

# systemd

Init der nächsten Generation

Stefan J. Betz

Backspace e.V.

19. Oktober 2014



- 1 Einleitung
- 2 Funktionen
- 3 Units
- 4 Praxisbeispiele
- 5 Migration
- 6 Anwendung, Tipps, Tricks



# Was ist systemd?

Wichtige Frage, wichtige Antwort!

Systemd ist eine moderne, von Grund auf neu entwickelte, Alternative zu klassischen Init Systemen. OK, es macht auch noch ein paar andere Dinge.



# Was ist systemd?

Wichtige Frage, wichtige Antwort!

Systemd ist eine moderne, von Grund auf neu entwickelte, Alternative zu klassischen Init Systemen. OK, es macht auch noch ein paar andere Dinge.



# Was ist systemd?

Wichtige Frage, wichtige Antwort!

Systemd ist eine moderne, von Grund auf neu entwickelte, Alternative zu klassischen Init Systemen. OK, es macht auch noch ein paar andere Dinge.



# Aktuelle Situation

Kleine (schmerzhafte) Bestandsaufnahme oder auch Realitätsabgleich

- Lose zusammenhängende Toolsammlung  
(`sysvinit`,`cron`,`atd`,`rsyslog`)
- `stdio/stderr` anyone?
- Abhängigkeiten anyone?
- Zu welchem Dienst gehört eigentlich `nsadatacopyagentd`?
- Akku? Welcher Akku?
- Schrödingers Prozesse



# Aktuelle Situation

Kleine (schmerzhafte) Bestandsaufnahme oder auch Realitätsabgleich

- Lose zusammenhängende Toolsammlung  
(`sysvinit`, `cron`, `atd`, `rsyslog`)
- `stdio/stderr` anyone?
- Abhängigkeiten anyone?
- Zu welchem Dienst gehört eigentlich `nsadatacopyagentd`?
- Akku? Welcher Akku?
- Schrödingers Prozesse



# Aktuelle Situation

Kleine (schmerzhafte) Bestandsaufnahme oder auch Realitätsabgleich

- Lose zusammenhängende Toolsammlung  
(sysvinit,cron,d,atd,rsyslog)
- `stdio/stderr` anyone?
- Abhängigkeiten anyone?
- Zu welchem Dienst gehört eigentlich `nsadatacopyagentd`?
- Akku? Welcher Akku?
- Schrödingers Prozesse





# Aktuelle Situation

Kleine (schmerzhafte) Bestandsaufnahme oder auch Realitätsabgleich

- Lose zusammenhängende Toolsammlung  
(`sysvinit`, `crond`, `atd`, `rsyslog`)
- `stdio/stderr` anyone?
- Abhängigkeiten anyone?
- Zu welchem Dienst gehört eigentlich `nsadatacopyagentd`?
- Akku? Welcher Akku?
- Schrödingers Prozesse



# Aktuelle Situation

Kleine (schmerzhafte) Bestandsaufnahme oder auch Realitätsabgleich

- Lose zusammenhängende Toolsammlung  
(sysvinit,cron,d,atd,rsyslog)
- `stdio/stderr` anyone?
- Abhängigkeiten anyone?
- Zu welchem Dienst gehört eigentlich `nsadatacopyagentd`?
- Akku? Welcher Akku?
- Schrödingers Prozesse



# Aktuelle Situation

Kleine (schmerzhafte) Bestandsaufnahme oder auch Realitätsabgleich

- Lose zusammenhängende Toolsammlung  
(sysvinit, crond, atd, rsyslog)
- `stdio/stderr` anyone?
- Abhängigkeiten anyone?
- Zu welchem Dienst gehört eigentlich `nsadacopyagentd`?
- Akku? Welcher Akku?
- Schrödingers Prozesse



# Aktuelle Situation

Kleine (schmerzhafte) Bestandsaufnahme oder auch Realitätsabgleich

- Lose zusammenhängende Toolsammlung  
(sysvinit, crond, atd, rsyslog)
- `stdio/stderr` anyone?
- Abhängigkeiten anyone?
- Zu welchem Dienst gehört eigentlich `nsadatacopyagentd`?
- Akku? Welcher Akku?
- Schrödingers Prozesse



# Aktuelle Anforderungen

Neue Anforderungen seit 1984. . .

