



Installation Manual

VSControl

Copyright (c) 2009 by Devpond,
Landis Holdings Inc.

Table of Contents

<i>Steps for Installation</i> -----	2
<i>System Requirement</i> -----	3
Hardware compatibility -----	3
Node installation -----	3
Physical node OS-----	3
Minimum OS packages-----	3
Disk partition-----	3
<i>XEN Installation</i> -----	5
Introduction-----	5
List of Pre-requisites-----	5
Installing Pre-requisites-----	6
Xen Installation-----	7
Configuration-----	7
Reboot and Finalization-----	8
<i>Apache, MySQL, mod_python, Pexpect installation</i> -----	9
<i>Django Installation</i> -----	9
<i>MySQL_Python installation</i> -----	10
<i>Vnstat installation</i> -----	10
<i>VSControl Installation</i> -----	11
<i>Configuring Backup Server</i> -----	14
<i>Download and Install OS Templates</i> -----	15

Steps for Installation

Please follow these steps for the successful system installation:

1. Make sure you read the **System Requirement** section and follow the instructions.
2. After OS install as per the system requirement you have to perform the **xen installation**.
3. Next step is **Apache, MySQL, mod_python, and Pexpect** installation.
4. Next **Django Installation** (that is a web framework for python).
5. Next step is **MySQL_Python** installation.
6. **Vnstat installation** will be performed next.
7. After that **VSControl installation** will be performed.
8. Optionally **Configure your Backup server**.
9. And finally **Download and Install OS Templates** for the VPS.

System Requirement

Hardware compatibility

For hardware compatibility please check the following Xen links:

<http://wiki.xensource.com/xenwiki/HardwareCompatibilityList>
http://wiki.xensource.com/xenwiki/HVM_Comppatible_Processors
http://wiki.xensource.com/xenwiki/HVM_Comppatible_Motherboards

Node installation

As VControl is a management panel for Xen VPS' so it has some requirement regarding physical node installation.

How the physical system should be installed i.e. What OS' are recommended?

What is minimum packages requirement?

How the disk should be partitioned?

Physical node OS

The VControl is developed under CentOS so the physical system should be installed with CentOS. It can work with any 5.x version (tested up to 5.3).

Minimum OS packages

During the installation of VControl it searches for pre-requisite packages so there is no specific requirement during the physical node installation, however make sure you install the system with basic packages at least i.e. make, rsync, wget, rpm, and development packages. Minimum installation can also work with development packages.

Disk partition

VControl uses Logical Volumes for VPS' as vbds (virtual block devices). So all the disk space that will be used as vbds for VPS should be under some volume group. Example partition table is given below:

Device	Type	Mount Point	Size
/dev/sda			
/dev/sda1	Ext3	/boot	100MB
/dev/sda2	LVM PV	Xen_repository	Remaining size
As the remaining disk is declared as LVM PV so all new partitions will be LVMs. Create LVMs as follows:			
Xen_repository			

VSControl Installation document

Node_swap	Swap		Equal to ram size
Node_root	Ext3	/	> 10G

Rest of the VG space will be used for vbds.

XEN Installation

Introduction

For complete Xen installation on the node we need to install following Xen components:

- **xen** (kernel that includes all the device drivers and will be used to boot the node. It is also used for the VPS.)
- **xen0** (kernel that includes all the device drivers and can be used to boot the node.)
- **xenU** (kernel that does not include device drivers and usually used for the vps to boot.)

This documents shows the steps needed to install xen, xen0 and xenU from source.

List of Pre-requisites

The following list is the pre-requisites for the xen. Please make sure all are installed before starting xen installation.

