Demystifying Systemd

Ben Breard, RHCA Solutions Architect, Red Hat bbreard@redhat.com



Agenda

- Systemd functionality
- Coming to terms
- Learning the basics
- More advanced topics
- Learning the journal
- Available resources



Systemd is more than a SysVinit replacement



Systemd is a system and service manager



Systemd Overview

- Controls "units" rather than just daemons
- Handles dependency between units.
- Tracks processes with service information
 - Services are owned by a cgroup.
 - Simple to configure "SLAs" based on CPU, Memory, and IO.
- Properly kill daemons
- Minimal boot times
- Debuggability no early boot messages are lost
- Easy to learn and backwards compatible.



Closer look at Units



Systemd - Units

- Naming convention is: name.type
 - httpd.service, sshd.socket, or dev-hugepages.mount
- **Service** Describe a daemon's type, execution, environment, and how it's monitored.
- **Socket** Endpoint for interprocess communication. File, network, or Unix sockets.
- **Target** Logical grouping of units. Replacement for runlevels.
- **Device** Automatically created by the kernel. Can be provided to services as dependents.
- Mounts, automounts, swap Monitor the mounting/unmounting of file systems.



Systemd – Units Continued

- **Snapshots** save the state of units useful for testing
- **Timers** Timer-based activation
- **Paths** Uses inotify to monitor a path
- **Slices** cgroup hierarchy for resource management.
- Scopes Organizational units that groups services' worker processes.



Systemd – Dependency Resolution

- Example:
 - Wait for block device
 - Check file system for device
 - Mount file system
- nfs-lock.service:
 - Requires=rpcbind.service network.target
 - After=network.target named.service rpcbind.service
 - Before=remote-fs-pre.target



That's all greatbut



Replace Init scripts!? Are you crazy?!



We're not crazy, I promise

- SysVinit had a good run, but leaves a lot to be desired.
- Often we work around init more than we realize.
 - One-node clusters
 - Daemon Monitoring with utilities such as monit
 - rc.local hacks
 - Tweaking symlinks under /etc/rc.d/rcX.d/S* to effect execution order
- Systemd encourages better standardization across distributions
 - LSB helped in this effort, but.....
 - Distribution standards benefit us all.



Fine, but isn't this just change for change's sake?



Not Really

- Systemd enables much "smarter" and easier to manage systems.
- PID 1 now handles dependency resolution.
 - No more adding things like `sleep 60; service [daemon] restart` to rc.local
- Services can be configured to autospawn and respawn
- Cgroup integration makes cgroups much easier to leverage.
- Most of us like Init because it's familiar and well understood.
- Systemd is simple to learn, and is easier for noobs



...but I just got used to Upstart in RHEL6.



...well, remember [deprecated technology]

- One of the best things about open source is that the *best* technology wins.
- Albeit, it can be frustrating to keep up, but comfort should not hinder innovation
- Upstart was a huge step forward from SysVinit, and was a great addition in RHEL 6.
- Upstart added the ability to respawn services and enabled some parallelization at boot.
- The downside is it failed to handle dependencies, and left it to the user/maintainer.
- Systemd solves that problem and many others.



....but I love System-V init scripts!!!



You're in luck!

- systemd maintains 99% backwards compatibility with initscripts and the exceptions are well documented.
- While we do encourage everyone to convert legacy scripts to service unit files, it's not a requirement.
 - ***hint: we'll show you how to do this in a few minutes.
- Incompatibilities are listed here: http://www.freedesktop.org/wiki/Software/systemd/Incompatibilities/
- Converting SysV Init Scripts: http://0pointer.de/blog/projects/systemd-for-admins-3.html



Isn't systemd just about fast boot times? I don't care about that on my servers!



You sure about that?

- Lennart Poettering says that "Fast booting isn't the goal of systemd, it's a result of a well designed system."
- As virt/cloud demand continues, the desire for lightweight, reliable/resilient, and *fast* images grows.
 - A stripped down image can boot in \sim 2 seconds.
 - Less CPU cycles burned during the boot process
 - Important for highly dense and dynamic environments.
 - Even more important for containers.





I don't like change. It makes me uncomfortable.

-Alf (R.I.P.)



Dude, seriously!?

Change is constant. Embrace rather than resist.



