

vserver.ping.de

Was ist LXC und warum machen wir damit VServer

LXC steht für Linux-Containers und ist eine Art virtueller Server für Linux. Dabei nutzen alle virtuellen Server-Instanzen den Kernel der Wirtsystems mit. Implementiert werden die Container in Linux über die Namespace Erweiterungen, die dafür sorgen, dass Prozesse in eigene Adressräume isoliert werden können. Da dies nicht nur Prozesse und Speicher, sondern auch Netzwerke umfasst, können hiermit vollständige virtuelle Server implementiert werden.

Bislang wurde bei PING die ältere Linux-VServer-Implementierung eingesetzt, die einen eigenen Kernel-Patch und Userspace-Tools implementiert. Bis zur Version Squeeze von Debian wurde dabei vom Debian-Kernel-Team ein entsprechend gepatchter Kernel zur Verfügung gestellt. Mit Release von Wheezy wurde dies allerdings zu Gunsten der weiter fortgeschrittenen Entwicklung der Linux-Containers entfernt.

Für Linux-Containers sind einige Optionen zu Namespaces zu aktivieren, was aber bei den Kernel-Images von Debian bereits der Fall ist. Es ist somit nur noch das Paket `lxc` zu installieren, welches die Userspace-Tools zu Erzeugung und Verwaltung der Container enthält.

Probleme

LXC ist noch in der Entwicklung, früher waren LXC-Container nicht ausbruchsicher, dies ist aber seit Debian Jessie nicht mehr der Fall siehe <https://wiki.debian.org/LXC>

Installation des Hosts

Der Anleitung im [Debian Wiki](#) folgend, muss zunächst das LXC-Paket installiert werden:

```
root@containers:~# apt-get install lxc
Reading package lists... Done
Building dependency tree
```

```
Reading state information... Done
The following additional packages will be installed:
  arch-test bridge-utils busybox-static cloud-image-utils debootstrap distro-info dns-root-data
Suggested packages:
  cloud-utils-euca ubuntu-archive-keyring squid-deb-proxy-client shunit2 wodim cdrkit-doc btrfs-
The following NEW packages will be installed:
  arch-test bridge-utils busybox-static cloud-image-utils debootstrap distro-info dns-root-data
0 upgraded, 23 newly installed, 0 to remove and 0 not upgraded.
Need to get 8030 kB of archives.
After this operation, 34.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Vorbereitung der unprivilegierten Root-Container

Da die als VServer gedachten Container von Root während des Startes des Systems erzeugt werden, diese aber unprivilegiert sein sollen, muss ein uid- und gid-Bereich dem Nutzer Root zugewiesen werden (vergl. <https://linuxcontainers.org/lxc/getting-started/#creating-unprivileged-containers-as-root>):

```
root@containers:/etc# git diff
diff --git a/lxc/default.conf b/lxc/default.conf
index d2d1d51..30231e3 100644
--- a/lxc/default.conf
+++ b/lxc/default.conf
@@ -1,3 +1,5 @@
  lxc.net.0.type = empty
  lxc.apparmor.profile = generated
  lxc.apparmor.allow_nesting = 1
+lxc.idmap = u 0 100000 65536
+lxc.idmap = g 0 100000 65536
diff --git a/subgid b/subgid
index dc7d014..44be50f 100644
--- a/subgid
+++ b/subgid
@@ -1,2 +1 @@
 -dh:100000:65536
+root:100000:65536
diff --git a/subuid b/subuid
```

```
index dc7d014...44be50f 100644
--- a/subuid
+++ b/subuid
@@ -1,2 +1 @@
-dh:100000:65536
+root:100000:65536
```

Netzwerk

Damit die LXC-Container auf das Netzwerk zugreifen können, müssen deren virtuellen Netzwerkkarten an das `lxcbr0` angebunden werden. Da `vserver.ping.de` selbst eine virtuelle Maschine ist und die IP-Netze außerhalb der VM ankommen, bzw. dorthin geroutet werden, muss für die Container einer Bridge eingerichtet werden, in der die an `vserver.ping.de` gerouteten Netze verwendet:

```
auto lxcbr0
iface lxcbr0 inet static
    bridge_ports none
    bridge_stp off
    bridge_fd 0
    address 176.9.167.1
    netmask 255.255.255.240

iface lxcbr0 inet6 static
    bridge_ports none
    bridge_stp off
    bridge_fd 0
    address 2a01:4f8:262:445e:300::1/72
```

Damit LXC für die erzeugten Container auch ein Interface im Host erzeugt und dies der Bridge hinzufügt, muss `/etc/lxc/default.conf` angepasst werden:

```
diff --git a/lxc/default.conf b/lxc/default.conf
index 30231e3..c6bd25e 100644
--- a/lxc/default.conf
+++ b/lxc/default.conf
@@ -1,4 +1,6 @@
-lxc.net.0.type = empty
```

```
+lxc.net.0.type = veth
+lxc.net.0.link = lxcbr0
+lxc.net.0.flags = up
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
lxc.idmap = u 0 100000 65536
lxc.idmap = g 0 100000 65536
lxc.cgroup.devices.allow = c 10:200 rwm
```

Container erstellen

Container speichern ihr Dateisystem unter `/var/lib/lxc/<name>/rootfs`. Aus diesem Grund sollte hier ein Dateisystem für den Container eingehängt werden (z.B. ein LV aus dem Dom0 durchgereicht).

Auf vserver.ping.de gibt es ein LVM-Volume `vserver` in dem für jeden VServer ein LV mit einem Dateisystem angelegt und eingehängt wird.

```
root@vserver:~# lvcreate -L 70G -n jmp100.prima.de vserver
Logical volume "jmp100.prima.de" created.

root@vserver:~# mkfs.ext4 /dev/vserver/jmp100.prima.de
mke2fs 1.47.2 (1-Jan-2025)
Discarding device blocks: done
Creating filesystem with 18350080 4k blocks and 4587520 inodes
Filesystem UUID: eb9c3ac5-baf0-4dba-92d0-d3ab68097894
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done

root@vserver:~# mkdir /var/lib/lxc/jmp100.prima.de

root@vserver:~# echo -e
```

```
"/dev/vserver/jmp100.prima.de\t/var/lib/lxc/jmp100.prima.de\ttext4\tdefaults\t0\t2" >>
/etc/fstab

root@vserver:~# systemctl daemon-reload

root@vserver:~# mount /var/lib/lxc/jmp100.prima.de
```

Das

Da die genutzten Container unprivilegiert sind, muss das `Download`-Template genutzt werden. Hier ist dann die `Distribution` (z.B.: `Debian`), das `Release` (z.B.: `buster`) und die `Architektur` (z.B.: `amd64`) angegeben werden.

```
root@vserver:~# lxc-create -n jmp100.prima.de -t download
Downloading the image index

