Centralized Logging with Rsyslog & Graylog (CentOS Stream 9)

This project demonstrates the implementation of centralized system logging using rsyslog clients forwarding logs to a Graylog server in a multi-node CentOS Stream 9 environment. The objective was to ensure reliable, centralized visibility into system activity across web, application, and performance hosts while following security and operational best practices.

Overview

Configured rsyslog on multiple CentOS Stream 9 servers to forward all system logs to a centralized Graylog instance using the Syslog Protocol (TCP/514).

Created a dedicated rsyslog configuration file (/etc/rsyslog.d/90-graylog.conf) to ensure clean, modular, and maintainable logging rules.

Restarted and enabled rsyslog to persist across reboots and validated service health using systemctl.

Verified log ingestion end-to-end by generating test events and confirming successful indexing and visibility in the Graylog web interface.

Key Tasks Performed

Installed and updated rsyslog packages

Enabled and validated rsyslog as an active system service

Configured centralized log forwarding to Graylog

Restarted services and confirmed runtime status

Generated test logs using logger

Validated ingestion, indexing, and searchability in Graylog

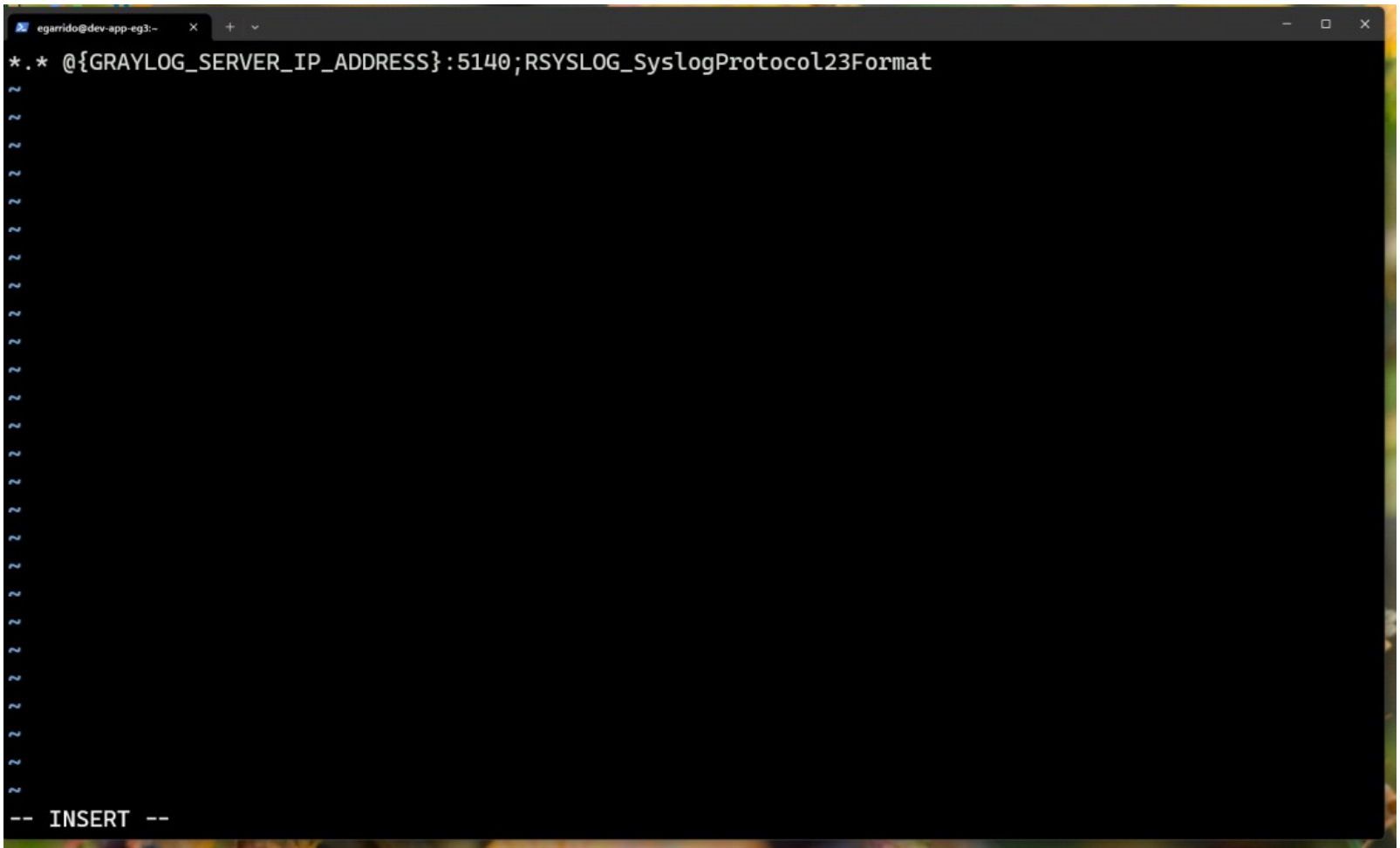
Confirmed host-specific log visibility using Graylog search filters