- GCC v3.4 or later
- GNU Make
- GNU Binutils
- Development install of zlib (e.g., zlib-dev)
- Development install of Python v2.4 or later (e.g., python-dev)
- Development install of curses (e.g., libncurses-dev)
- Development install of openssl (e.g., openssl-dev)
- Development install of x11 (e.g. xorg-x11-dev)
- bridge-utils package (/sbin/brctl)
- iproute package (/sbin/ip)
- hotplug or udev
- latex and transfig are required to build the documentation

Installing Pre-requisites

Most of the pre-requisites can be installed using yum as:

```
# yum install -y gcc make rsync zlib-devel zlib python-devel ncurses libncurses-  
devel ncurses-devel openssl openssl-devel bridge-utils tetex tetex-latex transfig  
libtool-ltdl dev86 glibc-devel binutils iproute udev  
  
# yum groupinstall -y "X Software Development" "X Window System"
```

For xen kernel installation we need to clone it using mercurial. So we need to install mercurial first as:

```
#rpm -ivh http://packages.sw.be/mercurial/mercurial-1.2.1-1.el5.rf.i386.rpm  
#rpm -ivh http://packages.sw.be/mercurial/mercurial-hgk-1.2.1-1.el5.rf.i386.rpm
```

Now clone the kernel in a directory where xen source resides i.e. both the kernel and xen should be in same directory.

```
#cd /usr/local/src  
# hg clone http://xenbits.xensource.com/linux-2.6.18-xen.hg
```

Xen Installation

Download and untar the xen-3.2.0 official release

URL : http://www.xen.org/download/index_3.2.0.html

```
#cd /usr/local/src
# wget http://bits.xensource.com/oss-xen/release/3.2.0/xen-3.2.0.tar.gz
# tar -xzvf xen-3.2.0.tar.gz
# cd xen-3.2.0
```

If memory size is greater than 4GB then we need to compile and install xen with PAE option enabled otherwise not. So from the following commands you will choose either one depending upon memory (ram) size.

```
# make world or
# make XEN_TARGET_X86_PAE=y world
```

During the execution it may ask for selection a yes / no for different components supports etc, accept default values for all these by pressing enter key.

```
# make install or
# make XEN_TARGET_X86_PAE=y install
```

Xen installation is completed at this stage.
Only some configuration is remaining. Please follow the following steps.

```
# depmod -a /lib/modules/2.6.18.8-xen/
# mkinitrd --allow-missing -f -v /boot/initrd-xen-3.2.0.img 2.6.18.8-xen
```

We have used the allow-missing parameter here because it was complaining about dm-cache missing module that is actually a known bug so using this parameter we can overcome to it. You should try without this parameter first.

Configuration

Now modify the grub file to add the newly installed kernel. e.g. by adding the following lines in the grub file and also set the default and fallback as required.

```
title Xen 3.2.0 / XenLinux(2.6.18-xen)
    root (hd0,0)
    kernel /xen-3.2.gz dom0_mem=500MB
    module /vmlinuz-2.6.18.8-xen ro root=/dev/Xen_repository/Linux_OS
    module /initrd-xen-3.2.0.img
```

After updating the grub.conf run the command below to make sure xend service is available in run level 3 and above.

```
# /sbin/chkconfig --level 345 xend on
```

Reboot and Finalization

Reboot the server so that xen kernel is loaded. Once the server back check the kernel and xend service whether it's running or not.

At this stage xen installation is completed. Now you need to install xen0 and xenU kernels and these are simple. If your xen installation is successful, the xen0 and xenU installation will be completed without any error. Just goto the xen source and execute the following commands but if the memory is greater than 4GB then you need to select the PAE option here otherwise not.

```
# make KERNELS="linux-2.6-xen0 linux-2.6-xenU" world or
# make KERNELS="linux-2.6-xen0 linux-2.6-xenU" XEN_TARGET_X86_PAE=y world

# make KERNELS="linux-2.6-xen0 linux-2.6-xenU" install or
# make KERNELS="linux-2.6-xen0 linux-2.6-xenU" XEN_TARGET_X86_PAE=y install

# sh install.sh
```

This is all about xen installation.

