systemd primer

DevOps Wroclaw meetup

2016-11-29

Maciej Lasyk

agenda

- systemd-bin
- 5-min break?
- service & unit file
- journal / loggins
- 5-min break?
- nspawn container
- integrating apps w/sd-notify

systemd - what is that?

systemd - learning

systemd - learning

```
man systemd.index
```

systemd - learning

```
man systemd.index
```

see what’s more…

```
man systemd[TAB][TAB]
```

systemd - bin

systemd - bin

systemctl

systemd - bin

systemctl

For dealing with unit files, services, targets etc.

systemd - bin

systemctl

- The know-how. man systemctl
- What’s happening on my system? systemctl status
- Show me loaded services. systemctl -t service
- Show me all unit files. systemctl list-unit-files
- Set vendor's default (enable / disable). systemctl preset docker
- What’s my system’s current state? systemctl is-system-running
- Which units are in failed state? systemctl --failed
- Please, show me dependencies of an *httpd*? systemctl list-dependencies httpd
- Enable the service. systemctl enable --now httpd
- Disable the service. systemctl disable --now httpd
- Show the service. systemctl show httpd

systemd - bin

analyzing boot process

systemd - bin

analyzing boot process

With systemd analyzing the boot process looks quite interesting (demo, pictures!):

systemd - bin

analyzing boot process

```
systemd-analyze time
systemd-analyze blame
systemd-analyze plot
systemd-analyze dump
systemd-analyze verify system.slice
systemd-analyze dot ‘docker.*’ | dot -Tsvg > docker.svg
systemd-analyze dot --to-pattern=’*.target’ --from-pattern=’*.target’ | dot -Tsvg > targets.svg
```

systemd - bin

coredumps

systemd - bin

coredumps

With systemd we may generate, browse and view any historical coredumps

systemd - bin

coredumps

```
coredumpctl dump
coredumpctl dump docker
coredumpctl dump _PID=666
    (journalctl general predicates; man systemd.directives)
coredumpctl dump /usr/sbin/httpd
coredumpctl gdb _PID=666
```

systemd - bin

cgroups

cgroups

```
systemd-cgtop
systemd-cgtop -d 5 -n 3
systemd-cgtop /system.slice/suditd.service
```

systemd - bin

killing processes

systemd - bin

killing processes

Actually units may have policy about how to be killed in a proper way by setting KillMode= in a unit file

systemd - bin

killing services

```
man systemd.kill
systemctl kill docker.service
```

systemd - bin

FHS!

systemd - bin

FHS!

Actually systemd takes care about FHS
You may easily see what’s the purpose of specific directories

systemd - bin

FHS!

```
man file-hierarchy
systemd-path*
systemd-path temporary
systemd-path system-state-logs
```

*do not confuse with a systemd.path (path activation)

systemd - bin

detecting virtualization

systemd - bin

detecting virtualization

systemd will tell you if you are on a bare, VM, container of chroot

detecting virtualization

```
man systemd-detect-virt
systemd-detect-virt
```

systemd - bin

DNS resolving

systemd - bin

DNS resolving

systemd provides resolver service which may be queried against DNS entries

DNS resolving

```
man systemd-resolve
systemd-resolve www.google.com
systemd-resolve -t mx google.com
```

systemd - bin

finger

systemd - bin

finger

```
man loginctl
loginctl list-users
loginctl list-sessions
loginctl user-status
loginctl session-status
```

systemd - bin

systemd time management

systemd time management

```
man timedatectl
timedatectl list-timezones
timedatectl set-time 2016-11-30 11:12:13
timedatectl status systemd-timesyncd.service
timedatectl set-ntp true
```

systemd - bin

inhibit

systemd - bin

inhibit

systemd provides a way to make sure that your hardware will not sleep / hibernate / poweroff during execution of given command

systemd - bin

inhibit

```
man systemd-inhibit
systemd-inhibit something
```

systemd - bin

d-bus

systemd - bin

d-bus

systemd uses a d-bus for an InterProcess Communication (IPC)