Validation

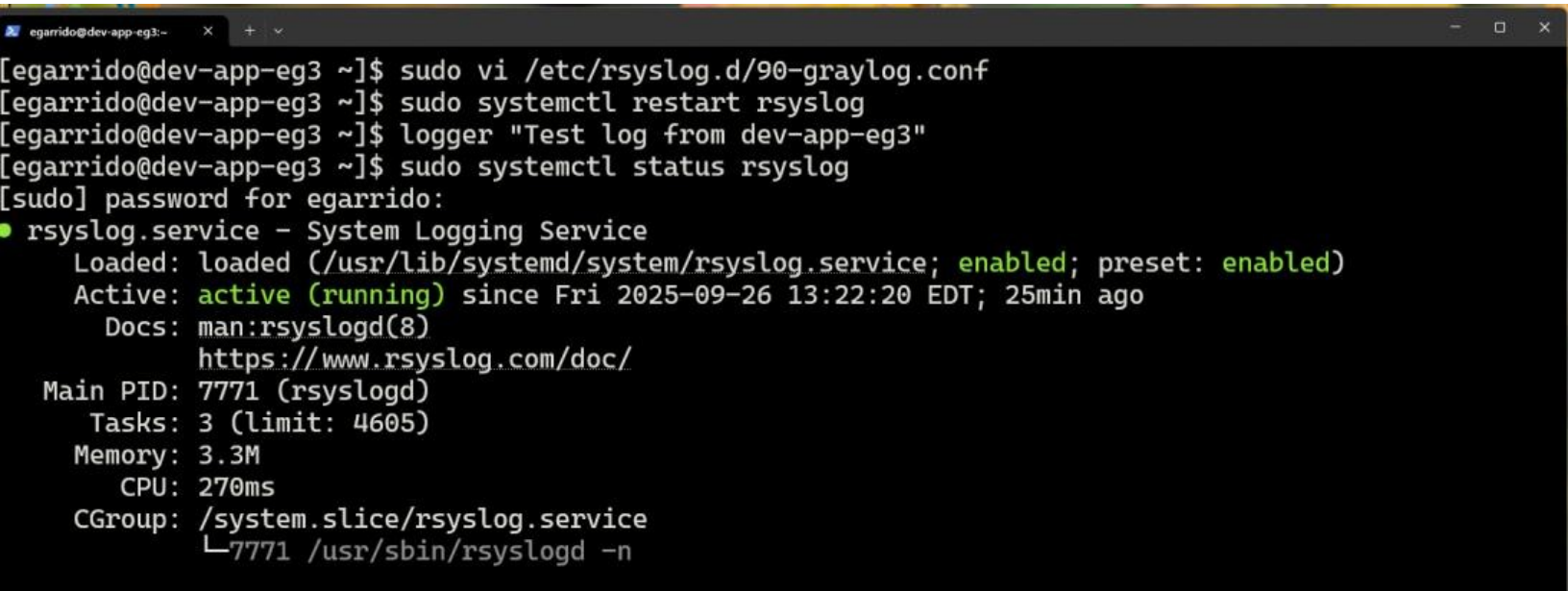
Graylog dashboards and search results confirm logs are being received in real time.

System events such as service restarts, sudo activity, and journal reloads are visible and correctly attributed to their source hosts.

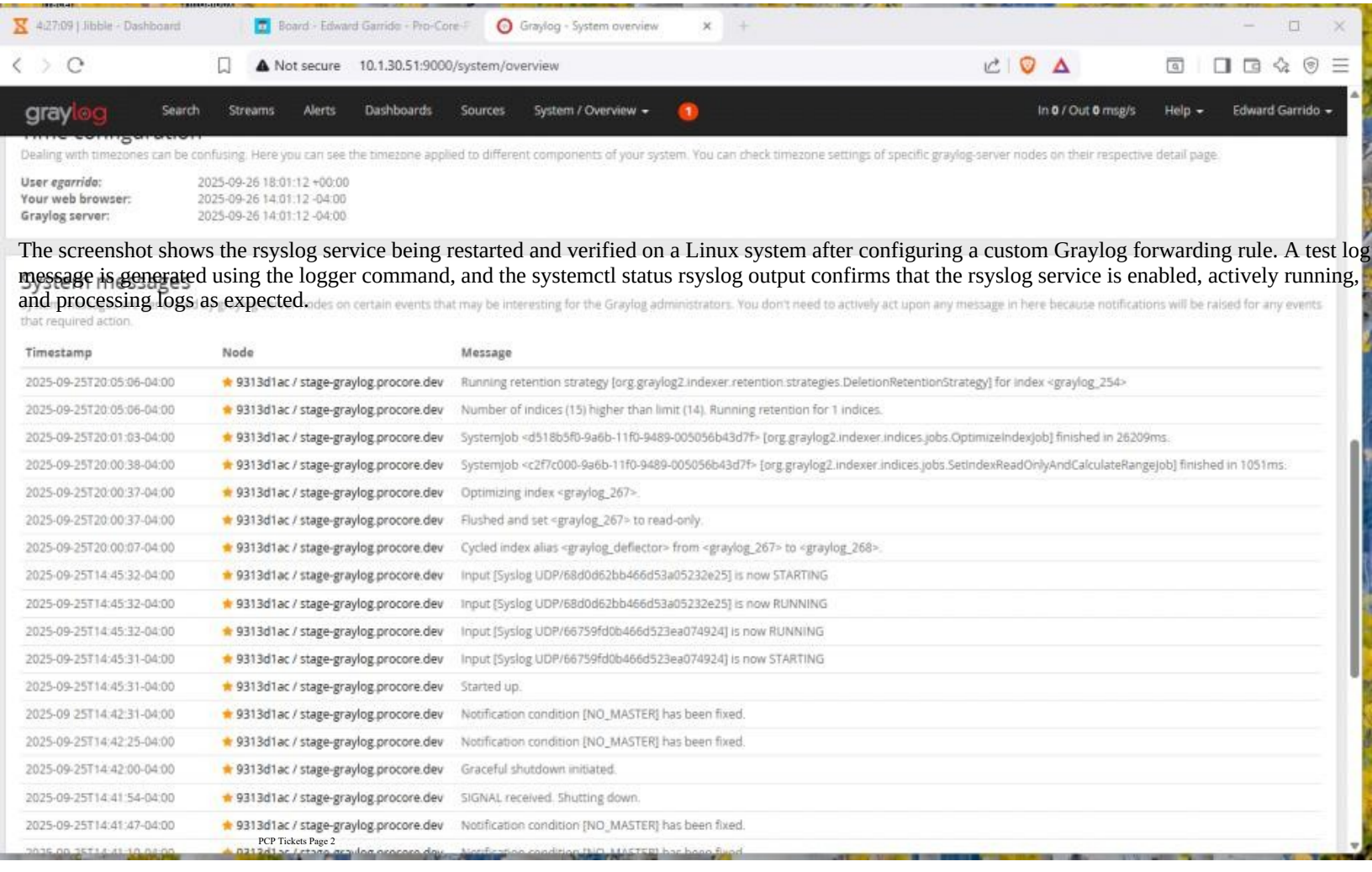
This project documents the configuration of centralized system logging using rsyslog to forward all system logs to a Graylog server over TCP. The setup ensures consistent log aggregation across Linux hosts, enabling improved observability, troubleshooting, and security monitoring. By standardizing the Syslog protocol format and directing logs to a centralized collector, this configuration supports enterprise logging best practices and prepares hosts for scalable monitoring and alerting workflows.



The screenshot shows the rsyslog service being restarted and verified on a Linux system after configuring a custom Graylog forwarding rule. A test log message is generated using the logger command, and the systemctl status rsyslog output confirms that the rsyslog service is enabled, actively running, and processing logs as expected.

