systemd Commands Cheat Sheet

This cheat sheet presents command-line executables that are used for working with systemd. The systemd service runs on Linux to consolidate service configuration and application behavior. systemd is found in Red Hat Enterprise Linux as well as other Linux distributions.
systemctl is the application that interacts with systemd from the command line. This cheat sheet demonstrates how to use various systemctl subcommands to control the behavior of particular services running on a computer. The journalctl command, which displays information about systemd activities from its logs, is also introduced.

APPLICATION MANAGEMENT USING SYSTEMCTL COMMANDS

The subcommands in this section control the behavior of particular applications, usually services (daemons) that run in the background.

systemctl enable

```
systemctl [options] enable <service_name>
```

Enables a service. Enabling a service causes the system to start the service upon reboot or whenever a computer starts up. The enable subcommand does not start the particular service immediately. To start a service immediately using systemctl, use the systemctl start command, described later.

Example:

The systemctl enable command in this example configures the system to invoke the sshd secure shell service at system startup. This command can be useful to ensure that users can securely log in from remote systems at any time when the local system is running.
The systemctl is-enabled command that follows verifies that the sshd services is enabled.
The systemctl enable command requires administrator permissions to execute. The command can be run as a subcommand to sudo. If sudo is not used, systemctl enable prompts for the administrator password. Calling systemctl is-enabled does not require administrator permissions.

```
$ sudo systemctl enable sshd

$ systemctl is-enabled sshd
enabled
```

systemctl restart

```
systemctl [options] restart <service_name>
```

Restarts a service.
The systemctl restart command requires administrator permissions to execute. The command can be run as a subcommand to sudo. If sudo is not used, systemctl restart prompts for the administrator password.

Example:

The systemctl restart command in this example restarts the httpd web server. This command can be useful after making configuration changes to the web server, so that the changes take effect for subsequent incoming requests.

```
$ sudo systemctl restart httpd
```

systemctl start

```
systemctl [options] start <service_name>
```

Starts a service.
The systemctl start command requires administrator permissions to execute. The command can be run as a subcommand to sudo. If sudo is not used, systemctl start prompts for the administrator password.

Example:

The systemctl start command in this example starts the httpd web server. This can be useful to run a service that is not normally running. systemctl start does not cause the service to restart after the system restarts.
The systemctl status httpd command that follows verifies that the httpd service is running. Calling systemctl status does not require administrator permissions.

```
$ sudo systemctl start httpd

$ 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 Mon 2022-01-24 09:31:10 PST; 2s ago
     Docs: man:httpd.service(8)
 Main PID: 41263 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 213 (limit: 49364)
   Memory: 36.7M
   CGroup: /system.slice/httpd.service
           ├─41263 /usr/sbin/httpd -DFOREGROUND
           ├─41264 /usr/sbin/httpd -DFOREGROUND
           ├─41265 /usr/sbin/httpd -DFOREGROUND
           ├─41266 /usr/sbin/httpd -DFOREGROUND
           └─41267 /usr/sbin/httpd -DFOREGROUND

Jan 24 09:31:09 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Jan 24 09:31:10 localhost.localdomain httpd[41263]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, us>
Jan 24 09:31:10 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Jan 24 09:31:10 localhost.localdomain httpd[41263]: Server configured, listening on: port 80
```

systemctl status

```
systemctl [options] status <service_name>
```

Reports status information about a service.

