



# systemd im Alltag

16. April 2016



Bernd Müller  
Linux/OpenSource Consultant & Trainer  
B1 Systems GmbH  
mueller@b1-systems.de

# Vorstellung B1 Systems

- gegründet 2004
- primär Linux/Open Source-Themen
- national & international tätig
- fast 100 Mitarbeiter
- unabhängig von Soft- und Hardware-Herstellern
- Leistungsangebot:
  - Beratung & Consulting
  - Support
  - Entwicklung
  - Training
  - Betrieb
  - Lösungen
- dezentrale Strukturen

# Schwerpunkte

- Virtualisierung (XEN, KVM & RHEV)
- Systemmanagement (Spacewalk, Red Hat Satellite, SUSE Manager)
- Konfigurationsmanagement (Puppet & Chef)
- Monitoring (Nagios & Icinga)
- IaaS Cloud (Native OpenStack & SUSE Cloud & Red Hat OpenStack Platform)
- Hochverfügbarkeit (Pacemaker)
- Shared Storage (GPFS, OCFS2, DRBD & CEPH)
- Dateiaustausch (ownCloud)
- Paketierung (Open Build Service)
- Administratoren oder Entwickler zur Unterstützung des Teams vor Ort

# Vorstellung Bernd Müller

- ausgebildeter Fachinformatiker-Systemintegration
- Teil der B1-Crew seit 03/2009
- Linux/Unix Consultant und Trainer
- Vortrag auf der LIT 2010 über Datenverschlüsselung

**Vortragslink** [https://www.luga.de/Angebote/Vortraege/  
Verschluesselung\\_LIT\\_2010](https://www.luga.de/Angebote/Vortraege/Verschluesselung_LIT_2010)

- Vortrag auf der LIT 2014 über ownCloud

**Vortragslink** [https://www.luga.de/Angebote/Vortraege/ownCloud\\_II\\_LIT\\_2014](https://www.luga.de/Angebote/Vortraege/ownCloud_II_LIT_2014)



# Übersicht

# Übersicht

- old and well-hung
- new hotness - Was ist systemd?
- systemd Informationen
- SysVinit vs systemd
  - Runlevel
  - Power Management
  - Service Management
  - Hostname
  - Journal vs Logging

# Übersicht

- systemd
  - On demand
  - Automount
  - Remote
  - Analyse
  - Netzwerk
  - Timer
- Links



old and well-hung



# old and well-hung

- SysVinit
- Upstart
- OpenRC
- ...

new hotness - Was ist systemd?

# Was ist systemd?

- System und Service Manager für Linux
- erstmals erschienen im April 2010
- kompatibel mit SysV und LSB Init Skripten
- Parallelisierungsfähig unter Verwendung von Sockets und D-Bus (Sockets halten Verbindung mit Programmen bei abgestürztem Dienst)
- on-demand Starten von Daemons
- Kontrolle der Prozesse unter Verwendung von Linux *control groups*
- Verwalten von Mount und Automount Points
- Remote-Verwaltung via SSH

# Was ist systemd?

- Servicekonfiguration unter `/usr/lib/systemd/system/`
- beim bootfest machen wird ein Link der Servicekonfiguration nach `/etc/systemd/system/target.wants` gesetzt
- selber geschriebene Servicekonfigurationen werden unter `/etc/systemd/system/` gespeichert
- Netzwerkkonfiguration unter `/etc/systemd/network/`



# systemd Informationen

# systemd Informationen

- Dbus `/etc/dbus-1/system.d/ & /usr/share/dbus-1/`
- PAM `/etc/pam.d/systemd-user`
- sysctl `/etc/sysctl.d/`
- systemd `/etc/systemd/`
- systemd generische Konfigurationen `/usr/lib/systemd/system/`
- tmpfiles `/etc/tmpfiles.d/, /usr/lib/tmpfiles.d` und `/run/tmpfiles.d`
- Kernel Install `/usr/lib/kernel/install.d/`
- Load Modules `/usr/lib/modules-load.d/`
- Udev Rules `/usr/lib/udev/rules.d/`
- DOC `/usr/share/doc/packages/systemd`
- Polkit/PolicyKit `/usr/share/polkit-1/actions/`

