

Installing And Using OpenVZ On Debian Etch

By Falko Timme

Published: 2008-05-22 19:03

Installing And Using OpenVZ On Debian Etch

Version 1.0

Author: Falko Timme <ft [at] falkotimme [dot] com>

Last edited 05/08/2008

In this HowTo I will describe how to prepare a Debian Etch server for OpenVZ. With OpenVZ you can create multiple Virtual Private Servers (VPS) on the same hardware, similar to Xen and the Linux Vserver project. OpenVZ is the open-source branch of Virtuozzo, a commercial virtualization solution used by many providers that offer virtual servers. The OpenVZ kernel patch is licensed under the GPL license, and the user-level tools are under the QPL license.

This howto is meant as a practical guide; it does not cover the theoretical backgrounds. They are treated in a lot of other documents in the web.

This document comes without warranty of any kind! I want to say that this is not the only way of setting up such a system. There are many ways of achieving this goal but this is the way I take. I do not issue any guarantee that this will work for you!

1 Installing OpenVZ

In order to install OpenVZ, we need to add the OpenVZ repository to our `/etc/apt/sources.list`:

```
vi /etc/apt/sources.list
```

```
[...]  
deb http://download.openvz.org/debian-systs etch openvz  
[...]
```

Run

```
wget -q http://download.openvz.org/debian-sysfs/dso_archiv_signing_key.asc -O- | apt-key add - && apt-get update
```

afterwards to download the key of that repository and update the package database.

The repository contains six OpenVZ kernel from which you must choose one. The *ovzkernel* packages use the original OpenVZ kernel configuration, the *fzkernel* packages use the default Debian kernel configuration plus OpenVZ settings:

- *ovzkernel-2.6.18*: uniprocessor | up to 4GB of RAM | i386 and amd64
- *ovzkernel-2.6.18-smp*: symmetric multiprocessor | up to 4 GB of RAM | i386 and amd64
- *ovzkernel-2.6.18-enterprise*: SMP + PAE support + 4/4GB split | up to 64 GB of RAM | i386 only
- *fzkernel-2.6.18-686*: uni- and multiprocessor | up to 4GB of RAM | i386
- *fzkernel-2.6.18-686-bigmem*: symmetric multiprocessor | up to 64 GB of RAM | i386
- *fzkernel-2.6.18-amd64*: uni- and multiprocessor | amd64

Pick one of them and install it as follows:

```
apt-get install fzkernel-2.6.18-686-bigmem
```

Next update the GRUB boot loader:

```
update-grub
```

Now we install some OpenVZ user tools plus a minimal Debian Etch OS template which we can use to create virtual machines:

```
apt-get install vzctl vzquota vzctl-ostmpl-debian vzprocps vzdump
```

Create a symlink from `/var/lib/vz` to `/vz` to provide backward compatibility:

```
ln -s /var/lib/vz /vz
```

Open `/etc/sysctl.conf` and make sure that you have the following settings in it:

```
vi /etc/sysctl.conf
```

```
[...]  
net.ipv4.conf.all.rp_filter=1  
net.ipv4.icmp_echo_ignore_broadcasts=1  
net.ipv4.conf.default.forwarding=1  
net.ipv4.conf.default.proxy_arp = 0  
net.ipv4.ip_forward=1  
kernel.sysrq = 1  
net.ipv4.conf.default.send_redirects = 1  
net.ipv4.conf.all.send_redirects = 0  
[...]
```

If you need to modify `/etc/sysctl.conf`, run

```
sysctl -p
```

afterwards.

The following step is important if the IP addresses of your virtual machines are from a different subnet than the host system's IP address. If you don't do this, networking will not work in the virtual machines!

Open `/etc/vz/vz.conf` and set `NEIGHBOUR_DEVS` to `all`:

```
vi /etc/vz/vz.conf
```

```
[...]  
NEIGHBOUR_DEVS=all  
[...]
```

Finally, reboot the system:

```
reboot
```

If your system reboots without problems, then everything is fine!

