Systemd Crashcourse

Stefan Hornburg

Contents

1 Einführung

Systemd ist ein umfangreiches Management-Tool für System und Services mit entsprechenden Vor- und Nachteilen. SysV Init ist nur für ein kleines Subset verantwortlich.

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

2 Bestandteile

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

2.1 Unit-Typen

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

- ...

Der Unit-Typ Slice erzeugt die Cgroup-Hierarchie für Resource-Management.
Der Unit-Typ Scope gruppiert die Worker-Prozesse für einen Service.

2.2 Targets

Targets sind einfach eine Gruppe von Unitfiles.

Außerdem gibt es Targets, die beim Bootvorgang aufgerufen werden und grob die Funktion von Runlevels ausüben:

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

Dabei entspricht graphical grob dem Runlevel 1, multiuser dem Runlevel 3, rescue dem Runlevel 1 und default ist ein Alias für das aktive Target beim Booten.

3 Kommandozeile

- systemctl
- journalctl

3.1 Services

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

```
~# systemctl status ntp
ntp.service - LSB: Start NTP daemon
   Loaded: loaded (/etc/init.d/ntp; generated; vendor preset: enabled)
   Active: active (running) since Mon 2018-03-26 10:15:10 CEST; 25min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 581 ExecStart=/etc/init.d/ntp start (code=exited, status=0/SUCCESS)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/ntp.service
           594 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 108:113
```

3.2 Fehlerbehandlung

systemctl start gibt keine Fehlermeldungen aus. Im Default-Modus kann es sogar dazu kommen, daß gar keine Meldung kommt.

```
~# 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.
```

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

```
~# systemctl --failed
UNIT                        LOAD   ACTIVE SUB    DESCRIPTION
elasticsearch.service loaded failed failed Elasticsearch
```

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

3.3 Units

Mit SysV Init finden sich die Skripte für die Services in `/etc/init.d`.

```
~# 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_process on;'
ExecStart=/usr/sbin/nginx -g 'daemon on; master_process on;'
ExecReload=/usr/sbin/nginx -g 'daemon on; master_process on;' -s reload
ExecStop=-/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.
TimeoutStopSec=5
KillMode=mixed

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

- Units auflisten:

  `systemctl list-units`

- Unitdateien:

  `systemctl list-unit-files`

3.4 Filter

- Unit-Typ (Services):

  `systemctl list-units --type=service`

- Inaktive und fehlende Services:

  `systemctl list-units --type=service --all`

3.5 Targets

- Hierarchie:

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

- Kinder:

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

- Anzeige

  ```
  systemctl get-default
  ```

- Voreinstellung

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

- Laufzeitänderung

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

4 Unitdateien

4.1 Dateisystem

- Standard

```
/lib/systemd/system
```

- Custom (Lokal, Ansible)

```
/etc/systemd/system
```

- Runtime

```
/run/systemd/system
```

4.1.1 target.wants

4.2 Struktur Unitdatei

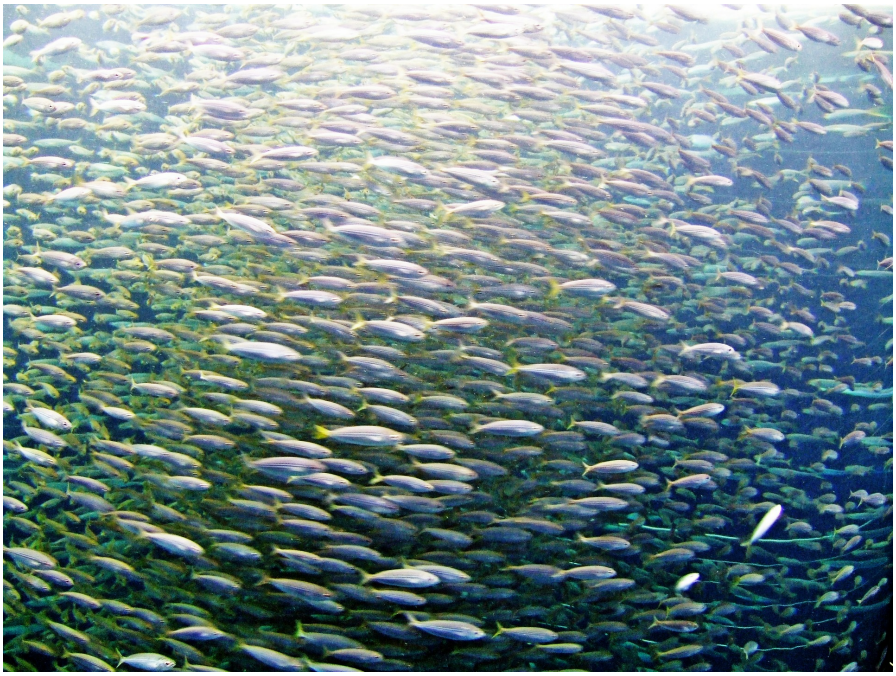
```
[Unit]
Description=The PHP FastCGI Process Manager
After=network.target

[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
```

4.3 Direktiven

- `[Unit]`
- `[Service]`
- `[Install]`



- ...

```
[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
```

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

4.4 Units bearbeiten

- Unitdatei kopieren und bearbeiten
- Drop-in erstellen

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

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

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

Hierbei gilt es auf die korrekte Syntax zu beachten, z.B. `[service]` statt `[Service]` führt zur Mißachtung der Anweisungen.

4.5 Service-Typen

- simple
- notify
- forking
- ...

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

- ExecStartPre

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

- Typ notify

  Änderung am Service erforderlich

5 Protokollierung

journalctl ist für die Protokollierung zuständig. Die Daten werden in einem Binärformat gespeichert.

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

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

5.1 Filter

- Service

  `journalctl -u nginx`
- Zeit

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

  `journalctl -p err`

6 Systemd & Sysvinit

- Initskripte
- Runlevel
- Inetd

6.1 Einbindung alte Services

- Systemstart
- Generierung Units

  `systemd-sysv-generator`

```
~# 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
```

```
[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
```

6.2 Runlevel

- Obsolet in Systemd
- Vergleichbar mit Targets

  1 emergency.target

  3 multi-user.target

  5 graphical.target

7 Abschluß

7.1 Vor- und Nachteile

- ☺ Abhängigkeiten
- ☺ Geschwindigkeit
- 😐 Unix-Philosophie
- ☹ Fallgrube Service-Typ ”simple”
- ☹ Dokumentation

7.2 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

7.3 Fragen

7.4 The end


END