

# **ADMINISTRACIÓN DE SISTEMAS OPERATIVOS**

## **TEMA 2**

**Ejercicios de control de servicios y  
daemons en Red Hat**

**2CFSH**

GERARDO CANO

**ASO**

**Capítulo 9: Control de Servicios y Daemons (systemd).....2**

1. Fundamentos de systemd y Unidades .....2

2. Monitoreo y Visualización de Unidades (systemctl) .....3

3. Control de Servicios .....3

    A. Control en Tiempo de Ejecución (Runtime) .....3

    B. Control Persistente (Boot-time).....4

    C. Enmascaramiento (Masking) .....4

**Trabajo de laboratorio:.....5**

# CAPÍTULO 9: CONTROL DE SERVICIOS Y DAEMONS (SYSTEMD)

Este capítulo introduce el sistema de inicialización y gestor de servicios **systemd**, el primer proceso del sistema (PID 1) en Red Hat Enterprise Linux.

## 1. FUNDAMENTOS DE SYSTEMD Y UNIDADES

El daemon `systemd` gestiona los procesos del sistema, aportando mejoras clave:

- **Paralelización:** Inicia múltiples servicios simultáneamente, acelerando el arranque.
- **Gestión de Dependencias:** Un servicio que depende de otro (ej. de la red) no se inicia hasta que su dependencia esté disponible.
- **Activación a Demanda:** Inicia daemons solo cuando son necesarios.

### Tipos de Unidades

Una **Unidad** es un archivo de configuración que define los recursos gestionados por `systemd`.

Tipo de Unidad	Extensión	Descripción	Uso Típico
Servicio	.service	Representa daemons de ejecución continua o acciones de una sola vez ( <i>oneshot</i> ).	Servidor web, SSH.
Socket	.socket	Monitorea una dirección de comunicación; inicia el servicio asociado solo cuando se recibe una conexión.	Ahorrar recursos demorando el inicio.
Ruta	.path	Demora la activación de un servicio hasta que un archivo o directorio específico cambie.	Servicios de cola de impresión o directorios de entrada.

## 2. MONITOREO Y VISUALIZACIÓN DE UNIDADES (SYSTEMCTL)

El comando `systemctl` es la herramienta central para inspeccionar y controlar los estados del sistema.

### Listado de Unidades

Comando	Función	Salida
<code>systemctl list-units --type=service</code>	Muestra solo los servicios que están <b>cargados y activos</b> .	UNIT, LOAD, ACTIVE, SUB, DESCRIPTION.
<code>systemctl list-units --all</code>	Muestra todas las unidades cargadas, independientemente de su estado de activación.	Incluye unidades <b>inactive</b> .
<code>systemctl list-unit-files --type=service</code>	Muestra <b>todos</b> los archivos de unidad instalados y su estado de arranque.	Muestra <b>enabled, disabled, static, masked</b> .
<code>systemctl list-dependencies &lt;UNIT&gt;</code>	Muestra una jerarquía de unidades de las que depende la unidad especificada para iniciarse.	Útil para la resolución de problemas de dependencias.

### Verificación de Estado

El comando `systemctl status <servicio>` ofrece un informe detallado. Alternativamente, se pueden verificar estados específicos:

Comando	Verifica	Posibles Resultados
<code>systemctl is-active &lt;servicio&gt;</code>	Si el servicio está en ejecución.	<b>active</b> o <b>inactive</b> .
<code>systemctl is-enabled &lt;servicio&gt;</code>	Si el servicio está configurado para iniciar en el arranque.	<b>enabled</b> o <b>disabled</b> .
<code>systemctl is-failed &lt;servicio&gt;</code>	Si el servicio experimentó un error al iniciar.	<b>active</b> o <b>failed</b> .

## 3. CONTROL DE SERVICIOS

### A. CONTROL EN TIEMPO DE EJECUCIÓN (RUNTIME)

Estos comandos afectan al servicio en la sesión actual, pero no cambian su configuración de arranque.

Comando	Función	Descripción
<code>systemctl start &lt;servicio&gt;</code>	Inicia el servicio.	
<code>systemctl stop &lt;servicio&gt;</code>	Detiene el servicio.	
<code>systemctl restart &lt;servicio&gt;</code>	Detiene y vuelve a iniciar el servicio (cambia el PID).	
<code>systemctl reload &lt;servicio&gt;</code>	Recarga la configuración sin interrumpir el servicio (no cambia el PID).	Solo funciona si el daemon lo soporta.
<code>systemctl reload-or-restart &lt;servicio&gt;</code>	Intenta recargar; si falla o no es soportado, reinicia.	
<code>systemctl daemon-reload</code>	<b>Obligatorio</b> para que <code>systemd</code> lea los cambios en los archivos de unidad.	