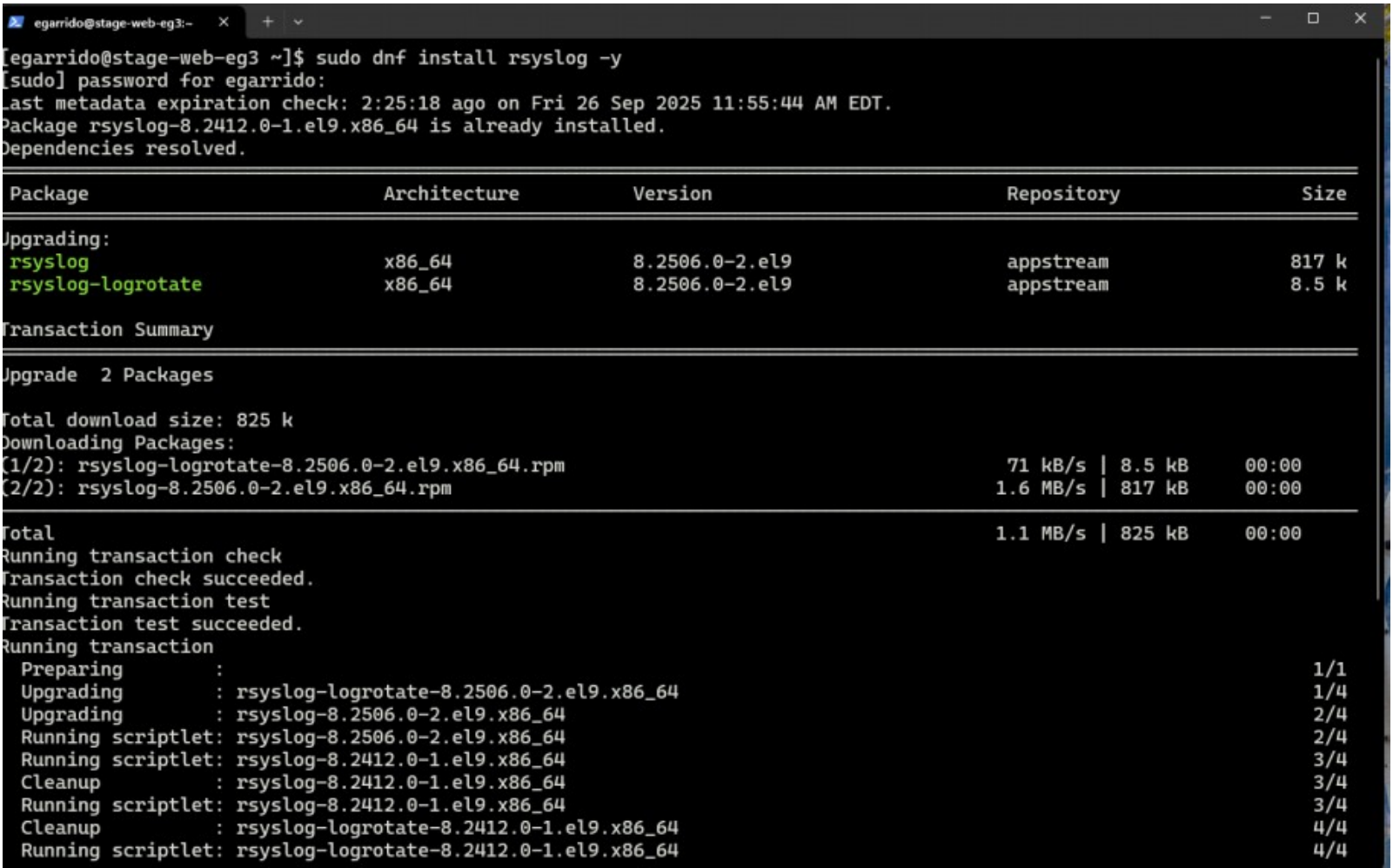


The screenshot shows the Graylog System → Overview page displaying system messages and operational events. It confirms that the Graylog server is running normally, with logs indicating index rotation, retention strategy execution, index optimization, and Syslog UDP inputs transitioning to a RUNNING state. The timestamps and messages demonstrate that log ingestion and backend maintenance tasks are functioning as expected, verifying successful log flow into Graylog after rsyslog configuration and testing

