redhat.

Demystifying systemd Hands On

Ben Breard
Product Manager – Linux Containers
October 2016

Agenda

- Intro and lab setup
- Basic concepts & components
- Unit files & drop-ins
- Resource management
- The journal
- Security capabilities
- Quick Hits

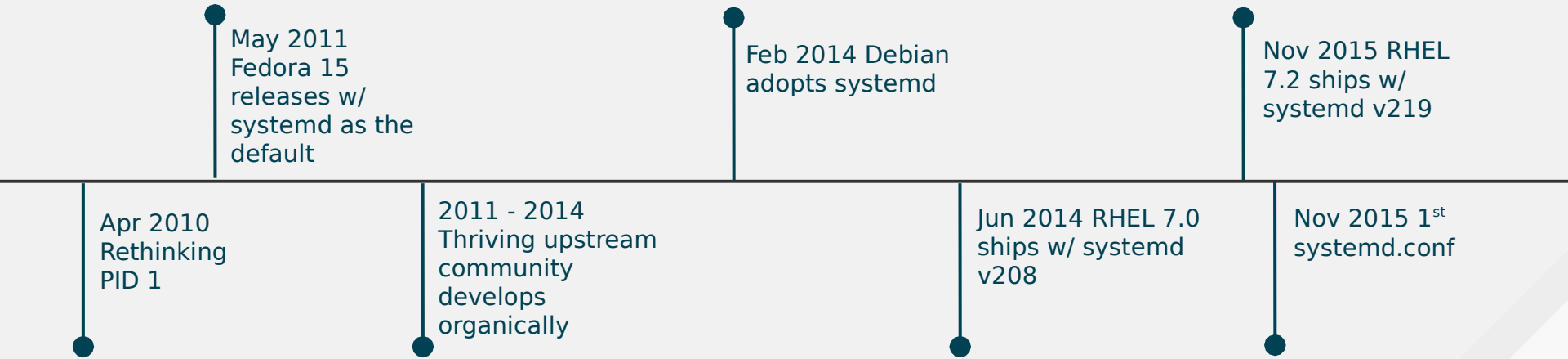

SYSTEM
V

systemd is a system & service manager

- Default init system for all major distributions
- Controls “units” rather than just daemons
- Handles dependency between units.
- Services are owned by a cgroup.
  - Tracks processes with service information
  - Simple to configure “SLAs” for CPU, Memory, and IO
  - Properly kill daemons
- Minimal boot times
- Debuggability – no early boot messages are lost
- Easy to learn and backwards compatible

systemd is not monolithic

Project Milestones



Lab Environment

- Run your favorite Linux distribution* in a container or VM
  - Note the lab was written using the Fedora 24 Cloud image.
  - Cloud-init ISO: http://people.redhat.com/bbreard/files/cloudinit.iso
- Log in!

Basic Concepts & Components

systemd Units

```
foo.service
bar.socket
baz.device
qux.mount
waldo.automount
thud.swap
```

```
grunt.target
snork.timer
grault.path
garply.snapshot
wibble.slice
flob.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

PrivateTmp=true

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

*Comments were removed for readability

systemd Units: Locations

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

`systemd-delta` – identify and compare overriding unit files

Note: unit files in /etc take precedence over /usr

Managing Services: Start/Stop

Init

```
service httpd {start,stop,restart,reload}
```

systemd

```
systemctl {start,stop,restart,reload} httpd.service
```

```
[root@host216 ~]# systemctl httpd start
Unknown operation 'httpd'.
[root@host216 ~]# son of a #@!$^!$%&%#*$@!$%!%*@^$*!
```

Managing Services: Start/Stop

- Glob units to work with multiple services
- `systemctl restart httpd mariadb`
- When the unit “type” isn't specified, .service is assumed.
  - `systemctl start httpd == systemctl start httpd.service`
- Make life easy and install shell completion
  - `yum install bash-completion`
  - `systemctl [tab] [tab]`
  - Add bash-completion to your SOE and minimal kickstarts
- Connect directly to remote hosts
  - `systemctl -H [hostname] restart httpd`

Managing Services: Status

Init

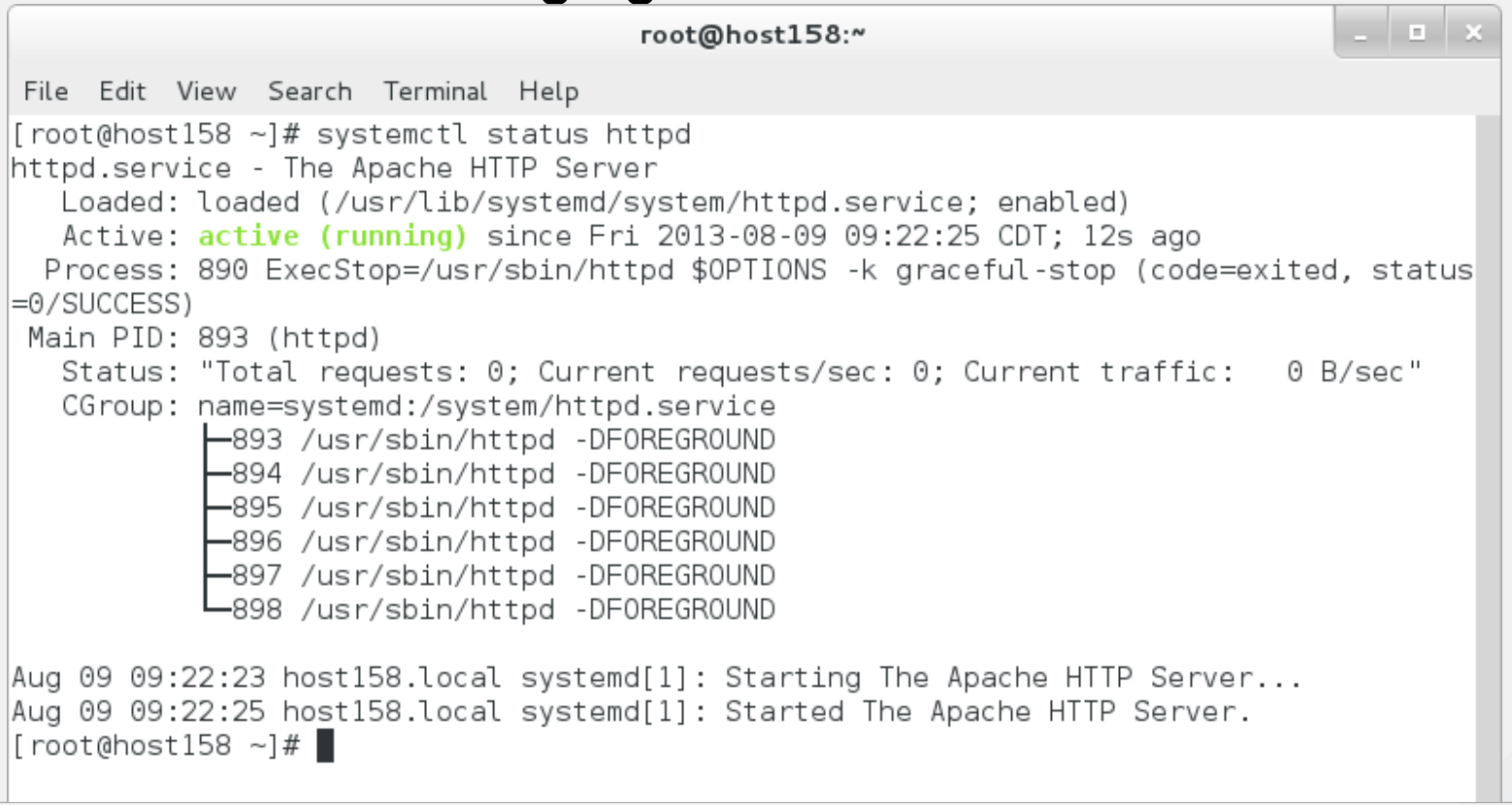
- `service httpd status`

systemd

- `systemctl status httpd`

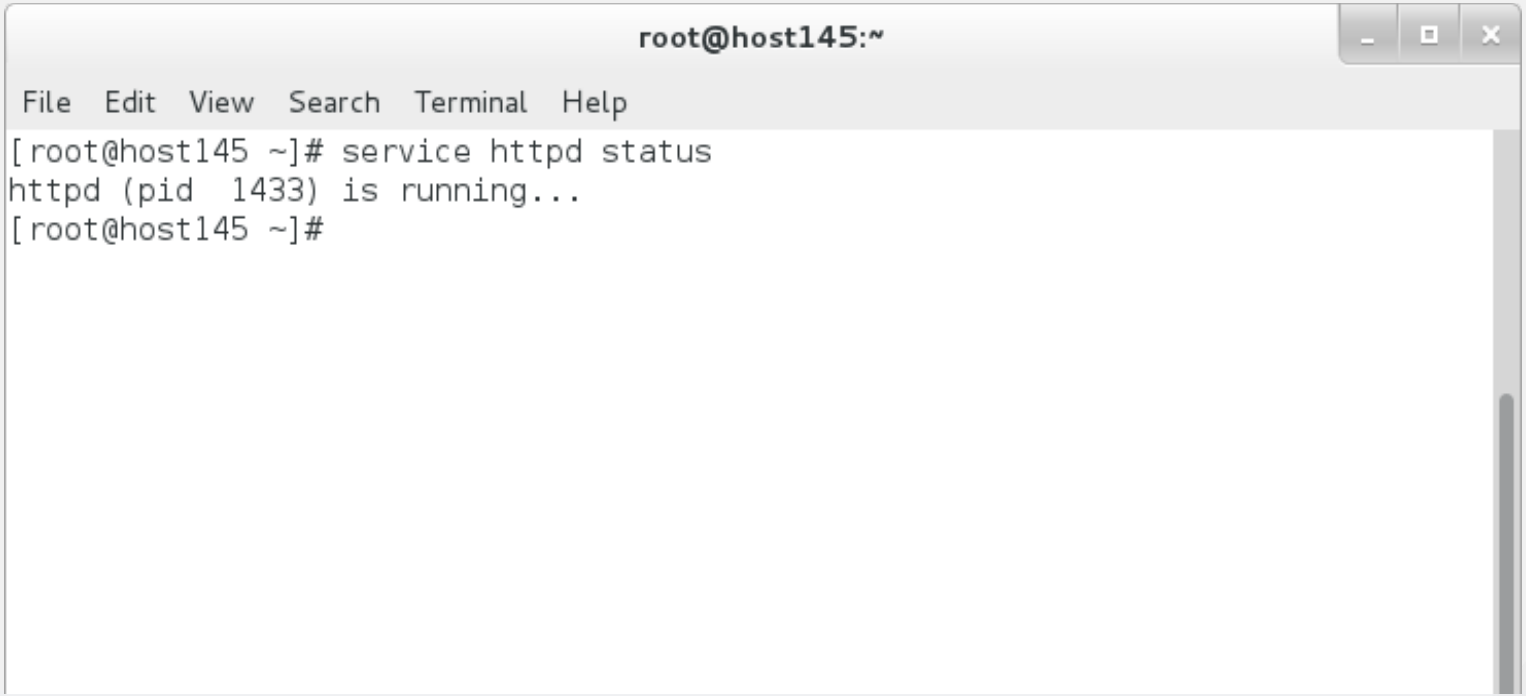
Tip: pass -l if the logs are cut off

Managing Services: Status



Managing Services: Status

That's a little more helpful than:



Managing Services: Status

- List loaded services:
  - `systemctl -t service`
- List installed services:
  - `systemctl list-unit-files -t service` (like chkconfig --list)
- Check for services in failed state:
  - `systemctl --state failed`

Managing Services: Enable/Disable

Init

- `chkconfig httpd {on,off}`

systemd

- `systemctl {enable, disable} httpd`

Tip: Start **and** enable services in one command:

- `systemctl enable --now httpd mariadb`

Targets

- Targets are groups of units
- “Runlevels” are exposed as target units
- Multiple targets can be active at once
- More meaningful names:
  - multi-user.target vs. runlevel3
  - graphical.target vs. runlevel5

Target / runlevel Mapping

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 -l /usr/lib/systemd/system/runlevel*target
lrwxrwxrwx. 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/runlevel5.target -> graphical.target
lrwxrwxrwx. 1 root root 13 Jul  3 21:37 /lib/systemd/system/runlevel6.target -> reboot.target
```

Targets

- View the default target: `systemctl get-default`
- Set the default target: `systemctl set-default [target]`
- Change at run-time: `systemctl isolate [target]`

Note: /etc/inittab is no longer used.

Sockets

tftp.socket

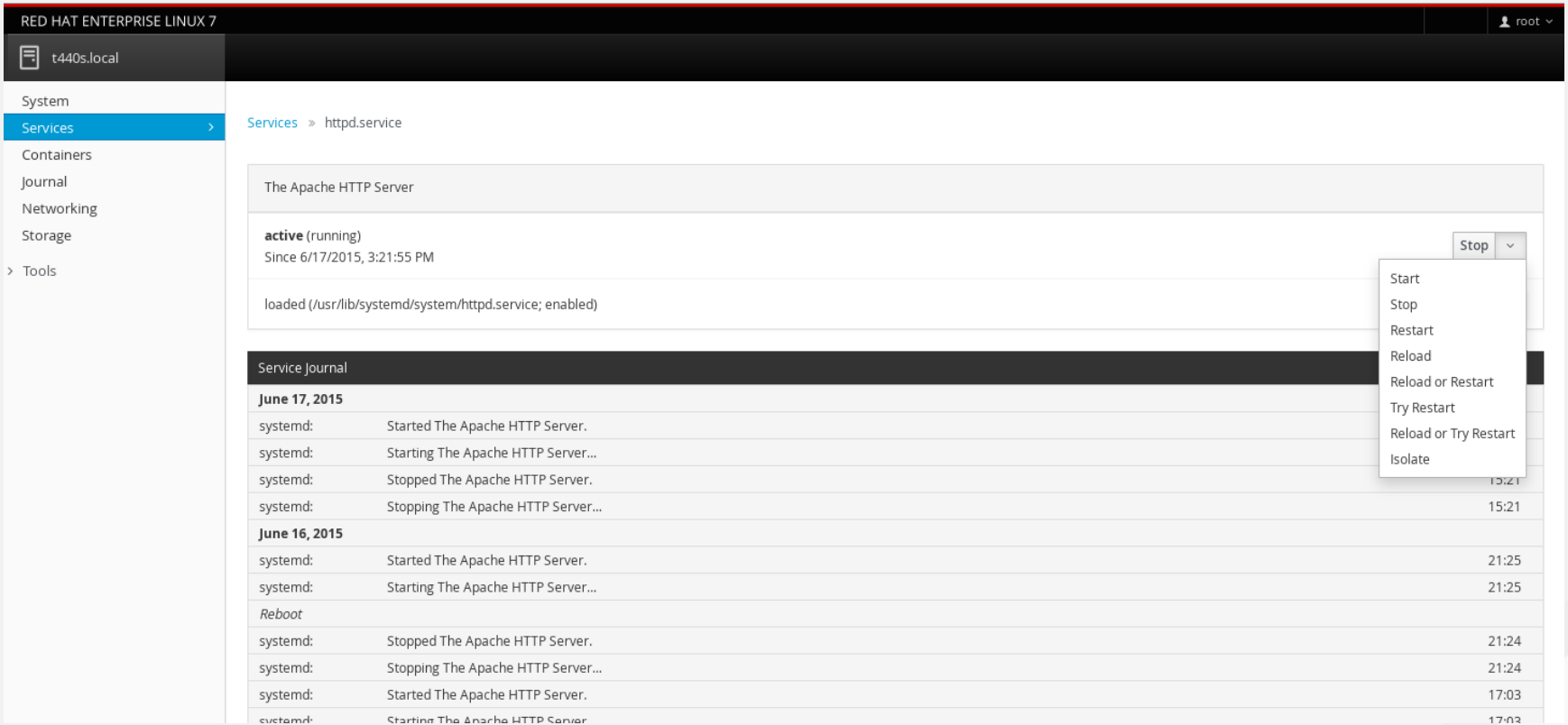
```
[Unit]
Description=Tftp Server
Activation Socket

[Socket]
ListenDatagram=69

[Install]
WantedBy=sockets.target
```

tftp.service

```
[Unit]
Description=Tftp Server

[Service]
ExecStart=/usr/sbin/in.tftpd
-s /var/lib/tftpboot
StandardInput=socket
```

```
man systemd.socket
```

Cockpit Provides an Powerful UI for systemd



Sockets

cockpit.socket

```
[Unit]
Description=Cockpit Web
Server Socket
Documentation=man:cockpit-
ws(8)

[Socket]
ListenStream=9090

[Install]
WantedBy=sockets.target
```

cockpit.service

```
[Unit]
Description=Cockpit Web Server
Documentation=man:cockpit-ws(8)

[Service]
ExecStartPre=/usr/sbin/remotectl cert
--ensure --user=root --group=cockpit-ws
ExecStart=/usr/libexec/cockpit-ws
PermissionsStartOnly=true
User=cockpit-ws
Group=cockpit-ws
```

Timers

fstrim.timer

```
[Unit]
Description=Discard unused
blocks once a week

[Timer]
OnStartupSec=10min
OnCalendar=weekly
AccuracySec=1h
Persistent=true
[Install]
WantedBy=multi-user.target
```

fstrim.service

```
[Unit]
Description=Discard unused blocks

[Service]
Type=oneshot
ExecStart=/usr/sbin/fstrim -a
```

```
man systemd.timer
```

