Linux Service Initialization

Services and

Init Daemons

John Karr
brainbuz@brainbuz.org
Philadelphia Linux User's Group Presentation
2012 February 1

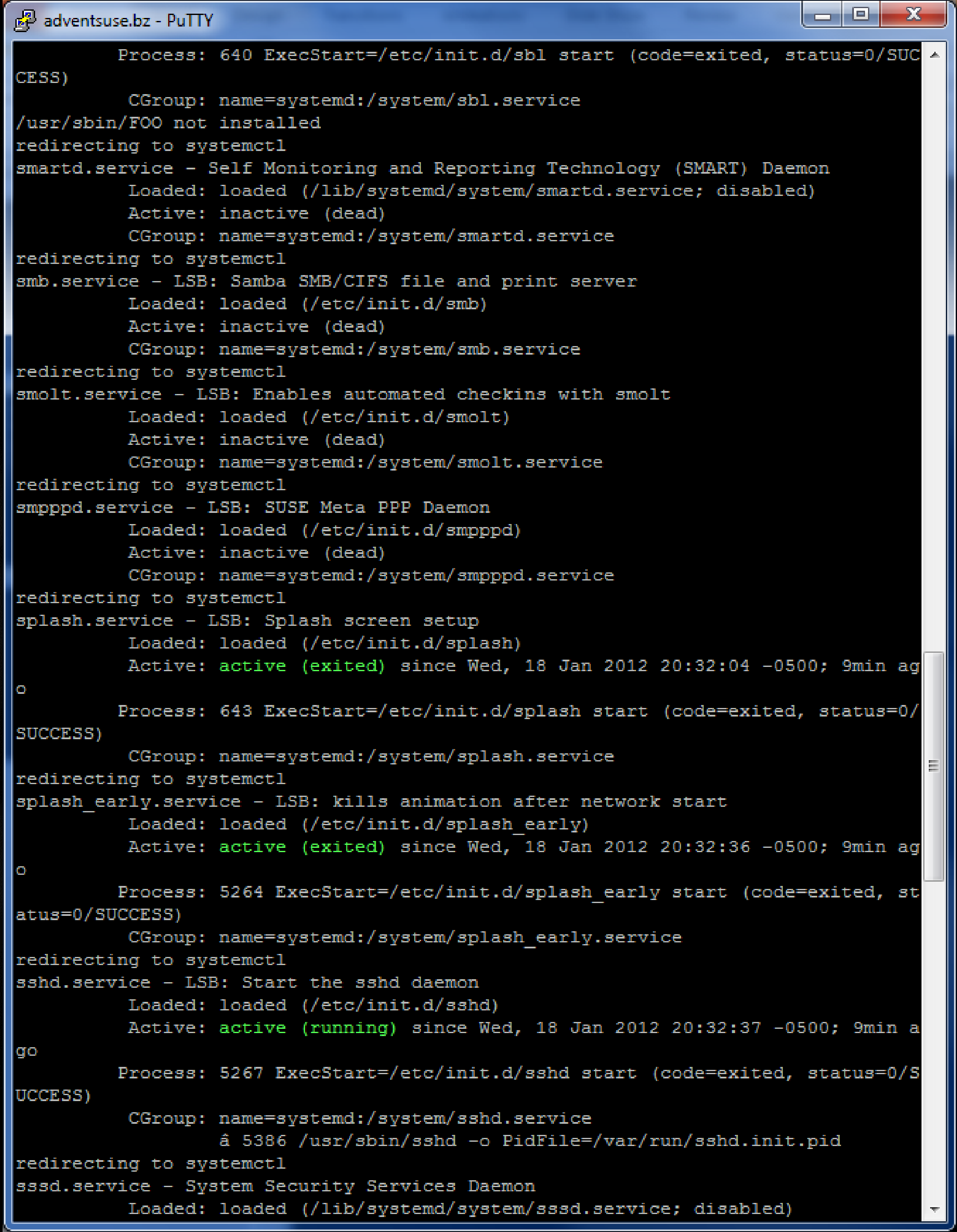
