



GUÍA DE SERVICIOS LINUX CON SYSTEMCTL

Elaboró: Luisa María Zapata Saldarriaga
luisa.zapatas@udea.edu.co

INTRODUCCIÓN

En esta guía encontrará como gestionar servicios a partir de systemctl, en donde se detallan los pasos para crear un archivo unitario de Python que podrá activar o desactivar. Además, podrá encontrar una tabla de comandos básicos para el manejo de consolas Linux.

Systemctl es la interfaz de control que conecta varios aspectos del sistema haciendo un seguimiento de los servicios controlados por systemd en función de la disponibilidad de estos. Esta interfaz se puede usar para monitorear, consultar y modificar los servicios y procesos que son controlados por systemd [1].

Systemd inicializa los componentes del espacio de usuario que se ejecutan una vez comienza el kernel del sistema Linux, los programas se comunican por medio de configuraciones conocidos como archivos unitarios que se ubican en la carpeta /etc/systemd/system/, en este directorio se pueden crear, habilitar o deshabilitar servicios.

A. Comandos básicos de Linux

A continuación, se presenta la Tabla 1 con los comandos más frecuentes en el manejo del sistema operativo Linux, junto con una descripción general. Esta información permitirá adaptarse más fácilmente con el entorno de Linux, y así poder aplicar otros procesos.

Tabla 1. Lista de comandos más recurrentes en el sistema operativo de Linux

Comandos linux	
Comando	Descripción
cd <dir>	Permite navegar por todos los directorios del sistema.
cd ..	Retroceder al directorio anterior.
cat <filename>	Muestra el contenido de un archivo, permite copiar contenido de un archivo a otro. También permite concatenar, sumar o unir archivos.
nano <filename>	Editor de texto que permite realizar modificaciones a los archivos.
mkdir <dir>	Creación de varios directorios a la vez, así como establecer los permisos para los directorios.
pwd	Localiza la ruta del directorio de trabajo en el que se encuentra el usuario.
cp	Permite copiar uno o más archivos a una ubicación específica.
mv	Se utiliza para mover o renombrar archivos o directorios del sistema de archivos.
ls	Enumera el contenido del directorio que desee, archivos y otros directorios anidados.
history	Identificar los últimos comandos que se han ejecutado en el servidor.

B. Conexión a un servidor

Para conectarse con un servidor ssh, se introduce el siguiente comando, en donde deberá especificar el nombre de usuario y el nombre del servidor. En caso tal de no contar con uno, puede utilizar particiones de Linux o VirtualBox que permite crear máquinas virtuales de cualquier sistema operativo.

```
ssh username@server
```

- Username: Nombre de usuario del sistema ssh.
- Server: Nombre del servidor del sistema ssh.

```
C:\Users\john.ochoa>ssh bioinstrumentation@bioinstrumentation
bioinstrumentation@bioinstrumentation's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

UA Infra: Extended Security Maintenance (ESM) is not enabled.

0 updates can be applied immediately.

262 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Jun 27 13:11:15 2022 from fe80::b88a:e438:d001:149%eno1
bioinstrumentation@bioinstrumentation:~$
```

C. Crear un servicio en Linux

Inicialmente para crear un servicio en Linux, primero rediríjase a la siguiente ruta, en donde se encuentran los archivos unitarios del sistema:

```
cd /etc/systemd/system/
```

Para ver los servicios existentes, con el comando “ls” se listan estos.

```

bioinstrumentation@bioinstrumentation:/$ cd etc/systemd/system
bioinstrumentation@bioinstrumentation:/etc/systemd/system$ ls
backend_graphics.service                                libvirtd.service
backend_preprocess_eeg                                 dbus-org.freedesktop.Avahi.service                multi-user.target.wants
backend_preprocess_eeg.service+                        dbus-org.freedesktop.nm-dispatcher.service        network-online.target.wants
backend_preprocess_eeg.servicees                       default.target.wants                               open-vm-tools.service.requires
backend_preprocess_eeg.service.save                    display-manager.service                            paths.target.wants
backend_preprocess_eeg.service.save.1                  display-manager.service.wants                     printer.target.wants
backend_preprocess_eeg.service.save.2                  final.target.wants                                 sockets.target.wants
backend_preprocess_eeg.service.save.3                  frontend.service                                   sshd.service
backend_preprocess_eeg.service.save.4                  getty.target.wants                                 suspend.target.wants
backend_preprocess_eeg.service.save.5                  graphical.target.wants                             sysinit.target.wants
backend_preprocess_eeg.service.save.6                  hibernate.target.wants                            syslog.service
bluetooth.target.wants                                 hybrid-sleep.target.wants                          timers.target.wants
cloud-final.service.wants                              iscsi.service
bioinstrumentation@bioinstrumentation:/etc/systemd/system$ _

```

Luego de esto, crear un fichero vacío con el siguiente comando:

```
sudo nano [your_new_service].service
```

Donde '[your_new_service]', hace referencia al nombre del servicio que desee crear. Para crear un servicio de un archivo de Python con un ambiente configure el archivo unitario de la siguiente forma:

```

[Unit]
Description=<description_about_your_project>

[Service]
User=<user_such_as_root>
WorkingDirectory=<path to your project directory>
ExecStart=/bin/bash -c 'cd /home/unixcop/project' && source venv/bin/activate && python test.py'

#replace /home/unixcop/project/ with your [Path] venv/bin/activate with your source and test.py with
your script name

[Install]
WantedBy=multi-user.target