Lab 1
https://goo.gl/alj2UB

Unit Files

What’s in the Unit

- First, view the unit file
  - `cat /usr/lib/systemd/system/httpd.service`
  - `systemctl cat httpd`

```
# /usr/lib/systemd/system/httpd.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=man:httpd(8)
Documentation=man:apachectl(8)

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/httpd
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
```

What’s in the Unit

- [Unit] describes the unit and defines it’s dependencies
  - man systemd.unit
- [Install] configures how the unit is installed w/ systemctl enable/disable
  - man systemd.unit
- [Service] specific configuration options
  - man systemd.service

What Options are 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) Create directory
   - `mkdir /etc/systemd/system/[name.type.d]/`
2) Create drop-in
   - `vim /etc/systemd/system/httpd.service.d/50-httpd.conf`

```
[Service]
Restart=always
CPUAffinity=0 1 2 3
OOMScoreAdjust=-1000
```

Remember the 'S' is capitalized

3) Notify systemd of the changes
   - `systemctl daemon-reload`

Drop-ins: The Easy Way

1) Create the drop-in

```
systemctl edit httpd
```

2) Add desired changes via the editor

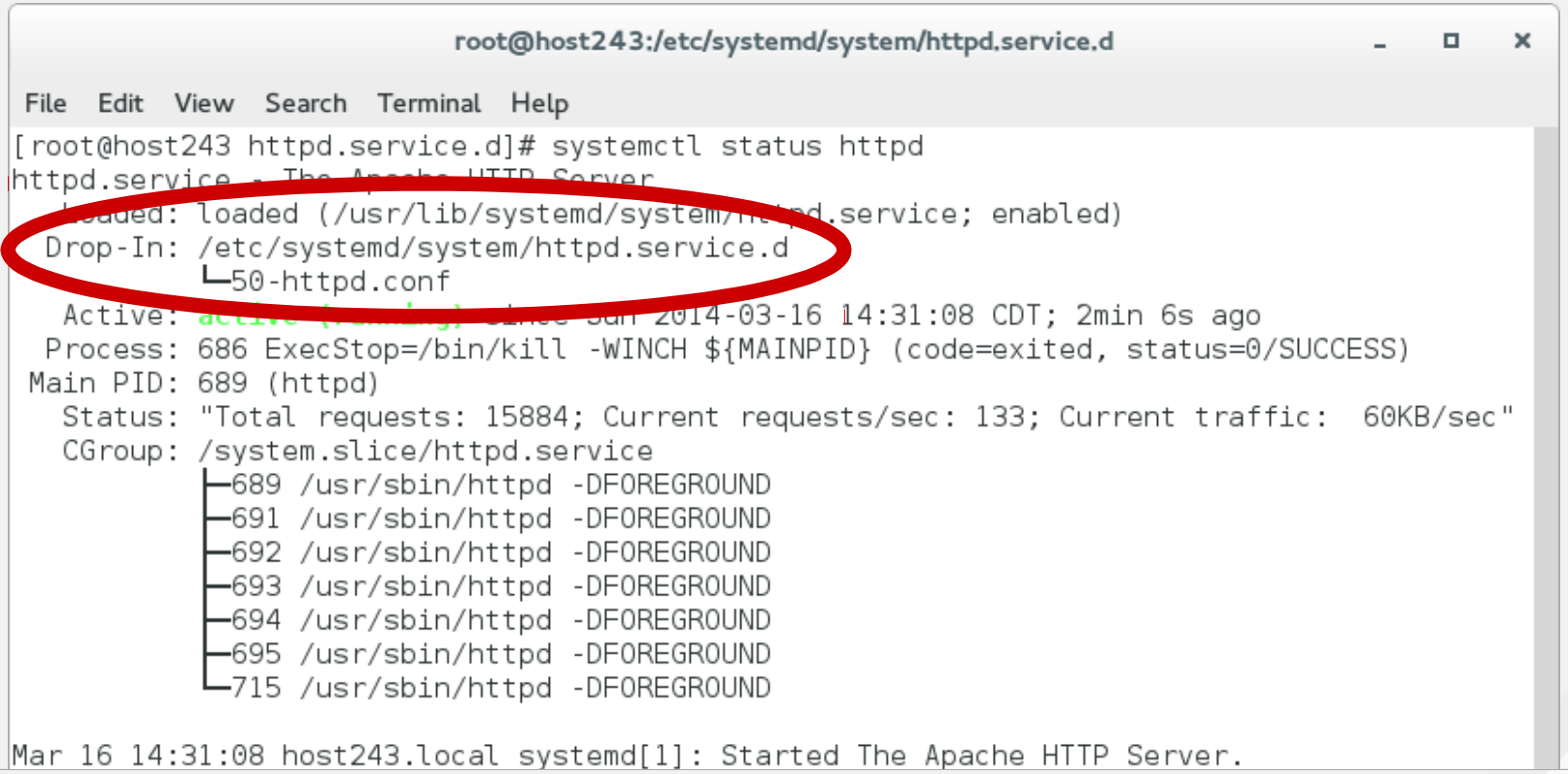
```
[Service]
Restart=always
```

3) Changes take effect upon writing the file

```
systemctl show -p Restart httpd
Restart=always
```

Tip: Set `$EDITOR` to specify an editor of your choice

Customizing Units: Drop-ins



Customizing Units: Drop-ins

- Safe to apply on running services
  - Note: some options will require a service restart to take effect
- Use **`systemd-delta`** to see what's been altered on a system

[EXTENDED] /usr/lib/systemd/system/httpd.service → /etc/systemd/system/httpd.service.d/override.conf

- Simple to use with configuration tools like Ansible, Puppet, etc.
- Simply delete the drop-in to revert to defaults.
- Don't forget **`systemctl daemon-reload`** when modifying units manually.

Quick Look at Converting Init Scripts

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

/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 $"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

/etc/init.d/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} $httpd
        RETVAL=$?
        ;;
```

/etc/init.d/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
```

