



systemd im Alltag

Augsburger Linux-Infotag

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



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 I/II

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` erstellt einen oder mehrere Sockets; bei Zugriff auf einen Socket wird der dazugehörige Service gestartet, ähnlich `inetd`
- `.target` Aufruf anderer Units, vergleichbar mit den Runlevels
- `.timer` arbeitet ähnlich Cron und startet `.service` zu bestimmten Zeiten
- `.device` von `udev` erstellte Pseudo-Units für Abhängigkeiten
- `.nspawn` zur Konfiguration von `nspawn`s

systemd Units II/II

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“

	systemd	SysVinit
Derzeitiger „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>	emergency

systemd – Power Management

Power Management

Power Management

	systemd	SysVinit
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	<code>systemctl suspend</code>	<code>pm-suspend</code>
Hibernate	<code>systemctl hibernate</code>	<code>pm-hibernate</code>

systemd – Service Management

Service Management I/III

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

Service Management II/III

Service Management

	systemd	SysVinit
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 III/III

	systemd	SysVinit
Alle fehlgeschlagenen Dienste listen	<code>systemctl --(state=)failed</code>	for-Schleife über alle Dienste und <code>grep</code>
Reset fehlgeschlagene Dienste	<code>systemctl reset-failed <pattern></code>	for-Schleife über alle Dienste, <code>grep</code> , <code>restart</code>
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

	systemd	SysVinit
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; unter `/etc/machine-info` gespeichert

„Static“ Name, der beim Starten des Computers dem Kernel übergeben wird; unter `/etc/hostname` abgelegt

„Transient“ als Fallback aus der Netzwerkkonfiguration bezogen

Die beiden letzten Namen unterliegen den Restriktionen, die für Internet Domänen gelten.

Hostname

`hostnamectl set-icon-name computer-laptop` Setzt den System Icon Namen, der von grafischen Anwendungen zum Anzeigen des eigenen Systems ausgewertet werden kann.

`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"`
Hier darf alles eingegeben werden; beschreibt den Standort des Betriebssystems.

systemd – Journal vs. Logging

Journal vs. Logging

Journal vs. Logging

	systemd	SysVinit
Alle Logeinträge in Echtzeit verfolgen	<code>journalctl -f</code>	<code>tail -f /var/log/* /var/log/httpd/*</code>
Logeinträge zu einem bestimmten Dienst verfolgen	<code>journalctl -f -u httpd</code>	<code>tail -f /var/log/httpd/* tail -f /var/log/messages grep httpd</code>
Alle Logeinträge seit dem letzten Systemstart	<code>journalctl -b</code>	<code>grep nach „jünger als Up- time“</code>
Alle Logeinträge eines Dienstes seit dem letzten Systemstart	<code>journalctl -b -u httpd</code>	<code>grep nach „jünger als Up- time“ und dem Dienst</code>



systemd

systemd – On-demand

systemd – On-demand

- systemd Socket Activation vergleichbar mit inetd/xinetd
- startet bei Bedarf einen Dienst
- für Dienst wird ein Socket geöffnet, der den eigentlichen Dienst bei einer Anfrage startet

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

- zwei Möglichkeiten des automatischen Einbindens mit systemd
- erste Möglichkeit: systemd liest `/etc/fstab`, erzeugt Mount Units → Verwendung von Mountoptionen
- automatisch erzeugte Mount Units unter `/run/systemd/generator`

```
/etc/fstab
```

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

systemd – Automount

- zweite Möglichkeit: Anlegen einer Mount Unit und einer Automount Unit
- kein Eintrag in der `/etc/fstab`, Werte werden in Units übernommen

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 – Datei kopieren

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

Automount Unit –

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

```
[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!!)`

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

- Kernelparameter 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

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

Link anlegen

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

- `systemd-resolved` kann über zwei Wege konfiguriert werden:
 - 1 global über `/etc/systemd/resolved.conf`
 - 2 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

- systemd-Funktionalität „Timer“ ist vergleichbar mit Cron
- wird am Beispiel von logrotate vorgestellt

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

Timer auflisten

```
# systemctl list-timers --all
```

NEXT	LEFT	LAST
Sun 2016-04-10 ...	1h 3min left	Sat 2016-04-09 13:20:00 CEST

PASSED	UNIT	ACTIVATES
9h ago	logrotate.timer	logrotate.service

Linksammlung I/II

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

<http://0pointer.de/blog/projects/systemd-for-admins-1.html> Lennart
Poetterings Blog über systemd

<http://cre.fm/cre209-das-linux-system> Chaos Radio Express (CRE) 209
Podcast: „Das Linux System“ mit Lennart Poettering

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

Linksammlung II/II

<http://ieeexplore.ieee.org/servlet/opac?punumber=6867> 802.3ad –
LACP

https://www.luga.de/Angebote/Vortraege/Verschlusselung_LIT_2010
Vortrag auf der LIT 2010 über Datenverschlüsselung

https://www.luga.de/Angebote/Vortraege/ownCloud_II_LIT_2014 Vortrag
auf der LIT 2014 über ownCloud



Fragen?

Vielen Dank für Ihre Aufmerksamkeit!

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