systemd - bin

d-bus

- see the current state of processes registered in the d-bus:

```
busctl
buctl tree
sudo busctl capture > test.pcap
```

demo?

```
busctl --user
```

systemd - bin

process confinement

systemd - bin

process confinement

you may run any process under systemd / cgroups confinement

systemd - bin

process confinement

```
systemd-run env
systemd-run -p BlockIOWeight=10 update

  -    Timers:
       `date; systemd-run --on-active=30 --timer-property=AccuracySec=100ms \
/bin/touch/tmp/foo`
       journalctl -b -u run-71.timer

systemd-run --scope -p BlockIOWeight=10 --user tmux
       tmuxls
```

5 minutes break?

services & unit files

services & unit files

imperativeness vs declarativeness

services & unit files

imperativeness vs declarativeness

compare httpd init scripts vs unit file

services & unit files

types of units

services & unit files

types of units

service
target
path
timer
socket
...

types of units

```
man systemd.(device | mount  | automount | swap | slice | scope)
```

services & unit files

runlevels & targets

services & unit files

runlevels & targets

we had *runlevels* before *systemd* (remember? chkconfig && 2,3,5?)
now we have units of type target and think of targets as unit aggregators / groups

runlevels & targets

- The know-how. man systemd.target
- Display possible targets. systemctl list-units --type=target
- Which is default (current runlevel)? systemctl get-default
- Change the default target? systemctl isolate [target] / AllowIsolate=

systemctl isolate multi-user.target (or) systemctl isolate runlevel3.target
systemctl isolate graphical.target (or) systemctl isolate runlevel5.target

services & unit files

services dependencies

services & unit files

services dependencies

Requires, Requisite, Wants, BindsTo, PartOf, Conflict, Before, Afetr, OnFailure, PropagatesReloadTo, ReloadPropagatedFrom, StopWhenUnneeded, DefaultDependencies, WantedBy, RequiredBy, Also

services & unit files

starting after installation

services & unit files

starting after installation

systemctl mask

services & unit files

starting after installation

systemctl mask
Debian, Ubuntu & autostart

services & unit files

starting after installation

systemctl mask
Debian, Ubuntu & autostart
http://maciej.lasyk.info/2016/Nov/29/systemd-mask/

services & unit files

cronjobs / timers

cronjobs / timers

```
[Unit]
Description=Run script every hour

[Timer]
OnBootSec=10min
OnUnitActiveSec=1h
Unit=script.service

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

services & unit files

socket activation

services & unit files

socket activation

ListenStream, ListenDatagram, ListenSequentialPacket, ListenFifo, ListenSpecial, ListenNetlink, ListenMessageQueue, ListenUSBFunction, SocketProtocol, BindToDevice, ...

socket activation

```
[Unit]
Description=Socket activation for simple systemd-notify app

[Socket]
ListenStream=1025

[Install]
WantedBy=sockets.target
```

services & unit files

cgroups control

services & unit files

cgroups control

CPUShares, CPUAccounting, MemoryAccounting, MemoryLimit, BlockIOAccounting, BlockIOWeight, BlockIOReadBandwidth, BlockIOWriteBandwidth

services & unit files

defining kill method

services & unit files

defining kill method

systemd-kill
KillMode, KillSignal, SendSIGHUP, SendSIGKILL

services & unit files

GUI?

services & unit files

GUI?

cockpit demo!

services & unit files

sysv import?

sysv import?

1. systemd maintains 99% backwards compatibility with LSB compatible initscripts and the exceptions are well documented
2. no need to convert
3. www.freedesktop.org/wiki/Software/systemd/Incompatibilities
4. 0pointer.de/blog/projects/systemd-for-admins-3.html

journal & logging

journal & logging

journald resolves security in syslog (authentication)

journal & logging

journald resolves security in syslog (authentication)

no more “disk is out of space” due to growing logs

journal & logging

journald resolves security in syslog (authentication)

no more “disk is out of space” due to growing logs

built-in anti ddos (rate limter)

basic filtering

`journalctl`

recently: `-e`
last 4 entries: `-n 4`
reverse: `-r`
kernel related: `-k`
since last boot: `-b`
no-paging: `--no-pager`
live tailing: `-f`

severity filtering

journalctl

logs severity: -p err
range: -p info..err

- emerg(0)
- alert(1)
- crit(2)
- err(3)
- warning(4)
- notice(5)
- info(6)
- debug(7)

output formatting

