

Linux-XEN-to-VMWARE

Summary

This document outlines a process for migrating RHEL/CentOS hosts from xen to vmware.

While this procedure is specifically targeted towards VM migrations, a similar plan can be used to migrate physical hosts between machines or for storage migration .

The example hosts we'll use are named **example**(192.168.1.11), **example-clone**(192.168.1.22) and **install**(192.168.1.33)

Overview

We will create a blank vmware vm, boot that into rescue mode.

Then we configure the storage media and copy all of the data from example to example-clone using rsync.

Next we update the system configuration to accomodate changes between xen and vmware.

Last we rebuild the initial ram disk file and install the grub boot loader in the MBR.

Prerequisites

First you will need to collect some information from the source host, example .

- Disk layout info (/etc/fstab, the output of fdisk -l)
- Various host info (#of cpus, amount of memory, amount of swap,ip address)

Next you'll need to create a vmware virtual machine for example-clone. The info from the prior step should help you to size your new vm.

After the vm is created you will want to write down the MAC address of the vm for booting purposes.

When prompted for network interface type, select intel e1000.

Prepare the source host, example

- Enable root login via ssh (/etc/ssh/sshd_config)
- Gather the root password for the system, backup /etc/shadow and change the password if necessary
- Update tcp_wrappers (/etc/hosts.allow) to permit inbound ssh traffic.
- Update host firewall to permit ssh traffic from the install server
- Install the 'normal' kernel on your system, the xen kernel won't run under vmware (yum -y install kernel)
- Disable kudzu if enabled (/sbin/chkconfig kudzu off)
- Update the source system with yum (yum -y update) This step is optional.

Prepare the install server

The install server will need a working PXE environment, including dhcpd and tftpd.

An entry in both dhcpd.conf

```
host example-clone {
    hardware ethernet 00:50:56:bb:00:2a;
    fixed-address 192.168.1.22;
}
```

And a file in the tftp hierarchy to tie the install media and rescue boot options together.

This is where you need to input the mac address from example-clone.

```
$ cat /usr/local/install/tftpboot/rhel5_x86_64/pxelinux.cfg/01-00-50-56-bb-00-2a
label linux
kernel vmlinuz
```

```
append load_ramdisk=1 initrd=initrd.img rescue network text ksdevice=eth0 method=http://192.168.1.33/OS/RHEL5.5_x86_64
```

If you need to configure the dhcpd server to listen on another interface you can do so in **/etc/sysconfig/dhcpd**

```
## default interface
DHCPDARGS=" eth1 "
## if you need to boot via eth0
#DHCPDARGS=" eth0 "
```

If you have a firewall running, configure it to accept dhcp and tftp from example-clone .

```
/sbin/iptables -i eth0 -I RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp \
--dport 68 -s 192.168.1.22 -j ACCEPT

/sbin/iptables -i eth0 -I RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp \
--dport 69 -s 192.168.1.22 -j ACCEPT
```

Migration Process

First, boot your fresh vmware vm, example-clone via 'Power On' .

If all goes well, the install server will hand example-clone the ip address you gave it (192.168.1.22) and will begin booting the rescue media via tftp.

The installer will ask you a few questions, after which you're dropped to a root shell .

Configuring the storage

Partitioning

Use fdisk to create a disk layout on the first scsi disk, /dev/sda .

```
fdisk /dev/sda
```

In this example we'll create a boot partition, and a lvm physical volume to build our ext3 and swap volumes

/boot

n,p,1,1,38

lvm

n,p,2,RETURN,RETURN

set second partition to linux lvm, and quit

t,2,8e,w

LVM and boot volume

Now you'll create lvm volumes and format your filesystems. If you have more partitions than / and /boot you should create and format those luns here. Substitute 30g, and 4g for the respective sizes of your / and swap volumes .

```
mkfs.ext3 -L/boot /dev/sda1
lvm pvcreate /dev/sda2
lvm vgcreate vglocal /dev/sda2
lvm lvcreate -L30g -nroot vglocal
lvm lvcreate -L4g -nswap vglocal
mkswap /dev/vglocal/swap
mkfs.ext3 -L/ /dev/vglocal/root
```

Mount your filesystems.

```
mkdir /mnt/root
mkdir /mnt/boot
mount /dev/vglocal/root /mnt/root
mount /dev/sda1 /mnt/boot
```

