LINUX BLIMP
One more Linux Magazine

APRIL 6, 2019 BY ROOT

How to Manage ‘Systemd’ Services and Units Using ‘Systemctl’ in Linux

Systemctl is a **systemd** utility which is responsible for Controlling the **systemd** system and service manager.

Systemd is a collection of system management daemons, utilities and libraries which serves as a replacement of **System V init** daemon. Systemd functions as central management and configuration platform for UNIX like system.

In the Linux Ecosystem **Systemd** has been implemented on most of the standard Linux Distribution with a few exception. Systemd is the parent Process of all other daemons oftenly but not always.

Manage Linux Services Using Systemctl

This article aims at throwing light on "**How to control System and Services**" on a system running systemd.

Starting with Systemtd and Systemctl Basics

1. First check if **systemd** is installed on your system or not and what is the version of currently installed Systemd?

```
# systemd --version

systemd 215
+PAM +AUDIT +SELINUX +IMA +SYSVINIT +LIBCRYPTSETUP +GCRYPT +ACL +XZ -SECCO
```

It's clear from the above example, that we have systemd **215** version Installed.

2. Check where the binaries and libraries of **systemd** and **systemctl** are installed.

```
# whereis systemd
systemd: /usr/lib/systemd /etc/systemd /usr/share/systemd /usr/share/man/m


# whereis systemctl
systemctl: /usr/bin/systemctl /usr/share/man/man1/systemctl.1.gz
```

3. Check whether **systemd** is running or not.

```
# ps -eaf | grep [s]ystemd

root         1     0  0 16:27 ?        00:00:00 /usr/lib/systemd/systemd -
```

```
root       444     1  0 16:27 ?        00:00:00 /usr/lib/systemd/systemd-j
root       469     1  0 16:27 ?        00:00:00 /usr/lib/systemd/systemd-u
root       555     1  0 16:27 ?        00:00:00 /usr/lib/systemd/systemd-l
dbus       556     1  0 16:27 ?        00:00:00 /bin/dbus-daemon --system
```

Notice: systemd is running as parent daemon (**PID=1**). In the above command **ps** with (**-e**) select all Processes, (**-a**) select all processes except session leaders and (**-f**) for full format listing (i.e. **-eaf**).

Also note the square brackets in the above example and rest of the examples to follow. Square Bracket expression is part of grep's character class pattern matching.

4. Analyze systemd boot process.

```
# systemd-analyze
Startup finished in 487ms (kernel) + 2.776s (initrd) + 20.229s (userspace)
```

5. Analyze time taken by each process at boot.

```
# systemd-analyze blame

8.565s mariadb.service
7.991s webmin.service
6.095s postfix.service
4.311s httpd.service
3.926s firewalld.service
3.780s kdump.service
3.238s tuned.service
1.712s network.service
1.394s lvm2-monitor.service
1.126s systemd-logind.service
....
```

6. Analyze critical chain at boot.

```
# systemd-analyze critical-chain

The time after the unit is active or started is printed after the "@" char
The time the unit takes to start is printed after the "+" character.

multi-user.target @20.222s
└─mariadb.service @11.657s +8.565s
  └─network.target @11.168s
    └─network.service @9.456s +1.712s
      └─NetworkManager.service @8.858s +596ms
        └─firewalld.service @4.931s +3.926s
          └─basic.target @4.916s
            └─sockets.target @4.916s
              └─dbus.socket @4.916s
                └─sysinit.target @4.905s
                  └─systemd-update-utmp.service @4.864s +39ms
                    └─auditd.service @4.563s +301ms
                      └─systemd-tmpfiles-setup.service @4.485s +69ms
                        └─rhel-import-state.service @4.342s +142ms
                          └─local-fs.target @4.324s
                            └─boot.mount @4.286s +31ms
                              └─systemd-fsck@dev-disk-by\x2duuid-79f594ad\
                                └─dev-disk-by\x2duuid-79f594ad\x2da332\x2d
```

Important: Systemctl accepts services (**.service**), mount point (**.mount**), sockets (**.socket**) and devices (**.device**) as units.

