Subject: Containers HOWTO? (Where do I start?)
Posted by Rob Landley on Wed, 08 Dec 2010 08:36:05 GMT
View Forum Message <> Reply to Message

I've been poking at the container code and reading the openvz wiki and the pages linked from http://lxc.sourceforge.net/index.php/about/kernel-namespaces / and random other things Google finds like http://www.howtoforge.com/how-to-do-live-migration-of-openvz -containers but I have yet to find a good "start here" document for what I want to do.

I want to build a containers test environment from source. I.E. I want to configure and build a kernel, build and configure my own root filesystem, invoke a chroot-with-benefits binary to set up a new init process in a container, and run test programs in the container.

I've found a lot of articles on how to use a distro to automate away the details, and I've found articles on several of the individual components, but if there's a design overview or a HOWTO on adding container support to a system you built yourself, I haven't found it yet. (A case insensitive search of Documentation didn't find a filename with "contain" or "vz" in it.)

I have no shortage of things to read, but it's all really unfocused. Any hints to help narrow my search?

Thanks,

Rob

_____
Containers mailing list
Containers@lists.linux-foundation.org
 https://lists.linux-foundation.org/mailman/listinfo/containe rs

Subject: RE: Containers HOWTO? (Where do I start?)
Posted by Rob Landley on Wed, 08 Dec 2010 21:17:45 GMT
View Forum Message <> Reply to Message

> A few places to start since you want to start from the ground up:
>
> 1. man clone

Yup, got that one already.

> 2. man pivot_root

I wrote the busybox switch_root implementation, which meant I had to
really _understand_ the full horror of the pivot_root implementation
and why it _wasn't_ switch_root (examining every process in the system
so you can move their chroot entry for kernel threads and daemons is
really ugly, but you can't unmount a filesystem that has processes
pinning it). Although come to think of it I thought kernel threads were

parked in initramfs these days? (Doesn't initrd overmount initramfs?
Now I have to go look at what the code's doing again...)

But how does pivot_root enter into this when you haven't got an initrd to
free? I thought when you killed a container's init process that killed all
the children and freed the resources, so how does pivot_root enter
into this? (You don't reparent existing processes, you span new ones,
right?)

> http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6 .git
;a=tree;f=Documentation/cgroups;h=8c6b3f6c41a929f8db38b51a39 442387ecbd5986;hb=HEAD

Ah, _that_ is what i was looking for. Documentation/cgroups.

I'll look at #3 and #5 after reading that.

Thanks,

Rob

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs