



How to Install Windows on Xen™ 3.0

A XenSource Technical Note for the Windows 2003 Server

Introduction

This note describes how to install Windows 2003 Server on Xen. It provides an overview of the Linux installation, and detailed steps for installing and configuring Xen and starting the Windows 2003 Server installation. It is recommended that this installation be done for demonstration and early non-performance evaluation only; this is not intended to produce a production-ready configuration.

The user is expected to have basic Linux knowledge, and be able to install Linux onto a computer.

XenEnterprise, when released from XenSource this year, will make this installation dramatically easier, and provide management, configuration, and monitoring features not found in the Open Source Xen software. Installations of Windows with XenEnterprise will be capable of deployment in production environments.

Requirements

- **A VT-enabled Intel system or an AMD-V enabled AMD system with 1GB or more of physical memory**
- **Scratch hard drive** – you'll need a hard drive that can be completely wiped clean. This disk will hold the Linux OS, Xen, and Windows.
- **Linux distro install CD/DVDs** – You'll need a Linux distro with a 2.6 or later Linux kernel.
- **Xen 3.0.2 tarball** – This can be downloaded from <http://www.xensource.com/xen/downloads>. Make sure that you download the tarball, and not the RPMs.
- **Windows 2003 Server CD**

Installation Overview

1. Install Linux and additional packages
2. Download and install Xen tarball
3. Create grub boot menu entry
4. Boot system with Xen
5. Prepare to install the Windows guest VM
6. Install Windows guest VM

Install Linux OS and Additional Packages

Most of the Linux installation is straightforward, except for the disk configuration. This is explained in some detail.

1. Boot the first Linux CD/DVD and start the installation.
2. The installation procedure will help you partition the disk. We suggest that the disk be partitioned using LVM. Put Linux in the first Volume Group, and create a second Volume Group for Guest VMs.
3. When you get to the Disk Setup page, you will need to make some changes.
 - a. Delete LVM Volume Group VolGroup00 – click on it and then select Delete
 - b. Delete /dev/hda2 (of type LVM PV) – This is the big partition that is the rest of the disk besides /boot. Click on it and then select Delete
 - c. Make a new, smaller LVM PV – Click on New. In the Add Partition dialog box, change the file system type to physical volume (LVM). Make the size such that you have a substantial amount of remaining space in which to install Windows. For instance, on a 40GB disk, I made a 16384MB partition plus a 2048MB swap file + 64MB (partition overhead) = 18496. I made the swap file

twice the physical memory in the system. The remainder of the disk will be used for Guest VM Partitions. Make sure you set it to Fixed Size, and clear force to be a primary partition.

- d. Now click on LVM. In the Make LVM Volume Group dialog box, we're going to add two logical volumes. First add the swap space (file system type = swap) of size 2048, and then add mount point "/" (type ext3) and give it the remaining space.
 - e. Click on New again to add a new physical volume. Once again, make the file system type physical volume (LVM). I selected Fill to maximum allowable size to be able to use the remainder of the hard drive for guest VMs.
 - f. Select the newly created partition (/dev/hda3 in my case), and select LVM. Make the volume group name VG_Guests (your choice). You don't need to add any logical volumes right now.
 - g. If the partitioning gets confused in some way, just click Back, and then click Next to start the partitioning over again.
4. For example, a sample partitioning scheme for a 40GB disk could be:

Device	Mount Point / RAID / Volume	Type	Size (MB)
LVM Volume Groups			
VG_Guests			19520
VolGroup00			18432
LogVol00		Swap	2048
LogVol01	/	ext3	16384
Hard Drives			
/dev/hda			
/dev/hda1	/boot	ext3	100
/dev/hda2	VolGroup00	LVM PV	18496
/dev/hda3	VG_Guests	LVM PV	19577

5. SELinux must be disabled. We suggest that any firewall be disabled.
6. In addition to the default package set, the X Window System and one of KDE or Gnome should be installed.
7. After the installation has completed, login and install the packages SDL, sysfsutils and bridge-utils if not installed as part of the installation.

Download and Install Xen tarball

If you have not already downloaded the Xen tarball from <http://www.xensource.com/xen/downloads> do so now. You may copy it anywhere; the next set of steps assume it is in /tmp.

