Chapter – 7 CONTROLLING SERVICES AND DAEMONS

Objectives:

- List system daemons and network services started by the systemd service and socket units.
- Control system daemons and network services, using systemctl.

IDENTIFYING AUTOMATICALLY STARTED SYSTEM PROCESSES

INTRODUCTION TO systemd

The **systemd** daemon manages start up for Linux, including service startup and service management in general. It activates system resources, server daemons, and other processes both at boot time and on a running system.

Daemons are processes that either wait or run in the background, performing various tasks. Generally, daemons start automatically at boot time and continue to run until shutdown or until they are manually stopped. It is a convention for names of many daemon programs to end in the letter d.

A few of the features provided by **systemd** include:

- Parallelization capabilities (starting multiple services simultaneously), which increase the boot speed of a system.
- On-demand starting of daemons without requiring a separate service.
- Automatic service dependency management, which can prevent long timeouts. For example, a network-dependent service will not attempt to start up until the network is available.
- A method of tracking related processes together by using Linux control groups.

DESCRIBING SERVICE UNITS

- Service units have a **.service** extension and represent system services.
- Socket units have a **.socket** extension and represent inter-process communication (IPC) sockets that systemd should monitor.
- Path units have a **.path** extension and are used to delay the activation of a service until a specific file system change occurs.

LISTING SERVICE UNITS

Use **systemctl** command to explore the current state of the system.

UNIT

The service unit name.

LOAD

Whether systemd properly parsed the unit's configuration and loaded the unit into memory.

ACTIVE

The high-level activation state of the unit. This information indicates whether the unit has started successfully or not.

SUB

The low-level activation state of the unit. This information indicates more detailed information about the unit. The information varies based on unit type, state, and how the unit is executed.

DESCRIPTION

The short description of the unit.

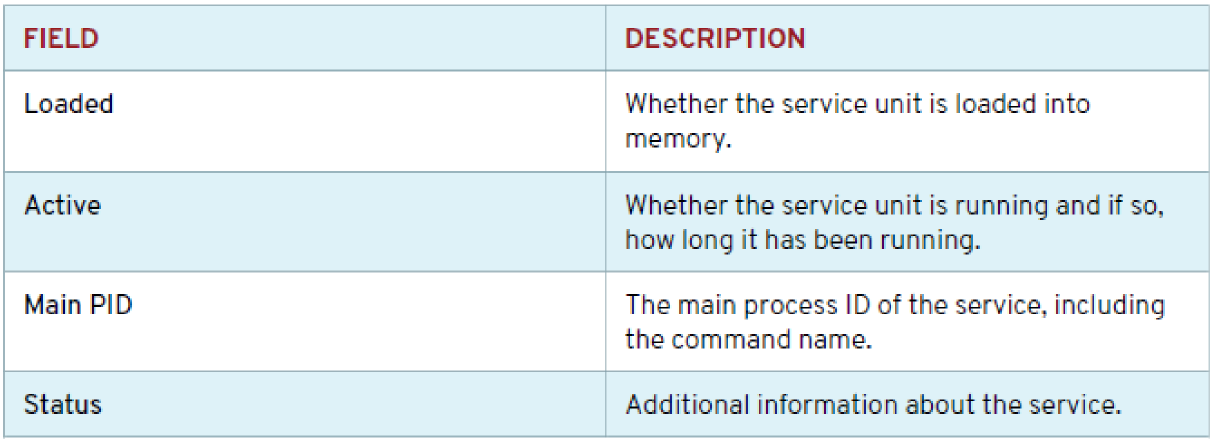
To see the state of all unit files installed, use the systemctl list-unit-files command.