The Basics: Managing Services



Managing Services – Unit files

Via Init:

Init scripts are stored in /etc/init.d & called from /etc/rc*

Via systemd:

Maintainer files: /usr/lib/systemd/system/

User modifications: /etc/systemd/system/

Note unit files under /etc/ will take precedence over /usr



Managing Services – Start/Stop

Via Init:

\$ service httpd {start,stop,restart,reload}

Via systemctl:

\$ systemctl {start,stop,restart,reload} httpd.service



Managing Services – Start/Stop

Note that:

- systemctl places the "action" before the service name.
- If a unit isn't specified, .service is assumed.
 - systemctl start httpd == systemctl start httpd.service
- Multiple services can be passed in one command.
 - systemctl start httpd mariadb
- Tab completion works great with systemctl
 - Install bash-completion



Via Init:

\$ service httpd status

Via systemctl:

\$ systemctl status httpd.service



```
root@host158:~
File Edit View Search Terminal Help
[root@host158 ~]# systemctl status httpd
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
   Active: active (running) since Fri 2013-08-09 09:22:25 CDT; 12s ago
 Process: 890 ExecStop=/usr/sbin/httpd $0PTIONS -k graceful-stop (code=exited, status
=0/SUCCESS)
 Main PID: 893 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/sec"
   CGroup: name=systemd:/system/httpd.service
           -893 /usr/sbin/httpd -DFOREGROUND
            -894 /usr/sbin/httpd -DFOREGROUND
            -895 /usr/sbin/httpd -DFOREGROUND
            -896 /usr/sbin/httpd -DFOREGROUND
            -897 /usr/sbin/httpd -DFOREGROUND
            -898 /usr/sbin/httpd -DFOREGROUND
Aug 09 09:22:23 host158.local systemd[1]: Starting The Apache HTTP Server...
Aug 09 09:22:25 host158.local systemd[1]: Started The Apache HTTP Server.
[root@host158 ~]#
```



• That's a little more helpful than:

root@host145:~	-	×
File Edit View Search Terminal Help		
[root@host145 ~]# service httpd status httpd (pid 1433) is running [root@host145 ~]#		



- List loaded services:
 - systemctl -t service
- List installed services:
 - systemctl list-unit-files -t service (similar to chkconfig --list)
- View state:
 - systemctl --state failed

 tip systemctl can connect to remote hosts over SSH using "-H"



Managing Services – Enable/Disable

Via Init:

\$ chkconfig httpd {on,off}

Via systemctl:

\$ systemctl {enable, disable, mask, unmask} httpd.service

mask - "This will link these units to /dev/null, making it impossible to start them. This is a stronger version of disable, since it prohibits all kinds of activation of the unit, including manual activation. Use this option with care."



Runlevels







Runlevels == Targets

- "Runlevels" are exposed via target units
- /etc/inittab is no longer used
- Target names are more relevant:
 - multi-user.target vs. runlevel3
 - graphical.target vs. runlevel5
- View the default target via: `systemctl get-default`
- Set the default target via: `systemctl set-default [target]`
- Change at run-time via: `systemctl isolate [target]`
- Change at boot time by appending systemd.unit=[target] to the kernel
 - Rescue mode: append single, s, S, or 1
 - Emergency (similar to init=/bin/bash): append -b or emergency



Runlevel Names

Runlevel	Systemd Target	Description
0	poweroff.target, runlevel0.target	System halt
1	rescue.target, runlevel1.target	Single user mode
3 (2,4)	multi-user.target, runlevel3.target	Multi-user, non graphical
5	graphical.target, runlevel5.target	Multi-user, graphical
6	reboot.target, runlevel6.target	System reboot

ls /lib/systemd/system/runlevel*target -l

lrwxrwxr. 1 root root 15 Jul 3 21:37 /lib/systemd/system/runlevel0.target -> poweroff.target lrwxrwxrwx. 1 root root 13 Jul 3 21:37 /lib/systemd/system/runlevel1.target -> rescue.target lrwxrwxrwx. 1 root root 17 Jul 3 21:37 /lib/systemd/system/runlevel2.target -> multi-user.target lrwxrwxrwx. 1 root root 17 Jul 3 21:37 /lib/systemd/system/runlevel3.target -> multi-user.target lrwxrwxrwx. 1 root root 17 Jul 3 21:37 /lib/systemd/system/runlevel4.target -> multi-user.target lrwxrwxrwx. 1 root root 16 Jul 3 21:37 /lib/systemd/system/runlevel4.target -> multi-user.target lrwxrwxrwx. 1 root root 16 Jul 3 21:37 /lib/systemd/system/runlevel5.target -> graphical.target lrwxrwxrwx. 1 root root 13 Jul 3 21:37 /lib/systemd/system/runlevel6.target -> reboot.target



getty


getty

- Append: console=ttyS0
 - Will enable first detected serial port
- Simply start additional getty's via:
 - systemctl start serial-getty@USB0.service
 - Started using template file: /usr/lib/systemd/system/serialgetty@.service
- To customize serial device configuration:
 - cp /usr/lib/systemd/system/serial-getty@.service /etc/systemd/system/serial-getty@ttyS2.service
 - Edit config
 - systemctl enable serial-getty@ttyS2.service
 - systemctl start serial-getty@ttyS2.service

http://0pointer.de/blog/projects/serial-console.html



Troubleshooting the Boot Process



Booting

- Boot process is too fast to watch
- Interactive boot append: systemd.confirm_spawn=1
- /var/log/boot.log still works the same
- Enable debugging from grub by appending:
 - debug systemd.log_target=kmsg log_buf_len=1M
 - Or send dbug info to a serial console:
 - debug systemd.log_target=console console=ttyS0
- Enable early boot shell on tty9
 - systemctl enable debug-shell.service
 - In -s /usr/lib/systemd/system/debug-shell.service \ /etc/systemd/system/sysinit.target.wants/
- systemctl list-jobs
 http://freedesktop.org/wiki/Software/systemd/Debugging/



Booting

- rc.local
 - touch /etc/rc.d/rc.local ; chmod +x /etc/rc.d/rc.local
 - Don't forget to add #!/bin/bash
- systemd-analyze
 - Use 'blame', 'plot', or 'critical-chain' for more details







- Unit files can be altered or extended by placing "dropins" under: /etc/systemd/system/foobar.service.d/*.conf
 - # cat /etc/systemd/system/httpd.service.d/50-httpd.conf
 - [Service] Restart=always StartLimitInterval=10 StartLimitBurst=5 StartLimitAction=reboot CPUShares=2048 Nice=-10 OOMScoreAdjust=-1000
- Changes are applied on top of maintainer unit files.



- Run `systemctl daemon-reload` after making changes to notify systemd
- Drop-ins will be shown from `systemctl status`

systemctl status httpd.service httpd.service - The Apache HTTP Server Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled) Drop-In: /etc/systemd/system/httpd.service.d ___50-httpd.conf



Customizing Service Unit Files – Tips!

- Changes to unit files under /usr/lib/systemd/system/ could be overwritten by updates. DON'T DO IT!
- /etc service files will take precedence over /usr
- Simply delete the drop-in to revert to defaults. Don't forget to run `systemctl daemon-reload`
- systemd-delta will show what is overridden and extended between /usr & /etc.



- Nice, CPUAffinity, CPUSchedulingPolicy, CPUSchedulingPriority, LimitCPU, IOSchedulingPriority, OOMScoreAdjust, IOSchedulingClass, etc
- For details see:
 - man 5 systemd.service
 - man 5 systemd.exec



Resource Management



Control Groups made simple

 Resource Management with cgroups can reduce application or VM contention and improve throughput and predictability





Resource Management

- View cgroup hierarchy via systemd-cgls
- View usage stats via systemd-cgtop (use for tuning)
- Default hierarchy
 - system.slice contains system services
 - user.slice contains user sessions
 - machine.slice contains virtual machines and containers
- Services can be promoted to their own slice if necessary.



Resource Management – systemd-cgls

```
File Edit View Search Terminal Help
 -1 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
 -machine.slice
   -machine-gemu\x2drhel7.scope
    └─17307 /usr/bin/qemu-system-x86_64 -machine accel=kvm -name rhel7 -S -machi
   machine-gemu\x2dEAP6.scope
    └─15290 /usr/bin/gemu-system-x86 64 -machine accel=kvm -name EAP6 -S -machin
  user.slice
   user-0.slice
    Luser@0.service
      -3289 /usr/lib/systemd/systemd --user
       -3299 (sd-pam)
   user-1000.slice
     -session-7.scope
      —13655 gdm-session-worker [pam/gdm-password]
       -13665 /usr/bin/gnome-keyring-daemon --daemonize --login
       —13710 qnome-session
      —13718 dbus-launch --sh-syntax --exit-with-session
      —13719 /bin/dbus-daemon --fork --print-pid 4 --print-address 6 --session
       —13784 /usr/libexec/gvfsd
       —13788 /usr/libexec//qvfsd-fuse /run/user/1000/qvfs -f -o big writes
       —13879 /usr/libexec/at-spi-bus-launcher
      —13883 /bin/dbus-daemon --confiq-file=/etc/at-spi2/accessibility.conf --n
       —13887 /usr/libexec/at-spi2-registryd --use-gnome-session
lines 1-23
```



Resource Management – configuration

- systemctl can configure and persist cgroup attributes
 - systemctl set-property httpd.service CPUShares=2048
- Add --runtime to **not** persist the settings:
 - systemctl set-property --runtime httpd.service \ CPUShares=2048
- Alternatively settings can be placed in unit files
 - [Service]
 - CPUShares=2048



Resource Management - CPU

- CPUAccounting=1 to enable
- CPUShares default is 1024.
- Increase to assign more CPU to a service
 - e.g. CPUShares=1600

https://www.kernel.org/doc/Documentation/scheduler/sched-design-CFS.txt



Resource Management - Memory

- MemoryAccounting=1 to enable
- Expose MemoryLimit and MemorySoftLimit
- Use K, M, G, T suffixes
 - MemoryLimit=1G

The idea behind soft limits is to allow control groups to use as much of the memory as needed, provided:

- a. There is no memory contention
- b. They do not exceed their hard limit

https://www.kernel.org/doc/Documentation/cgroups/memory.txt



Resource Management - BlkIO

- BlockIOAccounting=1
- BlockIOWeight= assigns an IO weight to a specific service (requires CFQ)
 - Similar to CPU shares
 - Default is 1000
 - Range 10 1000
 - Can be defined per device (or mount point)
- BlockIOReadBandwidth & BlockIOWriteBandwidth
 - BlockIOWriteBandwith=/var/log 5M

https://www.kernel.org/doc/Documentation/cgroups/blkio-controller.txt



Converting Init Scripts



But first, remember what init scripts look like?



/etc/init.d/httpd

```
. /etc/rc.d/init.d/functions
if [ -f /etc/sysconfig/httpd ]; then
    . /etc/svsconfig/httpd
fi
HTTPD LANG=${HTTPD LANG-"C"}
INITLOG ARGS=""
apachectl=/usr/sbin/apachectl
httpd=${HTTPD-/usr/sbin/httpd}
prog=httpd
pidfile=${PIDFILE-/var/run/httpd/httpd.pid}
lockfile=${LOCKFILE-/var/lock/subsys/httpd}
RETVAL=0
STOP TIMEOUT=${STOP TIMEOUT-10}
start() {
    echo -n $"Starting $prog: "
    LANG=$HTTPD LANG daemon --pidfile=${pidfile} $httpd $OPTIONS
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && touch ${lockfile}
    return $RETVAL
}
stop() {
    echo -n $"Stopping $prog: "
    killproc -p ${pidfile} -d ${STOP TIMEOUT} $httpd
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && rm -f ${lockfile} ${pidfile}
```

From RHEL 6.4; comments removed



Init – httpd continued

```
reload() {
  echo -n $"Reloading $prog: "
  if ! LANG=$HTTPD_LANG $httpd $OPTIONS -t >&/dev/null; then
    RETVAL=6
    echo $"not reloading due to configuration syntax error"
    failure $"not reloading $httpd due to configuration syntax error"
  else
    LSB=1 killproc -p ${pidfile} $httpd -HUP
    RETVAL=$?
    if [ $RETVAL -eq 7 ]; then
       failure $"httpd shutdown"
    fi
  fi
  echo
case "$1" in
 start)
    start
     ;;
 stop)
    stop
     ;;
 status)
    status -p ${pidfile} $httpd
    RETVAL=$?
     ;;
```



Init – httpd continued

```
restart)
    stop
    start
    ···
;;
condrestart|try-restart)
    if status -p ${pidfile} $httpd >&/dev/null; then
         stop
         start
    fi
    ;;
force-reload reload)
    reload
    •••
graceful|help|configtest|fullstatus)
    $apachectl $@
    RETVAL=$?
    ;;
*)
    echo $"Usage: $prog {start|stop|restart|condrestart|try-restart|force-reload|reload|status|fullstatus|graceful|help|
```

configtest}"

```
RETVAL=2
esac
```

exit \$RETVAL



Contrast that with a systemd unit file syntax



Unit file layout – httpd.service

[Unit] Description=The Apache HTTP Server After=network.target remote-fs.target nss-lookup.target

[Service] Type=notify EnvironmentFile=/etc/sysconfig/httpd ExecStart=/usr/sbin/httpd \$OPTIONS -DFOREGROUND ExecReload=/usr/sbin/httpd \$OPTIONS -k graceful ExecStop=/usr/sbin/httpd \$OPTIONS -k graceful-stop

```
KillSignal=SIGCONT
PrivateTmp=true
```

[Install] WantedBy=multi-user.target

*Comments were removed for readability



Unit file layout – Custom application example

```
[Unit]
Description=Describe the daemon
After=syslog.target network.target
```

[Service] ExecStart=/usr/sbin/[myapp] -D Type=forking PIDFile=/var/run/myapp.pid

[Install] WantedBy=multi-user.target



[Unit]

EAP Example

Description=JBoss Enterprise Application Platform

After=syslog.target network.target

User=jboss-as

Environment=JBOSS_USER=jboss-as

Environment=JBOSS_HOME=/usr/local/EAP-6.1.1/jboss-eap-6.1

Environment=JBOSS_CONSOLE_LOG=/var/log/jbossas/console.log

ExecStart=/usr/local/EAP-6.1.1/jboss-eap-6.1/bin/standalone.sh

PIDFile=/var/run/jboss-as/jboss-as-standalone.pid

SyslogIdentifier=jboss-as

LimitNOFILE=102642

CPUShares=1600

Restart=always

Slice=jboss.slice

[Install]

WantedBy=multi-user.target



EAP Example

root@host204:~	-		×						
File Edit View Search Terminal Help									
<pre>[root@host204 ~]# systemctl status jboss-as jboss-as.service - JBoss Enterprise Application Platform Loaded: loaded (/etc/systemd/system/jboss-as.service; enabled) Active: active (running) since Fri 2014-01-10 11:31:20 CST; 45s ago Main PID: 692 (standalone.sh) CGroup: /jboss.slice/jboss-as.service 692 /bin/sh /usr/local/EAP-6.1.1/jboss-eap-6.1/bin/standalone.s 1095 java -D[Standalone] -server -XX:+UseCompressedOops -Xms1303</pre>									
Jan 10 11:31:30 host204.local jboss-as[692]: 11:31:30,580 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,005 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,036 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,647 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,674 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,675 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,675 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,675 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,954 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,954 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,954 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,955 INF0 [org. Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,955 INF0 [org.	jboss apach jboss jboss jboss jboss jboss jboss jboss	.w e .a .a .a .a .a	.7 .0 .9 .9 .1 .7 .7 .0 .)						



EAP Example

root@host204:~	-		×					
File Edit View Search Terminal Help								
<pre>-jboss.slice -jboss-as.service - 692 /bin/sh /usr/local/EAP-6.1.1/jboss-eap-6.1/bin/standalone. -1095 java -D[Standalone] -server -XX:+UseCompressedOops -Xms130 -user.slice -user-0.slice -session-1.scope -1179 sshd: root@pts/0 -1185 -bash -1216 systemd-cgls -1217 systemd-cgls</pre>	sh 3m	-b 0.	0.0 303					
system.slice								
—1 /usr/lib/systemd/systemdswitched-rootsystemdeserialized —polkit.service	20)						
L=512 /usr/lib/polkit-1/polkitdno-debug								
<pre>-auditd.service -389 /sbin/auditd -n -systemd-udevd.service -343 /usr/lib/systemd/systemd-udevd -lvm2-lvmetad.service -314 /usr/sbin/lvmetad -systemd-journald.service</pre>								
-311 /usr/lib/systemd/systemd-journald			- 1					



Unit file layout – Test your unit file

- Copy the unit file
 - cp [myapp].service /etc/systemd/system/
- Alert systemd of the changes:
 - systemctl daemon-reload
- Start service
 - systemctl start [myapp].service
- View status
 - systemctl status [myapp].service



The Journal



Journal

- Indexed
- Formatted
 - Errors in red
 - Warnings in bold
- Security
- Reliability
- Intelligently rotated

http://0pointer.de/blog/projects/journalctl.html



Journal

- Does not replace rsyslog in RHEL 7
 - rsyslog is enabled by default
- Use rsyslog for traditional logging w/ enterprise features
- The journal is not persistent by default.
- Collects event metadata
- Stored in key-value pairs
 - man page: systemd.journal-fields(7)
- journalctl utility for to viewing the journal.
 - Simple (or complex) filtering
 - Interleave units, binaries, etc



Using the Journal

journalctl

	root@host151:~ 💶 🗵				
File	Ed	dit View	Search	Terminal	Help
0ct	28	15:04:58	host1	51.local	chronyd[329]: System clock wrong by -31.975399 seconds, adjustment
0ct	28	15:04:26	host1	51.local	chronyd[329]: System clock was stepped by -31.975 seconds
0ct	28	15:04:26	host1	51.local	systemd[1]: Time has been changed
0ct	28	15:04:52	host1	51.local	systemd[1]: Starting Stop Read-Ahead Data Collection
0ct	28	15:04:52	host1	51.local	systemd[1]: Started Stop Read-Ahead Data Collection.
0ct	28	15:05:32	host1	51.local	chronyd[329]: Selected source 174.133.168.194
0ct	28	15:06:08	host1	51.local	sshd[2040]: Accepted password for root from 192.168.122.1 port 4512
0ct	28	15:06:08	host1	51.local	systemd[1]: Starting user-0.slice.
0ct	28	15:06:08	host1	51.local	systemd[1]: Created slice user-0.slice.
0ct	28	15:06:08	host1	51.local	systemd[1]: Starting User Manager for 0
0ct	28	15:06:08	host1	51.local	systemd[1]: Starting Session 1 of user root.
0ct	28	15:06:08	host1	51.local	systemd[1]: Started Session 1 of user root.
0ct	28	15:06:08	host1	51.local	systemd-logind[322]: New session 1 of user root.
0ct	28	15:06:08	host1	51.local	<pre>sshd[2040]: pam_unix(sshd:session): session opened for user root by</pre>
0ct	28	15:06:08	host1	51.local	systemd[2044]: pam_unix(systemd-user:session): session opened for u
0ct	28	15:06:08	host1	51.local	systemd[2044]: Failed to open private bus connection: Failed to con
0ct	28	15:06:08	host1	51.local	systemd[2044]: Mounted /sys/kernel/config.
0ct	28	15:06:08	host1	51.local	systemd[2044]: Stopped target Sound Card.
0ct	28	15:06:08	host1	51.local	systemd[2044]: Starting Default.
0ct	28	15:06:08	host1	51.local	systemd[2044]: Reached target Default.
0ct	28	15:06:08	host1	51.local	systemd[2044]: Startup finished in 11ms.
0ct	28	15:06:08	host1	51.local	systemd[1]: Started User Manager for 0.
lin∈	es 9	962-983/9	83 (EN	D)	



Using the Journal

- Enable persistence: `mkdir /var/log/journal`
- View from boot: `journalctl -b`
- Tail -f and -n work as expected:
 - journalctl -f ; journalctl -n 50
- Filter by priority: `journalctl -p [level]`

0	emerg
1	alert
2	crit
3	err
4	warning
5	notice
6	debug



Using the Journal

- Other useful filters:
 - --since=yesterday or YYYY-MM-DD (HH:MM:SS)
 - --until=YYYY-MM-DD
 - -u [unit]
 - Pass binary e.g. /usr/sbin/dnsmasq
- View journal fields
 - journalctl [tab] [tab] ← bash-completion rocks!!
- Entire journal
 - journalctl -o verbose (useful for grep)



Systemd Resources

- RHEL 7 documentation: https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/
- Systemd project page: http://www.freedesktop.org/wiki/Software/systemd/
- Lennart Poettering's systemd blog entries: (read them all) http://Opointer.de/blog/projects/systemd-for-admins-1.html
- Red Hat System Administration II & III (RH134/RH254)
- FAQ
- Tips & Tricks




Questions?





