Hochschule
Albstadt-Sigmaringen

Albstadt-Sigmaringen University

Praktische Arbeit zur vorbereitenden Blockveranstaltung

Software-Container und Software-Development

Funktion von Software-Container und deren Einsatz in Entwicklung und Produktion

Autoren:

Maximilian Rieger
Technische Informatik
85581

Florian Lubitz
Technische Informatik
85900

Thomas Schöller
Technische Informatik
87113

Marc Bitzer
Technische Informatik
87117

Jonas Acker
Technische Informatik
85583

Inhaltsverzeichnis

1 Einleitung

Bis kurz vor der Jahrtausendwende führte die Virtualisierung von Servern ein Schattendasein und jeder Service wurde auf einem dedizierten Server zur Verfügung gestellt. Dabei war es keine Seltenheit, dass Server sehr gering ausgelastet waren, da der laufende Service nicht die gesamte Leistung der Hardware benötigte und der Ausfall eines nicht redundanten Servers einen Totalausfall eines Services bedeutete. Eine Beispielhafte dedizierte Serverkonstellation stellt folgende Grafik dar:

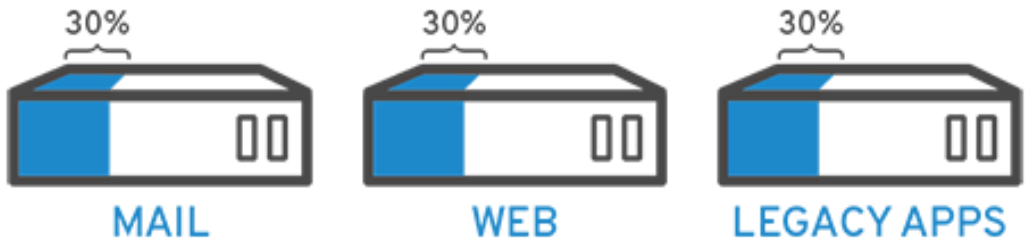

Abbildung 1: Serverauslastung ohne Virtualisierung [1]

[1] Quelle: https://www.redhat.com/cms/managed-files/server-usage-500x131.png

Um diese und weitere Probleme zu lösen, gewann die Virtualisierung von Servern zum Anfang des neuen Jahrtausends immer mehr an Bedeutung und ist heutzutage ein fester Bestandteil vieler großer Unternehmen. Dabei werden auf einem physikalischen System mehrere Dienste zusammengefasst, die sonst nur einen Bruchteil der Leistung benötigen würden. Dadurch kommen noch andere Vorteile wie z.B. das Erstellen von Snapshots und das dynamische Verschieben der virtuellen Maschinen zum tragen die folgende Grafik zeigt die Auslastung der virtualisieren Server:

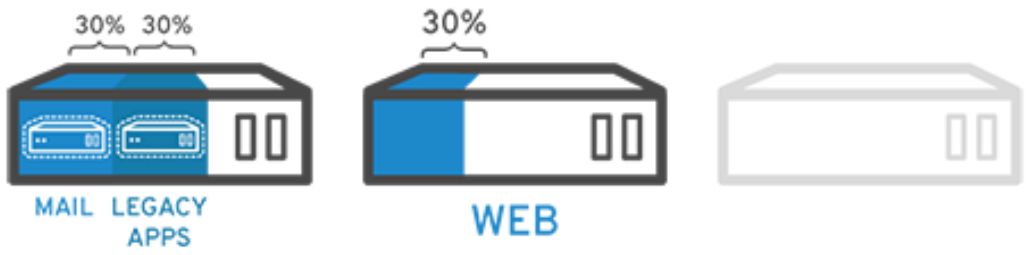

Abbildung 2: Serverauslastung mit Virtualisierung [2]

[2] Quelle: https://www.redhat.com/cms/managed-files/server-usage-for-virtualization-500x131.png

