

Tarea 4.2: Gestión de servicios con SystemD

Descripción de la tarea

Con la aparición del nuevo iniciador del sistema en 2010 conocido como SystemD surge una nueva forma de gestionar los servicios del sistema haciendo uso del comando `systemctl`. En este laboratorio gestionaremos el servicio Apache, el cual es un popular servidor de páginas web.

Para administrar servicios utilizaremos el comando `systemctl`, en este laboratorio utilizaremos el servicio `httpd.service` si estamos en distribuciones redhat o `apache2` si estamos en una distribución debian para lanzar los siguientes comandos.

Pasos de la tarea

1. Listar todos los servicios activos (ejecutándose) del sistema.

```
vagrant@debian:~$ systemctl list-units --type service
UNIT
LOAD    ACTIVE SUB    DESCRIPTION    >
acpid.service
loaded active running ACPI event daemon
...
```

2. Listar todos los servicios activos y también los inactivos.

```
vagrant@debian:~$ systemctl list-units --type service --all
UNIT
LOAD    ACTIVE SUB    DESCRIPTION>
acpid.service
loaded active running ACPI event >
apache2.service
loaded active running The Apache >
apparmor.service
loaded active exited Load AppArm>
● auditd.service
not-found inactive dead auditd.serv>
...
```

3. Instalar, iniciar, parar o reiniciar un servicio.

```
vagrant@debian:~$ sudo apt update
...
vagrant@debian:~$ sudo apt install -y apache2
...
vagrant@debian:~$ sudo su -
```

```
root@debian:~# systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset:
   enabled)
   Active: active (running) since Fri 2025-05-02 20:02:33 CEST; 50min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1826 ExecStart=/usr/sbin/apachectl start (code=exited,
   status=0/SUCCESS)
   Main PID: 1830 (apache2)
     Tasks: 55 (limit: 2303)
    Memory: 12.8M
       CPU: 1.325s
   CGroup: /system.slice/apache2.service
           └─1830 /usr/sbin/apache2 -k start
           └─1831 /usr/sbin/apache2 -k start
           └─1832 /usr/sbin/apache2 -k start

may 02 20:02:33 debian systemd[1]: Starting apache2.service - The Apache HTTP
Server...
may 02 20:02:33 debian systemd[1]: Started apache2.service - The Apache HTTP
Server.

root@debian:~# systemctl stop apache2
root@debian:~# systemctl start apache2
root@debian:~# systemctl restart apache2
```

4. Reiniciar únicamente si el servicio está ya iniciado.

```
root@debian:~# systemctl try-restart apache2
```

5. Recargar configuración.

```
root@debian:~# systemctl reload apache2
```

6. Comprobar el estado de un servicio o si se está habilitado (enabled).

```
root@debian:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset:
   enabled)
   Active: active (running) since Fri 2025-05-02 20:55:42 CEST; 1min 13s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 2450 ExecStart=/usr/sbin/apachectl start (code=exited,
   status=0/SUCCESS)
   Process: 2529 ExecReload=/usr/sbin/apachectl graceful (code=exited,
   status=0/SUCCESS)
   Main PID: 2454 (apache2)
```

```
Tasks: 55 (limit: 2303)
Memory: 8.8M
CPU: 270ms
CGroup: /system.slice/apache2.service
├─2454 /usr/sbin/apache2 -k start
├─2533 /usr/sbin/apache2 -k start
└─2534 /usr/sbin/apache2 -k start
```

```
may 02 20:55:41 debian systemd[1]: Starting apache2.service - The Apache HTTP
Server...
may 02 20:55:42 debian systemd[1]: Started apache2.service - The Apache HTTP
Server.
may 02 20:56:05 debian systemd[1]: Reloading apache2.service - The Apache HTTP
Server...
may 02 20:56:05 debian systemd[1]: Reloaded apache2.service - The Apache HTTP
Server.
root@debian:~# systemctl is-enabled apache2
enabled
```

7. Comprobar si un servicio está activado (ejecutándose).

```
root@debian:~# systemctl is-active apache2
active
```

8. Habilitar un servicio (para que se inicie al arrancar el sistema).

Habilitar un servicio (para que se inicie al arrancar el sistema)

9. Deshabilitar servicio.

```
root@debian:~# systemctl disable apache2
Synchronizing state of apache2.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed "/etc/systemd/system/multi-user.target.wants/apache2.service".
```

10. Hacer kill a un servicio (por defecto SIGTERM).

```
root@debian:~# systemctl kill apache2
root@debian:~# systemctl status apache2
o apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset:
enabled)
   Active: inactive (dead)
```

Docs: <https://httpd.apache.org/docs/2.4/>

...

11. Podemos especificar la señal que enviamos para hacer el kill con -s. Por ejemplo enviar SIGKILL.

```
root@debian:~# systemctl kill -s SIGKILL apache2
root@debian:~# systemctl status apache2
○ apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset:
   enabled)
   Active: inactive (dead)
     Docs: https://httpd.apache.org/docs/2.4/
   ...
```

12. También podemos hacer un reinicio, halt o apagado del sistema.

```
root@debian:~# systemctl reboot
root@debian:~# systemctl halt
root@debian:~# systemctl poweroff
```

13. También podemos enmascarar servicios, es decir incluso cuando deshabilitamos un servicio lo podemos arrancar, si queremos prohibir la ejecución de un servicio.

```
root@debian:~# systemctl mask apache2
Created symlink /etc/systemd/system/apache2.service → /dev/null.
root@debian:~# systemctl status apache2
● apache2.service
   Loaded: masked (Reason: Unit apache2.service is masked.)
   Active: active (running) since Fri 2025-05-02 21:30:57 CEST; 13s ago
 Main PID: 2931 (apache2)
    CPU: 113ms
   CGroup: /system.slice/apache2.service
           └─2931 /usr/sbin/apache2 -k start
             └─2932 /usr/sbin/apache2 -k start
               └─2933 /usr/sbin/apache2 -k start
```

Nota: Supongamos que tenemos como administrador que detener un servicio para gestionarlo y otro administrador en ese servicio ve que el servicio está parado y trata de activarlo, al estar enmascarado no va a poder realizarlo y así evitamos un conflicto.

```
root@debian:~# systemctl start apache2
Failed to start apache2.service: Unit apache2.service is masked.
```

14. Desenmascarar un servicio.

```

root@debian:~# systemctl unmask apache2
Removed "/etc/systemd/system/apache2.service".
root@debian:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset:
   enabled)
   Active: active (running) since Fri 2025-05-02 21:30:57 CEST; 39s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 2931 (apache2)
    Tasks: 55 (limit: 2303)
   Memory: 12.8M
      CPU: 122ms
   CGroup: /system.slice/apache2.service
           └─2931 /usr/sbin/apache2 -k start
           └─2932 /usr/sbin/apache2 -k start
           └─2933 /usr/sbin/apache2 -k start

```

15. Para ver cuánto tiempo ha empleado el arranque (de izquierda a derecha), del espacio del kernel, el espacio de los módulos del kernel, el espacio de los servicios y el tiempo total.

```

root@debian:~# systemd-analyze
Startup finished in 12.234s (kernel) + 9.559s (userspace) = 21.794s
graphical.target reached after 9.313s in userspace.

```

16. También podemos analizar el tiempo de inicio de cada uno de los servicios al arrancar el sistema con `systemd-analyze blame`:

```

root@debian:~# systemd-analyze blame
3.876s ifupdown-pre.service
2.749s dev-mapper-debian\x2d\x2d12\x2d\x2dvlg\x2droot.device
2.484s vboxadd.service
1.600s systemd-fsck@dev-disk-by\x2duuid-
33ff8406\x2dbd40\x2d4ce6\x2dafef\x2deaf5ead8acf9.service
1.586s e2scrub_reap.service
1.049s run-rpc_pipefs.mount
1.034s networking.service
933ms keyboard-setup.service
761ms systemd-udev-trigger.service
746ms lvm2-monitor.service
689ms chrony.service
...

```

Nota: La salida de este comando va a ordenar de lo que ha tardado más a menos, es interesante porque si tenemos por ejemplo un servicio nfs, con un punto de montaje que no podemos montar, veremos que el

servicio que mas tarda seria el rpc-statd.service.

17. También podemos ver el árbol de procesos iniciados por cada servicio.

```
root@debian:~# systemd-cgls
Control group /:
-.slice
├─user.slice (#218)
│  → user.invocation_id: 9d31c153634e4ddbafdc6d60e32bb99b
│  → trusted.invocation_id: 9d31c153634e4ddbafdc6d60e32bb99b
│  └─user-1000.slice (#3256)
│     → user.invocation_id: e23202c22a2a43d9b381890cbc6e2760
│     → trusted.invocation_id: e23202c22a2a43d9b381890cbc6e2760
│     └─session-9.scope (#3754)
│        └─3153 sshd: vagrant [priv]
│        └─3161 sshd: vagrant@pts/0
...

```

18. También podemos crear un archivo SVG que describe el proceso de arranque gráficamente y lo podemos abrir con nuestro navegador con el comando.

```
root@debian:~# systemd-analyze plot > /var/www/html/plot.html

```

Nota: En la IP de nuestra máquina en un navegador veremos el gráfico generado.