VPS Server Configuration Guide, Detailed for Linux Beginners
(Hosting Service Application)

Objective

The objective of this guide is to provide an easy to understand and follow tutorial that will help people who are not Linux experts configure their own VPS (virtual private server) suitable for use in offering commercial Internet hosting services. The final product will provide a graphical administration interface where client accounts, including hosting reseller clients, can be created and managed by non-technical personnel. This guide can also be used to configure a dedicated server, but it is geared towards the VPS for economic reasons discussed below.

This guide explains how to use the VPS control panel, install necessary hosting software, configure the system for hosting services, install a billing system suitable for managing recurring billing customers, then finally lock-down the system for security.

This guide is written from the perspective of configuring a Linux VPS while working from a Windows workstation, which is the most common case today.

The following software will be included with the installation.

- Linux operating system – CentOS 5.x 32-bit.
- Webmin – A low-overhead graphical administration and configuration interface for Linux.
- Kloxo control panel – System administration, add domain customers, add reseller accounts (who can add their own domain customers), manage DNS, add domains, add mysql databases, add email accounts, manage php & apache, and allow clients to manage their own accounts.
- BIND – DNS server.
- Qmail – Email server (i.e., a mail transfer agent, or MTA).
- Courier-IMAP – POP3 email server.
- MySQL – Database server.
- Apache – Web server (HTML server).
- Pure-FTP – FTP (file transfer protocol) server.
- Spamassassin, Spamdyke, & ClamAV – Email filters.
- APF, mod_evasive, mod_security, & chkrootkit – Server security applications.
- TheHostingTool – Real-time provisioned recurring billing application.

System Cost

The operating system, as well as all software required for this project, is open source. As such, the software for this server costs nothing. The only costs associated with the server are the subscription cost for the VPS, which should be under $10/month (hopefully more like half that much), and of course any domain name registration fees.

What exactly is a VPS anyway?

A VPS (virtual private server) is an account that leases an isolated portion of a commercial server hard drive (called a “canister”), as well as memory and CPU resources, where a dedicated operating system is installed. A VPS operates very much like a co-located dedicated server machine, where you can reboot the operating system, reinstall the operating system, and even have root access, all for a modest price. You'll find a VPS to be an entirely satisfactory solution for production service.

What are the system requirements for this server?

A Linux VPS account with 1 gb memory and 30 gb of drive space will do nicely for this project. It's possible to run with less memory than that, even as low as 256 mb memory, but you'll have to give up email virus scanning and the webmin graphical interface for Linux. I strongly recommend at least a 512 mb account, but 1 gb or more is preferred. If you run your server with everything, including the ClamAV email filter, you will need about 600 mb, so it's nice to have 1 gb of memory, or more if you can afford it. I happen to be running 2 gb.

It's easy to find economical accounts like that for under $10/month, and if you look around you can find them below $5/month. Even $2 to $3 per month is not unheard of if you pay quarterly or biannually. Look around at lowendbox.com for the best deals.

Why a VPS instead of a dedicated server?

In a word – cost. It's the objective of every successful business operator to keep costs down. A dedicated server has the up-front cost for building a computer system to be used as the server machine, as well as co-location costs to place the server in a commercial server room so it can have a commercial Internet connection. The VPS account includes a connection to a commercial Internet account.

You aren't going to get away with operating a hosting server on a residential broadband connection (residential DSL or cable service). That's because residential service will block certain ports that you'll need. You could arrange for a business broadband connection that doesn't block ports, but business broadband service is considerably more expensive than residential service. You're a lot better off financially with a VPS.

Why not a Windows server?

The knee-jerk reaction of many new server operators who aren't familiar with Linux is to consider a Windows server. While you can always do that, I don't recommend it. In the first place a Windows VPS is going to require a lot more resources than Linux, so it will cost you a lot more each month. That's because you can't get away from the huge graphical overhead of Windows (note: Windows Server 2008 & above can be installed as a command line interface), where Linux can be reduced to command line operation. There are also Windows server licensing fees. While you can easily find a suitable Linux VPS in the $5 to $10 per month range, a Windows VPS will start at more like $30/month.

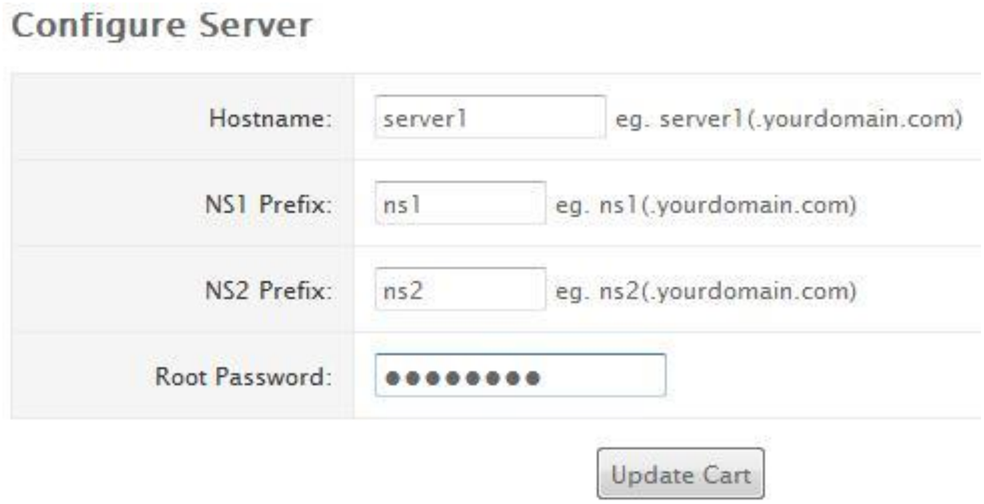
Then there is the dimension of reliability. The low-overhead command line environment of Linux is much more reliable than Windows environments, often being able to run for many months without the need to reboot. For those reasons I wouldn't recommend Windows for a production server platform.

Which Linux operating system to install?

When you open your VPS account you will probably have a choice of Linux operating systems, but most come with Centos 5.x 32-bit as the default. For this project I'm recommending CentOS 5.x 32-bit. I'm selecting CentOS 5.x 32-bit instead of 64-bit because 64-bit uses more memory, and the current version of Kloxo (6.1.15) doesn't work with CentOS 6.x. That may change with the next version, but we're stuck with CentOS 5.x for the time being. I have nothing in particular against other common flavors of Linux, but my familiarity is with Red Hat products, which CentOS is based upon. In short, I know where to find things in CentOS, and being based on RHEL (Red Hat Enterprise Linux) it's considered to be an enterprise product that's appropriate for production server purposes.

Basic VPS Configuration Instructions

Once you have selected a VPS service that suits your needs, you will sign-up for the service. During the sign-up process you will typically be asked for some configuration details, similar to this.

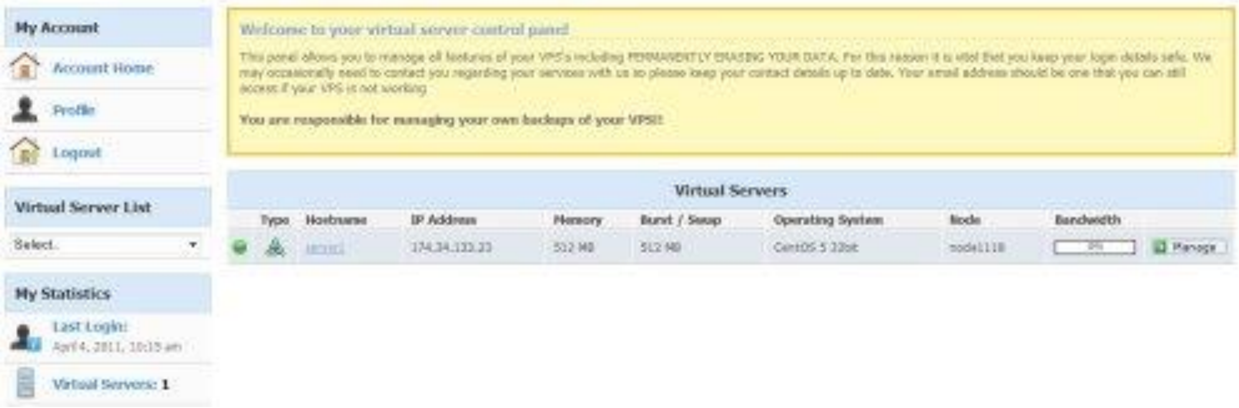


Configure Server

Hostname:	server1	eg. server1(.yourdomain.com)
NS1 Prefix:	ns1	eg. ns1(.yourdomain.com)
NS2 Prefix:	ns2	eg. ns2(.yourdomain.com)
Root Password:	●●●●●●●●	

Fill out the VPS order form like above, using a password that you can remember, but leave your root domain name out of it. If you enter your domain name as part of the Hostname or Nameserver prefixes it will be confusing to the Apache http server later on, since a third-party control panel will be handling domain hosting for you using a different directory structure than Apache uses. Therefore, it's best to just enter the Hostname and NS prefixes generically, as illustrated above, during the setup process.

Once you signup you'll be given login information for your VPS account. Go to the address they give to you and login. It should look something like this.



My Statistics

Last Login: April 4, 2011, 10:15 am

Virtual Servers: 1

Welcome to your virtual server control panel

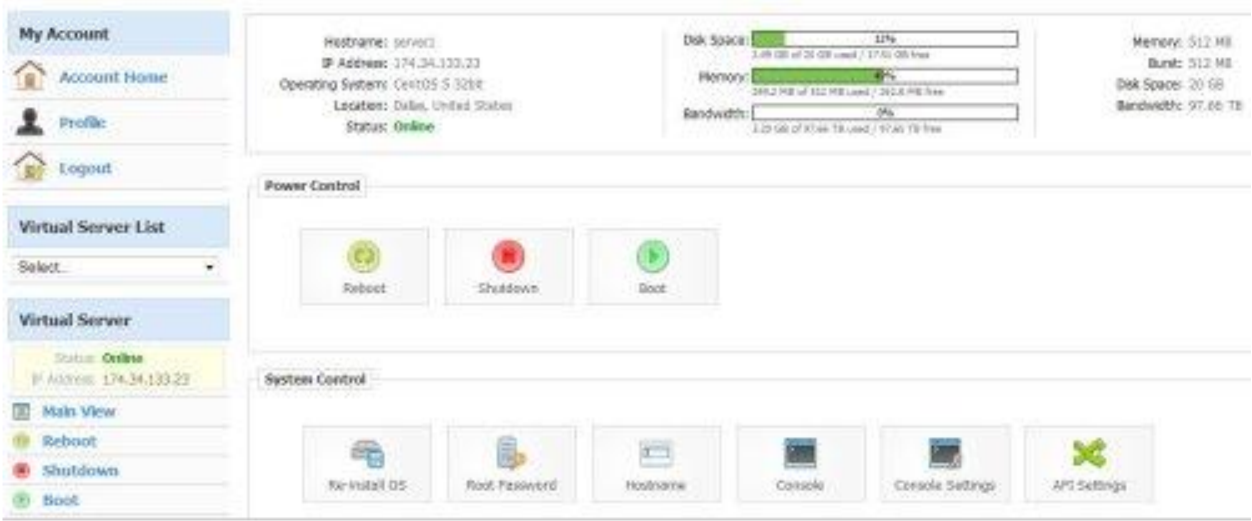
This panel allows you to manage all features of your VPS's including PERMANENTLY ERASING YOUR DATA. For this reason it is vital that you keep your login details safe. We may occasionally need to contact you regarding your services with us so please keep your contact details up to date. Your email address should be one that you can still access if your VPS is not working.

You are responsible for managing your own backups of your VPS!!

Virtual Servers

	Type	Hostname	IP Address	Memory	Burst / Swap	Operating System	Node	Bandwidth	
●		server1	174.34.133.23	512 MB	512 MB	CentOS 5 32bit	node1118	0%	Manage

Under “Virtual Servers” click the Manage button on the far right. The new view should look something like this.



Hostname: server1
IP Address: 174.34.133.23
Operating System: CentOS 5 32bit
Location: Dallas, United States
Status: Online

Disk Space: 12%
Memory:
Bandwidth: 0%

Memory: 512 MB
Burst: 512 MB
Disk Space: 20 GB
Bandwidth: 97.66 TB

Status: Online
IP Address: 174.34.133.23

You can use that panel to start, stop, or reboot your server. You can also click “Re-install OS” to reinstall the entire operating system to get a fresh start or change to a different flavor of Linux. If your operating system is not currently CentOS 5.x 32-bit then you will want to change that now by clicking “Re-install OS” and selecting CentOS 5 32-bit. After confirming that you want to reinstall, you might need give it maybe 20 to 30 minutes to complete the reinstall process, but a fast VPS might only take 5 minutes. It won’t actually tell you when it’s done, but you can check the disk & memory usage statistics. If they stay the same for two consecutive checks about a minute apart, you can be pretty sure that it’s done.

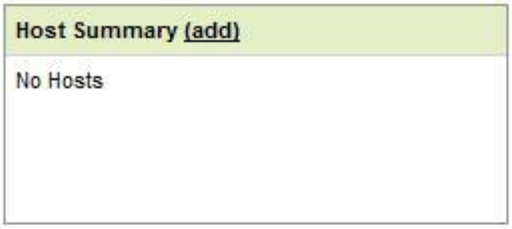
As an added note on VPS operating system templates, you may see an option to install CentOS 5.x 32-bit with Kloxo in the operating system list. While you will be using Kloxo, you will be installing Kloxo separately from the operating system. The reason is that installing to combination will automatically provision abbreviated web & DNS servers, where separate installs will provision Apache & BIND. For that reason you will want to select CentOS 5.x 32-but without Kloxo.

Configuring DNS with your new IP address

Before starting to work on the server itself, there is one item that you need to take care of. You need to do some preliminary configuration at your domain registrar using the IP address that your VPS has just been assigned. I use Godaddy.com as my registrar, so I can’t promise that these instructions will work at any registrar. We will be using the same IP address for both nameservers, since that’s all you’ve got. Some registrars require unique IP addresses, and some even require them to be in different subnets.

If you have not registered a domain name, do so now. This will be the primary domain name that you will run your hosting business as.