Doch auch die Virtualisierung von Servern birgt noch verbesserbare Nachteile. So entsteht durch das Betriebssystem der virtuellen Maschinen ein deutlicher Overhead, da diese zur Laufzeit etliche Services benötigen. Durch die virtualisierten Betriebssysteme wird die Hardware deutlich mehr beansprucht und die Startzeit ist relativ lange. Somit war die IT-Welt nicht in der Lage, wozu die Menschheit längst in der Lage ist: Güter in Container verpacken und diese Container aufgrund dem Standardisierten Format auf den verschiedensten Verkehrswegen zu transportieren. Um diese Lösung in die IT zu portieren, wurden auch hierfür sogenannte Software-Container entwickelt. Software-Container setzen wie die Schiffscontainer an dem Punkt Portabilität an. Es soll nicht für jeden Service ein zusätzliches Betriebssystem virtualisiert werden, sondern der Container soll nur das zusätzlich beinhalten was er für den Service benötigt und trotzdem isoliert von den anderen Container auf der Hardware laufen. Außerdem soll es wie bei den virtuellen Maschinen möglich sein, dynamisch Ressourcen zuzuweisen.Edwards [2016]; redhat Die folgenden Grafik verdeutlicht nochmals den eingesparten Overhead bei Containern verglichen mit virtuellen Maschinen:

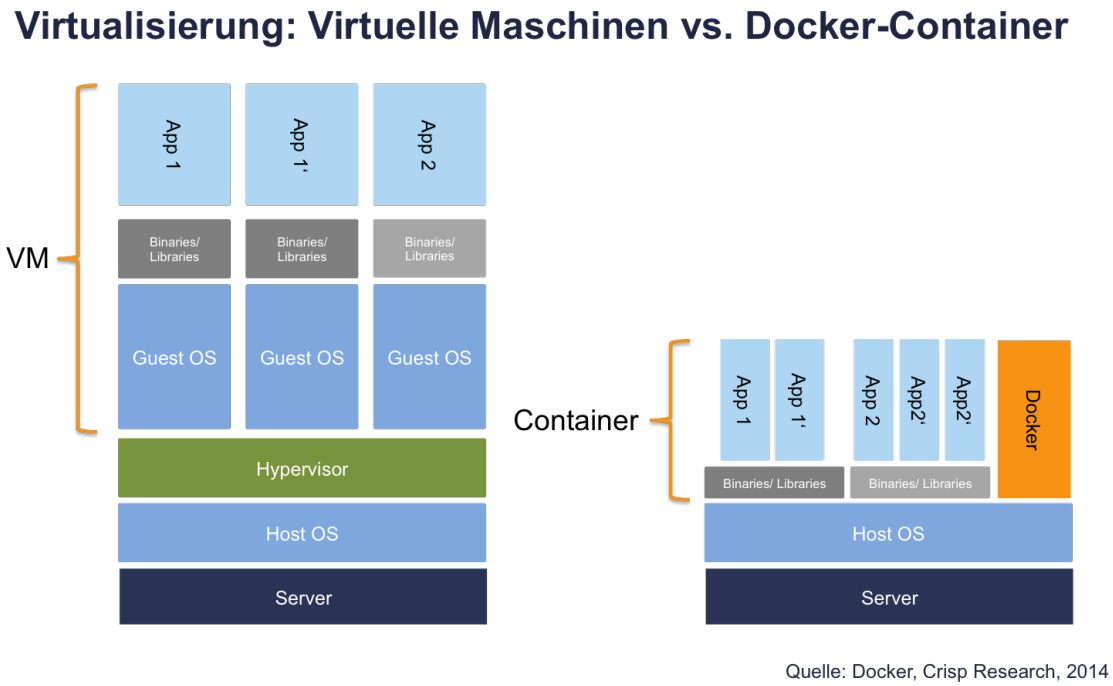

Abbildung 3: Vergleich Container und VM [3]

