systemd,the wave of the future

Don't let yourself get drowned by it

Mark Post
SUSE

Monday, August 4, 2014
Session 15565

#SHAREorg

SHARE in Pittsburgh 2014

Agenda

- What is systemd?
- What problem is systemd trying to fix? i.e., what was wrong with SysV init?
- systemd concepts
- Sample service unit files
- Backward compatibility
- I just got comfortable with SysV init. How do I do similar tasks with systemd?
- Questions?
- Additional material
  - Terminology, additional concepts
  - Links to more information

What is systemd?

- The new system startup and service manager for Linux, replacing the old Sys V init.
- According to the developers, systemd
  "provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, keeps track of processes using Linux cgroups, supports snapshotting and restoring of the system state, maintains mount and automount points and implements an elaborate transactional dependency-based service control logic."
  - It also promotes run-on sentences and excessive fervor in its adherents.
- It is intended to be compatible with SysV and LSB init scripts (but there have been some problems due to different design philosophies)
- Pretty much *everything* about it is different, which means a significant learning curve

What is systemd?

- A few things to keep in mind for both SLES12 and RHEL 7
  - A *lot* of things are different, just in the system startup area.
    - systemd versus SysV init
    - dracut versus mkinitrd
    - grub2 (Grand Unified Bootloader)
    - Plymouth (Graphical Boot Animation and Logger)
  - Trying to figure what you're seeing on the console belongs to which piece will be a new “learning experience (ouch).”
  - I'm not going to get into all that here. Talking about systemd alone could fill multiple session slots.
  - udev is now part of systemd. The RPM name is still just “udev” but the source comes from systemd.

What was wrong with SysV init?

- Depending on who you talk to, nothing or everything.
- There *are* a number of areas that could use improvement
  - It's slow
  - It's hard to parallelize
  - The concept of runlevels is rather coarse
  - Linux Standard Base (LSB) dependencies only do *part* of what most system administrators need
  - No automatic restart of services (outside of /etc/inittab)
  - No unified logging
  - No unified resource limit handling
  - Creating your own init scripts can be problematic

systemd concepts

- systemd introduces a lot of new concepts and terminology
    - Services are not started via scripts, but “unit files.”
    - Service starts, stops, and messages associated with those are logged in a journal, accessible via the “journalctl” command.
- systemd replaces the traditional init process
    - Comes with a bunch of helper services
    - Uses the DBus protocol
- You'll be typing the “systemctl” command a *lot*.
    - It's the primary way you interact with systemd.
    - You'll probably want to create an alias for it, such as “sc”.

systemd concepts

- One-stop shopping for boot, shutdown and power management
  - integrated logging
  - unified command line tools for almost everything
  - automatic restart of services
  - cgroups and security compartments for everything
  - multi-seat hosts
  - handling of ACPI power management events

systemd concepts

- Unit files describe targets (i.e. synchonization points) and services (what used to be init scripts)
  - Much of what the LSB standard used is modeled in unit files
  - Plus a few more, for instance with LSB you could never say “my service needs to be started before kdm”
- Runlevels are replaced by targets
  - Runlevel 0: poweroff.target
  - Runlevel 1: rescue.target
  - Runlevels 2-4: multi-user.target
  - Runlevel 5: graphical.target
  - Runlevel 6: reboot.target

Sample service unit file: sshd.service

/usr/lib/systemd/system/sshd.service

```
# This is a comment!
[Unit]
Description=OpenSSH Daemon
After=network.target

[Service]
EnvironmentFile=/etc/sysconfig/ssh
ExecStartPre=/usr/sbin/sshd-gen-keys-start
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=always

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

```
LSB Analogs/Equivalents
[Unit]
# Description: ...
# Required-Start: $network

[Service]
# <- All of these used to be
# <- open coded in the init script
# <-
# <-
# <-
Automated Restart: didn't exist previously

[Install]
# Default-Start: 3 5
```

Sample service unit file: cron.service

/usr/lib/systemd/system/cron.service

```
# This is a comment!
[Unit]
Description=Command Scheduler
After=ypbind.service nscd.service network.target
After=postfix.service sendmail.service exim.service

[Service]
ExecStart=/usr/sbin/cron -n
Restart=on-abort

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

Backward compatibility

- The systemd developers don't seem to believe in it.
- SUSE does, however
  - insserv, chkconfig and /sbin/service will still be supported
  - Old style commands (calling “rcfoobar”) will be redirected to systemctl automatically
  - LSB compatibility for targets like $network are still available
  - And of course init scripts are still supported
    - But you may find that you actually prefer unit files.

How do I do the things I used to do?

- First, make sure you have the systemd-sysvinit RPM installed. (It should be installed by default.)
  - This provides symbolic links for halt, telinit, reboot, runlevel, etc.
  - This will allow you to continue using the old commands while learning the new ones.
- Changing run levels
  - Old:
    ```
    telinit 1
    telinit S
    telinit 3
    ```
  - New:
    ```
    systemctl isolate runlevel1.target
    systemctl isolate rescue.target
    systemctl isolate default
    ```

How do I do the things I used to do?