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

PrivateTmp=true

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

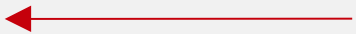
*Comments were removed for readability

Unit file layout – Custom application example

```
[Unit]
Description=foo, a demo application designed to break the
monotony of wordpress demos

[Service]
ExecStart=/usr/bin/foo -D
Type=forking
PIDFile=/var/run/foo.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=/var/run/jboss-as/jboss-as-standalone.pid
Slice=jboss.slice

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



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

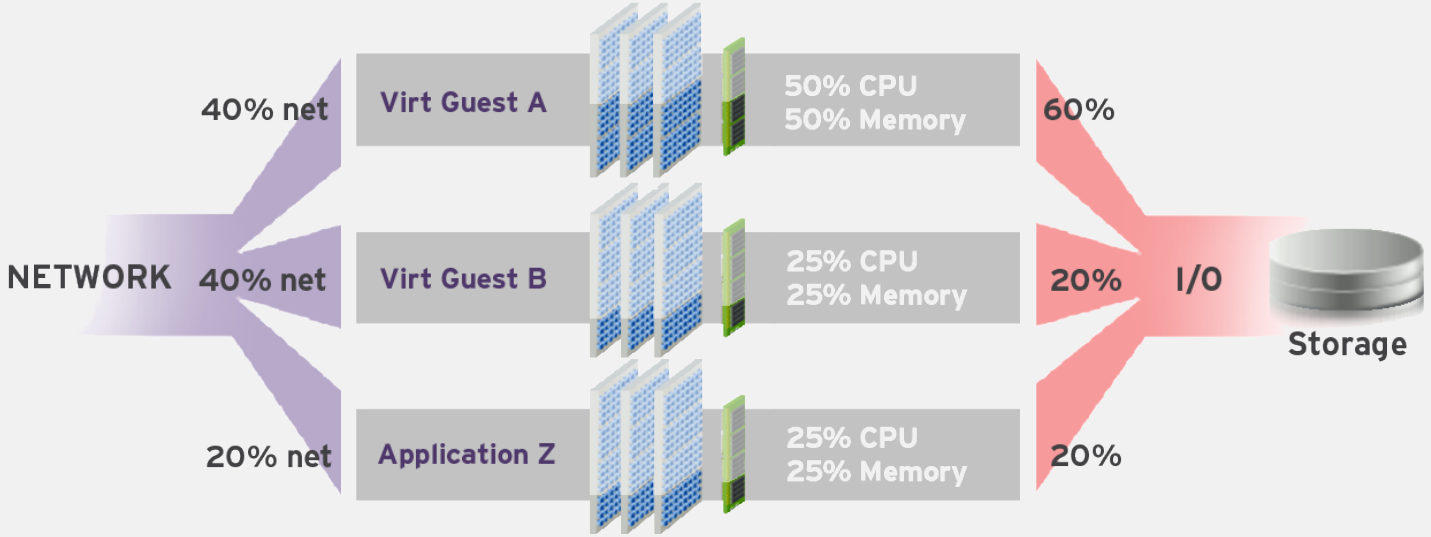
http://0pointer.de/blog/projects/systemd-for-admins-3.html

Lab 2

Resource Management

Control Groups Made Simple

Resource Management with cgroups can reduce contention and improve throughput and predictability



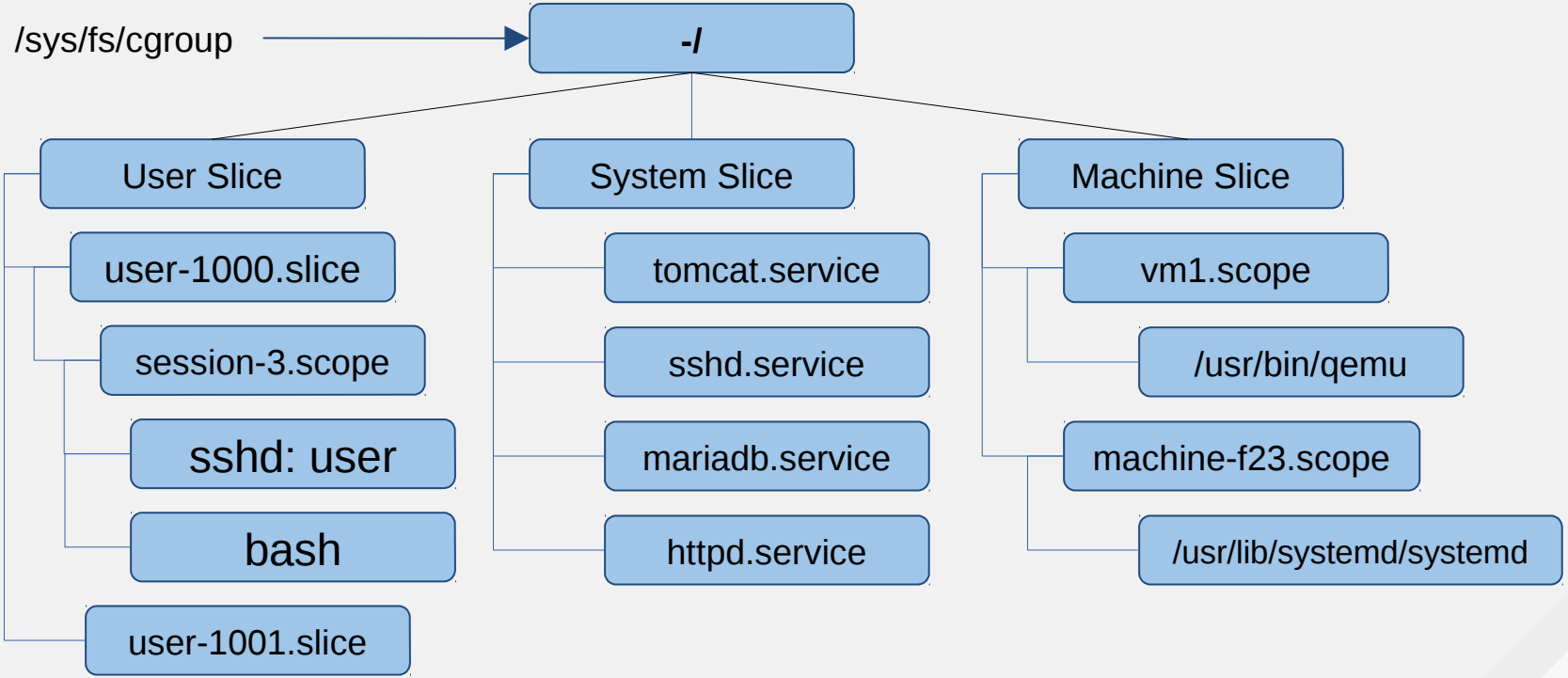
Slices, Scopes, Services

- **Slice** – Unit type for creating the cgroup hierarchy for resource management.
