

OpenVZ

Metin KAYA

EnderUNIX Yazılım Geliştirme Takımı

Akademi Üyesi

metin ~ Akademi.EnderUNIX.org

<http://www.EnderUNIX.org/>

Endersys Sistem Müh.

metin.kaya ~ Endersys.com.tr

<http://www.Endersys.com.tr/>

Tarih: 15 Tem 2007 Pazar EEST 06:34:17

İçindekiler

1. Giriş.....	3
2. OpenVZ Nedir?.....	3
3. OpenVZ ile Diğer Sanal Makine Uygulamalarının Karşılaştırılması.....	4
4. OpenVZ Kurulumu.....	5
5. Olası Hatalar ve Çözümleri.....	12
6. Teşekkürler.....	12
7. Kaynaklar.....	12

1. Giriş

Bu makalede OpenVZ sanal makine uygulamasından bahsedilecektir. Ayrıca OpenVZ ile diğer sanal makine yazılımlarıyla kıyaslanıp Red Hat türevi Linux sistemlere (Red Hat 3 ve 4, Fedora Core 6, CentOS 4 ve 5 kastedilmektedir) OpenVZ kurulumu anlatılacaktır.

2. OpenVZ Nedir?

OpenVZ; Linux tabanlı, işletim sistemi düzeyinde sunucu sanallaştırma yazılımıdır. OpenVZ aynı fiziksel sunucu üzerinde birden fazla yalıtılmış ve güvenli *sanal ortamlar* (Virtual Environments – VEs veya Virtual Private Servers – VPSs diyebiliriz) oluşturur. Her sanal ortam, fiziksel makine üzerinde sadece kendisi varmış gibi çalışır. Tüm sanal ortamlar birbirinden bağımsız olarak yeniden başlatılabilir ve her sanal ortamın farklı sistem kullanıcıları, IP adresleri, belleği, dosyaları, uygulamaları, sistem kütüphaneleri ve yapılandırmaları vardır.

OpenVZ, [SWsoft](#) tarafından desteklenen açık kaynak kodlu, [GNU GPL](#) lisanslı bir projedir ve yine bu firmanın ticari bir ürünü olan [Virtuozzo™](#) temellidir.

OpenVZ altında çalışan bir sanal sistem en fazla 64 GB belleği (RAM) destekleyebilir. Ayrıca bir OpenVZ çekirdeği aynı anda en fazla 100 sanal işletim sistemi yönetebilir.

3. OpenVZ ile Diğer Sanal Makine Uygulamalarının Karşılaştırılması

▲ OpenVZ şu an için sadece Asianux, CentOS, Debian, Fedora Core, Gentoo, Mandriva ve OpenSUSE Linux dağıtımlarını desteklemektedir. Ancak; OpenVZ 'in arkasında [SWsoft](#) olduğundan çok hızlı gelişmektedir.

▲ OpenVZ, GPL lisanslıdır. ESX gibi sanal makine yazılımları her ne kadar çok iyi işler yapabiliyor olsa da yüksek maliyete sahiptirler. OpenVZ ise özellikle VMware sanal makinelerine oranla Linux üzerinde çalıştırılan her uygulamada % 10 ile % 50 arasında değişen daha iyi performansa sahip olmasına rağmen **ücretsizdir**.

▲ OpenVZ kaynak kodunun % 95 'i platform bağımsızdır. Bu nedenle çok farklı mimarilerde (i386, i486, i586, i686, x86-64, IA64, PPC, PPC64, AMD, AMD64, SPARC, SPARC64) sorunsuz çalışabilmektedir. Ayrıca kodunun güvenilir, kararlı ve yüksek performanslı olduğu rahatlıkla söylenebilir.

▲ Linux tabanlı sanal bir sistem kullanıyorsanız OpenVZ, yapabileceğiniz en iyi tercih olacaktır. OpenVZ 'de işlemci zamanı, bellek, disk alanı, süreç (process) sayısı, oluşturulabilecek dosya sayısı, dosya kilidi sayısı ve açılacak TCP soket sayısı gibi konularda özelleştirmeler yapabilirsiniz. Bu bakımdan OpenVZ oldukça yetenekli.

