# Linux Container (LXC)

PC-Treff-BB
Roland Egeler

# Linux Container

Thema ist Virtualisierung

- LXC ist eine leichtgewichtige Virtualisierungslösung unter Linux
- Abgrenzung zur Hardwarevirtualisierung nach oben
- Abgrenzung zu Docker nach unten
- Reale Anwendung

# Erklärung

Was ist Virtualisierung?

- Virtualisierung ist die Abstraktion von Soft- oder Hardwarekomponenten
- Das Ergebnis ist eine Softwareschnittstelle
- Diese Schnittstelle kann von einer oder mehreren virtuellen Maschinen benutzt werden
- Dabei können die darunterliegenden Ressourcen pro virtuelle Maschine eingeschränkt werden

# Ressourcenverteilung

Was stellt die Schnittstelle der virtuellen Maschine zur Verfügung?

- Prozessorkerne
- Hauptspeicher
- Dateisystem (Plattenplatz)
- Optische Datenträger
- Images lassen sich als virtuelles Laufwerk einbinden
- Netzwerk (eigene IP-Adresse)
- Direkter Zugriff auf physikalische Hardware
  - USB, PCIe, ...

# Motivation

Warum Virtualisierung?

- Bessere Ausnutzung der Ressourcen
- Es wird weniger Hardware benötigt
- Test von Software ohne Auswirkung auf Produktivumgebung
- Virtuelle Maschinen lassen sich leichter migrieren
- Ausfallsicherheit bei Clustern
- Templates / Klone / Sicherung/Rückspielen

# Virtualisierung

Stufen der Virtualisierung bei Computern

- Kompletter Rechner (andere Architektur)
  - Android unter Windows
  - MAME (Multiple Arcade Machine Emulator)
- Gleiche Architektur (z.B. x86/x64)
  - Windows unter Linux
- Gleicher Kernel (z.B. Linux)
  - Container
- Applikationsvirtualisierung (Web Services)
  - z.B. Docker

# Virtualisierung

Beispiele von Virtualisierungssoftware

- ESXi (Hypervisor von VMware)
- Hyper-V (Hypervisor von Microsoft)
- Xen (Hypervisor, zeitweise Citrix, jetzt Linux Foundation)
- KVM (Linux, nutzt QEMU)
- VMware Workstation (Player, Pro)
- Virtual Box (Oracle)
- Container (LXC, jails, zones, Docker...)
- Java VM...

# Container

## Historie

- Chroot (Change root 1982 in BSD)
- BSD jails (2000)
- Solaris (Zones und Containers, 2005)
- Linux Container (LXC, 2008)
- Docker (2013)

# Warum LXC?

Abgrenzung von anderen Virtualisierungstechniken

- Vollvirtualisierung bindet mehr Ressourcen
  - Eigenes Betriebssystem
  - Feste Speichergröße
  - Festes Plattenimage
- Es läuft nur ein Kernel (im Gastgebersystem)
- Container ist eine abgeschottete Gruppe von Prozessen
- Virtualisierung von kompletten Servern im Gegensatz zur Applikationsvirtualisierung bei docker

# Warum LXC?

Abgrenzung von anderen Virtualisierungstechniken

- Keine Installation, sondern Templates (TurnKey Linux...)
  - LAMP
  - NextCloud
  - Wordpress...
- Templates können selbst erstellt werden
- Durchgriff auf physikalische Hardware ist möglich
  - Grafikkarte
  - Satellitenempfänger
  - USB...

# Warum LXC?

## Weitere Vorteile

- Braucht extrem wenig Ressourcen
  - Kein fest zugeordneter Hauptspeicher (Ballooning)
  - Kein fest zugeordneter Plattenplatz
- Braucht keine Kernelmodule
  - Problemloses Kernelupdate
  - Kein Kompilieren von Modulen

# Warum LXC?

## Weitere Vorteile

- Webmin läuft innerhalb von LXC
  - Macht Administration wesentlich leichter
- Docker läuft innerhalb von LXC
  - Alle Dockerinstanzen können dort gebündelt werden
  - Sicherer gegen „Ausbrüche“

# Worauf basiert LXC?

Benutzte Techniken

- chroot: Nur Teilbaum des Dateibaums sichtbar
- namespaces: Isolation des Containers
- cgroups: Ressourcenverwaltung
- Absicherung des Host gegen Container
- Unprivilegierte Container

# Benutzte Techniken

chroot

- Bedeutung: „Change Root“; setzt neue Wurzel des Dateibaums
- Darüberliegende Dateien sind nicht sichtbar
- Benutzt zur Isolation von Dateizugriff von Prozessen
- Bei Einbruch kein Zugriff auf darüberliegendes System
- Kann durchbrochen werden, wenn innerhalb „root“-Rechte erlangt werden
- Daher wurde „pivot_root“ entwickelt und benutzt

# Benutzte Techniken

## Namespaces

- Bedeutung: Namensräume
- Weitere Isolation von Prozessen
- Verschiedene Eigenschaften des Systems werden vor den im Namensraum laufenden Prozessen versteckt
- Beispiele:
  - Prozesse / Rechenzeit
  - Eingehängte Dateisysteme
  - Netzwerkschnittstellen
  - Benutzer

# Benutzte Techniken

Namespaces

- Beispiele:
  - mnt (Mount Namespace)
  - uts (Unix Time Sharing)
  - ipc (Inter Process Communication)
  - net (Network Namespace)
  - pid (Process Namespace)
  - user (User Namespace)

# Benutzte Techniken

cgroups

- Bedeutung: Controlling Groups („Kontrollgruppen“)
- Einschränkung der benutzten Ressourcen
  - Prozessorkerne
  - Speicher
  - Geräte
  - Einfrieren von Prozessen
  - Priorisierung

# Benutzte Techniken

Absicherung des Host gegen Container

- Das Gastsystem könnte kompromittiert werden
- Dadurch wäre auch der Host gefährdet
- Weitere Absicherung durch Sicherheitssysteme
- Beispiele
  - Seccomp
  - AppArmor
  - SELinux
  - Capabilities
- Komplex und fehleranfällig

# Benutzte Techniken

## Unprivilegierte Container

- Der Container läuft auf dem Host nicht als root
- Im Container existiert trotzdem ein (anderer) root
- Bei „Ausbruch“ eines Prozesses aus dem Container läuft dieser nicht als root
- Kann keinen Schaden im Host anrichten
- Könnte allerdings Container beschädigen

# Abgrenzung zu docker

## Ähnlichkeiten

- Ist ebenfalls ein Container
- Benutzt dieselben Techniken
  - cgroups
  - namespaces
  - …
- Noch leichtgewichtiger
- Erzeugung über Templates (Docker Hub)

# Abgrenzung zu docker

Anderer Ansatz

- Stellt im Normalfall genau einen Service zur Verfügung
  - html, mysql, ldap…
- Hält keine eigenen Daten (gedächtnislos)
- Datenhaltung auf dem Host oder einem „Datendocker“
- Konfiguration über Dateien
- Beim Start werden Templates heruntergeladen
- Administration über Kommandozeile
- „docker-daemon“ läuft als root

# Abgrenzung zu docker

Anderer Ansatz

- Aufspaltung von monolithischen Anwendungen in „microservices“
- Administration großer Mengen docker-Container (Cloud)
  - kubernetes
  - ranger...
- Alternative „podman“ (auch von Red Hat)
  - Kommandozeilenkompatibel
  - rootless

# Konkretes Projekt

Real existierende Hardware

- „Blechkasten“ (aufgebaut ca. 2009)
- AMD AthlonX2 5050e, 2 x 2,6 GHz, 45W
- 4 GiB DDR2 RAM
- 2 x 1 Terabyte Samsung SATA-Platten, gespiegelt (ZFS)
- Ein optisches Laufwerk für die Installation
- Gigabit Ethernet
- Eine DVB-S2 Satellitenempfängerkarte (PCI)
  - Geplant: Doppelempfängerkarte mit PCI Express

# Konkretes Projekt

Real existierende Software

- Hauptbetriebssystem ist „Proxmox VE“
- Ein Container für Webserver (Reverse Proxy)
- Ein Container für NextCloud
- Ein Container für Fileserver (samba)
- Ein Container für VDR (SAT-Karte durchgereicht)
- Geplant:
  - Migration auf Server mit Xeon-Vierkern (HT) und 16 GiB RAM
  - Ein weiterer Container mit Mailserver
  - Backup via Bacula

# Zusammenfassung

Linux Container

- Leichtgewichtige Virtualisierung unter Linux
- Open Source
- Genau ein Kernel
- Keine zusätzlichen Kernelmodule
- Flexible Ressourcenverteilung
- Durchgriff auf Hardware

# Quellen

- https://de.wikipedia.org/wiki/LXC
- https://linuxcontainers.org/
- https://wiki.ubuntuusers.de/LXC/
- https://de.wikipedia.org/wiki/Virtualisierung_(Informatik)/
- https://de.wikipedia.org/wiki/Docker_(Software)/
- https://www.heise.de/developer/artikel/Podman-Linux-Container-einfach-gemacht-Teil-1-4329067.html

# Vielen Dank!