SSH Service Configuration, Security, and Access Validation

TICKET Summary (Last 5 TICKETS)

This series of TICKETS documents the end-to-end configuration, validation, and hardening of secure SSH access on a Linux server. The work covers service availability, authentication auditing, permission enforcement, and firewall configuration to ensure reliable and secure remote access aligned with enterprise best practices.

Across these TICKETS, the Opens Sh service is verified to be enabled and running, authentication events are reviewed through system logs, and controlled service restarts are performed to apply configuration changes. User SSH key access is secured by enforcing correct ownership and permission settings on home directories, .ssh directories, and authorized_keys files, preventing unauthorized access and common SSH misconfigurations.

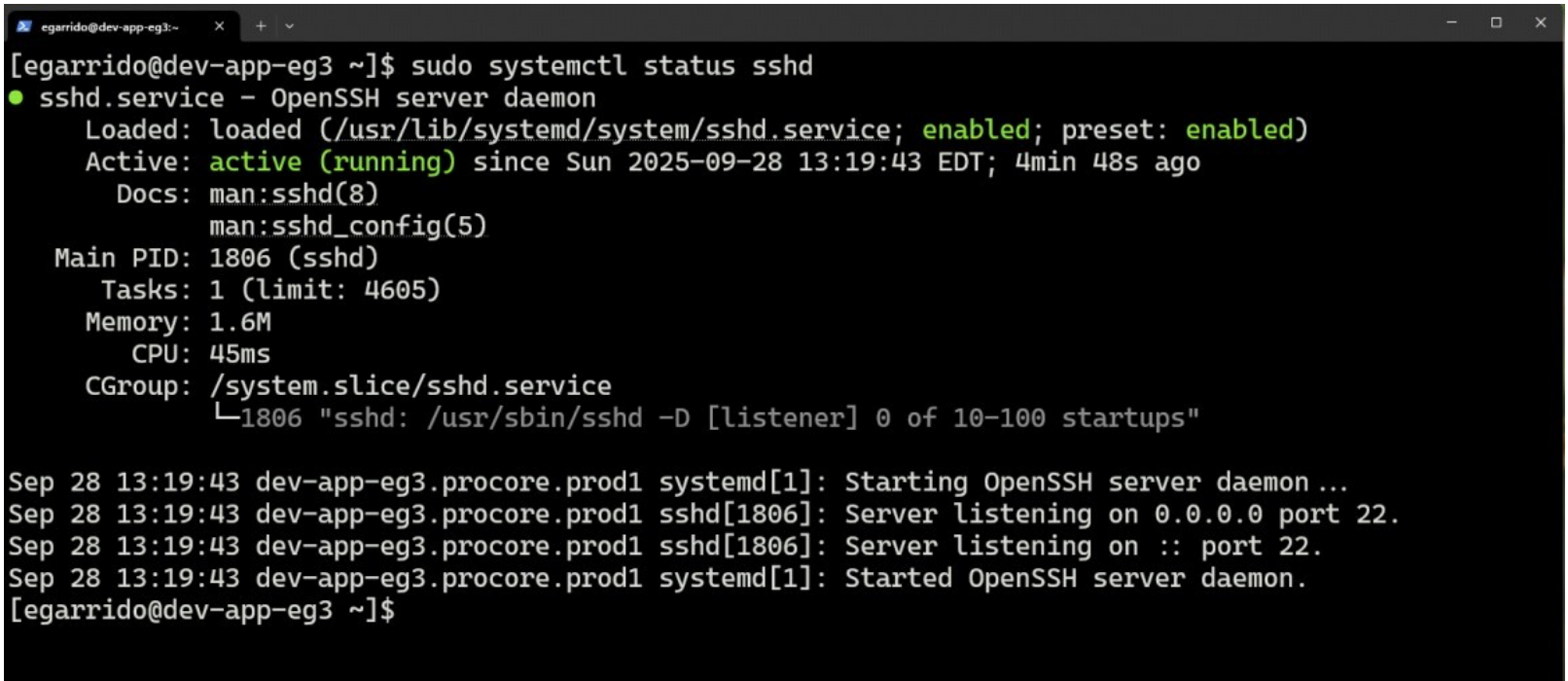
Firewall rules are validated and reloaded to ensure SSH traffic is persistently allowed without interrupting active services. System logs confirm successful authentication, session handling, and clean service restarts, providing full operational and audit visibility.