▲ OpenVZ tüm sanal sistemler için tek bir çekirdek kullandığından sanal sistemlere sabit bellek ve CPU kotası koymaz. OpenVZ sanal sisteminiz, ön tanımlı olarak Linux sisteminizin altında bir süreç olarak çalışır ve ihtiyacı olduğu kadar bellek ve CPU harcar. Eğer sanal sistem kaynaklarına kota koyduysanız, OpenVZ sanal sunucunuzun bellek ve CPU kotasını istediğiniz zaman, sanal sistem çalışırken bile, değiştirebilirsiniz. Diğer sanal sistem uygulamalarında sanal sistemin bellek veya CPU kotasını sanal sistem çalışır durumdayken değiştirmeniz mümkün değildir.

▲ OpenVZ ile kurduğunuz sanal sisteme ait tüm dosyaları, sanal sistem çalışır durumda olmasa bile, `/vz/private/VE_ID/` dizini altında bulabilirsiniz (`VE_ID` kavramı ilerleyen bölümlerde açıklanacaktır). Örneğin:

```
[root@enderunix]# ls /vz/private/23/
```

```
bin boot dev etc home initrd lib media mnt opt proc root sbin  
srv sys tmp usr var
```

NOT: OpenVZ 'in paralı sürümü olan Virtuozzo, daha yüksek performans sunmasının yanında sanal sistemleri yönetmek için bir web arayüzü de sunmaktadır. Ayrıca Virtuozzo, Windows sistemler üzerinde de çalışabilmektedir. Virtuozzo çok çekirdekli işlemcilerde OpenVZ 'den daha kararlı ve iyi bir performans sergilemektedir. Tüm bunlara rağmen söz konusu **sanal bir Linux kullanmak** ise **OpenVZ** 'in diğer tüm sanal makine uygulamalarından daha iyi olduğu söylenebilir. Çünkü OpenVZ, Linux çekirdek bileşenlerine tümleşik çalışmaktadır.

4. OpenVZ Kurulumu

4.1 Gereksinimler

- ▲ OpenVZ sadece *ext2* ve *ext3* dosya sistemlerinde çalışır. *ext3* kullanmanız tavsiye edilir.
- ▲ Her OpenVZ çekirdeği ön tanımlı olarak */vz/private/VE_ID* dizinini kullanır (*VE_ID*: sanal ortamın kimlik numarası). Bu sayede üzerine OpenVZ kurduğunuz Linux sistemin disk kotalarından bağımsız bir sanal ortam elde edersiniz. Sanal ortamlar için */vz/private/VE_ID* dizini yerine */root/* dizinini kesinlikle kullanmayınız. Çünkü bu durumda OpenVZ ‘de çalıştıracağınız birçok uygulama % 5 disk engeliyle karşılaşma riski artar.
- ▲ Donanımınızın Red Hat çekirdekleriyle uyumlu olması gerekir. OpenVZ uyumlu donanımların listesini <http://www.swsoft.com/en/products/virtuozzo/hcl/> adresinde bulabilirsiniz.
- ▲ Sorunsuz, rahat bir kurulum için *rpm* yerine *yum* kullanmanız önemle tavsiye edilir.

4.2 Ön Hazırlıklar

Bu kısımda OpenVZ ‘in *yum* paket deposu oluşturulacaktır. Aşağıdaki komutları *root* hakkı olan bir kullanıcı ile veriniz. *rpm* ile kurulum yaparsanız bir sonraki kısma geçebilirsiniz; ancak eksik paketleri temin etmek sizin sorumluluğunuzdadır.

