Demystifying systemd

2017: RHEL 7.3 Edition

Ben Breard, RHCA
Sr Product Manager - Linux Containers

Lennart Poettering
Sr Principal Engineer

AGENDA

- Concepts & Basic Usage
- Modifying Units
- Security Capabilities
- Resource Management

systemd is a System & Service Manager

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

systemd is not monolithic

“NO SANE PERSON wants systemd”

Random comment on public blog

LIFE BEYOND INIT
CONCEPTS & BASIC USAGE

Units

foo.**service**	grunt.**target**
bar.**socket**	snork.**timer**
baz.**device**	grault.**path**
qux.**mount**	garply.**snapshot**
waldo.**automount**	pizza.**slice**
thud.**swap**	tele.**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 removed for readability

systemd Units: Locations

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

systemd-delta - Identify and compare overriding unit files

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

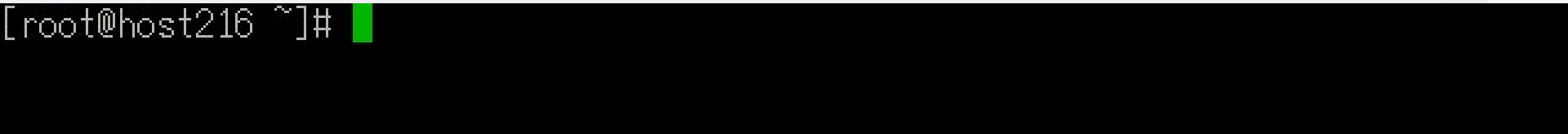
Managing Services: Start/Stop

Init

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

systemd

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



Managing Services: Start/Stop

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

Managing Services: Status

Init

```
service httpd status
```

systemd

```
systemctl status httpd
```

Tip: pass -l to see the full logs

Managing Services: Status

```
[root@camacho ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2017-04-27 18:32:35 CDT; 3h 47min ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 3235 (httpd)
   Status: "Total requests: 253156; Current requests/sec: 1e+04; Current traffic: 4.7MB/sec"
   CGroup: /system.slice/httpd.service
           ├─3235 /usr/sbin/httpd -DFOREGROUND
           ├─3239 /usr/sbin/httpd -DFOREGROUND
           ├─3241 /usr/sbin/httpd -DFOREGROUND
           ├─3242 /usr/sbin/httpd -DFOREGROUND
           ├─4071 /usr/sbin/httpd -DFOREGROUND
           ├─4073 /usr/sbin/httpd -DFOREGROUND
           ├─4075 /usr/sbin/httpd -DFOREGROUND
           ├─4076 /usr/sbin/httpd -DFOREGROUND
           ├─4078 /usr/sbin/httpd -DFOREGROUND
           ├─4356 /usr/sbin/httpd -DFOREGROUND
           ├─4358 /usr/sbin/httpd -DFOREGROUND
           ├─5741 /usr/sbin/httpd -DFOREGROUND
           ├─5744 /usr/sbin/httpd -DFOREGROUND
           └─5745 /usr/sbin/httpd -DFOREGROUND

Apr 27 18:32:34 t500.local systemd[1]: Starting The Apache HTTP Server...
Apr 27 18:32:35 t500.local systemd[1]: Started The Apache HTTP Server.
[root@camacho ~]#
```

I don’t care how awesome that is!

“systemd is the best example of Suck.”

http://suckless.org/sucks/systemd

Managing Services: Status

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

Managing Services: Enable/Disable

Init

```
chkconfig httpd {on,off}
```

systemd

```
systemctl {enable, disable} httpd
```

Tip: Clean up kickstarts by globing units:

```
    systemctl enable httpd mariadb lm_sensors
```

Usage Tips & Tricks

- Start **and** enable services in one command:
  - `systemctl enable --now httpd mariadb`
- Control remote hosts
  - `systemctl -H [hostname] restart httpd`
- rc.local is supported, but no longer runs last
  - `chmod +x /etc/rc.d/rc.local`
- `systemd-analyze`
  - Pass '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`

Targets

- Targets are simply 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

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.

