Linux-Container-Terminologie

Was ist ein "Docker"?

Jörg Kastning

Universität Bielefeld / BITS

16. Juni 2021

Container

Container sind neu.
Container sind in sich geschlossene Entitäten.
Container ersetzen Virtualisierung.
Container sind universell portabel.
Container sind prinzipiell sicher.
Container heißen Docker.
Sechs der größten Fehlannahmen im Hinblick auf das Thema Container und Docker!
Zitat von: Oliver Liebel, Autor und Linux-Enterprise-Experte

Warum sind wir hier?

- Linux-Container haben seit ca. 2015 in vielen Branchen Fuß gefasst.
- Sie haben sich in etlichen Bereichen etabliert und kommen langsam aber sicher auch im BITS an.
- Deshalb sollten wir mal darüber reden!

Was ist das Ziel?

- Noch sprechen wir wie Blinde über Farbe.
- Nach diesem Vortrag gibt es (hoffentlich) ein paar Einäugige unter den Blinden.

Worüber sprechen wir heute nicht?

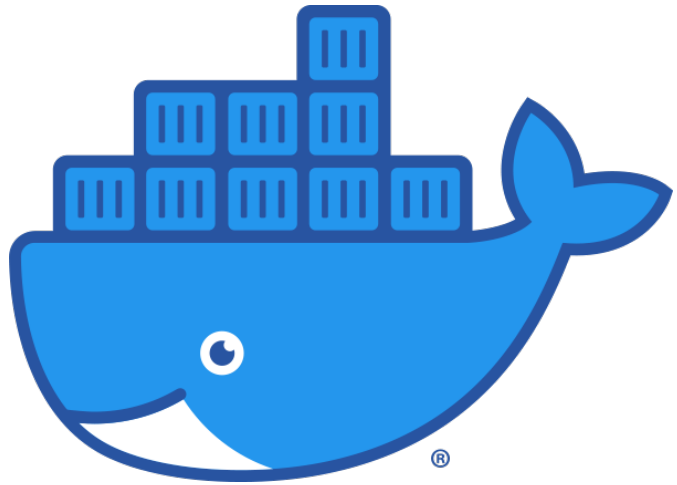
- Ist der rote Container gut? Oder nehmen wir ihn lieber in grün oder blau?
- Wie können wir den Einzug der Container verhindern?

Buzzwords


Runtime
railkar
LXD
Rkt
Tanzu
Swarm
Registry
CRI-O
Komposer
LXC
Image
compose
crun
Compose
Docker
json
Engine
OpenShift
katacontainer
runc
Kubernetes
Repository
Container
kubectl
k8s
Rancher
OCI
Podman
yaml

Docker ist...

- ... eine Firma.
- ... ein Container Image Format.
- ... eine Container Engine.
- ... ein Kommandozeilen-Werkzeug.
- ... ein Repository.
- ... ein Daemon.
- ... eine Registry.
- ... ein Alptraum.



Open Container Initiative

> *Die Open Container Initiative (OCI) ist ein Projekt der Linux Foundation, um offene Standards für die Containervirtualisierung auf Betriebssystemebene zu entwerfen. Derzeit sind zwei Spezifikationen in der Entwicklung und im Einsatz: Laufzeitspezifikation (runtime-spec) und Bildspezifikation (image-spec).*
>
> *OCI entwickelt runC, eine Container-Laufzeit, die die OCI-Spezifikation implementiert und als Basis für andere übergeordnete Werkzeuge dient. (Quelle: Wikipedia.de)*

Es erscheint sinnvoll, bei der Auswahl von Technologien und Werkzeugen aus diesem Bereich auf eine Kompatibilität zu den Standards der OCI zu achten, um nicht in einer Sackgasse zu landen.

Basis-Vokabular

Definition

Ein Container Host ist ein System, welches Container ausführt.

Example

Dies kann z. B. ein Gastbetriebssystem in einer VM sein, welches über eine Container Engine verfügt und Container ausführen kann.