Example:

The systemctl status command in this example reports status information about the sshd service. systemctl status also displays information about the service's activities via log entries that follow the status information. Earlier examples in this cheat sheet show reasons to use this command.

```
$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-01-21 10:13:49 PST; 2 days ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1026 (sshd)
    Tasks: 1 (limit: 49364)
   Memory: 4.5M
   CGroup: /system.slice/sshd.service
           └─1026 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-cbc,aes128-
gcm@openssh.c>

Jan 21 10:13:49 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Jan 21 10:13:49 localhost.localdomain sshd[1026]: Server listening on 0.0.0.0 port 22.
Jan 21 10:13:49 localhost.localdomain sshd[1026]: Server listening on :: port 22.
Jan 21 10:13:49 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Jan 21 10:23:25 localhost.localdomain sshd[2136]: Accepted password for reselbob from 192.168.86.20 port 59909 ssh2
Jan 21 10:23:25 localhost.localdomain sshd[2136]: pam_unix(sshd:session): session opened for user reselbob by (uid=0)
Jan 24 08:42:43 localhost.localdomain sshd[40279]: Accepted password for reselbob from 192.168.86.20 port 61945 ssh2
Jan 24 08:42:43 localhost.localdomain sshd[40279]: pam_unix(sshd:session): session opened for user reselbob by (uid=0)
lines 1-19/19 (END)
```

systemctl stop

```
systemctl [options] stop <service_name>
```

Stops a service. The systemctl stop command requires administrator permissions to execute. The command can be run as a subcommand to sudo. If sudo is not used, systemctl stop prompts for the administrator password.

Example:
The systemctl stop command in this example stops the httpd service. This command can be useful if you have to stop a service in order to backup its data, because you think it is being attacked by a malicious intruder, or for any other reason. The systemctl status httpd command that follows reports the status.

```
$ systemctl stop httpd

$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Mon 2022-01-24 09:56:53 PST; 3s ago
     Docs: man:httpd.service(8)
  Process: 1262 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
 Main PID: 1262 (code=exited, status=0/SUCCESS)
   Status: "Running, listening on: port 80"

Jan 24 09:32:27 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Jan 24 09:32:34 localhost.localdomain httpd[1262]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, usi>
Jan 24 09:41:29 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Jan 24 09:41:29 localhost.localdomain httpd[1262]: Server configured, listening on: port 80
Jan 24 09:56:52 localhost.localdomain systemd[1]: Stopping The Apache HTTP Server...
Jan 24 09:56:53 localhost.localdomain systemd[1]: httpd.service: Succeeded.
Jan 24 09:56:53 localhost.localdomain systemd[1]: Stopped The Apache HTTP Server.
```

Note the following line in the status output, which shows that the httpd service is inactive:

```
Active: inactive (dead) since Mon 2022-01-24 09:56:53 PST; 3s ago
```

COMPUTER CONTROL COMMANDS

The subcommands in this section reboot and shut down a computer.

systemctl poweroff

```
systemctl [options] poweroff
```

Shuts down the computer or virtual machine.
The systemctl poweroff command requires administrator permissions to execute.The command can be run as a subcommand to sudo. If sudo in not used, systemctl poweroff prompts for the administrator password.

Example:
The systemctl poweroff command in this example shuts down the computer or virtual machine.

```
sudo systemctl poweroff
```

systemctl reboot

```
systemctl [options] reboot
```

Shuts down the computer or virtual machine and immediately restarts it.

Example:
The systemctl reboot command in this example reboots the computer or virtual machine. The -i option ensures a shutdown by ignoring logged-in users and inhibitors (programs that prevent a shutdown in order to complete some long-running activity).
The systemctl reboot command requires administrator permissions to execute. The command can be run as a subcommand to sudo. If sudo is not used, systemctl reboot prompts for the administrator password.

```
sudo systemctl -i reboot
```

SYSTEM INFORMATION COMMANDS

The following shows how to use the journalctl and systemctl programs to get information about services running under systemd.

journalctl

```
journalctl [options]
```

journalctl works with systemd's logging capabilities. systemd stores the system, boot, and kernel log files in a central location in a binary format. journalctl presents information in the central logging system as human-readable text.

Example:

The journalctl command in this example displays everything stored recently in the log by systemd. The --follow option causes only the most recent journal entries to be displayed.

```
$ journalctl --follow
-- Logs begin at Mon 2022-01-24 09:31:39 PST. --
Jan 24 10:01:20 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Jan 24 10:01:20 localhost.localdomain httpd[2813]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using
localhost.localdomain. Set the 'ServerName' directive globally to suppress this message
Jan 24 10:01:20 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Jan 24 10:01:20 localhost.localdomain polkitd[876]: Unregistered Authentication Agent for unix-process:2787:124099 (system bus name :1.333,
object path /org/freedesktop/PolicyKit1/AuthenticationAgent, locale en_US.UTF-8) (disconnected from bus)
Jan 24 10:01:20 localhost.localdomain httpd[2813]: Server configured, listening on: port 80
Jan 24 10:03:29 localhost.localdomain systemd[1]: Starting dnf makecache...
Jan 24 10:03:34 localhost.localdomain dnf[3052]: Updating Subscription Management repositories.
Jan 24 10:03:35 localhost.localdomain dnf[3052]: Metadata cache refreshed recently.
Jan 24 10:03:35 localhost.localdomain systemd[1]: dnf-makecache.service: Succeeded.
Jan 24 10:03:35 localhost.localdomain systemd[1]: Started dnf makecache.
```

systemctl list-sockets

```
systemctl [options] list-sockets <unit_name_pattern>
```

Lists the sockets in memory available for interprocess communication (IPC).

Example:

The systemctl list-sockets command in this example lists the sockets in memory. By providing the pattern systemd*, the command shows only sockets that have a unit name beginning with the characters systemd. Also, the option --show-types is used to display the TYPE column in the output.

```
$ systemctl --show-types list-sockets 'systemd*'
LISTEN                   TYPE          UNIT                        ACTIVATES
/run/initctl             FIFO          systemd-initctl.socket      systemd-initctl.service
/run/systemd/coredump    SequentialPacket systemd-coredump.socket  systemd-coredump@0.service
/run/systemd/journal/dev-log Datagram  systemd-journald-dev-log.socket systemd-journald.service
/run/systemd/journal/socket  Datagram  systemd-journald.socket     systemd-journald.service
/run/systemd/journal/stdout  Stream    systemd-journald.socket     systemd-journald.service
/run/udev/control        SequentialPacket systemd-udevd-control.socket systemd-udevd.service
kobject-uevent 1         Netlink       systemd-udevd-kernel.socket systemd-udevd.service

7 sockets listed.
Pass --all to see loaded but inactive sockets, too.
```

systemctl list-units

```
systemctl [options] list-units <pattern>
```

Lists units that systemd has in memory. A *unit* refers to any resource that systemd knows how to operate on and manage. Units are listed with the following columns:

UNIT: Name of the unit

LOAD: Indicates whether the unit is loaded

ACTIVE: Indicates whether the unit is active

SUB: Indicates low-level activation state; for example: mounted or running

DESCRIPTION: Describes the service or unit

Example:
The systemctl list-units command in this example lists units in memory, using the pattern sys-k* to show only units whose names begin with the characters sys-k.

```
$ systemctl list-units 'sys-k*'
UNIT                  LOAD   ACTIVE SUB     DESCRIPTION
sys-kernel-config.mount  loaded active mounted Kernel Configuration File System
sys-kernel-debug.mount   loaded active mounted Kernel Debug File System
sys-kernel-tracing.mount loaded active mounted /sys/kernel/tracing

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.

3 loaded units listed.
To show all installed unit files use 'systemctl list-unit-files'.
```

systemctl list-unit-files

```
systemctl [options] list-unit-files <pattern>
```

Lists a unit's associated file, which describes how systemd starts and runs a unit. Unit files are listed with the following columns:
UNIT FILE: Name of a the file STATE: State of the file. The state can be static, generated or disabled.

Example:
The systemctl list-unit-files command in this example lists unit files, using the pattern sys-* to show only filenames that begin with the characters sys-.

```
$ systemctl list-unit-files 'sys-*'
UNIT FILE                  STATE
sys-fs-fuse-connections.mount static
sys-kernel-config.mount      static
sys-kernel-debug.mount       static

3 unit files listed.
```