▲ Sisteminiz Red Hat 3 ise:

- <http://linux.duke.edu/projects/yum/download/2.0> adresinden *yum* paketini (son sürüm 2.0.* indirmeniz tavsiye edilir) indirip aşağıdaki komutla kurunuz:

```
# rpm -ivh yum-2.0.*.noarch.rpm
```

- Aşağıdaki satırları */etc/yum.conf* dosyasına yazınız:

```
[freevps]
name=FreeVPS kernel and tools
baseurl=http://yum.freevps.com/3/$basearch/
```

```
gpgcheck=0
```

```
[hsphere-vps]
name=H-Sphere VPS scripts
baseurl=http://yum.freevps.com/3/noarch/
gpgcheck=0
```

▲ Sisteminiz Red Hat 4 ise:

- <http://www.freevps.com/download/yum/RHEL4> adresinden *yum* paketini (son sürüm 2.6.* indirmeniz tavsiye edilir) indirip aşağıdaki komutla *yum* ve bağımlı olduğu paketleri kurunuz:

```
# rpm -ivh sqlite-2.8.*.i386.rpm python-urlgrabber-2.9.*.noarch.rpm  
python-sqlite-1.0.*.i386.rpm python-elementtree-1.2.*.i386.rpm
```

```
# rpm -ivh yum-2.6.*.noarch.rpm
```

- */etc/yum.repos.d* dizinine geçip sisteminizde FreeVPS paket deposunu kullanılabilir hale getiriniz (*yum.freevps.com* sunucusu aktif değilse *freevps.kalou.net* sunucusunu kullanabilirsiniz):

```
# cd /etc/yum.repos.d  
# wget http://yum.freevps.com/4/freevps.repo
```

▲ Sisteminiz CentOS 4, CentOS 5 veya Fedora Core 6 ise:

```
# cd /etc/yum.repos.d  
# wget http://download.openvz.org/openvz.repo  
# rpm --import http://download.openvz.org/RPM-GPG-Key-OpenVZ
```

4.3 Çekirdek Kurulumu

Aşağıdaki komutla sisteminize uygun bir çekirdek kurulacaktır. Donanım ve sisteminize uygun farklı bir çekirdek kullanacaksanız http://wiki.openvz.org/Kernel_flavors sayfasını ziyaret ediniz.

```
# yum -y install ovzkernel
```

rpm ile kurulum yapacaksanız <http://download.openvz.org/kernel/> sayfasından kurmak istediğiniz çekirdeğin *rpm* paketini indirip aşağıdaki komutu veriniz:

```
# rpm -ihv ovzkernel*.rpm
```



Bu komutu verirken *-U* seçeneğini kesinlikle kullanmayınız. Aksi takdirde sistemde kullanılmakta olan çekirdek kaldırılıp yerine yeni indirdiğiniz çekirdek kullanılacaktır.

4.4 Yapılandırmalar

4.4.1 Açılış Yöneticisini Yapılandırma

Sisteminizin açılış yöneticisi *GRUB* ise */boot/grub/grub.conf* dosyasına (*/etc/grub.conf* dosyası bu dosyaya sembolik linktir) aşağıdakine çok benzer satırlar otomatik olarak eklenecektir:

```
title Fedora Core (2.6.8-022stab029.1)
    root (hd0,0)
    kernel /vmlinuz-2.6.8-022stab029.1 ro root=/dev/sda5 quiet rhgb vga=0x31B
    initrd /initrd-2.6.8-022stab029.1.img
```

Bu satırların ilkinde yazan çekirdek ismini (Fedora Core) *OpenVZ* olarak değiştirmeniz açılışta listelenecek olan çekirdekler arasından *OpenVZ* çekirdeğini kolaylıkla tanımanız açısından önerilir. Ayrıca *kernel* ile başlayan 3. satırdaki *root=/dev/sda5* ifadesinden sonraki tüm seçenekleri (*quiet*, *rhgb*, v.s.) silmeniz gerekmektedir.

OpenVZ çekirdeğinin ön tanımlı olarak başlatılmasını istiyorsanız */boot/grub/grub.conf* dosyasındaki *default* satırının karşısındaki değeri 0 yapınız.

