Linux Container (LXC) – Ein Überblick

Bernhard Wesely
Trivadis Delphi GmbH
Wien, Österreich

Schlüsselworte
Oracle Linux Container LXC Virtualisierung

Einleitung
Virtualisierungslösungen gehören schon seit langem zum IT-Standardrepertoire. Da ist es nicht verwunderlich, dass die letzten Jahre viele unterschiedliche Technologien hervorgebracht haben. Die Linux Container reihen sich hier nahtlos ein. Als besonders schlanke und einfache Lösung zur Kapselung von Linux Systemen und Applikationen versuchen sie dort Boden zu gewinnen wo keine „fremden" Betriebssysteme wie Windows oder Solaris benötigt werden.

Allgemeines
Viele verschiedene Virtualisierungslösungen existieren auf dem Markt, jede mit ihren Vor- und Nachteilen.
Der weitaus größte Anteil fällt den Hypervisor Lösungen zu. Vertreter dieser Technik sind beispielsweise *VMWare, PowerVM, LDOMs* oder *Xen*. Diese Technologien haben den Vorteil, dass sie generell Betriebssystem-unabhängig sein können. So können auf einem VMWare Server unterschiedliche Betriebssysteme gleichzeitig laufen. Dies ist jedoch ein recht aufwändiger und komplexer Prozess.
Bei der Kernel-Level Virtualisierung auf der anderen Seite stellt der laufende Kernel Funktionen bereit um laufende Prozesse von einander zu trennen. Dies bedeutet nur einen geringen Overhead, jedoch ist es somit nicht möglich andere Betriebssysteme als das des laufenden Kernels zu virtualisieren. Zu diesen Lösungen gehören *Solaris Zonen*, *KVM* oder eben *LXC* (Linux Container)

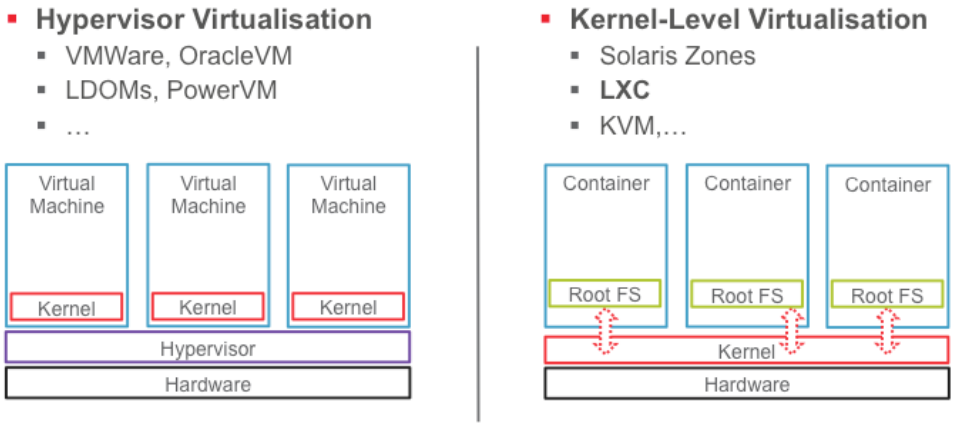

Abb. 1: Virtualisierungstechnologien

Installation
Die Installation von LXC gestaltet sich recht einfach. Viele Distributionen stellen in ihren Paketmanagern bereits Pakete zur Verfügung. Somit ist es dann meist nur mehr ein Befehl der abgesetzt werden muss

`apt-get install lxc`	unter Debian/Ubuntu oder
`yum install lxc`	unter Oracle Linux/Red Hat/Fedora/...

Um LXC verwenden zu können müssen einige Voraussetzungen erfüllt sein. Ob dem so ist, kann leicht mittels des Kommandos `lxc-checkconfig` überprüft werden. Im Idealfall sieht die Ausgabe dann so aus:

```
root@lxc01:~# lxc-checkconfig
Kernel configuration not found
  at /proc/config.gz; searching...
Kernel configuration found
  at /boot/config-3.13.0-35-generic
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled
--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: enabled
Cgroup cpuset: enabled
--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
```

Sollten einige Voraussetzungen nicht erfüllt sein, so liegt dies oft an einem alten Kernel und es sollte ein neuer Kernel eingespielt werden.

Was ist LXC?
LXC stellt im eigentlichen Sinn keine Virtuellen Maschinen zur Verfügung. Mehr werden laufenden Betriebssystemprozesse so von einander getrennt, dass der Eindruck entsteht, dass es sich um einen eigenen Server handelt. Für den Endanwender ist dieser Unterschied jedoch unerheblich.
Diese Ressourcentrennung basiert auf einem Linux Kernel-Feature namens „Namespaces“. Namespaces erlauben es, bestimmte Namen (Identifier) in einem Linux System mehrmals zu verwenden. So ist es ohne Namespaces beispielsweise nicht möglich eine PID (Prozess-ID) mehrmals zu verwenden. Prozesse können ihren Namespace nicht verlassen, so können auch keine Kollisionen entstehen sollten zwei Prozesse in unterschiedlichen Namespaces die gleiche PID erhalten haben.

