

Container Linux

Jannik Moritz und Lukas Schneider



LEHRSTUHL FÜR VERTEILTE SYSTEME
UND BETRIEBSSYSTEME



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Was ist Container Linux?

- Leichtgewichtige Distribution für Cluster
- Fokus auf Automation, Sicherheit und Skalierbarkeit
- Container stehen im Mittelpunkt



Weiteres zu Container Linux

- Basiert auf Gentoo
- Entwicklung durch CoreOS, Inc.
- Erster release im Oktober 2013
- Orientierung an ChromeOS für schnelle Updates
- Wird von vielen Cloud Anbietern bereitgestellt



Agenda

1. Container
2. Cluster
3. Updates
4. Entwicklung
5. Die Zukunft von Container Linux



Agenda

1. Container
2. Cluster
3. Updates
4. Entwicklung
5. Die Zukunft von Container Linux



Was sind Container?

■ Ein oder mehrere Prozesse, die vom System isoliert sind

- Sicherheit durch die Isolierung und Virtualisierung
- Verwendung des OS Kernels
- Kleiner und schneller als VMs

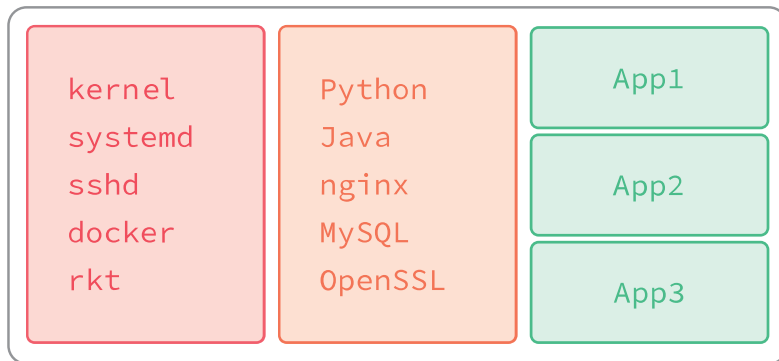
■ Alle notwendigen Abhängigkeiten separat im Container

- Konsistent über verschiedene Systeme
- Dadurch Konzentration auf App Entwicklung möglich
- Ermöglicht Portierbarkeit und Skalierbarkeit

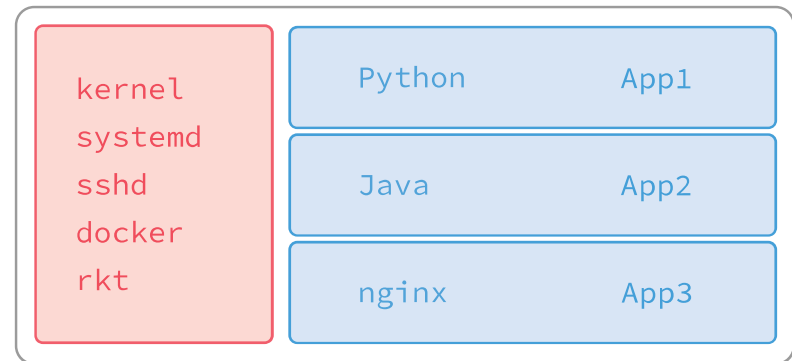


Was macht Container Linux?

Traditional Distro



CoreOS Your Containers



required software only



Wie verwendet Container Linux diese?

■ Docker

- Eine der führenden Umsetzungen für Container
- Zum entwickeln, teilen und ausführen von Containern

■ rkt

- Eigene Umsetzung von CoreOS
- Kann auch „Docker Images“ ausführen
- Implementiert eigenen Standard „App Container“
- Zielt in der Umsetzung auf bessere Sicherheit als Docker ab
- Kein zentraler Daemon



Agenda

1. Container
2. Cluster
3. Updates
4. Entwicklung
5. Die Zukunft von Container Linux



Was sind Cluster?