Outcome:

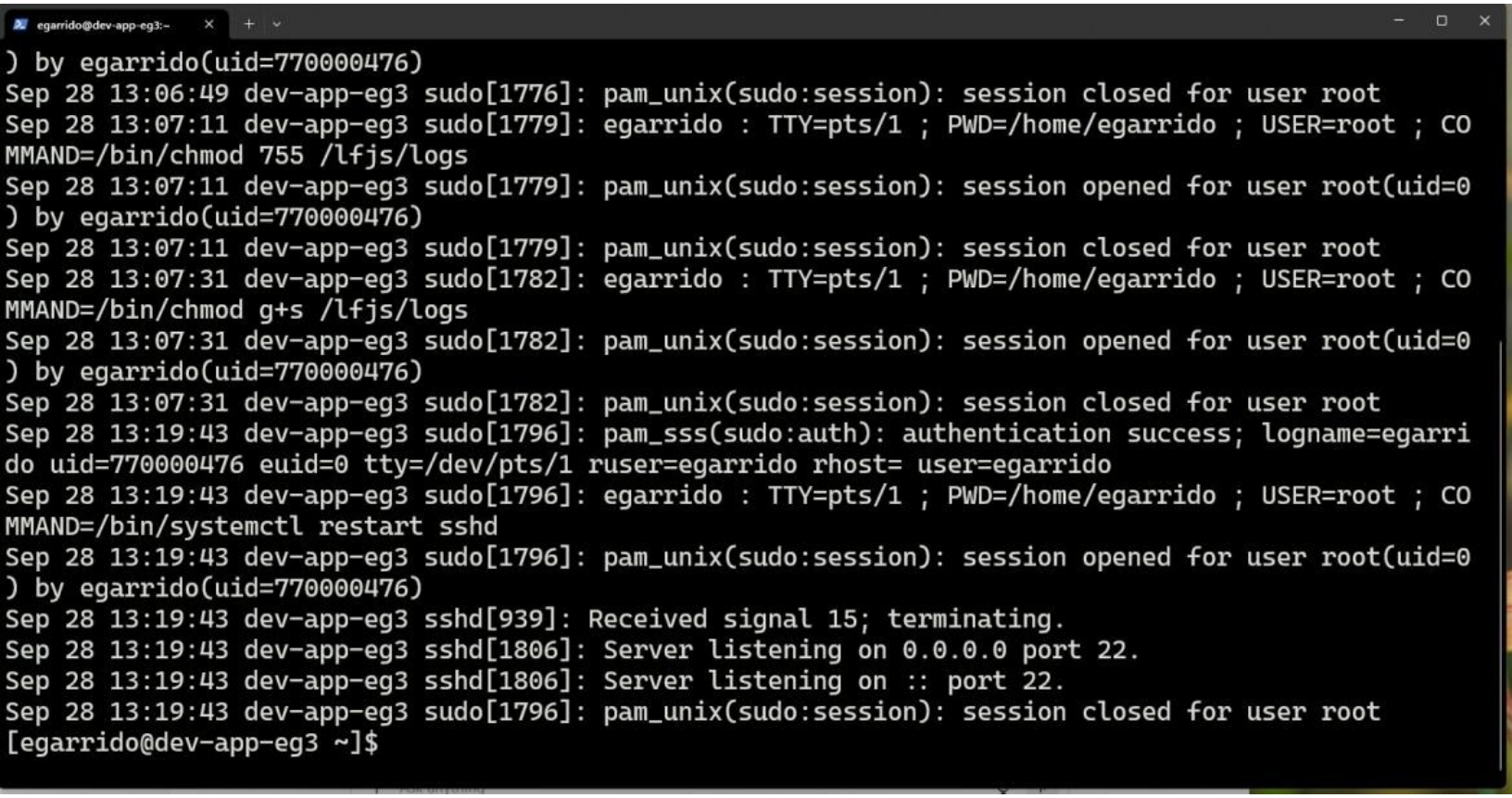
A fully functional and secure SSH configuration with verified service status, hardened key-based authentication, validated firewall access, and documented audit logs—demonstrating a production-ready approach to Linux remote access management.