# systemd Units

## systemd erkennt die Units an der Endung

- `.service` Hintergrunddienste
- `.mount` Dateisysteme
- `.automount` Dateisysteme bei Zugriff
- `.path` überwacht via Inotify Dateien und Verzeichnisse und startet bei Zugriff diese Unit
- `.socket` Startet einen oder mehrere Sockets, sobald Zugriff auf einen Socket wird der dazugehörige Service gestartet, ähnlich `inetd`
- `.target` Aufruf anderer Units, vergleichbar mit den Runlevel
- `.timer` arbeitet ähnlich Cron und startet `.service` zu bestimmten Zeiten
- `.device` Von `udev` erstellte Pseudo-Units für Abhängigkeiten
- `.nspawn` zur Konfiguration von `nspawns`

# systemd Units

## systemd erkennt die Units an der Endung

- `.network` Netzwerkkonfiguration
- `.netdev` Konfiguration virtueller Netzwerkinterfaces
- `.link` Link-Konfiguration in `/etc/systemd/network`
- `.negative|positive` Konfiguration für DNSSEC (`dnssec-trust-anchors.d (5)`)



# SysVinit vs systemd(229)

# systemd - Runlevel

# Runlevel

## Runlevel

	<b>systemd</b>	<b>SysVinit</b>
Derzeitiges Runlevel	<code>systemctl get-default</code>	<code>runlevel</code>
In Runlevel wechseln	<code>systemctl isolate graphical.target</code>	<code>init/telinit 5</code>
Default Runlevel einstellen	<code>systemctl set-default graphical.target</code>	<code>vim /etc/inittab</code>
System halt	<code>poweroff.target</code>	0
Single user mode	<code>rescue.target</code>	1,s,single
Multi user	<code>multi-user.target</code>	2
Multi user mit Netzwerk	<code>multi-user.target</code>	3
Experimental	<code>multi-user.target</code>	4
Multi user, Netzwerk, graphical	<code>graphical.target</code>	5
Reboot	<code>reboot.target</code>	6
Emergency	<code>emergency.target</code>	<code>emergency</code>



# systemd - Power Management

# Power Management

## Power Management

	<b>systemd</b>	<b>SysVinit</b>
Herunterfahren	<code>systemctl poweroff</code>	<code>shutdown -h now/halt/init 0</code>
Neustarten	<code>systemctl reboot</code>	<code>shutdown -r now/reboot/init 6</code>
Suspend/Hibernate	<code>systemctl suspend/hibernate</code>	<code>pm-suspend/pm-hibernate</code>

# systemd - Service Management

# Service Management

## Service Management

	<b>systemd</b>	<b>SysVinit</b>
Dienst starten	<code>systemctl start httpd(.service)</code>	<code>/etc/init.d/httpd start service httpd start</code>
Dienst stoppen	<code>systemctl stop httpd(.service)</code>	<code>/etc/init.d/httpd stop service httpd stop</code>
Dienst neustarten	<code>systemctl restart httpd(.service)</code>	<code>/etc/init.d/httpd restart service httpd restart</code>
Dienst neuladen	<code>systemctl reload httpd(.service)</code>	<code>/etc/init.d/httpd reload service httpd reload</code>
Dienst Status	<code>systemctl status httpd(.service)</code>	<code>/etc/init.d/httpd status service httpd status</code>
Dienst automatisch starten	<code>systemctl enable httpd(.service)</code>	<code>chkconfig httpd on</code>
Dienst nicht automatisch starten	<code>systemctl disable httpd(.service)</code>	<code>chkconfig httpd off</code>

# Service Management

## Service Management

	<b>systemd</b>	<b>SysVinit</b>
Status der automatisch startenden Dienste	<code>systemctl is-enabled httpd(.service)</code>	<code>chkconfig -l httpd</code>
Status der installierten Dienste	<code>systemctl list-unit-files</code>	<code>chkconfig -l</code>