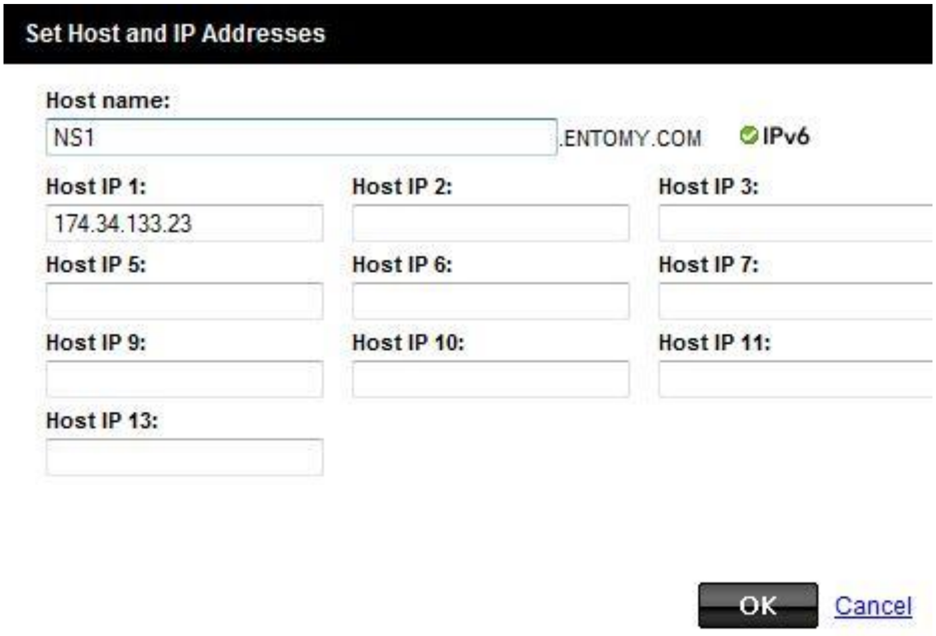
Login to your godaddy.com account and navigate to Manage Domains. You should see your primary hosting business domain in the list. Click on the domain name to go to the setup panel. You will find a box called “Host Summary” in the bottom left, which looks like this.



Host Summary (add)

No Hosts

Click the “add” link to enter a new host name. You will enter “ns1” as the host name prefix for your domain name (I’m using entomy.com in this example, but you will substitute your primary business domain name throughout this tutorial), and associate that name with the IP address that your VPS assigned to you by entering that IP address in the “Host IP 1” box. That entry should look like this.

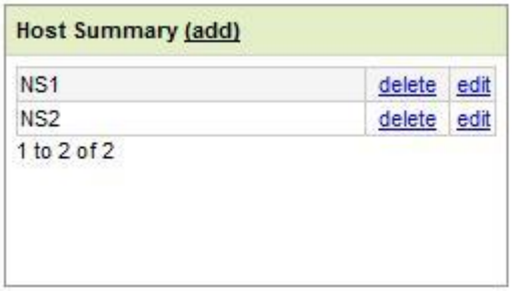


Set Host and IP Addresses

Host name: NS1 .ENTOMY.COM

Host IP 1: 174.34.133.23

Click OK. Now click “add” again. This time enter “ns2” in the host name box, and enter the same IP address in the “Host IP 1” box. Click OK. You should now see both entries in the Host Summary box, and it should look like this.

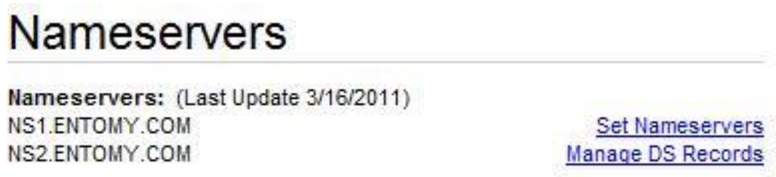


Host Summary (add)

NS1	delete	edit
NS2	delete	edit

1 to 2 of 2

You will give those name server addresses to your hosting customers for nameserver settings, and you will also enter those same nameserver addresses for your own domain. You can do that now by scrolling up the configuration panel about halfway to where is says Nameservers.

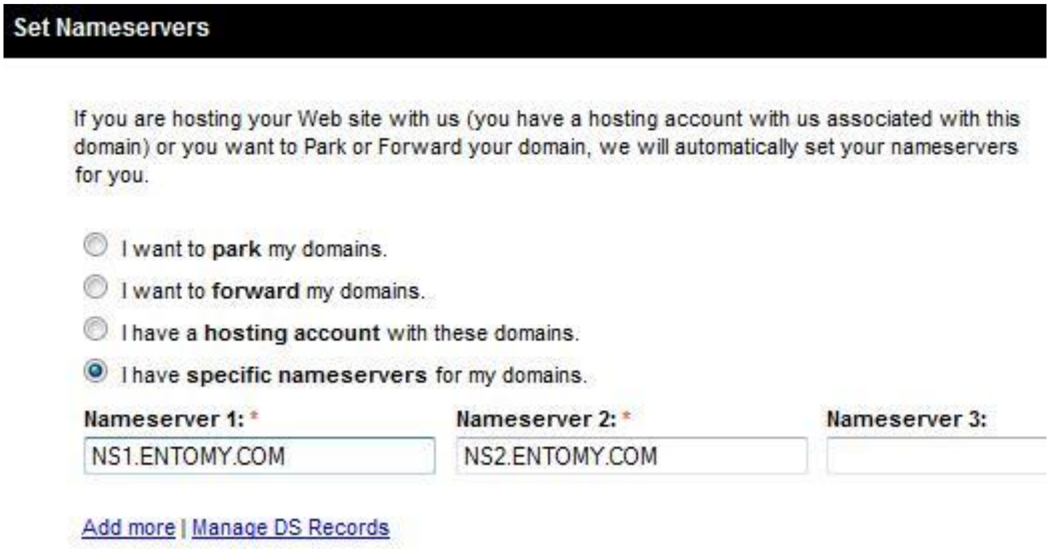


Nameservers

Nameservers: (Last Update 3/16/2011)
NS1.ENTOMY.COM
NS2.ENTOMY.COM

Click the “Set Nameservers” link. Select the “I have **specific nameservers** for my domains” radio button. Now enter the ns1 nameserver you just created (including your primary business domain name, like ns1.yourdomain.com) into the “Nameserver 1” box. Also enter ns2.yourdomain.com into the “Nameserver 2” box.

It should look something like the following image.



Set Nameservers

If you are hosting your Web site with us (you have a hosting account with us associated with this domain) or you want to Park or Forward your domain, we will automatically set your nameservers for you.

○ I want to **park** my domains.
○ I want to **forward** my domains.
○ I have a **hosting account** with these domains.
● I have **specific nameservers** for my domains.

Nameserver 1: * NS1.ENTOMY.COM
Nameserver 2: * NS2.ENTOMY.COM
Nameserver 3:

With the nameservers set to your new IP address you are done at godaddy.com. It will take a day or two for those new DNS entries to propagate throughout the Internet, so be patient.

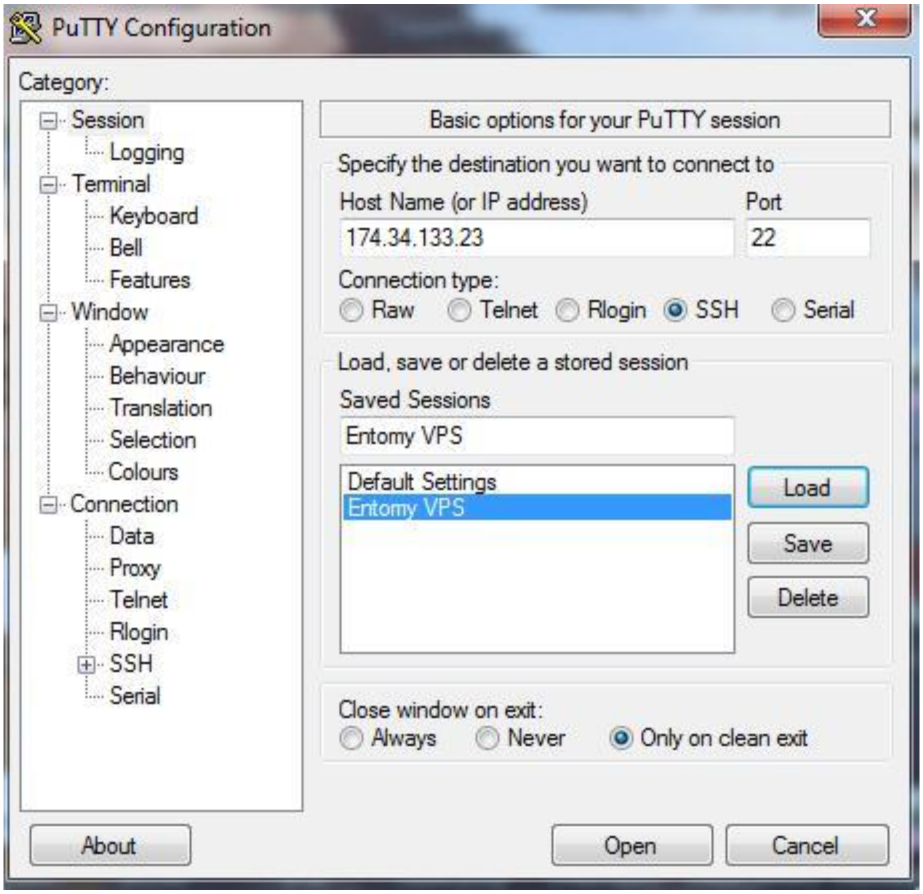
Gaining Access to Configure Your VPS

Primary configuration access will be by SSH terminal session, although we will also be installing webmin, which is a point & click method of administrating and configuring Linux.

The most common workstation platform is Windows, so this tutorial will assume that you will be accessing your Linux VPS from Windows. For SSH from a Windows workstation I normally recommend PuTTY as a SSH client. PuTTY is free and has the largest installed base of any Windows terminal client.

PuTTY is readily available for download by searching at Google.com, so download & install it now.

With PuTTY installed, open PuTTY and enter your VPS IP address into the Host Name box. Select SSH, which should change the port to 22. Give it a name (My VPS, for example) in the Save Sessions box and click the Save button, which will allow future sessions without the need to reenter your IP address. Now click the Open button to start your session.



Specify the destination you want to connect to

Host Name (or IP address): 174.34.133.23

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

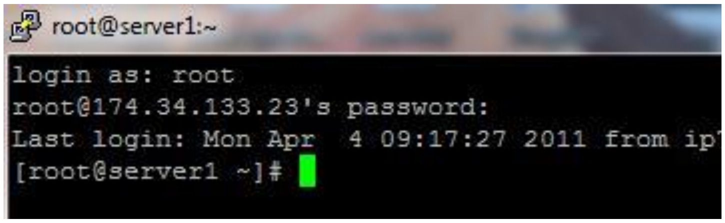
Load, save or delete a stored session

Saved Sessions: Entomy VPS

Default Settings
Entomy VPS

Close window on exit: ○ Always ○ Never ● Only on clean exit

Login as root, then enter the password you requested when you opened your VPS account. It should look like this.



```
login as: root
root@174.34.133.23's password:
Last login: Mon Apr  4 09:17:27 2011 from ip
[root@server1 ~]# 
```

*HINT: PuTTY supports pasting, not by Ctrl-v or Edit→Paste but by right-clicking the mouse. This feature will save you a mountain of work when following an install or configuration recipe with long text strings to enter. Simply copy (Ctrl-c) the text from the recipe, then right-click once in PuTTY to paste the text string. **Make sure to verify pasted text**, since it sometimes converts hyphens (-) to periods (.).*

ANOTHER HINT: Linux file names are typically very long, so that file names can be as descriptive as possible. While that's a handy feature for verifying that you have the correct file & version, typing long file names can be tedious. Linux therefore includes a feature called "tab completion", where you can type part of an existing file name, then press the tab key to have Linux enter the rest for you.

ONE MORE HINT: If you don't already know, everything in Linux is case sensitive. In other words, the filename test.txt is different from the filename Test.txt, and unlike Windows both of those filenames can coexist in the same directory as unique files. If you type a filename with the wrong capitalization, it will tell you that the file does not exist. Just so you know…

FINAL HINT: You can copy & paste from this document if you wish using Acrobat Reader, even though this is a pdf file. That will save you a ton of typing.

Now that you are logged-in to a terminal session the first thing you will want to do is download and install webmin, so that you can minimize your time on the command line. To do that, type these commands and press Enter after each line (I'll try to keep the latest version of webmin at that simplified download link, but since webmin automatically updates itself it's not that big of a deal if it's an older version).

```
# cd /tmp
# wget http://entomy.com/webmin.rpm
# rpm –ivh webmin.rpm
```

With webmin installed, go to your VPS IP address at port 10000. So if your IP address is 123.123.123.123 then your webmin address is:

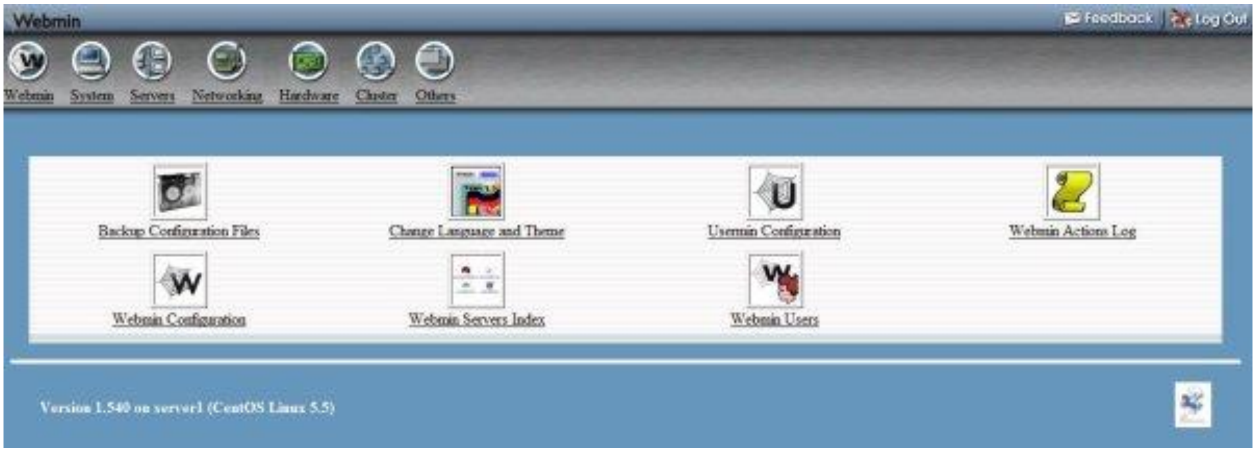
```
http://123.123.123.123:10000
```

Login to webmin as root using your password. You will see some options to do some updates, but leave those alone for now. It's better if you do your updates later. I'll tell when to do that.

