Independent Coursework

Trends in Container-Virtulisierung

bearbeitet von:	Tom Schubert
Studiengang:	Angewandte Informatik (Master)
Fachbereich:	Wirtschaftswisenschaften II
Matrikelnummer:	535279
zuständiger Prof.:	Prof. Dr. Hermann Heßling
zuständige Mitarbeiter (DESY):	Patrick Furhmann
	Yves Kemp
Datum (Version):	June 12, 2015 (Ver. 0.0)

Contents

1 Grundlagen

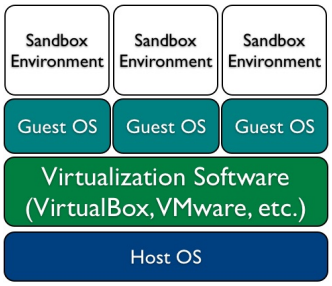
In diesem Kapitel werden die Grundlagen rund um das Thema Container erklärt. Hierzu wird darauf eingegangen, weshalb Container verwendet werden und welche Vorteile sie gegenüber anderen Verfahren haben.
Verschiedene Containerarten werden gezeigt und erklärt, in welchen Aspekten diese sich unterscheiden und / oder ähneln.

1.1 Container

Container bedienen sich einer gewissen Art der Abstraktion von verschiedenen Ressourcen des Host-Betriebssystems. Dabei werden virtuelle Ressourcen erstellt, mit denen das Gast-System arbeitet. Diese virtuellen Ressourcen sind vom Host-System abgekoppelt und bieten dadurch eine Trennung beider Systeme. Die Ressourcen, die virtualisiert werden sind abhängig von der Art der Container. Im folgenden werden zwei verschiedene Container-Arten vorgestellt.

1.1.1 Systemcontainer (kurz SC)

Systemcontainer oder auch virtuelle Maschinen sind eine Art von Containern, die das komplette System virtualisieren und nachbilden. In einer virtuellen Maschine werden alle Hardware-Komponenten nachgestellt und ein eigenes Betriebssystem innerhalb der virtuellen Maschine muss aufgesetzt werden.

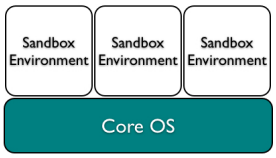


Hierbei hat jedes Gast-Betriebssystem (Guest OS) eine komplett eigene Hardware-Struktur, welche von der Virtualisierungssoftware zur Verfügung gestellt wird. Der Vorteil einer solchen Architektur ist, dass Software welche im Gast-System (im virtualisierten System) ausgeführt wird, sich wie in einer Sandbox verhält. Die virtuelle Maschine stellt der Software Ressourcen zur Verfügung, die wie gewohnt genutzt werden können - mit dem Unterschied, dass alle installierten Komponenten innerhalb der virtuellen Maschine

keinerlei Zugriff auf das Host-Betriebssytem besitzen.
Neben dem Sicherheitsaspekt ist ein weiterer Vorteil die Portabilität - denn virtuelle System sind auf dem Host-Betriebssytem einfache Dateien, welche auf andere Datenträger kopiert / verschoben werden können und auf anderen Rechnern im exakt identischen Zustand wieder gestartet werden können.

1.1.2 Anwendungscontainer (kurz APPC)

Ein Nachteil von Systemcontainern ist die Performance, da eine virtuelle Maschine immer eine komplette Rechnerarchitektur nachbildet und ein voll installiertes Betriebssystem beinhaltet. Um diesen Nachteil entgegenzuwirken werden Anwendungscontainer verwendet, die nicht das komplette System visualisieren, sondern gewisse Komponenten, die an das Host-System gekoppelt sind.



Containers

Der Unterschied ...

1.2 Images

1.2.1 App Container Image

1.2.2 App Container Pod

1.2.3 Signed Images

Image archives SHOULD be signed using PGP, the format MUST be ascii-armored detached signature mode.
Image signatures MUST be named with the suffix .and

1.3 Standardisierung

gibt es keine. Versuch https://github.com/appc/spec

1.4 Virtual Machines

Sind zu langsam (Performance-Probleme) Virtuelle Maschinen enthalten immer das komplette OS (Overhead)

1.4.1 Unterschiede bzw. Vergleichbarkeit

1.5 OS-Spezifikationen

Die meisten Container beruhen auf LXC (Linux). Es gibt auch schon Windows-Container (wie funktionieren diese?)

2 Containersoftware

https://github.com/appc/spec

2.1 LXC

Linux-Grundlage für Container-Technologie, alles andere beruht darauf.

2.2 LXD (SC)

2.3 Docker (APPC)

http://codefest.at/post/2014/11/25/Erste-Schritte-mit-Docker-Teil-1.aspx

2.4 Rocket (APPC)

2.5 Jetpack (APPC)

2.6 Libappc (APPC)

2.7 Kurma (APPC)

http://www.apcera.com/blog/apcera-open-sources-new-kurma-project/

3 Orchestration

3.1 Boot2Docker

3.2 VMWare

3.3 OpenStack

4 Performance

5 Security