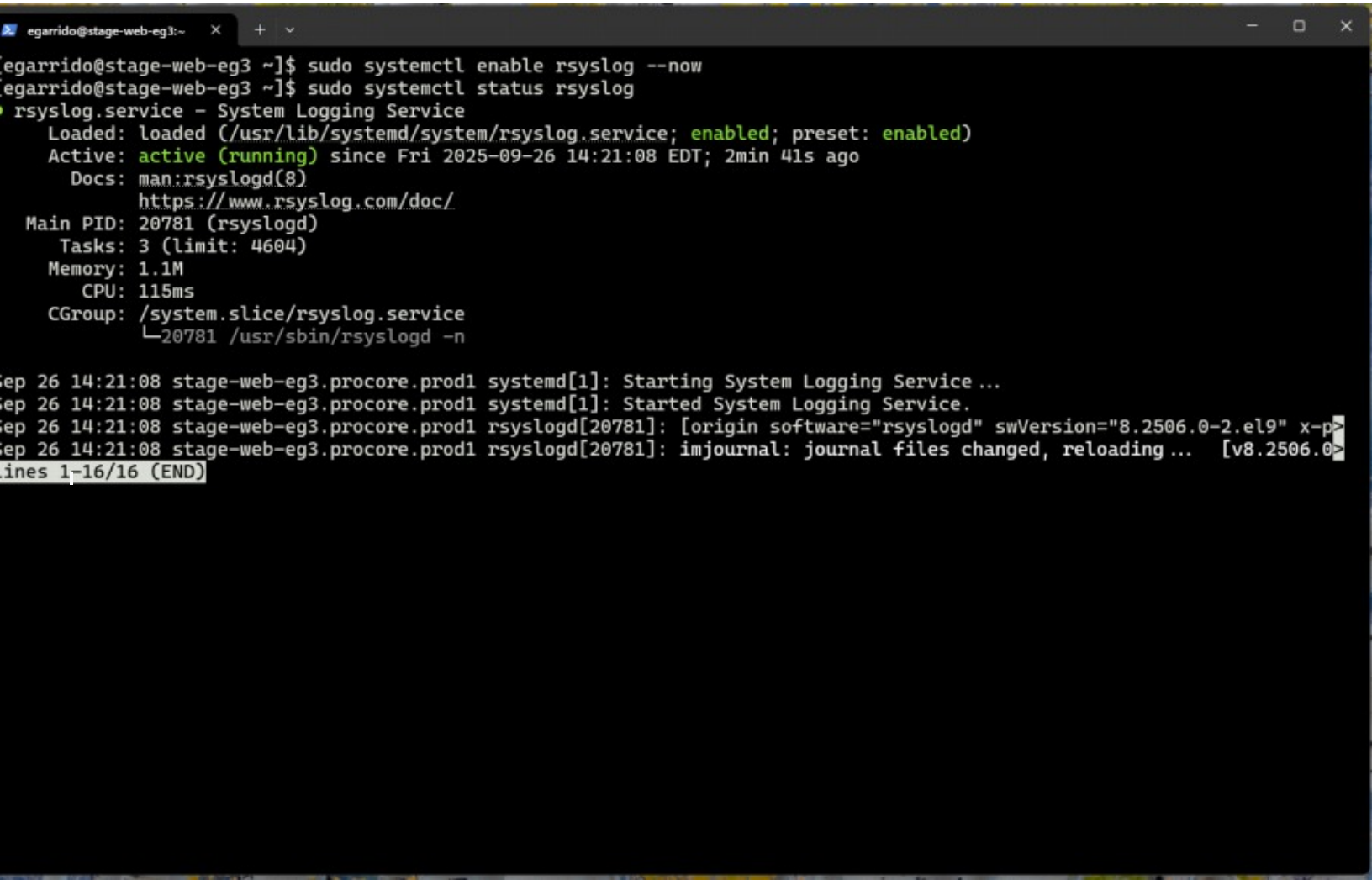


The screenshot shows the rsyslog service being restarted and verified on a Linux system after configuring a custom Graylog forwarding rule. A test log message is generated using the logger command, and the systemctl status rsyslog output confirms that the rsyslog service is enabled, actively running, and processing logs as expected.

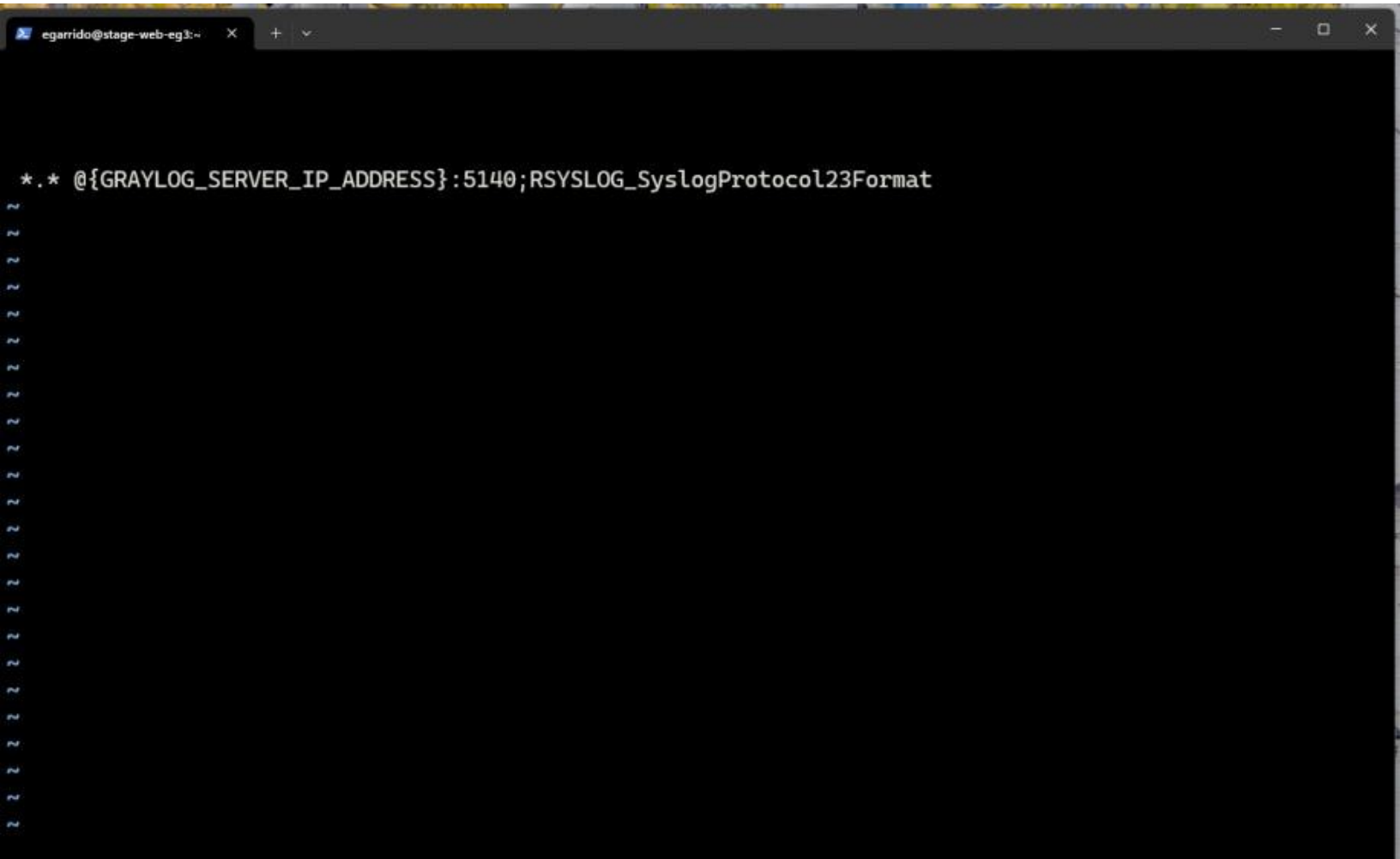
The screenshot shows the dnf install rsyslog -y command being executed on a CentOS Stream 9 system. The output confirms that rsyslog was already installed and successfully upgraded, along with the rsyslog-logrotate package, from the AppStream repository. Dependency checks, transaction tests, and post-install cleanup all completed without errors, verifying that the rsyslog logging service and its log rotation components are properly installed and up to date.



The screenshot shows the rsyslog service being enabled and started using systemctl enable rsyslog --now on a CentOS Stream 9 system. The systemctl status rsyslog output confirms that the System Logging Service is enabled at boot and actively running. Service details, including the main process ID and recent startup log messages, indicate that rsyslog initialized successfully and is ready to process system logs.