Things are arranged somewhat differently with the various webmin themes. You can select any theme that you wish when you are done with this tutorial but just so we are both on the same page we'll both need to use the same webmin theme. To change themes, click on "Webmin Configuration" in the upper left, then open the Webmin Themes icon. On the Change Themes tab, change it to MSC.Linux Theme, then click the Change button. You may need to restart webmin to see the new theme.

```
# service webmin restart
```

Now reload the webmin page. Webmin should now look something like this.



Version 1.540 on server1 (CentOS Linux 5.5)

Installing the Kloxo Control Panel

Before Kloxo is installed SELinux needs to be disabled. SELinux handles file and directory security in a way that's incompatible with Kloxo, so it's imperitive that it's disabled before attempting to install Kloxo. To disable SELinux, login to PuTTY as root and issue the following command.

```
# setenforce 0
```

(That's setenforce and a zero, not a capital "O")

Now that we have SELinux disabled, we need to make sure that it doesn't come back on after a reboot. To do that we need to create a file called "config" in the /etc/selinux/ directory.

```
# echo 'SELINUX=disabled' > /etc/selinux/config
```

That takes care of SELinux.

To begin the Kloxo installation, open PuTTY and issue the following commands, pressing the Enter key after each line.

```
# cd /tmp
# wget http://download.lxcenter.org/download/kloxo/production/kloxo-
installer.sh
# sh ./kloxo-installer.sh --type=master
```

Note: The wget statement is long, so it wrapped to the next line. It needs to be entered as a contiguous string.

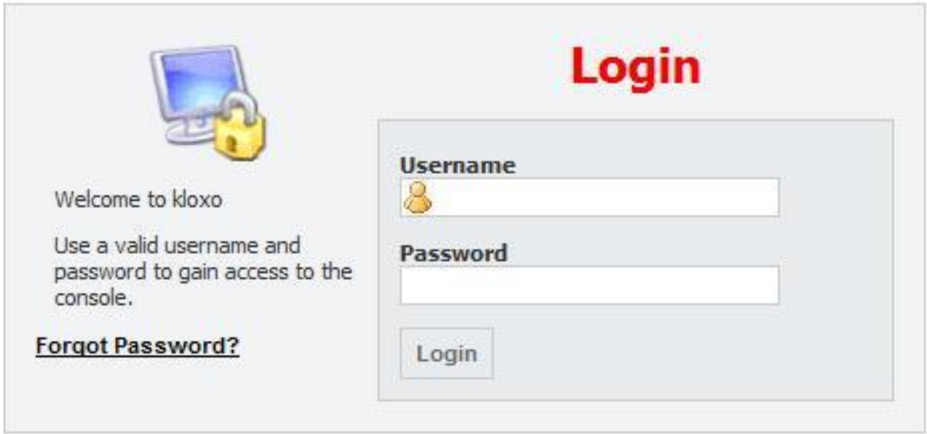
Upon entering the last command, the Kloxo installation will begin. It will ask you two questions, to which you should answer with "y" (less quotes, of course) to the first one and "n" for the second. The second question asks if you want to download Installapp applications, but Installapp is dangerously out of date. You will be disabling Installapp for your clients anyway.

Expect the install to take somewhere around 30 to 40 minutes to complete. It's a huge install, but is fully automated. Expect long pauses, sometimes for 5 or 10 minutes.

When the Kloxo install is complete you will get the command prompt back, and you will see instructions on how to access Kloxo in the few lines above the prompt. Basically, it will be your VPS IP address followed by :7778. So if your IP address is 123.123.123.123 your Kloxo login page will be.

```
http://123.123.123.123:7778
```

If you see the Kloxo login panel then you did everything correctly so far.



Welcome to kloxo

Use a valid username and password to gain access to the console.

Configuring Kloxo

Login with the username admin and the password admin. It will force you to change the administrator password first thing.

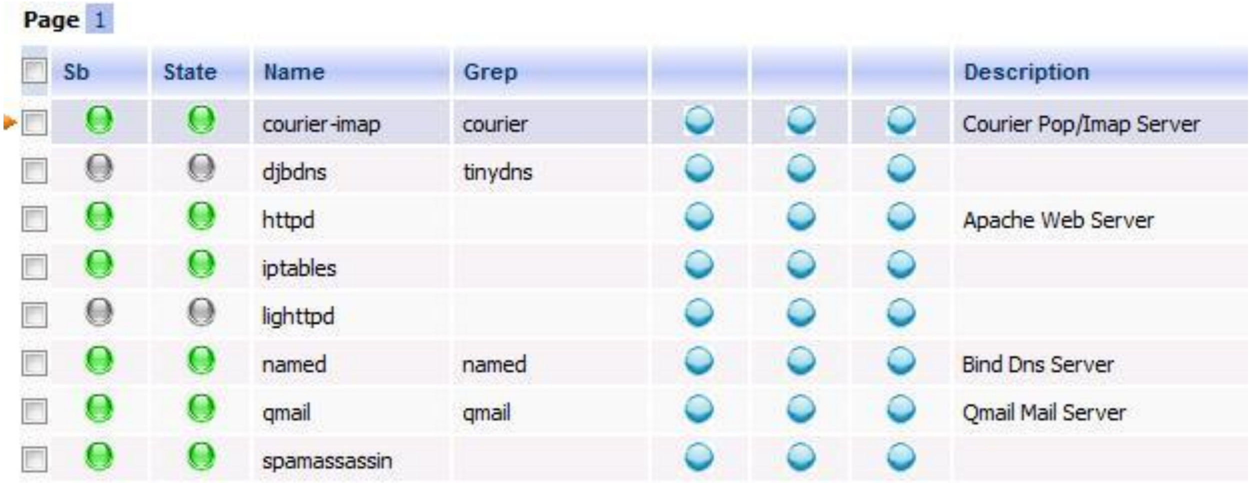
After you change your password and reach the main control panel, you will see a red warning at the top saying that you need to configure lxguard. Click the link to take care of that. Accept the default setting of 20, check the box that says you understand it, then click the Update button. Go back to the main page.

Again, you will see a warning at the top of the page, this time for the mail server. Click the link to take care of that. In the "My Name" box, enter your domain name with the prefix "mail". So if your domain name is yourdomain.com you would enter mail.yourdomain.com into that box. Go ahead and check the Enable Spamdyke box and leave the Enable Domainkey box checked. Those options don't use a lot of resources. However, the virus scanner will use about 100 mb of memory to run. Leave that unchecked for the time being. Also leave the rest of the settings in that panel the way they are. Click the Update button. Go back to the main Kloxo page.

Before we start adding things, let's make sure that all of the necessary services are running. As both a Kloxo and Linux administrator, it's important for you to understand that not all services are managed as the Linux administrator. Many services are controlled exclusively by Kloxo. For example, you probably won’t be able to start the courier-imap (pop email server) as the Linux administrator, but you can as Kloxo administrator. The two places to do that are:

Linux Service Administration – Webmin, by selecting the System icon at the top, then opening the “Bootup and Shutdown” icon.
Kloxo Service Administration – Main Kloxo admin panel in the Server:Linux box, then open the Services icon.

So lets get everything running that needs to be running by opening the Kloxo Services icon, as described above. What you want to see is in the image below, but you’ll probably see a lot of red at first.



Page 1

□	Sb	State	Name	Grep				Description
□	🟢	🟢	courier-imap	courier	🔵	🔵	🔵	Courier Pop/Imap Server
□	⚪	⚪	djbdns	tinydns	🔵	🔵	🔵	
□	🟢	🟢	httpd		🔵	🔵	🔵	Apache Web Server
□	🟢	🟢	iptables		🔵	🔵	🔵	
□	⚪	⚪	lighttpd		🔵	🔵	🔵	
□	🟢	🟢	named	named	🔵	🔵	🔵	Bind Dns Server
□	🟢	🟢	qmail	qmail	🔵	🔵	🔵	Qmail Mail Server
□	🟢	🟢	spamassassin		🔵	🔵	🔵	

The “State” column indicates whether a service is running. If it’s green it’s running, if it’s red it’s stopped, and if it’s gray it’s not installed. For any service that is red under “State”, click on the red button and see if it goes to green. It should. Do that for all stopped services. If for some reason you have difficulty getting services started in Kloxo, try with webmin under System in “Bootup and Shutdown.”.

The “Sb” column indicates whether the service is configured to start at boot time. You want to do that for all installed services, so click them to make them all green.

Djbdns (tinydns) button should be gray, and that’s fine because we are using named (BIND) instead, which is a much more advanced DNS server. Likewise lighttpd should be gray, and that’s fine because httpd (Apache) is used instead, which is a more advanced web server.

The above image shows that spamassassin is installed and running. Your spamassassin running lights will be gray, since spamassassin isn’t installed by default. That’s fine, since we’ll be using spamdkye as a spam filter instead. Just leave spamassassin uninstalled.

You may wish to reboot your server now to see if everything comes back up as it should.

Hint: You can reboot the system either from the VPS control panel, from Kloxo in the “Machine” box, using webmin (System category, Bootup and Shutdown icon), or by issuing this command in PuTTY.

```
# shutdown –r now
```

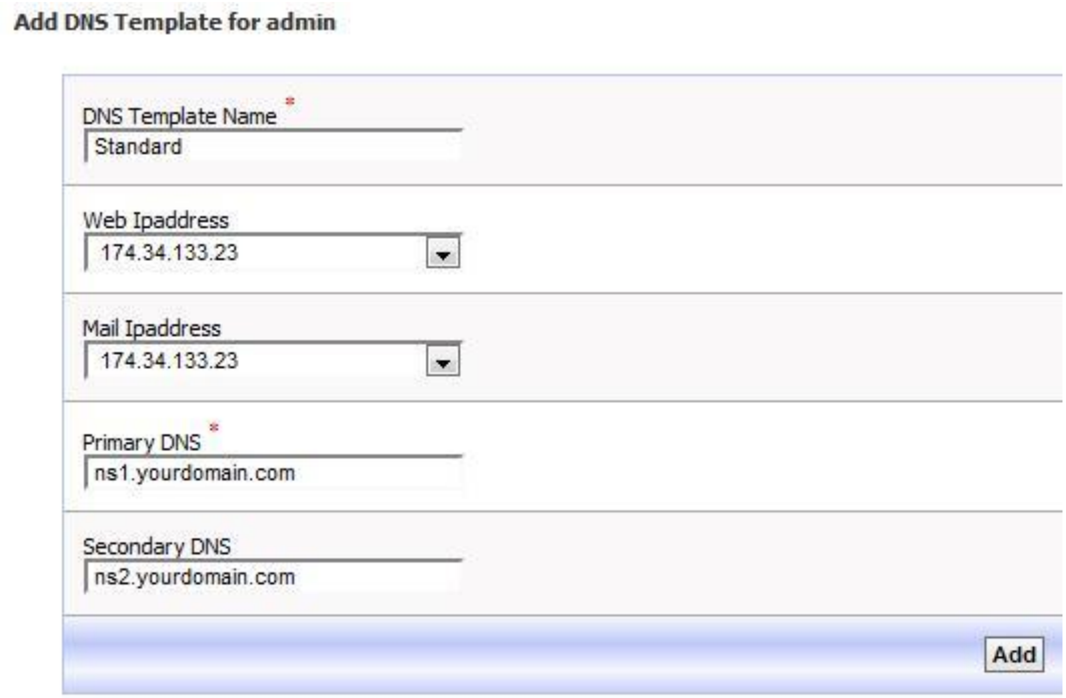
If everything came back up as it should, go to webmin and check some of the other critical services (click the System icon at the top of webmin, then open the “Bootup and Shutdown” icon. Pay particular attention to the following services.

Clamav – It will not be running because we specified to not use it in the email scanning settings. You may decide to use it in the future, but you should let Kloxo control clamav.
Crond – Should be running, start it if it is not running, and make sure it’s set to restart at boot time.
Mysqld – Should be running, start it if it is not running, and make sure it’s set to restart at boot time.

Let’s create your primary domain now. Log back in to Kloxo as admin.

```
http://youripaddress:7778
```

You will see some tabs near the top (Home, Domains, Subdomains, Mail accounts). Select the Domains tab. You should get a red warning that you haven’t setup a default DNS. Fill it out like the following image, substituting your primary business domain in place of yourdomain.com. Click Add.



Add DNS Template for admin

DNS Template Name *
Standard

Web Ipaddress
174.34.133.23

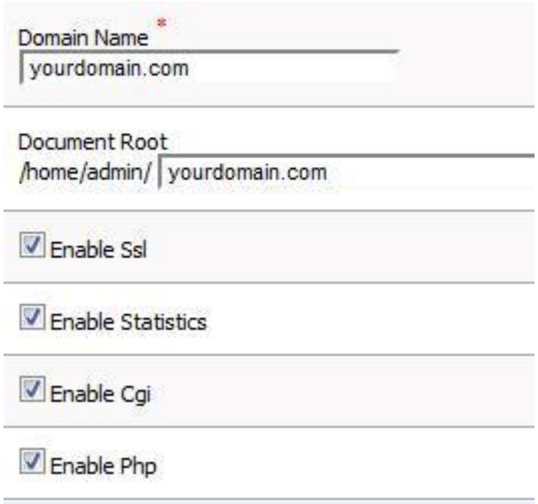
Mail Ipaddress
174.34.133.23

Primary DNS *
ns1.yourdomain.com

Secondary DNS
ns2.yourdomain.com

In the future, as you add reseller accounts, you will create DNS templates for the reseller’s business domain to give the appearance of private label branding for your resellers.

Once the DNS template is done, click on the Domains tab again. Now enter your domain name in the two boxes, as was done in the following image, then click Add.



Domain Name *
yourdomain.com

Document Root
/home/admin/ yourdomain.com

☑ Enable Ssl

☑ Enable Statistics

☑ Enable Cgi

☑ Enable Php

That created your web space, your DNS entries, and your FTP account, as well as a few other entries around your server. Since this is the first domain you've created you will want to verify that everything went well.

