Contribution ID: **597** Contribution code: **contribution ID 597** Type: **Poster**

OpenTop —Towards Composable and Configurable Containerisation

Containerisation is an elementary tool for sharing IT resources: It is more light-weight than full virtualisation, but offers comparable isolation. We argue that for many use-cases which are typically approached with standard containerisation tools, less than full isolation is sufficient: Sometimes, only networking or only storage or both need to be different from their native, unisolated state. An example is a chroot: It only isolates storage by presenting a different root file system to userspace, but doesn't interfere with other aspects.

We present OpenTop, an ongoing effort towards composable and configurable containerisation. It was decided to focus primarily on configurable networking; later research will add the remaining aspects. A domain specific language (DSL) is proposed to describe the network topology of "containers", including their network devices, routes, firewalls/packet filters etc. All the common tools for network administration (if applicable) should be available.

Only elementary techniques are used, e. g., Linux namespaces, control groups, iptables/nftables and virtual extensible LANs with veth pairs. For simplicity of use, these tools are applied directly without involving any frameworks, libraries (besides the libc) or init systems. This approach is deemed to be both more portable to different Linux distributions and more easily debugged/audited (both for developers and users of OpenTop). It also minimises attack surface, since there is no runtime demon involved.

Significance

1. The usual trend in container technology is to increasingly more aspects. This can go almost all the way to virtualisation sans the operating system. The proposed approach wants to reverse this trend and have containers with minimal weight (and therefore the right amount of isolation for the individual use case).
2. This tool should depend on as little infrastructure as possible; it does its job of setting up/tearing down and disappears. This is different from most container tools in common use which require a demon. Instead, the tool is generated based on a user-supplied specification.
3. Minimise "bloat": If a certain feature is not needed, no infrastructure for it is provided; e. g. no packet filters for inter-container communication, if no inter-container communication requested. The typical tools follow an all-encompassing template, whether needed or not.

References

Rosen, R. (2013). Resource management: Linux kernel namespaces and cgroups. Haifux, May, 186, 70.

Speaker time zone

Compatible with Europe

Primary authors: JAFFE, Ludwig Albert (Goethe University Frankfurt (DE)); ADLER, Alexander (Goethe University Frankfurt (DE)); KEBSCHULL, Udo Wolfgang (Goethe University Frankfurt (DE))

Presenters: JAFFE, Ludwig Albert (Goethe University Frankfurt (DE)); ADLER, Alexander (Goethe University Frankfurt (DE))

Session Classification: Posters: Broccoli

Track Classification: Track 1: Computing Technology for Physics Research