1. `cd /tmp`
2. `tar -xzf xen-3.0.2-install-x86_32.tgz`
3. The tar command will create a directory named xen-3.0.2-2-install
4. `cd xen-3.0.2-2-install`
5. Place all of the Xen files into their places in the file system by running
 - a. `sh ./install.sh`
6. Determine the name of the Xen Linux kernel version that was installed
 - a. `ls /lib/modules`
 - b. There should be a directory for the Xen Linux kernel (e.g., 2.6.16-xen)
7. Generate the module dependency list and map files
 - a. `/sbin/depmod 2.6.16-xen`
8. Create the Xen initial ramdisk image
 - a. `/sbin/mkinitrd /boot/initrd-2.6.16-xen.img 2.6.16-xen`
9. Disable tls
 - a. `mv /lib/tls /lib/tls.disabled`
10. Set xend service to start
 - a. `/sbin/chkconfig --add xend`
 - b. `/sbin/chkconfig xend on`

Create grub boot menu entry

The next step is to create a grub entry in the boot menu to boot Xen and the Xenified Linux kernel.

1. Edit `/boot/grub/menu.lst` and copy one of the existing boot entries to make the required changes for Xen. For example, here is the new menu entry to boot from `/dev/hda` (an IDE drive):

```
title Xen-i386-up (2.6.16-xen)
    root (hd0,0)
    kernel /xen-3.0.2-2.gz dom0_mem=262144
    module /vmlinuz-2.6.16-xen ro root=/dev/VolGroup00/LogVol01 rhgb quiet
    module /initrd-2.6.16-xen.img
```

The files specified in the kernel and module lines must exist in `/boot`. The title can be any string that you'd like it to be. The root device should be the same as the existing boot entries.

Boot system with Xen

Now, shutdown the system and reboot. When the system starts to reboot, select the new menu item that we just created.

If all goes well (and it will, since you followed these perfect instructions to the letter), the system will reboot with the Xenified kernel.

Once you have logged in, create a terminal window and `su` to root.

1. Type `/usr/sbin/xm info | grep caps` at the command prompt. You will see "hvm-3.0-x86_32" if Xen can detect the VT or AMD-V technology. If you do not see this, then your system does not have the appropriate processor technology, or it has been disabled in the BIOS. You must rectify this before proceeding.

Prepare to Install the Windows Guest VM

1. Create the LVM Guest partition
 - a. `/usr/sbin/lvcreate -L 10000 -n w2k3-001 VG_Guests`
 - b. This will create a 10GB partition named `w2k3-001` in the physical volume `VG_Guests` that was created during the installation
 - c. The full path to this partition is `/dev/VG_Guests/w2k3-001`
2. Transfer an ISO of the Windows 2003 Standard install CD to your system. This is used for the Windows 2003 installation.
 - a. Insert the CD into the CD drive
 - b. `dd if=/dev/cdrom of=/tmp/WINDOWS_2003_STANDARD.ISO`
3. Create the guest VM configuration file (`/etc/xen/w2k3.hvm`, for instance). A sample configuration file can be found at the end of this note.
 - a. Be sure that the lines for `disk=` and `cdrom=` are correct for your system. The `disk=` line must correspond to the LVM Guest partition created above. The `cdrom=` line must correspond to the ISO image created above.

Start the Windows Guest VM

At an `su` command prompt type `/usr/sbin/xm create w2k3.hvm` and a window should be created, with Windows Setup running in it.

Once the Windows installation has completed modify the configuration file to boot from the virtual hard drive.

1. Edit `/etc/xen/w2k3.hvm`
2. Change the line `boot=d` to `boot=c`

NOTE: Once you click the mouse in the window in which Windows 2003 is running, it will be captured there. Press the left-hand CTRL-ALT to exit grab.

Sample Windows VM Configuration File

```
# -*- mode: python; -*-
#-----
# Python configuration setup for 'xm create'.
# This script sets the parameters used when a domain is created using 'xm create'.
```

```

# You use a separate script for each domain you want to create, or
# you can set the parameters for the domain on the xm command line.
#=====

import os, re
arch = os.uname()[4]
if re.search('64', arch):
    arch_libdir = 'lib64'
else:
    arch_libdir = 'lib'

#-----
# Kernel image file.
kernel = "/usr/lib/xen/boot/hvmlloader"

# The domain build function. HVM domain uses 'hvm'.
builder='hvm'

# Initial memory allocation (in megabytes) for the new domain.
memory = 512

# A name for your domain. All domains must have different names.
name = "w2k3-001"

#-----
# the number of cpus guest platform has, default=1
#vcpus=1

# enable/disable HVM guest ACPI, default=0 (disabled)
#acpi=1

# enable/disable HVM guest APIC, default=0 (disabled)
#apic=0

# Pae
#pae=1

# List of which CPUS this domain is allowed to use, default Xen picks
#cpus = ""          # leave to Xen to pick
#cpus = "0"         # all vcpus run on CPU0
#cpus = "0-3,5,^1" # run on cpus 0,2,3,5

# Optionally define mac and/or bridge for the network interfaces.
# Random MACs are assigned if not given.
vif = [ 'type=ioemu, mac=00:16:3e:00:00:94, bridge=xenbr0' ]
# type=ioemu specify the NIC is an ioemu device not netfront
#vif = [ 'type=ioemu, mac=aa:00:b0:00:00:11, bridge=xenbr0' ]

#-----
# Define the disk devices you want the domain to have access to, and
# what you want them accessible as.
# Each disk entry is of the form phy:UNAME,DEV,MODE
# where UNAME is the device, DEV is the device name the domain will see,
# and MODE is r for read-only, w for read-write.

# This would be for using a QEMU qcow file as a virtual disk
#disk = [ 'file:/guests/1-w2k3/w2k3-demo-001.qcow,ioemu:hda,w' ]

# This is for using an LVM volume as a virtual disk
disk = [ 'phy:/dev/VG_Guests/w2k3-001,ioemu:hda,w' ]

```

```

#-----
# Configure the behaviour when a domain exits.  There are three 'reasons'
# for a domain to stop: poweroff, reboot, and crash.  For each of these you
# may specify:
#
# "destroy",          meaning that the domain is cleaned up as normal;
# "restart",         meaning that a new domain is started in place of the old
#                   one;
# "preserve",       meaning that no clean-up is done until the domain is
#                   manually destroyed (using xm destroy, for example); or
# "rename-restart", meaning that the old domain is not cleaned up, but is
#                   renamed and a new domain started in its place.
#
# The default is
#
# on_poweroff = 'destroy'
# on_reboot   = 'restart'
# on_crash    = 'restart'
#
# For backwards compatibility we also support the deprecated option restart
#
# restart = 'onreboot' means on_poweroff = 'destroy'
#                               on_reboot  = 'restart'
#                               on_crash   = 'destroy'
#
# restart = 'always'   means on_poweroff = 'restart'
#                               on_reboot  = 'restart'
#                               on_crash   = 'restart'
#
# restart = 'never'    means on_poweroff = 'destroy'
#                               on_reboot  = 'destroy'
#                               on_crash   = 'destroy'
#
#on_poweroff = 'destroy'
#on_reboot   = 'restart'
#on_crash    = 'restart'

#=====

# New stuff
device_model = '/usr/' + arch_libdir + '/xen/bin/qemu-dm'

#-----
# Disk image for
cdrom="/tmp/WINDOWS_2003_STANDARD.ISO"

#-----
# boot on floppy (a), hard disk (c) or CD-ROM (d)
boot="d"

#-----
# write to temporary files instead of disk image files
#snapshot=1

#-----
# enable SDL library for graphics, default = 0
sdl=1

```

```
#-----  
# enable VNC library for graphics, default = 1  
vnc=0  
  
#-----  
# enable spawning vncviewer(only valid when vnc=1), default = 1  
#vncviewer=1  
vncviewer=0  
  
#-----  
# no graphics, use serial port  
#nographic=0  
  
#-----  
#   serial port re-direct to pty device, /dev/pts/n  
#   then xm console or minicom can connect  
#serial='pty'  
  
#-----  
# enable ne2000, default = 0(use pcnet)  
ne2000=0  
  
#-----  
#  
#usb=1  
#usbdevice='tablet'  
  
#-----  
#   enable audio support  
#audio=1  
  
#-----  
#   set the real time clock to local time [default=0 i.e. set to utc]  
#localtime=1  
  
#-----  
#   start in full screen  
#full-screen=1  
  
#stdvga=0
```

About XenSource™, Inc. XenSource™, Inc. develops enterprise-grade virtualization solutions based on the industry's fastest and most secure server virtualization technology, the open source Xen™ hypervisor. Founded and run by the original Xen™ development team, XenSource™ products allow enterprises to realize the TCO savings that result from server consolidation, increased utilization, and reduced complexity in the datacenter. Find out more at www.xensource.com



Delivering the Power of Xen

XenSource, Inc. 2300 Geng Road Suite 250
Palo Alto, California 94303 USA

+1 650 798-5900 info@xensource.com
www.xensource.com

Xen Technical Note V07132006