B. CONTROL PERSISTENTE (BOOT-TIME)

Estos comandos modifican el comportamiento del servicio en el próximo arranque, creando o eliminando enlaces simbólicos en directorios como `/etc/systemd/system/*.target.wants/`.

Comando	Función	Consecuencia
<code>systemctl enable &lt;servicio&gt;</code>	Configura el servicio para que inicie en el arranque.	No lo inicia en la sesión actual.
<code>systemctl disable &lt;servicio&gt;</code>	Evita que el servicio inicie automáticamente en el arranque.	No lo detiene en la sesión actual.
<code>systemctl enable --now &lt;servicio&gt;</code>	Habilita el inicio automático y lo inicia en la sesión actual.	
<code>systemctl disable --now &lt;servicio&gt;</code>	Deshabilita el inicio automático y lo detiene en la sesión actual.	

C. ENMASCARAMIENTO (MASKING)

Enmascarar un servicio lo bloquea completamente, creando un enlace simbólico a `/dev/null` en el directorio de configuración.

- **`systemctl mask <servicio>`:** Evita que el servicio se inicie de forma manual o automática.
  - **Diferencia clave:** Un servicio `disabled` se puede iniciar manualmente; uno `masked` no.
- **`systemctl unmask <servicio>`:** Quita el enlace a `/dev/null` y permite que el servicio vuelva a iniciarse.

# TRABAJO DE LABORATORIO:

## Control de servicios y daemons

### Preparación Inicial

1. Inicio el laboratorio en la máquina workstation:

```
[student@workstation ~]$ lab start services-review
```

2. Inicio sesión en serverb como student:

```
[student@workstation ~]$ ssh student@serverb
```

(Todos los comandos siguientes se ejecutan en serverb.)

### Gestión del Servicio psacct

El objetivo es iniciar el servicio psacct y configurarlo para que se habilite (inicie) al arrancar.

1. Iniciar el Servicio psacct

Como estoy gestionando servicios que a menudo requieren privilegios de administrador, tengo que usar sudo.

```
[student@serverb ~]$ sudo systemctl start psacct
```

2. Habilitar el Servicio psacct

Esto asegura que el servicio se inicie automáticamente después de un reinicio.

```
[student@serverb ~]$ sudo systemctl enable psacct
```



## Gestión del Servicio rsyslog

El objetivo es detener el servicio rsyslog y configurarlo para que se deshabilite (no inicie) al arrancar.

### 1. Detener el Servicio rsyslog

```
[student@serverb ~]$ sudo systemctl stop rsyslog
```

### 2. Deshabilitar el Servicio rsyslog

Esto evita que el servicio se inicie automáticamente después de un reinicio.

```
[student@serverb ~]$ sudo systemctl disable rsyslog
```

