

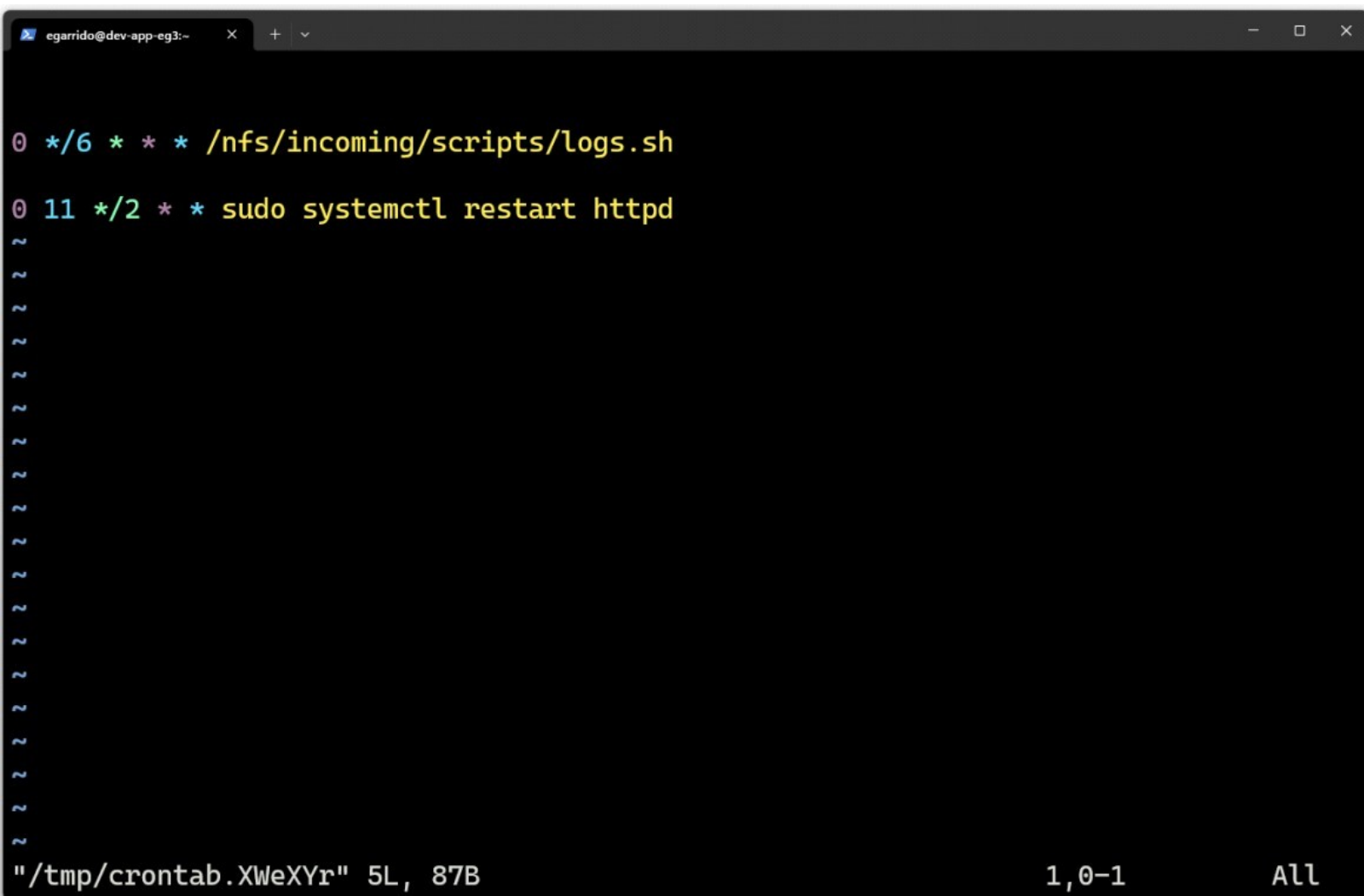
Privileged Command Delegation via Sudoers Configuration

This configuration demonstrates controlled privilege delegation using the Linux sudoers policy framework. Rather than granting full administrative access, a targeted sudo rule is implemented to allow a specific user to execute a defined system management command without requiring a password.

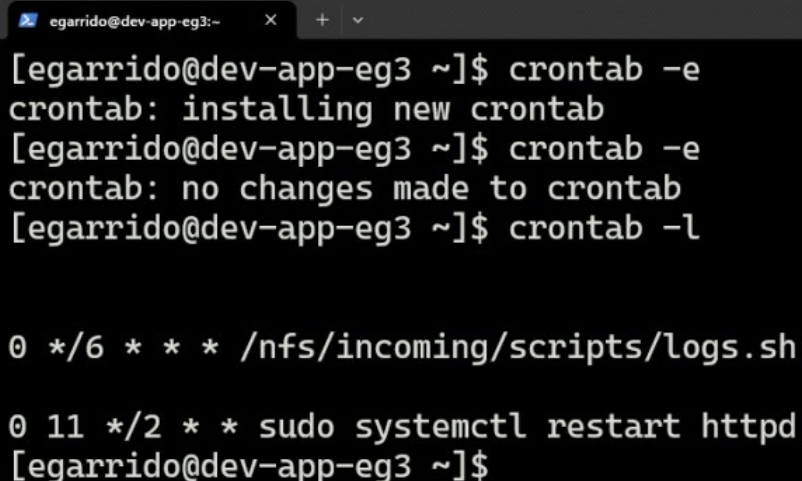
The sudoers file is reviewed and edited with elevated privileges to preserve default system policies while introducing a custom, least-privilege rule. This rule permits passwordless execution of an Apache HTTP service restart using systemctl, enabling operational tasks to be automated or delegated safely.

By limiting elevated access to a single command, this approach reduces risk while maintaining operational flexibility. It reflects common enterprise practices for secure service management, separation of duties, and automation readiness in Linux environments.

Two scheduled jobs are displayed as part of the user's crontab, viewed using the `crontab -e` command. One entry runs a script from an NFS-mounted directory every six hours, while another entry is configured to restart the Apache HTTP service at a defined recurring interval using `systemctl`, confirming automated task scheduling is in place.

A terminal window with a dark background and light-colored text. The window title bar shows 'egarrido@dev-app-eg3:~'. The terminal displays the output of the 'crontab -e' command, showing two cron entries. The first entry is '*/*6 * * * /nfs/incoming/scripts/logs.sh' and the second is '11 */2 * * sudo systemctl restart httpd'. There are several tilde characters (~) on the left side of the terminal, indicating a scrollable history. At the bottom, the prompt shows the file path '/tmp/crontab.XWeXYr' with a size of 5L, 87B. The bottom right corner shows '1,0-1' and 'All'.

The session shows the user creating and editing a crontab using the `crontab -e` command, followed by verification with `crontab -l`. The listed entries confirm two scheduled tasks are active: one running a script from an NFS-mounted directory every six hours, and another periodically restarting the Apache HTTP service using `systemctl`, demonstrating successful cron configuration and validation.

A terminal window with a dark background and light green text. The window title is 'egarrido@dev-app-eg3:~'. The user enters three commands: 'crontab -e', 'crontab -e', and 'crontab -l'. The first two commands result in messages about installing or not changing the crontab. The third command lists two cron jobs: one running a script every 6 hours and another restarting httpd every 11 minutes.

```
egarrido@dev-app-eg3:~$ crontab -e
crontab: installing new crontab
egarrido@dev-app-eg3:~$ crontab -e
crontab: no changes made to crontab
egarrido@dev-app-eg3:~$ crontab -l

0 */6 * * * /nfs/incoming/scripts/logs.sh
0 11 */2 * * sudo systemctl restart httpd
egarrido@dev-app-eg3:~$
```

A sudoers configuration file is shown being edited with elevated privileges, displaying default sudo policy entries and group-based permissions. A custom rule is added granting a specific user passwordless permission to restart the Apache HTTP service using systemctl, enabling controlled service management without requiring full administrative access.

```
root@dev-app-eg3:~  
##  
##      user      MACHINE=COMMANDS  
##  
## The COMMANDS section may have other options added to it.  
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)        ALL  
  
## Allows members of the 'sys' group to run networking, software,  
## service management apps and more.  
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCAT  
E, DRIVERS  
  
## Allows people in group wheel to run all commands  
%wheel  ALL=(ALL)        ALL  
  
## Same thing without a password  
# %wheel      ALL=(ALL)        NOPASSWD: ALL  
egarrido ALL=(ALL) NOPASSWD: /bin/systemctl restart httpd  
  
## Allows members of the users group to mount and unmount the  
## cdrom as root  
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom  
  
-- INSERT --
```

Summary

Core Linux service management and automation tasks were completed across the environment. Scheduled jobs were configured and validated using `crontab -e` and `crontab -l` to automate script execution from an NFS-mounted directory and perform recurring Apache service restarts.

To support automation without over-privileging users, a targeted sudoers rule was implemented granting passwordless access to restart the Apache HTTP service. This approach maintains system security while enabling reliable, non-interactive service management.

Together, these changes demonstrate practical enterprise Linux administration skills, including task scheduling, service automation, and least-privilege access control.