- **Scope** – Organizational unit that groups a services' worker processes. Used for user sessions, VMs, containers, etc.
- **Service** – Process or group of processes controlled by systemd

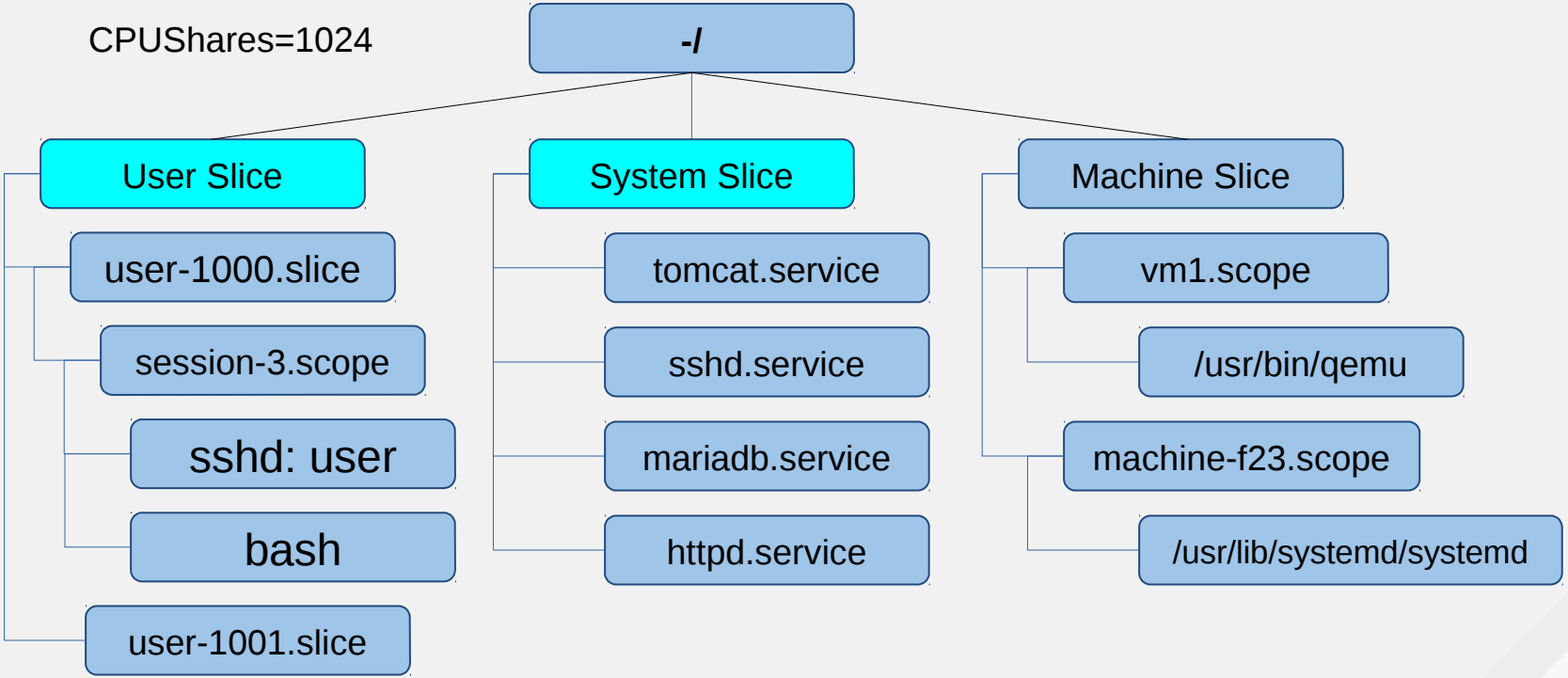
Slices, Scopes, Services

- By default, CPUShares=1024 for new slices, scopes, & services
- Under contention slices, scopes, & services will have equal “share” of the processor.

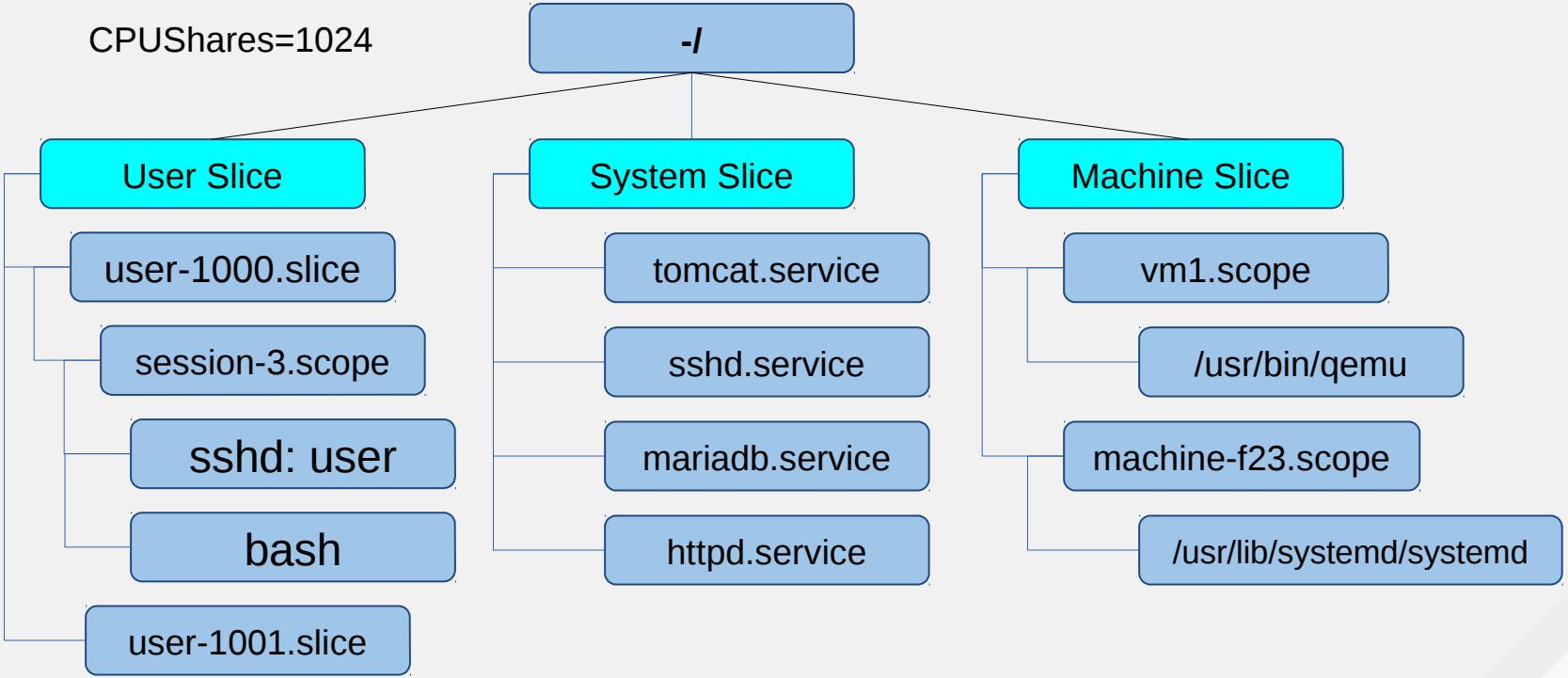
Slices, Scopes, Services



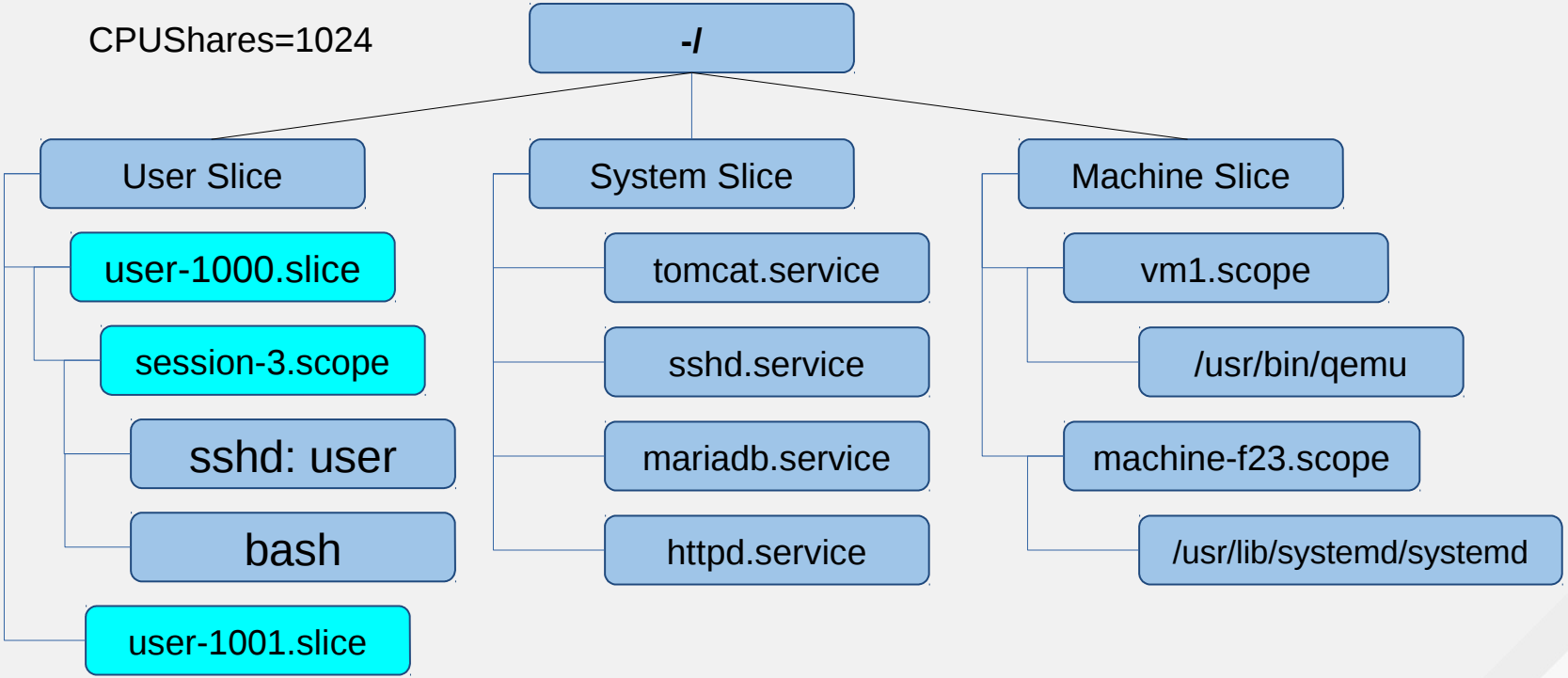
Slices, Scopes, Services



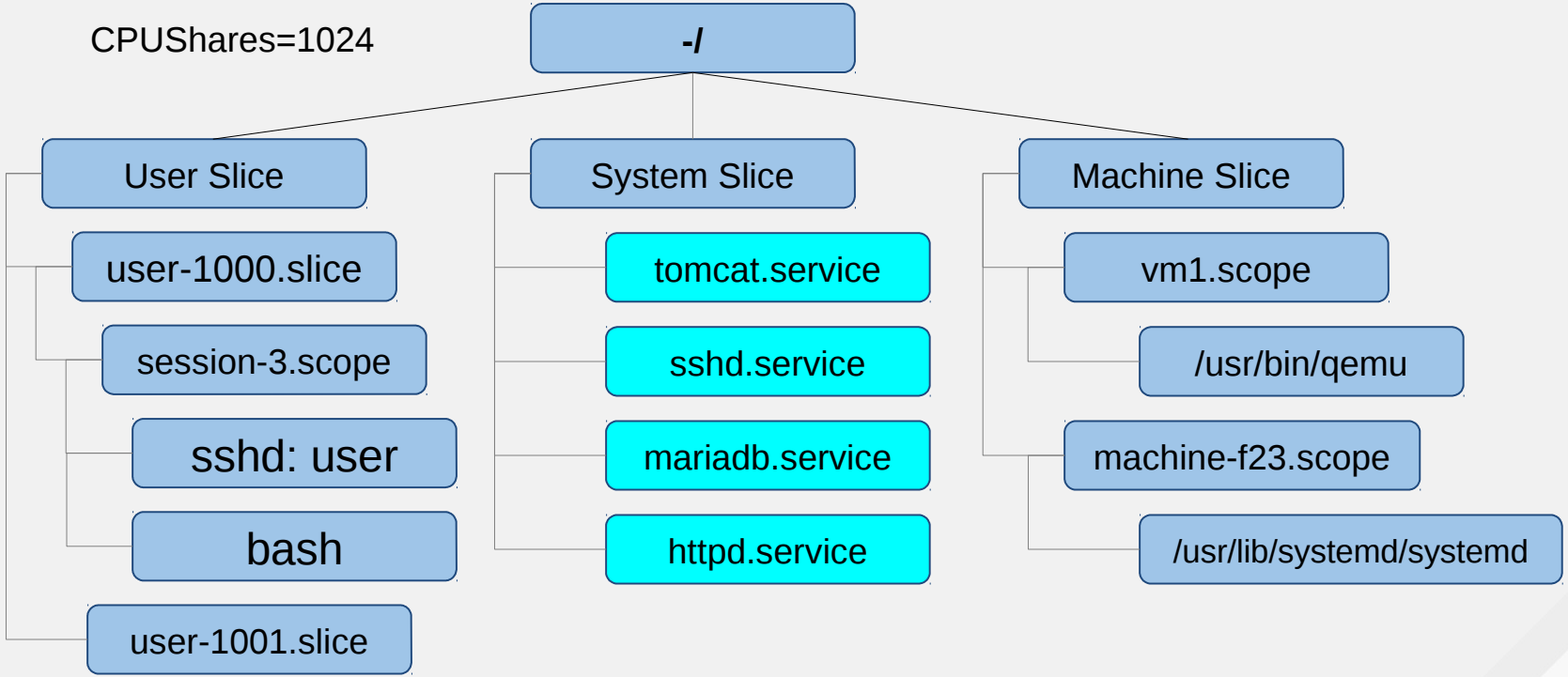
Slices, Scopes, Services



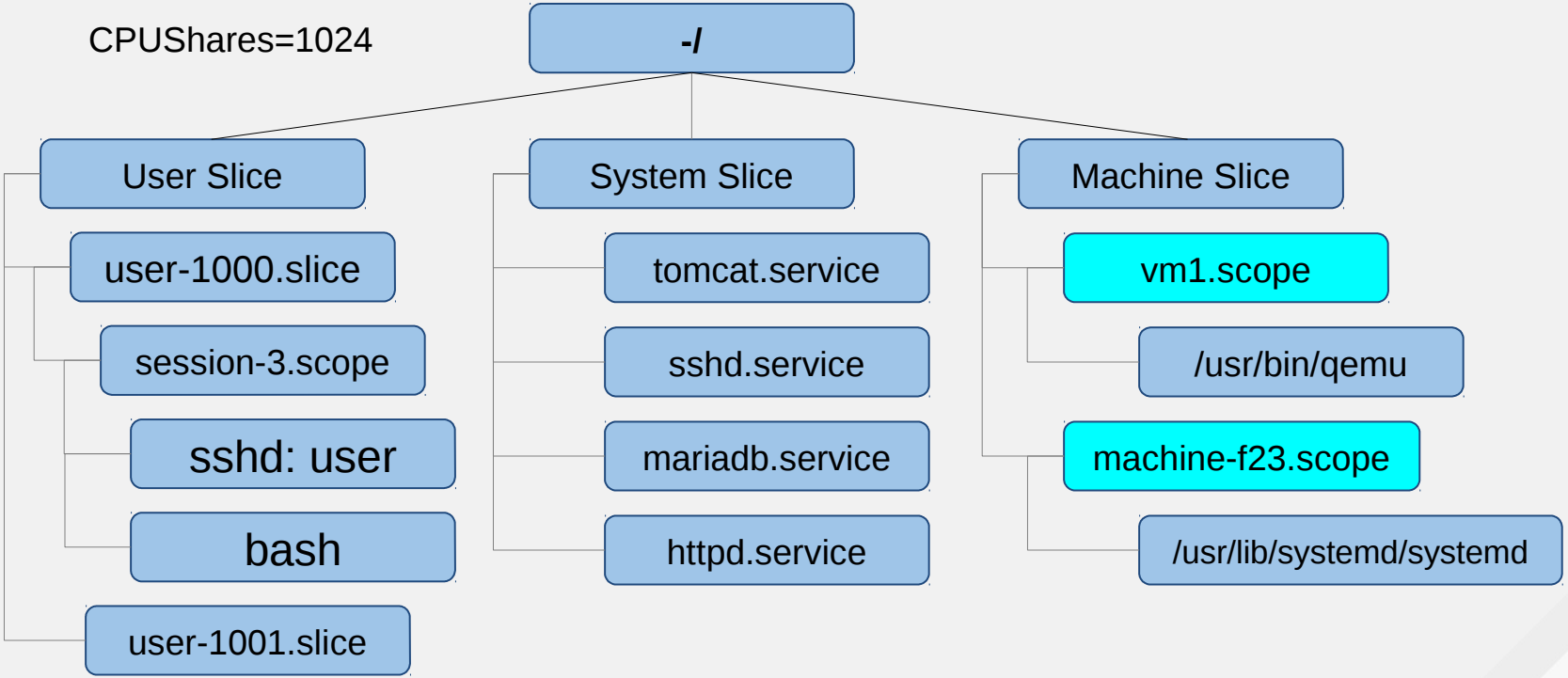
Slices, Scopes, Services



Slices, Scopes, Services



Slices, Scopes, Services



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 -machi
│ └─machine-qemu\x2dEAP6.scope
│   └─15290 /usr/bin/qemu-system-x86_64 -machine accel=kvm -name EAP6 -S -machin
├─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
lines 1-23
```