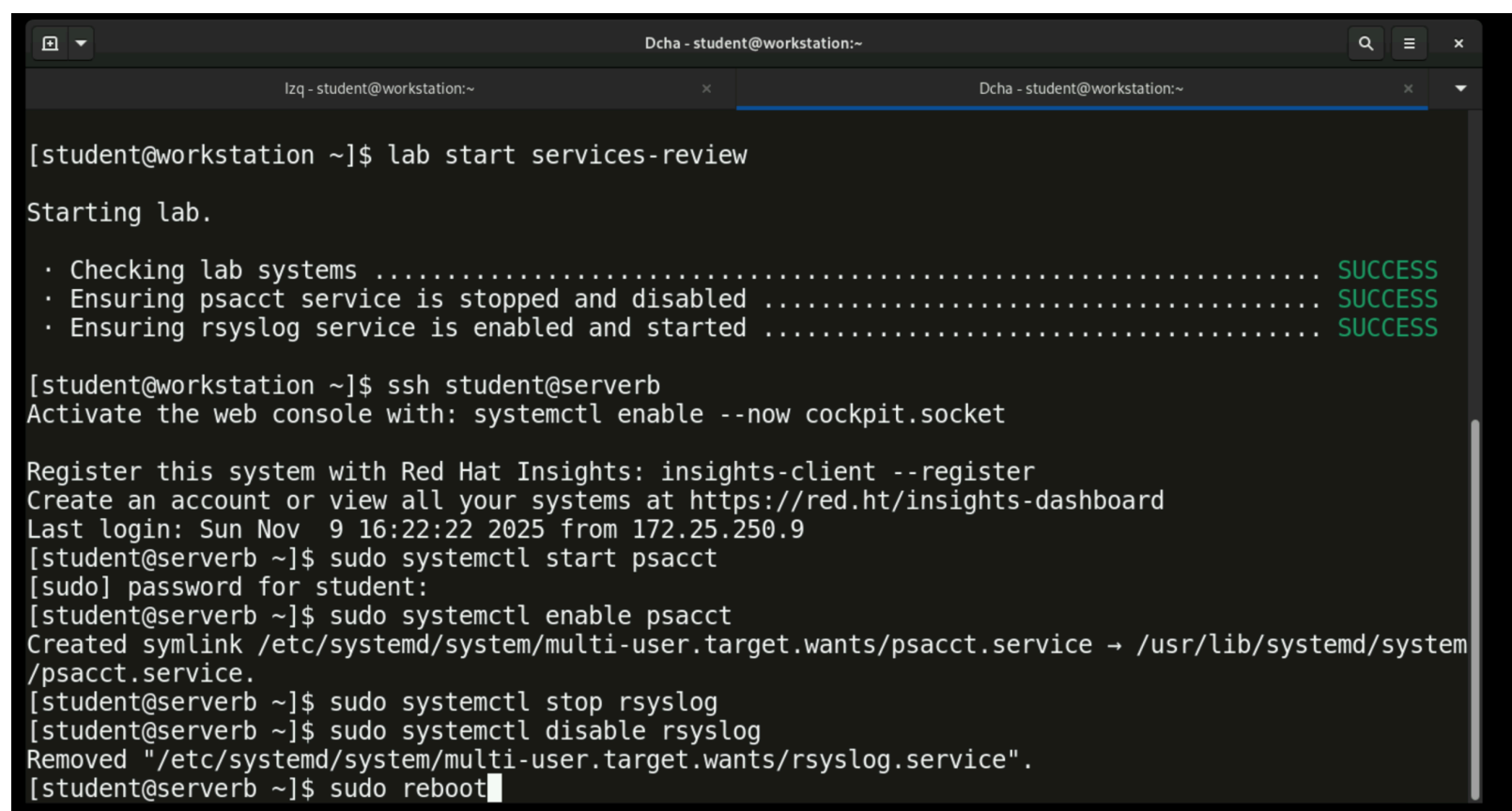
## Reinicio y Evaluación

El ejercicio requiere que se reinicie serverb para verificar que los ajustes de enable/disable funcionaron correctamente.

