Everyday systemd

Tom Most @ NBLUG
August 14, 2018
## Everyday systemd

<table>
<thead>
<tr>
<th>Overview</th>
<th>Example</th>
<th>Common Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>How does systemd model the world?</td>
<td>Running carrotbot.</td>
<td>Getting around a systemd machine.</td>
</tr>
</tbody>
</table>
What is a system manager?

Manages low-level details of the system across:

- Boot
- Runtime
- Shutdown

Responsibilities include:

- Hardware configuration
- Mounting filesystems
- Supervising services

And ordering the above.
Everything is a unit

- Service
- Socket
- Device
- Mount
- Automount
- Swap
- Target
- Path
- Timer
- Slice
- Scope

Units go in:

- /etc/systemd/system/*.system
- /run/systemd/system/*.system
- /lib/systemd/system/*.system

(Or user-specific directories. We won't discuss that here.)
A Simple Unit: hello.service

[Unit]
Description=My Service
Wants=network-online.target
After=network-online.target

[Service]
Type=simple
ExecStart=/bin/sh -c \\
  'while echo hello, world; do sleep 10; done'

[Install]
WantedBy=multi-user.target
Dependencies vs. ordering

[Unit]

Wants=network-online.target

“When you start this unit, be sure to start the network-online.target unit too”

After=network-online.target

“Only start this unit once network-online.target has started”
Enabling and disabling units

Enable a unit:

    systemctl enable --now hello.service

[Install] says what to do:

    WantedBy=multi-user.target

Is usually what you want.
Auto-generated units

- Services
  - Generated from /etc/init.d/

- Mount units
  - Generated from /etc/fstab and /etc/crypttab
  - Options provide more control:
    - x-systemd.requires=
    - Automount
    - Timeouts
  - See man pages:
    - systemd.mount(5)
    - systemd-fstab-generator(8)
    - systemd-cryptsetup-generator(8)
Common Tasks

Getting stuff done
What’s up?

```bash
# systemctl status
● jame
  State: running
  Jobs: 0 queued
  Failed: 0 units
  Since: Sun 2018-08-05 23:18:29 PDT; 1 weeks 1 days ago
  CGroup: /

  --init.scope
  ├── 1 /lib/systemd/systemd --system --deserialize 20
  │ 1048 /sbin/cgmanager -m name=systemd
  └─system.slice
      └─1008 /usr/sbin/thermald --no-daemon --dbus-enable
      └─1049 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-
      └─1123 /usr/sbin/ModemManager
      └─1016 /usr/sbin/cron -f
      └─311 /sbin/lvmetad -f
```
How’re ya doin’?

```
# systemctl status hello
● hello.service - My Service
   Loaded: loaded (/etc/systemd/system/hello.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2018-08-14 18:34:51 PDT; 7min ago
   Main PID: 1222 (sh)
      CGroup: /system.slice/hello.service
             └─1222 /bin/sh -c while echo hello, world; do sleep 10; done
            └─3651 sleep 10

Aug 14 18:40:51 omicron sh[1222]: hello, world
Aug 14 18:41:01 omicron sh[1222]: hello, world
Aug 14 18:41:11 omicron sh[1222]: hello, world
Aug 14 18:41:21 omicron sh[1222]: hello, world
Aug 14 18:41:31 omicron sh[1222]: hello, world
Aug 14 18:41:41 omicron sh[1222]: hello, world
Aug 14 18:41:51 omicron sh[1222]: hello, world
Aug 14 18:42:01 omicron sh[1222]: hello, world
Aug 14 18:42:11 omicron sh[1222]: hello, world
Aug 14 18:42:21 omicron sh[1222]: hello, world
```
### What’s broken?

```
$ systemctl --failed

<table>
<thead>
<tr>
<th>UNIT</th>
<th>LOAD</th>
<th>ACTIVE</th>
<th>SUB</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>mnt-Music.mount</td>
<td>not-found</td>
<td>failed</td>
<td>failed</td>
<td>mnt-Music.mount</td>
</tr>
<tr>
<td>networking.service</td>
<td>loaded</td>
<td>failed</td>
<td>failed</td>
<td>Raise network interfaces</td>
</tr>
</tbody>
</table>
```

**LOAD** = Reflects whether the unit definition was properly loaded.
**ACTIVE** = The high-level unit activation state, i.e. generalization of SUB.
**SUB** = The low-level unit activation state, values depend on unit type.

4 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
Tweaking units

Make small changes to upstream units:

# systemctl edit foo.service

Opens $EDITOR to create a file in “dropin” directory:

/etc/systemd/system/foo.service.d/

Don’t forget to systemctl daemon-reload!
What’s in a unit?

# systemctl cat hello.service
# /etc/systemd/system/hello.service
[Unit]
Description=My Service
Wants=network-online.target
After=network-online.target

[Service]
Type=simple
ExecStart=/bin/sh -c \\
  'while echo hello, world; do sleep 10; done'

[Install]
WantedBy=multi-user.target
Show me everything

# systemctl show hello.service
Restart=no
NotifyAccess=none
RestartUSec=100ms
TimeoutStartUSec=1min 30s
TimeoutStopUSec=1min 30s
RuntimeMaxUSec=infinity
WatchdogUSec=0
WatchdogTimestampMonotonic=0
FailureAction=none
PermissionsStartOnly=no
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
MainPID=0
ControlPID=0
FileDescriptorStoreMax=0
NFileDescriptorStore=0
StatusErrno=0
What have you been up to?

```
# journalctl -u hello.service
-- Logs begin at Tue 2018-08-14 18:34:27 PDT, end at Tue 2018-08-14 18:54:32 PDT.
--
Aug 14 18:34:51 omicron systemd[1]: Started My Service.
Aug 14 18:34:51 omicron sh[1222]: hello, world
Aug 14 18:35:01 omicron sh[1222]: hello, world
Aug 14 18:35:11 omicron sh[1222]: hello, world
Aug 14 18:35:21 omicron sh[1222]: hello, world
Aug 14 18:35:31 omicron sh[1222]: hello, world
Aug 14 18:35:41 omicron sh[1222]: hello, world
Aug 14 18:35:51 omicron sh[1222]: hello, world
Aug 14 18:36:01 omicron sh[1222]: hello, world
Aug 14 18:36:11 omicron sh[1222]: hello, world
Aug 14 18:36:21 omicron sh[1222]: hello, world
Aug 14 18:36:31 omicron sh[1222]: hello, world
Aug 14 18:36:41 omicron sh[1222]: hello, world
Aug 14 18:36:51 omicron sh[1222]: hello, world
Aug 14 18:37:01 omicron sh[1222]: hello, world
Aug 14 18:37:11 omicron sh[1222]: hello, world
```
Neat trick

Add to your crontab:

```
@daily journalctl -u hello --since '1 day ago' --priority notice --quiet
```

Get a daily e-mail with any errors.

(Does require the unit log to syslog or the journal.)
Documentation

Some notes

- Key manpages:
  - systemd.unit(5)
  - systemd.service(5)
  - systemctl(1)
- systemd for Administrators
  - Nice gradual introduction
Questions