4.4.2 *sysctl* Yapılandırması

OpenVZ çekirdeğin düzgün çalışabilmesi için */etc/sysctl.conf* dosyasındaki birçok çekirdek parametresinin uygun şekilde ayarlanması gerekir. Aşağıda olması gereken ayarlar görülmektedir (aşağıdaki herhangi bir satır */etc/sysctl.conf* dosyanızda varsa aşağıdaki gibi değiştiriniz, yoksa bu dosyaya aynen ekleyiniz):

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

4.4.3 SELINUX Ayarı

OpenVZ çekirdeğinin çalışma esnasında sorun yaşamaması için SELINUX 'un devre dışı bırakılması gerekmektedir. Bu nedenle */etc/sysconfig/selinux* dosyasında *SELINUX=* ile başlayan satırın aşağıdaki gibi olması gerekir:

```
SELINUX=disabled
```

4.4.4 *conntracking* Yapılandırması

netfilter bağlantı izleme (connection tracking) özelliğini engelleyen bir güvenlik duvarınız varsa ya güvenlik duvarını devre dışı bırakın ya da OpenVZ çekirdeğine (VE0) bağlantı izleme izni verin. OpenVZ 'e bağlantı izleme izni vermek için */etc/modprobe.conf* dosyasında aşağıdaki satırı ekleyiniz:

```
options ip_conntrack ip_conntrack_enable_ve0=1
```

4.5 Sistemi Yeniden Başlatma

Bu aşamada sisteminizi *shutdown -r now*, *reboot* veya *init 6* gibi komutlardan biriyle yeniden başlatmanız gerekmektedir. Açılış yöneticisi ekranında OpenVZ çekirdeğini seçip sistemin OpenVZ çekirdeğiyle başlatılmasını sağlayınız.



OpenVZ çekirdeğiniz *Segmentation Fault* hatası verip sistemi başlatamadıysa büyük olasılıkla donanımınıza uygun OpenVZ çekirdeğini seçmemişsinizdir. Örneğin; *i386* mimariye *i686* çekirdeği kurmuş olabilirsiniz.

4.6 OpenVZ Araçlarını Kurma

OpenVZ, kurduğunuz sanal sunucuyu yönetmeniz için *vzctl* ve *vzquota* gibi araçlar sunar. *vzctl* ile OpenVZ VPS 'inizi yönetebilirsiniz (*start*, *stop*, *create*, *destroy*, *set*, v.s.). *vzquota* ise genellikle *vzctl* tarafından dolaylı olarak kullanılır. Bu araçları yum ile kurarsanız aşağıdaki komutu veriniz:

```
# yum -y install vzctl vzquota
```

Bahsedilen araçları *rpm* ile kurarsanız <http://download.openvz.org/utills/> adresinden gerekli paketleri indirdikten sonra aşağıdaki komutu veriniz:

```
# rpm -Uhv vzctl*.rpm vzquota*.rpm
```

4.7 OpenVZ 'i Başlatma

Şimdi aşağıdaki komutla OpenVZ 'i başlatabilirsiniz:

```
# /sbin/service vz start
```

Bu komut, OpenVZ çekirdek modüllerini hemen başlattığı gibi sistemi bu modülleri her açılışta başlatacak şekilde yapılandırır.

4.8 İşletim Sistemi Kalıp Cebi Hazırlama

OpenVZ çekirdeği ile kullanacağınız sanal işletim sistemleri için belli bir şablon (Operating System Template Cache) oluşturmanız gerekmektedir. Bunun için gereken paketleri yum ile kurmak için aşağıdaki komutu veriniz:

```
# yum -y install vzpkg vzyum vzrpm43-python vzrpm44-python
```

Bu işlem için *rpm* kullanacaksanız gerekli paketleri <http://openvz.org/download/template/utills/> adresinden indirip aşağıdaki komutla kurunuz:

```
# rpm -ihv vzpkg*.rpm vzyum*.rpm vzrpm44*.rpm
```