7. List all the available units.

```
# systemctl list-unit-files

UNIT FILE                                   STATE
proc-sys-fs-binfmt_misc.automount           static
dev-hugepages.mount                         static
```

```
dev-mqueue.mount                                static
proc-sys-fs-binfmt_misc.mount                   static
sys-fs-fuse-connections.mount                   static
sys-kernel-config.mount                         static
sys-kernel-debug.mount                          static
tmp.mount                                       disabled
brandbot.path                                   disabled
.....
```

8. List all running units.

```
# systemctl list-units

UNIT                                            LOAD   ACTIVE SUB       DESCRI
proc-sys-fs-binfmt_misc.automount               loaded active waiting   Arbitr
sys-devices-pc...0-1:0:0:0-block-sr0.device loaded active plugged   VBOX_C
sys-devices-pc...:00:03.0-net-enp0s3.device loaded active plugged   PRO/10
sys-devices-pc...00:05.0-sound-card0.device loaded active plugged   82801A
sys-devices-pc...:0:0-block-sda-sda1.device loaded active plugged   VBOX_H
sys-devices-pc...:0:0-block-sda-sda2.device loaded active plugged   LVM PV
sys-devices-pc...0-2:0:0:0-block-sda.device loaded active plugged   VBOX_H
sys-devices-pl...erial8250-tty-ttyS0.device loaded active plugged   /sys/d
sys-devices-pl...erial8250-tty-ttyS1.device loaded active plugged   /sys/d
sys-devices-pl...erial8250-tty-ttyS2.device loaded active plugged   /sys/d
sys-devices-pl...erial8250-tty-ttyS3.device loaded active plugged   /sys/d
sys-devices-virtual-block-dm\x2d0.device    loaded active plugged   /sys/d
sys-devices-virtual-block-dm\x2d1.device    loaded active plugged   /sys/d
sys-module-configfs.device                  loaded active plugged   /sys/m
...
```

9. List all failed units.

```
# systemctl --failed

UNIT          LOAD   ACTIVE SUB    DESCRIPTION
kdump.service loaded failed failed Crash recovery kernel arming
```

```
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

1 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

10. Check if a Unit (**cron.service**) is enabled or not?.

```
# systemctl is-enabled crond.service

enabled
```

11. Check whether a Unit or Service is running or not?.

```
# systemctl status firewalld.service

firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
   Active: active (running) since Tue 2015-04-28 16:27:55 IST; 34min ago
 Main PID: 549 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─549 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Apr 28 16:27:51 tecmint systemd[1]: Starting firewalld - dynamic firewall
Apr 28 16:27:55 tecmint systemd[1]: Started firewalld - dynamic firewall d
```

Control and Manage Services Using Systemctl

12. List all services (including enabled and disabled).

```
# systemctl list-unit-files --type=service

UNIT FILE                                   STATE
arp-ethers.service                          disabled
auditd.service                              enabled
autovt@.service                             disabled
blk-availability.service                    disabled
brandbot.service                            static
collectd.service                            disabled
console-getty.service                       disabled
console-shell.service                       disabled
cpupower.service                            disabled
crond.service                               enabled
dbus-org.fedoraproject.FirewallD1.service   enabled
....
```

13. How do I start, restart, stop, reload and check the status of a service (**httpd.service**) in Linux.

```
# systemctl start httpd.service
# systemctl restart httpd.service
# systemctl stop httpd.service
# systemctl reload httpd.service
# systemctl status httpd.service

httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
   Active: active (running) since Tue 2015-04-28 17:21:30 IST; 6s ago
  Process: 2876 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=
 Main PID: 2881 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─2881 /usr/sbin/httpd -DFOREGROUND
           ├─2884 /usr/sbin/httpd -DFOREGROUND
           ├─2885 /usr/sbin/httpd -DFOREGROUND
           ├─2886 /usr/sbin/httpd -DFOREGROUND
           ├─2887 /usr/sbin/httpd -DFOREGROUND
```

```
            └─2888 /usr/sbin/httpd -DFOREGROUND

Apr 28 17:21:30 tecmint systemd[1]: Starting The Apache HTTP Server...
Apr 28 17:21:30 tecmint httpd[2881]: AH00558: httpd: Could not reliably de
Apr 28 17:21:30 tecmint systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

Note: When we use commands like **start**, **restart**, **stop** and **reload** with systemctl, we will not get any output on the terminal, only **status** command will print the output.

