

Gestión de servicios con Systemd

25/06/2025

Ángel Moreno García

CodeArts Solutions

Madrid

Introducción

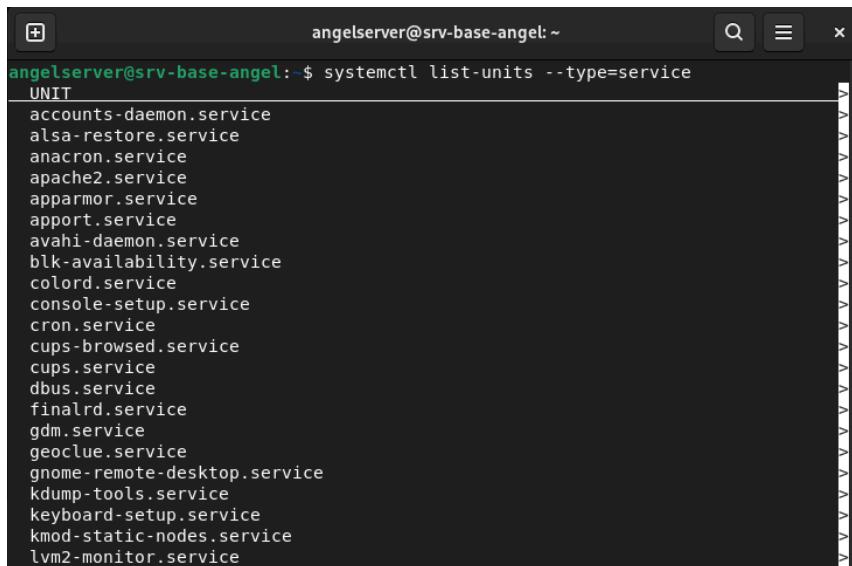
En este proyecto se ha instalado y configurado un servidor Linux (**Ubuntu Server 25.04 LTS**), con el objetivo de gestionar y configurar servicios con **Systemd**. También usaremos un script para personalizar servicios y la correspondiente monitorización y registro.

Fase 1: Análisis de servicios del sistema

Objetivos

1. Identificar los servicios activos del sistema.
2. Reconocer servicios fundamentales que garantizan el funcionamiento básico del servidor.
3. Comprobar el estado del servidor web.

Para ver los servicios activos del sistema usaremos el comando: **systemctl list-units --type=service**. Este comando lista todos los servicios que están cargados actualmente en el sistema, mostrando su estado (activo, inactivo, fallido, etc.).

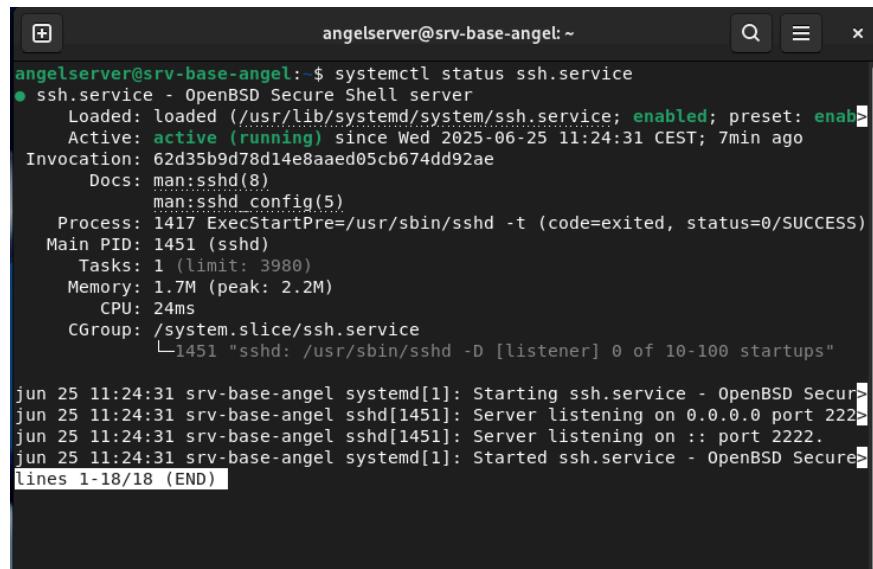


```
angelsrv@angelsrv:~$ systemctl list-units --type=service
UNIT
accounts-daemon.service
alsa-restore.service
anacron.service
apache2.service
apparmor.service
apport.service
avahi-daemon.service
blk-availability.service
colord.service
console-setup.service
cron.service
cups-browsed.service
cups.service
dbus.service
finalrd.service
gdm.service
geoclue.service
gnome-remote-desktop.service
kdump-tools.service
keyboard-setup.service
kmod-static-nodes.service
lvm2-monitor.service
```