“I find systemd’s lack of faith in UNIX disturbing”

https://bsdmag.org/randy_w_3/

Sockets

tftp.socket

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

[Socket]
ListenDatagram=69

[Install]
WantedBy=sockets.target
```

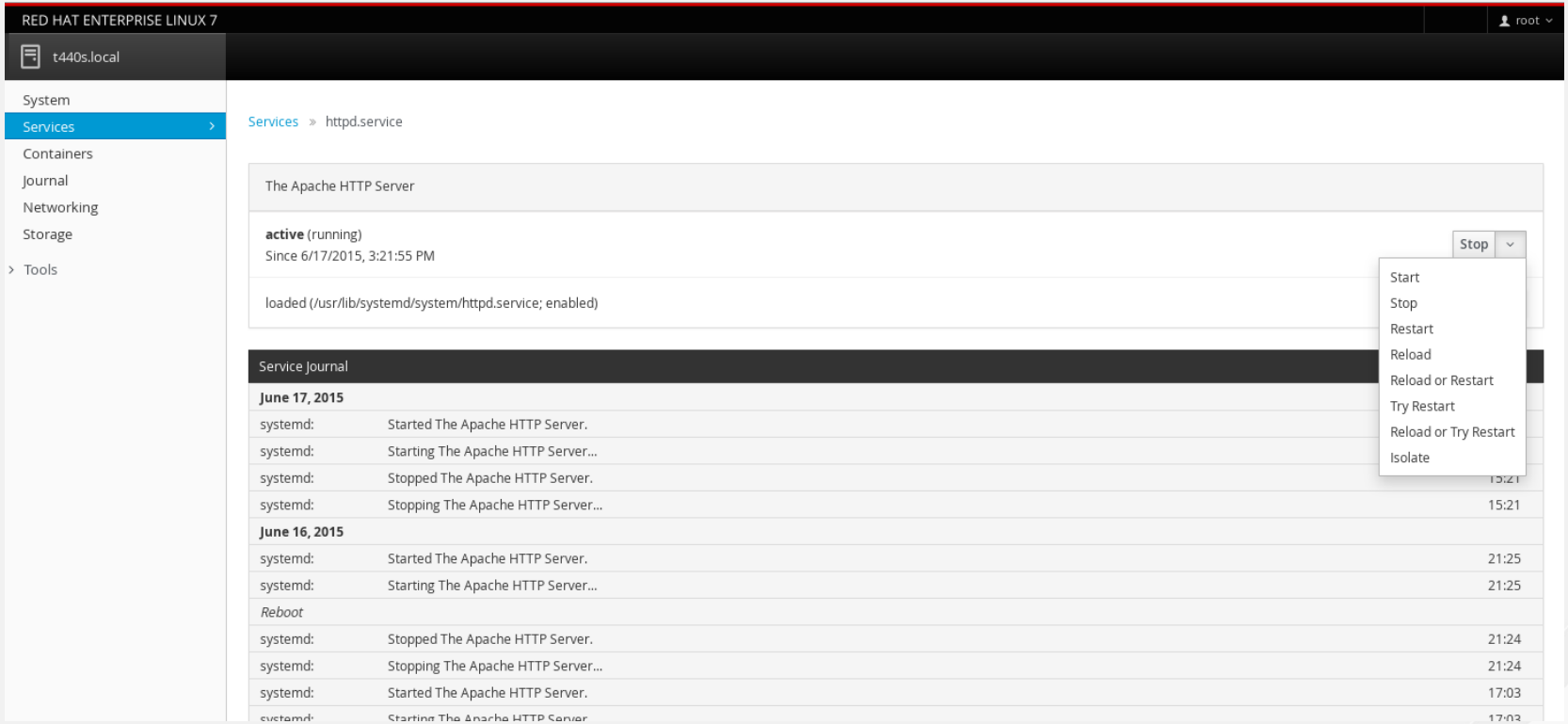
tftp.service

```
[Unit]
Description=Tftp Server

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

```
man systemd.socket
```

Cockpit - Linux Magic from Your Browser



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
```

```
man systemd.socket
```

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 -v /
```

```
man systemd.timer
```

I don’t want to live in a world without cron and xinentd!

CUSTOMIZING UNITS

Customizing Units: Viewing

- The hard way: `cat /usr/lib/systemd/system/httpd.service`
- The easy way: `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
```

Customizing Units: Available options

- 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**!

Customizing Units: Drop-in Manually

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] ← Remember the 'S' is capitalized

     Restart=always

     CPUAffinity=0 1 2 3

     OOMScoreAdjust=-1000
3) Notify systemd of the changes
   - systemctl daemon-reload

Customizing Units: Drop-in via systemctl

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: Pass --full to create a copy of the original unit file

Customizing Units: Viewing 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.
```

I don’t care!!