---
DIST          RELEASE     ARCH    VARIANT  BUILD
---
almalinux     10          amd64   default  20260123_23:25
almalinux     10          arm64   default  20260123_23:29
almalinux     8           amd64   default  20260123_23:23
almalinux     8           arm64   default  20260123_23:27
almalinux     9           amd64   default  20260123_23:25
almalinux     9           arm64   default  20260123_23:28
alpine        3.20        amd64   default  20260124_13:00
alpine        3.20        arm64   default  20260124_13:01
alpine        3.20        armhf   default  20260124_13:00
alpine        3.20        riscv64 default  20260124_13:07
alpine        3.21        amd64   default  20260124_13:00
alpine        3.21        arm64   default  20260124_13:01
alpine        3.21        armhf   default  20260124_13:02
alpine        3.21        riscv64 default  20260124_13:02
alpine        3.22        amd64   default  20260124_13:00
alpine        3.22        arm64   default  20260124_13:02
alpine        3.22        armhf   default  20260124_13:01
alpine        3.22        riscv64 default  20260124_13:02
alpine        3.23        amd64   default  20260124_13:00
alpine        3.23        arm64   default  20260124_13:00
alpine        3.23        armhf   default  20260124_13:00
```

alpine	3.23	riscv64	default	20260124_13:03
alpine	edge	amd64	default	20260124_13:00
alpine	edge	arm64	default	20260124_13:00
alpine	edge	armhf	default	20260124_13:01
alpine	edge	riscv64	default	20260124_13:06
alt	Sisyphus	amd64	default	20260124_03:08
alt	Sisyphus	arm64	default	20260124_03:22
alt	p11	amd64	default	20260124_03:40
alt	p11	arm64	default	20260124_03:17
amazonlinux	2	amd64	default	20260124_05:09
amazonlinux	2	arm64	default	20260124_05:09
amazonlinux	2023	amd64	default	20260124_05:09
archlinux	current	amd64	default	20260124_04:18
archlinux	current	arm64	default	20260124_04:18
archlinux	current	riscv64	default	20260124_04:18
busybox	1.36.1	amd64	default	20260124_06:00
busybox	1.36.1	arm64	default	20260124_06:00
centos	10-Stream	amd64	default	20260124_07:08
centos	10-Stream	arm64	default	20260124_07:35
centos	9-Stream	amd64	default	20260124_07:35
centos	9-Stream	arm64	default	20260124_07:08
debian	bookworm	amd64	default	20260124_05:24
debian	bookworm	arm64	default	20260124_05:24
debian	bookworm	armhf	default	20260124_05:24
debian	bullseye	amd64	default	20260124_05:24
debian	bullseye	arm64	default	20260124_05:24
debian	bullseye	armhf	default	20260124_05:39
debian	forky	amd64	default	20260124_05:24
debian	forky	arm64	default	20260124_05:24
debian	forky	armhf	default	20260124_05:24
debian	forky	riscv64	default	20260124_05:40
debian	trixie	amd64	default	20260124_05:24
debian	trixie	arm64	default	20260124_05:24
debian	trixie	armhf	default	20260124_05:36
debian	trixie	riscv64	default	20260124_05:24
devuan	chimaera	amd64	default	20260124_11:50
devuan	chimaera	arm64	default	20260124_11:50
devuan	daedalus	amd64	default	20260124_11:50
devuan	daedalus	arm64	default	20260124_11:50
devuan	excalibur	amd64	default	20260124_11:50

devuan	excalibur	arm64	default	20260124_11:50
fedora	41	amd64	default	20260124_00:01
fedora	41	arm64	default	20260123_23:38
fedora	42	amd64	default	20260123_23:40
fedora	42	arm64	default	20260124_00:09
fedora	43	amd64	default	20260124_00:01
fedora	43	arm64	default	20260124_00:39
kali	current	amd64	default	20260124_17:14
kali	current	arm64	default	20260124_17:14
mint	ulyana	amd64	default	20260124_08:51
mint	ulyssa	amd64	default	20260124_08:51
mint	uma	amd64	default	20260124_08:51
mint	una	amd64	default	20260124_08:51
mint	vanessa	amd64	default	20260124_08:51
mint	vera	amd64	default	20260124_08:51
mint	victoria	amd64	default	20260124_08:51
mint	virginia	amd64	default	20260124_08:51
mint	wilma	amd64	default	20260124_08:51
nixos	25.11	amd64	default	20260124_01:00
nixos	25.11	arm64	default	20260124_01:50
nixos	unstable	amd64	default	20260124_01:02
nixos	unstable	arm64	default	20260124_01:50
openeuler	20.03	amd64	default	20260120_15:48
openeuler	20.03	arm64	default	20260120_15:48
openeuler	22.03	amd64	default	20260120_15:48
openeuler	22.03	arm64	default	20260120_15:48
openeuler	24.03	amd64	default	20260120_15:48
openeuler	24.03	arm64	default	20260120_15:48
openeuler	25.03	amd64	default	20260120_15:48
openeuler	25.03	arm64	default	20260120_15:48
openeuler	25.09	amd64	default	20260120_15:48
openeuler	25.09	arm64	default	20260120_15:48
opensuse	15.6	amd64	default	20260124_04:20
opensuse	15.6	arm64	default	20260124_04:20
opensuse	16.0	amd64	default	20260124_04:20
opensuse	16.0	arm64	default	20260124_04:20
opensuse	tumbleweed	amd64	default	20260124_04:20
opensuse	tumbleweed	arm64	default	20260124_04:20
openwrt	23.05	amd64	default	20260124_11:57
openwrt	23.05	arm64	default	20260124_11:57

openwrt	24.10	amd64	default	20260124_11:57
openwrt	24.10	arm64	default	20260124_11:57
openwrt	25.12	amd64	default	20260124_11:57
openwrt	25.12	arm64	default	20260124_11:57
openwrt	snapshot	amd64	default	20260124_11:57
openwrt	snapshot	arm64	default	20260124_11:57
oracle	7	amd64	default	20260124_07:46
oracle	7	arm64	default	20260124_08:10
oracle	8	amd64	default	20260124_07:46
oracle	8	arm64	default	20260124_08:12
oracle	9	amd64	default	20260124_07:46
oracle	9	arm64	default	20260124_08:09
plamo	8.x	amd64	default	20260124_03:08
rockylinux	10	amd64	default	20260124_03:08
rockylinux	10	arm64	default	20260124_03:08
rockylinux	8	amd64	default	20260124_03:08
rockylinux	8	arm64	default	20260124_03:08
rockylinux	9	amd64	default	20260124_03:08
rockylinux	9	arm64	default	20260124_03:14
slackware	15.0	amd64	default	20260123_23:27
slackware	current	amd64	default	20260123_23:26
springdalelinux	7	amd64	default	20260124_06:38
springdalelinux	8	amd64	default	20260124_06:38
springdalelinux	9	amd64	default	20260124_06:38
ubuntu	jammy	amd64	default	20260124_07:42
ubuntu	jammy	arm64	default	20260124_07:42
ubuntu	jammy	armhf	default	20260124_08:02
ubuntu	jammy	riscv64	default	20260124_08:22
ubuntu	noble	amd64	default	20260124_07:42
ubuntu	noble	arm64	default	20260124_07:42
ubuntu	noble	armhf	default	20260124_07:42
ubuntu	noble	riscv64	default	20260124_07:42
ubuntu	plucky	amd64	default	20260124_07:42
ubuntu	plucky	arm64	default	20260124_07:48
ubuntu	plucky	armhf	default	20260124_08:07
ubuntu	plucky	riscv64	default	20260124_08:24
ubuntu	questing	amd64	default	20260124_07:42
ubuntu	questing	arm64	default	20260124_07:42
ubuntu	questing	armhf	default	20260124_07:42
voidlinux	current	amd64	default	20260124_17:10