# Service Management

## Service Management

	<b>systemd</b>	<b>SysVinit</b>
Alle fehlgeschlagenen Dienste listen	<code>systemctl --(state=)failed</code>	for Schleife über alle Dienst und grep
Reset fehlgeschlagene Dienste	<code>systemctl reset-failed PATTERN</code>	for Schleife über alle Dienste, grep, restart
Dienste maskieren	<code>systemctl mask (--runtime) (--now) httpd(.service)</code>	—
Dienste demaskieren	<code>systemctl unmask httpd(.service)</code>	—
Dienstekonfiguration neu einlesen	<code>systemctl daemon-reload</code>	<code>chkconfig httpd --add</code>
Hilfe anzeigen	<code>systemctl help httpd(.service)/PID</code>	—
Abhängigkeiten anzeigen	<code>systemctl list-dependencies httpd(.service)</code>	—

# systemd - Hostname

# Hostname

## Hostname

	<b>systemd</b>	<b>SysVinit</b>
Derzeitiger Hostname	<code>hostnamectl (status)</code>	<code>hostname (-f)</code>
Hostname setzen	<code>hostnamectl set-hostname vm.lit2016.aux</code>	<code>hostname vm.lit2016.aux</code>
„Pretty“ Hostname	<code>hostnamectl set-hostname -pretty „VM für den LIT 2016“</code>	—
„Static“ Hostname	<code>hostnamectl set-hostname -static vm</code>	—
„Transient“ Hostname	<code>hostnamectl set-hostname -transient vm-net</code>	—

# Hostname

- „Pretty“  
beschreibt einen frei wählbaren Namen mit sehr wenigen Restriktionen und wird unter `/etc/machine-info` gespeichert.
- „Static“  
ist derjenige, welche beim Starten des Computers dem Kernel übergeben wird und wird unter `/etc/hostname` abgelegt.
- „Transient“  
wird als Fallback aus der Netzwerkkonfiguration bezogen.

Die beiden letzten Namen haben die Restriktion, welche für Internet Domänen gilt.

# Hostname

- `hostnamectl set-icon-name computer-laptop`  
Setzt den System Icon Namen, welcher von grafischen Anwendungen ausgewertet werden kann zum Anzeigen des eigenen Systems. Icon Naming Specification
- `hostnamectl set-chassis laptop`  
Folgende Chassis Typen gibt es *desktop, laptop, server, tablet, handset, watch, embedded, vm* und *container*.
- `hostnamectl set-deployment production`  
Als Deployment Umgebungen können folgende Werte gesetzt werden, *development, integration, staging* und *production*.
- `hostnamectl set-location „World, Europe, Germany, Augsburg, Friedbergerstr. 2, Raum C,„`  
Hierbei darf alles eingegeben werden und beschreibt den Standort des Betriebssystems.

# systemd - Journal vs Logging

# Journal vs Logging

## Journal vs Logging

### systemd

Alle Logeinträge in Echtzeit verfolgen

```
journalctl -f
```

Logeinträge zu einem bestimmten Dienst verfolgen

```
journalctl -f -u httpd
```

Alle Logeinträge seit dem letzten Systemstart

```
journalctl -b
```

Alle Logeinträge eines Dienstes seit dem letzten Systemstart

```
journalctl -b -u httpd
```

### SysVinit

```
tail -f /var/log/*  
/var/log/httpd/*
```

```
tail -f /var/log/httpd/* tail  
-f /var/log/messages | grep  
httpd
```

grep nach jünger als uptime

grep nach jünger als uptime und dem Dienst



systemd



# systemd - On demand

## systemd - On demand

systemd Socket Activation ist vergleichbar mit inetd/xinetd und startet bei Bedarf einen Dienst. Dies wird ermöglicht, indem für einen Dienst ein Socket geöffnet wird, der den eigentlichen Dienst bei einer Anfrage startet. Dazu muss der Socket gestartet werden

### Socket Unit

```
systemctl start org.cups.cupsd.socket  
systemctl enable org.cups.cupsd.socket
```

# systemd - On demand

## Socket Unit

```
/usr/lib/systemd/system/org.cups.cupsd.socket  
[Unit]  
Description=CUPS Scheduler  
[Socket]  
ListenStream=/run/cups/cups.sock  
[Install]  
WantedBy=sockets.target
```

# systemd - On demand

## Service Unit

```
/usr/lib/systemd/system/org.cups.cupsd.service
[Unit]
Description=CUPS Scheduler
Documentation=man:cupsd(8)
[Service]
ExecStart=/usr/bin/cupsd -l
Type=simple
[Install]
Also=org.cups.cupsd.socket org.cups.cupsd.path
WantedBy=printer.target
```