“Systemd? More like $#!t-stemd”

http://wizardofbits.tumblr.com/post/45232318557/systemd-more-like-shit-stemd

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
  - (see man:capabilities(7) for details)

Security & Sandboxing?!

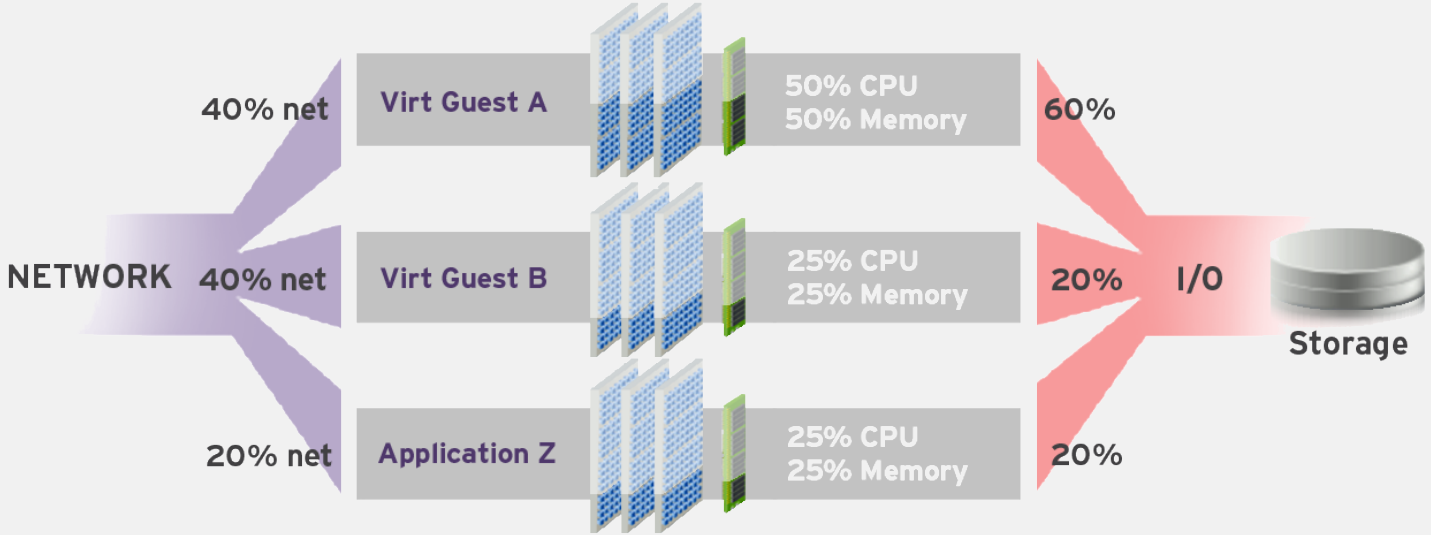
“systemd is a slap in the face to the Unix philosophy”

http://without-systemd.org

RESOURCE MANAGEMENT SLICES, SCOPES, SERVICES

Control Groups Made Simple

Resource Management with cgroups can reduce contention and improve throughput, predictability, and scalability.