### 1. Reinicio la máquina serverb:

```
[student@serverb ~]$ sudo reboot (
```

(Se cierra la sesión SSH.)



```
Dcha - student@workstation:~
lzq - student@workstation:~
Dcha - student@workstation:~

[student@workstation ~]$ lab start services-review
Starting lab.
• Checking lab systems ..... SUCCESS
• Ensuring psacct service is stopped and disabled ..... SUCCESS
• Ensuring rsyslog service is enabled and started ..... SUCCESS

[student@workstation ~]$ ssh student@serverb
Activate the web console with: systemctl enable --now cockpit.socket

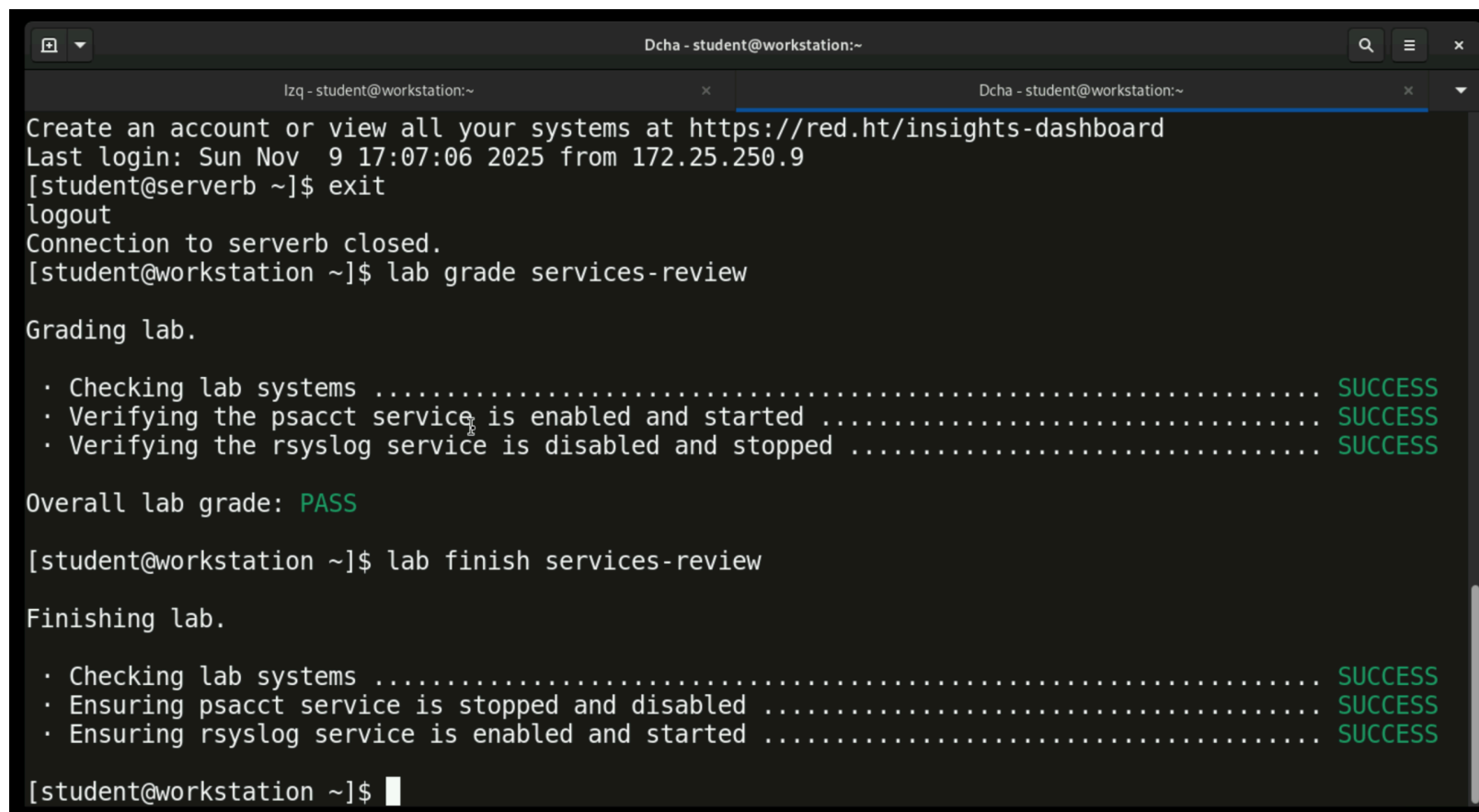
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Sun Nov  9 16:22:22 2025 from 172.25.250.9
[student@serverb ~]$ sudo systemctl start psacct
[sudo] password for student:
[student@serverb ~]$ sudo systemctl enable psacct
Created symlink /etc/systemd/system/multi-user.target.wants/psacct.service → /usr/lib/systemd/system/psacct.service.
[student@serverb ~]$ sudo systemctl stop rsyslog
[student@serverb ~]$ sudo systemctl disable rsyslog
Removed "/etc/systemd/system/multi-user.target.wants/rsyslog.service".
[student@serverb ~]$ sudo reboot
```

2. Vuelvo a iniciar sesión en serverb para confirmar que el reinicio fue exitoso (y salgo a workstation):

```
[student@workstation ~]$ ssh student@serverb  
[student@serverb ~]$ exit
```

3. Evalúo el trabajo en la máquina workstation:

```
[student@workstation ~]$ lab grade services-review
```



```
Dcha - student@workstation:~  
lzq - student@workstation:~  
Create an account or view all your systems at https://red.ht/insights-dashboard  
Last login: Sun Nov 9 17:07:06 2025 from 172.25.250.9  
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$ lab grade services-review  
Grading lab.  
· Checking lab systems ..... SUCCESS  
· Verifying the psacct service is enabled and started ..... SUCCESS  
· Verifying the rsyslog service is disabled and stopped ..... SUCCESS  
Overall lab grade: PASS  
[student@workstation ~]$ lab finish services-review  
Finishing lab.  
· Checking lab systems ..... SUCCESS  
· Ensuring psacct service is stopped and disabled ..... SUCCESS  
· Ensuring rsyslog service is enabled and started ..... SUCCESS  
[student@workstation ~]$
```

4. Finalizo el ejercicio en workstation:

```
[student@workstation ~]$ lab finish services-review
```