1:16 PM

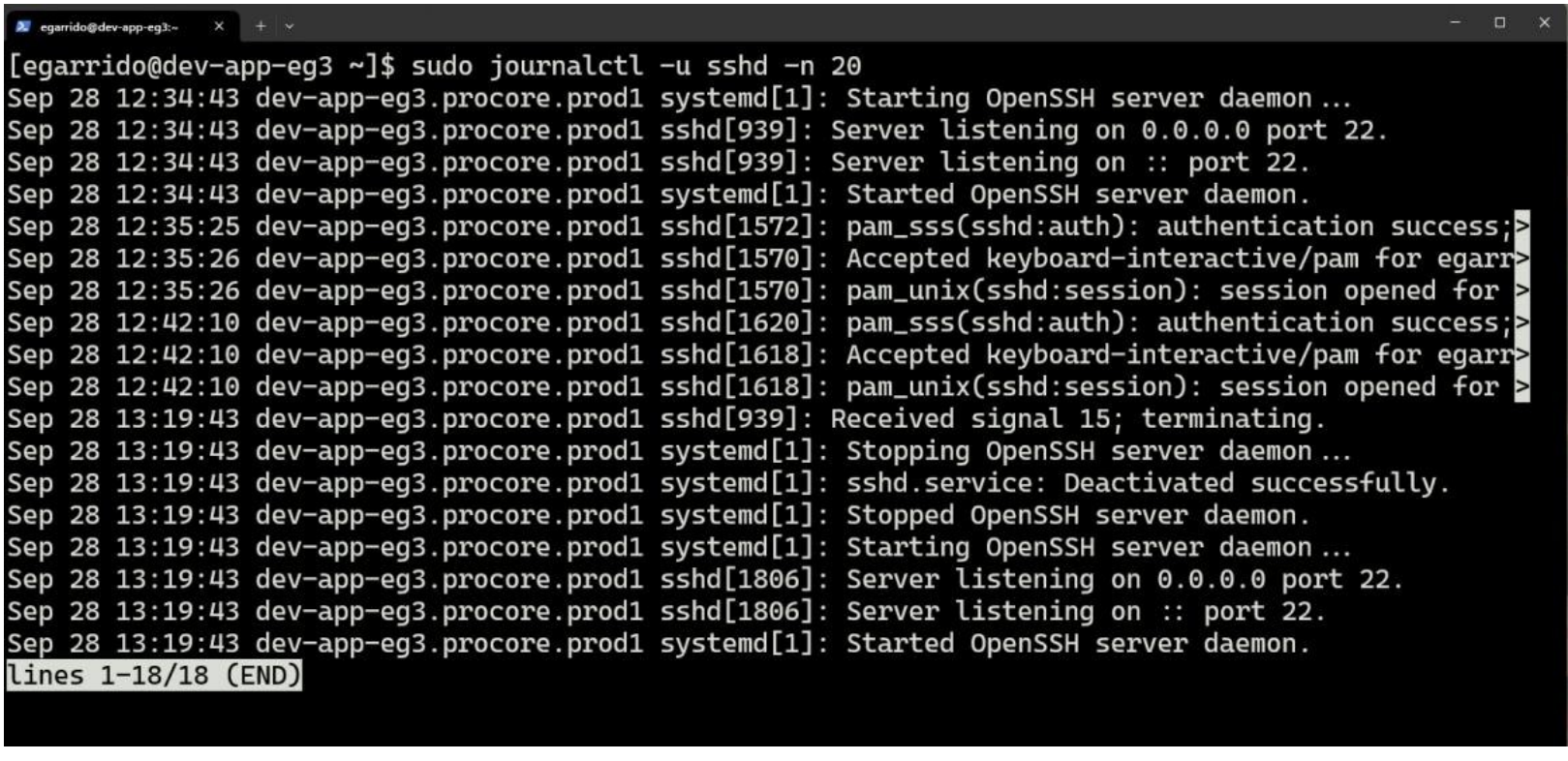
This output confirms that the OpenSSH server daemon (sshd) is installed, enabled, and actively running on the system. The service is loaded via systemd, set to start automatically at boot, and is currently listening on TCP port 22 for both IPv4 and IPv6 connections. The status also shows the main SSH daemon process ID, resource usage, and recent startup logs indicating a successful service launch.



