Subject: TEST IDEA - Scaling with openvz
Posted by kingneutron on Mon, 28 May 2007 06:21:44 GMT
View Forum Message <> Reply to Message

Say I have an external USB2 120GB drive. (Go on, say it!)

Now I would like to mount it for openvz use; no problem, I know how to do this.

I've made some basic mods to the standard Debian 4 openvz template and would like to use it as the "Base model" for testing, similar to "sysprep" builds in Windoze.

Can I just tar.gz up the 101 VE for Debian as a "master image", and vzctl create 102, etc based on that, if I copy the tarball to the Template dir? Or do I need "metadata"?

o Summary:

o Guest VE's will be batch-created to test scaling (10 at first, then 32 / 64 / 128 / 254 in order.)

o Host will have venet / veth of 172.16.0.1 // 255.255.0.0

o Master image will have IP of 172.16.1.1 // 255.255.0.0

oo Each guest VE will have IP like so:
VE 1002 == 172.16.1.2 ..
VE 1254 == 172.16.1.254 // 255.255.0.0

--I'm pretty sure I can do this programmatically with simple Bash scripting; but is there a way in-VE to have it determine what its number is on the host? (i.e. VE 1003 should "know" it's number is 1003 somehow without having networking enabled yet.)

--Once each VE gets powered on and sets up its IP, it will be netcat-listening on $port to make whatever local changes are needed after 1st-boot:

((' cd / ; time nc -l -p 32100 | tar xpf - '))

--We can then send a non-compressed tarball (what, you wanna overload your CPU? we may want to do this SIMULTANEOUSLY for stress-testing the disk subsystem) to each guest with whatever changes are needed on a localized basis, such as /etc/hostname, or the like.

--The basic plan is to easily setup an infrastructure for small ISPs, and network admins who want to deploy openvz on a semi-grand scale for customers or internal use.

--Any comments/suggestions appreciated... TIA

Subject: Re: TEST IDEA - Scaling with openvz

Posted by [rickb](#) on Mon, 28 May 2007 06:51:34 GMT

You have many ideas but I only see two questions.

Quote: based on that, if I copy the tarball to the Template dir?

Yes, see http://wiki.openvz.org/OS_template_cache

 Quote:Or do I need "metadata"?

it is not necessary. the simplest/fastest way to work with templates is to simply tar gz up an existing VE, then redeploy using that file.

Your other ideas seem interesting, but remember if you want to copy/sync a file to your ves, you can simply copy it into the VEs private. for exmaple, if you want to replace /etc/passwd on VE 103, from your HN, cp /path/passwd /vz/private/103/etc/passwd . This is the easiest way in my opinion, and more secure then a network based approach to moving files around as the root user.

Subject: Re: TEST IDEA - Scaling with openvz
Posted by [kingneutron](#) on Mon, 28 May 2007 06:59:25 GMT

  Yes, thanks... There are some openvz concepts I haven't quite wrapped my head around yet.

I have long experience with vmware, and to me directly modifying a VE's "private space" is a no-no -- kind of an "isolation violation." It's kinda like reading someone else's email when you're root. Yes we *can* do it, but *should* we...

I'd much rather prefer private disks per-VE, for $security. Only root will have direct VE-modify access anyhow, and the procedure I'm thinking of implementing would be available over the Internet for potential customers. (Just leave the VE running, and listening, and $client can send the tarball over the net whenever.)

Subject: Re: TEST IDEA - Scaling with openvz
Posted by [rickb](#) on Mon, 28 May 2007 07:13:02 GMT

I see. openvz is not a heavy duty or even a true virtualization solution (like vmware is). Rather it is a rather complete system of building linux process containers, which each resemble a conventional linux operating system. If you have worked with freebsd jails, it is very similar but (openvz is) also much more complete. the openvz network virtualization is far superior to freebsd

jails in particular. It seems like a security problem at first, that the HN an manage the VE's files, but having access to the HN, under any virtualization solution, could spell doom/insecurity under the same circumstances. ie, the HN gets hacked, all of the VE data is at risk under vmware, xen, openvz, etc.

Due to the fact that openvz is a single kernel, container approach to achieving virtualized operating systems, the complete filesystem of each VE is accessible from the hardware node. This is unlink xen, where each VE has a private filesystem. Actually, from the openvz hardware node, the VE processes look like normal processes and can be killed and interacted with like normal. This illustrates how openvZ is a container approach to virtualization.

If you want your customers to be able to [re]deploy operating systems on your hardware nodes, it is very possible and a very common requirement for a service provider. the high-level method I used to acheieve this is very simple and works well. My web application triggers a process on the HN, to make the appropriate vzctl calls which stop/destroy/create/set the VE. Using this basic strategy, you can manipulate all aspects of the VEs, including redeployment.

Rick Blundell

Subject: Re: TEST IDEA - Scaling with openvz
Posted by kingneutron on Mon, 28 May 2007 07:22:17 GMT
View Forum Message <> Reply to Message

Your writings intrigue me, and I wish to subscribe to your newsletter.

Where can I get a copy of your web app? I asked in another thread about a webmin module for openvz but have not heard anything.

// Boss prolly won't let openvz in for $semi-production unless he has a webmin (or similar) interface to be able to dink with it. He's not the biggest cmdline fan.

Subject: Re: TEST IDEA - Scaling with openvz
Posted by ugob on Tue, 29 May 2007 02:06:09 GMT
View Forum Message <> Reply to Message

http://sourceforge.net/projects/easyvz

Subject: Re: TEST IDEA - Scaling with openvz
Posted by kingneutron on Tue, 29 May 2007 02:22:07 GMT
View Forum Message <> Reply to Message

Thanks!   That looks just like what I was looking for.

Subject: Re: TEST IDEA - Scaling with openvz
Posted by kingneutron on Tue, 29 May 2007 06:43:15 GMT

[[
I'm pretty sure I can do this programmatically with simple Bash scripting; but is there a way in-VE
to have it determine what its number is on the host? (i.e. VE 1003 should "know" it's number is
1003 somehow without having networking enabled yet.)
]]

--What I've got so far is that beancounters knows what its VE number is:

BEGIN VE 101 /etc/init.d/rc.local (PARTIAL)

```
do_start() {
    if [ -x /etc/rc.local ]; then
        [ "$VERBOSE" != no ] && log_begin_msg "Running local boot scrip
        /etc/rc.local
        ES=$?
        [ "$VERBOSE" != no ] && log_end_msg $ES
        return $ES
    fi

# Openvz mods: 2007.0528
/bin/rm /dev/shm/THISVE

thisve=`cat /proc/user_beancounters \
  |grep kmemsize \
  |awk '{ print $1 }'`

thisve=${thisve//:/} # Take out the trailing ":"

echo $thisve > /dev/shm/THISVE

# If THISVE < 1000, it is a Master Image.

# If not, consider it a clone:
# o Setup IP
#  and start listening.

}

case "$1" in
    start)
```

Subject: Re: TEST IDEA - Scaling with openvz
Posted by rickb on Tue, 29 May 2007 06:46:56 GMT
View Forum Message <> Reply to Message

Quote:is there a way in-VE to have it determine what its number is on the host?

Here is a quick hack I whipped up, run it from in the VE

#cat /proc/user_beancounters | grep kmem | sed 's/\t/ /g'| tr -s " " | cut -d" " -f2| sed 's/://g'

Subject: Re: TEST IDEA - Scaling with openvz
Posted by kingneutron on Tue, 29 May 2007 11:56:32 GMT
View Forum Message <> Reply to Message

Thanks; that is a welcome contribution and does what I needed.

I managed to do it in a few less steps tho, which is important for resource-management when running a lot of things simultaneously.

Now I just have to dig out the ol' Bash manual and noodle it to see if I can get it done w/o even having to call awk...

Subject: Re: TEST IDEA - Scaling with openvz
Posted by kingneutron on Tue, 29 May 2007 23:02:01 GMT
View Forum Message <> Reply to Message

--Latest pseudocode update, comments/advice are welcome:

BEGIN VE 101 /etc/init.d/rc.local (PARTIAL)

Openvz mods: 2007.0528
/bin/rm /dev/shm/THISVE

thisve=`cat /proc/user_beancounters \
  |grep kmemsize \
  |awk '{ print $1 }'`

thisve=${thisve//:/} # Take out the trailing ":"

```
echo $thisve > /dev/shm/THISVE

# source /etc/openvz/clonesetup/VARS
# If NOT "clonesetupran" then

  # If THISVE > 100 AND < 1000, it is a Master Image.

  # If not, consider it a clone:
  # o Setup IP
  #  and start listening:

  # sed /etc/network/interfaces
  # /etc/init.d/networking restart

  # cd / ; time nc -l -p $port |tar xpvf -
  # > /root/log-this-somewhere.log

  # After NC tarball...
  # If exist /root/bin/vz-postprocess-exec (can be sent as part of the tarball),
  #   Source it and run it.
  #   Then rename it so it is not processed anymore after subsequent reboots.

  # Regardless:
  # mkdir -pv /etc/openvz/clonesetup
  # echo 'clonesetupran=1' > /etc/openvz/clonesetup/VARS

# Else

}
```

kingneutron wrote on Tue, 29 May 2007 01:43[[
I'm pretty sure I can do this programmatically with simple Bash scripting; but is there a way in-VE
to have it determine what its number is on the host? (i.e. VE 1003 should "know" it's number is
1003 somehow without having networking enabled yet.)
]]

--What I've got so far is that beancounters knows what its VE number is:

BEGIN VE 101 /etc/init.d/rc.local (PARTIAL)

```
do_start() {
      if [ -x /etc/rc.local ]; then
           [ "$VERBOSE" != no ] && log_begin_msg "Running local boot scrip
           /etc/rc.local
           ES=$?
```

```
        [ "$VERBOSE" != no ] && log_end_msg $ES
        return $ES
    fi

# Openvz mods: 2007.0528
/bin/rm /dev/shm/THISVE

thisve=`cat /proc/user_beancounters \
  |grep kmemsize \
  |awk '{ print $1 }'`

thisve=${thisve//:/} # Take out the trailing ":"

echo $thisve > /dev/shm/THISVE

# If THISVE < 1000, it is a Master Image.

# If not, consider it a clone:
# o Setup IP
#  and start listening.

}

case "$1" in
    start)
```