



Fakultät Informatik

Softwaretechnik und Medieninformatik

Ausarbeitung zum Thema

Evaluierung verschiedener Container Technologien

Corvin Schapöhler

751301

Semester 2018

Firma: NovaTec GmbH

Betreuer: Dipl.-Ing. Matthias Haeussler

Erstprüfer: Prof. Dr.-Ing. Dipl.-Inform. Kai Warendorf

Zweitprüfer: Prof. Dr. Dipl.-Inform. Dominik Schoop

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit von mir selbstständig und, ohne Hilfe Dritter und ausschließlich unter Verwendung der angegebenen Quellen angefertigt wurde. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen sind habe ich als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form, auch nicht in Teilen, keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Stuttgart, 9. März 2018

Ort, Datum

Corvin Schapöhler

Inhaltsverzeichnis

Kurzfassung	1
Abstract	1
1 Einleitung	2
2 Grundlagen	3
2.1 Standards	3
2.1.1 Open Container Initiative	3
2.1.2 Cloud Native Computing Foundation	3
2.2 Funktionsweise	3
2.3 Eigene Implementierung	3
Abbildungsverzeichnis	4
Tabellenverzeichnis	5
Listings	6
Akronyme	7

Kurzfassung

Stichwörter: Container, Docker, rkt, Cloud, OCI, cgroups, Linux

Abstract

Keywords: Container, Docker, rkt, Cloud, OCI, cgroup, Linux

1 Einleitung

Die Welt wird immer stärker vernetzt. Durch den Drang, Anwendungen für viele Nutzer zugänglich zu machen besteht der Bedarf an Cloud-Diensten wie Amazon Web Services (AWS). Eine dabei immer wieder auftretende Schwierigkeit ist es, die Skalierbarkeit des Services zu gewährleisten. Selbst wenn viele Nutzer zeitgleich auch einen Service zugreifen, darf dieser nicht unter der Last zusammen brechen.

Bis vor einigen Jahren wurde diese Skalierbarkeit durch Virtuelle Maschinen (VMs) gewährleistet. Doch neben großem Konfigurationsaufwand haben VMs auch einen großen Footprint und sind für viele Anwendungen zu ineffizient. Eine Lösung für dieses Problem stellen Container.

Diese Arbeit untersucht die Möglichkeiten, die Container dem Entwicklungszyklus geben. Um ein grundlegendes Verständnis der Technologie zu vermitteln, werden die verwendeten Techniken hinter Container zu Beginn der Arbeit erklärt und anhand eines praktischen Beispiels näher beleuchtet. Dazu wird ein eigener Prozess in einer Linuxumgebung isoliert und so eine eigene Container-Laufzeitumgebung für diesen geschaffen.

Im Anschluss wird ein historischer Blick auf die Container-Landschaft gegeben und untersucht, wie Docker der erfolgreichste Vertreter dieser Technologie wurde. Dabei wird vor allem auf die Anfänge der Isolierung und Virtualisierung eingegangen und historische Konzepte wie control groups (cgroups) näher beleuchtet.

Im weiteren Teil der Ausarbeitung werden Vergleiche zwischen verschiedenen aktuell verwendeten Container-Laufzeitumgebungen gezogen und aufgezeigt, wie gut diese auf dem Markt aufgenommen werden. Dabei wird auch ein Blick in verschiedene Container-Orchestrierungsplattformen gegeben und aufgezeigt, welche Plattformen welche Technologien unterstützen.

2 Grundlagen

Dieses Kapitel behandelt alle Grundlagen, die für Linux Container benötigt werden. Dabei liegt der Schwerpunkt auf den bestehenden Standards, die Funktionsweise hinter Containern und der Vorgehensweise, um eigene isolierte Prozesse zu instanziiieren.

Um ein besseres Verständnis für die Funktionsweise und die benötigten Technologien zu geben, wird zudem behandelt, wie man eigene Prozesse in einem Unix System vollständig unabhängig und isoliert voneinander laufen lassen kann und so die Separation erhält, die Container attraktiv machen.

2.1 Standards

2.1.1 Open Container Initiative

2.1.2 Cloud Native Computing Foundation

2.2 Funktionsweise

2.3 Eigene Implementierung

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Akronyme

AWS Amazon Web Services. 2

cgroup control group. 2

VM Virtuelle Machine. 2