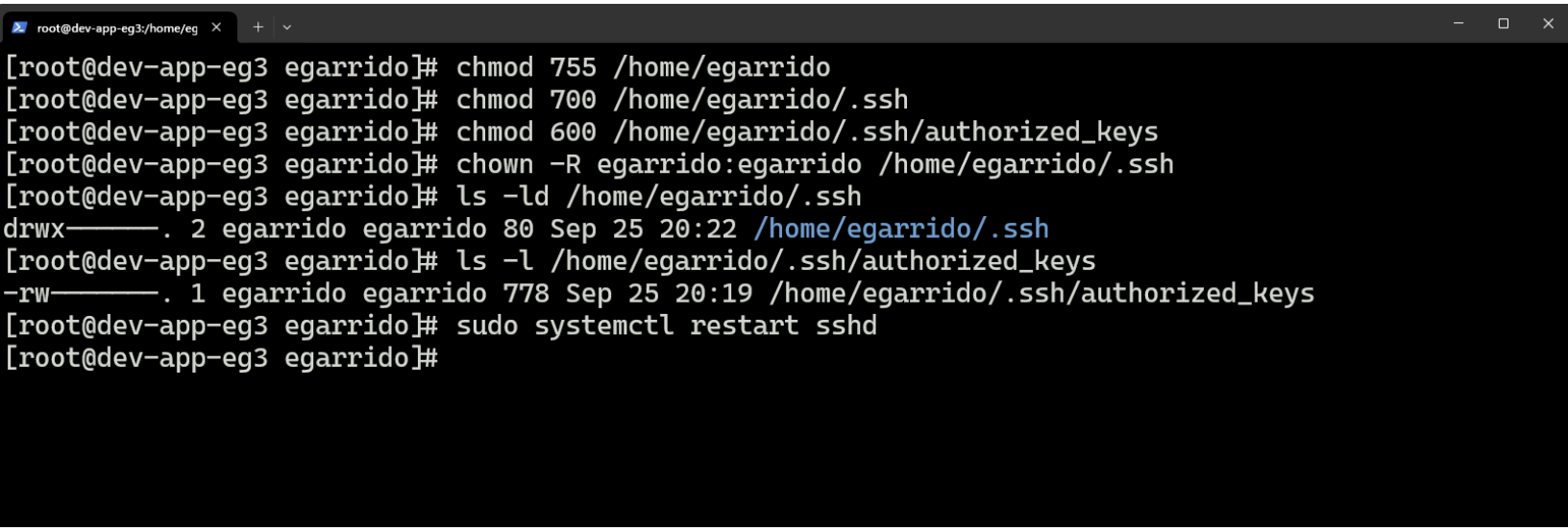
This log output shows authenticated sudo activity performed by the user egarrido, including permission changes on system directories and a controlled restart of the OpenSSH service. The entries confirm successful PAM authentication, privilege escalation to root, and proper session open/close handling. The SSH daemon receives a termination signal and restarts cleanly, resuming listening on TCP port 22 for both IPv4 and IPv6, indicating a successful and orderly service restart.