The Service Command

Output of

service –status-all

on OpenSUSE

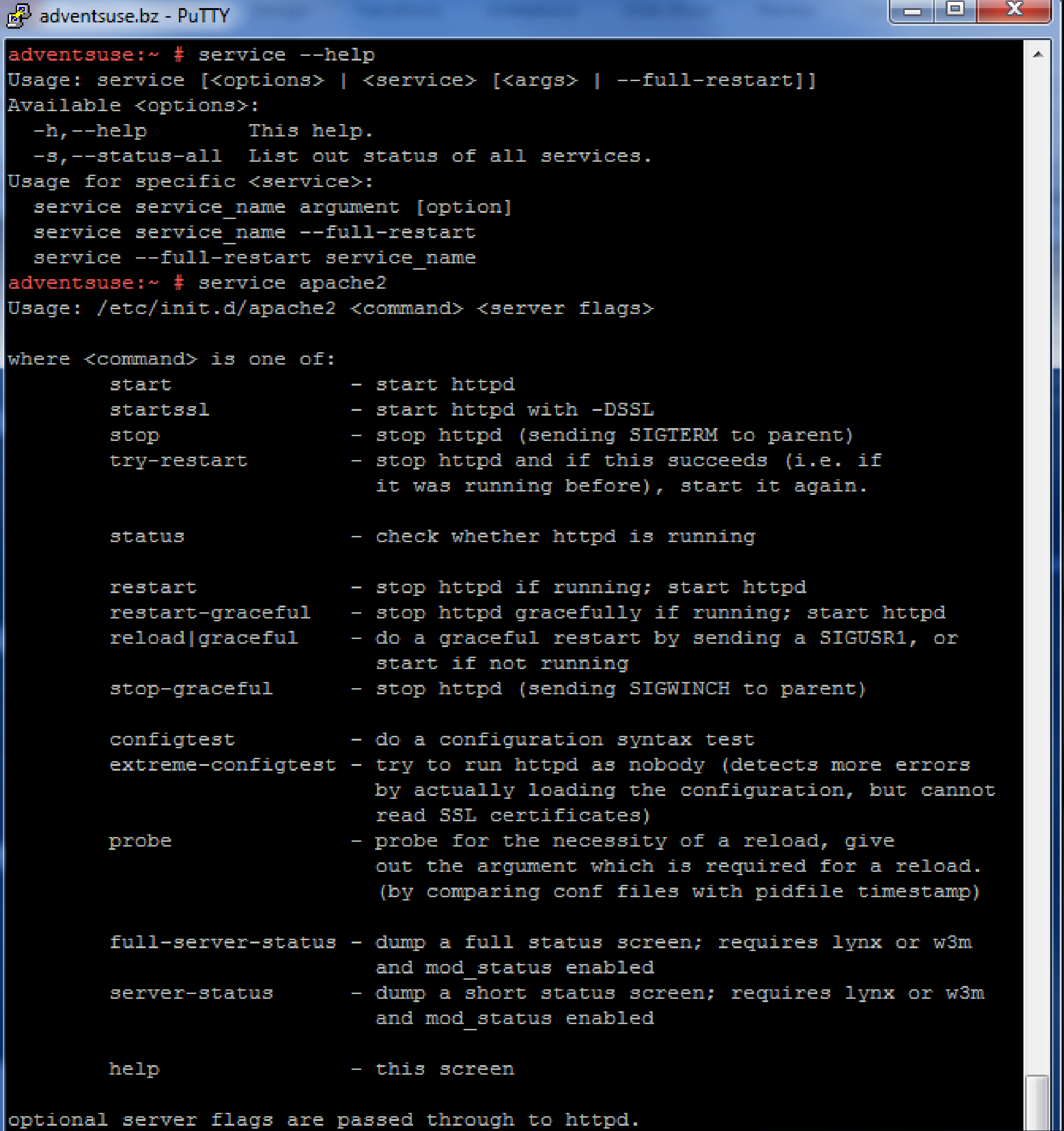
The Service Command
Is also used to start,
stop and restart
Daemons.