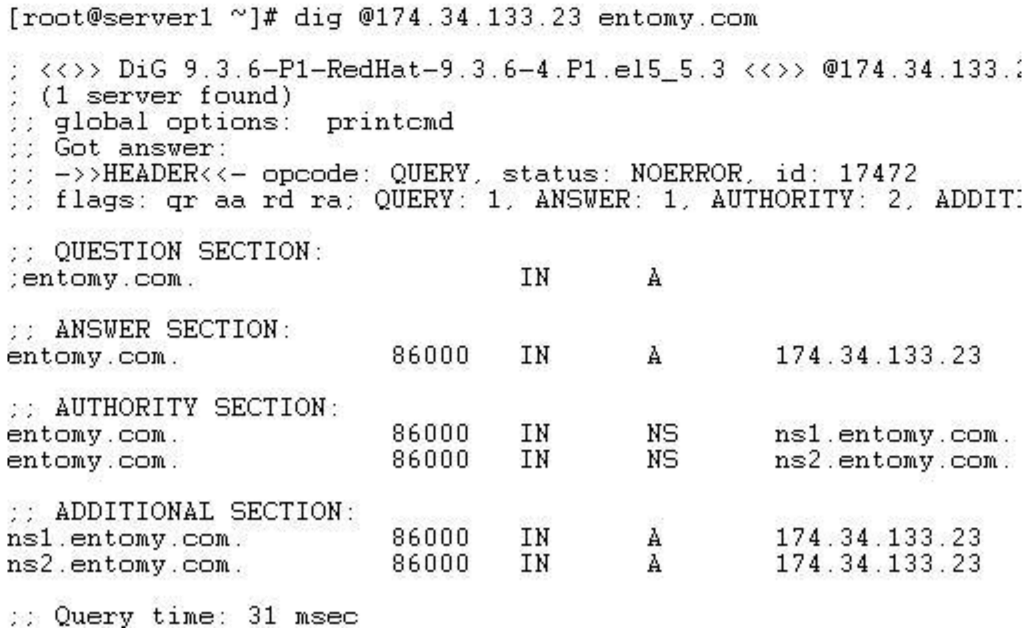
Verify that your web space exists. You can do that by using webmin. Click the Others button at the top, then select File Manager. Double-click the Home folder, then the admin folder. You should see a folder with your domain name. Open it, and you should see a cgi-bin folder, an image folder, and an index.html file. That's your web space. Client accounts will also be in the /home directory, but will have their own directory instead of being in the /home/admin directory.

Verify that your DNS entries were made. You can do that by clicking the Servers icon at the top of webmin, then clicking BIND DNS Server. You should see a Zone icon for your domain. Click on that icon, then click the A icon at the top left. That will show you all of the entries that are defined for your domain. If that's there then all is well. There will be a new zone created for each domain you add.

Verify that the DNS server is responding to queries about your domain. Do that by logging in to PuTTY as root, then issue the following "dig" command, substituting your IP address after the @ sign and your real domain name instead of yourdomain.com.

```
# dig @174.34.133.23 yourdomian.com
```

Output should look something like this.



```
[root@server1 ~]# dig @174.34.133.23 entomy.com

; <<>> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.el5_5.3 <<>> @174.34.133.
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17472
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDIT

;; QUESTION SECTION:
;entomy.com.                    IN      A

;; ANSWER SECTION:
entomy.com.             86000   IN      A       174.34.133.23

;; AUTHORITY SECTION:
entomy.com.             86000   IN      NS      ns1.entomy.com.
entomy.com.             86000   IN      NS      ns2.entomy.com.

;; ADDITIONAL SECTION:
ns1.entomy.com.         86000   IN      A       174.34.133.23
ns2.entomy.com.         86000   IN      A       174.34.133.23

;; Query time: 31 msec
```

Verify that your FTP account has been created and is working. You should try to connect with your FTP client to verify that. Use these settings to try to connect.

FTP Server: Your server IP address.
FTP Username: Your domain name
FTP Password: Your Kloxo admin password
Port: 21

There is currently a bug in the FTP server installation that might keep you from connecting. As a matter of course, you should enter the following commands to clear the bug.

```
# yum –y downgrade pure-ftpd
# /script/upcp
# /script/cleanup
# yum -y update
```

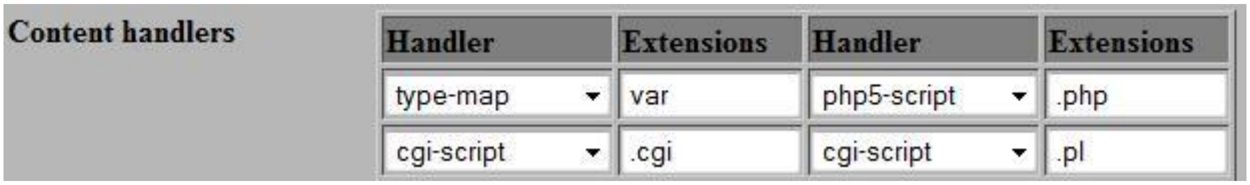
If you can connect, everything is fine. If not look in Kloxo admin in the Domains box and click on the FTP Users icon to reenter your authentication information.

Verify that your web page works. You can preview the web page in Kloxo. Click on the Domains tab, scroll down the page, and click on your domain name. That will open the domain control panel. Scroll down to the bottom to the Extra box, and click Dnsless Preview. If you see a panel that says your domain isn't configured properly then start a ssh session and issue the following command.

```
# sh /script/fixweb
```

Then reboot the server. That usually clears the error.

Next, you need to allow cgi scripts to run. I don't know why, but by default Apache has not been configured to know what to do with files with .cgi or .pl extensions. The result is that the web server will display .cgi files as if they were text files, so when you run a cgi script all you see is the code. To fix that you will need to add a couple of “handlers” to Apache. In webmin, click the Servers icon at the top, then click the Apache Webserver icon. Click on the Default Server icon. Click the MIME Types icon. In the “Content Handlers” section, add entries for .cgi and .pl extensions, selecting the handler type of “cgi-script” from the drop-down menu for both. When you get done it should look something like this.



Content handlers

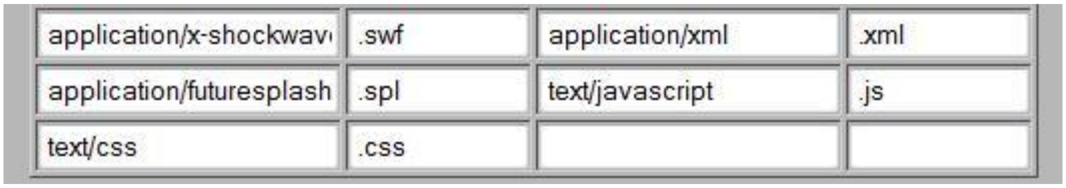
Handler	Extensions	Handler	Extensions
type-map ▾	var	php5-script ▾	.php
cgi-script ▾	.cgi	cgi-script ▾	.pl

Click the Save button at the bottom, then click the “Apply Changes” tab near the top. That should enable your CGI & perl scripts.

Now click the MIME Types icon again and add the following entries to the Extra MIME Types table, just to be sure that your web server will do most anything your customers want to do.

```
application/x-shockwave-flash    .swf
application/xml                  .xml
application/futuresplash         .spl
text/javascript                  .js
text/css                         .css
```

Those entries should look something like this.



application/x-shockwav	.swf	application/xml	.xml
application/futuresplash	.spl	text/javascript	.js
text/css	.css		

Hint: The table will only give you one row of open cells at a time. You will need to enter the top two, then click Save and click Apply Changes, then click MIME Types again. It will give you another row.

By default, Kloxo does not disable recursive DNS lookups. That can be a security problem, since anyone could use your DNS server for general DNS lookup use. Allowing recursive DNS lookups is a genuine risk, since a Denial of Service attack is possible if DNS recursion is not disabled.

To check that, login to Kloxo as admin. Click on the Domains tab and click on your primary business domain name (actually, you should get the same result with any of your hosted domains). In the domain control panel, go to the bottom in the Extra box and click on the Check Dns icon. It will take a minute before it displays. The only red warning you should see is a warning about recursive DNS lookups. If you see that error, you will want to take care of it.

To disable DNS recursion, go to webmin and click the Servers icon at the top. Click the BIND DNS Server icon, and then open the Edit Config File icon. Add the following lines of text to the top of the file.

```
options {
allow-recursion { localhost; };
};
```

HINT: The brackets in the above code are all braces (curly brackets).

That statement will disable recursive DNS lookups from anywhere except from within your own server. Click the Save button. To apply the change, click the Stop BIND tab, then click Start BIND. If you go back to Kloxo and click the Check Dns icon again you you should see that the recursive DNS error is now gone.

You should also disable logging in BIND. Even though you have disabled recursive lookups, BIND will still log recursive lookup attempts. During an attempted recursive attack the high-volume logging of each failed recursive lookup request can still slow the system, even resulting in the log file growing to hundreds of megabytes in a very short period of time. To prevent that it's recommended that logging be disabled during normal operation. To do that, add this code to the same file as above.

```
logging {
category default { null; };
};
```

Save, then restart named to apply.

This concludes the Kloxo configuration.

Billing Application

In order to properly manage a hosting service you will need a billing application capable of processing subscription accounts on a recurring billing basis. Most hosting operators want a billing application that can accept payments and create new accounts in real-time unattended. I'm not sure why that so important to them, since real-time provisioning of hosting accounts can't realistically make web pages available in real time. After all, it normally takes 48 hours for DNS changes to propagate anyway. But that's what people want, and such a free application is available so that's what we'll cover here.

We will be installing TheHostingTool in this tutorial. TheHostingTool is free and capable of adding & removing accounts & domains in Kloxo. Probably the biggest drawback is that it only accepts payments from PayPal. If that's what you want then it's an asset, but otherwise it could be a drawback. If you want to accept major credit cards directly then you will need to look for something else. But at the time of this writing no other billing application is known to be capable of automatically provisioning accounts in Kloxo.

You can still provision accounts manually, so you can accept payments from check draft customers. Check drafts are checks that you print and deposit. The way it works is that you ask a customer for his checking account information over the phone or in a sign-up page (customer name & address, bank name & address, routing number, and account number), then you print a check draft each month to cover subscription payments and deposit into your checking account. The customer does not need to sign the check draft, and there are usually no processing fees (except that sometimes banks charge 10 cents or so apiece if you exceed a certain number of items). In fact, check drafts are so inexpensive that you might consider offering a discount for drafting checking accounts instead of using PayPal or a merchant account. All banks accept check drafts, and you don't need a special account to deposit them. Any checking account will do, even a personal account. You can learn more about check drafts and find check draft software to help you create check drafts by Googling for more information.

That said, you can get TheHostingTool application software from the developer's website, thehostingtool.com, or from a link at my server. Currently the version of TheHostingTool for Kloxo is a branch of TheHostingTool called "Reworked." However, the people supporting the Reworked project are still at TheHostingTool website. It's a very active software community so I expect future versions to be updated quickly, to track Kloxo updates when necessary. These instructions are aimed at installing TheHostingTool for Kloxo 6.1.12. If Kloxo comes out with a newer version before I can update this tutorial I suggest you visit the forum at TheHostingTool's website.

Before installing the billing application you may want to wait a day or two until the nameservers propagate for your primary business domain. You won't be able to test and configure your installation until that happens. You can come back to this section of the tutorial at any time to install the billing application, since it operates independently from other server functions.

To begin the install, create a directory in your web site for the billing application using PuTTY (substitute your domain name for yoursite.com below).

```
# mkdir /home/admin/yoursite.com/billing
```

Then download the billing application

```
cd /home/admin/yoursite.com/billing
wget http://entomy.com/THT-1.3.5-Reworked.zip
```

Staying in the billing directory, issue the following commands to begin the install.

```
# unzip THT-1.3.5-Reworked.zip
# cd ..
# chmod -R 755 billing/
# chown -R admin:admin billing/
# cd billing/includes
# touch conf.inc.php
# chmod 666 conf.inc.php
# chown admin:admin conf.inc.php
```

Go to Kloxo admin panel and create a MySQL database. Name the database "billing" and use a password you can remember.

Go to the following address with a web browser, substituting your domain name for "yourdomain.com."

http://yourdomain.com/billing/install

Follow instructions for the installation. For database info, enter the info from the database you just created. The database username will be the same as the database name (billing, in the example). Accept localhost for hostname and tht_ for prefix. Click next, then create an administrative account (usually admin) with a password you can remember. Enter your email and your real name (or something like administrator). Click next to finish install.

Issue these Linux commands.

```
# cd /home/admin/yourdomain.com/billing
# rm -r install
(answer y to all questions)
# cd includes
# chmod 444 conf.inc.php
```

The install is now complete, but you will need to fix a few things to make it work properly. You will need to enable curl_exec in PHP by removing it from the disable_functions list. I suggest that you remove it from the main Kloxo admin control panel, and also in the domain control panel for the domain you are hosting THT on. To do that open the PHP Config icon, select the Advanced PHP Config tab, then remove "curl_exec," from the Disable Functions list. After changing that in both the main admin panel and in the domain panel, edit the disable_functions parameter in the following php.ini files. You will find it about 20% of the way down.

```
/etc/php.ini
/home/httpd/yourdomain.com/php.ini
```

Restart the httpd server.

```
# service httpd restart
```

There is a bug in the /includes/class_main.php script that prevents packages from being displayed. To fix that, you will need to patch that file according to this guide.

http://thehostingtool.com/forum/thread-2043-post-11253.html# pid11253

Finally, you will need to disable ssl connections to the control panel, since ssl connections won't work between THT & Kloxo. To do that, login to the billing application by going to this link, substituting your actual domain name.

```
http://yourdomain.com/billing/admin
```

Click General Settings. Click Security Settings. Change "Connect to WHM via SSL" to No.

Create an auxiliary admin user in Kloxo for TheHostingTool (THT from now on) to use for connecting to Kloxo. Login to Kloxo and click the Auxiliary Logins icon and create a new login. To use that login you will always need to use a suffix of ".aux". For example, if you create a login name of tht, then the username you will need to use in THT will be tht.aux.

In THT, create a server. To do that, click Servers from the admin menu, then click Add Server. Call your Kloxo server Master. Enter the Kloxo IP address. Enter 7778 for both the reseller and CP ports. Enter the name servers you want your customers to use, one per line. Username will be the auxiliary login we created in the previous paragraph. Enter the name if the DNS template you normally use to create domains in Kloxo, and be sute to include the .dnst suffix. Set email to Yes, and set the server type to

Kloxo.

Create a test package in Kloxo. Click the Resource Plans icon, then create the test plan. Notice the name after you create it. If you names the plan "test" then it should look something like:

```
test___client-admin
```

You can now create a test package in TheHostingTool. Click the Packages link, then Add Package. The name might be test, and the Backend will be the full name of the test package you created in Kloxo (test___client-admin). Test the package by making it free, and select Master as the server.

You can now test the signup link. It should create a new client and domain for you in Kloxo.

