

By Falko Timme

Published: 2009-01-14 18:01

How To Convert Physical Systems And Xen VMs Into OpenVZ Containers (Debian Etch)

Version 1.0

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

Last edited 12/17/2008

This guide explains how you can convert physical systems (running Debian Etch) or Xen domUs (also running Debian Etch) into an OpenVZ container. This procedure should also work for converting VMware VMs, VirtualBox VMs, or KVM VMs into OpenVZ containers, but I haven't tried this. It should work for other Linux distributions as well, with minor modifications (for example, the network configuration is not located in `/etc/network/interfaces` if you're not on Debian/Ubuntu).

I do not issue any guarantee that this will work for you!

1 Preliminary Note

I want to convert the physical system or Xen domU `server1.example.com` (with the IP address `192.168.0.100`) into an OpenVZ container. I have a second server, `server2.example.com`, where I have set up OpenVZ according to this guide: [Installing And Using OpenVZ On Debian Etch](#). I will use the VEID 211 for the new OpenVZ container on `server2.example.com`.

I will use `/vz` here as the OpenVZ directory, so make sure that `/vz` is a symlink to `/var/lib/vz`:

server2:

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

2 Converting server1.example.com To OpenVZ

First we create two empty directories for the new OpenVZ container 211 on *server2* and also a standard OpenVZ configuration file for the new container:

[server2:](#)

```
mkdir /vz/root/211 /vz/private/211  
  
cat /etc/vz/conf/ve-vps.basic.conf-sample > /etc/vz/conf/211.conf
```

Next we install rsync on both systems as we will use rsync to transfer the files from *server1* to the OpenVZ container:

[server1/server2:](#)

```
apt-get install rsync
```

Now let's transfer *server1*'s files to the OpenVZ container:

[server2:](#)

```
rsync -arvpz --numeric-ids --exclude dev --exclude proc --exclude tmp -e "ssh -l root@192.168.0.100" root@192.168.0.100:/ /vz/private/211/
```

This can take a few minutes, depending on the number and size of the files on *server1*.

Afterwards, we can configure our new OpenVZ container with the *vzctl* command, as you would do if you would create a new container from scratch.

OpenVZ needs to know which distribution is used in the container, therefore we must set the *OSTEMPLATE* value:

[server2:](#)

```
vzctl set 211 --ostemplate debian-4.0-i386-minimal --save
```

Afterwards, you can set the network parameters and also the disk space of the new container:

server2:

```
vzctl set 211 --onboot yes --save
```

```
vzctl set 211 --hostname server1.example.com --save
```

```
vzctl set 211 --ipadd 192.168.0.100 --save
```

```
vzctl set 211 --numothersock 120 --save
```

```
vzctl set 211 --nameserver 145.253.2.75 --nameserver 213.191.92.86 --save
```

```
vzctl set 211 --diskspace 10000000:11000000 --save
```

(This would set a disk space soft limit of 10GB and a hard limit of 11GB.)

If you like, you can give the container an IP address different from that used on the original system; in that case you might have to adjust a few other files in the container (e.g. Apache configuration, etc., depending on what is installed in the container). If you use the same IP address, don't forget to shut down the original system before you power on the OpenVZ container!

These are the basic settings. You can, of course, also change the other settings with the `vzctl` command.

Before we start the OpenVZ container, we have to do a few OpenVZ-specific changes to it:

server2:

```
sed -i -e '/getty/d' /vz/private/211/etc/inittab
```

```
rm -f /vz/private/211/etc/mtab
```

```
ln -s /proc/mounts /vz/private/211/etc/mtab
```

```
cp /vz/private/211/etc/fstab /vz/private/211/etc/fstab.old
```

```
grep devpts /vz/private/211/etc/fstab.old > /vz/private/211/etc/fstab
```

```
mkdir /vz/private/211/dev
```

```
mknod --mode 666 /vz/private/211/dev/ptmx c 5 2
```

```
mkdir /vz/private/211/dev/pts
```

```
cp -a /dev/tty* /dev/ptyp* /vz/private/211/dev/
```

```
rm -f /vz/private/211/dev/null
```

```
mknod --mode 666 /vz/private/211/dev/null c 1 3
```

```
mknod --mode 444 /vz/private/211/dev/urandom c 1 9
```

```
mkdir /vz/private/211/proc
```

```
cat /dev/null > /vz/private/211/etc/network/interfaces
```

That's it, you can now start the new OpenVZ container ([don't forget to shut down the old system first if both are using the same IP!](#))...

[server2:](#)

```
vzctl start 211
```

... and set a password:

```
vzctl exec 211 passwd
```

3 Links

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