4.8 İşletim Sistemi Kalıbı Kurma

Kullanmak istediğiniz işletim sisteminin kalıbını kurmanız gerekmektedir. Aşağıdaki komut OpenVZ geliştiricileri tarafından kalıbı oluşturulmuş işletim sistemlerini listeler:

```
# yum search vztmpl
```

Bu komutun ardından seçtiğiniz işletim sistemi kalıbını şöyle kurabilirsiniz (Burada CentOS 4 tercih edildiği için ona uygun kalıp kurulmuştur. CentOS kuracaksanız CentOS 5 'e göre çok daha kararlı olan CentOS 4 'ü kurmanız tavsiye edilir.):

```
# yum -y install vztmpl-centos-4
```

rpm ile kurulum yapacaksanız uygun paketi <http://download.openvz.org/template/metadata/> sitesinden indirip aşağıdaki komutla kurabilirsiniz:

```
# rpm -ihv vztmpl-*.rpm
```

Şimdi sıra geldi kalıbını hazırladığımız işletim sistemini kullanılabilir hale getirmeye:

```
# vzpkgcache centos-4-i386-minimal
```

▲ Dilerseniz makalenin 4.8 İşletim Sistemi Kalıbı Kurma bölümünü es geçip OpenVZ geliştiricileri tarafından önceden hazırlanmış kalıplardan birini kullanabilirsiniz. Bunun için <http://download.openvz.org/template/precreated/> sitesinden istediğiniz bir kalıbı */vz/template/cache/* dizinine indirip arşivlenmiş dosyayı **açmadan** kullanmaya başlayabilirsiniz (*vzpkgcache* komutu seçilen bir işletim sistemi kalıbı için *tar* arşivi oluşturur. Burada zaten arşiv dosyası indirildiğinden arşivi açmaya gerek yoktur. Ayrıntılı bilgi için: *man 8 vzpkgcache*). Aşağıdaki komutlarla Debian 4 işletim sistemi için gerekli dosyalar sağlanmıştır:

```
# cd /vz/template/cache/  
# wget http://download.openvz.org/template/precreated/debian-4.0-i386-minimal.tar.gz
```

4.9 Son Adımlar

Önceki kısımlarda gerekli dosyalarını temin ettiğiniz ve çalışma ortamını hazırladığınız sanal sistemi aşağıdaki komutlarla çalıştırıp kullanmaya başlayabilirsiniz:

```
# vzctl create 47 --ostemplate debian-4.0-i386-minimal (1)
# vzctl set 47 --ipadd 10.0.0.247 -save (2)
# vzctl start 47 (3)
```

1. satırdaki komutun yazımı `vzctl create VE_ID --ostemplate osname` şeklinde olup `VE_ID` parametresi kullanacağınız sisteme vereceğiniz kimlik numarasıdır. Her sanal sisteme bir kimlik numarası vermeniz gerekir ve bu sayıyı istediğiniz gibi seçebilirsiniz. `osname` ise önceden kalıbını hazırladığınız veya hazır kalıbını indirdiğiniz işletim sisteminin adıdır (`debian-4.0-i386-minimal.tar.gz` için `debian-4.0-i386-minimal` yazmanız gerekir).

2. satırda sanal sisteme IP adresi atanmıştır.

3. satırda ise 47 kimlik numaralı sanal sistemimiz başlatılmıştır. Sistemin gerçekten çalışmaya başlayıp başlamadığını anlamak için aşağıdaki komutu kullanabilirsiniz:

```
[root@enderunix cache]# vzctl exec 47 ps auxww
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3   1952    644 ?        Ss   03:33   0:00 init [2]
root      7718  0.0  0.3   1728    696 ?        Ss   03:33   0:00 /sbin/syslogd
root      7741  0.0  0.5   4928   1060 ?        Ss   03:33   0:00 /usr/sbin/sshd
root      7753  0.0  0.3   2196    748 ?        Ss   03:33   0:00 /usr/sbin/cron
```

Yukarıdakine benzer bir çıktı alıyorsanız; yani 47 kimlik numaralı sanal sisteminize ait süreçleri görebiliyorsanız sanal sisteminiz başarıyla başlatılmış demektir.