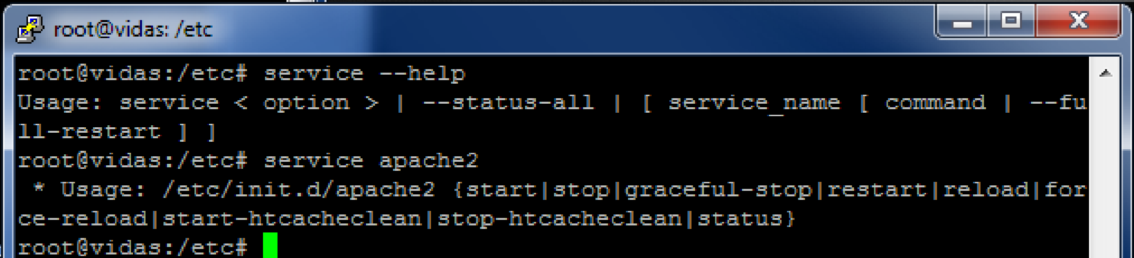
```
         Process: 640 ExecStart=/etc/init.d/sbl start (code=exited, status=0/SUC
CESS)
          CGroup: name=systemd:/system/sbl.service
/usr/sbin/FOO not installed
redirecting to systemctl
smartd.service - Self Monitoring and Reporting Technology (SMART) Daemon
          Loaded: loaded (/lib/systemd/system/smartd.service; disabled)
          Active: inactive (dead)
          CGroup: name=systemd:/system/smartd.service
redirecting to systemctl
smb.service - LSB: Samba SMB/CIFS file and print server
          Loaded: loaded (/etc/init.d/smb)
          Active: inactive (dead)
          CGroup: name=systemd:/system/smb.service
redirecting to systemctl
smolt.service - LSB: Enables automated checkins with smolt
          Loaded: loaded (/etc/init.d/smolt)
          Active: inactive (dead)
          CGroup: name=systemd:/system/smolt.service
redirecting to systemctl
smpppd.service - LSB: SUSE Meta PPP Daemon
          Loaded: loaded (/etc/init.d/smpppd)
          Active: inactive (dead)
          CGroup: name=systemd:/system/smpppd.service
redirecting to systemctl
splash.service - LSB: Splash screen setup
          Loaded: loaded (/etc/init.d/splash)
          Active: active (exited) since Wed, 18 Jan 2012 20:32:04 -0500; 9min ag
o
         Process: 643 ExecStart=/etc/init.d/splash start (code=exited, status=0/
SUCCESS)
          CGroup: name=systemd:/system/splash.service
redirecting to systemctl
splash_early.service - LSB: kills animation after network start
          Loaded: loaded (/etc/init.d/splash_early)
          Active: active (exited) since Wed, 18 Jan 2012 20:32:36 -0500; 9min ag
o
         Process: 5264 ExecStart=/etc/init.d/splash_early start (code=exited, st
atus=0/SUCCESS)
          CGroup: name=systemd:/system/splash_early.service
redirecting to systemctl
sshd.service - LSB: Start the sshd daemon
          Loaded: loaded (/etc/init.d/sshd)
          Active: active (running) since Wed, 18 Jan 2012 20:32:37 -0500; 9min a
go
         Process: 5267 ExecStart=/etc/init.d/sshd start (code=exited, status=0/S
UCCESS)
          CGroup: name=systemd:/system/sshd.service
                  â 5386 /usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
redirecting to systemctl
sssd.service - System Security Services Daemon
          Loaded: loaded (/lib/systemd/system/sssd.service; disabled)
```

The Service Command

We can see here that the service command is different on OpenSUSE and DEBIAN based distributions, but both implement a similar feature set.



```
adventsuse:~ # service --help
Usage: service [<options> | <service> [<args> | --full-restart]]
Available <options>:
  -h,--help        This help.
  -s,--status-all  List out status of all services.
Usage for specific <service>:
  service service_name argument [option]
  service service_name --full-restart
  service --full-restart service_name
adventsuse:~ # service apache2
Usage: /etc/init.d/apache2 <command> <server flags>

where <command> is one of:
        start              - start httpd
        startssl           - start httpd with -DSSL
        stop               - stop httpd (sending SIGTERM to parent)
        try-restart        - stop httpd and if this succeeds (i.e. if
                             it was running before), start it again.

        status             - check whether httpd is running

        restart            - stop httpd if running; start httpd
        restart-graceful   - stop httpd gracefully if running; start httpd
        reload|graceful    - do a graceful restart by sending a SIGUSR1, or
                             start if not running
        stop-graceful      - stop httpd (sending SIGWINCH to parent)

        configtest         - do a configuration syntax test
        extreme-configtest - try to run httpd as nobody (detects more errors
                             by actually loading the configuration, but cannot
                             read SSL certificates)
        probe              - probe for the necessity of a reload, give
                             out the argument which is required for a reload.
                             (by comparing conf files with pidfile timestamp)

        full-server-status - dump a full status screen; requires lynx or w3m
                             and mod_status enabled
        server-status      - dump a short status screen; requires lynx or w3m
                             and mod_status enabled

        help               - this screen

optional server flags are passed through to httpd.

adventsuse:~ #
```



```
root@vidas:/etc# service --help
Usage: service < option > | --status-all | [ service_name [ command | --fu
ll-restart ] ]
root@vidas:/etc# service apache2
 * Usage: /etc/init.d/apache2 {start|stop|graceful-stop|restart|reload|for
ce-reload|start-htcacheclean|stop-htcacheclean|status}
root@vidas:/etc#
```

The System V Init Script

AT&T released System V in 1983.

From that time the initialization process has been pretty much the same.

The service command, which borrows its terminology from Windows is a relatively recent innovation in System V. It is now used on most linuxes.

The Init Process

Unix systems have a number of arbitrary run levels. Traditionally 7 (0-6).

The system starts at runlevel 1.

The /sbin/init daemon is the first user process started by the kernel and gets PID 1.

Linux Distributions through Linux Standards Base (LSB) agreed on what the runlevels should mean.

