

Manage Systemd Services

In many Linux based operating systems such as Debian 8, Red Hat Enterprise Linux (RHEL) and CentOS 7 systemd is now the default init system and is used for service management.

Here we will cover service management with the systemctl command, which is used to control the state of the systemd system and service manager.

Systemctl Examples

Here are some examples outlining how to use the systemctl command to manage various services.

Check the Status of a Service

The current status of a service can be checked as shown below.

```
[[email protected] ~]# systemctl status chronyd.service
chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled)
   Active: active (running) since Mon 2015-08-24 15:52:20 AEST; 2 days ago
     Main PID: 718 (chronyd)
        CGroup: /system.slice/chronyd.service
                └─718 /usr/sbin/chronyd -u chrony
```

Here we can see that the chronyd.service unit is currently in the “active (running)” state, as well as that the service is enabled meaning that it is currently configured to start up on system boot. We do not need to specify .service on the end of the service each time as this is automatically placed on the end by default, it will be omitted henceforth.

Alternatively we can also check if a service is active by using ‘is-active’ which will show if the service is currently running or not.

```
[[email protected] ~]# systemctl is-active chronyd
active
```

Similarly we can also check if a service is enabled to start on boot by using ‘is-enabled’.

```
[[email protected] ~]# systemctl is-enabled chronyd
enabled
```

Starting, Stopping and Restarting Services

Systemctl can be used to start, stop and restart services as demonstrated.

Here we stop the chronyd service and confirm that it is no longer actively running.

```
[[email protected] ~]# systemctl stop chronyd
[[email protected] ~]# systemctl is-active chronyd
inactive
```

We can start the service back up and confirm that it is active once again.

```
[[email protected] ~]# systemctl start chronyd
[[email protected] ~]# systemctl is-active chronyd
active
```

Rather than perform the stop then start in two steps, we can instead restart the service. If you check the status of the service after this, the active since time will have changed as the service was restarted and the PID will also change.

```
[[email protected] ~]# systemctl restart chronyd
```

If you attempt to restart a service that is not currently actively running, it will start up. Instead we can use 'try-restart' to only perform a restart if the service is already currently running. If the service is not already running and you use 'try-restart' it will not start up.

```
[[email protected] ~]# systemctl is-active chronyd
active
[[email protected] ~]# systemctl stop chronyd
[[email protected] ~]# systemctl is-active chronyd
inactive
[[email protected] ~]# systemctl try-restart chronyd
[[email protected] ~]# systemctl is-active chronyd
inactive
[[email protected] ~]# systemctl restart chronyd
[[email protected] ~]# systemctl is-active chronyd
active
```

A service can be reloaded which will just refresh things like configuration file changes, the main process will continue to run however so the PID will not change.

```
[[email protected] ~]# systemctl reload sshd
```

Enabling and Disabling Services

If a service is enabled it will be started automatically during system boot, however if a service is disabled it will not automatically start up during system boot. It is possible for a user or another service to manually start up the disabled service.

Below we can see that the chronyd service is enabled, after disabling it the symlink is removed. Once disabled the service will still be actively running, however if the system is rebooted the service will not start up unless manually started.

```
[[email protected] ~]# systemctl is-enabled chronyd enabled [[email protected] ~]# systemctl
disable chronyd rm '/etc/systemd/system/multi-user.target.wants/chronyd.service' [[email protected]
~]# systemctl is-enabled chronyd disabled [[email protected] ~]# systemctl is-active chronyd active
The symlink will be recreated when the service is enabled to start up during system boot.
```

```
[[email protected] ~]# systemctl enable chronyd ln -s '/usr/lib/systemd/system/chronyd.service'
'/etc/systemd/system/multi-user.target.wants/chronyd.service' If you try to enable a service that is
already enabled, it will not recreate the symlink which already exists so there will be no output. We
can reset the symlink using 'reenable' which will first delete the symlink and then recreate it.
```

```
[[email protected] ~]# systemctl enable chronyd [[email protected] ~]# systemctl reenale chronyd
rm '/etc/systemd/system/multi-user.target.wants/chronyd.service' ln -s
'/usr/lib/systemd/system/chronyd.service' '/etc/systemd/system/multi-
user.target.wants/chronyd.service' To prevent a service from being started automatically and
manually by the user or other services we can mask the service. Masking the service essentially
directs the symlink to /dev/null so that when it is used, nothing happens.
```

```
[[email protected] ~]# systemctl mask chronyd ln -s '/dev/null' '/etc/systemd/system/chronyd.service'
Once the service is masked it will not start on boot and can not be manually started. If a running
process is masked it will remain active.
```

```
[[email protected] ~]# systemctl is-active chronyd active [[email protected] ~]# systemctl stop
chronyd [[email protected] ~]# systemctl start chronyd Failed to issue method call: Unit
chronyd.service is masked. To demonstrate this, the test server has been rebooted while the service
is masked and a status on the service has been performed after the server had booted back up,
confirming that it is not running.
```

```
[[email protected] ~]# systemctl status chronyd chronyd.service
```

```
Loaded: masked (/dev/null)
Active: inactive (dead)
```

```
[[email protected] ~]# systemctl is-active chronyd inactive
```

To reverse this the service can be unmasked, it can then be started up successfully.

```
[[email protected] ~]# systemctl unmask chronyd rm '/etc/systemd/system/chronyd.service' [[email
protected] ~]# systemctl start chronyd [[email protected] ~]# systemctl is-active chronyd active
View Status of all Services
```

A list of the current status of all services can be viewed with the command below, remove `-all` to only list active services.

`systemctl list-units -type service -all` A list of all services can be viewed to see if they are currently enabled with the command below.

`systemctl list-unit-files -type service` There are many more things that `systemctl` can do, for further information check out the manual page by entering '`man systemctl`' at the command line.

Summary

With `systemctl` we can check if a service is currently active or enabled to start up automatically during system boot. We can start, stop, restart and reload services, as well as disable them from starting up during system boot and even mask them to prevent them being started up completely.

As `systemd` gains popularity and replaces older alternatives, it is more important to understand how to work with services with `systemctl`.

<https://www.rootusers.com/how-to-manage-linux-systemd-services-with-systemctl/>

Last update: **2017/09/01 15:11**