Aşağıda kurulan sisteme girilip bazı komutların çalıştırılması gösterilmiştir:

```
[root@enderunix cache]# vzctl enter 47
entered into VE 47
```

```
goebbels:/# ls
```

```
bin boot dev etc home initrd lib media mnt opt proc root sbin
srv sys tmp usr var
```

```
goebbels:/# hostname
```

```
goebbels
```

```
goebbels:/# ifconfig
```

```
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

venet0     Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
            inet addr:127.0.0.1  P-t-P:127.0.0.1 Bcast:0.0.0.0  Mask:255.255.255.255
            UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 b)  TX bytes:108 (108.0 b)

venet0:0   Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
            inet addr:10.0.0.247 P-t-P:10.0.0.247 Bcast:0.0.0.0  Mask:255.255.255.255
            UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
```

```
goebbels:/# pwd
```

```
/
```

```
goebbels:/# whoami
```

```
root
```

```
goebbels:/# uname -a
```

```
Linux goebbels 2.6.18-8.1.4.el5.028stab035.1 #1 SMP Sat Jun 9 01:43:20 MSD
2007 i686 GNU/Linux
```

Görüldüğü 47 kimlik numaralı sisteme başarıyla girilip çeşitli komutlar çalıştırılmıştır. Debian sistemi hazır kalıp olarak kullandığımız için sanal sistemin bilgisayar adı önceden belirlenmiş bir isimdir (goebbels). Eğer kalıbı kendiniz hazırlamış olsaydınız üzerinde kurulum yaptığınız sistemle OpenVZ sisteminizin makine adı aynı olacaktır.

exit komutuyla içinde bulunduğunuz sanal sistemden çıkabilirsiniz. Çalışmakta olan sanal sistemi aşağıdaki komutla kapatabilirsiniz:

```
# vzctl stop 47
```

vzctl komutuna parametre olarak verilen *start* normal bir işletim sistemini başlatmaya karşılık gelirken *stop* parametresi normal bir işletim sistemini kapatmaya denktir.

Sisteminize kurduğunuz bir sanal işletim sistemini bir daha kullanmamak üzere sisteminizden tamamen kaldırmak istiyorsanız aşağıdaki komutu kullanabilirsiniz (bu komut sadece ilgili sanal sisteme ait dosyaları ve kaynakları sistemden silip kaldırır; OpenVZ paketine bir zarar vermez):

```
# vzctl destroy 47
```

▲ Bu adımdan sonra OpenVZ altında yeni bir işletim sistemi daha kurmak istiyorsanız makalenin 4.8 ve 4.9 bölümlerinde anlatılanları yapmanız yeterlidir.

5. Olası Hatalar ve Çözümleri

▲ *vzpkgcache osname* komutu, sistemde *coreutils-5.2.1-31.6* paketi olmadığından hata verirse aşağıdaki komutları çalıştırdıktan sonra tekrar deneyiniz (bilgisayar mimarinize uygun paketi, *i386*, *i586*, v.s., seçmeyi unutmayınız):

```
# wget ftp://fr2.rpmfind.net/linux/ASPLinux/i386/updates/10.1/i386/  
coreutils-5.2.1-31.6.i386.rpm  
# rpm -Uvh coreutils-5.2.1-31.6.i386.rpm
```

6. Teşekkürler

- UNIX/Linux sistem yöneticiliğindeki değerli hocam [İsmail YENİGÜL](#) 'e teşekkürler...

7. Kaynaklar

- <http://www.openvz.org>
- <http://wiki.openvz.org/>
- http://en.wikipedia.org/wiki/Comparison_of_virtual_machines
- http://searchServerVirtualization.techtarget.com/tip/0,289483,sid94_gci1192371,00.html