Agenda

• systemd 101
• Customizing Units
• Resource Management
• Converting init scripts
• The journal
systemd

• 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
RESISTANCE IS FUTILE!
RESISTANCE IS FUTILE!
systemd 101
units, systemctl, troubleshooting
systemd Units

- service.service
- socket.socket
- device.device
- mount.mount
- automount.automount
- swap.swap
- target.target
- path.path
- timer.timer
- snapshot.snapshot
- slice.slice
- scope.scope
systemd Units – httpd.service

[Unit]
Description=The Apache HTTP Server
After=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
Managing Services – Unit files

Init
• Init scripts: /etc/init.d & called from /etc/rc*

systemd
• Maintainer files: /usr/lib/systemd/system
• Administrator files: /etc/systemd/system
• Non-persistent, runtime data: /run/systemd

Note: unit files under /etc will take precedence over /usr
Managing Services – Start/Stop

Init
• service httpd {start, stop, restart, reload}

systemd
• systemctl {start, stop, restart, reload} httpd.service
Managing Services – Start/Stop

Init

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

systemd

• systemctl \{start, stop, restart, reload\} httpd.service
Managing Services – Start/Stop

• Glob units when needed.
  • systemctl restart httpd mariadb
  • systemctl enable httpd mariadb ntpd lm_sensors [etc]
• If a unit type isn't specified, .service is assumed.
  • systemctl start httpd == systemctl start httpd.service
• Shell completion is highly recommended
  • Install bash-completion
  • Add bash-completion to minimal kickstarts
• Connect to remote hosts over SSH using “-H”
Managing Services – Status

Init
• service httpd status

systemd
• systemctl status httpd.service

Tip: -l won't truncate the output
Managing Services – Status

```
[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 $OPTIONS -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...
[root@host158 ~]#
```
Managing Services – Status

That's a little more helpful than:

```
[root@host145 ~]# service httpd status
httpd (pid 1433) is running...
[root@host145 ~]#
```
Managing Services – Status

• List loaded services:
  • systemctl -t service

• List installed services:
  • systemctl list-unit-files -t service (similar to chkconfig --list)

• View state:
  • systemctl --state failed
Managing Services – Enable/Disable

Init
• `chkconfig httpd {on, off}`

systemd
• `systemctl {enable, disable, mask, unmask} httpd.service`
Targets == Runlevels

• “Runlevels” are exposed as target units
• 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]`

Note: /etc/inittab is no longer used.
Troubleshooting

• Append `systemd.unit=[target]` to the kernel
  • Rescue mode: single, s, S, or 1
  • Emergency (similar to `init=/bin/bash`): -b or emergency
• Interactive boot append: `systemd.confirm_spawn=1`
  • Enable debugging append:
    • debug
    • `debug systemd.log_target=kmsg log_buf_len=1M`
    • `debug systemd.log_target=console console=ttyS0`

http://freedesktop.org/wiki/Software/systemd/Debugging/
Troubleshooting

• Early boot shell on tty9
  • systemctl enable debug-shell.service
  • Equivalent to: `ln -s /usr/lib/systemd/system/debug-shell.service /
    /etc/systemd/system/sysinit.target.wants/`

• systemctl list-jobs

• systemd-analyze
  • Use 'blame', 'plot', or 'critical-chain' for more details

• rc.local is supported, but no longer runs last
  • chmod +x /etc/rc.d/rc.local
Customizing Units

drop-ins
What's available??

• List a unit's properties:
  • systemctl show --all httpd
• Query a single property:
  • systemctl show -p Restart httpd
    • Restart=no
• Helpful man files: systemd.exec and systemd.service
  • Restart, Nice, CPUAffinity, OOMScoreAdjust, LimitNOFILE, etc

Disclaimer: just because you can configure something doesn't mean you should!
Drop-ins