Apache, MySQL, mod_python, pexpect installation

These modules can be installed using yum as follows:

```
# yum install mod_ssl httpd distcache mysql* mod_python pexpect
```

Django Installation

Download the source tarball from: <http://www.djangoproject.com/download/1.0.2/tarball/>

```
# tar -xvzf Django-1.0.2-final.tar.gz
# cd Django-1.0.2-final
# python setup.py install
```

Check the django installation by importing the django module in python terminal. If it is imported successfully that means installation is fine otherwise not. Just run the python command to get into python terminal as:

```
# python

[root@vsc2 scripts]# python
Python 2.4.3 (#1, Jan 21 2009, 01:10:13)
[GCC 4.1.2 20071124 (Red Hat 4.1.2-42)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>>
```

You can see no error here so django installation is fine. Press Ctrl+D to exit python command line.

MySQL_Python installation

Download the source tarball from:

<http://internap.dl.sourceforge.net/sourceforge/mysql-python/MySQL-python-1.2.2.tar.gz>

```
# tar -xvzf MySQL-python-1.2.2.tar.gz
# cd MySQL-python-1.2.2
# python setup.py build
# python setup.py install
```

Vnstat installation

Download the source tarball from: <http://humdi.net/vnstat/vnstat-1.7.tar.gz>

```
# tar -xvzf vnstat-1.7.tar.gz
# cd vnstat-1.7
# make
# make install
# vnstat -u -i eth0
```

VSControl Installation

- Buy the license and download the software from <http://devpond.com/license/>

```
# tar -xvzf vscontrol.tgz
# cp -r vscontrol /var/www/django/
# cd /var/www/django/vscontrol/
```

- create mysql database with name “vsc_db” e.g.

```
# mysqladmin create vsc_db
```

- Run the following two scripts as:

```
# python manage.py syncdb
# python setup.pyc
```

- Follow the instructions and provide required information the setup script asks.
- Now its time to configure django, mod_python and apache as per requirement:

To configure Django with mod_python, first make sure you have Apache installed with the mod_python module activated.

This usually means having a LoadModule directive in your Apache configuration file (/etc/httpd/conf/httpd.conf). It will look something like this:

```
LoadModule python_module modules/mod_python.so
```

- Then, edit your Apache configuration file and add the following (usually at the end of config file):

```
<Location "/vscontrol/">
  SetHandler python-program
  PythonHandler django.core.handlers.modpython
  SetEnv DJANGO_SETTINGS_MODULE vscontrol.settings
  SetEnv PYTHON_EGG_CACHE /var/www/django/vscontrol/python-eggs
  PythonOption django.root /vscontrol
  PythonPath "['/var/www/django/vscontrol', '/var/www/django'] + sys.path"
  PythonDebug On
</Location>
```

At this stage the installation completed. Next step is to configure vscontrol.

- open the vsconf.py script that exist inside vscontrol directory (/var/www/django/vscontrol/).

This file contains all the config option for vscontrol. You need to set some of them like `vg_name`, `hostname`, `kernel`, `ramdisk`, `gateway`, `presolver`, `sresolver`, `tresolver` and `node_id`. These are some configuration that will vary from machine to machine. So please set the values for them appropriately.

You can see some configuration related to paths, you don't need to change them if you have installed the vscontrol at default path "/var/www/django" otherwise you need to set these paths.

- Make the startup script executable as:

```
# chmod 755 /etc/init.d/vsdaemon.sh
```

- Start the vsdaemon as:

```
# /etc/init.d/vsdaemon.sh start
```

As this is the first time you are starting vscontrol daemon, it will ask for the license information. It will ask for the username, password and license key. Enter the required information correctly otherwise vsdaemon will not work properly. It will be started but will not perform any actions.

Now restart your apache,

```
# service httpd restart
```

Open your browser and enter the URL. e.g. http://hostname_or_ipaddr/vscontrol/login/ Where `hostname_or_ipaddr` is the hostname or IP address of the hardware node. You can login now using your admin user you created at VSControl setup.