Copy the data

We'll use rsync to copy over the system data. If you have any applications or databases running on example, now is a good time to stop them. You'll need the root password for example at this time .

```
rsync -avpor -e ssh root@192.168.1.11:/boot/ /mnt/boot/
rsync -avpor -e ssh --exclude=proc --exclude=sys --exclude=boot "root@192.168.1.11:/*" /mnt/root
```

Now reboot example clone back into rescue mode. We haven't touched the MBR of /dev/sda yet so the system will hopefully boot via PXE

```
# sync;reboot
```

Finishing up

Your example-clone host should now be rebooted in rescue mode. You should permit the installer to search for an existing linux system. If you performed the prior steps correctly the installer should locate and mount your new volumes under **/mnt/sysimage**

Should the installer fail to find your system, try mounting /dev/vglocal/root on /mnt/root . Update **/mnt/root/etc/fstab** as follows below.

Reboot

GRUB

Edit the grub configuration file at **/mnt/sysimage/etc/grub.conf** and change the default kernel from the -xen kernel to the 'normal' kernel we installed earlier on example. If for some reason you only have -xen kernels listed in your configuration, you may have forgotten this step earlier.

Make certain to change the root= line to match the cloned systems root device. Your config should end up looking like this:

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-238.1.1.el5xen)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-238.1.1.el5xen ro root=/dev/vglocal/root
    initrd /initrd-2.6.18-238.1.1.el5xen.img
title Red Hat Enterprise Linux Server (2.6.18-238.1.1.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-238.1.1.el5 ro root=/dev/vglocal/root
    initrd /initrd-2.6.18-238.1.1.el5.img
```

Take note of the kernel version, you'll need that information later .

MODPROBE.CONF

Edit the kernel module configuration file at **/mnt/sysimage/etc/modprobe.conf**, disable the xen network and disk drivers and add an intel e1000 adapter for every interface from the example host .

```
alias eth0 e1000
#alias eth0 xenet
#alias scsi_hostadapter xenblk
```

NETWORK CONFIGURATION

Edit each interface configuration file from `/etc/sysconfig/network-scripts/ifcfg-eth*` and update the ip and mac addresses.

ie: `/etc/sysconfig/network-scripts/ifcfg-eth0`

```
DEVICE=eth0
BOOTPROTO=static
DHCPCLASS=
##OLD HWADDR=00:16:36:64:41:46
##OLD IPADDR=192.168.1.11
HWADDR=00:50:56:BB:00:2A
IPADDR=192.168.1.22
NETMASK=255.255.254.0
ONBOOT=yes
```

FSTAB

The filesystem configuration file, `/etc/fstab` needs to be updated .

Change all references to to match your new lvm partitions:

BEFORE

```
/dev/vgfunky/root      /                ext3    defaults    1 1
LABEL=/boot            /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
/dev/vgfunky/swap      swap            swap    defaults    0 0
```

AFTER

```
/dev/vglocal/root     /                ext3    defaults    1 1
LABEL=/boot           /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
/dev/vglocal/swap     swap            swap    defaults    0 0
```

CHROOT

At this point you have your new system almost ready. Now we just need to install grub and rebuild initrd and we're done.

```
mount --bind /sys /mnt/sysimage/sys
mount --bind /dev /mnt/sysimage/dev
mount --bind /proc /mnt/sysimage/proc
chroot /mnt/sysimage
```

CREATE INITRD

You'll notice that the kernel version here needs to match your new default kernel in `grub.conf`

```
cd /boot
mkinitrd -f initrd-2.6.18-238.1.1.el5.img 2.6.18-238.1.1.el5
```

INSTALL GRUB

```
grub-install --recheck /dev/sda
```

Post Migration

Now you should have two live systems; example and example-clone.

SET CLOCK

The system clock on example-clone will probably be wrong. If ntpdate is not installed you can get it from the ntp package (yum -y install ntp)

```
/sbin/service ntpd stop
/usr/sbin/ntpdate -s address-of-your-ntp-server
/sbin/clock -w
/sbin/service ntpd start
```

RESIZING FILESYSTEMS

If you created /dev/sda with a larger size, now is the time to add or increase the size of your filesystems.

```
/usr/sbin/lvextend -L+1G /dev/vglocal/root
/sbin/resizefs /dev/vglocal/root
```

NETWORKING

Presumably you will want to halt example and then rename, renumber example-clone, but thats your business :)