# systemd - Automount

## systemd - Automount

Das automatische Einbinden kann mit `systemd` auf zwei Arten gemacht werden. Da `systemd` die `/etc/fstab` einliest und Mount Units erzeugt, kann dies über die Mountoptionen eingerichtet werden.

```
/etc/fstab
```

```
UUID=1865c6d8-aa95-4789-b7a4-e9ad38b2377c /mnt/ssd ext4 x-systemd.automount,noauto 0 1
```

Die automatisch erzeugten Mount Units sind unter `/run/systemd/generator` zu finden.

## systemd - Automount

Die zweite Art ist über das Anlegen eines Mount und eines Automount Units möglich. Dazu wird der Eintrag in der `/etc/fstab` wie folgt verarbeitet

```
/etc/fstab
```

```
#UUID=1865c6d8-aa95-4789-b7a4-e9ad38b2377c /mnt/ssd ext4 defaults 0 1
```

# systemd - Automount

## Mount Unit

```
/etc/systemd/system/mnt-ssd.mount
[Unit]
Description=Mount for external SSD
[Mount]
What=/dev/disk/by-uuid/1865c6d8-aa95-4789-b7a4-e9ad38b2377c
Where=/mnt/ssd
Type=ext4
Options=defaults
[Install]
WantedBy=multi-user.target
```



# systemd - Automount

## Automount Unit

```
cp /etc/systemd/system/mnt-ssd.mount /etc/systemd/system/mnt-ssd.automount
```

# systemd - Automount

## Automount Unit

```
[Unit]
```

```
Description=Automount for external SSD
```

```
[Automount]
```

```
Where=/mnt/ssd
```

```
TimeoutIdleSec=60
```

```
[Install]
```

```
WantedBy=multi-user.target
```

# systemd - Automount

## Aktiviere Automount Unit

```
systemctl daemon-reload  
systemctl enable mnt-ssd.automount  
systemctl start mnt-ssd.automount
```

# systemd - Automount

## Prüfe Automount Unit

```
mount
```

```
lsblk
```

```
ls
```

```
mount
```

```
lsblk
```



# systemd - Remote

# systemd - Remote

## Remote über das Netzwerk

- `systemctl -H hostname (status)`
- `systemctl -H hostname get-default`
- `systemctl -H hostname poweroff`
- `hostnamectl -H hostname (Machine/Boot ID Bug)`

# systemd - Remote

## Remote lokale Container

- `systemctl -M containername (status)`
- `systemctl -M containername get-default`
- `hostnamectl -M containername`
- `journalctl -M containername -f`

# systemd - Analyse



## systemd - Analyse

- `systemd-analyze (time)`: zeigt die Kernel und Userspace Zeit des Systemstarts an
- `systemd-analyze blame`: zeigt alle Dienste und deren Startzeit an
- `systemd-analyze critical-chain`: zeigt in einer Baumstruktur die zeitkritischen Einheiten einer Reihe
- `systemd-analyze plot > /dev/shm/local.svg`: erstellt eine SVG Grafik des Systemstarts
- `systemd-analyze plot -H 192.168.122.184 > /dev/shm/remote.svg`: erstellt die SVG Grafik eines Remotesystems

# systemd - Analyse

- folgenden Eintrag dem Kernel übergeben  
`init=/usr/lib/systemd/systemd-bootchart` und es werden folgende Daten gesammelt
  - CPU Last
  - HDD IO Last
  - RAM Auslastung
  - Informationen zu den einzelnen Prozessen
- Konfigurationsdatei `/etc/systemd/bootchart.conf`

# systemd - Netzwerk

## systemd - Netzwerk

Die Netzwerkfunktionalität in systemd wird über `systemd-networkd` und `systemd-resolved` gestartet. Folgender Link muss angelegt werden:

### resolv.conf

```
ln -s /run/systemd/resolve/resolv.conf /etc/resolv.conf
```

`systemd-resolved` kann über zwei Wege konfiguriert werden. Entweder global über `/etc/systemd/resolved.conf` oder direkt in der Netzwerkkonfiguration `/etc/systemd/network/*.network`.