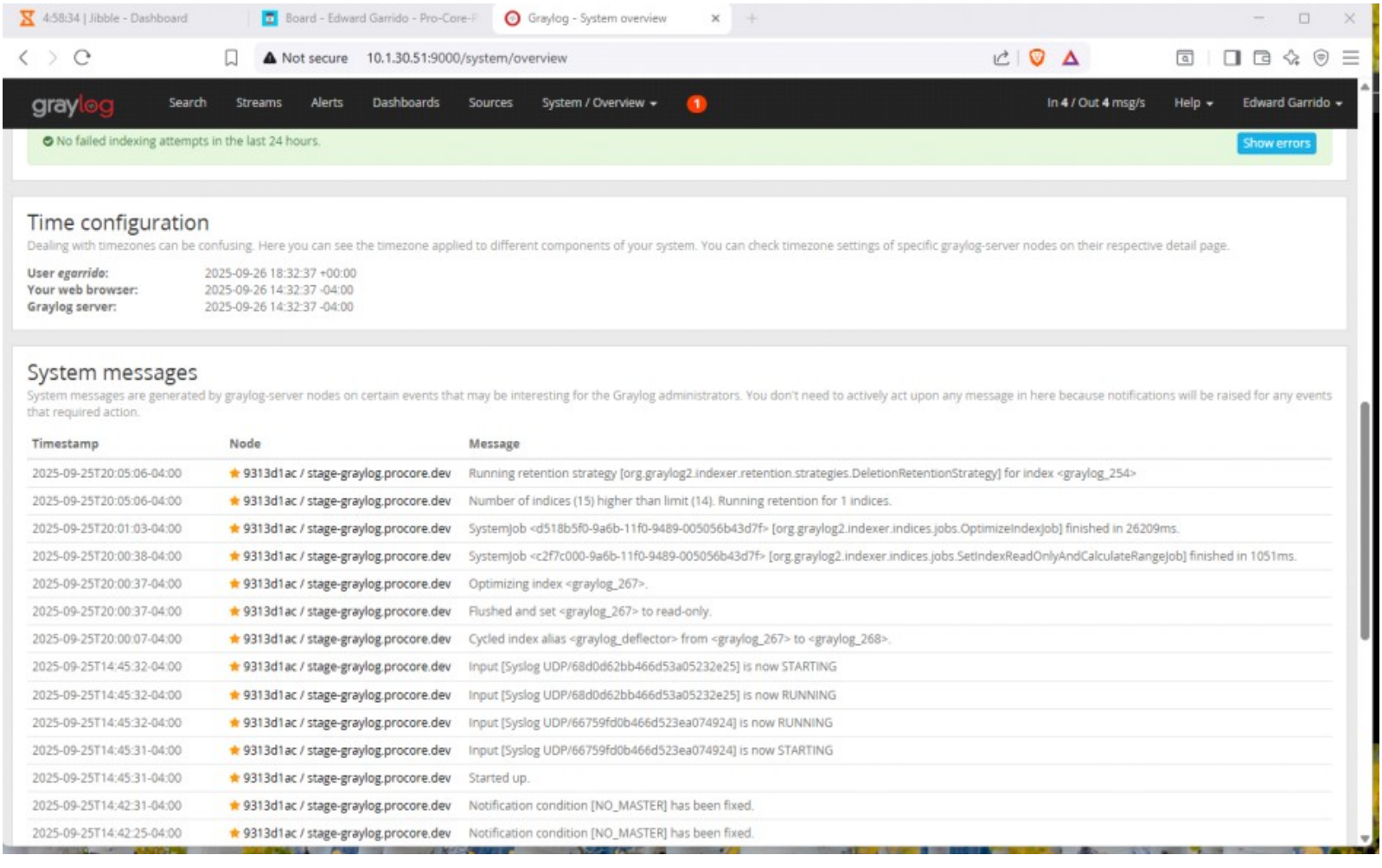


The screenshot shows the /etc/rsyslog.d/90-graylog.conf configuration file open in a text editor. It contains a single rsyslog rule that forwards all system logs (*.*) to a Graylog server over UDP port 5140, using the Syslog Protocol 23 format. The Graylog server address is referenced via a variable (@{GRAYLOG_SERVER_IP_ADDRESS}), indicating a centralized logging setup that standardizes log forwarding while keeping the configuration clean and reusable.