- USB, SATA und anderes Hotplug Spielzeug
- Akkus
- Abhängigkeiten
- Container (Docker, LXC), VMs (ESX, Xen, Virtualbox)
- (Anwender die Geräte ausschalten/einschalten)



- USB, SATA und anderes Hotplug Spielzeug
- Akkus
- Abhängigkeiten
- Container (Docker, LXC), VMs (ESX, Xen, Virtualbox)
- (Anwender die Geräte ausschalten/einschalten)



# Aktuelle Anforderungen

Neue Anforderungen seit 1984...

- USB, SATA und anderes Hotplug Spielzeug
- Akkus
- Abhängigkeiten
- Container (Docker, LXC), VMs (ESX, Xen, Virtualbox)
- (Anwender die Geräte ausschalten/einschalten)



# Aktuelle Anforderungen

Neue Anforderungen seit 1984. . .

- USB, SATA und anderes Hotplug Spielzeug
- Akkus
- Abhängigkeiten
  - Container (Docker, LXC), VMs (ESX, Xen, Virtualbox)
  - (Anwender die Geräte ausschalten/einschalten)





- USB, SATA und anderes Hotplug Spielzeug
- Akkus
- Abhängigkeiten
- Container (Docker, LXC), VMs (ESX,Xen,Virtualbox)
- (Anwender die Geräte ausschalten/einschalten)



- USB, SATA und anderes Hotplug Spielzeug
- Akkus
- Abhängigkeiten
- Container (Docker, LXC), VMs (ESX,Xen,Virtualbox)
- (Anwender die Geräte ausschalten/einschalten)



# Unterschiede

Der eine hat den Dill, der andere die Gurken.

Unterschied zu sysvinit. . .

- Konfiguration (Unit) statt Scripting (bash)

Unterschied zu upstart. . .

- Abhängigkeiten statt Ereignisse



# Unterschiede

Der eine hat den Dill, der andere die Gurken.

## Unterschied zu sysvinit. . .

- Konfiguration (Unit) statt Scripting (bash)

## Unterschied zu upstart. . .

- Abhängigkeiten statt Ereignisse



# Unterschiede

Der eine hat den Dill, der andere die Gurken.

Unterschied zu sysvinit. . .

- Konfiguration (Unit) statt Scripting (bash)

Unterschied zu upstart. . .

- Abhängigkeiten statt Ereignisse



# Unterschiede

Der eine hat den Dill, der andere die Gurken.

Unterschied zu sysvinit. . .

- Konfiguration (Unit) statt Scripting (bash)

Unterschied zu upstart. . .

- Abhängigkeiten statt Ereignisse



# Unterschiede

Der eine hat den Dill, der andere die Gurken.

Unterschied zu sysvinit. . .

- Konfiguration (Unit) statt Scripting (bash)

Unterschied zu upstart. . .

- Abhängigkeiten statt Ereignisse



# Funktionen von systemd

## Features vom neuen Spielzeug

- Abhängigkeiten
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)





### ■ Abhängigkeiten

- ... zu Mountpoints
- ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

## ■ Abhängigkeiten

- ... zu Mountpoints
- ... zu Geräten

## ■ Socket Activation

## ■ Units (Konfiguration) statt Scripts

## ■ Control Groups

## ■ Parallel

## ■ User Sessions

## ■ Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)

## ■ (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

- **Abhängigkeiten**
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

- **Abhängigkeiten**
  - ... zu Mountpoints
  - ... zu Geräten
- **Socket Activation**
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

