



Server Virtualization Open Source Project

User-level Tools

vzctl is a high-level command line tool to control VEs. It is used to create/start/stop/delete a VE, and to set various VE parameters.

vzpkg is a set of utilities to aid in VE installation and package maintenance. It currently supports most RPM-based distros. The main idea behind vzpkg is to have "OS template metadata" which is basically a list of packages to form a VE, and a list of repositories to get those packages from. After having that metadata, creating an "OS template" (an image to be installed into a VE being created) is as easy as running a single command, which will in turn fetch all packages, verify their consistency and create (or update) an OS template.

There are a few other handy utilities, like **vzlist** which prints a list of VEs available on the server, and **vzsplit** which is used to create VE configuration suitable to run *N* VEs on the server.

Templates

The following pre-created templates are currently available:

- | | | |
|--------------------|------------------|------------------|
| - CentOS 4 | - Fedora Core 5 | - SUSE 9.3 |
| - Debian 3.1 Sarge | - Gentoo 2005.1 | - Ubuntu 6.06 |
| - Fedora Core 3 | - OpenSUSE 10 | - Slackware 10.2 |
| - Fedora Core 4 | - Slackware 10.2 | - ALTlinux 2.4 |

Most templates are available for both x86 and x86_64 (AMD64, EM64T) platforms, and in two flavors: default and minimal (bare minimum set of packages). For Fedora Core and CentOS distributions template metadata is also available, so you can build templates in-house using a single command; it is also possible to modify lists of packages and/or create your own.

It is easy to create your own template for OpenVZ - basically, you have to install a consistent set of packages that forms the base of operating system userland.

Project Links

Main site: <http://openvz.org/>
Downloads: <http://ftp.openvz.org/>
Wiki: <http://wiki.openvz.org/>
Forum: <http://forum.openvz.org/>
Bug tracking: <http://bugzilla.openvz.org/>
GIT repo: <http://git.openvz.org/>
Blog: <http://blog.openvz.org/>
IRC channel: #openvz at freenode.net

The OpenVZ way of virtualization is so-called OS-level virtualization: instead of doing hardware emulation or running many modified kernel instances under a hypervisor, virtualization is done by one single kernel. That gives OpenVZ superior scalability, manageability and native hardware performance.

OpenVZ is easy to install: what you need is any preinstalled Linux system which supports kernel 2.6. Then you download and install the OpenVZ kernel and tools, and reboot to the freshly installed kernel. Next you download a few pre-created templates for the distributions you want to have in VEs, and create those VEs. All that can be done in a matter of minutes.

From the point of view of a VE owner, VE is just like a real server: owner can log in as root via ssh, install/remove/recompile any usual Linux software, add users, halt or reboot his VE, etc.

From the point of view of a physical server owner, a VE is a separate entity: it can be created, started, stopped, and deleted. The physical server

Typical vzctl commands:

```
# vzctl create 101 --ostemplate fedora-core-5
# vzctl set 101 --ipadd 10.1.2.2 --save
# vzctl start 101
# vzctl exec 101 ps ax
# vzctl enter 101
# vzctl stop 101
# vzctl destroy 101
```

Typical vzpkg commands:

```
# vzpkgcache
# vzpkgls
# vzyum 101 install gcc
```

<http://openvz.org/>