

Linux Administration

Managing services

Xavier Belanger

**This work is licensed under
a Creative Commons Attribution-ShareAlike 4.0 International License.**

<http://creativecommons.org/licenses/by-sa/4.0/>

You are free to:

- **Share** — copy and redistribute the material in any medium or format
- **Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

What is a service?

- The generic definition of a service is of an application running on a dedicated system (a server) that is used by various other systems (the clients).
- This doesn't apply to all services; some services are running in the background on a system (not necessarily a server) just to provide a specific functionality.

Historical context

- Originally, a service running in the background was called a *daemon*.
- This translates into the fact that many services have their main program ending with the letter 'd'. For instance, the program for the SSH service is */usr/bin/sshd*.

Services components

Beyond the application itself, a service will usually include the following:

- a management script
- one or more configuration files
- a process id file (PID)
- some log files

systemd

- Initially, all services were started independently by the `init` process, using an architecture called “System V”, in reference to the original UNIX system.
- In 2010, Lennart Poettering started working on a replacement to manage all services and system functions once the system has booted. That application is called *systemd*.
- `systemd` is now the main initialization system for most Linux distributions.

Managing a service

- Once a service application is installed via a package, it should be integrated with systemd.
- The main operations are to start, stop, enable or disable the service, and check on its status when needed.

systemctl commands

The systemd's *systemctl* command will let you manage any service:

- `systemctl start <service>`
- `systemctl stop <service>`
- `systemctl enable <service>`
- `systemctl disable <service>`
- `systemctl status <service>`

systemctl status

- The basic command *systemctl status* will give you an overview of all services running on the system.
- The *systemctl list-unit-files* will summarize the status and state of all services.
- When checking on a specific service, you will find the name of the unit configuration file and the most recent lines from the service log file.

Unit configuration files

- Default unit configuration files are usually located in */usr/lib/systemd/system*.
- Additional unit configuration files specific to the system can be found in */etc/systemd/system*.
- When modifying those files, you must use the *systemctl daemon-reload* command for any change to be effective.

journalctl

- *journalctl* can display the content of the journal, starting with the oldest event (use the *-r* option to start with the most recent one).
- To check on the logs of a specific service, use the *-u* option with the name of the service:

```
journalctl -u sshd
```

PID file

- The PID file is usually used to track the main process for a service.
- It's also used as a lock: if the service is already running with a valid PID, the program will not launch a second instance.
- The PID can be used for monitoring that a service is running properly.

Most common and useful services

- OpenSSH - remote access, command execution and file transfer
- Apache HTTPD, Apache Tomcat, Nginx - Web service
- Postfix - Email (SMTP)
- OpenLDAP - Directory and authentication
- BIND - Name resolution (DNS)
- ISC DHCPD, ISC Kea - IP addressing (DHCP)
- CUPS - Printing (IPP)
- Samba - File and printer sharing
- NTP - Time synchronization