systemctl list-unit-files --type=service

VIEWING SERVICE STATES

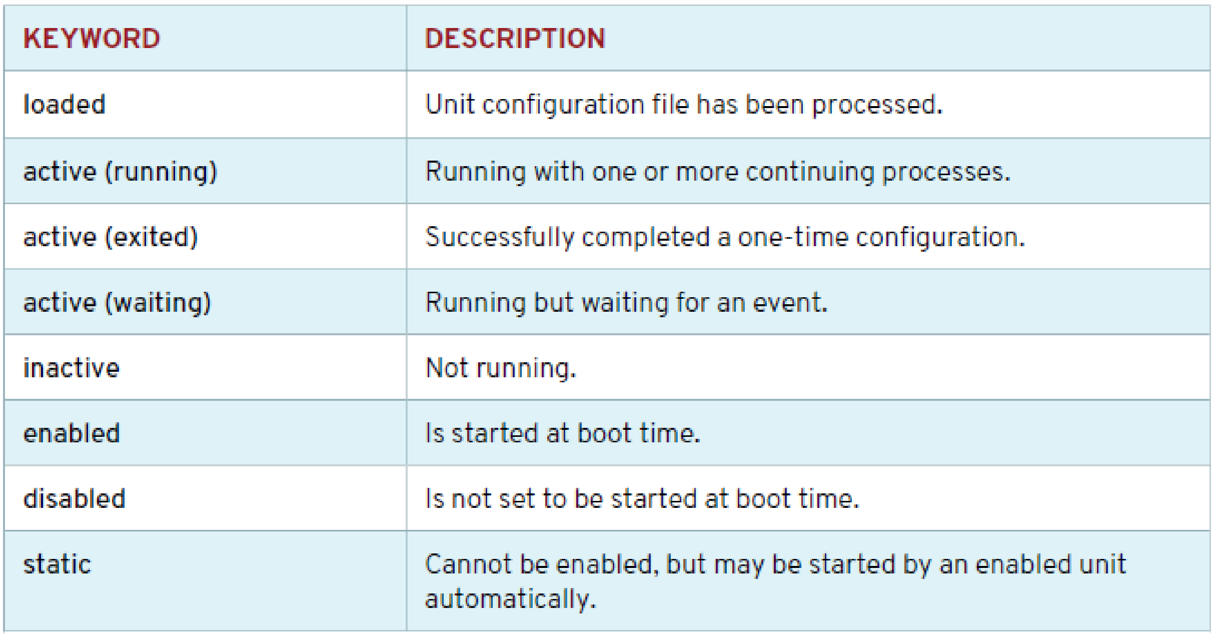
systemctl status *name.type*

Service Unit Information



FIELD	DESCRIPTION
Loaded	Whether the service unit is loaded into memory.
Active	Whether the service unit is running and if so, how long it has been running.
Main PID	The main process ID of the service, including the command name.
Status	Additional information about the service.

Service States in the Output of `systemctl`



KEYWORD	DESCRIPTION
loaded	Unit configuration file has been processed.
active (running)	Running with one or more continuing processes.
active (exited)	Successfully completed a one-time configuration.
active (waiting)	Running but waiting for an event.
inactive	Not running.
enabled	Is started at boot time.
disabled	Is not set to be started at boot time.
static	Cannot be enabled, but may be started by an enabled unit automatically.

VERIFYING THE STATUS OF A SERVICE

systemctl is-active service_name

The command returns state of the service unit, which is usually active or inactive.