- Abhängigkeiten
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

- Abhängigkeiten
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
  - Parallel
  - User Sessions
  - Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
  - (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

- Abhängigkeiten
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

- Abhängigkeiten
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)





# Funktionen von systemd

Features vom neuen Spielzeug

- Abhängigkeiten
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)



# Funktionen von systemd

Features vom neuen Spielzeug

- Abhängigkeiten
  - ... zu Mountpoints
  - ... zu Geräten
- Socket Activation
- Units (Konfiguration) statt Scripts
- Control Groups
- Parallel
- User Sessions
- Ressource Limits (CPU, RAM, IO, ...), Permissions (Netzwerk, /tmp...)
- (Init Script kompatibel)



# Funktionen von journald

Binär und böse gut

- Integration in systemd
- Logrotate überflüssig
- Assoziation zwischen Dienst und Journal
- Forward Secure Sealing (FSS)
- (Speichert Daten binär)



# Funktionen von journald

Binär und böse gut

- Integration in systemd
- Logrotate überflüssig
- Assoziation zwischen Dienst und Journal
- Forward Secure Sealing (FSS)
- (Speichert Daten binär)



# Funktionen von journald

Binär und böse gut

- Integration in systemd
- Logrotate überflüssig
- Assoziation zwischen Dienst und Journal
- Forward Secure Sealing (FSS)
- (Speichert Daten binär)



# Funktionen von journald

Binär und böse gut

- Integration in systemd
- Logrotate überflüssig
- Assoziation zwischen Dienst und Journal
  - Forward Secure Sealing (FSS)
  - (Speichert Daten binär)



# Funktionen von journald

Binär und böse gut

- Integration in systemd
- Logrotate überflüssig
- Assoziation zwischen Dienst und Journal
- Forward Secure Sealing (FSS)
- (Speichert Daten binär)



- Integration in systemd
- Logrotate überflüssig
- Assoziation zwischen Dienst und Journal
- Forward Secure Sealing (FSS)
- (Speichert Daten binär)





# Weitere Bestandteile

Tools, Tools, Tools, ...

- `udev`
- `systemd-timesyncd`
- `systemd-networkd`
- `systemd-analyze`
- `systemd-nspawn`
- `systemd-tmpfiles`
- `systemd-run`



- udev
- systemd-timesyncd
- systemd-networkd
- systemd-analyze
- systemd-nspawn
- systemd-tmpfiles
- systemd-run



- `udev`
- `systemd-timesyncd`
- `systemd-networkd`
- `systemd-analyze`
- `systemd-nspawn`
- `systemd-tmpfiles`
- `systemd-run`



# Weitere Bestandteile

Tools, Tools, Tools, ...

- `udev`
- `systemd-timesyncd`
- `systemd-networkd`
- `systemd-analyze`
- `systemd-nspawn`
- `systemd-tmpfiles`
- `systemd-run`



- `udev`
- `systemd-timesyncd`
- `systemd-networkd`
- `systemd-analyze`
- `systemd-nspawn`
- `systemd-tmpfiles`
- `systemd-run`



- `udev`
- `systemd-timesyncd`
- `systemd-networkd`
- `systemd-analyze`
- `systemd-nspawn`
- `systemd-tmpfiles`
- `systemd-run`



# Weitere Bestandteile

Tools, Tools, Tools, ...

- `udev`
- `systemd-timesyncd`
- `systemd-networkd`
- `systemd-analyze`
- `systemd-nspawn`
- `systemd-tmpfiles`
- `systemd-run`



- `udev`
- `systemd-timesyncd`
- `systemd-networkd`
- `systemd-analyze`
- `systemd-nspawn`
- `systemd-tmpfiles`
- `systemd-run`





Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar



Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar



Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar



Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar



Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar



Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar



Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar



Fast alle Dinge an systemd können über Units erledigt werden.

- Definieren wie Dinge erfolgen
- Abhängigkeiten der Dinge
- Textdateien
- Überladbar
- Instanzen
- Teilweise automatisch von sog. Generatoren (z.B. `systemd-fstab-generator`) erzeugt
- Unterschiedliche Typen verfügbar





## Auswahl verschiedener Unit Typen:

- `service`, für *normale* Dienste
- `timer`, als `crond/atd` Ersatz
- `mount`, für Mountpoints (`/etc/fstab`)
- `socket`, für Sockets
- `device`, für Geräte



Auswahl verschiedener Unit Typen:

- **service**, für *normale* Dienste
- timer, als crond/atd Ersatz
- mount, für Mountpoints (/etc/fstab)
- socket, für Sockets
- device, für Geräte



Auswahl verschiedener Unit Typen:

- service, für *normale* Dienste
- timer, als crond/atd Ersatz
- mount, für Mountpoints (/etc/fstab)
- socket, für Sockets
- device, für Geräte



Auswahl verschiedener Unit Typen:

- `service`, für *normale* Dienste
- `timer`, als `crond/atd` Ersatz
- `mount`, für Mountpoints (`/etc/fstab`)
- `socket`, für Sockets
- `device`, für Geräte



Auswahl verschiedener Unit Typen:

- service, für *normale* Dienste
- timer, als crond/atd Ersatz
- mount, für Mountpoints (/etc/fstab)
- socket, für Sockets
- device, für Geräte



Auswahl verschiedener Unit Typen:

- `service`, für *normale* Dienste
- `timer`, als `crond/atd` Ersatz
- `mount`, für Mountpoints (`/etc/fstab`)
- `socket`, für Sockets
- `device`, für Geräte



OpenVPN Init Script als Negativbeispiel. Funktionsumfang:

- Mehrere Instanzen für verschiedene VPN Verbindungen nötig
- Möglichkeit (einfach) Konfigurationen zu starten/stoppen



OpenVPN Init Script als Negativbeispiel. Funktionsumfang:

- Mehrere Instanzen für verschiedene VPN Verbindungen nötig
- Möglichkeit (einfach) Konfigurationen zu starten/stoppen





OpenVPN Init Script als Negativbeispiel. Funktionsumfang:

- Mehrere Instanzen für verschiedene VPN Verbindungen nötig
- Möglichkeit (einfach) Konfigurationen zu starten/stoppen



# OpenVPN, mit sysvinit

Wie läuft das bisher?

Bestandsaufnahme vom Ubuntu 14.04 OpenVPN Init Script:

- 314 Zeilen `bash` Code (9.9 KiB)
- Deaktivieren von Instanzen durch umbenennen von Dateien in `/etc/openvpn`



# OpenVPN, mit sysvinit

Wie läuft das bisher?

Bestandsaufnahme vom Ubuntu 14.04 OpenVPN Init Script:

- 314 Zeilen bash Code (9.9 KiB)
- Deaktivieren von Instanzen durch umbenennen von Dateien in `/etc/openvpn`



# OpenVPN, mit sysvinit

Wie läuft das bisher?

Bestandsaufnahme vom Ubuntu 14.04 OpenVPN Init Script:

- 314 Zeilen bash Code (9.9 KiB)
- Deaktivieren von Instanzen durch umbenennen von Dateien in `/etc/openvpn`



# OpenVPN, mit systemd

Wie läuft das jetzt?

Bestandsaufnahme vom Arch Linux OpenVPN Service Unit:

```
[Unit]
```

```
Description=OpenVPN connection to %i
```

```
[Service]
```

```
Type=forking
```

```
ExecStart=/usr/bin/openvpn --cd /etc/openvpn\  
--config /etc/openvpn/%i.conf\  
--daemon openvpn@%i\  
--writepid /run/openvpn@%i.pid
```

```
PIDFile=/run/openvpn@%i.pid
```

```
[Install]
```

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



# Praxisbeispiel journald

Lasst uns doch mal Logs suchen...

```
[root@server ~]# systemctl status ntpd
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled)
   Active: active (running) since Fri 2014-09-26 16:16:17 CEST; 2 weeks 4 days ago
 Main PID: 770 (ntpd)
   CGroup: /system.slice/ntpd.service
           └─770 /usr/bin/ntpd -g -u ntp:ntp

Sep 26 16:16:17 server ntpd[770]: Listen normally on 3 enp5s0 192.168.1.1:123
Sep 26 16:16:17 server ntpd[770]: Listen normally on 4 guestnet 192.168.2.1:123
Sep 26 16:16:17 server ntpd[770]: Listen normally on 5 tun2 172.18.148.21:123
Sep 26 16:16:17 server ntpd[770]: Listen normally on 6 lo [::1]:123
Sep 26 16:16:17 server ntpd[770]: Listen normally on 7 enp5s0 [fd07:4763:c4fd:8192::1]:123
Sep 26 16:16:17 server ntpd[770]: Listen normally on 8 enp5s0 [fe80::6a05:caff:fe2b:3246%2]:123
Sep 26 16:16:17 server ntpd[770]: Listen normally on 9 guestnet [fe80::6a05:caff:fe2b:3246%3]:123
Sep 26 16:16:17 server ntpd[770]: Listening on routing socket on fd #26 for interface updates
Sep 26 16:16:17 server ntpd[770]: restrict default: KOD does nothing without LIMITED.
Sep 26 16:16:17 server ntpd[770]: switching logging to file /var/log/ntp.log
```

Abbildung: systemctl status ntpd



# Migration

## Migration nach systemd

- Init Scripts kicken...
- ... sind teilweise ohnehin kaputt.
- Upstart Jobs sind für systemd komplett unbrauchbar
- supervisord wird (meist) überflüssig
- logrotate, crond, atd und sysvlog können weg
- Upstream (Entwickler) Units sind wenn vorhanden zu bevorzugen



- Init Scripts kicken...
- ... sind teilweise ohnehin kaputt.
- Upstart Jobs sind für systemd komplett unbrauchbar
- supervisord wird (meist) überflüssig
- logrotate, crond, atd und sysvlog können weg
- Upstream (Entwickler) Units sind wenn vorhanden zu bevorzugen





- Init Scripts kicken...
- ...sind teilweise ohnehin kaputt.
- Upstart Jobs sind für systemd komplett unbrauchbar
- supervisord wird (meist) überflüssig
- logrotate, crond, atd und sysvlog können weg
- Upstream (Entwickler) Units sind wenn vorhanden zu bevorzugen



- Init Scripts kicken...
- ...sind teilweise ohnehin kaputt.
- Upstart Jobs sind für systemd komplett unbrauchbar
- supervisord wird (meist) überflüssig
- logrotate, crond, atd und sysvlog können weg
- Upstream (Entwickler) Units sind wenn vorhanden zu bevorzugen



# Migration

## Migration nach systemd

- Init Scripts kicken...
- ...sind teilweise ohnehin kaputt.
- Upstart Jobs sind für systemd komplett unbrauchbar
- supervisord wird (meist) überflüssig
  - logrotate, crond, atd und sysvlog können weg
  - Upstream (Entwickler) Units sind wenn vorhanden zu bevorzugen



# Migration

## Migration nach systemd

- Init Scripts kicken...
- ...sind teilweise ohnehin kaputt.
- Upstart Jobs sind für systemd komplett unbrauchbar
- supervisord wird (meist) überflüssig
- logrotate, crond, atd und sysvlog können weg
- Upstream (Entwickler) Units sind wenn vorhanden zu bevorzugen



# Migration

## Migration nach systemd

- Init Scripts kicken...
- ...sind teilweise ohnehin kaputt.
- Upstart Jobs sind für systemd komplett unbrauchbar
- supervisord wird (meist) überflüssig
- logrotate, crond, atd und sysvlog können weg
- Upstream (Entwickler) Units sind wenn vorhanden zu bevorzugen



# Anwendung

Kurzanleitung in sehr kurz

Dienst aktivieren und starten:

```
sudo systemctl enable dienst  
sudo systemctl start dienst
```

Problematische Dienste:

```
systemctl --failed
```

Vollständiges Log vom aktuellen Bootvorgang:

```
journalctl -b
```

Log von einem Dienst seit dem Bootvorgang anzeigen, umgekehrt chronologisch sortiert:

```
journalctl -b -r -u dienst
```



Dienst aktivieren und starten:

```
sudo systemctl enable dienst  
sudo systemctl start dienst
```

Problematische Dienste:

```
systemctl --failed
```

Vollständiges Log vom aktuellen Bootvorgang:

```
journalctl -b
```

Log von einem Dienst seit dem Bootvorgang anzeigen, umgekehrt chronologisch sortiert:

```
journalctl -b -r -u dienst
```



Dienst aktivieren und starten:

```
sudo systemctl enable dienst  
sudo systemctl start dienst
```

Problematische Dienste:

```
systemctl --failed
```

Vollständiges Log vom aktuellen Bootvorgang:

```
journalctl -b
```

Log von einem Dienst seit dem Bootvorgang anzeigen, umgekehrt chronologisch sortiert:

```
journalctl -b -r -u dienst
```





Dienst aktivieren und starten:

```
sudo systemctl enable dienst
sudo systemctl start dienst
```

Problematische Dienste:

```
systemctl --failed
```

Vollständiges Log vom aktuellen Bootvorgang:

```
journalctl -b
```

Log von einem Dienst seit dem Bootvorgang anzeigen, umgekehrt chronologisch sortiert:

```
journalctl -b -r -u dienst
```



Dienst aktivieren und starten:

```
sudo systemctl enable dienst  
sudo systemctl start dienst
```

Problematische Dienste:

```
systemctl --failed
```

Vollständiges Log vom aktuellen Bootvorgang:

```
journalctl -b
```

Log von einem Dienst seit dem Bootvorgang anzeigen, umgekehrt chronologisch sortiert:

```
journalctl -b -r -u dienst
```



Anzeigen wann welches Timer Unit lief bzw. als nächstes laufen wird:

```
systemctl list-timers
```

Wie lange hat eigentlich der Bootvorgang gedauert?

```
systemd-analyze
```

Ist das eine virtuelle Maschine?

```
systemd-detect-virt
```

Programm in einem sauberen Environment ausführen, ohne ein Unit zu erstellen:

```
systemd-run programm
```



Anzeigen wann welches Timer Unit lief bzw. als nächstes laufen wird:

```
systemctl list-timers
```

Wie lange hat eigentlich der Bootvorgang gedauert?

```
systemd-analyze
```

Ist das eine virtuelle Maschine?

```
systemd-detect-virt
```

Programm in einem sauberen Environment ausführen, ohne ein Unit zu erstellen:

```
systemd-run programm
```



Anzeigen wann welches Timer Unit lief bzw. als nächstes laufen wird:

```
systemctl list-timers
```

Wie lange hat eigentlich der Bootvorgang gedauert?

```
systemd-analyze
```

Ist das eine virtuelle Maschine?

```
systemd-detect-virt
```

Programm in einem sauberen Environment ausführen, ohne ein Unit zu erstellen:

```
systemd-run programm
```



Anzeigen wann welches Timer Unit lief bzw. als nächstes laufen wird:

```
systemctl list-timers
```

Wie lange hat eigentlich der Bootvorgang gedauert?

```
systemd-analyze
```

Ist das eine virtuelle Maschine?

```
systemd-detect-virt
```

Programm in einem sauberen Environment ausführen, ohne ein Unit zu erstellen:

```
systemd-run programm
```



Vielen Dank für die Aufmerksamkeit!

Die Folien unterliegen der CreativeCommons  
„Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Unported“.



2014 Stefan J. Betz