```
voidlinux          current      arm64  default  20260124_17:10
```

```
---
```

Distribution:

debian

Release:

trixie

Architecture:

amd64

Downloading the image index

Downloading the rootfs

Downloading the metadata

The image cache is now ready

Unpacking the rootfs

```
---
```

You just created a Debian trixie amd64 (20260124_05:24) container.

To enable SSH, run: `apt install openssh-server`

No default root or user password are set by LXC.

Um den Container beim Booten automatisch starten zu lassen, muss eine Zeile zu seiner Config hinzugefügt werden:

```
echo "lxc.start.auto = 1" >> /var/lib/lxc/<name>/config
```

Damit der Container auch Netzwerk hat, wird ganz normal seine `/etc/systemd/network/eth0.network` angepasst. Z.B:

```
cat >>/var/lib/lxc/<name>/rootfs/etc/systemd/network/eth0.network <<EOF
```

```
[Match]
```

```
Name=eth0
```

```
[Network]
```

```
Address=176.9.167.13/28
```

```
Gateway=176.9.167.1
```

```
DNS=10.10.10.3
```

```
DNS=2a01:4f8:262:445e:100::3
```

```
Address=2a01:4f8:262:445e:300::d/72
```

```
Gateway=2a01:4f8:262:445e:300::1
```

```
EOF
```

Um Probleme beim Neustarten der Container zu vermeiden ist es des Weiteren sinnvoll eine feste MAC-Adresse für das Netzwerk-Interface festzulegen:

```
echo "lxc.net.0.hwaddr = f2:93:8c:e9:93:98" >> /var/lib/lxc/<name>/config
```

Mit `lxc-autostart` kann der Container gestartet werden. Mit `lxc-attach -n jmp100.prima.de` betritt man den Container und installiert alles wichtige, wie einen SSH-Server.

Revision #1

Created 2026-01-24 22:38:24 UTC by Daniel Hess

Updated 2026-01-24 23:19:35 UTC by Daniel Hess