Basis-Vokabular

Definition

Die Container Engine ist ein Stück Software, welches Eingaben über CLI oder API entgegen nimmt und verarbeitet. Aus Perspektive der Nutzer*innen führt die Container Engine die Container aus. Dazu bedient sich die Container Engine einer sogenannten Container Runtime.

Example

Beispiele für Container Engines sind Docker, RKT, CRI-O und LXD. Diese basieren auf der OCI Referenzimplementierung runc.

Basis-Vokabular

Definition

Eine Container Runtime ist eine Low-Level-Komponente der Container Engine. Die OCI Runtime Standard Referenz Implementierung ist runc. Die Runtime ist verantwortlich für:

- Verarbeitung von Container Images und Meta-Daten aus der Container Engine
- Kommunikation mit dem Kernel zur Instanziierung von Container-Prozessen
- Konfguration von cgroups, SELinux Policies, AppArmor Rules, etc.

Example

OCI kompatible Runtimes sind runc, crun, railkar und katacontainers.

Basis-Vokabular

Definition

Als Container Image wird der lokale Mountpoint bezeichnet, welcher zur Instanziierung von Containern genutzt werden kann. Diese Images können aus einem oder mehreren Layern bestehen und werden von sogenannten Registry-Servern bezogen.

Hierbei handelt es sich um einen der am häufigsten überladenen Begriffe im Container-Universum. Missverständnisse sind vorprogrammiert.

Basis-Vokabular

Definition

Das Container Image Format definiert den Aufbau eines Container Images, bestehend aus Layern und Metadaten.

Example

Das OCI Container Image Format definiert, dass sich Images aus TAR-Archiven für jeden Layer und einem JSON-Manifest mit Metadaten zusammensetzen.

Basis-Vokabular

Definition

Kernel Namespaces sind Datenstrukturen, welche Containern Ressourcen wie z. B. Einhängepunkte, Netzwerk-Schnittstellen, UIDs, GIDs, etc. bereitstellen.

Ohne Kernel Namespaces gebe es keine Linux-Container, wie wir sie heute kennen.

Basis-Vokabular

Definition

Ein Container ist die Laufzeit-Instanz eines Container Images. Es handelt sich dabei um einen Prozess, wecher in einem Kernel Namespace läuft.

Example

Unter Linux ist ein Container ein Standard-Prozess, welcher durch den Systemaufruf clone() erzeugt wird. Häufig werden Container durch Mechanismen wie cgroups, SELinux oder AppArmor isoliert.

Basis-Vokabular

Definition

Container Orchestration kümmert sich um die dynamische Verteilung und Ausführung von Containern in einem Cluster. Darüber hinaus stellt sie eine standardisierte Form zur Definition von Anwendungen bereit.