Resource Management – systemd-cgtop

Path	Tasks	%CPU	Memory	Input/s	Output/s
/	72	99.8	329.4M	-	-
/user.slice	20	49.1	-	-	-
/system.slice	16	49.1	287.2M	-	-
/system.slice/httpd.service	20	31.1	39.5M	-	-
/system.slice/mariadb.service	2	18.0	168.3M	0B	5.9M
/system.slice/NetworkManager.service	2	-	-	-	-
/system.slice/alsa-state.service	1	-	-	-	-
/system.slice/atd.service	1	-	-	-	-
/system.slice/auditd.service	1	-	-	-	-
/system.slice/chronyd.service	1	-	-	-	-
/system.slice/crond.service	1	-	-	-	-
/system.slice/dbus.service	1	-	-	-	-
/system.slice/libstoragemgmt.service	1	-	-	-	-
/system.slice/polkit.service	1	-	-	-	-
/system.slice/smartd.service	1	-	-	-	-
/system.slice/sshd.service	1	-	-	-	-
/system.slice/systemd-journald.service	1	-	-	-	-
/system.slice/systemd-logind.service	1	-	-	-	-
/system.slice/systemd-udevd.service	1	-	-	-	-
/user.slice/...0.slice/session-1.scope	2	-	-	-	-

Resource Management – Configuration

- Configure cgroup attributes:
  - `systemctl set-property --runtime httpd CPUShares=2048`
- Drop “--runtime” to persist (will create a drop-in):
  - `systemctl set-property httpd CPUShares=2048`
- Or place in the unit file:
  - `[Service]`
  - `CPUShares=2048`

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

Resource Management – CPU & MEM

- CPUAccounting=1 to enable
- CPUShares – default is 1024.
  - e.g. CPUShares=1600
- CPUQuota – Max percentage of single CPU.
  - e.g. CPUQuota=200%

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

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

https://www.kernel.org/doc/Documentation/scheduler/sched-design-CFS.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
- BlockIODeviceWeight - Can be defined per device (or mount point)
- BlockIOReadBandwidth & BlockIOWriteBandwidth
  - BlockIOWriteBandwith=/var/log 5M

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

Lab 3

The Journal

Journal

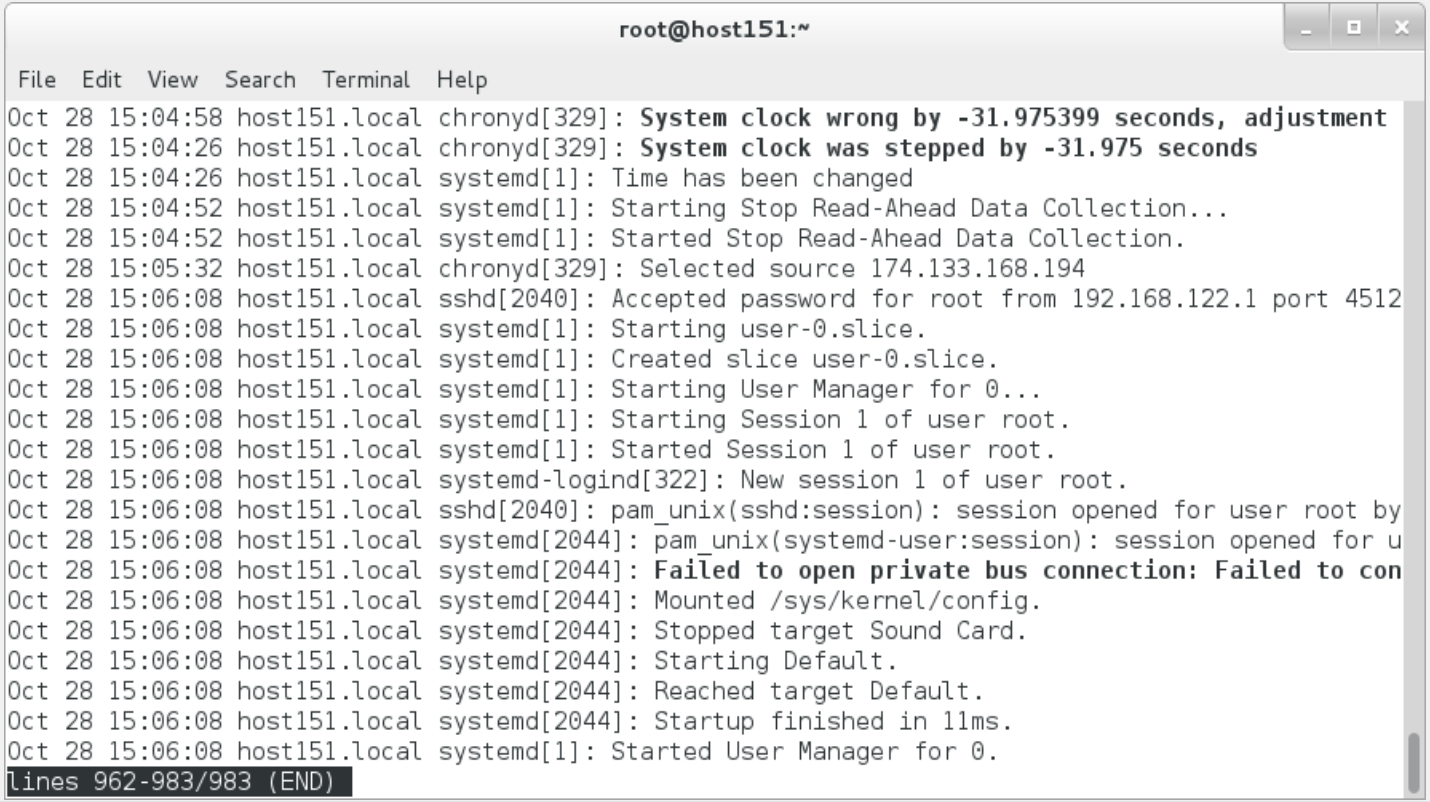
- Indexed
- Formatted
  - Errors in red
  - Warnings in bold
- Security
- Reliability
- Intelligently rotated