LSB Runlevels

Runlevel	Description
0	Halt
1	Single-User mode
2	Multi-user mode console logins only (without networking)
3	Multi-User mode
4	Not used/User-definable
5	Multi-User mode, with display manager as well as console logins (X11)
6	Reboot

The Startup scripts

On DEBIAN scripts for each runlevel are in:

rc0.d rc1.d rc2.d rc3.d rc4.d rc5.d rc6.d rcS.d

Init.d contains scripts for multiple runlevels that are linked by the various runlevels.

Some commands available on debian for managing runlevels

invoke-rc.d (8) - executes System-V style init script actions

runlevel (8) - find the previous and current system runlevel.

update-rc.d (8) - install and remove System-V style init script links

RedHat and SUSE have chkconfig, and the rc? Scripts are beneath rc.d.

Problems with System V Init

A small debian init script is 1.5K, the Debian apache2 init script (from squeeze) is 7,621 bytes. Scripts this large are difficult to write and maintain.

Not Dependency Based. Even when you have professionals to write your init scripts (ie a major distribution), getting them in order is still a challenge. The numbering system used does not support the number of potential daemons facing a modern distribution.

System V is strictly sequential, allowing processes to initialize in parallel can speed boot time.

The Debian (Non) Solution

About 10 years ago the Debian project decided to use duct tape and strict adherence to the rules that were in place. Debian’s init structure is less broken than many other distributions.

What Debian and Gentoo and many smaller distributions have done, is wait for someone else to solve the problem.

Ironically by not being out in front Debian is now stuck there.

The Ubuntu Upstart Solution

In 2006 Ubuntu released their solution, called upstart.

Upstart integrates well with the existing Debian init script structure so that Upstream Debian packages' init scripts can be used and no one has to convert the more complicated existing init scripts.

With upstart you define a .conf file for a service and activate it by symbolically linking the global upstart script.

Create an Upstart Service

Contents of starman_testapp.conf (in /etc/init).

```
description “Starman testapp"
author      "Based on a script by Steve Langasek <steve.langasek@ubuntu.com>"
# Copy this script with the name of the actual script to run embedded
# place your copy at /etc/init/jobname.conf
start on filesystem or runlevel [2345]
stop on runlevel [!2345]
respawn limit 10 5
umask 022
expect fork
exec /bin/starman --daemonize  --l localhost:5000 /var/www/TestApp/testapp.psgi
```

The above configuration file is what is necessary to use starman (a Perl specific replacement for fcgi) to serve a Perl application, testapp.

Then cd to /etc/init.d and type:

```
ln -s /lib/init/upstart-job starman_testapp
```

That’s it, starman_testapp will start with the system, and be managed through the service command.

Why Hasn’t the World Already Switched to Upstart?

Making our upstart job was so easy and painless.

So why doesn’t everyone use it? Momentum was building for upstart, RedHat Enterprise had adopted. But then…

Lennart Poettering (author of PulseAudio) didn’t like Upstart. More accurately, he liked it a lot, but didn’t like how it dealt with process dependencies and the way Canonical was running the project.

In April 2010 Lennart Poettering released SystemD.

RedHat, SUSE, and Mandriva have already switched their community editions to SystemD, and it is on the roadmap for their Enterprise editions.

Gentoo and Arch are experimenting with it.

Debian has both upstart and systemd in testing.

SystemD is even better than Upstart

Create Just ONE File:

/etc/systemd/system/starman_testapp.service

```
[Unit]
Description=Starman TestApp
After=syslog.target

[Service]
Type=forking
ExecStart=/usr/bin/starman \
   --daemonize -l :5000 \
   --pid /var/run/starman_testapp.pid \
   /Path_To/TestApp/testapp.psgi
Restart=always

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

Then type: sudo systemctl enable starman_testapp.service

Comparison

Upstart does not support setting a pid file, this is required by SystemD.

SystemD seems to have more options than upstart and to be better documented.

Lennart Poettering has blogged extensively on the advantages to SystemD over Upstart. I have not found much in the way of response from the upstart community, which I take as a concession.

My conclusion from having experimented with SystemD on OpenSUSE and worked with Upstart on Ubuntu is that SystemD is better than Upstart, but not monumentally better, but definitely the init daemon I would choose.

References

Wikipedia

The Ubuntu Upstart Homepage: http://upstart.ubuntu.com

Lennart Poettering’s blog: http://opointer.de/

SystemD homepage: http://www.freedesktop.org/wiki/Software/systemd

Manpages.ubuntu.com does not have the entry for upstart, but the manpage on a system redirects to a sparse entry for init.

Fedora has a lengthy SystemD page: https://fedoraproject.org/wiki/Systemd#systemd_documentation

The numerous SystemD manpages are available at:

http://opointer.de/public/systemd-man/

An exhaustive Comparison of inits from 2011:

http://opointer.de/blog/projects/why.html