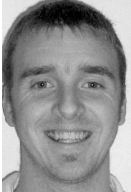


LEIGH GRIFFIN AND JOHN RONAN

Xen installation and configuration



Leigh is a student assistant researcher at the TSSG, Waterford Institute of Technology. He has now returned to WIT to complete his studies in the B.Sc. in Applied Computing Degree.

lgriffin@tssg.org



John Ronan is a senior researcher at the TSSG, Waterford Institute of Technology, and also a radio ham (EI7IG). When not experimenting with new network technologies, he can be found bouncing AX25 packets through LEO satellites.

jronan@tssg.org

THIS GUIDE IS DESIGNED TO GET A Xen server up and running in no time. It is a simple copy-and-paste guide which should get you through a bare-bones install with minimal trouble and time. Xen is an exciting and still relatively new technology; more information can be obtained by visiting the Xen homepage at <http://www.xensource.com> and also by reading “The Inevitability of Xen,” by Crowcroft et al. [1]. The only limitations you will find with Xen will derive from your hardware or your ingenuity.

As this guide is aimed at beginners, it is going to cover the installation of Xen from the binary packages. The binary packages are recommended for people who are new to Xen and are uncomfortable with the range of configuration options that the source install offers.

The latest stable release of Xen can be found at <http://www.xensource.com/products/downloads/>. The sources are also available from BitTorrent sites, among others; however, we recommend going with the official releases to ensure validity, security, and stability [2]. We obtained the 3.0.1 binary install file and thus will use this as a reference for the rest of the guide; simply replace 3.0.1 with your 3.0.x in the relevant positions in order to install the software successfully.

The Linux Filesystem Hierarchy Standard recommends placing the source file in the `/usr/src` folder, and that’s where we will put it and unpack it [3].

Prerequisites

We ran the following commands to install the dependency packages and to remove some outdated and unnecessary packages (using Debian):

```
xen:# apt-get remove exim4 exim4-base lpr nfs-  
common portmap pidentd pcmcia-cs pppoe  
pppoeconf ppp pppconfig
```

```
xen:# apt-get install screen ssh debootstrap  
python python2.3-twisted iproute bridge-utils  
libcurl3-dev
```

Now that we have the necessary files installed, let’s extract the software from its `.tar` file and run the install script:

```
xen:# /usr/src$ cd xen-3.0.1-install
xen:# /usr/src/xen-3.0.1-install$ ./install.sh
xen:# /usr/src/xen-3.0.1-install$ mv /lib/tls /lib/tls.disabled
```

The last command is necessary to avoid the emulation slowdown problems with the glibc libraries that are installed by default [2].

You should now have the Xen software installed on your computer. To start the Xen services at boot time, the following commands need to be run:

```
xen:# update-rc.d xend defaults 20 21
xen:# update-rc.d xenddomains defaults 21 20
```

The final additions to be made are to add the Xen kernel to the bootloader program (Grub). Scroll through the file until you find the line that reads:

```
### BEGIN AUTOMAGIC KERNELS LIST
```

Just above that is the place where we must make our addition to the file. Enter the following text:

```
title Xen 3.0 / XenLinux 2.6.12
kernel /boot/xen.gz dom0_mem=64000
module /boot/vmlinuz-2.6.12-xen0 root=/dev/hda1 ro console=tty0
```

Note that it is important to make sure that your root is indeed /dev/hda1. If it is not, simply change the value after root= to match it. If you are unsure what your root name is, scroll further down the menu.lst file and you will see the default kernel and its root value.

Reboot the machine; at the boot prompt, Grub will now list Xen 3.0/ XenLinux 2.6.12 as the first kernel and boot it automatically. Everything should load normally and you will be given your standard login. If the machine does not boot, the following may fix the problem.

If the Xen machine executes a hard reboot as it is starting up, the problem rests with the amount of RAM in your machine. You will get no error message with this problem, and the last thing you will see is a line that says:

```
"Scrubbing free RAM. . . . ."
```

Then the screen will go black and do a hard reboot. The solution is to remove the excess RAM (while still keeping the DIMMs balanced) and reboot the machine. Currently, the binary install can only cope with a maximum of 3583 MB of RAM. To use more RAM, a source install needs to be performed and PAE support must be built into the kernel; however, this topic lies outside the scope of this guide.

Creation of Domains

Now that we have the Xen software installed, it is time to get to the creation of the virtual machines. First, we are going to create a storage area for our virtual machines:

```
xen:# mkdir /virtual && cd /virtual
```

Here we are going to create two directories in which to store and configure the images:

```
xen:/virtual# mkdir vm_base
xen:/virtual# mkdir images
```

We will create a default image and swap image from which our virtual machines will be derived. Execute the following commands:

```
xen:/virtual# dd if=/dev/zero of=/virtual/images/vm_base.img bs=1024k
count=xxxx
xen:/virtual# dd if=/dev/zero of=/virtual/images/vm_base-swap.img
bs=1024k count=xxx
```

Note that the value that `count=` specifies is the size the image will be in megabytes. Simply change it to a value that will suit your needs; only your machine capacity is the limit. (See p. 65 for sample file sizes.)