Servicios fundamentales seleccionados

1. ssh.service

- **Función:** Permite las conexiones remotas seguras mediante el protocolo SSH (puerto 22 o el que se configure). Es crucial para administrar el servidor de forma remota.
- **Comando:** `systemctl status ssh.service`
- **Estado esperado:** activo y habilitado.



```
angelsrv@srv-base-angel: ~
angelsrv@srv-base-angel: $ systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Wed 2025-06-25 11:24:31 CEST; 7min ago
    Invocation ID: 62d35b9d78d14e8aaed05cb674dd92ae
      Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 1417 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1451 (sshd)
     Tasks: 1 (limit: 3980)
    Memory: 1.7M (peak: 2.2M)
       CPU: 24ms
      CGroup: /system.slice/ssh.service
              └─1451 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

jun 25 11:24:31 srv-base-angel systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
jun 25 11:24:31 srv-base-angel sshd[1451]: Server listening on 0.0.0.0 port 2222.
jun 25 11:24:31 srv-base-angel sshd[1451]: Server listening on :: port 2222.
jun 25 11:24:31 srv-base-angel systemd[1]: Started ssh.service - OpenBSD Secure Shell server.

lines 1-18/18 (END)
```

2. cron.service

- **Función:** Servicio de programación de tareas automáticas. Es esencial para ejecutar scripts, copias de seguridad, actualizaciones programadas, etc.
- **Comando:** `systemctl status cron.service`
- **Estado esperado:** activo (para garantizar ejecución de tareas programadas).

```
angelserv@srv-base-angel:~$ systemctl status cron.service
● cron.service - Regular background program processing daemon
  Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled; preset: ena
  Active: active (running) since Wed 2025-06-25 11:24:31 CEST; 8min ago
    Invocation: cd5846efc38248f8a940608815bebb0c
      Docs: man:cron(8)
    Main PID: 1413 (cron)
       Tasks: 1 (limit: 3980)
      Memory: 472K (peak: 2M)
        CPU: 21ms
       CGrou: /system.slice/cron.service
              └─1413 /usr/sbin/cron -f -P

jun 25 11:24:31 srv-base-angel systemd[1]: Started cron.service - Regular backg>
jun 25 11:24:31 srv-base-angel (cron)[1413]: cron.service: Referenced but unset>
jun 25 11:24:31 srv-base-angel cron[1413]: (CRON) INFO (pidfile fd = 3)
jun 25 11:24:31 srv-base-angel cron[1413]: (CRON) INFO (Running @reboot jobs)
jun 25 11:25:01 srv-base-angel CRON[3630]: pam_unix(cron:session): session open>
jun 25 11:25:01 srv-base-angel CRON[3631]: (root) CMD (command -v debian-sal >>
jun 25 11:25:01 srv-base-angel CRON[3630]: pam_unix(cron:session): session clos>
jun 25 11:30:01 srv-base-angel CRON[3718]: pam_unix(cron:session): session open>
jun 25 11:30:01 srv-base-angel CRON[3719]: (root) CMD ([ -x /etc/init.d/anacron>
jun 25 11:30:01 srv-base-angel CRON[3718]: pam_unix(cron:session): session clos>
lines 1-22/22 (END)
```

3. networking.service

- **Función:** Controla y configura las interfaces de red durante el arranque. Sin este servicio, el servidor no podría conectarse a otras redes o servicios.
- **Comando:** *ip a*
- **Estado esperado:** activo, ya que garantiza la conectividad del sistema.

```
angelserv@srv-base-angel:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:45:f3:29 brd ff:ff:ff:ff:ff:ff
    altname enx08002745f329
    inet 192.168.1.21/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fe45:f329/64 scope link proto kernel ll
            valid_lft forever preferred_lft forever
angelserv@srv-base-angel:~$
```

Para la verificación del servicio web **Apache2** escribiremos en la consola: ***sudo systemctl status apache2***

```
angelserver@srv-base-angel:~$ systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
   Active: active (running) since Wed 2025-06-25 11:24:31 CEST; 12min ago
     Invocation: a6a36150a5584d008b596e12cb6f4133
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1409 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
   Main PID: 1483 (apache2)
      Tasks: 55 (limit: 3980)
     Memory: 7.8M (peak: 8.1M)
        CPU: 69ms
       CGroup: /system.slice/apache2.service
               ├─1483 /usr/sbin/apache2 -k start
               ├─1486 /usr/sbin/apache2 -k start
               └─1487 /usr/sbin/apache2 -k start

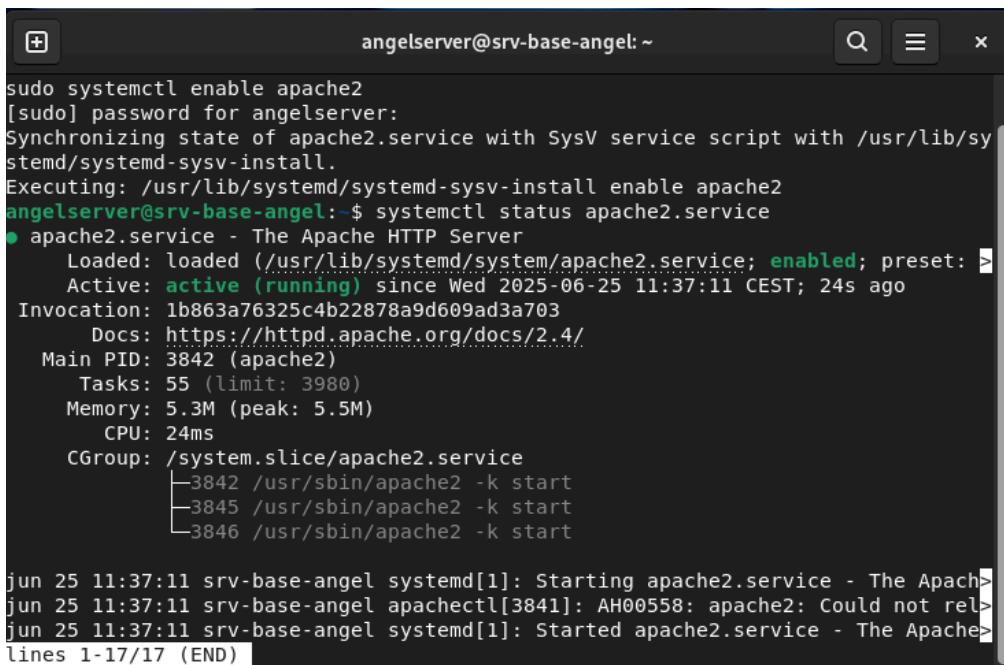
jun 25 11:24:30 srv-base-angel systemd[1]: Starting apache2.service - The Apache>
jun 25 11:24:31 srv-base-angel apachectl[1446]: AH00558: apache2: Could not rel>
jun 25 11:24:31 srv-base-angel systemd[1]: Started apache2.service - The Apache>
lines 1-18/18 (END)
```

Fase 2: Gestión avanzada con systemd

Objetivos

- Controlar el ciclo de vida del servidor web.
- Automatizar su recuperación en caso de fallo.
- Crear accesos directos personalizados.

Los comandos utilizados para detener, iniciar y reiniciar el servicio web Apache2 son: ***sudo systemctl stop apache2***, ***sudo systemctl start apache2*** y ***sudo systemctl restart apache2***. También habilitaremos el arranque automático con: ***sudo systemctl enable apache2***



```

sudo systemctl enable apache2
[sudo] password for angelserver:
Synchronizing state of apache2.service with SysV service script with /usr/lib/sys
temd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
angelserver@srv-base-angel: $ systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
   Active: active (running) since Wed 2025-06-25 11:37:11 CEST; 24s ago
     Invocation: 1b863a76325c4b22878a9d609ad3a703
       Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 3842 (apache2)
      Tasks: 55 (limit: 3980)
     Memory: 5.3M (peak: 5.5M)
        CPU: 24ms
      CGroup: /system.slice/apache2.service
              └─3842 /usr/sbin/apache2 -k start
                  ├─3845 /usr/sbin/apache2 -k start
                  ├─3846 /usr/sbin/apache2 -k start

jun 25 11:37:11 srv-base-angel systemd[1]: Starting apache2.service - The Apache>
jun 25 11:37:11 srv-base-angel apachectl[3841]: AH00558: apache2: Could not rel>
jun 25 11:37:11 srv-base-angel systemd[1]: Started apache2.service - The Apache>
lines 1-17/17 (END)

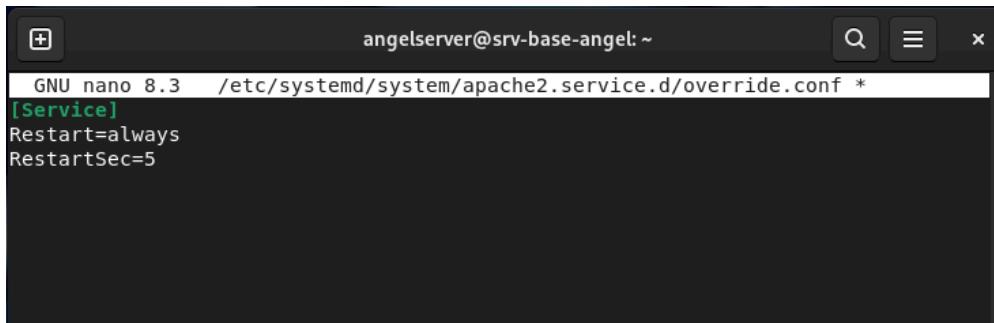
```

Vamos a editar la configuración del servicio en caso de fallo con el comando: ***sudo systemctl edit apache2***. Esto abre un archivo de override en */etc/systemd/system/apache2.service.d/override.conf*. Añadimos:

[Service]

Restart=always

RestartSec=5



```

GNU nano 8.3  /etc/systemd/system/apache2.service.d/override.conf *
[Service]
Restart=always
RestartSec=5

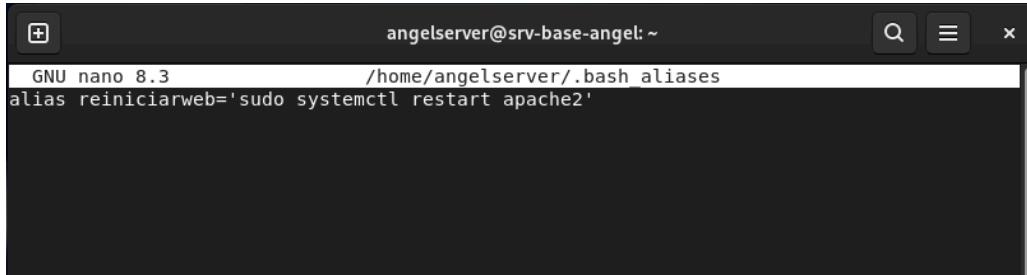
```

Aplicamos los cambios con:

sudo systemctl daemon-reexec, sudo systemctl restart apache2

Crear un alias

Abriremos el archivo /home/usuario/.bash_aliases con **sudo nano** **/home/usuario/.bash_aliases** e introduciremos en el archivo: **alias reiniciarweb='sudo systemctl restart apache2'**



```
GNU nano 8.3          /home/angelserver/.bash_aliases
alias reiniciarweb='sudo systemctl restart apache2'
```

Fase 3: Servicio personalizado – saludo.service

Objetivo

Automatizar un mensaje en el arranque del sistema que confirme que el servidor se ha iniciado correctamente.

1. Crear el script

Usaremos el comando **sudo nano /usr/local/bin/saludo.sh** con el contenido :

```
#!/bin/bash
echo "¡Servidor iniciado correctamente!" >> /var/log/saludo.log
```

Introduciremos el comando **sudo chmod +x /usr/local/bin/saludo.sh** para darle los permisos adecuados.



```
angelserv@srv-base-angel:~$ sudo nano /usr/local/bin/saludo.sh
angelserv@srv-base-angel:~$ sudo chmod +x /usr/local/bin/saludo.sh
angelserv@srv-base-angel:~$
```

```
GNU nano 8.3          /usr/local/bin/saludo.sh *
#!/bin/bash
echo "¡Servidor iniciado correctamente!" >> /var/log/saludo.log
```

2. Crear el archivo de servicio

Abriremos el archivo utilizando ***sudo nano /etc/systemd/system/saludo.service*** y dentro añadiremos:

[Unit]

Description=Servicio de saludo al iniciar el sistema

[Service]

ExecStart=/usr/local/bin/saludo.sh

Type=oneshot

[Install]

WantedBy=multi-user.target

```
GNU nano 8.3          /etc/systemd/system/saludo.service *
[Unit]
Description=Mensaje de bienvenida al arrancar el servidor
After=network.target

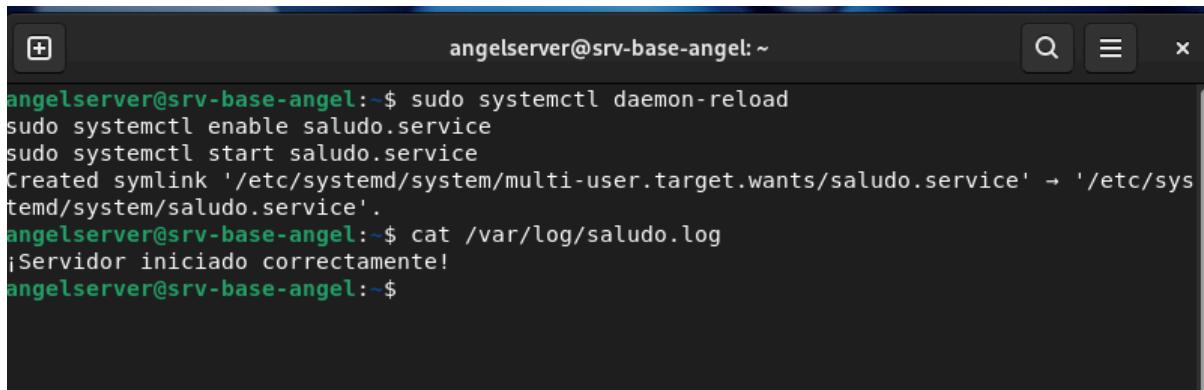
[Service]
ExecStart=/usr/local/bin/saludo.sh
Type=oneshot
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

3. Activación y verificación

Para activar el script usaremos: ***sudo systemctl daemon-reload, sudo systemctl enable saludo.service, y sudo systemctl start saludo.service***

Después,verificaremos que el contenido del script se ejecuta y nos muestra el mensaje que hemos introducido previamente. Para ello utilizaremos el comando: ***cat/var/log/saludo.log***



A screenshot of a terminal window titled "angelserver@srv-base-angel: ~". The terminal displays the following command-line session:

```
angelserv...$ sudo systemctl daemon-reload
sudo systemctl enable saludo.service
sudo systemctl start saludo.service
Created symlink '/etc/systemd/system/multi-user.target.wants/saludo.service' → '/etc/systemd/system/saludo.service'.
angelserv...$ cat /var/log/saludo.log
¡Servidor iniciado correctamente!
angelserv...$
```



Fase 4: Monitorización y registro

Objetivo

Supervisar el funcionamiento de los servicios usando `journalctl` y guardar logs para auditoría.

Comandos utilizados

Ver logs del servicio saludo: `journalctl -u saludo.service`

Filtrar errores o advertencias del sistema: `journalctl -p 3 -xb`

Guardar logs en carpetas personalizadas: `sudo mkdir -p /srv/logs, sudo journalctl -u saludo.service > /srv/logs/saludo_journal.log`

Verificar el contenido del archivo: `cat /srv/logs/saludo_journal.log`

```
GNU nano 8.3          /srv/logs/saludo_journal.log
jun 25 11:56:02 srv-base-angel systemd[1]: Starting saludo.service - Mensaje de bienvenida >
jun 25 11:56:02 srv-base-angel systemd[1]: Finished saludo.service - Mensaje de bienvenida >
```

Conclusión

Durante este trabajo se ha aprendido a gestionar y controlar servicios del sistema en un entorno **Linux**. Se identificaron servicios clave del sistema, se configuró el reinicio automático de un servicio web y se creó un servicio personalizado que genera un log al arrancar. También se aplicaron mejoras como la creación de alias y el cambio de configuraciones para mejorar el control. Se utilizaron herramientas como **systemctl** y **journalctl** para gestionar y monitorizar los servicios. Todo se documentó con pruebas, capturas y scripts, asegurando que el sistema funciona correctamente y es fácil de mantener.