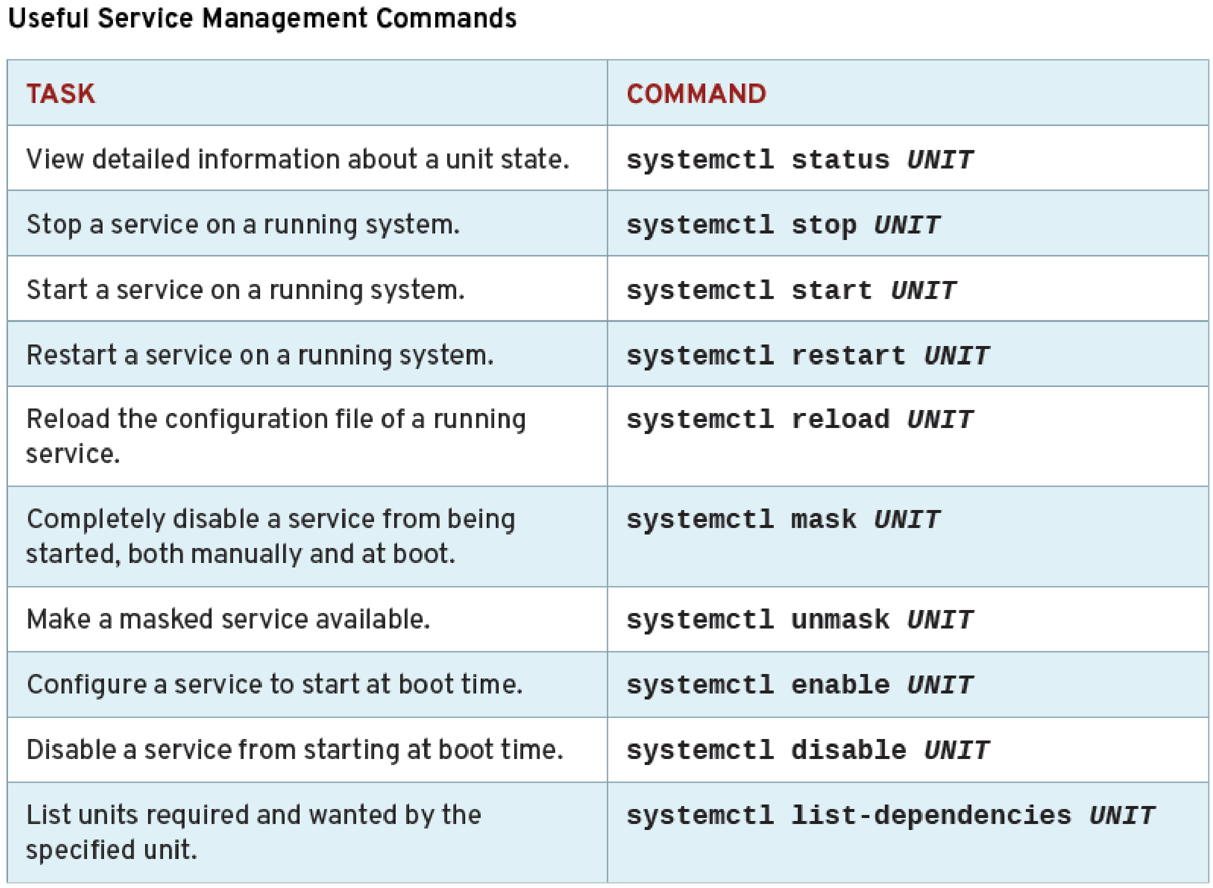
systemctl is-enabled service_name

The command returns whether the service unit is enabled to start at boot time, which is usually enabled or disabled.

systemctl is-failed service_name

The command either returns active if it is properly running or failed if an error has occurred during start up. In case the unit was stopped, it returns unknown or inactive.

CONTROLLING SYSTEM SERVICES



Useful Service Management Commands

TASK	COMMAND
View detailed information about a unit state.	`systemctl status UNIT`
Stop a service on a running system.	`systemctl stop UNIT`
Start a service on a running system.	`systemctl start UNIT`
Restart a service on a running system.	`systemctl restart UNIT`
Reload the configuration file of a running service.	`systemctl reload UNIT`
Completely disable a service from being started, both manually and at boot.	`systemctl mask UNIT`
Make a masked service available.	`systemctl unmask UNIT`
Configure a service to start at boot time.	`systemctl enable UNIT`
Disable a service from starting at boot time.	`systemctl disable UNIT`
List units required and wanted by the specified unit.	`systemctl list-dependencies UNIT`