- Check the default runlevel
  - Old: `runlevel`
  - New: `systemctl get-default`
- Set/change the default runlevel
  - Old: Use YaST, or edit `/etc/inittab`
  - New: `systemctl set-default --force multi-user.target`
- Enabling/disabling a service
  - Old:
    ```
    chkconfig <name> on
    chkconfig <name> off
    insserv [-r]
    ```
  - New:
    ```
    systemctl enable <name>.service
    systemctl disable <name>.service
    ```

How do I do the things I used to do?

- Starting/stopping a service
  - Old:
    ```
    rc<name> start
    rc<name> stop
    /etc/init.d/<name> start
    /etc/init.d/<name> stop
    service <name> start
    service <name> stop
    ```
  - New:
    ```
    systemctl enable <name>.service
    systemctl disable <name>.service
    ```

To kill all the processes in the cgroup of a service:

```
systemctl kill <name>.service
```

How do I do the things I used to do?

- Check the status of a service
  - Old:
    ```
    rc<name> status
    /etc/init.d/<name> status
    service <name> status
    ```
  - New: `systemctl status <name>.service`

    To check the status of *all* services
    ```
    systemctl
    ```
- Re-read /etc/inittab
  - Old:
    ```
    telinit q
    telinit Q
    ```
  - New:
    ```
    systemctl daemon-reload
    systemctl reload “*”
    ```

How do I do the things I used to do?

- Re-spawn/re-execute init
  - Old: **`telinit u`**
    `telinit U`
  - New: **`systemctl daemon-reexec`**
- Show the running process “tree”
  - Old: **`pstree`** (part of the psmisc package)
  - New: **`systemd-cgls`**

How do I do the things I used to do?

- Shutting down
  - Old: `halt`

    ```
    halt
    shutdown -h
    telinit 0
    ```
  - New: `systemctl halt`
- Rebooting
  - Old:

    ```
    reboot
    shutdown -r
    telinit 6
    ```
  - New: `systemctl reboot`

File locations

- RPM-provided unit files
  `/usr/lib/systemd/system`
- User overrides to unit files
  `/etc/systemd/system/`
  `/etc/systemd/system/<name>.service.d`
- `/etc/systemd/system.conf`
  At the moment, all defaults (as comments)
- `/etc/systemd/user.conf`
  At the moment, all defaults (as comments)
- Don't bother looking for `/etc/inittab`. It doesn't exist any longer.

Questions?

Terminology

- Unit file - Encodes information about things such as a service, socket, device, mount, automount, target, snapshot, etc.
- Target - A unit configuration file whose name ends in ".target" encodes information about a target unit of systemd, which is used for grouping units and as well-known synchronization points during start-up.
- Slice - A concept for hierarchially managing resources of a group of processes.
- Seat - The set of hardware available at one work place (graphics card, keyboard, mouse, usb devices)
  - This doesn't seem to be relevant to System z

Terminology

- Session - A session is created once a user is logged on, using a specific seat
  - Only one session can be active per seat
  - Default seat (for Linux consoles) is seat0
  - Hardware is assigned to seats
    - This replaces ConsoleKit

Additional concepts

- Cgroups (control groups) for *everything*
  - systemd puts each service and each session into a separate cgroup
    - Sessions also get assigned an audit ID matching their cgroup ID
  - You can restrict these cgroups in all the ways the kernel supports
    - IO bandwidth, memory or CPU consumption, etc

Additional concepts

- Improve Security for Everything
  - Restrict services and sessions using namespaces
    - Linux kernel namespaces are the technology underlying Linux containers
    - blacklist directories
    - require private /tmp directory
    - whitelist devices to which access is granted
  - Specify user/group to run as
  - Assign Linux kernel capabilities (CAP_FOOBAR)
  - Set ulimit values

Additional concepts

- Overriding defaults for a service
  - With SysV init, if you want to do anything more advanced that enable/disable a service, you need to edit the init script
    - Obviously this doesn't go well with package updates
- systemd supports this
  - To modify settings for <name>.service
    - Create /etc/systemd/system/<name>.service.d
    - Create a file named mysettings.conf in there:

```
[Service]
InaccessibleDirectories=/precious
MemoryLimit=1G
```

    - Only the settings you want to add/override need to be in the file

Additional concepts

- Session handling
  - List all sessions
    `loginctl [list-sessions]`
  - Show session details
    `loginctl session-status <session-number>`
  - Forcefully terminate a session
    `loginctl kill-session|kill-user|terminateseat <name>`

Additional concepts

- See how your configuration differs from the vendor defaults
    - `systemd-delta`
- Analyze boot times and bottlenecks
    - `systemd-analyze`

Links to more information

- The various man pages on an installed system.
    - They're pretty verbose, but as do most man pages, they tend to assume some background you may or may not have. See below for where to find that background.
- http://www.freedesktop.org/wiki/Software/systemd/
- http://0pointer.de/blog/

SUSE®

We adapt. You succeed.

Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.