5.4 Docker in Relation to LXC (Linux Container)

Quick notes in Docker in relation to LXC.

Docker started as a front-end application virtualization engine based on OS level virtualization. Docker started with LXC (Linux Containers) under the hoods but since version 0.9 switched to it's own libcontainer as the default virtual engine.

The two technologies appear to have similar functions. Initially I had a hard time wrapping my head around why to use Docker over LXC which to me is much more flexible and I don't lose ssh access.

However, interestingly enough as I used LXC more, I started to appreciate Docker's marketing (which will mean it will do well), documentation and spirit of being **restrictive**. From the view of writing applications and operating them:

- Ephemeral - Being stateless you build must externalize the right things (logs, application property files) from day 1.
- Single Process as Best Practice - I lose SSH access, but I then stop thinking of stacks (OS, Java, JEE Server, Application Code) and instead of a single package.
- No Upgrading, Just Rebuild your Day 1 script to update your application and OS creating consistency.

It's a fundamental mind shift. If forces both Ops and Developers into a repeatable automated approach to building servers and coding applications. This shifts work instead to using good orchestration software (which I am in process of researching). The Docker base OS template is pared down to a single app environment and does not endorse an init, services, daemons, syslog, cron or running multiple applications.

Again though, I'll note all these feature is possible with LXC and by the way (with a slightly different spin) happens with Cloud Foundry powered by Docker.

You can run Dockers inside of Linux containers with virtually no performance impact.

Where I use LXC is for long running systems and creating isolation density within the Cloud. For example, a bunch of LXC's inside of an Azure VM which further separates my apps (maybe running inside of Docker) btw PreProd and Production.

Which one to learn first? If you want to really understand things, learn LXC and you'll know Docker. If using Cloud Foundry, learn LXC too. It's like starting with DOS then learning Windows.

Is one better than the other. No, they have different uses.

References

See my hosting page for Virtualization technologies.

LXC vs Docker - http://www.flockport.com/lxc-vs-docker/

Discusses security - https://blog.docker.com/2013/08/containers-docker-how-secure-are-they/