



OpenVZ

Servervirtualisierung Open Source Project

Tools auf Benutzerebene

vzctl ist ein Kommandozeilen-Tool auf höchster Ebene zur Kontrolle von VEs. Es wird zum Erstellen, Starten, Stoppen und Löschen von VEs verwendet, sowie zum Festlegen verschiedener VE-Parameter.

Bei **vzpkg** handelt es sich um einen Satz von Utilities für die VE-Installation und Wartung der Pakete. Es unterstützt gegenwärtig die meisten RPM-basierten Distributionen. Der Hauptgedanke, der vzpkg zugrunde liegt, ist es, "Betriebssystem-Template-Metadaten" zur Verfügung zu haben, bei denen es sich grundsätzlich um eine Liste von Paketen zur Erstellung eines VE handelt, sowie um eine Liste von Repositories, aus denen diese Pakete abgerufen werden können. Mithilfe dieser Metadaten ist das Erstellen eines "Betriebssystem-Templates" (eines Images, das in ein VE, welches gerade erstellt wird, installiert werden soll) genauso einfach wie das Ausführen eines einzelnen Kommandos, das seinerseits alle Pakete abrufen, deren Konsistenz verifiziert und ein Betriebssystem-Template erstellt (oder aktualisiert).

Es gibt eine Reihe von weiteren praktischen Utilities, wie **vzlist**, das eine Liste der auf dem Server verfügbaren VEs ausgibt, und **vzsplit**, das zum Erstellen einer VE-Konfiguration verwendet wird, die zum Ausführen von n VEs auf dem Server geeignet ist.

Templates

Die folgenden bereits erstellten Templates stehen momentan zur Verfügung:

- | | | |
|----------------------|------------------|------------------|
| - 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 |

Die meisten Templates sind sowohl für x86- als auch für x86_64 (AMD64, EM64T)-Plattformen verfügbar, und zwar in zwei Ausführungen: Standard und Minimal (nur der mindestens benötigte Satz von Paketen). Für Fedora Core- und CentOS-Distributionen stehen auch Template-Metadaten zur Verfügung, so dass Sie Templates vor Ort mithilfe eines einzigen Kommandos erstellen können; es ist ebenfalls möglich, Listen von Paketen zu modifizieren und/oder eigene Listen zu erstellen.

Sie können ganz einfach Ihr eigenes Template für OpenVZ erstellen – grundsätzlich müssen Sie dazu einen konsistenten Satz von Paketen installieren, der die Grundlage des Betriebssystems Userland darstellt.

Projektlinks

Hauptsite:	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-Repository:	http://git.openvz.org/
Blog:	http://blog.openvz.org/
IRC-Kanal:	#openvz at freenode.net

Der Virtualisierungsansatz von OpenVZ besteht in der sogenannten Virtualisierung auf Betriebssystemebene: Statt der Nachbildung von Hardware oder der Ausführung vieler modifizierter Kernel-Instanzen unter einem Hypervisor wird die Virtualisierung von einem einzigen Kernel durchgeführt. Dies verleiht OpenVZ eine überaus hohe Skalierbarkeit, Verwaltbarkeit und native Hardware-Performance.

OpenVZ ist einfach zu installieren: Sie benötigen lediglich ein beliebiges bereits installiertes Linux-System, das den Kernel 2.6 unterstützt. Sie laden den OpenVZ-Kernel und die dazugehörigen Tools herunter und installieren beides, und führen dann den Neustart zu dem gerade installierten Kernel durch. Als Nächstes laden Sie einige bereits erstellte Templates für die Distributionen herunter, die Sie in VEs haben möchten, und erstellen diese VEs. Für all dies benötigen Sie nur wenige Minuten.

Vom Standpunkt eines VE-Besitzers aus betrachtet ist ein VE genau wie ein echter Server: Der Besitzer kann sich über SSH als Root anmelden, jede gebräuchliche Linux-Software installieren, entfernen oder neu kompilieren, Benutzer hinzufügen, sein VE anhalten oder neu starten, etc.

Vom Standpunkt des Besitzers eines physikalischen Servers aus betrachtet ist ein VE eine separate Einheit: Es kann erstellt, gestartet, gestoppt und gelöscht werden.

Gebräuchliche vzctl-Kommandos:

```
# vzctl create 101 --ostemplate fedora-core-4
# 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
```

Gebräuchliche vzpkg-Kommandos:

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

<http://openvz.org/>