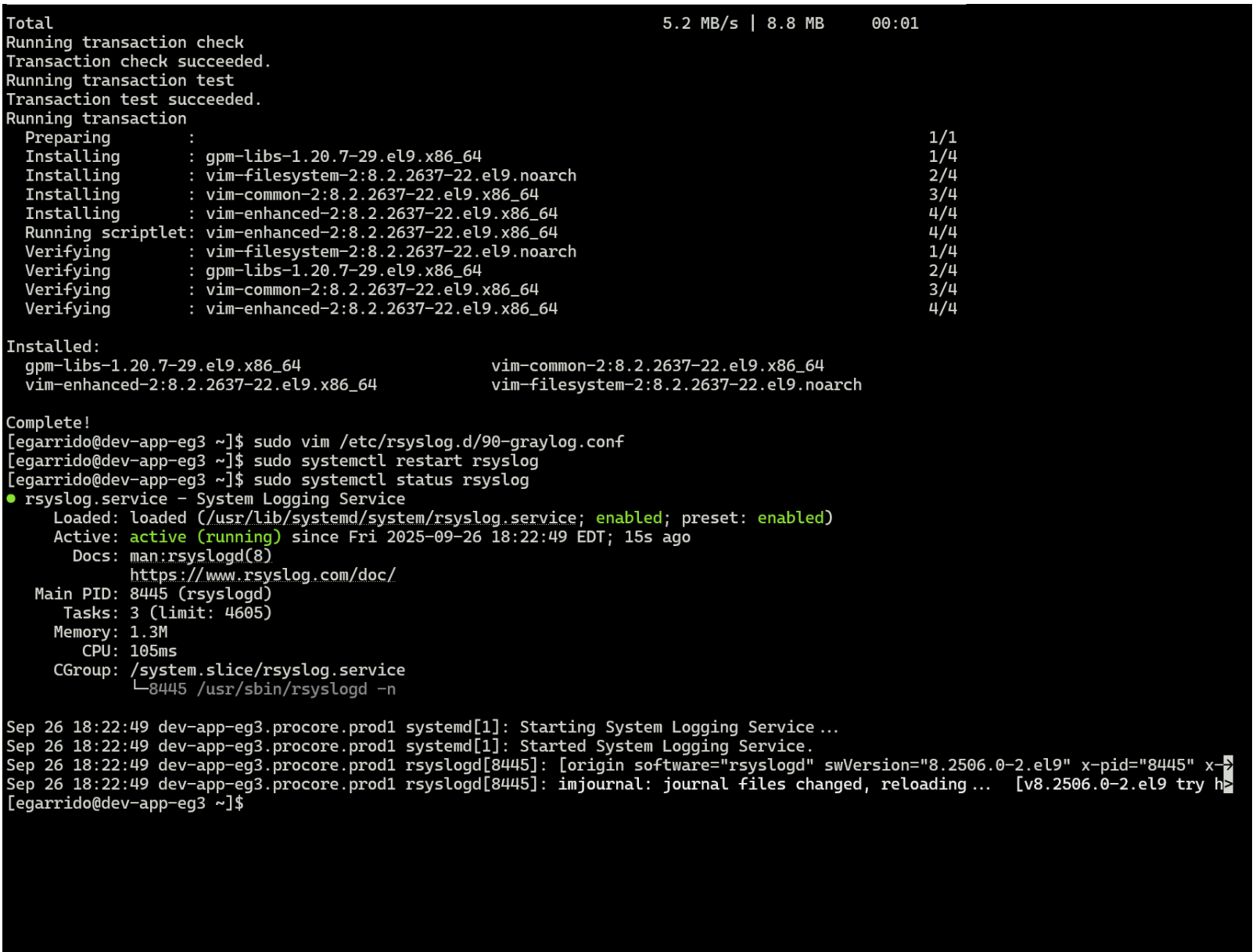


The screenshot displays the Graylog System → Overview page confirming the health and operational status of the logging platform. It shows no failed indexing attempts in the last 24 hours, indicating stable log ingestion. The Time configuration section confirms consistent timestamps across the user session, browser, and Graylog server.

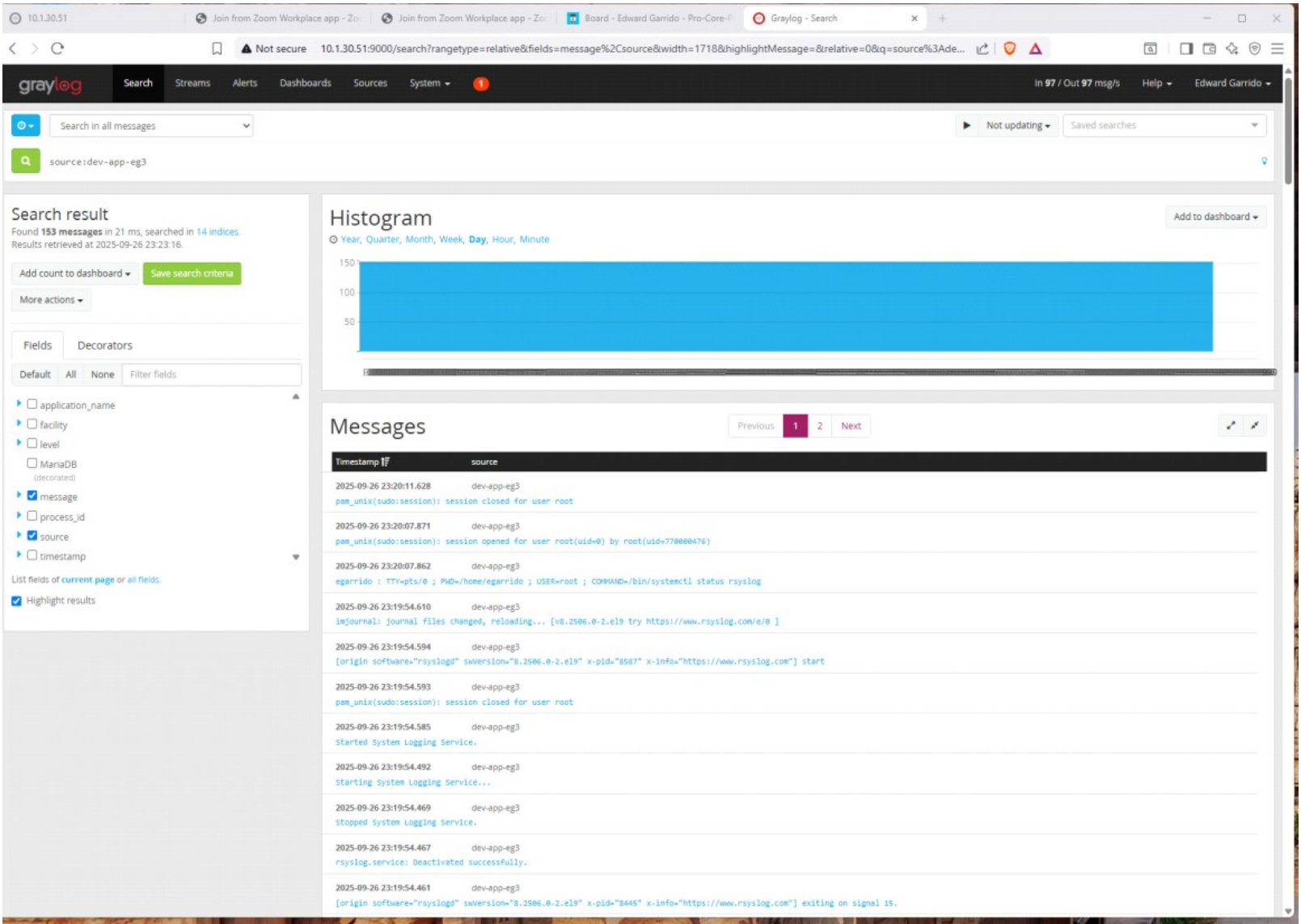
Below, the System messages panel lists normal Graylog activities such as index rotation, retention strategy execution, index optimization, and Syslog UDP inputs starting and running successfully. Together, these details verify that Graylog is functioning correctly and actively receiving and managing logs from connected systems.



