vcpus = 1
bootloader = "/usr/bin/pygrub"
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
vfb = [ "type=vnc,vncunused=1" ]
disk = [ "tap:aio:/virt/apache.img,xvda,w" ]
vif = [ "mac=00:16:3e:01:13:97,bridge=xenbr4" ]
```

This file should then be copied to the other node in the cluster, for example:

```
# cd /etc/xen
# scp apache centos5s2:$PWD
```

Once final note is that you may need to customise this configuration file on the other node; for example on the host “centos5s2” the “xenbr” device was “xenbr0” rather than “xenbr4”, so on that node the final line of the above configuration file was changed to:

```
vif = [ "mac=00:16:3e:01:13:97,bridge=xenbr0" ]
```

Once the virtual machine was shutdown, but whilst the “/virt” file system was still mounted, the process of defining the new linuxha.net application could begin.

Defining the Application in Linuxha.net

All the commands in this section were carried out on “centos5s1” where the Xen virtual machine had been built, and the “/virt” file system was still mounted.

The first steps were to create a “xapache” directory (the name we wish to give our linuxha.net application since the intention is to use it to host Apache), for example:

```
# cd /etc/cluster
# mkdir xapache
# cp appconf.example.xml xapache/appconf.xml
# cp lems.local.xml.example xapache/lems.local.xml
```

The contents of the “/etc/cluster/xapache/appconf.xml” were edit to the following contents [all

comments removed for clarity].

```
<?xml version="1.0"?>
<appconf>
  <global>
    <version>0.1</version>
    <name>xapache</name>
    <takeover>normal</takeover>
    <syncrate>10000</syncrate>
    <preferred_node>LEAST_CPU_LOAD</preferred_node>
  </global>
  <networks>
    <network net="public" ip="MONITOR"/>
  </networks>
  <vg>
    <name>virtvg</name>
    <type>filesystems</type>
  </vg>
  <application>
    <startscript>/usr/sbin/xm create apache</startscript>
    <stopscript>/usr/sbin/xm shutdown -w apache</stopscript>
    <maxstoptime>600</maxstoptime>
    <maxstarttime>600</maxstarttime>
  </application>
</appconf>
```

Several points to notice about the configuration of the application:

- The network used for the IP address which the Xen application appears is given an IP address of "MONITOR". This is a special setting which indicates to Linuxha.net that the application needs this network to exist to be able to run, but Linuxha.net does not need to apply any IP addresses for the application to it.
- No matter when file systems the Xen image has internally it is presented only as a single file or series of files in a single location - "/virt" in this instance.
- The scripts to start and stop the application are simply calls to "xm" to create and shutdown the virtual machine respectively. Notice that when starting the virtual machine up there is no "wait" option, so the application will appear "STARTED" in Linuxha.net as soon as the boot process starts, rather than when the booting completes. Fortunately this is not the case with the shutdown - the "-w" ensures the virtual machine is stopped before attempting to un-mount file systems.
- The time-outs given to the application start and stop are very large; smaller values will most likely be fine for real hardware.

The other configuration file that is required for a Linuxha.net application to function is the "lems.local.xml" file – a file containing monitoring required by Linuxha.net to ensure the application remains functional. Here the bare minimum configuration is used:

```
<?xml version="1.0"?>
<lems_config>
  <globals modules="/sbin/cluster/lems/modules"
    programs="/sbin/cluster/lems/programs"
    logs="/var/log/cluster/lems"
  />

  <check>
    <name>fsmonitor</name>
    <type>internal</type>
    <module>fsmon xapache</module>
    <interval>10</interval>
    <action_list>
      <action rc="0" action="NOP"/>
      <action rc="1" action="PAUSE 30"/>
      <action rc="2" action="STOP"/>
      <action rc="3" action="FAILOVER"/>
      <action rc="10" action="PAUSE 60"/>
    </action_list>
  </check>
</lems_config>
```

In the above configuration there are just one monitor; for the file system. A more detailed example might include checks to monitor the health of the virtual machine via ssh.

Building the Application

Once the configuration files are in place the usual steps of building the application and then synchronisation of the storage are required. These commands are both run on "centos5s1" where the "/virt" file system is still mounted.

```
# clbuildapp -A xapache
```

If this command fails check the log file and correct any issues, and then repeat until the command runs successfully. Once it has been run then run the application synchronisation:

```
# clbuildapp -A xapache -sync
```

This may take some time depending on the performance of the network and the exact configuration used. In this case a synchronisation rate of 10Mbytest/sec was specified for the application – meaning that approximately 14 minutes after running the command the 8Gb was fully synchronised and the application was ready for use.

Running and Testing the Xen application instance

At this point the application file system and storage will have been un-mounted since all access to it now must be via the replicated storage device that the application has been allocated.

Starting the application on a node is simply the normal Linuxha.net process of running:

```
# clstartapp -A xapache -V
```

Since the start of the virtual machine is asynchronous the start-up should be quite speedy. At any point the application can be stopped, again using the usual command:

```
# clhaltapp -A xapache -V
```

Of course the shutdown is synchronous – and thus will take longer than the start-up.

Future Improvements

This document currently describes the minimum number of steps necessary to get a Xen virtual machine running under Linuxha.net. There are many potential improvements that could be added at a later date, including:

- The start-up command used could be a more detailed script that used ping and ssh to attempt to communicate with the virtual machine, and only when connection was established the script should exit. This would ensure the virtual machine was usable prior to Linuxha.net reporting the application as “usable”.
- A monitor to check the status of the node hosting the virtual machine should be considered. Using the “xm list” command to check the status of the virtual machine, checking memory and swap usage locally too.
- Additional custom monitors could be written for Lems to check the health of the virtual machine – and if not healthy perhaps shutdown and migrate the application? This could be done by attempting to ssh into the virtual machine, check memory, applications etc – quite a bit of work, but a useful refinement for many sites probably.