Run

```
uname -r
```

and your new OpenVZ kernel should show up:

```
server1:~# uname -r  
2.6.18-fza-028stab053.5-686-bigmem  
server1:~#
```

2 Using OpenVZ

I will now show you the basic commands for using OpenVZ.

To set up a VPS from the default minimal Debian Etch template (you can find it in `/var/lib/vz/template/cache`), run:

```
vzctl create 101 --ostemplate debian-4.0-i386-minimal --config vps.basic
```

The `101` must be a unique ID - each virtual machine must have its own unique ID. You can use the last part of the virtual machine's IP address for it. For example, if the virtual machine's IP address is `1.2.3.101`, you use `101` as the ID.

If you want to have the vm started at boot, run

```
vzctl set 101 --onboot yes --save
```

To set a hostname and IP address for the vm, run:

```
vzctl set 101 --hostname test.example.com --save
```

```
vzctl set 101 --ipadd 1.2.3.101 --save
```

Next we set the number of sockets to 120 and assign a few nameservers to the vm:

```
vzctl set 101 --numothersock 120 --save
```

```
vzctl set 101 --nameserver 213.133.98.98 --nameserver 213.133.99.99 --nameserver 213.133.100.100 --nameserver 145.253.2.75 --save
```

(Instead of using the `vzctl set` commands, you can as well directly edit the vm's configuration file which is stored in the `/etc/vz/conf` directory. If the ID of the vm is `101`, then the configuration file is `/etc/vz/conf/101.conf`.)

To start the vm, run

```
vzctl start 101
```

To set a root password for the vm, execute

```
vzctl exec 101 passwd
```

You can now either connect to the vm via SSH (e.g. with [PuTTY](#)), or you enter it as follows:

```
vzctl enter 101
```

To leave the vm's console, type

```
exit
```

To stop a vm, run

```
vzctl stop 101
```

To restart a vm, run

```
vzctl restart 101
```

To delete a vm from the hard drive (it must be stopped before you can do this), run

```
vzctl destroy 101
```

To get a list of your vms and their statuses, run

```
vzlist -a
```

```
server1:~# vzlist -a
      VEID      NPROC STATUS IP_ADDR      HOSTNAME
      101        6 running 1.2.3.101  test.example.com
server1:~#
```

To find out about the resources allocated to a vm, run

```
vzctl exec 101 cat /proc/user_beancounters
```

```
server1:~# vzctl exec 101 cat /proc/user_beancounters
Version: 2.5
  uid resource          held   maxheld barrier   limit   failcnt
101: kmemsize           500737 517142 11055923 11377049 0
     lockedpages        0        0      256      256      0
     privvmpages        2315     2337   65536   69632    0
     shmpages           640      640    21504   21504    0
     dummy              0        0        0        0        0
     numproc            7        7       240     240      0
     physpages          1258     1289    0 2147483647 0
     vmguarpages        0        0    33792 2147483647 0
     oomguarpages       1258     1289    26112 2147483647 0
     numtcpsock         2        2       360     360      0
     numflock           1        1       188     206      0
     numpty             1        1        16     16       0
     numsiginfo         0        1       256     256      0
     tcpsndbuf          17856    17856  1720320 2703360 0
     tcprcvbuf          32768    32768  1720320 2703360 0
     othersockbuf       2232     2928   1126080 2097152 0
     dgramrcvbuf        0        0    262144  262144  0
     numothersock       1        3       120     120      0
     dcachesize         0        0   3409920 3624960 0
     numfile            189     189    9312    9312     0
```

<i>dummy</i>	0	0	0	0	0
<i>dummy</i>	0	0	0	0	0
<i>dummy</i>	0	0	0	0	0
<i>numiptent</i>	10	10	128	128	0

```
server1:~#
```

The *failcnt* column is very important, it should contain only zeros; if it doesn't, this means that the vm needs more resources than are currently allocated to the vm. Open the vm's configuration file in */etc/vz/conf* and raise the appropriate resource, then restart the vm.

To find out more about the *vzctl* command, run

```
man vzctl
```

3 Links

- OpenVZ: <http://openvz.org>
- Debian: <http://www.debian.org>