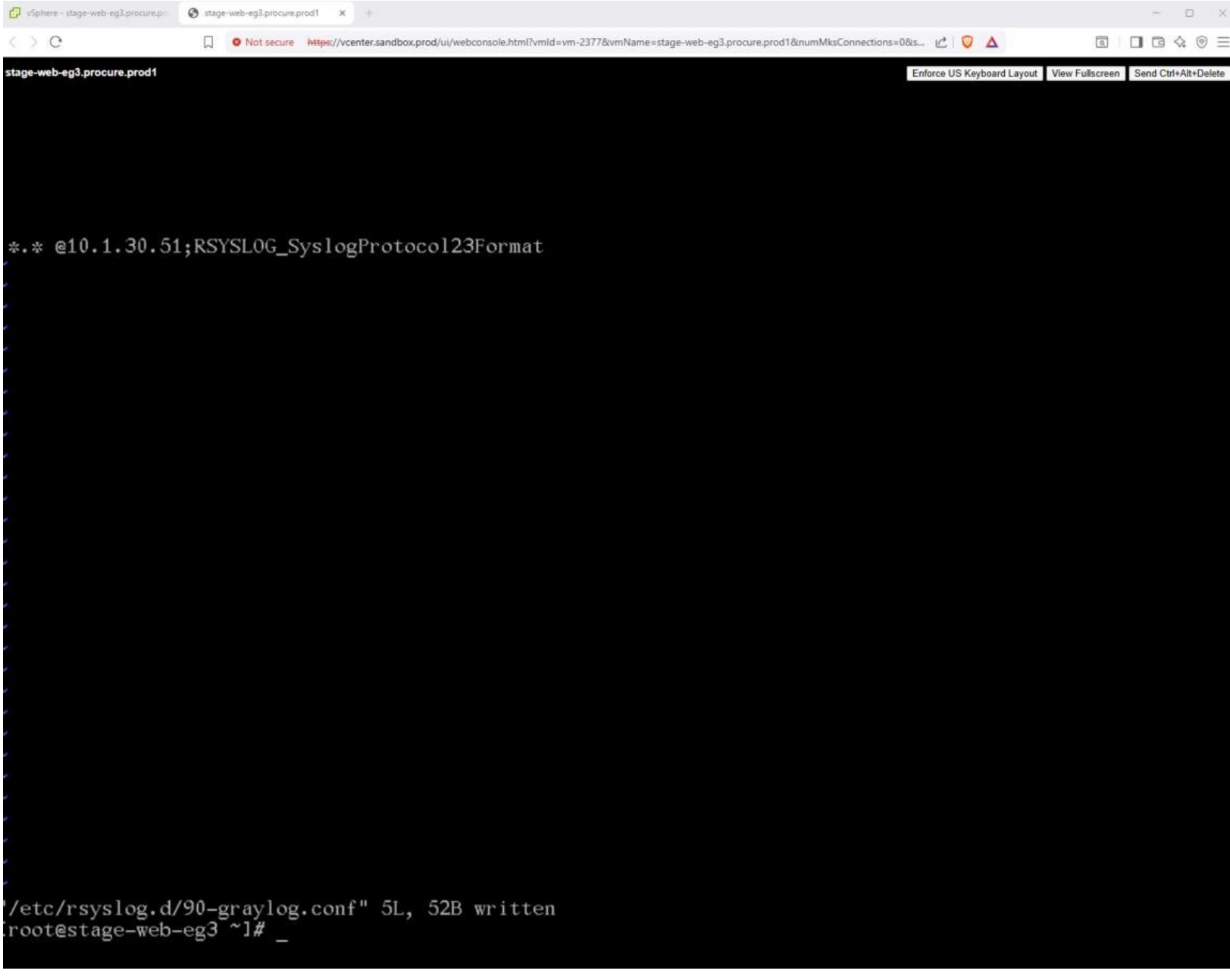
The screenshot shows the installation of required editor packages (including vim-enhanced) followed by configuration and validation of rsyslog on a CentOS Stream 9 system. After editing the Graylog configuration file (/etc/rsyslog.d/90-graylog.conf), the rsyslog service is restarted and its status is checked. The output confirms that rsyslog is enabled, actively running, and restarted successfully, with system logs indicating proper initialization and readiness to forward logs to the centralized Graylog server.



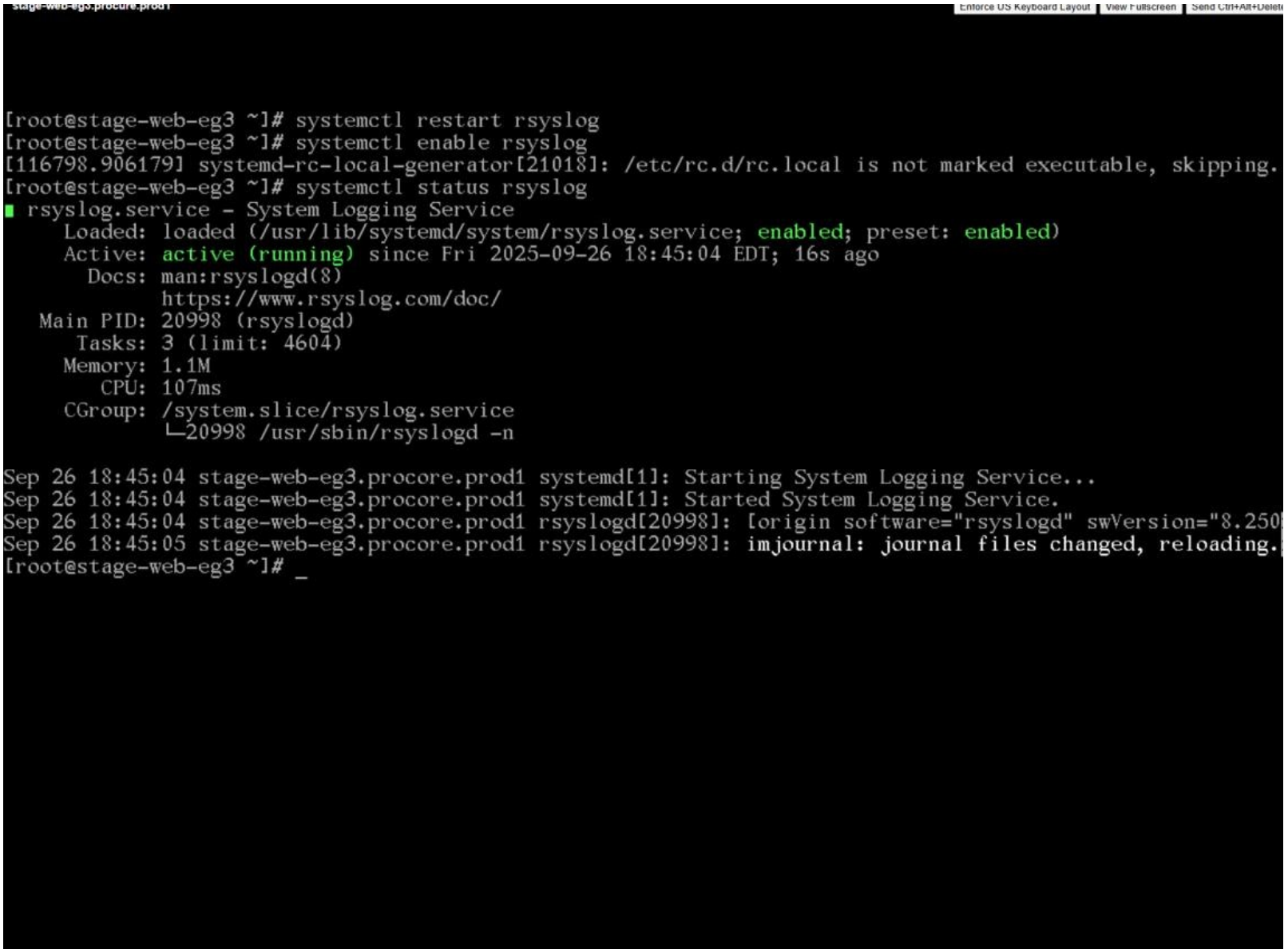
The screenshot shows the Graylog Search interface confirming successful log ingestion from the host dev-app-eg3. A filtered search (source:dev-app-eg3) returns multiple messages across recent timestamps, visualized in the histogram at the top and detailed in the message list below.



