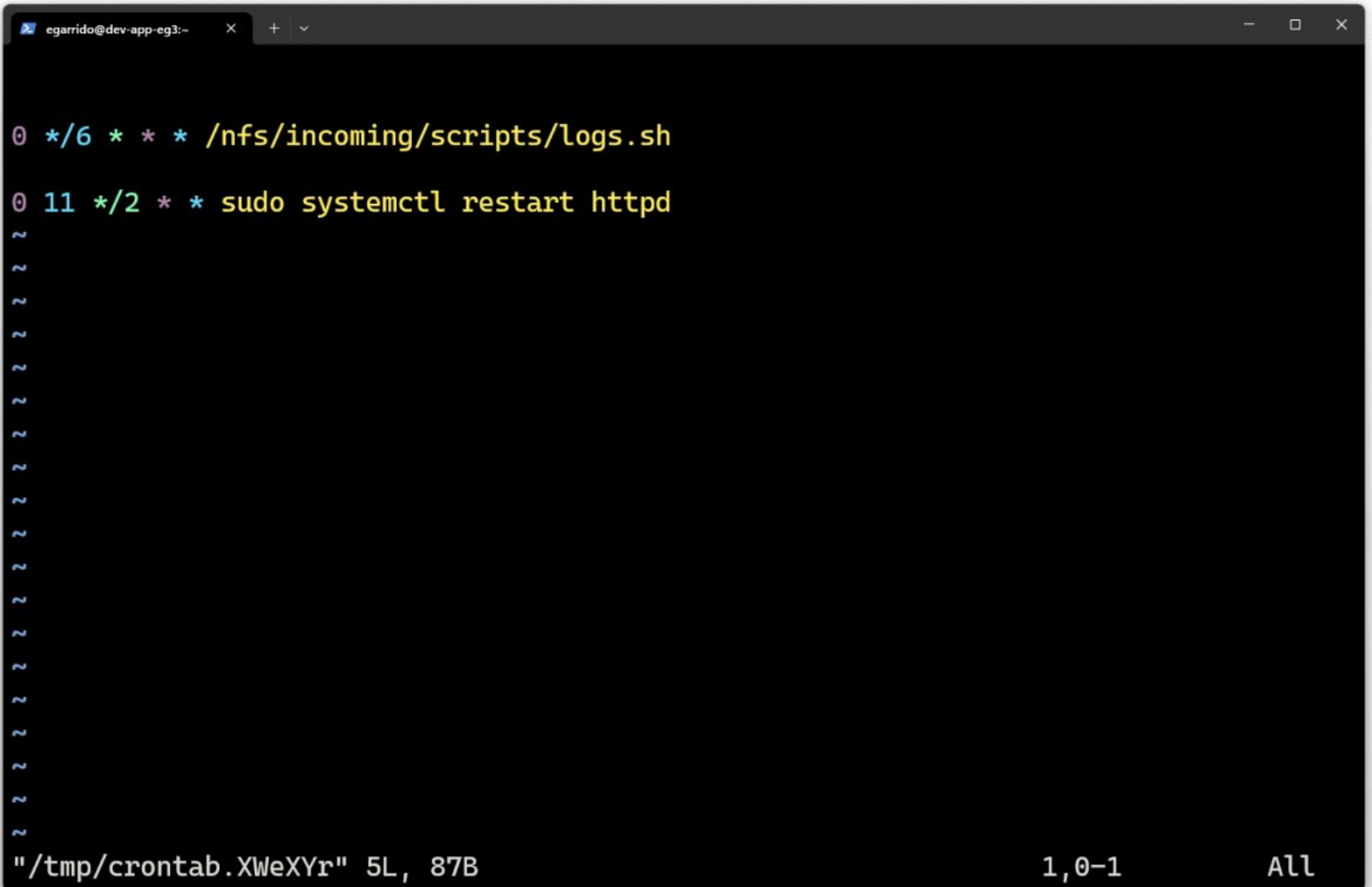
Privileged Command Delegation via Sudoers Configuration

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.

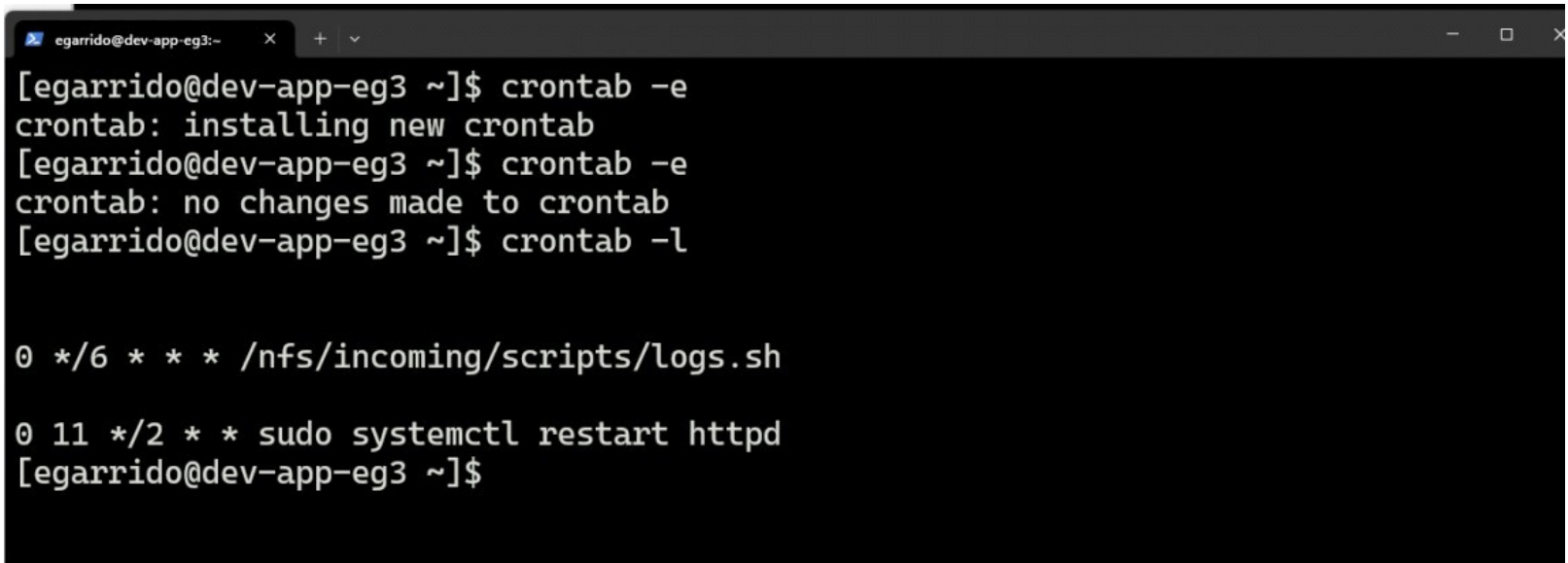
The sudoers file is reviewed and edited with elevated privileges to preserve default system policies while introducing a custom, least-privilege rule. This rule permits passwordless execution of an Apache HTTP service restart using systemctl, enabling operational tasks to be automated or delegated safely.

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.

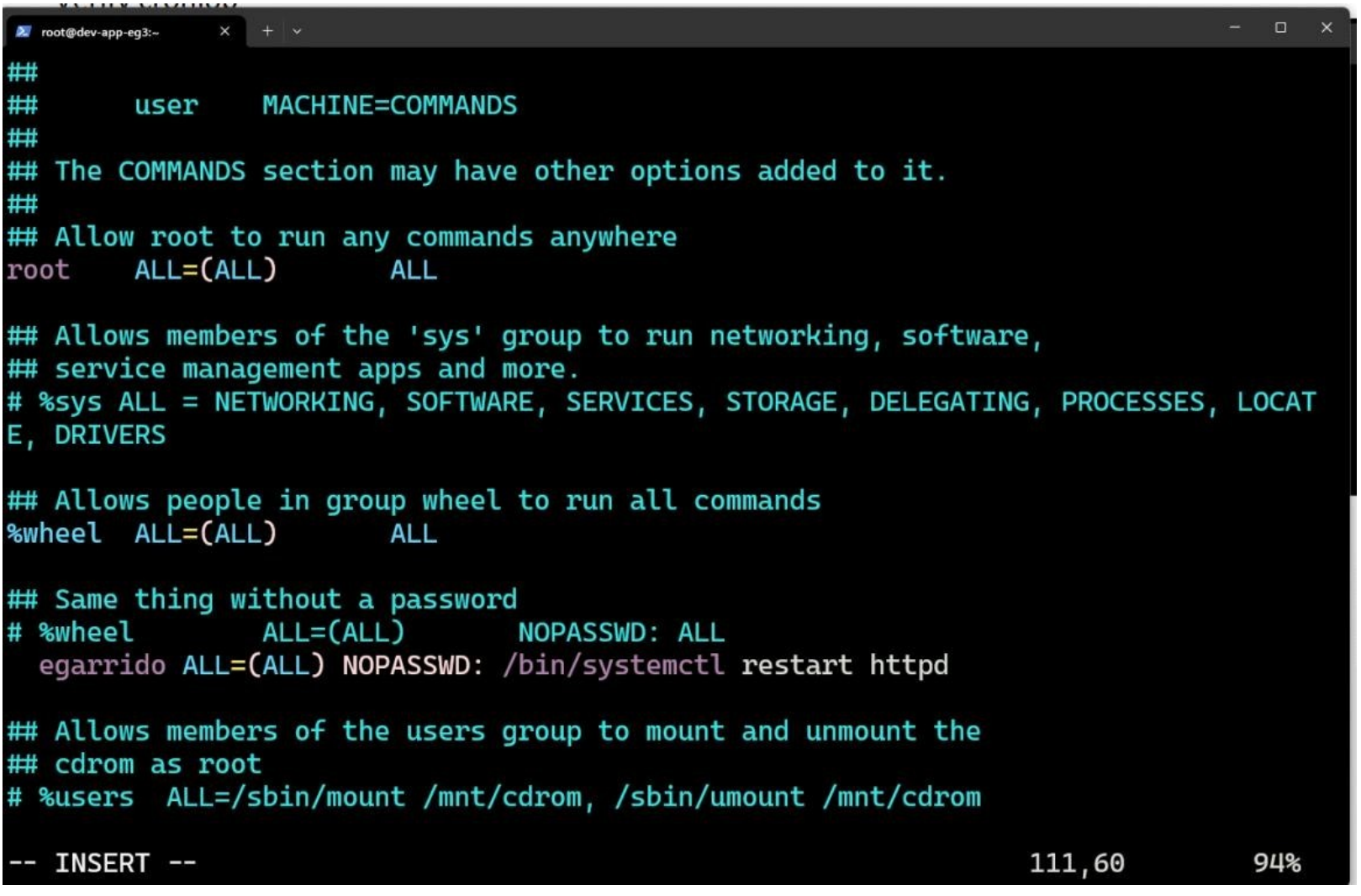
Two scheduled jobs are displayed as part of the user’s crontab, viewed using the crontab -e command. One entry runs a script from an NFS-mounted directory every six hours, while another entry is configured to restart the Apache HTTP service at a defined recurring interval using systemctl, confirming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



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 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.



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.