1) mkdir /etc/systemd/system/[name.type.d]/*.conf
2) vim /etc/systemd/system/httpd.service.d/50-httpd.conf
   [Service] Remember the 'S' is capitalized
   Restart=always
   CPUAffinity=0 1 2 3
   OOMScoreAdjust=-1000
3) systemctl daemon-reload
   • Changes will be applied on top of maintainer unit files.
Drop-ins

```
[root@host243 httpd.service.d]# systemctl status httpd
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
   Active: active (running) since Sun 2014-03-16 14:31:08 CDT; 2min 6s ago
   Process: 686 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=0/SUCCESS)
   Main PID: 689 (httpd)
   Status: "Total requests: 15884; Current requests/sec: 133; Current traffic:  60KB/sec"
   CGroup: /system.slice/httpd.service
          └─689 /usr/sbin/httpd -DFOREGROUND
          └─691 /usr/sbin/httpd -DFOREGROUND
          └─692 /usr/sbin/httpd -DFOREGROUND
          └─693 /usr/sbin/httpd -DFOREGROUND
          └─694 /usr/sbin/httpd -DFOREGROUND
          └─695 /usr/sbin/httpd -DFOREGROUND
          └─715 /usr/sbin/httpd -DFOREGROUND

Mar 16 14:31:08 host243.local systemd[1]: Started The Apache HTTP Server.
```
Drop-ins

• `systemd-delta` is your friend.
• Simple to use with configuration tools like Satellite, Puppet, etc.
• Simply delete the drop-in to revert to defaults.
• Don't forget `systemctl daemon-reload` when modifying units.
Resource Management

slices, scopes, services
Control Groups Made Simple

Resource Management with cgroups can reduce application or VM contention and improve throughput and predictability
Easily the best RHEL 6 feature that no one uses
Contain Your Applications and Guarantee Service

• Configure how applications, containers, and VMs will behave when resources are under contention.
• Trivial to keep poorly written applications from stomping on your system.

My daughter was not harmed during the making of this presentation.
Slices, Scopes, Services

• **Slice** – Unit type for creating the cgroup hierarchy for resource management.

• **Scope** – Organizational unit that groups a services' worker processes.

• **Service** – Process or group of processes controlled by systemd
Understanding the Hierarchy

- /

• systemd implements a standard, single-root hierarchy under /sys/fs/cgroup/systemd
Understanding the Hierarchy

• Each slice gets equal CPU time on the scheduler.
Understanding the Hierarchy

- Each slice gets equal CPU time on the scheduler.

-/-

  user.slice
  CPUShares=1024

  system.slice
  CPUShares=1024

  machine.slice
  CPUShares=1024
Understanding the Hierarchy

- `/`
  - `user.slice`
    - `user-1000.slice`
      - `session-3.scope`
        - `sshd: user`
        - `bash`
    - `user-1001.slice`
  - `system.slice`
  - `machine.slice`
Understanding the Hierarchy

/-
  
user.slice
  
user-1000.slice
  
session-3.scope
    sshd: user
    bash
  
user-1001.slice

system.slice

machine.slice

CPUShares=1024
Understanding the Hierarchy

- `/`
  - `user.slice`
    - `user-1000.slice`
      - `session-3.scope`
        - `sshd: user`
          - `bash`
    - `user-1001.slice`
  - `system.slice`
    - `tomcat.service`
    - `sshd.service`
    - `mariadb.service`
    - `httpd.service`
  - `machine.slice`
Understanding the Hierarchy

- `/`
  - `user.slice`
    - `user-1000.slice`
      - `session-3.scope`
        - `sshd: user`
          - `bash`
    - `user-1001.slice`
  - `system.slice`
    - `tomcat.service`
    - `sshd.service`
    - `mariadb.service`
    - `httpd.service`
  - `machine.slice`

CPUShares=1024
Understanding the Hierarchy

- `/`
  - `user.slice`
    - `user-1000.slice`
      - `session-3.scope`
        - `sshd: user`
          - `bash`
    - `user-1001.slice`
  - `system.slice`
    - `tomcat.service`
    - `sshd.service`
    - `mariadb.service`
    - `httpd.service`
  - `machine.slice`
    - `vm1.scope`
      - `/usr/bin/qemu`
    - `vm2.scope`
      - `/usr/bin/qemu`
Resource Management – systemd-cgls

```
1 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
   machine.slice
   |machine-qemu\x2drhel7.scope
   |   17307 /usr/bin/qemu-system-x86_64 -machine accel=kvm -name rhel7 -S -machine-qemu\x2dEAP6.scope
   |   15290 /usr/bin/qemu-system-x86_64 -machine accel=kvm -name EAP6 -S -machine-qemu\x2dEAP6.scope
user.slice
   |user-0.slice
   |   user@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 gnome-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/gvfsd-fuse /run/user/1000/gvfs -f -o big Writes
   |   13879 /usr/libexec/at-spi-bus-launcher
   |   13883 /bin/dbus-daemon --config-file=/etc/at-spi2/accessibility.conf --n
   |   13887 /usr/libexec/at-spi2-registryd --use-gnome-session
```
<table>
<thead>
<tr>
<th>Path</th>
<th>Tasks</th>
<th>%CPU</th>
<th>Memory</th>
<th>Input/s</th>
<th>Output/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>72</td>
<td>99.8</td>
<td>329.4M</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/user.slice</td>
<td>20</td>
<td>49.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice</td>
<td>16</td>
<td>49.1</td>
<td>287.2M</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/httpd.service</td>
<td>20</td>
<td>31.1</td>
<td>39.5M</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/mariadb.service</td>
<td>2</td>
<td>18.0</td>
<td>168.3M</td>
<td>0B</td>
<td>5.9M</td>
</tr>
<tr>
<td>/system.slice/NetworkManager.service</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/alsa-state.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/atd.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/auditd.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/chronyd.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/crond.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/dbus.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/libstoragegmnt.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/polkit.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/smawrd.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/sshd.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/systemd-journald.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/systemd-logind.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/system.slice/systemd-udevd.service</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>/user.slice/...0.slice/session-1.scope</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Resource Management – Configuration

• Enable the desired controller(s) – CPU, Memory, BlockIO

• Configure cgroup attributes:
  • systemctl set-property --runtime httpd.service CPUShares=2048

• Drop “--runtime” to persist:
  • systemctl set-property httpd.service CPUShares=2048

• Or place in the unit file:
  • [Service]
  • CPUShares=2048

http://0pointer.de/blog/projects/resources.html
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
• MemoryLimit=
• Use K, M, G, T suffixes
  • MemoryLimit=1G

https://www.kernel.org/doc/Documentation/cgroups/memory.txt
Resource Management - BlockIO

• 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)
• BlockIOWriteBandwidth & BlockIOWriteBandwidth
  • BlockIOWriteBandwidth=/var/log 5M

https://www.kernel.org/doc/Documentation/cgroups/blkio-controller.txt
Converting Init Scripts
You can do it! It's easy!
Remember what init scripts look like?
/etc/init.d/httpd

. /etc/rc.d/init.d/functions
if [ -f /etc/sysconfig/httpd ]; then
  . /etc/sysconfig/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=${STOP_TIMEOUT-10}
start() {
  echo -n "$prog: 
  LANG=$HTTPD_LANG daemon --pidfile=${pidfile} $httpd $OPTIONS
  RETVAL=$?
  echo
  [$RETVAL = 0 ] && touch ${lockfile}
  return $RETVAL
}
stop() {
  echo -n "$prog: 
  killproc -p ${pidfile} -d ${STOP_TIMEOUT} $httpd
  RETVAL=$?
  echo
  [$RETVAL = 0 ] && rm -f ${lockfile} ${pidfile}
  return $RETVAL
}
From RHEL 6.4; comments removed
Init – httpd continued

reload() {
    echo -n "$Prog: "
    if ! LANG=${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} $httpd
        RETVAL=$?
    ;;
    *)
        failure "$not a command"
    esac

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
Systemd – httpd.service

[Unit]
Description=The Apache HTTP Server
After=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*
To be clear

• Systemd maintains 99% backwards compatibility with LSB compatible 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.

• 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
Unit file layout – Custom application example

[Unit]
Description=Describe the daemon

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

[Install]
WantedBy=multi-user.target
EAP Example

[Unit]
Description=JBoss Enterprise Application Platform

[Service]
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=/run/jboss-as/jboss-as-standalone.pid
SyslogIdentifier=jboss-as
LimitNOFILE=102642
Slice=jboss.slice

[Install]
WantedBy=multi-user.target

Note: If you don't define “Type=” it will be “simple” by default
EAP Example

[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.sh
             -1095 java -D[standalone] -server -XX:+UseCompressedOops -Xms1303...

Jan 10 11:31:30 host204.local jboss-as[692]: 11:31:30,580 INFO [org.jboss.w...7
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,005 INFO [org.apache....0
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,036 INFO [org.apache....0
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,647 INFO [org.jboss.a...9
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,674 INFO [org.jboss.a...s
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,675 INFO [org.jboss.a...]
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,679 INFO [org.jboss.a...7
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,954 INFO [org.jboss.a...t
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,954 INFO [org.jboss.a...0
Jan 10 11:31:31 host204.local jboss-as[692]: 11:31:31,955 INFO [org.jboss.a...]
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`

http://0pointer.de/blog/projects/systemd-for-admins-3.html
The Journal

journalctl 101
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
- The journal is not persistent by default.
  - Enable persistence: `mkdir /var/log/journal`
- Stored in key-value pairs
  - `journalctl` [tab] [tab]
  - Man 7 systemd.journal-fields
- Collects event metadata along with the message
- Simple to filter
  - Interleave units, binaries, etc.
Journal – journalctl

Oct 28 15:04:58 host151.local chronyd[329]: System clock wrong by -31.975399 seconds, adjustment
Oct 28 15:04:26 host151.local chronyd[329]: System clock was stepped by -31.975 seconds
Oct 28 15:04:25 host151.local systemd[1]: Time has been changed
Oct 28 15:04:52 host151.local systemd[1]: Starting Stop Read-Ahead Data Collection...
Oct 28 15:04:52 host151.local systemd[1]: Started Stop Read-Ahead Data Collection.
Oct 28 15:05:32 host151.local chronyd[329]: Selected source 174.133.168.194
Oct 28 15:06:08 host151.local sshd[2940]: Accepted password for root from 192.168.122.1 port 4512
Oct 28 15:06:08 host151.local systemd[1]: Starting user-0.slice.
Oct 28 15:06:08 host151.local systemd[1]: Created slice user-0.slice.
Oct 28 15:06:08 host151.local systemd[1]: Starting User Manager for 0...
Oct 28 15:06:08 host151.local systemd[1]: Starting Session 1 of user root.
Oct 28 15:06:08 host151.local systemd[1]: Started Session 1 of user root.
Oct 28 15:06:08 host151.local systemd-logind[322]: New session 1 of user root.
Oct 28 15:06:08 host151.local sshd[2940]: pam_unix(sshd:session): session opened for user root by
Oct 28 15:06:08 host151.local systemd[2044]: pam_unix(systemd-user:session): session opened for u
Oct 28 15:06:08 host151.local systemd[2044]: Failed to open private bus connection: Failed to con
Oct 28 15:06:08 host151.local systemd[2044]: Mounted /sys/kernel/config.
Oct 28 15:06:08 host151.local systemd[2044]: Stopped target Sound Card.
Oct 28 15:06:08 host151.local systemd[2044]: Starting Default.
Oct 28 15:06:08 host151.local systemd[2044]: Reached target Default.
Oct 28 15:06:08 host151.local systemd[2044]: Startup finished in 11ms.
Oct 28 15:06:08 host151.local systemd[1]: Started User Manager for 0.

lines 962-983/983 (END)
Using the Journal

• Tail the journal: `journalctl -f`
• Show X number of lines: `journalctl -n 50`
• View from boot: `journalctl -b`
• Filter by priority: `journalctl -p [level]`

<table>
<thead>
<tr>
<th>Level</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>emerg</td>
</tr>
<tr>
<td>1</td>
<td>alert</td>
</tr>
<tr>
<td>2</td>
<td>crit</td>
</tr>
<tr>
<td>3</td>
<td>err</td>
</tr>
<tr>
<td>4</td>
<td>warning</td>
</tr>
<tr>
<td>5</td>
<td>notice</td>
</tr>
<tr>
<td>6</td>
<td>debug</td>
</tr>
</tbody>
</table>
Using the Journal

- Other useful filters:
  - -r reverse order
  - -u [unit]
  - binary e.g. /usr/sbin/dnsmasq [additional binaries]
  - --since=yesterday or YYYY-MM-DD (HH:MM:SS)
  - --until=YYYY-MM-DD

- View 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://0pointer.de/blog/projects/systemd-for-admins-1.html
• Red Hat System Administration II & III (RH134/RH254) http://redhat.com/training/
• Systemd FAQ
• Tips & Tricks
“Questions?”
THANK YOU