```

En la sección de 'Description', podrá colocar una breve explicación que apunte al servicio que se esté corriendo. En 'User' se espera el nombre de usuario en donde se encuentra trabajando. Para conocer el usuario con el comando 'whoami' se identifica el usuario en donde se encuentra actualmente el navegando.

```
bioinstrumentation@bioinstrumentation:/$ whoami
bioinstrumentation
bioinstrumentation@bioinstrumentation:/$ _
```

En ‘*WorkingDirectory*’ se introduce la ruta de trabajo, es decir la ruta que contiene el ejecutable. Para conocer esta, podrá utilizar el comando ‘*pwd*’ en donde se encuentre el archivo y este retornará la ruta absoluta. En ‘*ExecStart*’ debemos indicar la ruta absoluta al script que se desea ejecutar para arrancar el servicio en cuestión.

Para verificar que nuestro archivo se haya configurado de forma correcta, con el comando ‘*cat* <filename>’ podemos visualizar lo que hay dentro de nuestro fichero.

```
bioinstrumentation@bioinstrumentation:/etc/systemd/system$ cat backend_graphics.service
[Unit]
Description=<estariamos muy felices si esto funciona>

[Service]
ExecStart=bioinstrumentation
WorkingDirectory=/home/bioinstrumentation/Documents/TD6_LUISA/Backend_service_graphics
ExecStart=/bin/bash -c 'cd /home/bioinstrumentation/Documents/TD6_LUISA/Backend_service_graphics && source my_env/bin/activate && python3 src/main.py'
Replace=/home/unixcop/project/ with your Path, venv/bin/activate with your source and test.py with your script name

[Install]
```

Una vez creado el servicio, es necesario otorgarle permisos, con la siguiente línea de código:

```
sudo chmod +x
```

Después de terminar, cargue los archivos de servicio para incluir el nuevo servicio con las siguientes líneas de código.

```
sudo systemctl daemon-reload
sudo systemctl start [your_new_service].service
sudo systemctl enable [your_new_service].service
```

Con el comando ‘*status*’, se puede verificar si el servicio está corriendo correctamente.

```
sudo systemctl status [your_new_service].service
```

```
bioinstrumentation@bioinstrumentation:/etc/systemd/system$ systemctl status backend_graphics.service
● backend_graphics.service - <estariamos muy felices si esto funciona>
   Loaded: loaded (/etc/systemd/system/backend_graphics.service; static; vendor preset: enabled)
   Active: active (running) since Mon 2022-06-27 10:20:29 -05; 5h 47min ago
     Main PID: 14529 (bash)
       Tasks: 4
      Memory: 717.7M
         CPU: 5h 22min 34.645s
    CGroup: /system.slice/backend_graphics.service
            └─14529 /bin/bash -c cd /home/bioinstrumentation/Documents/TD6_LUISA/Backend_service_graphics && source my_env/bin/activate && python3 src/main.py
              └─14531 python3 src/main.py
                └─14557 /home/bioinstrumentation/Documents/TD6_LUISA/Backend_service_graphics/my_env/bin/python3 -c from multiprocessing.resource_tracker import main;main(4)
                  └─14558 /home/bioinstrumentation/Documents/TD6_LUISA/Backend_service_graphics/my_env/bin/python3 -c from multiprocessing.spawn import spawn_main; spawn_main(

Jun 27 10:20:48 bioinstrumentation bash[14529]: INFO: Uvicorn running on http://localhost:8000 (Press CTRL+C to quit)
Jun 27 10:20:48 bioinstrumentation bash[14529]: 2022-06-27 10:20:48,510 - uvicorn.error - INFO - Uvicorn running on http://localhost:8000 (Press CTRL+C to quit)
Jun 27 10:20:48 bioinstrumentation bash[14529]: INFO: Started reloader process [14531] using statreload
Jun 27 10:20:48 bioinstrumentation bash[14529]: 2022-06-27 10:20:48,510 - uvicorn.error - INFO - Started reloader process [14531] using statreload
Jun 27 10:20:54 bioinstrumentation bash[14529]: INFO: Started server process [14558]
Jun 27 10:20:54 bioinstrumentation bash[14529]: 2022-06-27 10:20:54,034 - uvicorn.error - INFO - Started server process [14558]
Jun 27 10:20:54 bioinstrumentation bash[14529]: INFO: Waiting for application startup.
Jun 27 10:20:54 bioinstrumentation bash[14529]: 2022-06-27 10:20:54,034 - uvicorn.error - INFO - Waiting for application startup.
Jun 27 10:20:54 bioinstrumentation bash[14529]: INFO: Application startup complete.
Jun 27 10:20:54 bioinstrumentation bash[14529]: 2022-06-27 10:20:54,035 - uvicorn.error - INFO - Application startup complete.
lines 1-23/23 (END)
```

Después de cada reinicio habilite su servicio de la siguiente manera:

```
sudo systemctl enable [your_new_service].service
```

Para detener un servicio utilice los comandos ‘stop’ y ‘disable’ de la siguiente forma:

```
sudo systemctl stop [your_new_service].service  
sudo systemctl disable [your_new_service].service
```

Referencias

- [1] P. Kirkbride, “systemd,” *Basic Linux Termin. Tips Tricks*, pp. 221–234, 2020, doi: 10.1007/978-1-4842-6035-7_11.