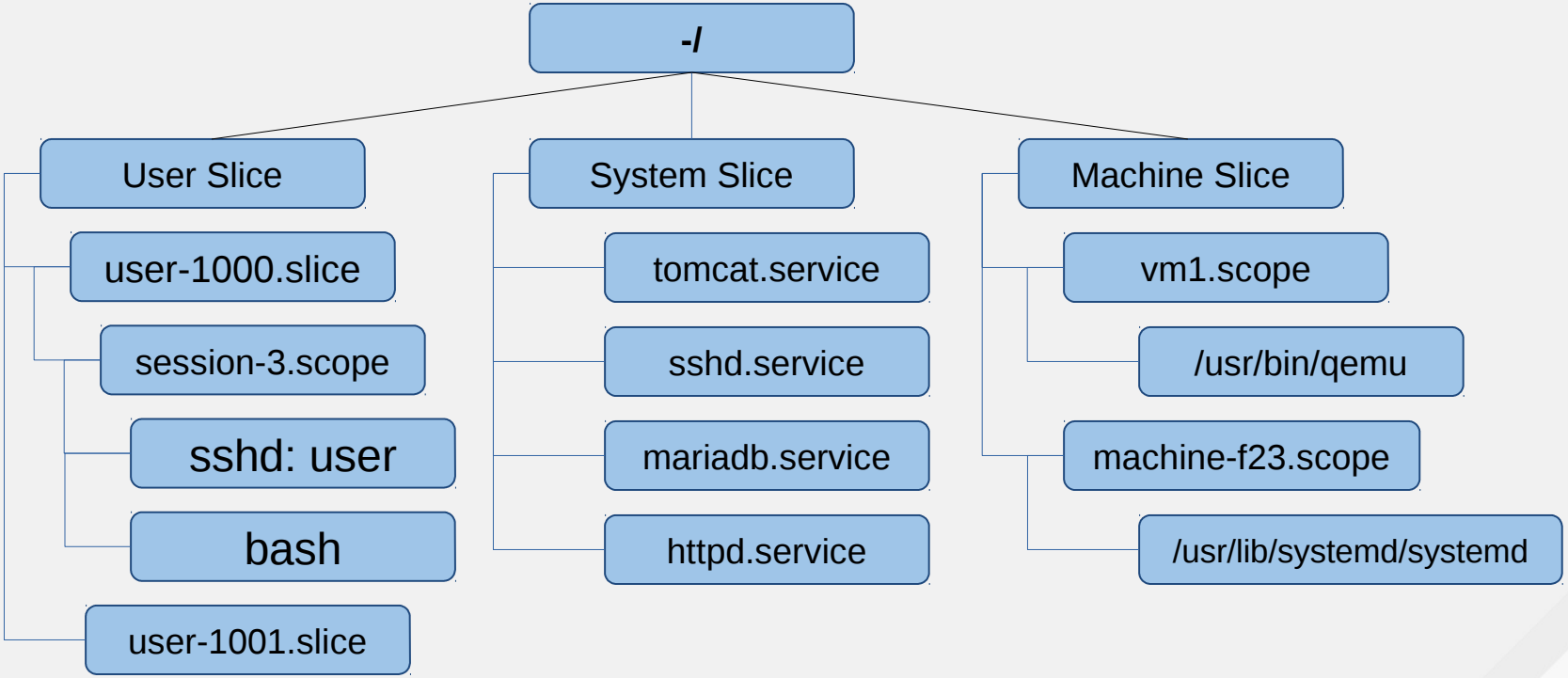
Slices, Scopes, Services

- **Slice** – Unit type for creating the cgroup hierarchy for resource management.
- **Scope** – Organizational unit that groups a daemon’s worker processes.
- **Service** – Process or group of processes controlled by systemd