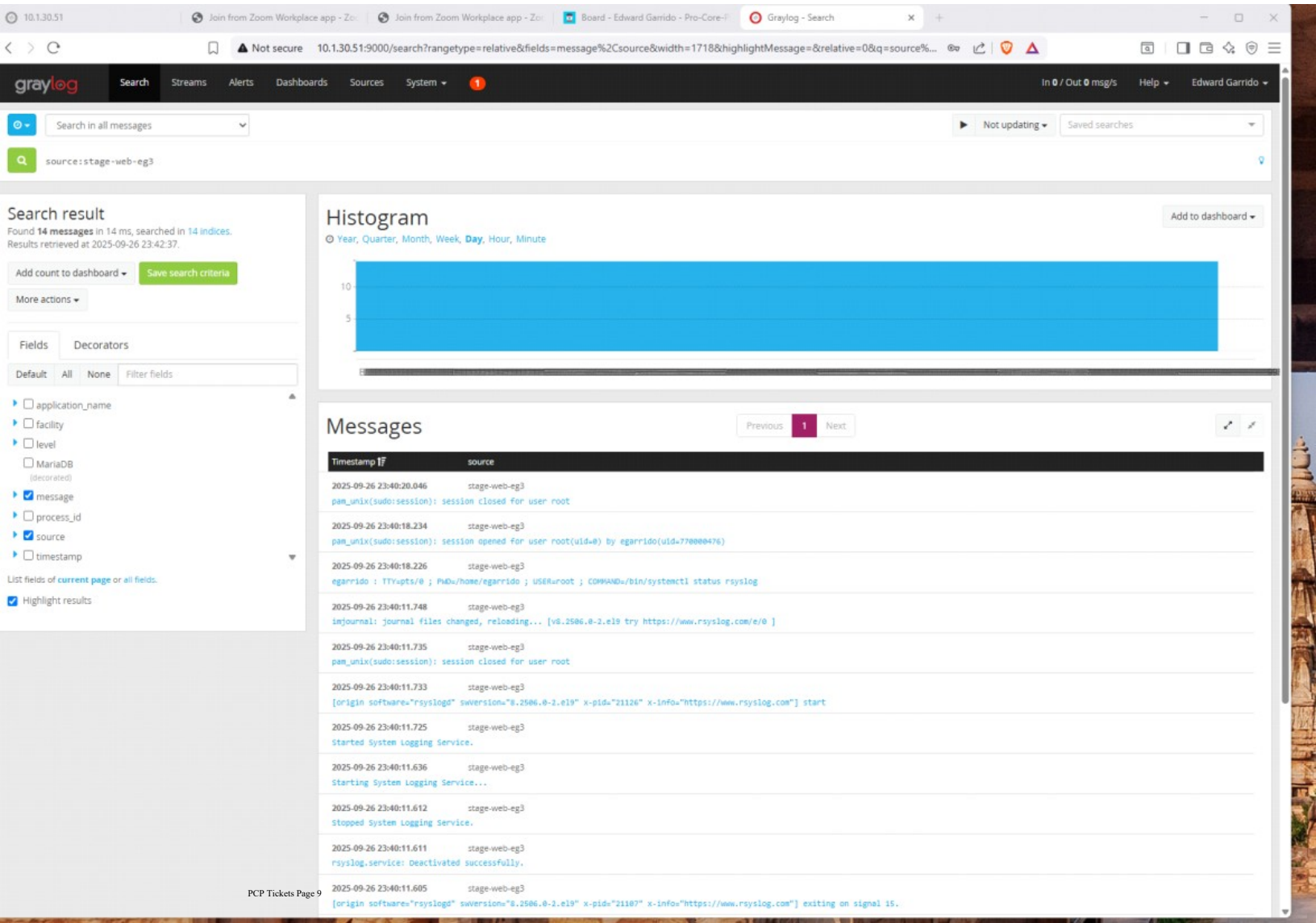
The screenshot shows the /etc/rsyslog.d/90-graylog.conf file open on the stage-web-eg3 server after being edited and saved. The configuration contains a rule that forwards all system logs (*.*) to a centralized Graylog server using the RSYSLOG Syslog Protocol 23 format. The file save confirmation at the bottom indicates the configuration was successfully written, preparing the system for centralized log forwarding via rsyslog.



This screenshot verifies that the rsyslog service on the stage-web-eg3 server was successfully restarted, enabled at boot, and is actively running. The output confirms the service is loaded, enabled, and operational, with recent log entries showing the system logging service starting and reloading journal files—indicating the updated logging configuration has been applied correctly.



This screenshot confirms that logs from the stage-web-eg3 host are successfully being ingested and indexed in Graylog. The search filter is scoped to source:stage-web-eg3, and the results show recent system and rsyslog events, including service restarts, journal reloads, and sudo session activity. The populated histogram and message list verify end-to-end log forwarding from the server to Graylog and successful processing by the logging pipeline.



Summary

Centralized logging was implemented using rsyslog to forward system logs from multiple CentOS Stream 9 servers to a Graylog server. Logging services were installed, enabled, and validated on each host, with log forwarding configured through a dedicated rsyslog configuration file. End-to-end functionality was confirmed by generating test events and verifying successful ingestion, indexing, and searchability within the Graylog interface. All environment details and IP addresses shown are sanitized for security