Without a doubt, the biggest sticking point in making THT work properly is in making sure that curl_exec is enabled. That's particularly true since curl_exec is disabled by default when Kloxo is installed. The classic symptom of curl_exec being disabled is seeing the following error in Step 5 of the signup process.

```
An error has occurred. Please inform your system administrator.
```

The problem is that a number of configuration glitches can generate the same error message. So before you go off on a wild goose chase searching THT for configuration glitches, you should first make ABSOLUTEY SURE that curl-exec is enabled. To do that, create a file named testcurl.php and paste the following contents.

```
<?php
echo '<pre>';
var_dump(curl_version());
echo '</pre>';
?>
```

Save the file, then run it as a php script.

```
# php testcurl.php
```

If curl_exec is not enabled then you will see this error.

```
Fatal error: Call to undefined function curl_version() in testcurl.php
on line 2
```

If you see that error then you still need to remove curl_exec from the disabled list from one of the php.ini files. Don't forget to restart httpd to make sure that php changes are applied.

If curl_exec is enabled properly then you will see a large output with the curl parameters. When you have curl_exec enabled you should be able to get past Step 5. If the Step 5 problem persists then you probably have a problem authenticating with your hosting package or your Kloxo aux login credentials.

There are a lot of other problems that can cause the step 5 error. Usually the culprit is a leading or trailing space in a setting. If you continue to get the error check for leading & trailing spaces in the aux user, package name, server IP address, and in just about everything else. It's usually something minor.

Before you can start charging for packages you will need to edit your PayPal information. To do that, click Paid Configuration. Enter your PayPal email address and select "live" for mode.

Check your cron jobs, since the install should have added a command to run /includes/cron.php. Make sure it is active and test run the script to verify that the path is correct. It set mine to run every 5 minutes, which is a whole lot more than needed. Really, once per day should be enough. I set mine to run once per hour.

Create your actual commercial packages, and direct customers to the signup links.

Problems and advanced questions should be posted at the THT forum for the Reworked application.

http://thehostingtool.com/forum/forum-46.html

This concludes the installation and configuration of the billing application.

System Update

As with all operating systems, updates are critical to both bug-free operation and security. Linux is no exception. One thing I'll promise you; if you don't apply updates regularly you'll eventually get hacked and lose your server through some vulnerability. Fortunately there is a free automated process available to update CentOS Linux packages for us. To invoke the update routine, use PuTTY to login as root and issue the following command.

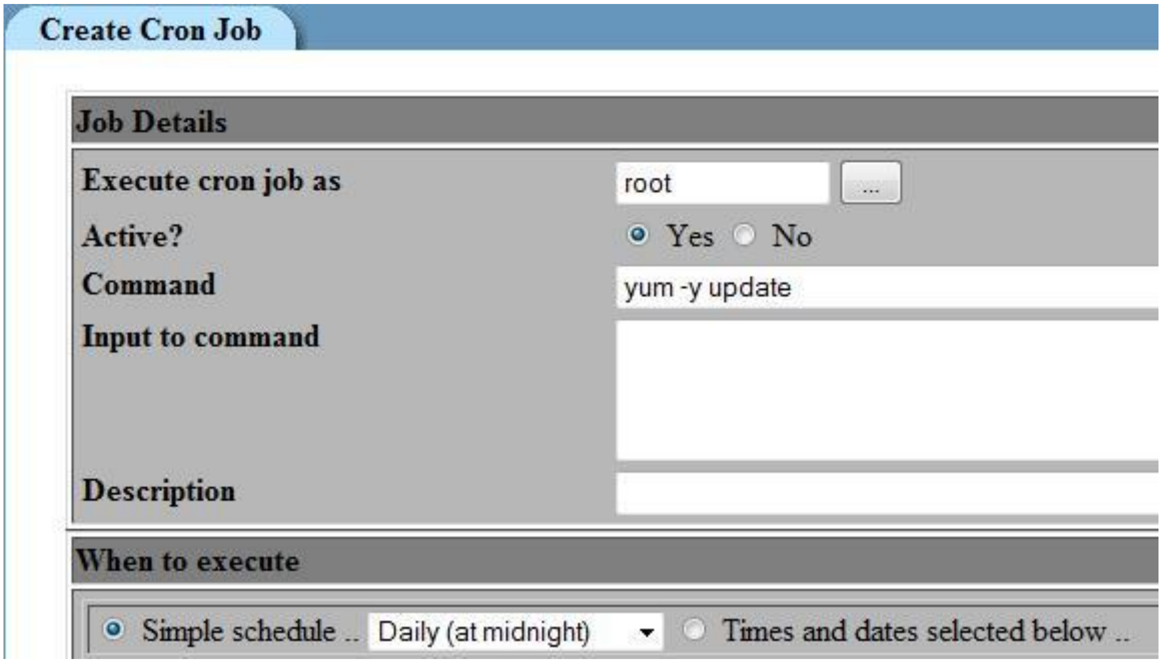
```
# yum –y update
```

Hint: The –y parameter tells yum to answer yes to all conditional questions and continue the install automatically. You can leave it out and answer those conditional questions manually.

Don't be surprised to find that there are 100 or more packages in need of updating. Be patient, it may take 10 minutes or more. When it's finished you should reboot your server, since a reboot will be necessary if there was a Linux kernel update. Otherwise the kernel update can't take effect.

You should now setup a "cron job" to automatically update Linux in the future automatically. Cron is a Linux application that will execute commands at designated times. We will use cron to execute an update command similar to the one we issued above. You can set that up in webmin.

Click the System icon at the top of webmin, then click the Scheduled Cron Jobs icon. Click the text link "Create a new scheduled cron job." Fill it out like this.



Create Cron Job

Job Details

Execute cron job as: root

Active? ● Yes ○ No

Command: yum -y update

Input to command:

Description:

When to execute

● Simple schedule .. Daily (at midnight) ○ Times and dates selected below ..

You will need to execute the update command as root, as I specified above, in order for the command to have the necessary permissions.

Under "When to execute", select "Simple schedule". In the drop-down menu, I selected "Daily (at midnight)". Weekly updates would probably be satisfactory, but it doesn't hurt anything to do it daily. By the way, note that midnight to the server will most likely be midnight Greenwich time, not local time, so keep that in mind if you are watching the cron log to verify that the command is running.

Click the Create button at the bottom. You should see the new cron job in the list, and also see that it's active.

You should start a SSH session perhaps once a month to run "yum update", just to verify that it's running on cron okay. Problems can arise. Checking yum activity in the /var/log/yum.log file isn't enough, since it only tracks packages that were installed or updated. You really need to test yum yourself from time to time. Don't turn your back on it for 6 months or a year assuming that everything is fine. You don't want to risk accumulating vulnerabilities in your system.

Memory Reduction

There are a lot of things you can do to save memory, but you can easily cut your memory usage in half by adjusting mysqld to reasonable settings. If you don't do that your server performance will really suffer, and you'll have difficulty with some tasks that require a lot of resources (like some updates might). You can expect mysqld to consume about 280 mb of memory as delivered, but you can easily lower that figure to 60-80 mb while still making ample database resources available to your subscribers.

To lower mysql memory usage login to webmin. Click the Others icon at the top of webmin and then click the File Manager icon. Open the /etc directory and click once on the my.conf file to select it. Click the edit key. Paste the following entry immediately below [mysqld], which should be at the top of the file.

```
skip-innodb
```

Save & close. Apply changes by restarting mysqld.

```
# service mysqld restart
```

Your entire server should now be running at around 200 mb of memory.

Basic Protection Against Denial of Service (DoS) Attcks

A Denial of Service (DoS) attack is a situation where system resources are deliberately drawn by vandals to the point where the server can't service legitimate users. In its most rudimentary form, web pages are requested over and over, perhaps hundreds of times per second, by multiple sources at the same time. Most web hosts are not high risk targets for DoS attacks, since websites of limited popularity are not particularly interesting to vandals who launch DoS attacks. Still it makes sense for the prudent server administrator to have some kind of protection in place.

Most data centers have a sophisticated strategy to protect their customers from DoS attacks, so most data center customers assume that DoS protection is in place. They're probably correct most of the time. However, there is no way to know what protections, if any, are being used in your data center. Your data center isn't going to tell you what protections they have, since making their security policy public would provide a roadmap to exploit their vulnerabilities. Whatever protections they have in place are best kept secret, so you can't blame them for doing so. That being the case, you should have your own basic protection in place that at least can discourage unsophisticated attacks.

The kind of protection that works best for preventing request-based DoS attacks is a behavior-triggered IP address blocker. So if an IP address exhibits suspicious behavior then that IP address is temporarily blocked automatically. If suspicious behavior persists during the blocking period, then the blocking period is extended.

One free application that provides behavior-triggered DoS protection is "mod_evasive", which is an Apache add-on module. With mod_evasive you can set the blocking policy to any threshold settings that suit you. You can also have mod-evasive email you when there is any suspicious activity going on. I strongly recommend that mod_evasive be installed.

Before mod-evasive can be installed the Apache development package needs to be present in the system. To do that open PuTTY, login as root, type the following command and press Enter.

```
# yum –y install httpd-devel
```

With those utilities installed you are now ready to download and install mod_evasive. Using PuTTY again as root, issue the following commands. Press Enter after each command.

```
# cd /tmp
# wget http://entomy.com/security/mod_evasive_1.10.1.tar.gz
# tar xfz mod_evasive_1.10.1.tar.gz
# cd mod_evasive
# /usr/sbin/apxs -cia mod_evasive20.c
```

That last command might take a minute or so to execute.

Before applying mod_evasive to Apache we need to enter policy parameter settings. Do that with webmin. Click the Others icon at the top of webmin and then click the File Manager icon. On the left side, navigate to the following directory.

```
/etc/httpd/conf/
```

Click on the httpd.conf file to select it, then click the Edit tile. That's a large file, but maybe 20% of the way down you should see a line that says this:

```
LoadModule evasive20_module /usr/lib/httpd/modules/mod_evasive20.so
```

That line was added to httpd.conf when mod_evasive was installed. Right below that line you will insert the policy parameters, which you can modify to your liking. Here is the text to insert.

```
#
# Settings for mod_evasive
#
<IfModule mod_evasive20.c>
   DOSHashTableSize 2048
   DOSPageCount 10
   DOSSiteCount 50
   DOSPageInterval 1
   DOSSiteInterval 1
   DOSBlockingPeriod 30
</IfModule>
```

Here is an explanation of the variables:

DOSHashTableSize – This setting specifies the size of the hash table (the table that keeps track if IP address activity). Increasing this number will provide faster performance by decreasing the number of iterations required to get to the record, but will consume more memory for table space. You should increase this if you have a busy web server. The default setting is 1024, but I use 2048 to improve performance.

DOSPageCount & DOSPageInterval – These settings work together to set the page request violation threshold. With the settings above, if a web page is requested by the same IP address 10 times within 1 second then the IP address is blacklisted.

DOSSiteCount & DOSSiteInterval – These settings work together to set the site object request violation threshold. With the settings above, if any single object (an image, for example) is requested by the same IP address more than 50 times within 1 second then the IP address is blacklisted.

DOSBlockingPeriod – This is the number of seconds that an IP address is blacklisted for after a violation, in this case 30 seconds. Some server administrators set this pretty high, like 5 minutes (300 seconds), while others set it as low as 10 seconds. However, if additional requests are made during the blacklisting period then the blacklisting period is extended, so long blacklisting periods don't seem necessary to me. I use 30 seconds, but use your own judgment.

To apply mod_enable, use webmin. Click the Servers icon at the top, then click on the Apache Webserver icon at the top. Click the "Apply Changes" tab near the top.

There is a utility that can test mod_evasive included in the install package. You can run it by opening PuTTY and connecting to your server as root. Enter these commands.

```
# cd /tmp/mod_evasive
# perl test.pl
```

That will hammer your server for a few seconds. You should see some error 403 message returned. If not, lower your DOSPageCount parameter a little and try again.

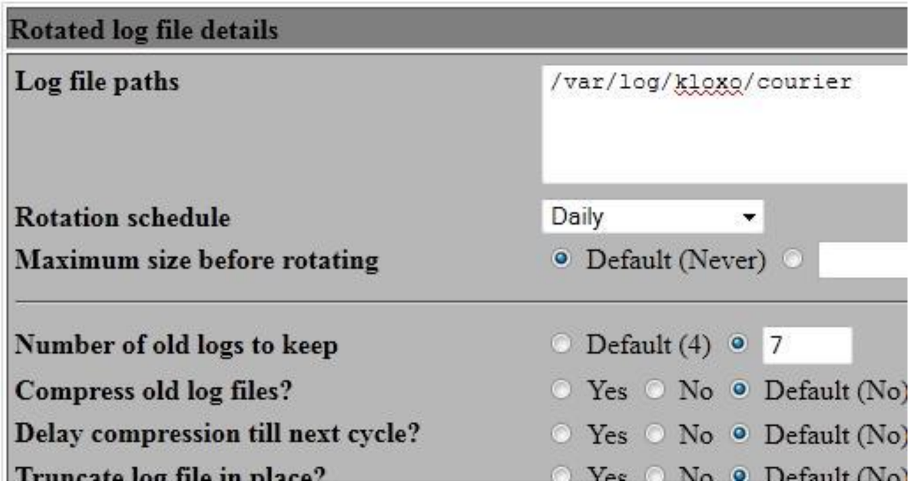
Log Rotation

Maintaining log files may seem like a minor housekeeping chore, but it can be critical to your disk space. Log files that aren't on a rotation schedule can grow to become quite large. A rogue log file that's growing out of control can even fill your entire disk space, which could shut down a server. To prevent that from happening you need to setup log rotation for any log files that aren't already setup, then browse the log directory (/var/log/) from time to time to make sure that there aren't any new rogue logs that are growing.

By default, the Kloxo log files aren't rotated, but they need to be. Mostly the logs involve email and FTP activity, which can become very large in a short amount of time with enough subscribers. To setup a rotation schedule for the Kloxo log files, use webmin. Click the Others icon at the top and then click the File Manager icon. Navigate to the following directory.

```
/var/log/kloxo
```

You should see four log files in that directory. You will need to setup a log rotation schedule for each. To do that click the System icon at the top of webmin, then click the Log File Rotation icon. Click the "Add a new log file to rotate" link at the top. Fill it out like this for the courier log file.



