Subject: Re: Re: containers and cgroups mini-summit @ Linux Plumbers
Posted by Serge E. Hallyn on Thu, 26 Jul 2012 18:16:29 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Glauber Costa <glommer@parallels.com> writes:
>
> >>>
> >>> Another old issue is that unless I have missed something control groups
> >>> are still broken for generic use in containers. Does anyone care?
> >>> Are there any plans on fixing this issue?
> >>>
> >
> > What is "generic use in containers" ? I am using them alright, but not
> > sure if this counts as generic or specific =)
>
> The general container use case would be.
>
> - Create a new mount namespace.
> - Create fresh mounts of all of the control groups like I would do at
>   boot, with no consideration to any other control group state.
> - Start forking processes.
>
> The expected semantics would be something like chroot for control
> groups, where all of the control groups that are created by fresh mounts
> are relative to whatever state the process of being in a control group
> that the process that mounted them was in.
>
> Last I looked the closest you could come to that was bind mounts, and
> even with bind mounts you get into weird things where control groups are
> bound into hierarchies and you may be running a distribution that wants
> it's control groups bound into different hierarchies.
>
> Last I looked this was just about a total disaster, and the only thing
> that allowed systemd to run in containers was the fact that systemd did
> not user controllers.
>
> Eric

(Sorry, please disregard my last email :)

Yes, what we do now in ubuntu quantal is the bind mounts you mention,
and only optionally (using a startup hook).
Each container is brought up in say
/sys/fs/cgroup/devices/lxc/container1/container1.real, and that dir is
bind-mounted under /sys/fs/cgroup/devices in the guest. The guest
is not allowed to mount cgroup fs himself.

It's certainly not ideal (and in cases where cgroup allows you to raise your own limits, worthless). The 'fake cgroup root' has been mentioned before to address this. Definately worth discussing.

thanks,
-serge