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

Journal

- Can forward to a traditional syslog
- The journal is persistent on most distros.
  - 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.

journalctl



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]`

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

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)

Lab 4

Security Capabilities

Security Capabilities

- `PrivateTmp=`
  - File system namespace with `/tmp & /var/tmp`
  - Files are under `/tmp/systemd-private-*-[unit]-*/tmp`
- `PrivateNetwork=`
  - Creates a network namespace with a single loopback device
- `JoinsNamespaceOf=`
  - Enables multiple units to share PrivateTmp & PrivateNetwork
- `SELinuxContext=`
  - Specify an SELinux security context for the process/service

https://www.freedesktop.org/software/systemd/man/systemd.exec.html

Security Capabilities

- `ProtectSystem=`
  - If enabled, /usr & /boot directories are mounted read-only
  - If “full”, /etc is also read-only
- `ProtectHome=`
  - If enabled, /home, /root, /run/user will appear empty
  - Alternatively can set to “read-only”
- `PrivateDevices=`
  - If enabled, creates a private `/dev namespace.`
  - Includes pseudo devices like `/dev/null, /dev/zero,` etc
  - Disables `CAP_MKNOD`

https://www.freedesktop.org/software/systemd/man/systemd.exec.html

Security Capabilities

- `ReadWriteDirectories=, ReadOnlyDirectories=, InaccessibleDirectories=`
  - Configure file system namespaces
- `NoNewPrivileges=`
  - Ensure a process & children cannot elevate privileges
- `CapabilityBoundingSet=`
  - CAP_SYS_ADMIN
  - ~CAP_NET_ADMIN

Quick Hits

Tips & Troubleshooting

- Early boot shell on tty9
  - `systemctl enable debug-shell.service`
  - `Append systemd.debug-shell`
- `systemctl list-jobs`
- 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/

Tips & Troubleshooting

- `rc.local is supported, but no longer runs last`
  - `chmod +x /etc/rc.d/rc.local`
- `systemd-analyze`
  - Use 'blame', 'plot', or 'critical-chain' for more details
- Append systemd.unit=[target] to the kernel
  - Rescue mode: single, s, S, or 1
  - Emergency (similar to init=/bin/bash): -b or emergency

networkd

- DHCP srv/clt, bridge, bond, vlan, vxlan, macvlan, tun, etc
- Disable NetworkManager, network, etc
- systemctl enable --now systemd-networkd systemd-resolvd

/etc/systemd/network/80-dhcp.network

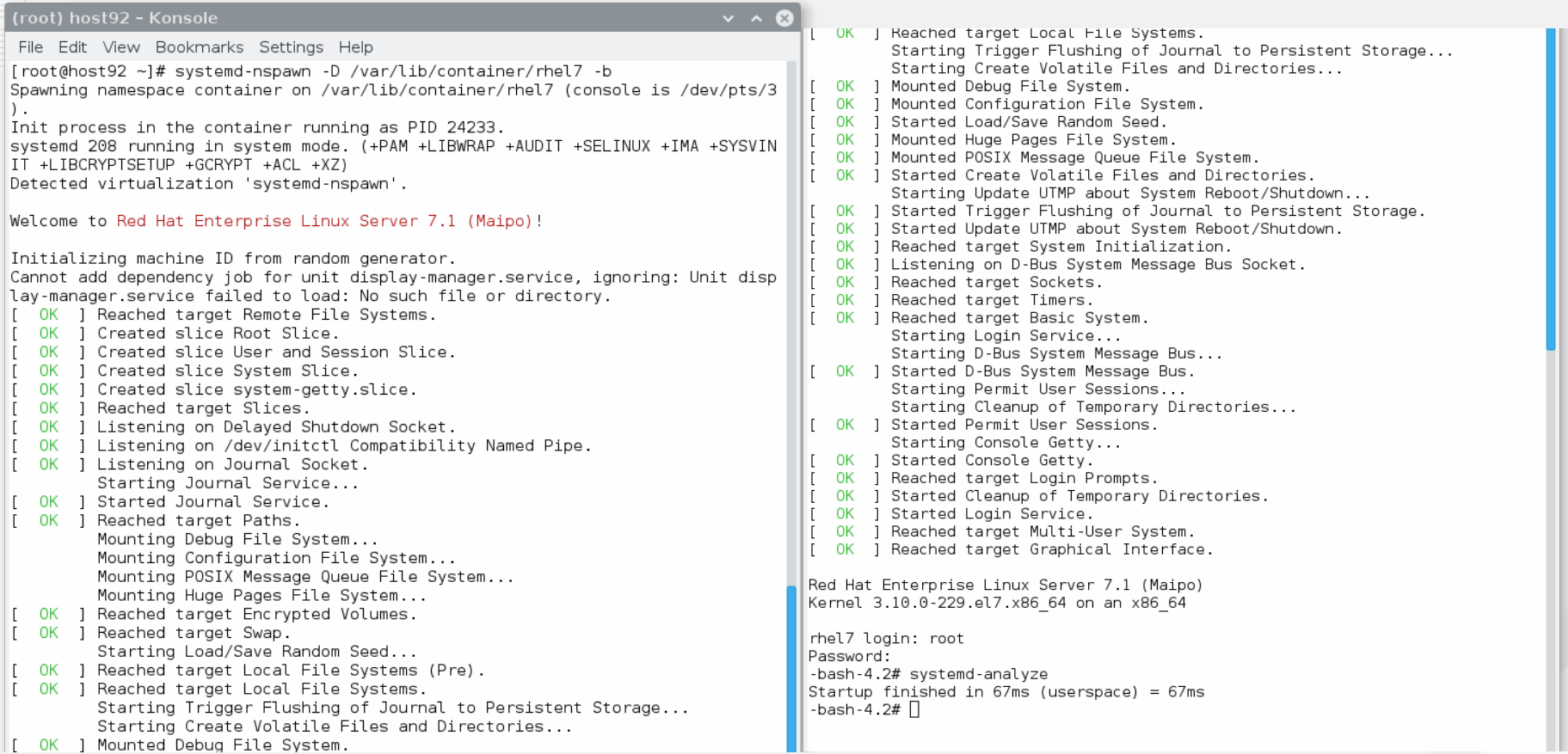
```
[Match]
Name=en*

[Network]
DHCP=yes
```