Rotated log file details

Log file paths	/var/log/kloxo/courier
Rotation schedule	Daily
Maximum size before rotating	● Default (Never) ○
Number of old logs to keep	○ Default (4) ● 7
Compress old log files?	○ Yes ○ No ● Default (No)
Delay compression till next cycle?	○ Yes ○ No ● Default (No)
Truncate log file in place?	○ Yes ○ No ● Default (No)

As is indicated above, fill-in the “Log file path” with the fully-qualified location of the file. Change the “Rotation schedule“ from weekly to daily so the mail logs don’t get too large. Also change the “Number of old logs to keep” to 7, so you’ll have at least a week of back logs. Click Save.

Do that for all four Kloxo log files. The one log that’s critical is the smtp log, since operating your own email server comes with certain responsibilities. If you get a spam abuse report you’ll need to access the smtp log to deal with it. Failure to deal with spam reports can get your mail server blacklisted. Don’t let it happen! Save at least one week of smtp logs so you can research abuse reports.

IMPORTANT: As a side note, and this is totally unrelated to log rotation, but you MUST monitor your postmaster@yourdomain.com email address if you operate an email server. That’s where abuse reports are going to be sent. ***If you fail to reply to abuse reports sent to your postmaster account in a timely manner then your mail server will be blacklisted.*** *You might consider forwarding your postmaster mail to an email account that you check all the time. That can be setup on Kloxo.*

Browse your /var/log/ directory, and it’s subdirectories for any other log files that might need to have rotation setup. Make sure that all logs files are on rotation.

You should return to /var/log/ from time to time to look for large log files. Pay attention to the file size column. Look for any file size reported in MB instead of KB, as they might be large. This is necessary since installing new applications will deposit new log files in /var/log/ that may not have rotation setup automatically.

Scanning for Rootkits

Linux is vulnerable to becoming infected with rootkits. Basically, a rootkit is a virus that enables privileged access to a server without the knowledge of the system administrator. Rootkits are common, since they are an integral part of DoS attacks. Before a DoS attack can be launched, the vandal will first look for “weak” systems where he can install rootkits. Those systems will be used as launch points for an organized Distributed Denial of Service attack (DDoS attack), where the attack is launched by multiple servers at the same time. You can get into a lot of trouble with your VPS host if you consume his bandwidth by unknowingly taking part in a DDoS attack because you have picked-up a rootkit. Therefore, you need to do what you can to keep rootkits out of your system.

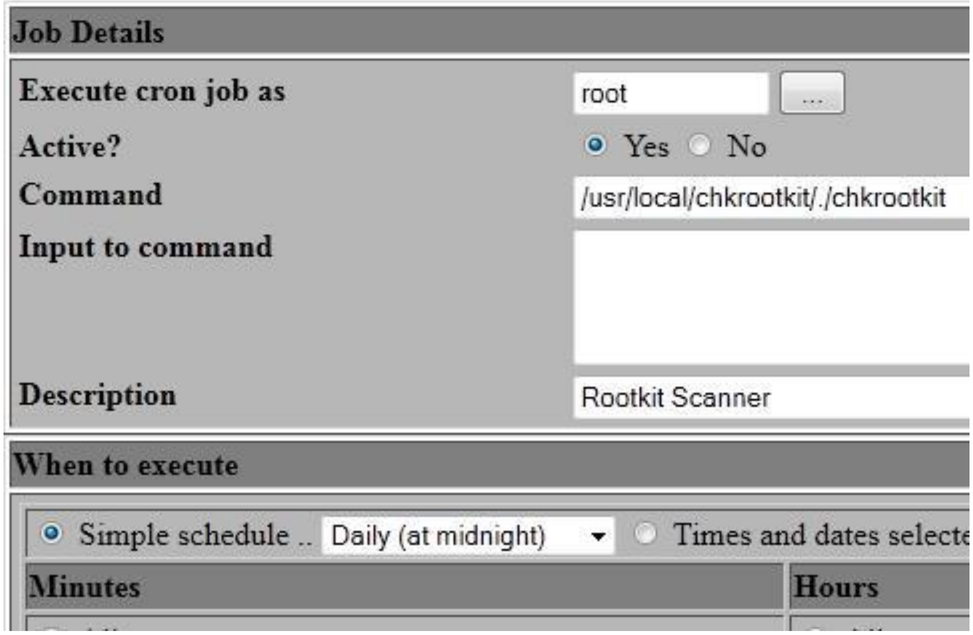
Unfortunately, there is currently no real-time rootkit blocker for Linux. That being the case, you should scan your system for common rootkits daily so you can detect any infections that you might have picked-up. I suggest using chkrootkit for a scanning utility. To download and install chkrootkit, open PuTTY and execute the following commands, pressing Enter after each line.

```
# cd /tmp
# wget http://entomy.com/chkrootkit.tar.gz
# tar –xzvf chkrootkit.tar.gz
# mkdir /usr/local/chkrootkit
# mv /tmp/chkrootkit*/* /usr/local/chkrootkit
# cd /usr/local/chkrootkit
# make sense
```

That last line may take a minute to run. To test run the scanner issue the following command.

```
# /usr/local/chkrootkit/./chkrootkit
```

You will see quite a lot of output as the scan is done. Expect the scan to take 1 to 2 minutes. That scan should be done every day to be sure that you don’t have any problems. To run chkrootkit automatically you should create a cron job. Using webmin, click on the System icon at the top and then click the Scheduled Cron Jobs icon. Click the “Create a new scheduled cron job” link. Fill out the form like this.



Job Details

Execute cron job as	root ...
Active?	● Yes ○ No
Command	/usr/local/chkrootkit/./chkrootkit
Input to command	
Description	Rootkit Scanner

When to execute

● Simple schedule .. Daily (at midnight) ○ Times and dates selecte

Minutes	Hours

You will execute that job as root. Enter the command string exactly like this.

```
/usr/local/chkrootkit/./chkrootkit | mail -s 'CHKROOTKIT' you@email.com
```

Of course you need to change “you@email.com” to your real email address. The entry ‘CHKROOTKIT’ is the subject line of the email to be sent to you, so you may wish to make the subject more descriptive, particularly if you operate more than one server that will be doing this. Of so, have the text between the single-quotes specify your server name, such as 'CHKROOTKIT Daily Output for server.com'.

You can give it a Description (I used Rootkit Scanner), but you can leave it blank if you want. Under "When to execute" select the "Simple schedule" radio button and then select "Daily (at midnight)" from the drop-down menu. Click Save. If you wish to test run the job, click on the command to reenter the job editor and click the "Run Now" button at the bottom. You won't see any output generated in webmin if you test run the cron job, but all of the output will be sent to you in an email message. Review the output each day when you get it so you can deal with any problems that it might detect.

Installing a Firewall

A server firewall is very different from a workstation firewall. In addition to the ability to block unused ports, a server firewall is also capable of detecting and blocking suspicious behavior. We already installed mod_evasive, which monitors for certain types of suspicious behavior, but a server firewall can do things that mod_evasive can't do. The firewall can track users as they access the system, looking for suspicious behavior by what visitors are doing, and can also block IP addresses of known network abusers by accessing listing services. Using a server firewall is highly recommended.

However, be aware that installing and configuring a server firewall is an involved process, and if you really foul it up you can find yourself locked-out of your own system. Please follow all of the instructions below very carefully to avoid that.

The firewall I'm recommending is the Advanced Policy Firewall (APF), which is a software firewall that you can download and use indefinitely for free. APF has become the security standard for hosting server firewall applications.

Before installing APF, verify that you have all of the iptables utilities installed, and that you have the latest versions. To do that, use PuTTY login as root. Issue the following command, then press Enter.

```
# yum install iptables*
```

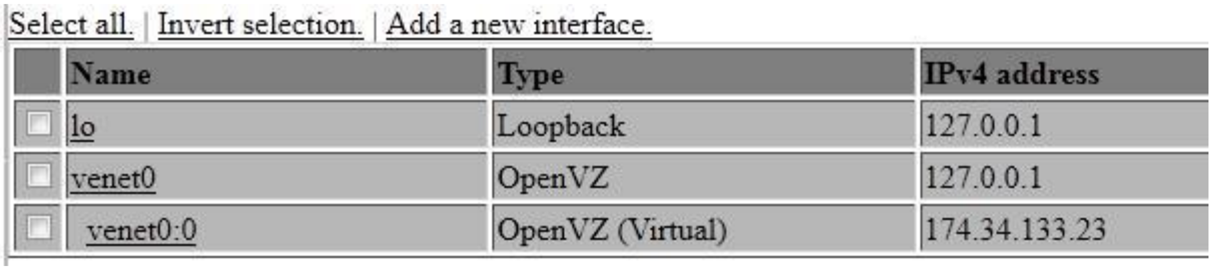
To download and install APF, issue the following commands and press Enter after each line.

```
cd /usr/local/src
wget http://www.rfxn.com/downloads/apf-current.tar.gz
tar -zxf apf-current.tar.gz
cd apf-9.7-2
./install.sh
```

Note that if a future version of APF comes out that the line that says "cd apf-9.7-2" may need to be adjusted to reflect the newer version directory name.

Before we begin editing the APF configuration file we need to take care of a few things. First, we need to find out the designation of the network interface (i.e., the name of your network adapter). If you have a dedicated server machine that will normally be eth0 or eth1. If you have a VPS then you will have a virtual network adapter, which has a designation of something like venet0. To find out the exact designation of your network interface use webmin. Click the Networking icon at the top, then click the

Network Configuration icon. Finally, click the Network Interfaces icon. You should see something like this.



Select all. | Invert selection. | Add a new interface.

	Name	Type	IPv4 address
☐	lo	Loopback	127.0.0.1
☐	venet0	OpenVZ	127.0.0.1
☐	venet0:0	OpenVZ (Virtual)	174.34.133.23

Logically, you would think that the interface associated with the Internet IP address (in this case venet0:0) would be correct, but it won't work. The correct network interface above is venet0, even though it's associated with the loopback address. Remember that device name for future reference.

Since we'll be setting ports for the firewall, now is a good time to change SSH to a non-standard port. Rather than use port 22, it's best to change it to some unused port, like port 522 (you can use any port number you want, as long as it's not currently in use). To see which ports are currently in use, issue the following command with PuTTY.

```
# netstat -nlp
```

We'll use 522 for non-standard SSH port for this tutorial. To change that port, go to Kloxo and login as admin. Look in the Security box, and click the SSH Config icon. Enter 522 in the SSH Port box and click Update. From now on you will need to use port 522 to connect with PuTTY. Our port entries in the firewall will reflect port 522, and port 22 won't be entered at all.

While you're at it, you might take this opportunity to also change the webmin port from 10000 to some non-standard port. Any unused port will do. To change the webmin port edit the following file.

```
/etc/webmin/miniserv.conf
```

When you edit that file you will need to change the port number in two places; the "port=" and the "listen=" variables. Change both of those from 10000 to the unused non-standard port you wish to use, then be sure to replace that new non-standard port where you see port 10000 in the instructio0ns below (you'll see port 10000 in three places).

We also need to take care of a communication port problem with pure-ftp. If you don't take care of this then your FTP client won't be able to list file and directories with your FTP client in passive mode. To fix that, use webmin. Click on the Others icon at the top, then click the File Manager icon. Navigate to the /etc/pure-ftpd/ directory, then click on the pure-ftpd.conf file once to select it. Click the Edit button. Look for the following line.

```
# PassivePortRange          30000 50000
```

Remove the hash mark from in front of that line so it isn't commented any more. Click the Save & Close button at the bottom. That change will open up the port range of 30000 through 50000 for use in passive mode FTP communications. To apply that change to pure-ftpd, restart xinetd.

```
# service xinetd restart
```

With pure-ftpd restarted you are ready to begin the firewall configuration. Using webmin, click the Others icon at the top, then click the File Manager icon. Navigate to the /etc/apf/ directory and click on the conf.apf file to select it. Click the Edit tile.

The first entry that isn't commented-out with a hash mark is the DEVEL_MODE parameter. That is extremely important, and it's critical that you don't change that now. If it is set to "1" then it's in testing mode, which is what you want. The setting of "1" will tell the firewall to run for 5 minutes then shut off. It does that in case you did something that locked you out of your own machine. In DEVEL_MODE="1" the worst case result of a screw-up is that you have to wait 5 minutes to get back in. LEAVE THAT SETTING AT "1" UNTIL YOU ARE POSITIVE THAT YOU HAVE APF EXACTLY THE WAY YOU WANT IT.

The next thing you will change is IFACE_IN and IFACE_OUT. Those settings are where you put your network interface name that we looked up with webmin ("venet0" in the VPS example). The same name goes in both settings.

The next thing we'll do is to set the ports for ingress and egress. Look perhaps two-thirds of the way down that page for the parameter IG_TCP_CPORTS. Here is what I have for that one, and the next several port settings. Edit those settings to match what I have below. (Note the "30000_50000" port range syntax used for the passive mode FTP ports.)

IG_TCP_CPORTS="21,25,53,80,110,143,522,953,993,995,3306,7777,7778,7779,10000,30000_50000"
IG_UDP_CPORTS="21,53,80,10000"
IG_ICMP_TYPES="3,5,11,21,30,8"
EG_TCP_CPORTS="21,522,25,26,53,80,110,113,443,465,873,2098,3306,7777,7778,7779,10000"
EG_UDP_CPORTS="20,21,37,53,80,873"

Everything seems to work fine for me with those settings. It might be that some of those ports aren't necessary, but it works. Note that port 522 appears in the IG_TCP_CPORTS but port 22 does not. Since we changed the SSH port we also need to change the following setting to port 522, as shown below.

```
HELPER_SSH_PORT=”522”
```

With the ports edited, search the conf.apf file for the following parameters and change them to the indicated settings (Hint: there is a Find utility in the editor). You will see some pretty descriptive documentation above each setting, so you can understand why each variable is being changed and what the settings mean.

```
RAB="1"
RAB_PSCAN_LEVEL="3"
TCR_PASS="0"
DLIST_PHP="1"
DLIST_SPAMHAUS="1"
DLIST_DSHIELD="1"
DLIST_RESERVED="1"
SET_MONOKERN=”1"
```

To avoid a large number of errors during APF startup, place a hash mark in front of the following two lines to comment them out, as indicated below.

```
# BLK_P2P_PORTS="1214,2323,4660_4678,6257,6699,6346,6347,6881_6889,6346,7778"
# BLK_PORTS="135_139,111,513,520,445,1433,1434,1234,1524,3127"
```

(Actually it's a good thing we comment those lines, since the Kloxo port (7778) is blocked in the default APF configuration.)

That pretty-well finishes the configuration, so click the Save button at the bottom of the editor. Now it's time to test the firewall. Using PuTTY, issue the following command.

```
/usr/local/sbin/apf –r
```

That command will restart APF to apply the changes to the configuration file, and will run the firewall for 5 minutes. You should test everything you can think of; websites, webmin, Kloxo, your FTP client – everything. If you need more time to test then issue the above command again to get another 5 minutes. If you have any trouble it's usually because a TCP or UDP port needs to be added to ingress or egress. Do what you need to do to get things working properly.

When you are satisfied that the firewall is the way you want it, return to the config.apf file and edit the DEVEL_MODE setting so the firewall won't shut off after 5 minutes. Change that setting to zero, as indicated below.

```
DEVEL_MODE="0"
```

Save the conf.apf and close the editor. Issue the restart command again to apply the final change.

To be sure that APF will restart on a server reboot, look in webmin. Click the System icon on the top, then click the Bootup and Shutdown icon. You should see APF at or near the top, and see if it's configured to start on boot.

To learn more about the capabilities and options in APF there is a comprehensive readme file located in your server at the following location.

```
/etc/apf/doc/README.apf
```

Securing the System

Spend a few days becoming accustomed to your new server, browsing the various features of webmin. But when you are done looking around you should recognize that there are certain security risks in having SSH and webmin running. Therefore, within a few days you should consider locking down the server by restricting access to both.

To disable webmin, go to the System icon at the top of webmin, then click on the "Bootup and Shutdown" icon. Click the webmin service. Next to "Start at boot time?" select the "No" radio button, then click the Save button. Now go back into the webmin service again and click the "Stop Now" button. Webmin should no longer be accessible.

Kloxo recommends changing the SSH communications port from 22 to something else for security (perhaps 522). While I have no objection to doing that, and you might just go ahead and do it to get Kloxo to stop bugging you about it, I suggest that terminal access isn't necessary most of the time for maintaining a Kloxo hosting server. For that reason I simply disable SSH when I'm not actively using it, and you might consider doing the same. But recognize that disabling SSH is controversial, since you are locked out of the system and have no way to make command line repairs if you lose your Kloxo control panel. Consider the disabling of SSH carefully. As an alternative, you might consider SSH Authorized Key access instead, which can be configured through Kloxo.

Remember! If you change any ports to non-standard ports they will need to be added to the SPF firewall configuration, and APF will need to be restarted to apply changes, as follows.

```
/usr/local/sbin/apf –r
```

To disable SSH terminal access, login to Kloxo as administrator. In the security box, click the SSH Config icon. Put a check mark in the box next to "Completely Disable Password Based Access". Click the Update button.

Your system is now locked out. You can still administrate Kloxo and you still have basic VPS services access, but otherwise your system is inaccessible.

You should reboot the system from time to time (perhaps every month or so), to make sure that any kernel updates are applied and to flush memory. You can do that from either the VPS control panel or Kloxo administration (look in the Machine box in the admin panel).

Regaining System Access

You can regain SSH terminal access by going back to the Kloxo administration page. In the security box, click the SSH Config icon. Remove the check mark from the box next to "Completely Disable Password Based Access". Click the Update button.

To regain webmin access, open PuTTY and login as root. Issue the following command.

```
# service webmin start
```

You can now login to webmin. If you will be doing system maintenance & configuration for a while you should go back into the System icon and into Bootup and Shutdown to set webmin to start on boot. Otherwise you will need to issue the above command with PuTTY each time the system is rebooted.

Email Spam Filtering

By default, Kloxo's spam filtering capabilities are not configured to reduce spam messages. You will need to make some changes to make it work for you and your clients.

When you configure your first email addresses you will notice that a lot of the email you get will be marked as spam, by spamdyke adding ******SPAM****** to the email subject. By sending all email to your inbox, marked or not, it gives you an opportunity to see which good messages are false positives and how much spam is getting through.

The first step is to redirect email marked as spam to the spam folder. Do that by going to the Kloxo panel, select the Mail Accounts tab, then go to the email account Kloxo panel by clicking on the email account name. Click the Filter Config icon. Change "What To Do With Spam" to spambox, then click update.

Now return to the email account Kloxo panel and click the Spam Training icon. You will see the messages judged as spam listed in that page. If you see any good messages in that list you should click the box next to the message then click the "train as ham" button (spam=bad, ham=good). If you train a message as ham, any future messages from the same sender should not be marked as spam. You will want to monitor the messages marked as spam for a few days to make sure good messages aren't being tossed.

Hint: If you are logged-in as admin you will also see buttons for Train As System Spam & Ham. If you use those buttons it will whitelist & blacklist messages for the entire system, rather than for individual email accounts.

Another Hint: Login to RoundCube to view good messages that landed in the spam folder. You may need to "subscribe" to the Junk folder in RoundCube settings to view the spam folder.

One More Hint: You will notice a tab in the Spam Training icon called Clear Spam Db. That's not really a tab, it's a command that wipes-out all of your spam/ham training entries. If you click on that tab it will delete all training entries for that email account without so much as a confirmation. ***Don't click on that tab*** *unless you have a compelling reason to do so.*

You can adjust the level of spam filtering by going to the domain's Kloxo panel and clicking the Spam Status icon. The default is 5. Setting it to a lower number will mark more email as spam, and setting it higher will mark less email as spam. I usually set it higher, to maybe 6 or 7, since I don't want to miss any messages. We will be blocking most of the spam with DNS blacklists anyway.

DNS blacklists will turn away a lot of spam before it reaches filtering. These are lists of sending servers that have been blacklisted for spamming. Once you subscribe to DNS blacklists, any email received from rogue sending servers will be deleted. Depending on your spam situation, 80% or more of all unsolicited email can be removed by simply subscribing to some major blacklists. To do that, go to the Kloxo admin panel and click the Server Mail Settings icon. Click the Spamdyke tab. In the box at the bottom marked "Space separated DNS RBL servers" enter bl.spamcop.net and zen.spamhaus.org separated by a single space. Click the update button. Those blacklists are free, and are the most well-respected spam authorities.

It's a good idea to also select Reject Servers Without RDNS Names. There really aren't any legitimate email servers that don't have rDNS properly set.

Your spam should be pretty well under control now.

Secondary DNS

The previous tutorial described a DNS server on the same machine as the web server using a single IP address. That works just fine if all you are doing is hosting simple websites. In that case, a good argument can be made that if the web server goes down for some reason (along with the DNS server, most likely), that a secondary DNS server can't help visitors get to your website because the web server is down anyway. And all of that is true.

Where it makes a difference is when you have clients who are counting on email that is handled by your server. So if you have a business that is hosting its email with you and your only DNS server is down, mail will be immediately returned to senders as a bad email domain. Obviously having email returned to customers would not be good for business. But if you had a secondary DNS server running someplace the sender's email server will hold the email message until your email server comes back up, then try sending again.

So if your business reaches a point where you believe that a secondary DNS server is necessary, you need to consider a solution. If you only have a limited number of domains then you might consider a free service to host your secondary DNS. You might consider BuddyNS.com for that. It's very good service, and you can host an unlimited number of DNS zones (domains) with them, but you will have to enter new domains manually.

If you are going to be running a hosting business where a lot of domains are likely to be created or deleted, particularly if you are going to have resellers who will be creating domains that you aren't even aware of, then you need to find a more automated solution. To that end, I am going to provide an automated method of making the slave DNS server aware of new or deleted domains.

It's important to understand that while BIND comes with the capability to automatically update zones, it does not have the capability to know when new zones are added in the master DNS server. The first part of this tutorial will be to take care of that by transferring a zone list to the slave server.

These steps are done in the Master DNS server

The first thing that needs to be done is to create a zone listing file in the master DNS server machine that can be used as a .conf file by BIND in the slave DNS server. However, we'll use the file extension .txt in this script, since Apache is funny about transferring .conf files.

You will need to determine where the active .conf file is on the master DNS server. The BIND default location is /etc/named.conf, but I use the Kloxo control panel so my conf file happens to be:

```
/var/named/chroot/etc/kloxo.named.conf
```

Also you need to decide where your zone files should be in the slave machine. In my case I wanted the files to be in the following directory.

```
/var/named/chroot/var/named/slaves/dallas/named/
```

Note that I used "dallas" in the path, since my master DNS server VPS happens to be in Dallas. That not only leaves room for future server locations, but also keeps files orderly. With my master DNS server in Dallas and my slave DNS server in Denver, I keep things straight by referencing those locations. For that reason, I'll called the script we are going to create updatedenver. So you should use a text editor to create a file in /var/named/ called updatedenver, then paste the following code into it.

```
#!/bin/sh
#
for domain in `grep ^zone /var/named/chroot/etc/kloxo.named.conf |grep "type
master" |/bin/awk '{print $2}' |/bin/awk -F\" '{print $2}'`
do
/usr/bin/printf "zone \"${domain}\" { type slave; file
\"/var/named/chroot/var/named/slaves/dallas/named/${domain}\"; masters {
174.34.133.23; }; };\n"
done > /home/admin/entomy.com/dns/updatedenver.txt
```

In the first line of code (the "for" statement), the only thing you need to change is the path to your named.conf file. In this case it will be the path to the kloxo.named.conf.

In the "printf" statement, change the path to where you want to zone files to be put. Change the IP address (174.34.133.23) to the IP address of your master server. When this conf file is eventually applied in the slave DNS server, BIND will create empty zone files in the slave server according to where you specify in this statement.

In the last line (i.e., the"done" statement) enter the path to a web accessible location to deposit the output file of this script, usually someplace in /var/www… or someplace in /home/…

Once you are finished editing the file, save it. I called my script updatedenver and put it in the /var/named/ directory, which will work fine.

Now login to SSH as root and enter the following command to make sure that the web accessible script is writable. You will edit the path to the web accessible location of your choice.

```
# mkdir /home/admin/entomy.com/dns/
# chmod 755 /home/admin/entomy.com/dns/
```

Now test run the script.

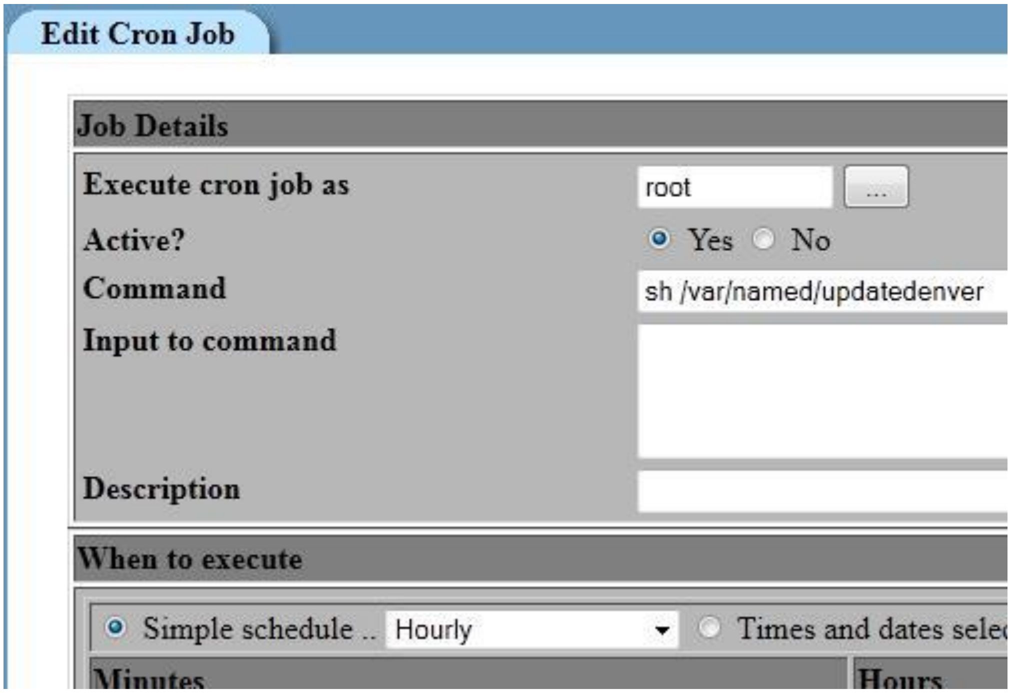
```
# sh /var/named/updatedenver
```

Check the output of the script buy navigating to the web accessible directory and opening the output file with a text editor. Mine is located at.

```
/home/admin/entomy.com/dns/updatedenver.txt
```

You should see a line in that file for each existing zone, specifying the path location for the zone file in the slave DNS machine.

Once you get the script running properly, you should create a cron job to run the script at regular intervals. In this case, we'll run it once per hour so there will always be a fresh listing of your zones available to the slave DNS server to fetch. Using webmin, click the System icon at the top and click the Scheduled Cron Jobs icon. Click the "Create a new scheduled cron job" link. Fill it out like this:



Edit Cron Job

Job Details

Execute cron job as: root

Active? ● Yes ○ No

Command: sh /var/named/updatedenver

Input to command:

Description:

When to execute

● Simple schedule .. Hourly ○ Times and dates selec

Minutes | Hours

That concludes the Master DNS server configuration, at least as far as updating zone names is concerned.

These steps are done in the Slave DNS server

The objective of this section is to configure the slave DNS server to fetch a zone information file from the master DNS server, place the file in a location that is accessible to BIND, and then apply those zones to BIND by restarting the application. To do that you need to create a script.

Create a text file and paste the following code into it:

```
#!/bin/sh
wget http://entomy.com/dns/updatedenver.txt -O /var/named/chroot/var/named/slaves/dallas.conf
/etc/init.d/named restart
```

The wget statement fetches the file from the master DNS server, and then saves it as a file called dallas.conf in the specified location. You will need to edit the URL of the text file location to the actual web accessible location you put it in on the master DNS server. Likewise, edit the name and location of the .conf file to your liking. When done, save the file. I called the file getdallas, since my master DNS server is in Dallas, and saved it in the `var/named/chroot/var/named/slaves/` directory, the same location as I put the .conf file.

Now open your named.conf file (normally found in /etc) and place this statement at the end of the file.

```
include "/var/named/chroot/var/named/slaves/dallas.conf";
```

You will need to edit the path and file name to match the location of the .conf file you specified in the script. Save the file. Also not that if BIND is running “chrooted” (as it is under Kloxo) that the path will be shortened to make the command look like this.

```
include "/var/named/slaves/dallas.conf";
```

Now login to SSH as root and create the .conf file so it can be written to by your script, editing for the proper path and file name, of course.

```
# touch /var/named/chroot/var/named/slaves/dallas.conf
```

Create the directory for the zone files to be created in, make the directory writable, and then restart BIND.