Example

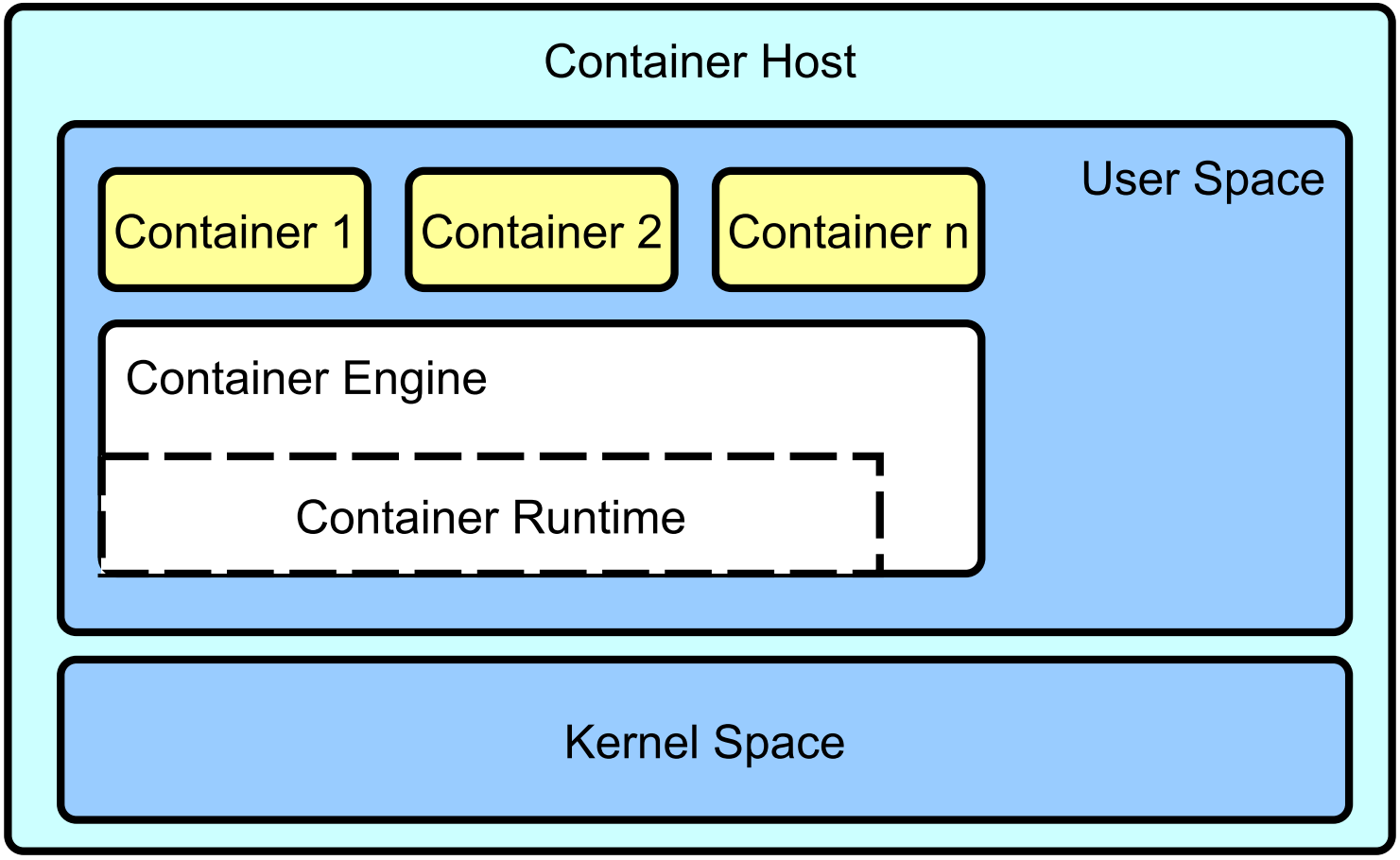
Anwendungen werden bspw. als *kube yaml*, *docker compose*, etc. definiert. Die unterschiedlichen Definitionen sind meist nicht kompatibel zueinander. Konkrete Produkte sind z. B. VMware Tanzu, Red Hat OpenShift, SUSE Rancher, Vanilla Kubernetes oder Docker Swarm (deprecated).

Zu Container Orchestration

Beim Betrieb von Container-Clustern waren zwei Probleme zu lösen. Das der Infrastruktur und das von Vanilla Kubernetes. Mit der Infrastruktur wären wir fertig geworden.

Frei nach Wernher von Braun

Einordnung der Vokabeln



Erweitertes-Vokabular

Hausaufgabe: Lernen und üben des erweiterten Vokabulars. Dieses

findet ihr unter:
https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction

Abschluss und Ausblick

- Lesestoff zum Thema bieten Kanboard im Container und die Artikel in meiner Linksammlung.
- Ebenfalls empfehlenswert:
  - Kubernetes Failure Stories.
  - Detecting Agile BS from US-Department of Defense.
- Folgethemen: Container-Architektur und Use Cases.
- Ich freue mich, wenn ihr wieder dabei seid.