- Motivation: Bei großen Anwendungen ist ein Computer nicht mehr ausreichend
- Große Anwendungen laufen auf vielen untereinander vernetzten Computern
 - Erhöhung der Rechenleistung
 - Erhöhung der Ausfallsicherheit



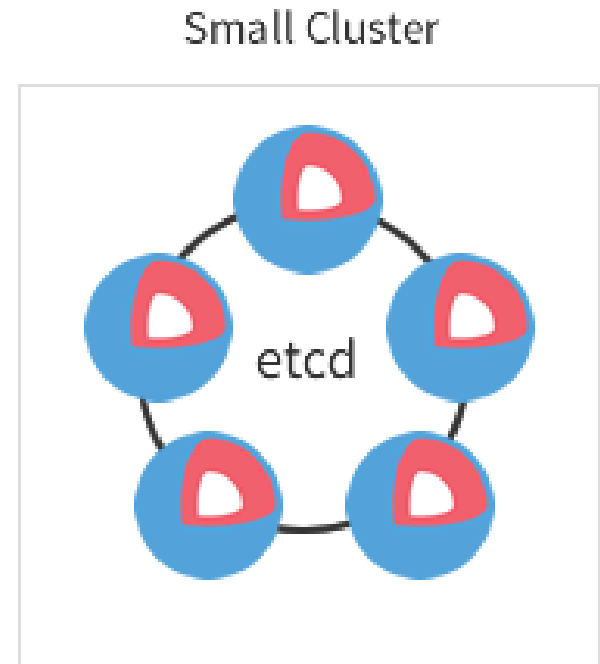
Cluster bei Container Linux

- Container Linux ist für Cluster entwickelt worden
- Etcd als wichtigstes Hilfsmittel
 - „Key-value store“
 - (Zentrale) Instanz für die Konfiguration eines Clusters
- Durch Anwendungen wie Kubernetes kann man einen Container Linux Cluster leichter verwalten



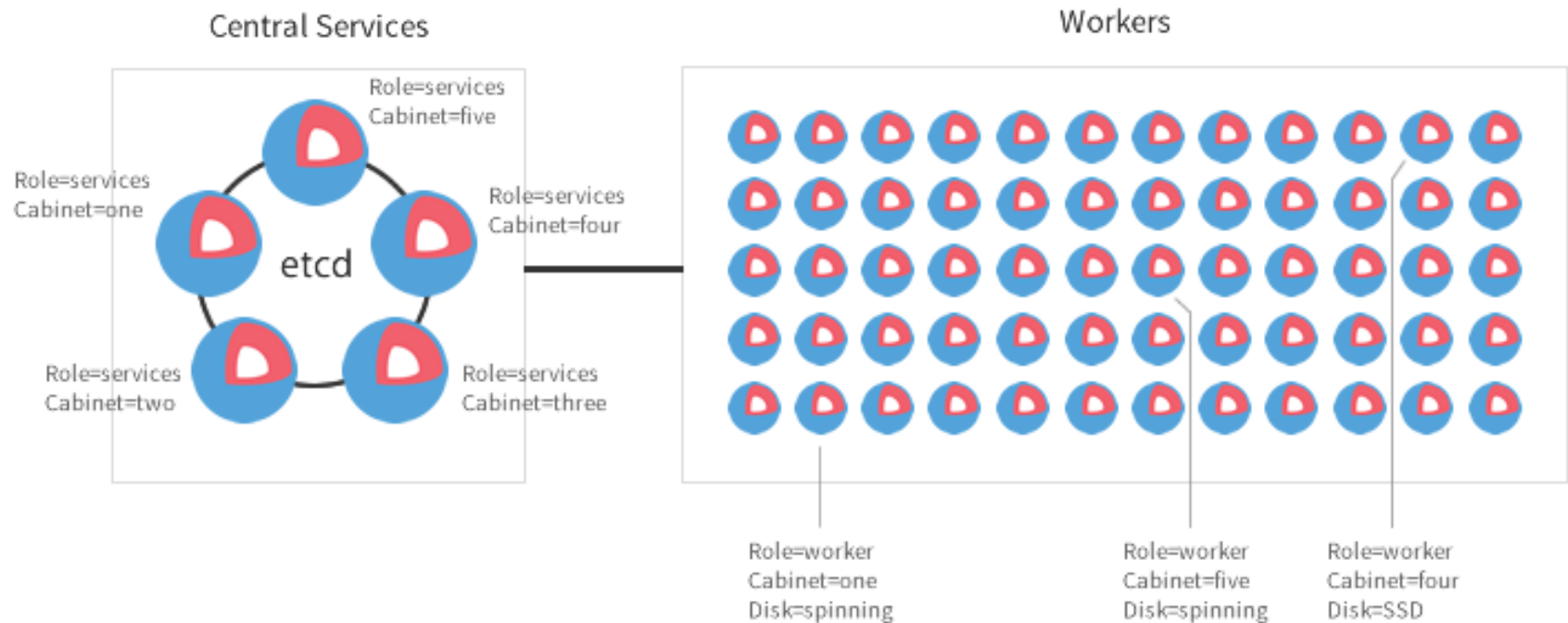
Kleiner Cluster

- Für kleinere Anwendungen geeignet
- Etcd läuft auf jeder Maschine



Produktionscluster mit zentralen Diensten

- Für große Anwendungen
- Extra Maschinen, auf denen zentrale Dienste laufen
- Auf den Arbeiter Maschinen läuft die eigentliche Anwendung



