

Gestión de Servicios y Automatización con Systemd en Linux

1. Análisis de servicios del sistema.

- **Listar todos los servicios activos del sistema:** Utilizando systemctl
list-units --type=service
- **Identificar y documentar 3 servicios activos fundamentales:** ssh, cron y networking.
- **Comprobar si el servidor web está activo, habilitado y funcionando.**

2. Gestión avanzada de servicios con Systemd.

- **Detener, reiniciar y habilitar al arranque el servicio web:** apache2
- **Modificar la configuración de uno de los servicios para que se reinicie automáticamente si falla:** Restart=always en su .service.
- **Crear un alias para reiniciar rápidamente el servicio:** Desde .bashrc o .zshrc.

3. Creación de un servicio personalizado.

- **Crear un script Bash:** Llamado [saludo.sh](#) que escriba "¡Servidor iniciado correctamente!" en un archivo /var/log/saludo.log.
- **Crear un nuevo servicio de systemd:** Llamado saludo.service que ejecute este script automáticamente al iniciar el sistema.
- **Comprobar que el servicio:** Esté habilitado, se ejecuta al arrancar y crea el archivo de log correctamente.

4. Monitorización y logs.

- **Visualizar los logs de los servicios anteriores con journalctl.**
 - **Filtrar los mensajes de error o advertencia:** journalctl -p 3 -xb
 - **Registrar el estado del servicio saludo y guardar una copia del log en /srv/logs/saludo_journal.log**
-

1. Análisis de servicios del sistema.

Lo primero será listar todos los servicios activos del sistema con el comando:

systemctl list-units --type=service

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ systemctl list-units --type=service
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
accounts-daemon.service            loaded active running Accounts Servi
alsa-restore.service               loaded active exited Save/Restore S
apache2.service                    loaded active running The Apache HTT
apparmor.service                   loaded active exited Load AppArmor
apport.service                     loaded active exited automatic cras
avahi-daemon.service               loaded active running Avahi mDNS/DNS
blk-availability.service            loaded active exited Availability o
colord.service                     loaded active running Manage, Instal
console-setup.service              loaded active exited Set console fo
cron.service                       loaded active running Regular backgr
cups-browsed.service               loaded active running Make remote CU
cups.service                       loaded active running CUPS Scheduler
dbus.service                       loaded active running D-Bus System M
finalrd.service                    loaded active exited Create final r
fwupd.service                      loaded active running Firmware updat
gdm.service                        loaded active running GNOME Display
gnome-remote-desktop.service        loaded active running GNOME Remote D
keyboard-setup.service              loaded active exited Set the consol
kmod-static-nodes.service           loaded active exited Create List of
lvm2-monitor.service               loaded active exited Monitoring of
ModemManager.service               loaded active running Modem Manager
multipathd.service                 loaded active running Device-Mapper
```

Ahora para documentar los servicios ssh, cron y networking, introduciremos los comando:

systemctl status ssh.service

systemctl status cron.service

systemctl status NetworkManager

Esto nos servirá para conocer la información de cada uno de los servicios.

El servicio ssh: Nos permite el acceso remoto seguro al servidor mediante el protocolo SSH.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ 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 Fri 2025-06-20 07:27:22 CEST; 2h 51min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 1222 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 1237 (sshd)
    Tasks: 1 (limit: 4605)
   Memory: 2.2M (peak: 2.4M)
      CPU: 14ms
   CGroup: /system.slice/ssh.service
           └─1237 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Warning: some journal files were not opened due to insufficient permissions.
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

El servicio **cron**: Ejecuta tareas programadas automáticamente.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ systemctl status cron.service
● cron.service - Regular background program processing daemon
   Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-06-20 07:27:22 CEST; 2h 57min ago
     Docs: man:cron(8)
    Main PID: 1217 (cron)
      Tasks: 1 (limit: 4605)
     Memory: 464.0K (peak: 2.1M)
        CPU: 137ms
    CGroup: /system.slice/cron.service
            └─1217 /usr/sbin/cron -f -P

Warning: some journal files were not opened due to insufficient permissions.
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

El servicio **NetworkManager**: Gestiona la configuración de red del sistema.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; prese>
   Active: active (running) since Fri 2025-06-20 07:27:22 CEST; 2h 59min ago
     Docs: man:NetworkManager(8)
    Main PID: 962 (NetworkManager)
      Tasks: 4 (limit: 4605)
     Memory: 24.5M (peak: 41.1M)
        CPU: 188ms
    CGroup: /system.slice/NetworkManager.service
            └─962 /usr/sbin/NetworkManager --no-daemon

Warning: some journal files were not opened due to insufficient permissions.
lines 1-12/12 (END)
```

Lo siguiente será comprobar si el servidor web está activo, habilitado y funcionando.

Para comprobar que está activo utilizamos el comando:

systemctl status apache2

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enab>
   Active: active (running) since Fri 2025-06-20 07:27:22 CEST; 3h 4min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Process: 1214 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 1245 (apache2)
      Tasks: 55 (limit: 4605)
     Memory: 7.6M (peak: 7.9M)
        CPU: 196ms
    CGroup: /system.slice/apache2.service
            └─1245 /usr/sbin/apache2 -k start
              └─1246 /usr/sbin/apache2 -k start
                └─1247 /usr/sbin/apache2 -k start
```

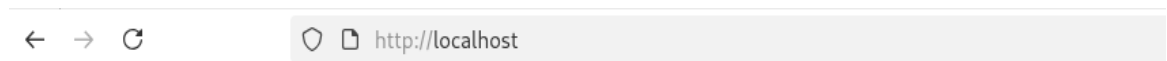
Para comprobar si está habilitado utilizamos el comando:

sudo systemctl is-enabled apache2

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl is-enabled apache2
[sudo] password for AZDINFARISSE:
enabled
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

Para comprobar si funciona correctamente, entramos desde el navegador a:

<http://localhost>



¡Apache funcionando desde /srv/www!

2. Gestión avanzada de servicios con Systemd.

Para detener, reiniciar y habilitar al arranque el servicio web de apache2 introduciremos los siguientes comandos:

Para detener: **sudo systemctl stop apache2**

Para reiniciar: **sudo systemctl restart apache2**

Para habilitarlo al arranque: **sudo systemctl enable apache2**

Y verificamos con el comando:

sudo systemctl is-enabled apache2

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl stop apache2
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl restart apache2
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl is-enabled apache2
enabled
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

Ahora vamos a modificar Apache2 para que se reinicie automáticamente si falla.

Para ello tenemos que crear un override de la unidad con systemctl edit.

Para crear un override del servicio utilizamos el comando:

sudo systemctl edit apache2

Con este comando se nos abrirá un archivo con encabezado [Service] en el que añadiremos las siguientes líneas:

[Service]

Restart=always

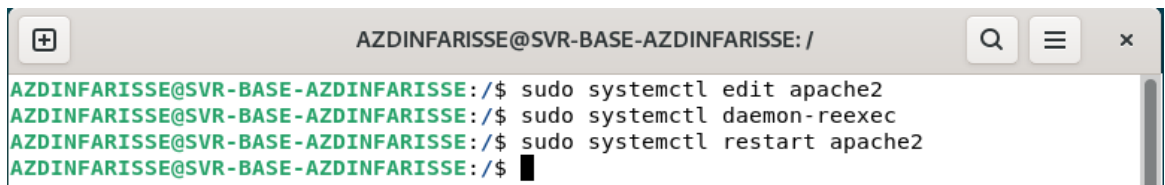
RestartSec=3

Esto hará que el servicio se reinicie aunque falle y que espere 3 segundos antes de reiniciarse.

Lo siguiente será recargar la configuración de systemd y reiniciar el servicio con los comandos:

sudo systemctl daemon-reexec

sudo systemctl restart apache2



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl edit apache2
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl daemon-reexec
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl restart apache2
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

Ahora crearemos un alias para reiniciar rápidamente Apache desde bash. Para ello editaremos el archivo **.bashrc** abriéndolo con el comando:

nano ~/.bashrc

Y añadiendo al final del archivo la siguiente línea:

alias restartapache='sudo systemctl restart apache2'



```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
GNU nano 7.2 /home/AZDINFARISSE/.bashrc
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ] && echo terminal || echo er
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
alias restartapache='sudo systemctl restart apache2'
```

Ahora recargamos el shell para activar el alias con:

source ~/.bashrc

y probamos el alias:

restartapache

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ nano ~/.bashrc
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ source ~/.bashrc
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ restartapache
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

3. Creación de un servicio personalizado.

Ahora vamos a crear un script Bash llamado [saludo.sh](#) que escriba "¡Servidor iniciado correctamente!" en un archivo /var/log/saludo.log. Para ello introduciremos el comando:

sudo nano /usr/local/bin/[saludo.sh](#)

y le introduciremos la siguiente línea:

#!/bin/bash

echo "¡Servidor iniciado correctamente! - \$(date)" >> /var/log/saludo.log

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
GNU nano 7.2 /usr/local/bin/saludo.sh *
#!/bin/bash
echo "¡Servidor iniciado correctamente! - $(date)" >> /var/log/saludo.log
```

y le daremos permisos de ejecución con el comando:

sudo chmod +x /usr/local/bin/[saludo.sh](#)

Ahora crearemos un nuevo servicio de **systemd** llamado **saludo.service** que ejecute el script creado automáticamente al iniciar el sistema. Para ello introduciremos el comando:

sudo nano /etc/systemd/system/saludo.service

Y le introduciremos las siguientes líneas:

[Unit]

Description=Mensaje de saludo al iniciar el servidor

After=network.target

[Service]

Type=oneshot

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

[Install]

WantedBy=multi-user.target

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
GNU nano 7.2 /etc/systemd/system/saludo.service *
[Unit]
Description=Mensaje de saludo al iniciar el servidor
After=network.target

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

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

Ahora vamos a recargar la configuración, habilitar el servicio para que se ejecute al arrancar y ejecutarlo manualmente para probarlo con los comandos:

```
sudo systemctl daemon-reload
sudo systemctl enable saludo.service
sudo systemctl start saludo.service
```

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl daemon-reload
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl enable saludo.service
Created symlink /etc/systemd/system/multi-user.target.wants/saludo.service -> /etc/systemd/system/saludo.service.
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo systemctl start saludo.service
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ █
```

Ahora comprobamos que se creó el archivo de log correctamente con:

```
cat /var/log/saludo.log
```

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ cat /var/log/saludo.log
¡Servidor iniciado correctamente! - vie 20 jun 2025 12:03:41 CEST
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$
```

Y comprobamos el estado del servicio con el comando:

```
systemctl status saludo.service
```

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ systemctl status saludo.service
○ saludo.service - Mensaje de saludo al iniciar el servidor
   Loaded: loaded (/etc/systemd/system/saludo.service; enabled; preset: enabled)
   Active: inactive (dead) since Fri 2025-06-20 12:03:41 CEST; 3min 42s ago
     Process: 8337 ExecStart=/usr/local/bin/saludo.sh (code=exited, status=0/SUCCESS)
    Main PID: 8337 (code=exited, status=0/SUCCESS)
       CPU: 2ms
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ █
```


4. Monitorización y logs.

Para visualizar los logs de los servicios anteriores con journalctl lo haremos con los comandos:

```
sudo journalctl -u apache2
```

```
sudo journalctl -u saludo.service
```

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo journalctl -u apache2
jun 19 12:25:40 SVR-BASE-AZDINFARISSE systemd[1]: Starting apache2.service - The Apache
jun 19 12:25:40 SVR-BASE-AZDINFARISSE apachectl[11965]: AH00558: apache2: Could not reli
jun 19 12:25:40 SVR-BASE-AZDINFARISSE systemd[1]: Started apache2.service - The Apache H
jun 19 12:32:37 SVR-BASE-AZDINFARISSE systemd[1]: Stopping apache2.service - The Apache
jun 19 12:32:37 SVR-BASE-AZDINFARISSE apachectl[12936]: AH00558: apache2: Could not reli
jun 19 12:32:37 SVR-BASE-AZDINFARISSE systemd[1]: apache2.service: Deactivated successfu
jun 19 12:32:37 SVR-BASE-AZDINFARISSE systemd[1]: Stopped apache2.service - The Apache H
jun 19 12:32:37 SVR-BASE-AZDINFARISSE systemd[1]: Starting apache2.service - The Apache
jun 19 12:32:37 SVR-BASE-AZDINFARISSE apachectl[12944]: AH00558: apache2: Could not reli
jun 19 12:32:37 SVR-BASE-AZDINFARISSE systemd[1]: Started apache2.service - The Apache H
jun 19 12:34:39 SVR-BASE-AZDINFARISSE systemd[1]: Stopping apache2.service - The Apache
jun 19 12:34:39 SVR-BASE-AZDINFARISSE apachectl[14080]: AH00558: apache2: Could not reli
jun 19 12:34:39 SVR-BASE-AZDINFARISSE systemd[1]: apache2.service: Deactivated successfu
jun 19 12:34:39 SVR-BASE-AZDINFARISSE systemd[1]: Stopped apache2.service - The Apache H
jun 19 12:34:39 SVR-BASE-AZDINFARISSE systemd[1]: Starting apache2.service - The Apache
jun 19 12:34:39 SVR-BASE-AZDINFARISSE apachectl[14087]: AH00558: apache2: Could not reli
jun 19 12:34:39 SVR-BASE-AZDINFARISSE systemd[1]: Started apache2.service - The Apache H
jun 19 12:42:18 SVR-BASE-AZDINFARISSE systemd[1]: Stopping apache2.service - The Apache
jun 19 12:42:18 SVR-BASE-AZDINFARISSE apachectl[17059]: AH00558: apache2: Could not reli
jun 19 12:42:18 SVR-BASE-AZDINFARISSE systemd[1]: apache2.service: Deactivated successfu
jun 19 12:42:18 SVR-BASE-AZDINFARISSE systemd[1]: Stopped apache2.service - The Apache H
jun 19 12:42:18 SVR-BASE-AZDINFARISSE systemd[1]: Starting apache2.service - The Apache
jun 19 12:42:18 SVR-BASE-AZDINFARISSE apachectl[17068]: AH00558: apache2: Could not reli
jun 19 12:42:18 SVR-BASE-AZDINFARISSE systemd[1]: Started apache2.service - The Apache H
jun 19 12:51:30 SVR-BASE-AZDINFARISSE systemd[1]: Stopping apache2.service - The Apache
jun 19 12:51:30 SVR-BASE-AZDINFARISSE apachectl[18129]: AH00558: apache2: Could not reli
jun 19 12:51:30 SVR-BASE-AZDINFARISSE systemd[1]: apache2.service: Deactivated successfu

AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo journalctl -u saludo.service
jun 20 12:03:41 SVR-BASE-AZDINFARISSE systemd[1]: Starting saludo.service - Mensaje de s
jun 20 12:03:41 SVR-BASE-AZDINFARISSE systemd[1]: saludo.service: Deactivated successfu
jun 20 12:03:41 SVR-BASE-AZDINFARISSE systemd[1]: Finished saludo.service - Mensaje de s
lines 1-3/3 (END)
```

Ahora vamos filtrar los mensajes de error o advertencia con el comando:

```
sudo journalctl -p 3 -xb
```

Este comando significa:

-p 3: Que muestre solo mensajes con prioridad error.

-x: Añade explicaciones extendidas si están disponibles.

-b: Sólo desde el último boot.

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/$ sudo journalctl -p 3 -xb
jun 20 07:27:20 SVR-BASE-AZDINFARISSE systemd[1]: Invalid DMI field header.
jun 20 07:27:20 SVR-BASE-AZDINFARISSE kernel: vmwgfx 0000:00:02.0: [drm] *ERROR* vmwgfx
jun 20 07:27:20 SVR-BASE-AZDINFARISSE kernel: vmwgfx 0000:00:02.0: [drm] *ERROR* This co
jun 20 07:27:20 SVR-BASE-AZDINFARISSE kernel: vmwgfx 0000:00:02.0: [drm] *ERROR* Please
jun 20 07:27:20 SVR-BASE-AZDINFARISSE (udev-worker)[415]: vboxguest: /etc/udev/rules.d/60
jun 20 07:27:20 SVR-BASE-AZDINFARISSE (udev-worker)[402]: vboxuser: /etc/udev/rules.d/60
jun 20 07:27:25 SVR-BASE-AZDINFARISSE (udev-worker)[1419]: vboxguest: /etc/udev/rules.d/6
jun 20 07:27:25 SVR-BASE-AZDINFARISSE (udev-worker)[1419]: vboxuser: /etc/udev/rules.d/6
jun 20 07:27:25 SVR-BASE-AZDINFARISSE systemd[1]: Failed to start vboxadd.service.
Subject: A start job for unit vboxadd.service has failed
Defined-By: systemd
Support: http://www.ubuntu.com/support

A start job for unit vboxadd.service has finished with a failure.

The job identifier is 164 and the job result is failed.
jun 20 07:27:35 SVR-BASE-AZDINFARISSE gdm-password[2801]: gkr-pam: unable to locate dae
jun 20 07:27:35 SVR-BASE-AZDINFARISSE gdm3[2331]: Gdm: on_display_added: assertion 'GDM_
jun 20 07:27:35 SVR-BASE-AZDINFARISSE systemd[2816]: Failed to start app-gnome-gnome\x2d
Subject: A start job for unit UNIT has failed
Defined-By: systemd
Support: http://www.ubuntu.com/support

A start job for unit UNIT has finished with a failure.

The job identifier is 217 and the job result is failed.
jun 20 07:27:37 SVR-BASE-AZDINFARISSE gdm3[2331]: Gdm: on_display_removed: assertion 'GD
```


Ahora vamos a registrar el estado del servicio saludo y guardar una copia del log en **/srv/logs/saludo_journal.log**.

Primero crearemos la carpeta /srv/logs con el comando:

sudo mkdir -p /srv/logs

Y exportamos los logs del servicio saludo con el comando:

sudo journalctl -u saludo.service | sudo tee /srv/logs/saludo_journal.log > /dev/null

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /srv
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$ sudo mkdir -p /srv/logs
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$ ls
docs  logs  repositorios  www
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$ sudo journalctl -u saludo.service > /srv/logs/saludo_journal.log
bash: /srv/logs/saludo_journal.log: Permission denied
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$ sudo journalctl -u saludo.service | sudo tee /srv/logs/saludo_journal.log > /dev/null
[sudo] password for AZDINFARISSE:
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$ █
```

Y ahora vemos el archivo con el comando:

cat /srv/logs/saludo_journal.log

```
AZDINFARISSE@SVR-BASE-AZDINFARISSE: /srv
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$ cat /srv/logs/saludo_journal.log
jun 20 12:03:41 SVR-BASE-AZDINFARISSE systemd[1]: Starting saludo.service - Mensaje de saludo al iniciar el servidor...
jun 20 12:03:41 SVR-BASE-AZDINFARISSE systemd[1]: saludo.service: Deactivated successfully.
jun 20 12:03:41 SVR-BASE-AZDINFARISSE systemd[1]: Finished saludo.service - Mensaje de saludo al iniciar el servidor.
AZDINFARISSE@SVR-BASE-AZDINFARISSE:/srv$
```