2 Funktionalität von Container

Container setzen direkt auf dem Kernel eines Linux-Betriebssystems auf. Um auf den Kernel durchgreifen zu können, verwenden Container standard-Linux-Techniken wie Cgroups und Namespaces oder selbst entwickelte Schnittstellen. Dadurch wird das Betriebssystem innerhalb des Containers, ohne vollständige Virtualisierung emuliert. Die folgende Grafik verdeutlicht den Kernelzugriff:

[3] Quelle: https://images.computerwoche.de/bdb/2668601/738x415_f5f5f5.jpg

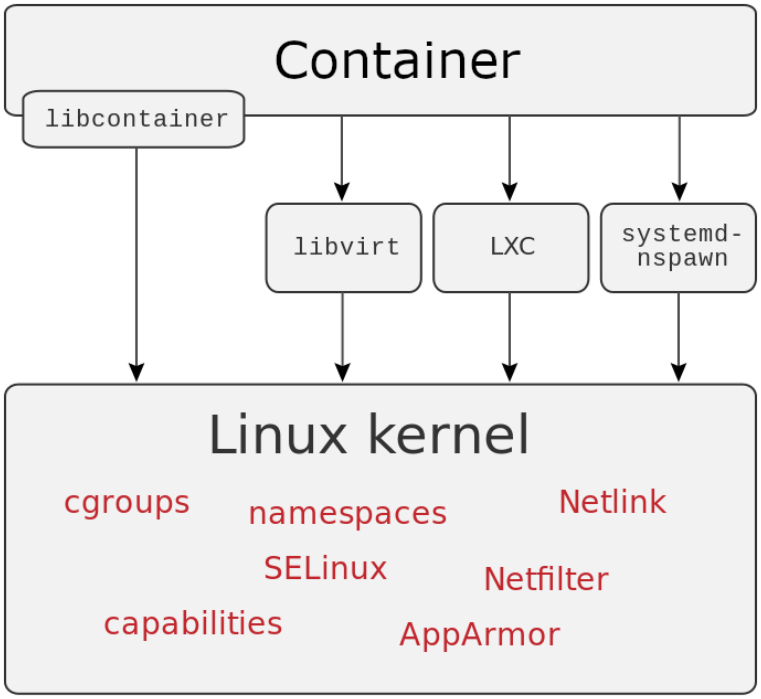

Abbildung 4: Schnittstelle vom Container zum Kernel [4]

Somit können CPU-Zyklen, Arbeitsspeicher, Blockspeicher und sonstige Schnittstellen über den Kernel angefordert und isoliert in dem jeweiligen Container zur Verfügung gestellt werden. DATACENTERINSIDER

Da sie nur als einzelnes Image abgelegt sind, und kein Betriebssystem beinhalten, welches aktualisiert und gewartet werden müsste, beschränken sich die Installation und Deinstallation auf ein einfaches kopieren oder löschen des Containers. Aus einem Image können beliebig viele Container-Instanzen aufgerufen werden, da Schreibzugriffe nicht auf das Image auf das Image zugreifen sondern auf ein eigenes Dateisystem des Containers. Dieses Verhalten sorgt für eine sehr hohe Skalierbarkeit, da bei Bedarf einfach neue Instanzen der Anwendung gestartet werden können. DEVINSIDER

[4] Quelle: https://www.datacenter-insider.de/container-technik-docker-co-a-480855/index2.html

3 Containertechnologien

Im Laufe der Containertechnologie traten verschiedene Implementierungen derselben auf. Hierbei waren die ersten Umsetzungen noch sehr einfach aufgebaut und wurden mit den Anforderungen an die Containerdienste immer komplexer. Im Folgenden findet sich eine Übersicht über die wichtigsten Technologien in der Containerisierung.