14. How to active a service and enable or disable a service at boot time (auto start service at system boot).

```
# systemctl is-active httpd.service
# systemctl enable httpd.service
# systemctl disable httpd.service
```

15. How to mask (making it impossible to start) or unmask a service (**httpd.service**).

```
# systemctl mask httpd.service
ln -s '/dev/null' '/etc/systemd/system/httpd.service'

# systemctl unmask httpd.service
rm '/etc/systemd/system/httpd.service'
```

16. How to a Kill a service using systemctl command.

```
# systemctl kill httpd
# systemctl status httpd
```

```
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
   Active: failed (Result: exit-code) since Tue 2015-04-28 18:01:42 IST; 2
 Main PID: 2881 (code=exited, status=0/SUCCESS)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:

Apr 28 17:37:29 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 17:37:29 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 17:37:39 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 17:37:39 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 17:37:49 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 17:37:49 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 17:37:59 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 17:37:59 tecmint systemd[1]: httpd.service: Got notification messag
Apr 28 18:01:42 tecmint systemd[1]: httpd.service: control process exited,
Apr 28 18:01:42 tecmint systemd[1]: Unit httpd.service entered failed stat
Hint: Some lines were ellipsized, use -l to show in full.
```

Control and Manage Mount Points using Systemctl

17. List all system mount points.

```
# systemctl list-unit-files --type=mount

UNIT FILE                      STATE
dev-hugepages.mount            static
dev-mqueue.mount               static
proc-sys-fs-binfmt_misc.mount static
sys-fs-fuse-connections.mount static
sys-kernel-config.mount        static
sys-kernel-debug.mount         static
tmp.mount                      disabled
```

18. How do I mount, unmount, remount, reload system mount points and also check the status of mount points on the system.

```
# systemctl start tmp.mount
# systemctl stop tmp.mount
# systemctl restart tmp.mount
# systemctl reload tmp.mount
# systemctl status tmp.mount

tmp.mount - Temporary Directory
   Loaded: loaded (/usr/lib/systemd/system/tmp.mount; disabled)
   Active: active (mounted) since Tue 2015-04-28 17:46:06 IST; 2min 48s ag
    Where: /tmp
     What: tmpfs
     Docs: man:hier(7)
           http://www.freedesktop.org/wiki/Software/systemd/APIFileSystems
  Process: 3908 ExecMount=/bin/mount tmpfs /tmp -t tmpfs -o mode=1777,stri

Apr 28 17:46:06 tecmint systemd[1]: Mounting Temporary Directory...
Apr 28 17:46:06 tecmint systemd[1]: tmp.mount: Directory /tmp to mount ove
Apr 28 17:46:06 tecmint systemd[1]: Mounted Temporary Directory.
```

19. How to active, enable or disable a mount point at boot time (auto mount at system boot).

```
# systemctl is-active tmp.mount
# systemctl enable tmp.mount
# systemctl disable  tmp.mount
```

20. How to mask (making it impossible to start) or unmask a mount points in Linux.

```
# systemctl mask tmp.mount

ln -s '/dev/null' '/etc/systemd/system/tmp.mount'

# systemctl unmask tmp.mount
```

```
rm '/etc/systemd/system/tmp.mount'
```

Control and Manage Sockets using Systemctl

21. List all available system sockets.

```
# systemctl list-unit-files --type=socket

UNIT FILE                    STATE
dbus.socket                  static
dm-event.socket              enabled
lvm2-lvmetad.socket          enabled
rsyncd.socket                disabled
sshd.socket                  disabled
syslog.socket                static
systemd-initctl.socket       static
systemd-journald.socket      static
systemd-shutdownd.socket     static
systemd-udevd-control.socket static
systemd-udevd-kernel.socket  static

11 unit files listed.
```

