Linux-Vserver on Debian Testing (Etch), the easy way

In this tutorial, I'll show you how to install Linux-Vserver on Debian Testing (Etch), the easy way. This is my first tutorial here, so please tell me if you like it or not. You may find it a bit verbose, as I try to explain things in enough detail so that everyone understands what I mean :-)

What is Linux-Vserver, you ask? It's simple. Basically, Linux-Vserver is an open-source system used to separate a single physical server into multiple virtual servers. From the Linux-Vserver website:

"Linux-VServer allows you to create virtual private servers and security contexts which operate like a normal Linux server, but allow many independent servers to be run simultaneously in one box at full speed. All services, such as ssh, mail, Web, and databases, can be started on such a VPS, without modification, just like on any real server. Each virtual server has its own user account database and root password and doesn't interfere with other virtual servers."

Two terms you will need to know are:

* **Host System**: This is the main system (the physical server) which you install Linux-Vserver onto.
* **Guest System**: These are the virtual systems you create using the Linux-Vserver utilities. They run on top of the Host System, and are all isolated from each other.

Most tutorials mention that you need to compile your own kernel in order to use Linux-Vserver. However, Debian Etch includes a Linux-Vserver kernel as standard, so you don't need to compile it yourself. This has the advantage of being easier and faster to install, and it's easy to keep up-to-date with security updates.

### First Steps

The first thing you'll need to do is have a system with a fresh installation of Debian Etch. To do so, download the Debian Testing CD from <http://www.debian.org/>, and install the base system only. After you've installed the base system, the next step is to make sure that the repositories are enabled. Firstly, make sure you're the superuser (the prompt is a **#**, not a **$**). If not, become the superuser (root):

$ su  
#

Once you're sure you're root, we need to edit /etc/apt/sources.list:

# nano /etc/apt/sources.list

Make sure a line similar to this is in the file:

deb ftp://ftp.au.debian.org/debian etch main contrib non-free

If it's not, add it in. Press CTRL+O and hit ENTER to save the file, and then press CTRL+X to exit.  
After this, we need to update the package list, so type the following command:

# apt-get update

### The Packages

Now that we've updated the package list, the next step is to install all the required packages. The packages required by Linux-Vserver are:

* **linux-image-vserver-686** - This is the actual kernel
* **util-vserver** - These are the utilities used to administer the guests
* **vserver-debiantools** - These are special Vserver tools for Debian, used to create and duplicate Debian guest systems.
* **ssh** - This is probably already installed, but just in case it isn't

So, let's install them:

# apt-get install linux-image-vserver-686 util-vserver vserver-debiantools ssh

Once that's done, we need to reboot:

# reboot

When the GRUB menu appears, make sure you choose the vserver kernel. Once your system boots, you'll be running the new kernel. You may check this by running

# uname -r   
2.6.16-2-vserver-686

Notice how the output has 'vserver' in it? This means you're running the VServer kernel.

Guess what? That's the *whole* installation! Everything is now setup and ready to use :-). There's one very important thing to do though: Create the guest systems.

### Creating a virtual server (guest)

So, now that Linux-Vserver is set up and ready to use, we need to create a guest system. On a Debian system, you may do so using the newvserver command. The syntax of this command is:

newvserver --vsroot /var/lib/vservers/ --hostname <Hostname> \--domain <Domain> --ip <IP Address>/<CIDR Range> \--dist etch --mirror <Debian Mirror> --interface <Net Interface>

*(the backslashes at the end of the lines mean that it continues onto the next line. You may copy it as-is [with the backslashes], or put the whole command on one line [and exclude the backslashes], as I do below)*

The command line arguments are:

* **vsroot** - This is where the files for the guests are kept. On a default installation, this will be at /var/lib/vservers. Alternatively, some people create a separate partition for their vservers.
* **Hostname** - The hostname of the system (eg. test1)
* **Domain** - The domain of the system. This is usually the same as the domain you chose for the host system (eg. dan-network.local. This doesn't need to be real, it's only used internally.)
* **IP Address**- The IP address for the guest system (eg. 10.1.1.7)
* **CIDR Range** - The CIDR Range for your local network. For a 10.x.x.x network, this is usually /8. For a 172.16.x.x network, this is usually /16. For a 192.168.x.x network, this is usually /24. If your network is subnetted, this will be different. When in doubt, choose /8 :)
* **Dist** - The distribution to use. For the purposes of this exercise, we use etch.
* **Debian Mirror** - The Debian mirror you use (when in doubt, choose ftp://ftp.debian.org/debian)
* **Interface** - Your network interface, if it's not eth0 (eg. eth1).

So, let's make a test server. The settings for our test guest are like so:

* Hostname: test1
* Domain: example.com
* IP Address: 10.1.1.7
* CIDR Range: /8
* Debian Mirror: http://ftp.au.debian.org/debian/
* Interface: eth1

Let's go!

# newvserver --vsroot /var/lib/vservers/ --hostname test1 --domain example.com --ip 10.1.1.7/8 --dist etch --mirror http://ftp.au.debian.org/debian/ --interface eth1

This will begin a Debian net install, which will only take a few minutes. Once the packages are all downloaded and installed, the Debian base setup will come up. This will prompt you for your time zone, and also ask for a root password. Once you've completed this, the VServer will be ready to use.

### Entering the guest

OK, so we've set up the guest, now to do anything useful, we need to start the guest, and enter into its context. To do so, we use the 'vserver' command. The basic syntax for it is:

vserver <name> [start | stop | restart | enter]

So, to start and enter the guest we created earlier, type the following:

vserver test1 start; vserver test1 enter

The output will be a bit like:

root@server1:/home/daniel# vserver test1 start; vserver test1 enterStarting system log daemon: syslogd.Starting internet superserver: no services enabled, inetd not started.Starting periodic command scheduler....test1:/#  
  
We're now 'inside' the virtual server. You can run any command you would normally run. Note that this is a *very* basic installation, so you should use apt to install whatever you want.