

Playing on the frontier

A log of random stuff that interests me.

Docker on LXC – Use the LXC execution driver

Running docker inside LXC containers In my have [proven](#) to be [no small task](#) container, at least in my setup of a Gentoo host and Ubuntu LXC guest. One of the remaining issues is the `cpuset.cpus` error with Docker 1.4 and 1.5



```
1 | /sys/fs/cgroup/cpu/lxc/cpuset/cpuset.cpus: no such file or directory
```

I have found a way to get around this issue.

Docker in its younger days was dependent on LXC to create its containers. That changed some time ago when it [migrated to using its own code in the form of libcontainer](#). You could however still opt to use LXC. A change to the execution driver from native (libcontainer) to lxc and the `cpuset.cpus` error was gone. You can check your current execution driver by running `docker info`. native-0.2 is libcontainer:

```
1 | # docker info
2 | Containers: 22
3 | Images: 3
4 | Storage Driver: overlay
5 | Backing Filesystem: extfs
6 | Execution Driver: native-0.2
7 | Kernel Version: 3.18.7-gentoo
8 | Operating System: Gentoo/Linux (containerized)
9 | CPUs: 4
10 | Total Memory: 7.735 GiB
```

To change the execution driver, you can run the docker daemon with the “-e lxc” option, or add it to the config of your startup script. In Gentoo this was `/etc/conf.d/docker`. Change `DOCKER_OPTS`:

```
1 | # any other random options you want to pass to docker
2 | DOCKER_OPTS="-e lxc"
```

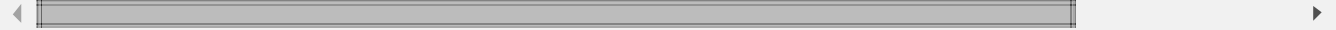
After the change, `docker info` should show LXC:

```
1 | # docker info
2 | Containers: 22
3 | Images: 3
4 | Storage Driver: overlay
5 | Backing Filesystem: extfs
6 | Execution Driver: lxc-1.0.7
7 | Kernel Version: 3.18.7-gentoo
8 | Operating System: Gentoo/Linux (containerized)
9 | CPUs: 4
10 | Total Memory: 7.735 GiB
```

However, after switching to LXC, I still had a few hurdles to jump through. First, there was a bug that made LXC look for sysRq when it doesn't exist. This is a [LXC bug](#). For now, I recompiled my kernel with CONFIG_MAGIC_SYSRQ=y. You can find the magic key option under "Kernel development" menu.

Next issue was to do with LXC failing to set devices.allow:

```
1 | lxc-start: cgfs.c: do_setup_cgroup_limits: 1908 Error setting devices.allow to c
```

A terminal window showing a command prompt with a vertical bar, followed by the error message: lxc-start: cgfs.c: do_setup_cgroup_limits: 1908 Error setting devices.allow to c. The terminal has a light gray background and a dark gray border.

To fix this, I added lxc.cgroup.devices.allow = c 4:* rwm to the LXC config of the parent container.

This was enough for my Gentoo LXC guest to run docker. For my Ubuntu LXC guest, I got this error:

```
1 | lxc-cgmanager.c: lxc_cgmanager_escape: 329 call to cgmanager_move_pid_abs_sync(n
```

A terminal window showing a command prompt with a vertical bar, followed by the error message: lxc-cgmanager.c: lxc_cgmanager_escape: 329 call to cgmanager_move_pid_abs_sync(n. The terminal has a light gray background and a dark gray border.

so I disabled CGManager:

```
1 | initctl stop cgmanager
2 | echo manual | sudo tee /etc/init/cgmanager.override
```

Now I can use all the [latest docker features like overlays!](#)

This entry was posted in [docker](#) and tagged [docker](#), [gentoo](#), [lxc](#), [ubuntu](#) on March 15, 2015

[<http://siphon9.net/loune/2015/03/docker-on-lxc-use-the-lxc-execution-driver/>] .