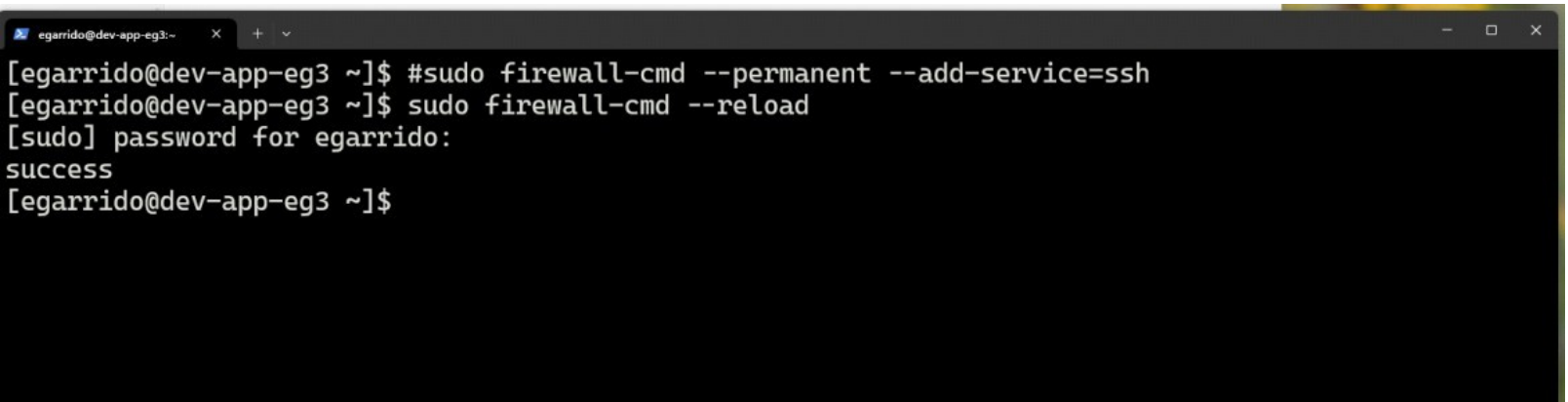
This output shows recent OpenSSH service events retrieved via journalctl, including service startup, authenticated user access, and a controlled restart of the SSH daemon. The logs confirm successful PAM authentication for user access, session creation, and orderly shutdown and restart of the sshd service. After receiving a termination signal, the service is restarted successfully and resumes listening on TCP port 22 over both IPv4 and IPv6, verifying stable and secure remote access availability



This sequence shows corrective permission and ownership configuration for a user's SSH directory to meet OpenSSH security requirements. The home directory is set to appropriate execute permissions, the .ssh directory is restricted to the owner, and the authorized_keys file is secured with read/write access for the user only. Ownership is recursively assigned to the correct user and group, and permissions are verified using ls. The SSH service is then restarted to ensure the changes are applied successfully.



This step shows the firewall being reloaded after ensuring SSH access is permitted. The firewall rules are applied successfully, confirming that SSH connectivity on TCP port 22 is allowed through the system firewall. Reloading the firewall activates the updated configuration without disrupting active services, ensuring secure and persistent remote access.



These TICKETS document the verification, configuration, and security hardening of SSH access on a Linux server. The OpenSSH service is confirmed to be enabled and running, with authentication and session activity validated through system and journal logs. Controlled service restarts are performed to safely apply configuration changes.

User SSH access is secured by enforcing proper ownership and permission settings on home directories, .ssh directories, and authorized_keys files to meet OpenSSH security requirements. Firewall rules are reloaded to ensure persistent SSH access on TCP port 22 without disrupting services. Log entries confirm successful authentication, clean service restarts, and reliable remote connectivity.