Subject: Re: [PATCH 0/6] containers: Generic Process Containers (V6)
Posted by serue on Fri, 12 Jan 2007 18:42:07 GMT

View Forum Message <> Reply to Message

Quoting Paul Menage (menage@google.com):
> Hi Serge,
>
> On 1/3/07, Serge E. Hallyn <serue@us.ibm.com> wrote:
> >From: Serge E. Hallyn <serue@us.ibm.com>
> >Subject: [RFC] [PATCH 1/1] container: define a namespace container
> >subsystem
> >
> >Here's a stab at a namespace container subsystem based on
> >Paul Menage's containers patch, just to experiment with
> >how semantics suit what we want.
>
> Thanks for looking at this.
>
> What you have here is the basic boilerplate for any generic container
> subsystem. I realise that my current containers patch has some
> incompatibilities with the way that nsproxy wants to work.

In retrospect I don't like the changes in behavior. So my next
version will aim for closer to the original (non-containerfs)
behavior.

> >A few things we'll want to address:
> >
> >   1. We'll want to be able to hook things like
> >   rmdir, so that we can rm -rf /containers/vserver1
> >   to kill all processes in that container and all
> >   child containers.
>
> The current model is that rmdir fails if there are any processes still
> in the container; so you'd have to kill processes by looking for pids
> in the "tasks" info file. This was behaviour inherited from the
> cpusets code; I'd be open to making this more configurable (e.g.
> specifying that rmdir should try to kill any remaining tasks).

Ok - of course I suspect I'll have to just start coding away before
i can guess at what help I might need from your code.

> >
> >   2. We need a semantic difference between attaching
> >   to a container, and being the first to join the
> >   container you just created.
>
> Right - the way to do this would probably be some kind of

> "container_clone()" function that duplicates the properties of the
> current container in a child, and immediately moves the current
> process into that container.
>
> >   3. We will want to be able to give the container
> >   attach function more info, so that we can ask to
> >   attach to just the network namespace, but none of
> >   the others, in the container we're attaching to.
>
> If you want to be able to attach to different namespaces separately,
> then possibly they should be separate container subsystems?

That's one possibility, but imo somewhat unpalatable.

As I mentioned in the last email, I really like the idea of having
files representing each namespace under each namespace container
directory, creating a new container by linking some of those
namespace files, and entering containers by echoing the pathname
to the new container into /proc/$$/ns_container. (either upon
the echo, or, I think preferably, upon a subsequent exec)

-serge