Nach und nach wurden im Linux Kernel kritische Komponenten mit Namespace Unterstützung ausgestattet, hierzu zählen:

- **pid** - Prozesse
- **net** - Netzwerk-Interfaces,…
- **mnt** - Mount points, Filesysteme
- **uts** - Hostname
- **user** - UIDs
- **ipc** - System V IPC

Jeder startende Linux Container macht also nichts anderes als einen neuen Namespace anzulegen und in diesem dann init aufzurufen. Da Prozesse immer den Namespace ihres Mutterprozesses erben und init für das Starten aller anderen Linux Prozesse eines System verantwortlich ist, entsteht so ein „Parallelsystem“, der Linux Container.

Natürlich wird auch ein (Root-)Filesystem benötigt. Dieses ist einfach ein Verzeichnis auf dem Host, welches chroot-artig in den Container gemounted wird. So kann jeder Container seine eigenen Dateien verwalten.

Grundlagen
Zu den Grundlagen von LXC gehört das Anlegen, Starten, Stoppen und Löschen von Containern

Erzeugen eines neuen Containers
Ein neuer Container wird mittels des Kommandos `lxc-create` angelegt. Es werden mindestens zwei Argumente erwartet, `-n` für den Namen des Containers (und damit auch Namespaces) und `-t` für den Namen des anzuwendenden Templates. Templates sind nichts anderes als Shellscripts welche eine Neu-Installation einer bestimmten Distribution beschreiben. So würde das oracle Template beispielsweise ein Oracle Linux in den Container installieren.
Templates unterstützen in den meisten Fällen eigene Distributions-spezifische Argumente. Diese werden mittels „`-- -<Argument>`“ übergeben.

Folgende Templates stehen standardmässig zur Verfügung:

```
lxc-alpine                          lxc-gentoo
lxc-altlinux                        lxc-openmandriva
lxc-archlinux                       lxc-opensuse
lxc-busybox                         lxc-oracle
lxc-centos                          lxc-plamo
lxc-cirros                          lxc-sshd
lxc-debian                          lxc-ubuntu
lxc-download                        lxc-ubuntu-cloud
lxc-fedora
```

Hier ein Beispiel eines erfolgreich erzeugten Oracle Linux Containers. Als Template-spezifischer Parameter wird die gewünschte Release Nummer übergeben.

```
root@lxc03:~# lxc-create -n cn-ol5u10 -t oracle -- -R 5.10
Host is Ubuntu 14.04
Create configuration file /var/lib/lxc/cn-ol5u10/config
Downloading release 5.10 for x86_64
...
Setting up Install Process
...
Downloading Packages:
(1/121): MAKEDEV-3.23-1.2.x86_64.rpm                    | 135 kB     00:00
(121/121): zlib-1.2.3-7.el5.x86_64.rpm                  |  52 kB     00:00
--------------------------------------------------------------------------
Total                                          2.0 MB/s |  82 MB     00:41
...
Running Transaction
  Installing : libgcc-4.1.2-54.el5.x86_64
1/121
  Installing : rootfiles-8.1-1.1.1.noarch
121/121
...
Complete!
Fixing (downgrading) rpm database from version 9
Rebuilding rpm database
Patching container rootfs /var/lib/lxc/cn-ol5u10/rootfs for Oracle Linux
5.10
Configuring container for Oracle Linux 5.10
Added container user:oracle password:oracle
Added container user:root password:root
Container : /var/lib/lxc/cn-ol5u10/rootfs
Config    : /var/lib/lxc/cn-ol5u10/config
Network   : eth0 (veth) on virbr0
```

Starten eines Containers

Das Kommando `lxc-start -n <Container Name>` startet einen zuvor erstellten Container. Gestartet wird der Container im Vordergrund, um die Terminal Session wiederzuerlangen muss der Container herunterfahren werden. Um dies zu umgehen kann der Container im Hintergrund gestartet werden. Dies passiert mittels des Parameters `-d`.

Stoppen eines Containers

Laufende Container werden mittels `lxc-stop -n <Container Name>` gestoppt. Dies führt einen geordneten Shutdown im Gast-Betriebssystem durch. Sollte das Betriebssystem im Container nach 60sek noch nicht gestoppt sein, so wird der Container gekillt.

Löschen eines Containers

Container können mittels des Kommandos `lxc-destroy -n <Container Name>` gelöscht werden. Dies löscht sowohl das Config File als auch das rootfs Verzeichnis von der Festplatte. Laufende Container können nur mittels des Parameters `-f` (Force) gelöscht werden.

Zusammenfassung
Linux Container sind eine einfache, performante und robuste Methode um schnell Linux Systeme zu virtualisieren. Durch die enge Verzahnung mit CGroups ist auch ein Resourcen Management möglich, welches gerade in virtualisierten Umgebungen dringend benötigt wird. Eigene Anforderungen können einfach mit der LXC-API und diversen Scriptsprachen umgesetzt werden.

Kontaktadresse:
Bernhard Wesely
Trivadis Delphi GmbH
Handelskai 94-96, Millennium Tower
A-1200 Wien

E-Mail Bernhard.Wesely@trivadis.com
Internet: www.trivadis.com