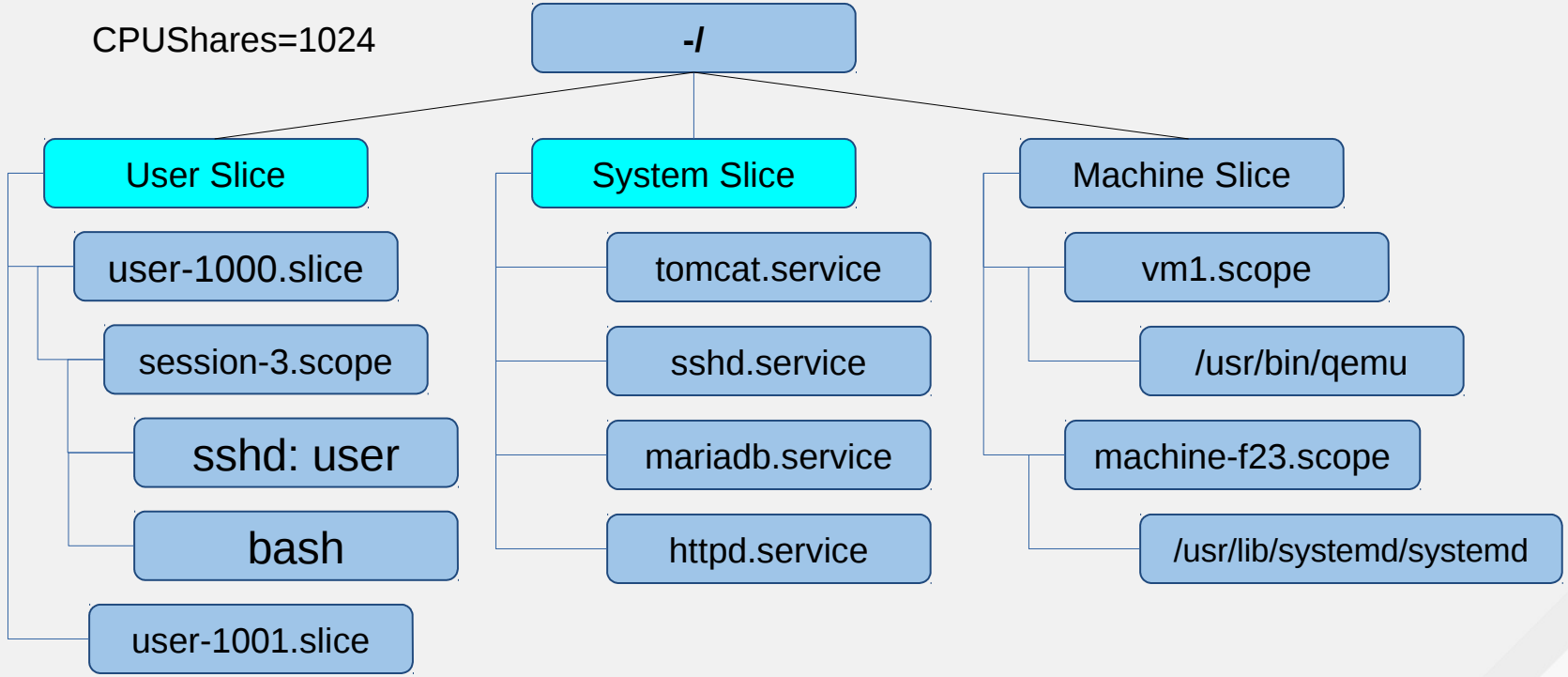
Slices, Scopes, Services

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

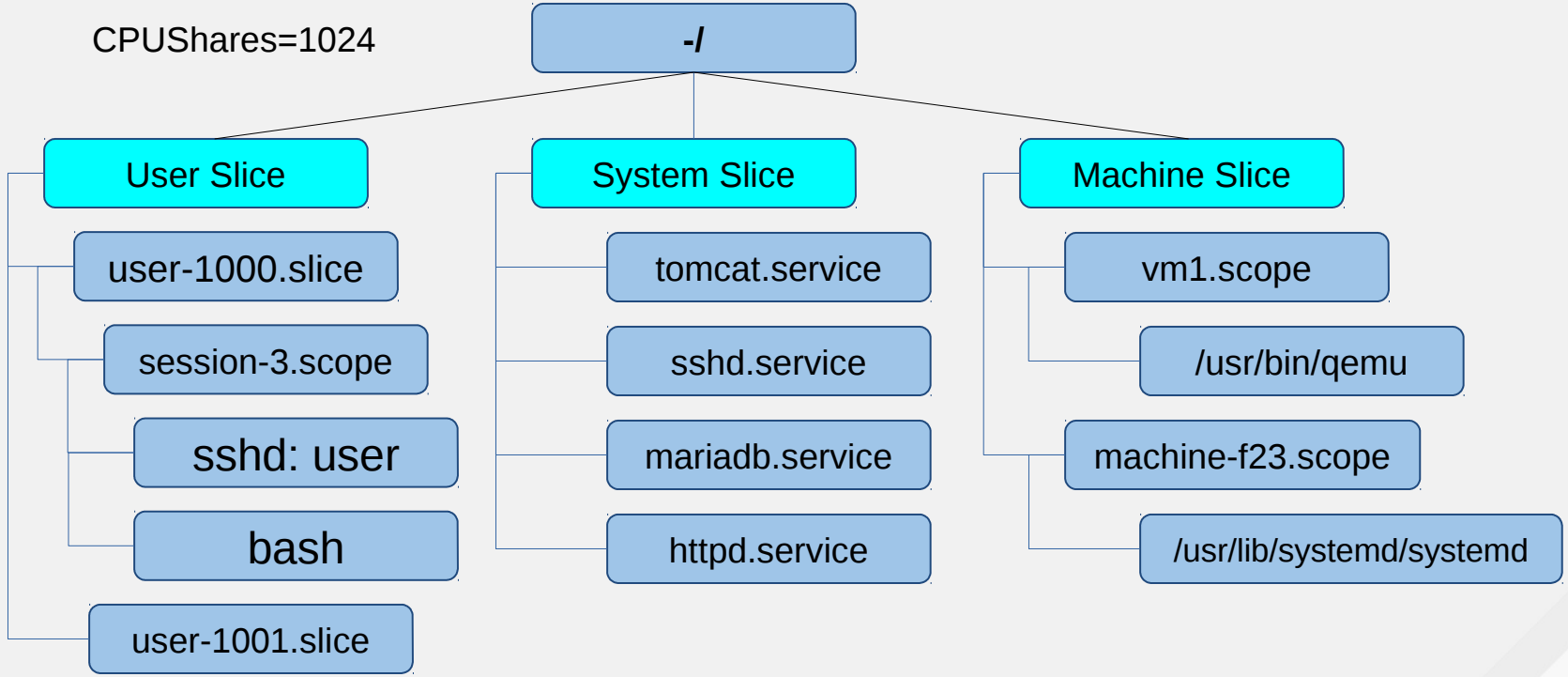
Slices, Scopes, Services



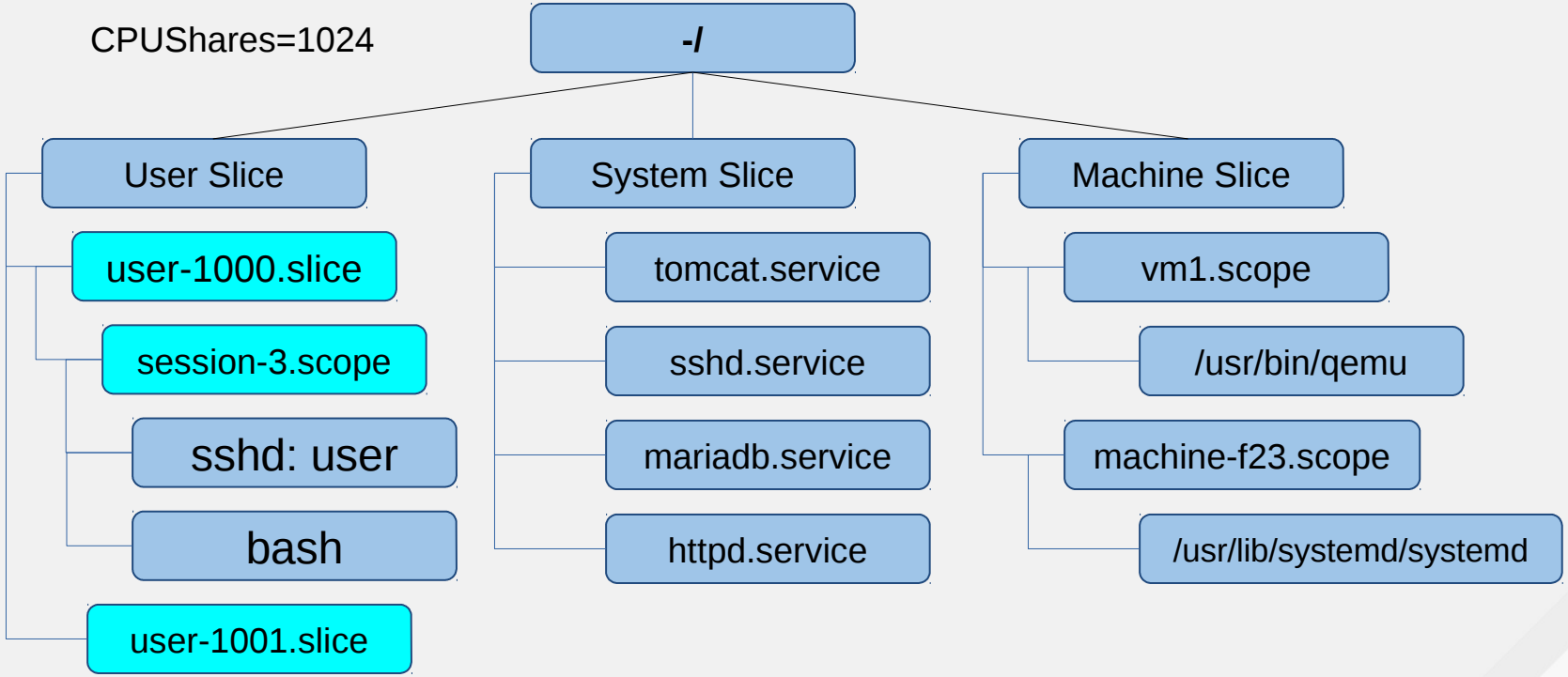
Slices, Scopes, Services



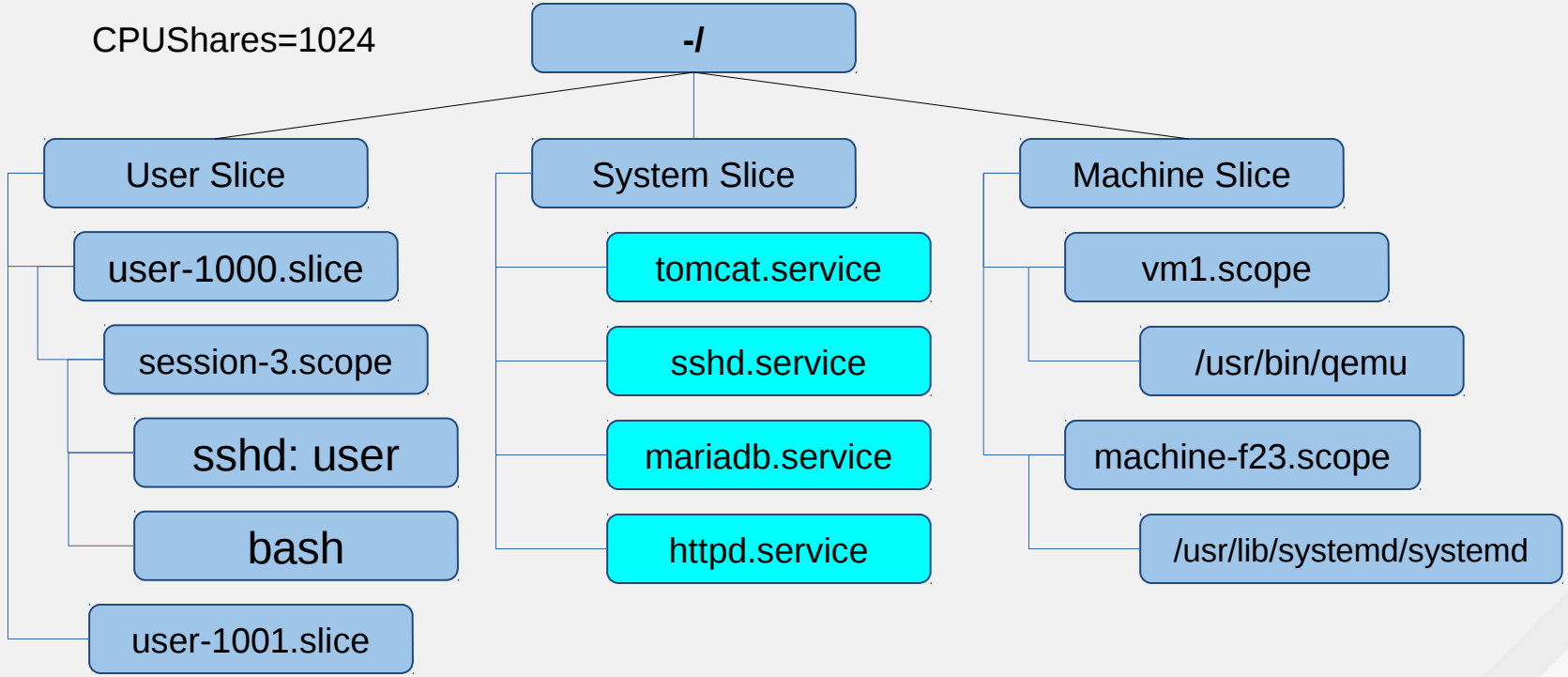
Slices, Scopes, Services



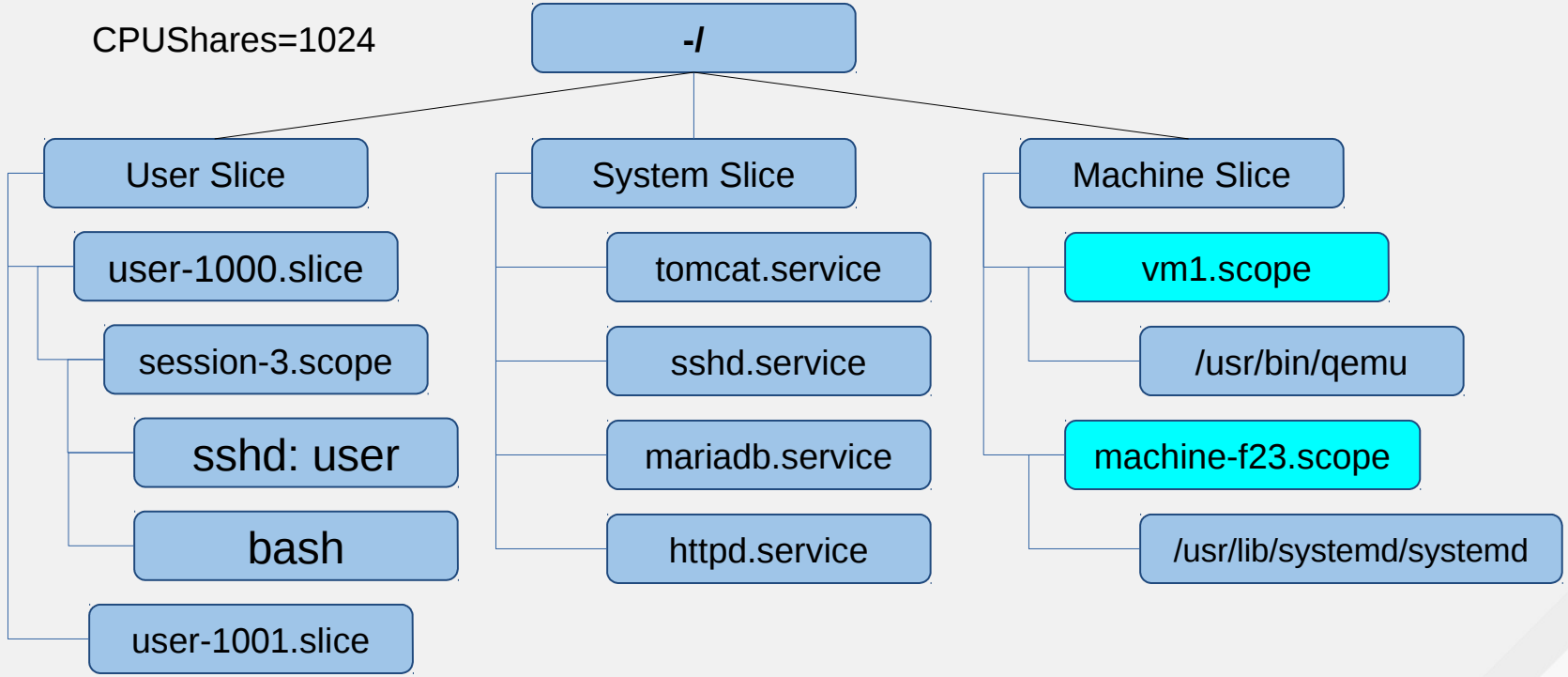
Slices, Scopes, Services



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	-	-	-	-

Usable cgroups?!

“SystemD is broken by design!”

https://ewontfix.com/14/

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
- StartupCPUShares= Applies only during the system startup
- 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

Resource Management – PIDs

- TasksAccounting=1
- TasksMax=
  - assigns the maximum number of tasks the unit can create.

- Coming soon in RHEL 7.4

https://www.kernel.org/doc/Documentation/cgroup-v1/pids.txt

“Ah nuts! ...my kiddie scripts depend on fork-bombs!”

-NoOne Ever

Additional 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?

RED HAT SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.

Customizing Units: Drop-ins

- systemctl daemon-reload is safe to run
  - Note: some service options will require the service to restart before taking 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/50-httpd.conf
[EXTENDED]   /usr/lib/systemd/system/httpd.service → /etc/systemd/system/httpd.service.d/90-CPUShares.conf
```

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

Boot Troubleshooting

- Early boot shell on tty9
  - `systemctl enable debug-shell.service`
  - `ln -s /usr/lib/systemd/system/debug-shell.service \ /etc/systemd/system/sysinit.target.wants/`
- `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/Softwar
systemd/Debugging/