

# Systemd Crashcourse

Stefan Hornburg

# Systemd

- All-In-One
- Abhängigkeiten
- Parallele Ausführung
- Geschwindigkeit
- Prozesse überwachen
- Ressourcen + Security

# Bestandteile

- Services
- Logging
- Hostname/Locale/Keyboard/Time/...
- Temporäre Dateien
- Timers (Cron-Ersatz)
- ...

# Units

- Services (.service)
- Gruppen (.target)
- Slice (.slice)
- Scope (.scope)
- Dateisysteme (.mount / .automount)
- Überwachung (.path)
- Sockets (.socket)
- Netzwerke (.network)
- Device (.device)
- ...

# Targets

- graphical.target
- multiuser.target
- rescue.target
- default.target

# Kommandozeile

- systemctl
- journalctl

# Services

```
systemctl status nginx  
systemctl stop nginx  
systemctl start nginx  
systemctl enable nginx  
systemctl disable nginx
```

# Service Status

```
~# systemctl status ntp
ntp.service - LSB: Start NTP daemon
   Loaded: loaded (/etc/init.d/ntp; generated; vendor prese
   Active: active (running) since Mon 2018-03-26 10:15:10 C
   Docs: man:systemd-sysv-generator(8)
  Process: 581 ExecStart=/etc/init.d/ntp start (code=exited
   Tasks: 2 (limit: 4915)
   CGroup: /system.slice/ntp.service
           594 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 10
```



# Fehlerbehandlung

```
~# systemctl start nginx
```

Job for nginx.service failed because the control process exited with error code.

See "systemctl status nginx.service" and "journalctl -xe" for details.

# Fehlerbehandlung

```
~# systemctl start elasticsearch  
~#
```

# Fehlerbehandlung

```
~# systemctl --failed
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
elasticsearch.service	loaded	failed	failed	Elasticsearch

# Fehlerbehandlung



# Fehlerbehandlung

```
# There is insufficient memory for the Java Runtime Environment to  
continue.  
# Native memory allocation (mmap) failed to map 3046768640 bytes fo  
committing reserved memory.
```

# Unit anzeigen

```
~# systemctl cat nginx
# /lib/systemd/system/nginx.service
[Unit]
After=network.target

[Service]
Type=forking
PIDFile=/run/nginx.pid
ExecStartPre=/usr/sbin/nginx -t -q -g 'daemon on; master_pr
ExecStart=/usr/sbin/nginx -g 'daemon on; master_process on;
ExecReload=/usr/sbin/nginx -g 'daemon on; master_process on
ExecStop=-/sbin/start-stop-daemon --quiet --stop --retry QU
TimeoutStopSec=5
KillMode=mixed

[Install]
WantedBy=multi-user.target
```

# Liste der Units

- Units auflisten:  
`systemctl list-units`
- Unitdateien:  
`systemctl list-unit-files`

# Filter

- Unit-Typ (Services):  
`systemctl list-units --type=service`
- Inaktive und fehlende Services:  
`systemctl list-units --type=service --all`



# Abhängigkeiten anzeigen

- Hierarchie:

```
systemctl list-dependencies multi-user.target
```

- Kinder:

```
systemctl show -p Wants multi-user.target
```

# Default Target

- Anzeige

```
systemctl get-default
```

- Voreinstellung

```
systemctl set-default multi-user.target
```

- Laufzeitänderung

```
systemctl isolate multi-user.target
```

# Dateisystem

- Standard

`/lib/systemd/system`

- Custom (Lokal, Ansible)

`/etc/systemd/system`

- Runtime

`/run/systemd/system`

# Struktur Unitdatei

```
[Unit]
```

```
Description=The PHP FastCGI Process Manager
```

```
After=network.target
```

```
[Service]
```

```
Type=notify
```

```
PIDFile=/var/run/php5-fpm.pid
```

```
ExecStartPre=/usr/lib/php5/php5-fpm-checkconf
```

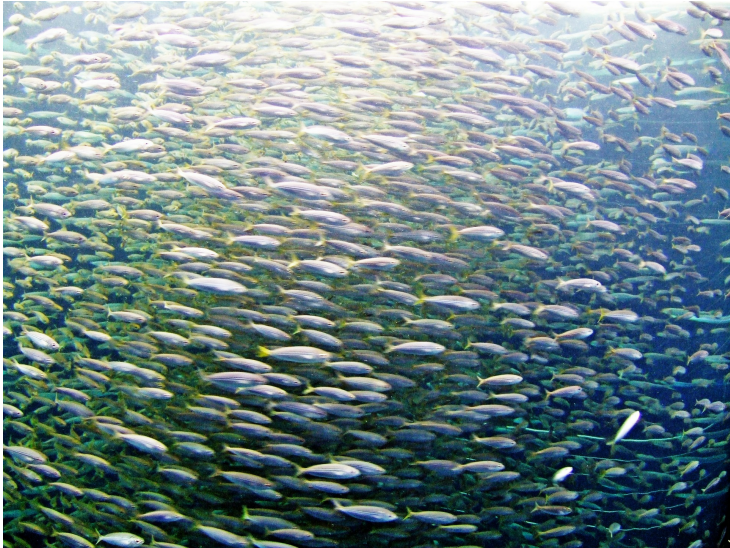
```
ExecStart=/usr/sbin/php5-fpm --nodaemonize \  
--fpm-config /etc/php5/fpm/php-fpm.conf
```

```
ExecReload=/bin/kill -USR2 $MAINPID
```

```
[Install]
```

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

# Direktiven



# Abschnitte

- [Unit]
- [Service]
- [Install]
- ...

# Environment

```
[Service]
```

```
Environment=ES_HOME=/usr/share/elasticsearch
```

```
Environment=CONF_DIR=/etc/elasticsearch/es1
```

```
EnvironmentFile=-/etc/default/es1_elasticsearch
```

```
WorkingDirectory=/usr/share/elasticsearch
```

```
User=elasticsearch
```

```
Group=elasticsearch
```

# Restart

```
[Service]  
Restart=always
```



# Units bearbeiten

- Unitdatei kopieren und bearbeiten
- Drop-in erstellen

# Drop-in erstellen

```
# systemctl show -p Restart nginx  
Restart=no
```

```
# systemctl edit nginx  
[Service]  
Restart=always
```

```
# systemctl show -p Restart nginx  
Restart=always
```

# Service-Typen

- simple
- notify
- forking
- ...

# Typ simple

```
ExecStart=/usr/share/elasticsearch/bin/elasticsearch \  
          -p ${PID_DIR}/elasticsearch.pid \  
          --quiet
```

# Fire and Forget



# Workarounds

- ExecStartPre  
ExecStartPre=/usr/lib/php5/php5-fpm-checkconf
- Typ notify  
Änderung am Service erforderlich

# Protokollierung

- Binärformat
  - `/run/log/journal` (gelöscht beim Booten)
  - `/var/log/journal` (persistent)
- Syslog
  - rsyslog liest Journal
  - `ForwardToSyslog=yes`

# Protokollierung

```
journalctl -n 1000 -f
```



# Filter für journalctl

- Service

```
journalctl -u nginx
```

- Zeit

```
journalctl --since 09:00 --until "1 hour ago"
```

- Priorität

```
journalctl -p err
```

# Systemd & SysV-Init

- Initskripte
- Runlevel
- Inetd

# Einbindung alte Services

- Systemstart
- Generierung Units  
`systemd-sysv-generator`

# NTP Service I

```
~# systemctl cat ntp  
# /run/systemd/generator.late/ntp.service  
# Automatically generated by systemd-sysv-generator
```

[Unit]

Documentation=man:systemd-sysv-generator(8)

SourcePath=/etc/init.d/ntp

Description=LSB: Start NTP daemon

Before=multi-user.target

Before=multi-user.target

Before=multi-user.target

Before=graphical.target

After=network-online.target

After=remote-fs.target

Wants=network-online.target

# NTP Service II

```
[Service]
Type=forking
Restart=no
TimeoutSec=5min
IgnoreSIGPIPE=no
KillMode=process
GuessMainPID=no
RemainAfterExit=yes
SuccessExitStatus=5 6
ExecStart=/etc/init.d/ntp start
ExecStop=/etc/init.d/ntp stop
```

# Runlevel

- Obsolet in Systemd
- Vergleichbar mit Targets
  - 1 emergency.target
  - 3 multi-user.target
  - 5 graphical.target

# Vor- und Nachteile

- 😊 Abhängigkeiten
- 😊 Geschwindigkeit
- 😞 Unix-Philosophie
- 😞 Fallgrube Service-Typ "simple"
- 😞 Dokumentation

# Referenzen

- ArchWiki
- Demystifying systemd
- <https://www.digitalocean.com/community/tutorials/understanding-systemd-units-and-unit-files>
- <https://www.digitalocean.com/community/tutorials/how-to-use-journalctl-to-view-and-manipulate-systemd-logs>



# Fragen



# The end