Agenda

1. Container
2. Cluster
3. Updates
4. Entwicklung
5. Die Zukunft von Container Linux



■ Orientierung an ChromeOS

■ Zwei Partitionen USR-A und USR-B für OS

- Tausch der Partitionen von aktiv zu passiv nach Update
- Dadurch schnelle Updates möglich

■ Verschieden Update Strategien

- Updatezeitpunkt festlegbar
- Strategien für Cluster für geringe/keine Downtime

■ Rollback fehlerhafter Updates

- Automatisch wenn Boot fehlschlägt
- Manuell auch möglich direkt nach dem Update



Agenda

1. Container
2. Cluster
3. Updates
- 4. Entwicklung**
5. Die Zukunft von Container Linux



- **Der Linux Kernel ist fast unverändert**
 - Aktuell nur drei kleine Patches
- **Das Container Linux Projekt besteht aus vielen Git Repositories**
 - Alle Open Source und auf GitHub verfügbar
- **CoreOS stellt ein SDK zur Verfügung, um an Container Linux mit zu arbeiten**



■ Mittels Pull Request auf GitHub

- Automatisierter Test muss erfolgreich sein
- Ein CoreOS Mitarbeiter muss den Pull Request Überprüfen
- War beides erfolgreich, kann ein CoreOS Mitarbeiter den Pull Request mergen

■ Vorgaben für Patches befinden sich in dem jeweiligen Git Repository

- Stil der Commit Message (ähnlich wie in Linux)
- Code Stil
- Können von Repository zu Repository leicht unterschiedlich sein



Agenda

1. Container
2. Cluster
3. Updates
4. Entwicklung
5. Die Zukunft von Container Linux



Die Zukunft von Container Linux

- Core OS Inc. wurde 2018 von Red Hat gekauft
- Die Entwicklung von Container Linux wird zurück gefahren
 - Soll aber weiterhin unterstützt werden
- Der Nachfolger Red Hat CoreOS ist zur Zeit in der Entwicklung
 - Basierend auf Fedora



■ Container

- Trennung des OS von den Applikationen
- Sicher, schnell, skalierbar und portierbar

■ Cluster

- Container Linux ist für große Cluster entworfen
- Etcid und andere Verwaltungsprozesse können ausgelagert werden

■ Updates

- Schnell durch zwei Partitionen für OS
- Sicher durch einfaches Rollback

■ Entwicklung

- Fast unveränderter Linux Kernel
- Einreichen von Patches mittels Pull Requests
- Vorgaben befinden sich im jeweiligen Git Repository



Quellen

<https://coreos.com/why/>

<https://coreos.com/os/docs/latest/>

<https://coreos.com/os/docs/latest/sdk-disk-partitions.html>

<http://www.chromium.org/chromium-os/chromiumos-design-docs/disk-format>

<https://www.redhat.com/de/topics/containers/whats-a-linux-container>

<https://www.redhat.com/de/topics/containers>

<https://www.docker.com/why-docker>

<https://coreos.com/rkt/>

<https://coreos.com/blog/rocket.html>

<https://coreos.com/rkt/docs/latest/>

https://wiki.gentoo.org/wiki/Distributions_based_on_Gentoo

<https://github.com/coreos/manifest/releases/>

https://groups.google.com/forum/?utm_medium=email&utm_source=footer#!forum/coreos-user

<https://de.wikipedia.org/wiki/Rechnerverbund>

<https://coreos.com/blog/coreos-tech-to-combine-with-red-hat-openshift>

<https://kubernetes.io/docs/concepts>

<https://kubernetes.io/docs/concepts/overview/components/>

<https://github.com/etcd-io/etcd>

<https://etcd.io/>

<https://github.com/coreos/>



Partition Table:

Nr.	LABEL
1	EFI-SYSTEM
2	BIOS-BOOT
3	USR-A
4	USR-B
5	ROOT-C
6	OEM
7	OEM-CONFIG
8	(unused)
9	ROOT



Einfacher Entwicklungs- und Testcluster

- Für die Entwicklung von Containern
- Etcd läuft auf einer extra Maschine