Now we need to format the base image to be `ext3` so that it can serve as our journaling filesystem. We have chosen `ext3` as it is faster than `ext2` and has stronger guarantees for data integrity [4].

```
xen:/virtual# mkfs.ext3 /virtual/images/vm_base.img
```

Answer yes to the question prompted regarding the warning about the block special device. Now we need to configure the swap file to be a swap area:

```
xen:/virtual# mkswap /virtual/images/vm_base-swap.img
```

Next it's time to install the Debian base system to our newly created image. First, though, we need to mount our image:

```
xen:/virtual# mount -o loop /virtual/images/vm_base.img /virtual/vm_base
```

Debootstrapping the Base Image

We run the `debootstrap` command to download all the prerequisite packages, using the following command:

```
xen:/virtual# debootstrap --arch i386 sarge /virtual/vm_base/
http://ftp2.de.debian.org/debian
```

Now change root and configure the images `apt` program to specify how we want to pull down our software and updates:

```
xen:/virtual# chroot /virtual/vm_base
xen:# apt-setup
```

During the standard `apt` setup, you will be asked some basic questions regarding your location and which mirror you wish to use to speed up the process. When this is done, edit the `sources.list` that comes with `apt` and change the word `testing` to `stable` wherever it appears in the file. Now update your software repository:

```
xen:# apt-get update
```

The next step in the installation process involves setting up the locales for your region:

```
xen:# apt-get install localeconf
```

Choose the locales to install depending on your country (e.g., `en_IE ISOxxxx` for Ireland or `en_US ISOxxxx` for the United States).

Next, configure the base system using `base-config`. A menu with various installation options will be presented to you. The important things to configure are:

1. Users and passwords. This is where you set the default user name, password, and root password. This is an important part as each image subsequently created from the base image will have these default passwords, which will need to be changed.
2. The time zone.

3. Which software to install. When the program prompts for additional software to be installed we choose “none,” as this is the base image, from which the other virtual machines will later be derived. Each derived machine can be customized when it is ready.

When you are satisfied with the system, simply hit return and you are finished configuring the base system.

There are some small configurations still to be completed. First, remove the hostname from the system. We remove the hostname because `debootstrap` copies this from the host machine to the newly created image so both will have the same hostname:

```
xen:# rm -f /etc/hostname
```

Now we need to create our networking interfaces by editing `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback
    address 127.0.0.1
    netmask 255.0.0.0
```

Next we edit the `fstab` file; it must end up looking exactly like the following in order to represent the internal structure of the virtual image, its mountpoints, and its filesystem types:

```
/dev/hda1 /      ext3    defaults 1 2
/dev/hda2 none  swap   sw       0 0
/dev/pts  devpts gid=5,mode=620 0 0
none     /dev/shm tmpfs  defaults 0 0
```

These values will map to the configuration file values for the root and swap later on in the configuration of the virtual domains themselves.

Our last configuration option sees the creation of the hosts file:

```
127.0.0.1    localhost.localdomain localhost
# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
ff02::3     ip6-allhosts
```

Now we leave the `chroot` environment with `exit`. All that is left is for us to copy the kernel modules to our virtual machine and unmount the image.

```
xen:/virtual/vm_base# cp -dpR /lib/modules/2.6.12.6-xenU\
/virtual/vm_base/lib/modules/
xen:/virtual/vm_base# mv /virtual/vm_base/lib/tls /virtual\
/vm_base/lib/tls.disabled
xen:/virtual/vm_base# umount /virtual/vm_base
```

The base image is now complete.

Creation of Virtual Domains

Now that we have a base image to work off of, it is time to go and make some virtual machines. We will do this by copying the base image like so:

```
xen:/virtual/vm_base# cp -pf /virtual/images/vm_base.img\
/virtual/images/vm01.img
```

```
xen:/virtual/vm_base# cp -pf /virtual/images/vm_base-swap.img\  
/virtual/images/vm01-swap.img
```

Now we create a configuration file for this new domain. Xen is located in `/etc/xen`, so that is the place where we will leave the configuration files, because the Xen software automatically scans this directory for the matching file. We use `/etc/xen/myfirstdomain.sxp` as the name for our first domain.

Here is a copy of the domain that we created:

```
name="myfirstdomain"  
kernel="/boot/vmlinuz-2.6.12.6-xenU"  
root="/dev/hda1"  
memory=64  
disk=['file:/virtual/images/vm01.img,hda1,w','file:/virtual/images\  
/vm01-swap.img,hda2,w']  
  
# network  
vif=[ " ]  
dhcp="off"  
ip="10.0.0.50"  
netmask="255.0.0.0"  
gateway="10.0.0.254"  
hostname="myfirstdomain.yourdomain.org"  
  
extra="3"
```

The ip addresses should match ranges within your organization's network. It is simply a case of sorting the networking out more than anything else. We set DHCP to be off in our instance, but if your network requires DHCP to be on as a means of dealing with addresses it's simply a matter of changing "off" to "on."

The root value is the value that we set earlier in the `/etc/fstab` file. The mappings between the swap and root values that we set earlier in the `/etc/fstab` file are also evident in the specification of the disk value.

Note that the memory value is the amount of RAM in megabytes that you are going to give to this domain. In this case, our domain is going to receive 64 MB of RAM to work with.

Now to start the machine, you need to be logged in as root.

Once in as root, you have access to the Xen software. If you type in `xm help`, you will get a listing of the available commands and how to use them. To create a domain we will execute the following command:

```
xen:# xm create -c myfirstdomain.sxp
```

There is no need to specify the exact path to the `myfirstdomain` file, as the Xen software automatically looks in `/etc/xen` for a file matching the configuration file you are using. If you placed the configuration file elsewhere, simply insert the complete path to the file.

The `-c` flag is used to ask for a console for the domain you have just launched. If all goes ok, you should see the machine booting up and eventually you will get to the login prompt. If you get an error saying the domain failed to balloon, it is an error associated with allocating too little memory to the virtual machine. You have not allocated enough RAM to allow domU to boot successfully. You will have to use the `xm destroy myfirstdomain` to stop the domain, then edit the configuration file to allocate more memory to your domain, and use the create command to launch the domain.

Log in with the default username and password that you specified in the configuration of the base system. It is a good idea now to change the default password. This is a major security issue, as each domain is created from the same base system and thus has the same username and passwords!

With your domain you should be able to ping the master Xen server, other xen domains floated, and other hosts on the same network as yours. It should also be possible to ssh into the domain.

When you are finished with your domain and wish to exit it, you can do a shutdown as normal, which will send you back to the original Xen domain from where you came, or if you wish to leave it running and wish to return to Xen, simply hold down CTRL +]. This will take you back to Xen. If you run `xm list` you should see your domains that are successfully floated, including information such as how much memory they are allocated and their domain name. To get a console to one of them simply run `xm console myfirstdomain` where `myfirstdomain` is the name of the domain we specified within the configuration file and is the name that appears in the list of domains we see when `xm list` is run.

If you wish to create more domains it is simply a matter of copying the base image:

```
xen:# cp -pf /virtual/images/vm_base.img /virtual/images/vm0X.img
xen:# cp -pf /virtual/images/vm_base-swap.img /virtual/images\
/vm0X-swap.img
```

The `vm0X` just needs to be changed to a new unique number or name.

A corresponding config file needs to be created in `/etc/xen`, which references the newly created image file in its `disk=` parameter.

If you wish to have your domains started automatically at startup, a link must be created in the `auto` folder that Xen scans as the system boots. This can be achieved by doing this:

```
xen:# ln -s /etc/xen/myfirstdomain.sxp /etc/xen/auto
```

Restart the machine and see if the domains come up successfully.

Extra Configuration

The final configuration that must be done in order to create more than three domains may need to be performed now.

Each virtual image and its swap area run on a loop each. The default number of loops is 7. If you attempt to float a fourth or fifth domain you will get this error:

```
"Error: Device 769 (vbd) could not be connected. Backend device not found."
```

This means that we can only create at most three domains with this setup (as each requires two loops to run). So we need to do some editing to vital files. Again, ensure a backup has been made in case things go wrong.

We need to edit the modules configuration file `/etc/modules.conf` and add these options anywhere in the file:

```
options loop max_loop=64
rmmod loop
modprobe loop
```

Once done, if you are running `devfs`, the new loops should have been automatically created.

If you still only see seven values for loop, you need to edit /dev/MAKEDEV and recompile it to make the changes take place. This is a very big file; you need to scroll down until you see the following:

```
loop)
    for part in 0 1 2 3 4 5 6 7
    do
        makedev loop$part b 7 $part $disk
    done
    ;;
```

This needs to be changed to:

```
loop)
    for part in `seq 0 63`
    do
        makedev loop$part b 7 $part $disk
    done
    ;;
```

Then recompile by running `makedev loop`.

Verify in /dev that there are now 64 loops created, which is enough for 32 machines to be created. If you need more, change the 63 to a number you desire.

When this is done, restart the machine and everything should be working fine.

RESOURCES AND LINKS

- [1] J. Crowcroft et al., “The Inevitability of Xen,” *login*, 30, no. 4 (2005): 10–13. Available at <http://www.usenix.org/publications/login/2005-08/pdfs/crowcroft.pdf>.
- [2] *Xen User's Manual*. Available at <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user/>.
- [3] Filesystem Hierarchy Standard Group, R. Russell et al., eds., *File System Hierarchy Standard*, 2004. Available at <http://www.pathname.com/fhs/pub/fhs-2.3.pdf>.
- [4] M.K. Johnson, “Red Hat's New Journaling File System: ext3,” 2001. Available at <http://www.redhat.com/support/wpapers/redhat/ext3/#advantages>.
- [5] F. Timme, “The Perfect Xen 3.0 Setup for Debian,” 2006. Available at http://www.howtoforge.com/perfect_setup_xen3_debian.

Acknowledgment: This work is partially supported by the IST ENABLE project under the European Commission's 6th framework program.