Next step is to configure the VSControl cron scripts. VSControl has three types of cron scripts. These scripts are as under

1. Script for daily backup of VPS data. Will be executed once in 24 hours. Example settings are as under:

```
# crontab -e
```

and enter:

```
15 04 * * * python /var/www/django/vscontrol/vscbackups.pyc
```

2. Scripts for traffic logs update and traffic monitoring for outage. Will be executed every three minute and 10 minute respectively, e.g.

```
* / 3 * * * * python /var/www/django/vscontrol/vsctraffics.pyc  
10 * * * * python /var/www/django/vscontrol/vsctraffics.pyc
```

3. Script for port monitoring. Will be executed after every five minutes, e.g.

```
* / 5 * * * * python /var/www/django/vscontrol/vscports.pyc
```

Configuring Backup Server

To backup data on a remote server, we need to setup ssh key authorization.

SSH Keys

First log in on physical node as user root and generate a pair of authentication keys. Do not enter any passphrase:

```
node@root:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/a/.ssh/id_rsa):
Created directory '/home/a/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/a/.ssh/id_rsa.
Your public key has been saved in /home/a/.ssh/id_rsa.pub.
The key fingerprint is:
3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 node@root
```

Append physical nodes's new public key to backup@root:~/.ssh/authorized_keys and enter backup server's password:

```
node@root:~# cat .ssh/id_rsa.pub | ssh backup@root 'cat >>
.ssh/authorized_keys'
backup@root's password:
```

Now on you can log into your backup server as root from your physical node without password.

A note from one of our readers: Depending on your version of SSH you might also have to do the following changes:

- Put the public key in `.ssh/authorized_keys2`
- Change the permissions of `.ssh` to 700
- Change the permissions of `.ssh/authorized_keys2` to 640

Download and Install OS Templates

For VPS installation VSControl needs OS templates. We have several OS templates available that licensed users can download and use. Here is information about how to download and install these templates with VSControl.

Download

We have templates available at: <http://www.devpond.com/downloads/os/>

The file centos5-32.tar.gz means CentOS 32bit. debian5-64.tar.gz means debian 5 64bit.

From here you can download these templates on your system, e.g.

```
# wget http://www.devpond.com/downloads/os/centos5-32.tar.gz
```

Install

After downloading the template you can install it by using the following steps:

1. Untar the downloaded template. e.g.

```
# tar -xvzf centos5-32.tar.gz
```

2. Go to vscontrol source. E.g.

```
# cd /var/www/django/vscontrol/scripts
```

3. Run the create_lvm.py script to create an LVM (to copy the OS template data into it) and pass it the name of the LVM and size either through command line or as user input, e.g.

```
# python create_lvm.pyc centos5 1536M or  
# python create_lvm.pyc centos5 1.5G or  
# python create_lvm.pyc
```

4. Mount your newly created LVM at some mount point, e.g here 'Xen_repository' is the Volume Group name and it can be different at your machine.

```
# mkdir -p /mnt/centos5  
# mount /dev/Xen_repository/centos5 /mnt/centos5
```

5. Copy the template data on this mounted LVM, e.g assuming your untared OS data is at /root/centos5

```
# cd /root/centos5  
# rsync -zaHlv -r * /mnt/centos5/
```

6. Copy the xenU kernel from your /lib/modules/ into your mounted LVM at same path, e.g.

```
# cp -r /lib/modules/2.6.18.8-xenU/ /mnt/centos5/lib/modules/.
```

7. Umount the template

```
# umount /mnt/centos5/
```

8. Login to VSControl and add the templates. Check admin manual for further detail <http://www.devpond.com/downloads/VSControlAdminManual.pdf>
9. OS template is installed now and can be used for the creation and installation of VPS.
10. Repeat these steps for each template.