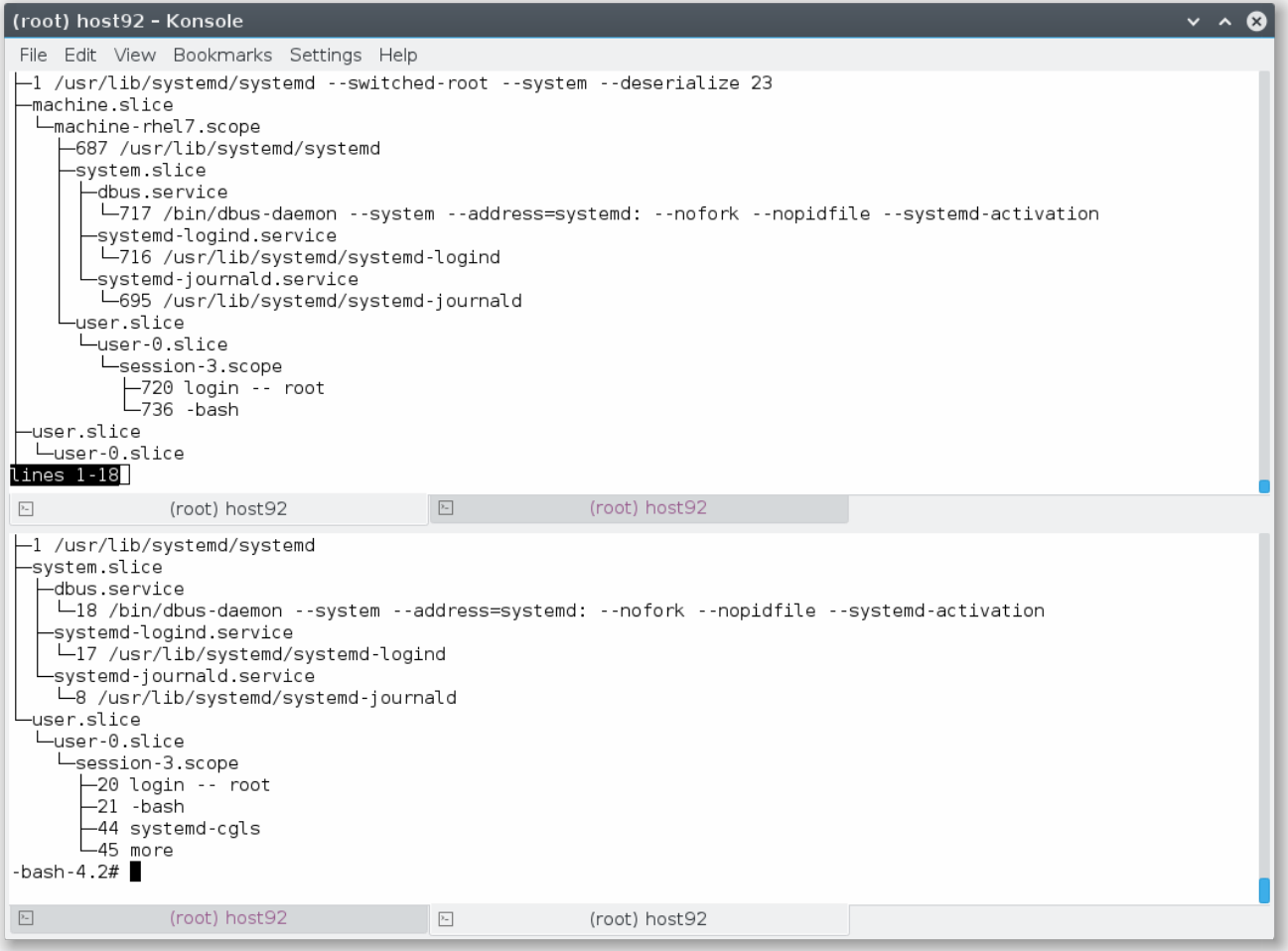
/etc/systemd/network/50-static.network

```
[Match]
Name=enp0s25

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
DNS=192.168.0.1
```

nspawn

- Install a minimal OS w/ yum; 306 rpms ~360MB on disk:
  - `Fedora: sudo dnf -y --releasever=24 –installroot=/var/lib/machines/f24 install systemd passwd dnf fedora-release`
  - `Debian: debootstrap --arch=amd64 stable /var/lib/machines/DebianJessie`
- Point nspawn at the directory and go
  - `systemd-nspawn -D /var/lib/machines/f24`
- To “boot with an init system”, we need to, start the container, set the root password, and configure the system if necessary, etc
  - `systemd-nspawn -D /var/lib/machines/f24`
  - `passwd ; systemctl disable [unwanted units]`


(root) host92 - Konsole
File Edit View Bookmarks Settings Help
[root@host92 ~]# systemd-nspawn -D /var/lib/container/rhel7 -b
Spawning namespace container on /var/lib/container/rhel7 (console is /dev/pts/3
).
Init process in the container running as PID 24233.
systemd 208 running in system mode. (+PAM +LIBWRAP +AUDIT +SELINUX +IMA +SYSVIN
IT +LIBCRYPTSETUP +GCRYPT +ACL +XZ)
Detected virtualization 'systemd-nspawn'.
Welcome to Red Hat Enterprise Linux Server 7.1 (Maipo)!
Initializing machine ID from random generator.
Cannot add dependency job for unit display-manager.service, ignoring: Unit disp
lay-manager.service failed to load: No such file or directory.
[OK] Reached target Remote File Systems.
[OK] Created slice Root Slice.
[OK] Created slice User and Session Slice.
[OK] Created slice System Slice.
[OK] Created slice system-getty.slice.
[OK] Reached target Slices.
[OK] Listening on Delayed Shutdown Socket.
[OK] Listening on /dev/initctl Compatibility Named Pipe.
[OK] Listening on Journal Socket.
Starting Journal Service...
[OK] Started Journal Service.
[OK] Reached target Paths.
Mounting Debug File System...
Mounting Configuration File System...
Mounting POSIX Message Queue File System...
Mounting Huge Pages File System...
[OK] Reached target Encrypted Volumes.
[OK] Reached target Swap.
Starting Load/Save Random Seed...
[OK] Reached target Local File Systems (Pre).
[OK] Reached target Local File Systems.
Starting Trigger Flushing of Journal to Persistent Storage...
Starting Create Volatile Files and Directories...
[OK] Mounted Debug File System.
[OK] Reached target Local File Systems.
Starting Trigger Flushing of Journal to Persistent Storage...
Starting Create Volatile Files and Directories...
[OK] Mounted Debug File System.
[OK] Mounted Configuration File System.
[OK] Started Load/Save Random Seed.
[OK] Mounted Huge Pages File System.
[OK] Mounted POSIX Message Queue File System.
[OK] Started Create Volatile Files and Directories.
Starting Update UTMP about System Reboot/Shutdown...
[OK] Started Trigger Flushing of Journal to Persistent Storage.
[OK] Started Update UTMP about System Reboot/Shutdown.
[OK] Reached target System Initialization.
[OK] Listening on D-Bus System Message Bus Socket.
[OK] Reached target Sockets.
[OK] Reached target Timers.
[OK] Reached target Basic System.
Starting Login Service...
Starting D-Bus System Message Bus...
[OK] Started D-Bus System Message Bus.
Starting Permit User Sessions...
Starting Cleanup of Temporary Directories...
[OK] Started Permit User Sessions.
Starting Console Getty...
[OK] Started Console Getty.
[OK] Reached target Login Prompts.
[OK] Started Cleanup of Temporary Directories.
[OK] Started Login Service.
[OK] Reached target Multi-User System.
[OK] Reached target Graphical Interface.
Red Hat Enterprise Linux Server 7.1 (Maipo)
Kernel 3.10.0-229.el7.x86_64 on an x86_64
rhel7 login: root
Password:
-bash-4.2# systemd-analyze
Startup finished in 67ms (userspace) = 67ms
-bash-4.2#


(root) host92 - Konsole
File Edit View Bookmarks Settings Help
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 23
├─machine.slice
│ └─machine-rhel7.scope
│ ├─687 /usr/lib/systemd/systemd
│ ├─system.slice
│ │ ├─dbus.service
│ │ │ └─717 /bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation
│ │ ├─systemd-logind.service
│ │ │ └─716 /usr/lib/systemd/systemd-logind
│ │ └─systemd-journald.service
│ │ └─695 /usr/lib/systemd/systemd-journald
│ └─user.slice
│ └─user-0.slice
│ └─session-3.scope
│ ├─720 login -- root
│ └─736 -bash
├─user.slice
│ └─user-0.slice
lines 1-18
(root) host92
(root) host92
├─1 /usr/lib/systemd/systemd
├─system.slice
│ ├─dbus.service
│ │ └─18 /bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation
│ ├─systemd-logind.service
│ │ └─17 /usr/lib/systemd/systemd-logind
│ └─systemd-journald.service
│ └─8 /usr/lib/systemd/systemd-journald
└─user.slice
└─user-0.slice
└─session-3.scope
├─20 login -- root
├─21 -bash
├─44 systemd-cgls
└─45 more
-bash-4.2#
(root) host92
(root) host92

nspawn

- To start on boot in EL 7.0 & 7.1:
  - cp /usr/lib/systemd/system/systemd-nspawn\@.service /etc/systemd/system/systemd-nspawn@f24.service
  - systemctl enable systemd-nspawn@f24.service

Lab 5

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos