INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 06, Issue 01, January 2026, pp : 207-214
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

CENTRALIZED LOG ANALYSIS AND MONITORING OF SYSTEM, WEB APPLICATION AND CONTAINER LOGS USING ELK STACK

Ritvik S[1], Surya J[2]

[1,2]UG Student, Department of IT and Cognitive Systems, Sri Krishna Arts And Science College, Coimbatore, Tamil Nadu, India.

ABSTRACT

With the rapid growth of web applications, containerized deployments, and distributed systems, managing and analyzing logs from multiple sources has become increasingly complex. Traditional log analysis methods are often decentralized, manual, and inefficient, leading to delayed detection of system issues and security threats. This paper proposes a centralized log analysis and monitoring solution using the ELK Stack (Elasticsearch, Logstash, and Kibana) to efficiently collect, process, store, and visualize logs in real time. The system is implemented on the Ubuntu operating system and integrates logs from the Nginx web server, system services, and Docker containers. Logstash is used to parse and structure heterogeneous log data, Elasticsearch enables scalable storage and fast querying, and Kibana provides interactive dashboards for monitoring and analysis. The proposed solution improves system visibility, enables real-time threat detection, reduces incident response time, and enhances overall system reliability and cybersecurity monitoring in modern IT environments.

Keywords: ELK Stack, Log Analysis, Cybersecurity Monitoring, Elasticsearch, Logstash, Kibana, Web Application Security, Container Logs, System Logs, Threat Detection.

1. INTRODUCTION

The complexity of contemporary IT infrastructures has dramatically increased due to the quick development of web applications, container-based deployments, and distributed system architectures. To increase scalability and deployment efficiency, organisations are depending more and more on technologies like cloud platforms, microservices, and containerisation. Large amounts of logs are thus produced by operating systems, web servers, apps, and containers. These logs are crucial for efficient monitoring and management because they include vital information about system behaviour, performance metrics, user activity, and security events.

Logs are essential for tracking system health, identifying security risks, and diagnosing malfunctions. They offer thorough records of things like system modifications, user requests, authentication attempts, and application errors. Administrators can detect cyberattacks, spot unusual activity, and guarantee system dependability by promptly analysing log data. Centralised and automated log monitoring has become essential for preserving operational visibility and cybersecurity as infrastructures grow more dispersed.

Despite the significance of logs, there are a number of difficulties with conventional log analysis techniques. Traditional approaches frequently entail manually reviewing log files and storing logs locally on separate servers, which is laborious and ineffective. These methods struggle to keep up with the growing volume and speed of log data produced by contemporary systems. Critical events might therefore go unreported, delaying incident response and possibly resulting in security breaches.

The dispersed and unstructured nature of log data presents another significant obstacle. Correlation and analysis are challenging because different operating systems, web servers, and container platforms generate logs in different formats. The lack of centralized storage and real-time analysis further complicates troubleshooting and security monitoring, highlighting the need for an efficient, scalable, and unified log management solution.

This project's main goal is to use the ELK Stack to design and implement a centralised log analysis and monitoring system. The system's goal is to gather and combine logs from various sources—such as the Nginx web server, Docker containers, and the Ubuntu operating system—into a single, cohesive platform. This centralisation enhances overall system visibility and streamlines log management.

Enabling real-time log data monitoring and analysis is another important goal. The system enables administrators to promptly detect abnormal behaviour, system failures, and performance problems by processing logs as they are generated. By using log analysis and visualisation to identify suspicious activity and possible cyberthreats, the project also aims to improve security monitoring. Additionally, by offering structured logs, robust search capabilities, and interactive dashboards, the system seeks to increase troubleshooting efficiency. By achieving these goals, the suggested solution aims to improve operational effectiveness, shorten incident response times, and fortify the general security posture of contemporary IT infrastructures.

2. REVIEW OF LITERATURE

Bavaskar et al. (2019): Performed a comprehensive survey on log analysis techniques and highlighted the key capabilities of the ELK Stack as a leading open-source solution for collecting, parsing, storing, and visualizing log data. They emphasized ELK's potential in handling high-velocity logs from multiple sources and enabling interactive analysis dashboards. Their work laid a foundational understanding of how Elastic technologies can replace manual and semi-automated logging systems in IT environments.