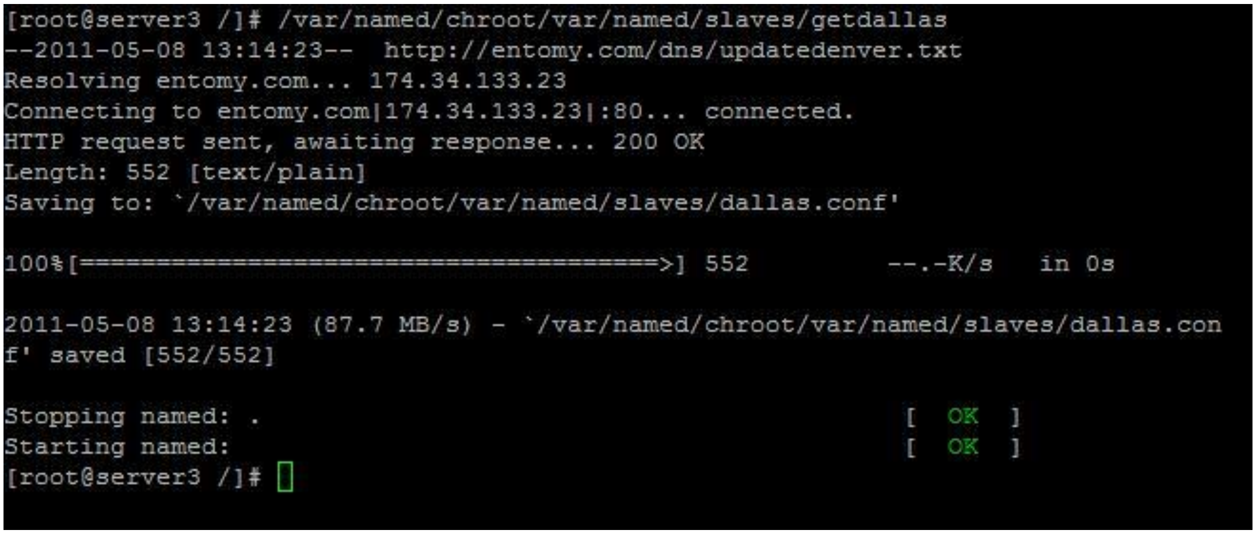
```
# mkdir /var/named/chroot/var/named/slaves/dallas/
# mkdir /var/named/chroot/var/named/slaves/dallas/named/
# chmod 777 /var/named/chroot/var/named/slaves/dallas/
# chown named:named /var/named/chroot/var/named/slaves/dallas/
# chmod 777 /var/named/chroot/var/named/slaves/dallas/named/
# chown named:named /var/named/chroot/var/named/slaves/dallas/named/
# /etc/rc.d/init.d/named restart
```

Of course, all of the above can be done a lot quicker & easier using the webmin file manager, but it works fine from the command prompt.

With those tasks done, test run the script from the command prompt.

```
/var/named/chroot/var/named/slaves/getdallas
```

With any luck, your output will look like this.



```
[root@server3 /]# /var/named/chroot/var/named/slaves/getdallas
--2011-05-08 13:14:23--  http://entomy.com/dns/updatedenver.txt
Resolving entomy.com... 174.34.133.23
Connecting to entomy.com|174.34.133.23|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 552 [text/plain]
Saving to: `/var/named/chroot/var/named/slaves/dallas.conf'

100%[======================================>] 552         --.-K/s   in 0s

2011-05-08 13:14:23 (87.7 MB/s) - `/var/named/chroot/var/named/slaves/dallas.con
f' saved [552/552]

Stopping named: .                                          [  OK  ]
Starting named:                                            [  OK  ]
[root@server3 /]# 
```

You can see from the output that the file was fetched, saved in the proper location, then named was restarted successfully.

Once you have the script running the way you want it to, create a cron job to run the script soon after the script on the master DNS server runs. Scheduling the scripts to run once or twice an hour is normally sufficient for redundant DNS.

Note that the zone files that were created from these scripts will be empty files. These scripts are only intended to make the slave DNS server aware of changes in the master DNS server zone listing. You will need to configure another solution to fetch the zone file contents, such as AXFR, rsync, or scp. However, it's important to note that any zone files that existed in the slave DNS server before the script was run that already contained zone details will not be overwritten.

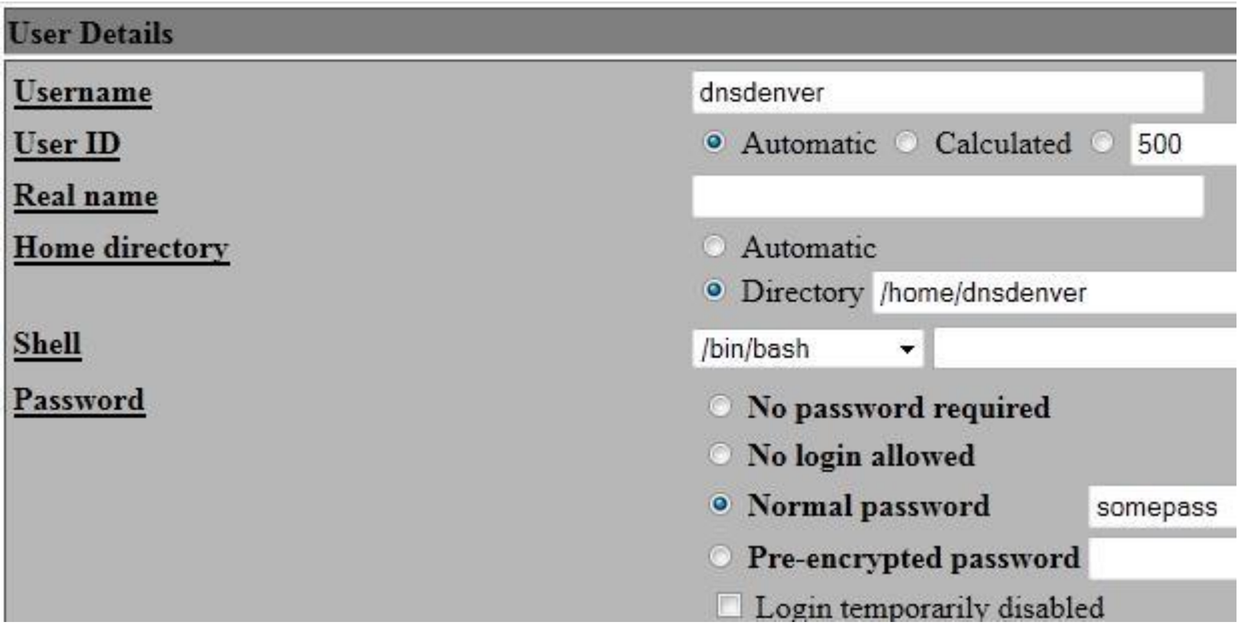
Update Zone Files

Now that we have the zone files entered into the slave DNS server successfully, the zone files need to be populated with current information from the master DNS server. There are a number of ways to do that; AXFR (or IXFR), rsync, scp, etc. Setting up any of those methods is an involved process, and they all work (some better than others), but I've found rsync to be the most satisfactory solution for updating zone files.

Before starting, confirm that both the master DNS server and the slave DNS server machines have the latest rsync installed.

```
# yum install rsync
```

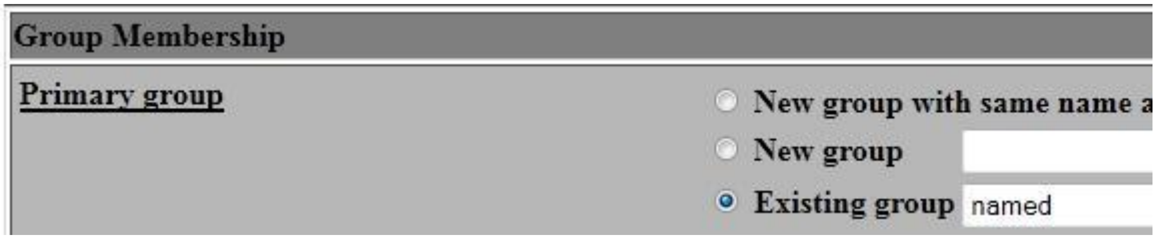
On the master DNS server machine, create a user that has a password and login access. Using webmin you can click the System icon at the top and then click the Users and Groups icon. Click the "Create a new user" link. Fill it out like this.



User Details

Username	dnsdenver
User ID	● Automatic ○ Calculated ○ 500
Real name	
Home directory	○ Automatic ● Directory /home/dnsdenver
Shell	/bin/bash
Password	○ No password required ○ No login allowed ● Normal password somepass ○ Pre-encrypted password □ Login temporarily disabled

Also, farther down the page next to “Primary group” select “Existing group” and enter named, like this.


Group Membership

Primary group

○ New group with same name a

○ New group

● Existing group named

Click the Create button at the bottom.

Determine where your zone files are located in your master DNS server. Those are usually in one of these two locations.

```
/var/named/
/var/named/chroot/var/named/
```

If you are in the correct directory you will see individual files for each zone you maintain. The file naming convention is simply the domain name that the zone services.

On the slave DNS server machine, test rsync by logging into SHH as root and issuing the following command. This step is more than just a test, since you must login to rsync manually at least once to add your remote server to the “allow” list before a non-human login is allowed.

```
# rsync -avz -e ssh dnsdenver@example.com:/var/named /var/named/slaves/
```

Where “dnsdenver” is the user you created in the master DNS machine and example.com is the domain name (or IP address) of your master DNS server. The first /var/named entry is the location in your master DNS server where the zone files are located, and /var/named/slaves/ is the location in the slave DNS server where you want the zone files to be replicated.

Note that if you are using a non-standard SSH port (555 in this example) that you will need to enter the command like this:

```
# rsync -avz -e "shh -p 555" dnsdenver@entomy.com:/var/named/ /var/named/slaves/dallas/
```

The test run will still ask for the password of the user, which you will need to provide, but we’ll take care of automating the process later. Right now you just want to get your source and target directories doing exactly what you want them to do. That is, to make sure that rsync is finding the zone files in the master DNS server machine, and then updating the zone files in the slave DNS server machine where you want them to go. Test run rsync as many times as you need to in order to get it working right.

Note that rsync might be moving some sub directories to the slave DNS server machine that you don’t need. Don’t worry about that now, we’ll exclude those later.

Once you are satisfied that rsync is doing exactly what you want it to do, we’ll move-on make rsync run without user interaction by installing a key.

On the slave DNS server machine, login to SSH as root and issue the following commands.

IMPORTANT: Don't enter your password when prompted. If you enter your password rsync will still require user interaction. Just press Enter when prompted for a password.

```
# mkdir /root/rsync
# ssh-keygen -t dsa -b 1024 -f /root/rsync/mirror-rsync-key
```

Now copy the key you just created to the master DNS server machine.

```
# scp /root/rsync/mirror-rsync-key.pub dnsdenver@example.com:/home/dnsdenver/
```

If you are using an alternate ssh port, such as 522 suggested earlier in this paper, then you would need to enter that command this way:

```
# scp –P 522 /root/rsync/mirror-rsync-key.pub dnsdenver@example.com:/home/dnsdenver/
```

If you have difficulty with the above command for some reason (perhaps a firewall issue from a non-standard SSH port) don't fret over it too much. Use whatever method you wish to transfer the file, just make sure that it's in the home directory for the user you created in the master DNS server machine.

On the master DNS server machine, login to SSH as the user you created (dnsdenver in my example) but NOT AS ROOT. You MUST login as the user you created. Enter the following commands.

```
# mkdir ~/.ssh
# chmod 700 ~/.ssh
# mv ~/mirror-rsync-key.pub ~/.ssh/
# cd ~/.ssh
# touch authorized_keys
# chmod 600 authorized_keys
# cat mirror-rsync-key.pub >> authorized_keys
```

Your key should be installed in the master DNS machine and is now ready for testing.

On the slave DNS server machine, login to SSH as root and issue this more detailed rsync command (it is all one command, but wrapping to a new line).

```
rsync -avz --delete --exclude=**/data --exclude=**/slaves -e "ssh -p 555 -i
/root/rsync/mirror-rsync-key" dnsupdate@entomy.com:/var/named /var/named/slaves/dallas
```

Note that we've added a few things since we tested rsync earlier. The "--delete" entry tells rsync to remove any files that it finds on the slave DNS server that are no longer on the master DNS server. The "--exclude=" entries tell rsync to ignore the /data and /slaves directories, so they won't be transferred any longer. The "ssh" commands between the double-quotes tells SSH to use the non-standard SSH port 555, and to authenticate using the specified key. If you are using the standard SSH port of 22 then you can enter the "ssh" entry like this.

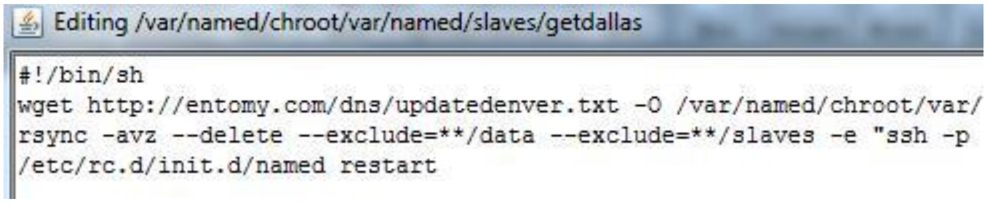
```
"ssh -i /root/mirror-rsync-key"
```

The rest of the entries should be entered exactly as what worked well for you during the testing phase.

If the command does what you want it to do, and runs without asking for a password, then you are ready to automate the process. To do that you will run the rsync to update zone file contents each time the zone file list is updated. We can do that by entering the final rsync command in the script that we wrote for the slave DNS server. In the example we called it getdallas and placed it in the following directory.

```
/var/named/chroot/var/named/slaves/
```

Edit the script with a text editor, adding the rsync command on a new line before the named restart command, so it looks like this.



Editing /var/named/chroot/var/named/slaves/getdallas

```
#!/bin/sh
wget http://entomy.com/dns/updatedenver.txt -O /var/named/chroot/var/
rsync -avz --delete --exclude=**/data --exclude=**/slaves -e "ssh -p
/etc/rc.d/init.d/named restart
```

Test run the script using SSH as root on the slave DNS machine, just to be sure everything works as it should. You don’t need to do anything else, since the cron job for that script was created in a previous step. You can run the scripts in the two machines as often as you wish, but bear in mind that the DNS server in the slave machine will be unavailable for the time it takes to restart named. Running it once or twice per hour should be more than adequate.

End of Tutorial

Best of luck with your hosting business!