

Evaluation verschiedener Container-Technologien

Corvin Schapöhler*, Kai Warendorf, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2018

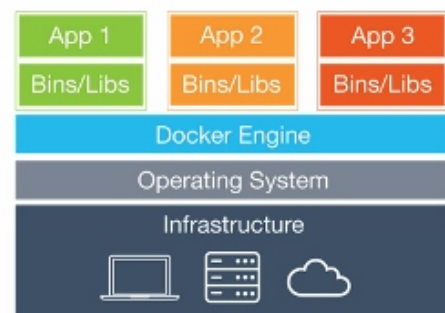
Der Trend zur Cloud ist unumkehrbar. Bereits heute setzen Firmen wie Microsoft verstärkt auf das Cloudgeschäft [1]. Die treibende Technologie hinter diesem Trend sind Container. Im folgenden wird diese Technologie genauer betrachtet. Dabei soll die Frage beantwortet werden, wie Docker die populärste Technologie wurde, welche aktuellen Probleme bestehen und wie versucht wird, diese zu lösen.

Container und Virtuelle Maschinen

Container dienen der Isolation von Prozessen. Dabei sind sie ressourcensparender und deutlich schneller als Virtuelle Maschinen. Diese Einsparungen werden erreicht, indem nicht ein gesamtes Betriebssystem virtualisiert, sondern lediglich eine Isolation einzelner Systemkomponenten vorgenommen wird.



Virtual Machines



Containers

Abbildung 1: Vergleich: Virutalisierung von VM und Isolation von Container

Durch diese Methode sind Container hoch skalierbar und können binnen Bruchteile von Sekunden gestartet oder zerstört werden. Diese Skalierbarkeit und Isolation sind die tragenden Faktoren der heutigen Cloud-Architekturen.

Historische Entwicklung

Die grundlegenden Konzepte zur Isolation wurde bereits in den frühen 80er mit der Einführung des Unix Systemaufrufs chroot implementiert. Der erste große Durchbruch kam allerdings erst 2008 mit LXC [2]. Figure 2 Timeline der wichtigsten Container Releases

*Diese Arbeit wurde durchgeführt bei der Firma NovaTec GmbH, Leinfelden-Echterdingen

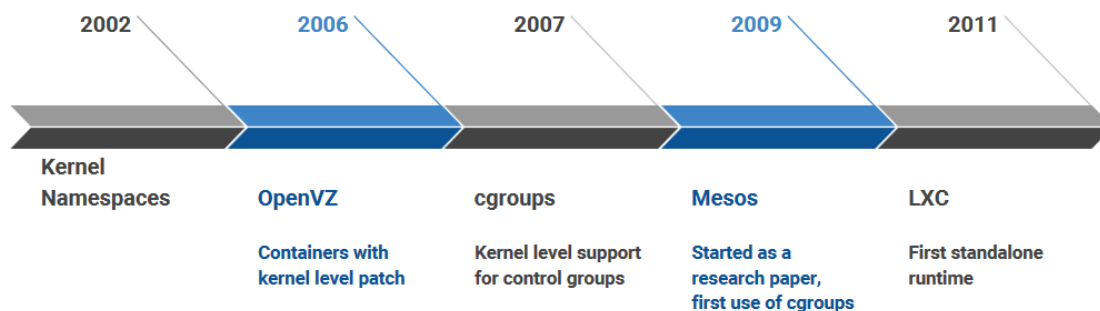


Abbildung 2: Timeline der wichtigsten Konzepte zur Isolation im Linux-Kernel

Dieses implementiert eine vollständige Isolation des Linux-Kernels. In den folgenden Jahren wurden die unterliegenden Konzepte erweitert und vereinfacht. Die Popularität, die Container heute für sich haben kam mit dem Release der Container-Plattform Docker im Jahr 2013 [2]. Docker setzte zu Beginn auf LXC auf, wechselte bald aber zu einer eigenen Implementierung. Zudem bietet Docker mit dem Docker Hub eine SaaS-Plattform, die die Wiederverwendung von Containern ermöglicht.

Alternativen zu Docker

Neben Docker hat sich ein großer Markt an Alternativen Container-Runtimes gebildet. So bietet CoreOS mit rkt eine Alternative, die auf Sicherheit und Wiederverwendbarkeit einzelner Container setzt. Zudem setzt rkt darauf, eine der ersten Container-Runtimes zu sein, die den AppContainer Standard implementiert. Eine weitere Alternative bietet Canonical mit LXD, einer Weiterführung der Container-Runtime LXC. LXD setzt, recht ähnlich zu ersten Versionen von Docker auf eine RESTful API zur Steuerung von LXC. Im Gegensatz zu rkt oder Docker ist LXD dafür gedacht, komplette Linux-Distributionen zu isolieren.

Durch das rapide Wachstum am Interesse für Container wurden Standards für diese gefordert. Neben AppContainer wurde 2015 unter der Obhut der Linux Foundation die Open Container Initiative gegründet [3]. Diese definiert Standards für Container-Images und Runtimes, die durch die Implementierung runc genutzt wird. Mittlerweile nutzen Docker und viele weitere Tools die Spezifikation und bauen ihre Angebote auf runc um.

Aktuelle Probleme und Lösungen

Die enorme Popularität von Container und die ansteigende Nutzung werden auch viele Probleme mit dieser Art der Isolation erkenntlich. Bereits 2014 startete Google alle Dienste in Containern und musste somit jede Woche zwei Milliarden Container verwalten [4]. Zudem werden zunehmend Sicherheitslücken im Linux-Kernel bekannt, die es ermöglichen aus der Isolation auszubrechen. Um diesen Problemen entgegenzuwirken werden Orchestrierungstools wie Kubernetes und Container-Runtimes wie gVisor entwickelt, die mehr Sicherheit und vereinfachte Verwaltung versprechen.

- [1] Satya Nadella. *Annual Report 2017*. Financial Report. Microsoft, 2017. url: <https://www.microsoft.com/investor/reports/ar17/index.html> (besucht am 09.05.2018).
- [2] Rani Osnat. *A Brief History of Containers: From the 1970s to 2017*. Blog. Mar. 21, 2018. url: <https://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016> (besucht am 09.05.2018).
- [3] Open Container Initiative. *Open Container Initiative*. The Linux Foundation. 2018. url: <https://www.opencontainers.org/> (besucht am 09.05.2018)
- [4] Joe Beda. *Containers At Scale. At Google, the Google Cloud Platform and Beyond*. In: GlueCon 2014. May 22, 2014. url: <https://speakerdeck.com/jbeda/containers-at-scale> (besucht am 09.05.2018).

Bildquellen:

- Abbildung 1: http://www.serverspace.co.uk/hubfs/Blog_Images/container-vs-vm.jpg?t=1524959203460
- Abbildung 2: https://sched.ws/hosted_files/kccnceu18/08/What%E2%80%99s%20Up%20With%20All%20the%20Container%20Runtimes.pdf