Swati Chaudhari et al. (2020): Proposed a real-time traffic and log monitoring framework leveraging ELK-based pipelines to support proactive incident detection across networked systems. Their research demonstrated that integrating centralized log processing with interactive dashboards enables administrators to detect abnormal access patterns and potential security breaches quicker than traditional tools.

Anishkumar Sargunakumar (2020): Explored Logstash and Kibana for full-stack monitoring. The study showed that structuring log data into fields such as timestamps, error types, and source identifiers significantly improves troubleshooting and enhances the accuracy of performance analytics. This work also discussed best practices for building scalable Logstash pipelines capable of handling heterogeneous log formats, a challenge common in distributed systems.

Koneri (2025): Investigated centralized logging solutions in cloud environments, focusing on AWS-integrated ELK deployments. The research demonstrated that a centralized platform increases security visibility and compliance reporting capability across hybrid cloud environments. It also highlighted the need for scalable indexing strategies to manage rapidly growing log volumes.

Mihail Alexandru Stan (2021): Automated log analysis using the ELK Stack coupled with machine learning techniques for network vulnerability detection. The study demonstrated that combining ELK with unsupervised anomaly detection algorithms enhances the precision of identifying unusual events and reduces false positive alerts.

Amarasinghe (2023): Evaluated ELK-based log analytics for small and medium enterprises, emphasizing its feasibility as a cost-effective monitoring solution. The research highlighted improvements in network security visibility and efficient storage of large log datasets.

Maduranga (2021): Reinforced similar findings with a focus on ELK as an affordable SIEM-like system. This study provided compelling evidence that open-source stacks can rival commercial solutions in delivering centralized security monitoring when configured with effective parsing and alert capabilities.

3. SYSTEM ARCHITECTURE

The system architecture defines the structural design and operational workflow of the proposed centralized log analysis and monitoring solution. It is designed to support real-time log ingestion, scalable storage, efficient querying, and interactive visualization of logs generated from system-level, application-level, and containerized environments. The architecture emphasizes modularity, fault tolerance, and extensibility, enabling seamless integration of additional log sources and analytical components in the future.

3.1 Overall Architecture Description

The proposed architecture has adopted a centralized logging approach whereby logs are collected from distributed systems and then processed on a single platform. The proposed architecture employs the use of Ubuntu as its operational system and ELK Stack with Nginx and Docker applications for log collection and processing. The proposed architecture comprises five layers: log generation, log collection, log processing, log storage, and lastly visualization.

At the log generation layer, different components continuously generate logs of system events, application activities, user requests, and runtime behavior. The input plugins configured will forward them to the collection layer. The processing layer then uses Logstash for normalizing and structuring heterogeneous log formats from different logging sources. The processed data goes to Elasticsearch, a distributed search and analytics engine that can handle massive volumes of log data. Kibana acts as the visualization layer, providing real-time dashboards, filters, and alerts for monitoring and cybersecurity analytics.

The architecture allows for centralized visibility over the entire infrastructure, complexity reduction in operations, as well as swift detection of anomalies as well as incidents. Through the aggregation of log data from varied layers into one platform, administrators can have a single point of perspective regarding the operations of the systems as well as the performance of the application. The architecture also promotes proactive monitoring through the capability to perform real-time analysis with the generation of alarms, hence reducing downtime in the systems. In addition, the

architecture is very scalable. It also facilitates efficient troubleshooting and informed decision-making by providing accurate, real-time insights through centralized log analysis.

3.2 Log Sources

Ubuntu System Logs:

Ubuntu system logs capture critical information related to operating system events, user authentication, background services, and hardware interactions. Logs such as syslog, auth.log, and kernel logs provide valuable insights into system stability, resource utilization, and potential security violations. Centralizing these logs allows administrators to detect unauthorized access attempts, system failures, and abnormal system behavior at an early stage.

Nginx Web Server Logs:

Nginx plays a crucial role in hosting and managing web traffic. Its access logs record detailed information about client requests, including IP addresses, request methods, URLs, response codes, and response times. Error logs capture application failures, configuration issues, and backend connectivity problems. Analyzing Nginx logs helps in identifying performance bottlenecks, detecting malicious requests such as brute-force attacks, and ensuring the availability and reliability of web services.

Docker Container Logs:

Docker enables application deployment using lightweight and isolated containers. Each container generates logs related to application execution, exceptions, and runtime events. Collecting Docker logs provides visibility into containerized applications and microservices. Centralized analysis of container logs helps identify application crashes, resource contention issues, and security vulnerabilities within containerized environments.

3.3 Data Flow Diagram

The process begins with log generation from Ubuntu system services, Nginx web applications, and Docker containers. These logs are collected through configured Logstash input plugins or log shippers that continuously monitor log files and streams. In the processing stage, Logstash applies filters to parse unstructured log data, extract relevant fields, normalize timestamps, and enrich logs with metadata such as source type and host information. This structured data is then forwarded to Elasticsearch, where it is indexed and stored in a distributed manner to support fast searches and scalable analytics.

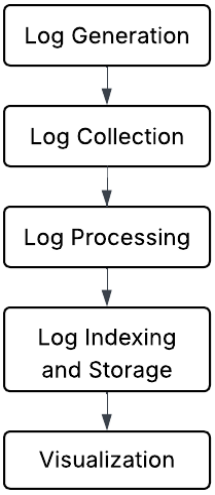

Figure 1: Data Flow Diagram of Centralized log collection system

4. TECHNICAL STACK

4.1 Operating System – Ubuntu:

Ubuntu is used as the primary operating system for deploying and managing the entire infrastructure. It provides a stable, secure, and widely supported Linux environment suitable for hosting ELK Stack components, Nginx web server, and Docker containers. Ubuntu offers efficient resource management, strong community support, and seamless compatibility with open-source monitoring tools. System-level logs generated by Ubuntu, such as authentication logs and service logs, form a critical input for system health monitoring and security analysis.

4.2 Web Server – Nginx:

Nginx is employed as the web server to host web applications and manage incoming client requests. It is chosen for its high performance, low resource consumption, and ability to handle concurrent connections efficiently. Nginx

generates detailed access and error logs that provide valuable insights into web traffic patterns, request behavior, response status codes, and server-side errors. These logs play a key role in identifying performance issues, detecting suspicious activities, and monitoring application availability.

4.3 Container Platform – Docker

Docker is used to deploy and manage applications in isolated containers. Containerization ensures consistency across development and deployment environments while enabling efficient resource utilization. Docker generates logs related to container lifecycle events, application output, and runtime errors. Centralizing Docker logs allows better visibility into containerized workloads, simplifies troubleshooting in microservices architectures, and supports security monitoring in dynamic container environments.

4.4 Log Shipper – Filebeat

Filebeat is used as a lightweight log shipper to collect log files from multiple sources and forward them to the processing layer. It continuously monitors specified log files such as Ubuntu system logs, Nginx access and error logs, and Docker container logs. Filebeat is designed for minimal resource usage and reliable log delivery, ensuring that log data is transmitted efficiently and without loss, even under high-load conditions.

4.5 Log Processor – Logstash

Logstash serves as the core log processing engine in the system. It receives logs from Filebeat and applies filters to parse, structure, and normalize raw log data. Logstash enables extraction of key fields such as timestamps, IP addresses, error messages, and log levels. This processing step transforms unstructured logs into structured formats suitable for efficient indexing and analysis, improving the accuracy and usefulness of log insights.

4.6 Storage and Search – Elasticsearch

Elasticsearch is used as the centralized storage and search engine for processed log data. It provides distributed indexing, fast querying, and high scalability, making it suitable for handling large volumes of log data. Elasticsearch enables real-time search and aggregation across logs from multiple sources, supporting efficient troubleshooting, trend analysis, and security investigations. Its scalable architecture ensures consistent performance as log volume grows.

4.7 Visualization Tool – Kibana

Kibana is used as the visualization and monitoring interface for the system. It connects to Elasticsearch to retrieve indexed log data and presents it through interactive dashboards, charts, and graphs. Kibana enables users to analyze log patterns, monitor system health, detect anomalies, and gain real-time insights into system and application behavior. Custom dashboards and filters enhance usability and support effective decision-making for administrators and security teams.

5. METHODOLOGY

The methodology for the proposed project **"Centralized Log Analysis and Monitoring of System, Web Application, and Container Logs using ELK Stack"** is designed to systematically collect, process, store, analyze, and visualize logs from heterogeneous sources. This approach ensures centralized monitoring, real-time insights, improved security, and efficient troubleshooting for modern distributed systems. The methodology is divided into several key phases: log generation, log collection, log processing and parsing, indexing and storage, and visualization and monitoring. Each phase is critical for achieving accurate and actionable insights from logs while maintaining system performance and scalability.

5.1 Log Generation:

Log generation forms the first phase of the methodology. Logs are continuously produced by multiple components of the infrastructure, including Ubuntu system services, the Nginx web server, and Docker containers. System logs from Ubuntu provide details about authentication events, background service operations, kernel messages, and system errors, which are critical for monitoring the health and security of the operating system. Nginx generates access and error logs that capture web traffic, HTTP requests, client IPs, server responses, and potential application errors. Docker containers produce logs that record runtime events, application outputs, and container lifecycle information. Collectively, these logs form the raw data foundation for centralized analysis, allowing administrators to detect errors, performance issues, and security anomalies across all system layers.

5.2 Log Collection:

In the log collection phase, Filebeat is employed as a lightweight and reliable log shipper. Filebeat monitors designated log files and directories in real time, capturing updates as they occur. It collects logs from Ubuntu system directories, Nginx server logs, and Docker container outputs, ensuring continuous and uninterrupted log delivery to the processing layer. Its minimal resource usage ensures that log collection does not interfere with system performance.

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 06, Issue 01, January 2026, pp : 207-214
e-ISSN : 2583-1062
Impact Factor : 7.001

Filebeat also supports secure communication, guaranteeing that sensitive log data is transmitted reliably to Logstash for processing.

5.3 Log Processing and Parsing:

The log processing and parsing phase is handled by Logstash, which acts as the core engine for transforming raw, unstructured log data into structured and analyzable formats. Logstash pipelines use grok patterns, filters, and mutation operations to extract meaningful fields such as timestamps, IP addresses, request methods, error messages, and container identifiers. During this phase, logs are also enriched with metadata, including source type, hostname, and environment tags, and normalized to maintain consistent formatting across heterogeneous sources. This structured processing ensures accurate indexing, enables efficient querying, and improves the precision of subsequent analysis and visualization.

5.4 Log Indexing and Storage:

After processing, logs are sent to Elasticsearch for indexing and storage. Elasticsearch provides a scalable, distributed, and fault-tolerant storage solution capable of handling large volumes of log data. Logs are indexed by source type, time, and relevant fields, making it possible to retrieve information quickly for real-time analysis. The storage layer supports advanced querying, aggregation, and search operations, allowing administrators to perform historical analysis, trend detection, and forensic investigations. Elasticsearch's clustering capabilities ensure high availability and horizontal scalability, maintaining consistent performance even as log volume grows.

5.5 Visualization and Alerting:

Finally, visualization and monitoring are achieved using Kibana. Kibana dashboards provide real-time insights into system performance, web traffic, and container behavior. Interactive visualizations, charts, and graphs allow administrators to monitor error rates, system resource usage, web requests, and abnormal activities. Custom dashboards and alert configurations enable proactive detection of anomalies, helping in timely troubleshooting and security incident response. Kibana also supports reporting features, allowing administrators to generate summaries of log activity for operational audits and compliance purposes.

Overall, the methodology ensures a seamless and efficient flow of log data from generation to visualization, with a focus on **real-time monitoring, centralized visibility, and enhanced security**. By integrating Ubuntu, Nginx, Docker, Filebeat, Logstash, Elasticsearch, and Kibana, the system achieves **comprehensive log analysis, reduced troubleshooting time, and improved operational efficiency**. This modular and scalable methodology also allows future enhancements, such as the integration of machine learning-based anomaly detection, support for additional log sources, and cloud-native deployment, making it adaptable to evolving IT infrastructures.

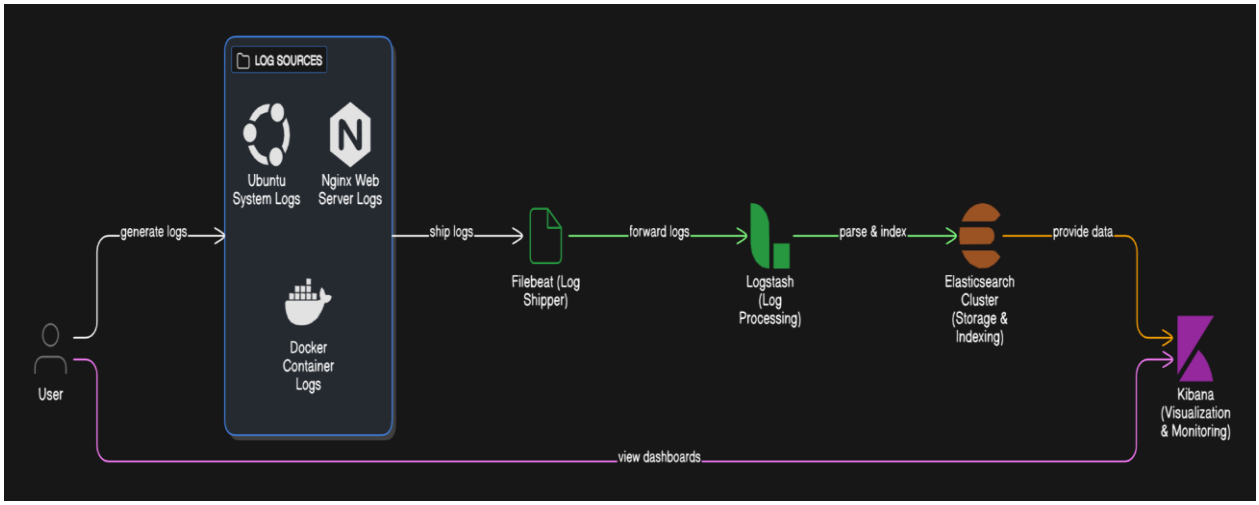

Figure 2: Methodology of how the system behaves

6. CHALLENGES AND GOVERNANCE

6.1 Key Technical and Operational Challenges:

Implementing a centralized log analysis and monitoring system using the ELK Stack presents several technical and operational challenges. One primary challenge is handling high-volume log data. Large-scale environments with multiple servers, web applications, and containerized workloads generate massive amounts of logs, which can lead to storage and performance bottlenecks. Ensuring efficient log ingestion, processing, and indexing without impacting system performance requires careful configuration and resource management.

Another significant challenge is log heterogeneity. Logs from Ubuntu system services, Nginx web servers, and Docker containers differ in structure and format, making parsing and normalization difficult. Developing robust Logstash pipelines to extract meaningful fields and maintain consistent log formats is crucial for accurate analysis and visualization.

Security and compliance also pose challenges. Logs often contain sensitive information such as user credentials, IP addresses, and internal system data. Protecting log data during collection, transmission, storage, and visualization is essential to prevent unauthorized access, data breaches, or tampering. Additionally, organizations may need to comply with regulatory requirements such as GDPR, ISO, or other standards, which requires implementing proper retention policies, access control, and audit trails.

Maintaining real-time monitoring and alerting in dynamic environments, especially with containerized workloads, is another challenge. Containers can scale up or down rapidly, generating logs at high velocity, which can create delays if the pipeline is not optimized. Finally, scalability and resource management remain ongoing challenges. Elasticsearch clusters, Logstash pipelines, and visualization dashboards must be carefully scaled to handle increasing log volumes while maintaining system performance and reliability.

6.2 System Governance and Compliance Measures:

Robust governance is essential to ensure that the centralized log analysis system operates securely, efficiently, and in compliance with organizational policies. Data governance includes defining clear policies for log collection, storage, retention, and deletion, ensuring that only relevant logs are retained and sensitive data is protected. Role-based access control (RBAC) in Elasticsearch and Kibana ensures that only authorized personnel can access or modify log data, dashboards, or configurations.

Operational governance involves maintaining monitoring processes, updating log shippers and pipelines, and regularly auditing system performance. This ensures the system remains reliable, efficient, and capable of handling large-scale logs. **Security governance** protects log data across its lifecycle through encryption, secure storage, authentication mechanisms, and access audits. Compliance governance ensures that the system meets organizational and regulatory standards, including data privacy, retention, and reporting requirements.

Scalability governance is also critical for future growth. Policies for adding new log sources, expanding Elasticsearch clusters, and managing resource allocation help maintain consistent performance. Documentation, incident response procedures, and administrator training further strengthen governance, ensuring that the system remains **reliable, secure, and adaptable** as organizational needs evolve.These governance measures ensure that the centralized log analysis system remains **robust, secure, and compliant** with organizational policies and regulatory standards. They also provide a framework for **scalable growth and operational consistency** as log volumes and infrastructure complexity increase. Overall, effective governance guarantees **reliable performance, data integrity, and actionable insights** for administrators and security teams.

7. FUTURE ENHANCEMENTS

The first major area for future enhancement is the integration of **machine learning-based anomaly detection** into the centralized log analysis system. While the current ELK Stack setup provides real-time log collection and visualization, it primarily relies on predefined filters and manual observation to identify anomalies. By leveraging machine learning algorithms such as clustering, classification, or deep learning-based sequence models, the system can automatically detect unusual patterns or behaviors in system, web, and container logs. For example, sudden spikes in failed login attempts, unusual error patterns in Nginx logs, or abnormal resource usage in Docker containers could be identified proactively. This intelligent analysis reduces the dependency on human intervention, improves detection accuracy, and enables predictive monitoring, allowing administrators to address issues before they escalate into critical incidents.

Another enhancement involves the implementation of a **robust alerting and notification system**. Currently, administrators need to actively monitor dashboards in Kibana to identify critical events. By introducing automated alert mechanisms, the system can instantly notify administrators whenever predefined thresholds or unusual activities are detected. Alerts could be sent via multiple channels, including email, SMS, instant messaging, or integration with incident management tools. This would enable faster response to security threats, performance bottlenecks, or operational failures. Additionally, an advanced notification system could support **prioritization and categorization** of alerts, ensuring that critical issues receive immediate attention while minor events are logged for later analysis.

A further enhancement is the integration of **cloud environments** into the ELK-based monitoring system. As organizations increasingly adopt cloud infrastructure for scalability and cost efficiency, the centralized log analysis platform can be extended to collect logs from cloud servers, virtual machines, and cloud-based applications. This would allow administrators to monitor hybrid or multi-cloud architectures in real time, providing centralized visibility

across on-premise and cloud resources. Additionally, cloud integration can leverage auto-scaling, load balancing, and distributed storage, ensuring that the monitoring system remains resilient and efficient even under high log volume scenarios.

Closely related to cloud integration is **Kubernetes support** for containerized workloads. Modern applications are often deployed as microservices using Kubernetes clusters, where containers can dynamically scale, move between nodes, or be created and destroyed frequently. Integrating Kubernetes logging capabilities into the ELK Stack would enable the collection of pod, node, and cluster-level logs, providing full observability of containerized environments. This would facilitate real-time monitoring of application health, error detection, performance tracking, and security monitoring at the microservices level. Features such as centralized pod logging, automated log aggregation from multiple namespaces, and correlation of logs across services would make the system highly effective for cloud-native and containerized workloads.