# systemd - Netzwerk

## DHCP

```
/etc/systemd/network/enp1s0.network
```

```
[Match]
```

```
Name=enp1s0
```

```
[Network]
```

```
DHCP=yes
```

# systemd - Netzwerk

## Statische Konfiguration

```
/etc/systemd/network/enp1s0.network
[Match]
Name=enp1s0
[Network]
Address=192.168.122.184/24
Gateway=192.168.122.1
DNS=192.168.122.1
Domains=aux
[Link]
MTUBytes=1500
MACAddress=XX:XX:XX:XX:XX:XX
```

# systemd - Netzwerk

## Bond Kernel

```
/etc/modprobe.d/bonding.conf  
options bonding mode=802.3ad miimon=100
```

# systemd - Netzwerk

## Bond NICs

```
/etc/systemd/network/10-create-bond0.network  
[Match]  
Name=en0*/eno1 enp1s0  
[Network]  
Bond=bond0
```



# systemd - Netzwerk

## Bond virtuelles Netzwerkinterface

```
/etc/systemd/network/20-setup-bond0.netdev  
[NetDev]  
Name=bond0  
Kind=bond  
[Bond]  
Mode=802.3ad  
LACPTransmitRate=fast  
MIIMonitorSec=1s  
UpDelaySec=2s  
DownDelaySec=8s
```

# systemd - Netzwerk

## Bond Netzwerkkonfiguration

```
/etc/systemd/network/30-IP-bond0.network
```

```
[Match]
```

```
Name=bond0
```

```
[Network]
```

```
Address=192.168.122.2/24
```

```
Gateway=192.168.122.1
```

```
DNS=192.168.122.1
```

```
Domains=lit2016.aux
```



# systemd - Timer

# systemd - Timer

Die systemd Funktionalität Timer ist vergleichbar mit Cron und wird am Beispiel von logrotate erklärt.

# systemd - Timer

## Service Unit

```
/usr/lib/systemd/system/logrotate.service
[Unit]
Description=Rotate log files
[Service]
Type=oneshot
ExecStart=/usr/bin/logrotate /etc/logrotate.conf
Nice=19
IOSchedulingClass=best-effort
IOSchedulingPriority=7
```

# systemd - Timer

## Timer Unit

```
/usr/lib/systemd/system/logrotate.timer
[Unit]
Description=Daily rotation of log files
[Timer]
OnCalendar=daily
AccuracySec=12h
Persistent=true
```

# systemd - Timer

## Aktiviere Timer Unit

```
systemctl start logrotate.timer  
systemctl enable logrotate.timer
```

# systemd - Timer

```
systemctl list-timers --all
NEXT                LEFT                LAST                PASSED            UNIT
Sun 2016-04-10 00:00:00 CEST 1h 3min left Sat 2016-04-09 13:20:00 CEST 9h ago logrotate.timer
```



# Linksammlung

- offizielle systemd Webseite

<https://www.freedesktop.org/wiki/Software/systemd/>

- Lennart Poetterings Blog über systemd

[http:](http://0pointer.de/blog/projects/systemd-for-admins-1.html)

[//0pointer.de/blog/projects/systemd-for-admins-1.html](http://0pointer.de/blog/projects/systemd-for-admins-1.html)

- Chaos Radio Express (CRE) 209 Das Linux System Podcast mit Lennart Poettering

<http://cre.fm/cre209-das-linux-system>

- Icon Naming Specification

<http://standards.freedesktop.org/icon-naming-spec/icon-naming-spec-latest.html>

# Linksammlung

- 802.3ad - LACP

<http://ieeexplore.ieee.org/servlet/opac?punumber=6867>

- Vortrag auf der LIT 2010 über Datenverschlüsselung

[https://www.luga.de/Angebote/Vortraege/  
Verschluesselung\\_LIT\\_2010](https://www.luga.de/Angebote/Vortraege/Verschluesselung_LIT_2010)

- Vortrag auf der LIT 2014 über ownCloud

[https:  
//www.luga.de/Angebote/Vortraege/ownCloud\\_II\\_LIT\\_2014](https://www.luga.de/Angebote/Vortraege/ownCloud_II_LIT_2014)



Fragen?

Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an [info@b1-systems.de](mailto:info@b1-systems.de)  
oder +49 (0)8457 - 931096