22. How do I start, restart, stop, reload and check the status of a socket (example: **cups.socket**) in Linux.

```
# systemctl start cups.socket
# systemctl restart cups.socket
# systemctl stop cups.socket
# systemctl reload cups.socket
# systemctl status cups.socket
```

```
cups.socket - CUPS Printing Service Sockets
   Loaded: loaded (/usr/lib/systemd/system/cups.socket; enabled)
   Active: active (listening) since Tue 2015-04-28 18:10:59 IST; 8s ago
   Listen: /var/run/cups/cups.sock (Stream)

Apr 28 18:10:59 tecmint systemd[1]: Starting CUPS Printing Service Sockets
Apr 28 18:10:59 tecmint systemd[1]: Listening on CUPS Printing Service Soc
```

23. How to active a socket and enable or disable at boot time (auto start socket at system boot).

```
# systemctl is-active cups.socket
# systemctl enable cups.socket
# systemctl disable cups.socket
```

24. How to mask (making it impossible to start) or unmask a socket (cups.socket).

```
# systemctl mask cups.socket
ln -s '/dev/null' '/etc/systemd/system/cups.socket'

# systemctl unmask cups.socket
rm '/etc/systemd/system/cups.socket'
```

CPU Utilization (Shares) of a Service

25. Get the current CPU Shares of a Service (say httpd).

```
# systemctl show -p CPUShares httpd.service

CPUShares=1024
```

Note: The default each service has a CPUShare = **1024**. You may increase/decrease CPU share of a process.

26. Limit the CPU Share of a service (httpd.service) to **2000** CPUShares/

```
# systemctl set-property httpd.service CPUShares=2000
# systemctl show -p CPUShares httpd.service

CPUShares=2000
```

Note: When you set **CPUShare** for a service, a directory with the name of service is created (**httpd.service.d**) which contains a file **90-CPUShares.conf** which contains the CPUShare Limit information. You may view the file as:

```
# vi /etc/systemd/system/httpd.service.d/90-CPUShares.conf

[Service]
CPUShares=2000
```

27. Check all the configuration details of a service.

```
# systemctl show httpd

Id=httpd.service
Names=httpd.service
Requires=basic.target
Wants=system.slice
WantedBy=multi-user.target
Conflicts=shutdown.target
Before=shutdown.target multi-user.target
After=network.target remote-fs.target nss-lookup.target systemd-journald.s
Description=The Apache HTTP Server
LoadState=loaded
```

```
ActiveState=active
SubState=running
FragmentPath=/usr/lib/systemd/system/httpd.service
....
```

28. Analyze critical chain for a services(httpd).

```
# systemd-analyze critical-chain httpd.service

The time after the unit is active or started is printed after the "@" char
The time the unit takes to start is printed after the "+" character.

httpd.service +142ms
└─network.target @11.168s
  └─network.service @9.456s +1.712s
    └─NetworkManager.service @8.858s +596ms
      └─firewalld.service @4.931s +3.926s
        └─basic.target @4.916s
          └─sockets.target @4.916s
            └─dbus.socket @4.916s
              └─sysinit.target @4.905s
                └─systemd-update-utmp.service @4.864s +39ms
                  └─auditd.service @4.563s +301ms
                    └─systemd-tmpfiles-setup.service @4.485s +69ms
                      └─rhel-import-state.service @4.342s +142ms
                        └─local-fs.target @4.324s
                          └─boot.mount @4.286s +31ms
                            └─systemd-fsck@dev-disk-by\x2duuid-79f594ad\x2
                              └─dev-disk-by\x2duuid-79f594ad\x2da332\x2d47
```