Finally, combining these enhancements—**machine learning, alerting, cloud, and Kubernetes integration**—would transform the centralized log analysis system into a **proactive, intelligent, and scalable monitoring platform**. Such a platform would not only provide real-time visualization and analysis but also predictive insights, automated incident response, and seamless scalability for modern IT infrastructures. Organizations would be able to achieve enhanced operational efficiency, improved system reliability, and stronger security posture. Furthermore, these enhancements would allow the system to evolve with emerging technologies and complex distributed architectures, ensuring long-term adaptability and relevance in rapidly changing IT environments.

8. CONCLUSION

In this project, a centralized log analysis and monitoring system was designed and implemented using the ELK Stack to manage logs from Ubuntu system services, Nginx web servers, and Docker containers. The system efficiently collects, processes, indexes, and visualizes logs, providing real-time monitoring and centralized visibility across all components of the infrastructure. By integrating Filebeat for log collection, Logstash for parsing and processing, Elasticsearch for storage and indexing, and Kibana for visualization, the platform ensures that logs are handled reliably and presented in a structured and meaningful way.

The implementation of this system demonstrates significant improvements in operational efficiency and security monitoring. Administrators can now quickly identify errors, detect anomalies, and troubleshoot issues without manually analyzing individual log files from multiple sources. The centralized approach also allows historical analysis of trends and patterns, enabling proactive management of system performance and security incidents. By providing clear insights through interactive dashboards, the system supports informed decision-making and reduces downtime.

Overall, the project proves that a well-structured ELK-based monitoring system can greatly enhance system performance, improve security visibility, and simplify the management of complex, distributed, and containerized environments. The modular design of the system allows it to be easily extended in the future, such as incorporating advanced alerting mechanisms, machine learning-based anomaly detection, and support for cloud or Kubernetes environments. This makes the system adaptable and capable of meeting the evolving needs of modern IT infrastructures.

9. REFERENCES

[1] N. M. K. Koneru, “Centralized Logging and Observability in AWS- Implementing ELK Stack for Enterprise Applications,” International Journal of Computational and Experimental Science and Engineering, vol. 11, no. 2, pp. 3428–3451, 2025.

[2] B. Lyu, “Design and Implementation of Real-Time Log Analysis Based on ELK,” Applied and Computational Engineering, vol. 77, 2024. EWA Pub

[3] P. P. Bavaskar, O. Kemker, and A. K. Sinha, “A Survey On: Log Analysis With ELK Stack Tool,” SSRN Electronic Journal, vol. 6, no. 4, pp. 965-968, 2019. ResearchGate

[4] V. Soma, “Monitoring and Observability with ELK (Elasticsearch, Logstash, Kibana),” Journal of Engineering and Applied Sciences Technology (JEAST), Mar. 2024. Online Scientific Research

[5] R. Banerjee, “A Comparative Analysis of System Monitoring Architecture: Evaluating Prometheus, ELK Stack and Custom Dashboards,” International Journal of Science and Research (IJSR), vol. 14, no. 9, 2025. ResearchGate

[6] Ruan et al., “Integrating Logstash and Kibana for Full-Stack Monitoring Solutions,” International Journal of Innovative Research in Multidisciplinary Physical Sciences (IJIRMPS), 2020. IJIRMPS

[7] Y. Li et al., “A Cloud-Based Framework for Large-Scale Log Mining through Apache Spark and Elasticsearch,” Applied Sciences, vol. 9, no. 6, 2019. MDPI

[8] Satish Yerram, "Kibana Deployment in AWS for Log Analysis: A Research-Based Study on Observability and Elastic Stack Integration," International Journal of Scientific and Technical Advancements (IJSAT), vol. 16, no.3, 2025. IJSAT

[9] "Centralized Log Management Using Elasticsearch, Logstash and Kibana," Semantic Scholar, study on ELK log centralization and analysis components. Semantic Scholar

[10] T. Zhang, H. Qiu, G. Castellano, M. Rifai, C. S. Chen, and F. Pianese, "System Log Parsing: A Survey," arXiv Preprint, 2022 (journal-style survey on log parsing relevant to ELK processing).