Subject: Re: [RFD] reboot / shutdown of a container
Posted by Bruno Pr on Thu, 13 Jan 2011 20:09:18 GMT
View Forum Message <> Reply to Message

On Thu, 13 January 2011 Daniel Lezcano <daniel.lezcano@free.fr> wrote:
> in the container implementation, we are facing the problem of a process
> calling the sys_reboot syscall which of course makes the host to
> poweroff/reboot.
>
> If we drop the cap_sys_reboot capability, sys_reboot fails and the
> container reach a shutdown state but the init process stay there, hence
> the container becomes stuck waiting indefinitely the process '1' to exit.
>
> The current implementation to make the shutdown / reboot of the
> container to work is we watch, from a process outside of the container,
> the <rootfs>/var/run/utmp file and check the runlevel each time the file
> changes. When the 'reboot' or 'shutdown' level is detected, we wait for
> a single remaining in the container and then we kill it.
>
> That works but this is not efficient in case of a large number of
> containers as we will have to watch a lot of utmp files. In addition,
> the /var/run directory must *not* mounted as tmpfs in the distro.
> Unfortunately, it is the default setup on most of the distros and tends
> to generalize. That implies, the rootfs init's scripts must be modified
> for the container when we put in place its rootfs and as /var/run is
> supposed to be a tmpfs, most of the applications do not cleanup the
> directory, so we need to add extra services to wipeout the files.
>
> More problems arise when we do an upgrade of the distro inside the
> container, because all the setup we made at creation time will be lost.
> The upgrade overwrite the scripts, the fstab and so on.
>
> We did what was possible to solve the problem from userspace but we
> reach always a limit because there are different implementations of the
> 'init' process and the init's scripts differ from a distro to another
> and the same with the versions.
>
> We think this problem can only be solved from the kernel.
>
> The idea was to send a signal SIGPWR to the parent of the pid '1' of the
> pid namespace when the sys_reboot is called. Of course that won't occur
> for the init pid namespace.

Wouldn't sending SIGKILL to the pid '1' process of the originating PID
namespace be sufficient (that would trigger a SIGCHLD for the parent
process in the outer PID namespace.
(as far as I remember the PID namespace is killed when its 'init' exits,
if this is not the case all other processes in the given namespace would

have to be killed as well)

Only issue is how to differentiate the various reboot() modes (restart, power-off/halt) from outside, though that one also exists with the SIGPWR signal.

Bruno

> Does it make sense ?
>
> Any idea is very welcome :)
>
> -- Daniel

_____

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