29. Get a list of dependencies for a services (httpd).

```
# systemctl list-dependencies httpd.service

httpd.service
├─system.slice
```

```
└─basic.target
  ├─firewalld.service
  ├─microcode.service
  ├─rhel-autorelabel-mark.service
  ├─rhel-autorelabel.service
  ├─rhel-configure.service
  ├─rhel-dmesg.service
  ├─rhel-loadmodules.service
  ├─paths.target
  ├─slices.target
  | ├─-.slice
  | └─system.slice
  ├─sockets.target
  | ├─dbus.socket
....
```

30. List control groups hierarchically.

```
# systemd-cgls

├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 23
├─user.slice
| └─user-0.slice
|   └─session-1.scope
|     ├─2498 sshd: root@pts/0
|     ├─2500 -bash
|     ├─4521 systemd-cgls
|     └─4522 systemd-cgls
└─system.slice
  ├─httpd.service
  | ├─4440 /usr/sbin/httpd -DFOREGROUND
  | ├─4442 /usr/sbin/httpd -DFOREGROUND
  | ├─4443 /usr/sbin/httpd -DFOREGROUND
  | ├─4444 /usr/sbin/httpd -DFOREGROUND
  | ├─4445 /usr/sbin/httpd -DFOREGROUND
  | └─4446 /usr/sbin/httpd -DFOREGROUND
  ├─polkit.service
  | └─721 /usr/lib/polkit-1/polkitd --no-debug
```

```
....
```

31. List control group according to CPU, memory, Input and Output.

```
# systemd-cgtop

Path                                                                Tasks

/                                                                      83
/system.slice                                                           -
/system.slice/mariadb.service                                           2
/system.slice/tuned.service                                             1
/system.slice/httpd.service                                             6
/system.slice/NetworkManager.service                                    1
/system.slice/atop.service                                              1
/system.slice/atopacct.service                                          1
/system.slice/auditd.service                                            1
/system.slice/crond.service                                             1
/system.slice/dbus.service                                              1
/system.slice/firewalld.service                                         1
/system.slice/lvm2-lvmetad.service                                      1
/system.slice/polkit.service                                            1
/system.slice/postfix.service                                           3
/system.slice/rsyslog.service                                           1
/system.slice/system-getty.slice/getty@tty1.service                     1
/system.slice/systemd-journald.service                                  1
/system.slice/systemd-logind.service                                    1
/system.slice/systemd-udevd.service                                     1
/system.slice/webmin.service                                            1
/user.slice/user-0.slice/session-1.scope                                3
```

Control System Runlevels

32. How to start system rescue mode.

```
# systemctl rescue

Broadcast message from root@tecmint on pts/0 (Wed 2015-04-29 11:31:18 IST)

The system is going down to rescue mode NOW!
```

33. How to enter into emergency mode.

```
# systemctl emergency

Welcome to emergency mode! After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" to try agai
to boot into default mode.
```

34. List current run levels in use.

```
# systemctl get-default

multi-user.target
```

35. How to start Runlevel 5 aka graphical mode.

```
# systemctl isolate runlevel5.target
OR
# systemctl isolate graphical.target
```

36. How to start Runlevel 3 aka multiuser mode (commandline).

```
# systemctl isolate runlevel3.target
OR
```

```
# systemctl isolate multiuser.target
```

36. How to set multiusermode or graphical mode as default runlevel.

```
# systemctl set-default runlevel3.target

# systemctl set-default runlevel5.target
```

37. How to reboot, halt, suspend, hibernate or put system in hybrid-sleep.

```
# systemctl reboot

# systemctl halt

# systemctl suspend

# systemctl hibernate

# systemctl hybrid-sleep
```

For those who may not be aware of runlevels and what it does.

1. **Runlevel 0** : Shut down and Power off the system.
2. **Runlevel 1** : Rescue?Maintainance Mode.
3. **Runlevel 3** : multiuser, no-graphic system.
4. **Runlevel 4** : multiuser, no-graphic system.
5. **Runlevel 5** : multiuser, graphical system.
6. **Runlevel 6** : Shutdown and Reboot the machine.

That's all for now. Keep connected! Keep commenting. Don't forget to provide us with your valuable feedback in the comments below. Like and share us and help us get spread.

Source