```
journalctl -o json
journalctl -o json-pretty
   -    short
   -    verbose
   -    export
   -    json
   -    cat
```

time filtering

```
man systemd.time
```

```
journalctl --since=”2016-08-01”
journalctl --until=”2016-09-01”
```

Timezone is default, local but may add a definition e.g. UTC

```
journalctl --since=”2016-08-01 07:00:00 UTC”
```

Additional settings:
today, yesterday, tomorrow, -1week, -1month, -20day

journal & logging

grepping

journal & logging

grepping

```
journalctl -b -u some.service --no-pager | grep -i ‘some_keyword’
```

journal & logging

managing disk space

journal & logging

managing disk space

persistent storage? mkdir /var/log/journal

managing disk space

persistent storage? mkdir /var/log/journal

- Show current disk usage: journalctl --disk-usage
- Truncate logs to given size: journalctl --vacuum-size=2.8GB
- Set logs retention: journalctl -vacuum-time=1years

- Define it in the configuration: man journald.conf

journal & logging

metadata

metadata

- show detailed metadata: `journalctl -o verbose`
    - `journalctl -F [TAB]`
    - `man systemd.directives`
- specific PID: `journalctl _PID=1 _PID=n`
    - `journalctl -F _SYSTEMD_UNIT`
    - `journalctl -SE[TAB]`
- filter by hostname: `journalctl _HOSTNAME=somehost`
    - `journalctl _UID=x _GID=y`
- add more contectual info: `journalctl -x`

journal & logging

pipelining stdout/err into journal

pipelining stdout/err into journal

- catch stdout and stderr: `systemd-cat cat /proc/loadavg`
- catch stdout only: `cat /proc/loadavg | systemd-cat`

HTTPD logs viewer

```
> dnf install systemd-journal-remote
> systemctl endble --now systemd-journal-gateway
```

http://localhost:19531/browse
http://localhost:19531/machine

man systemd-journal-gatewayd

journal & logging

sealing journal

sealing journal

FSS - Forwad Secure Sealing used by journald to ensure the integrity of the journal and to seal the logs cryptographically

https://eprint.iacr.org/2013/397.pdf

journal & logging

sealing journal

- check the integrity of the journal: `journalctl --verify`
- generate the keys: `journalctl --setup-keys`
- verify integrity w/FSS keys: `journalctl --verify-key [path_to_key] --verify`

journal & logging

journal & Python

journal & logging

journal & Python

another demo

5 minutes break?

nspawn containers

nspawn containers

very simple containers

nspawn containers

very simple containers

no daemon behind

no need to do anything with the storage or network

nspawn containers

very simple containers

no daemon behind

no need to do anything with the storage or network

just dnf / yum install

nspawn containers

installation

```
> dnf --releasever=25 --installroot=/var/lib/machines/f25 install systemd passwd dnf
fedora-release
> systemd-nspawn -D /var/lib/container/f25
> passwd
> cp /usr/lib/systemd/systemd-nspawn@.service
/etc/systemd/system/systemd-nspawn@f25.service
> systemctl enable --now systemd-nspawn@f25.service
```

sd-notify

sd-notify

even more demos...

#learningsystemd

man systemd.linux

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

http://0pointer/de/blog/projects (look for systemd*)

http://0pointer.de/blog/projects/the-biggest-myths.html

http://maciej.lasyk.info/tag/learning-systemd.html

Thanks, Q&A?

Maciej Lasyk

@docent-net
http://maciej.lasyk.info