Abbildung 5: Containertechnologie im Laufe der Zeit

3.1 chroot

Chroot ist ein Befehl, der schon früh in Unix-Systemen eingebaut wurde. Er ermöglicht es einem Prozess ein andres Rootverzeichnis zu geben. Wird in einem Programm `chroot()` aufgerufen wechselt es das Verzeichnis und kann nicht auf Dateien außerhalb der zugewiesenen Struktur zugreifen. Diese Abschottung eines Prozess war nie als Sicherheitsfeature vorgesehen und wird hauptsächlich zur Virtualisierung eingesetzt. Mit dem Befehl können einzelne Prozess auf Dateiebene von anderen Anwendungen getrennt werden, weitere Sicherheitsmechanismen oder Isolierungen gibt es nicht.KAUR UND SINGH [2016]; SMITH [1996]; MANPAGES

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

Literaturverzeichnis

DataCenterInsider

DataCenterInsider: *Container-Technik: Docker und Co.* https://www.datacenter-insider.de/container-technik-docker-co-a-480855/, Abruf: 24.07.2018

DevInsider

DevInsider: *Was sind Docker-Container?* https://www.dev-insider.de/was-sind-docker-container-a-597762/, Abruf: 24.07.2018

Edwards 2016

EDWARDS, Chris: Containers Push Toward the Mayfly Server. In: *Communications of the ACM* 59 (2016), Nr. 12, 24 - 26. http://www.redi-bw.de/db/ebsco.php/search.ebscohost.com/login.aspx%3fdirect%3dtrue%26db%3degs%26AN%3d120050683%26site%3dehost-live. – ISSN 00010782

Kaur und Singh 2016

KAUR, N. ; SINGH, M.: Improved file system security through restrictive access. In: *2016 International Conference on Inventive Computation Technologies (ICICT)* Bd. 3, 2016, S. 1–5

Manpages

MANPAGES, Linux: *chroot - Wurzelverzeichnis wechseln*. https://manpages.debian.org/stretch/manpages-de-dev/chroot.2.de.html, Abruf: 24.07.2018

redhat

REDHAT: *Was ist Virtualisierung?* https://www.redhat.com/de/topics/virtualization/what-is-virtualization, Abruf: 24.07.2018

Smith 1996

SMITH, R. E.: Mandatory protection for Internet server software. In: *Proceedings 12th Annual Computer Security Applications Conference*, 1996. – ISSN 1063–9527, S. 178–184

A Anhang

A.1 Begründung der ausgewählten Literatur

Zur Verfügung stand lediglich sehr aktuelle Literatur, da die Containervirtualisierung erst seit Erscheinen von Docker im Jahr 2013 in der IT-Branche an Bedeutung gewonnen hat. Daher sind auch viele der hier betrachteten Werkzeuge erst in den vergangen Jahren entwickelt worden.

In Anbetracht des kurzen Zeitraumes, der den Autoren zur Verfügung stand, konnte keine Fernleihe durchgeführt werden. Eine Vorbestellung der Literatur war daher ebenfalls nicht möglich. Am ersten Tag der Bearbeitung des Artikels stand außerdem die Bibliothek aufgrund des Betriebsausflugs nicht zur Verfügung, weshalb auch nicht auf die physischen Medien zurückgegriffen werden konnte. Deshalb hat sich die Bücher- bzw. Artikelauswahl auf die über die Hochschule verfügbaren digitalen Medien beschränkt.

Die Literaturauswahl umfasst außerdem Dokumentationen der gängisten Software zum Thema Container-Technologie. Diese wurde zum Verständnis des Aufbaus und der Nutzung des jeweiligen Werkzeugs genutzt. Die Dokumentationen sind online bzw. zusammen mit dem jeweiligen Source Code verfügbar und werden von den Entwicklern zur Verfügung gestellt. Daher handelt sich bei den Dokumentationen um eine verlässliche Quelle über das jeweilige Werkzeug.

Blog Einträge dienten den Autoren als Ideengeber für einen Teil des Inhalts der vorliegenden Arbeit. Da diese am Puls der Zeit sind, zeigen sie aktuelle Trends und populäre Software zum Thema Container-Technologie auf. Ein Blog wird nicht überprüft und stellt daher selbstverständlich keine zuverlässige Quelle dar. Zur weiteren Recherche wurden aufgrund dessen wissenschaftlich verlässliche Quellen verwendet.