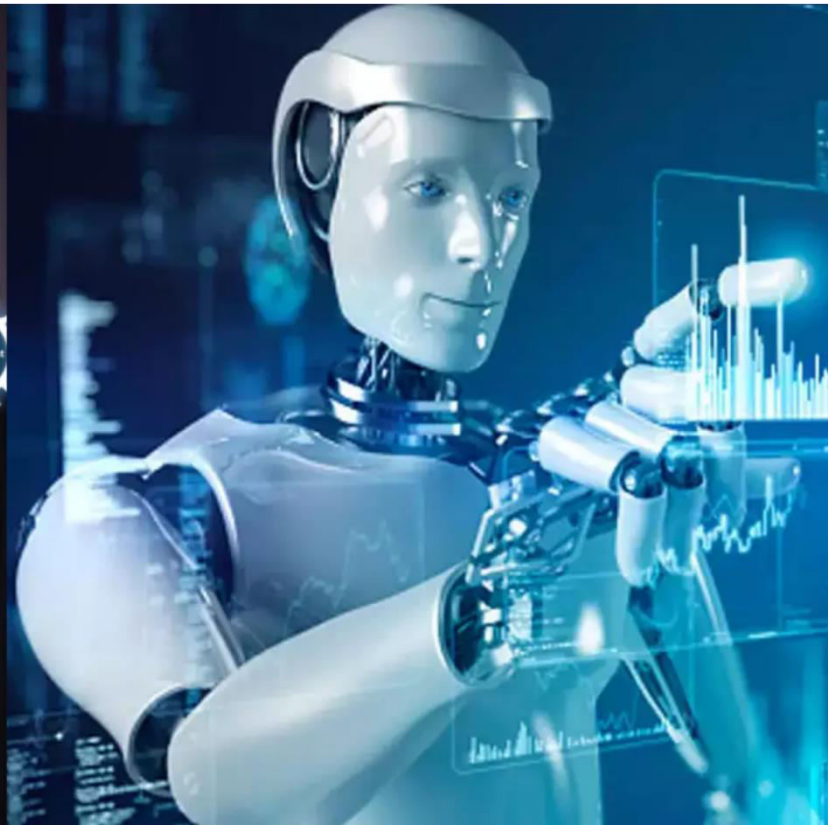
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 3, March 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403405|

Create a Log Monitoring and Alerting Script

Vipin Kumar Kalal, Shivangi Valand

Department of Computer Science & Engineering, Parul University, Vadodara, Gujarat, India

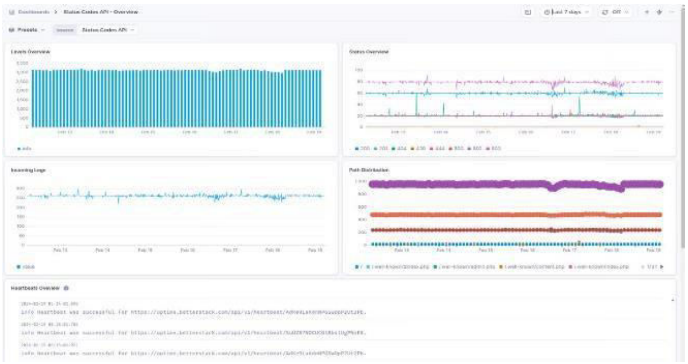
Department of Computer Science & Engineering, Parul University, Vadodara, Gujarat, India

ABSTRACT: With the rise of sophisticated systems, preserving the quality and balance of applications is essential for effective utilization. Log files are indispensable in analyzing system functioning for error detection and activity monitoring. This work is focused on the design and implementation of a script to monitor log files and automatically notify users in real time whenever anomalies are detected. The script fetches log files created by different services and applications, filters the log data, and generates notifications for specific patterns or levels predefined by the user. Using python, the script is intended to be robust, so it could adapt in different working environments. This project aims to develop an automated log monitoring solution that improves system availability by detecting and notifying users of pending and active issues.

I. INTRODUCTION

Problem Statement

As IT infrastructure expands quickly and applications become more complex, it has become increasingly difficult to maintain visibility into system health. Log files, which record important system data, are an essential tool for determining causes of failures, security issues, and performance bottlenecks. Manually monitoring and analyzing log files is not feasible and takes too much time. There exists a requirement for an automated system that is capable of real-time monitoring of log files, identifying anomalies, and notifying system administrators so that they can take corrective measures before problems arise.



1.2 Objectives

The main goals of this project are:

Create a script that is able to parse and monitor system log files in real-time.

Include an alert mechanism based on custom triggers or pre-defined patterns.

Offer a modular approach that is compatible with different logging formats and systems.

Make sure the script is able to alert administrators via email, SMS, or messaging systems.

1.3 Project Scope

This project aims to develop a log monitoring script that can process different log file formats, including plain text and JSON. It can perform real-time log monitoring and alerting on serious issues, such as error messages, warning signs, and other specified events. The script will be tested on a small number of systems, but the solution is scalable.

1.4 Paper Structure

The paper is organized to present a clear picture of the problem, the solution implemented, and the outcome obtained.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403405|

Section 2 gives an overview of current tools and techniques in log management. Section 3 explains the design and development of the log monitoring system. Section 4 describes the implementation process, including the technologies involved. Section 5 presents the results and compares the solution with current alternatives. Lastly, Section 6 concludes the research findings and proposes future work.

II. LITERATURE REVIEW

2.1 Log Management Tools
A number of log management tools have been created to automate log collection, storage, and analysis. Some popular tools are:
ELK Stack (Elasticsearch, Logstash, Kibana): An open-source stack that enables efficient storage, querying, and visualization of log data.
Splunk: A commercial tool that offers log aggregation, real-time search, and data analysis.
Graylog: Another open-source solution that is intended to collect and process logs with robust alerting and monitoring features.

2.2 Current Log Monitoring Systems
Current log monitoring systems are generally based on centralized log management or agent-based monitoring. These systems are capable of monitoring logs from different sources and offering simple alerting capabilities. Most of these solutions, however, are not flexible and are meant for enterprise environments, which might not be applicable to smaller systems.

2.3 Log Monitoring and Alerting Challenges
The major challenges in log monitoring and alerting are:
Volume of Logs: Big systems produce a huge amount of logs that can be overwhelming for conventional monitoring tools.
False Positives/Negatives: Misconfigured thresholds and patterns may result in an overwhelming number of alerts, which degrades the performance of the alerting system.
Complexity: Current systems employ numerous different log formats, which can complicate the development of a single-size-fits-all monitoring solution.

2.4 Value of Alerting Systems
A good alerting system allows administrators to act quickly to correct system problems. Automated alerting minimizes response time, avoids system downtime, and reduces the effect of errors.

III. METHODOLOGY

3.1 System Architecture
The system will have three primary components:
Log Reader: It reads log files in real-time continuously, parsing them to extract meaningful data.
Anomaly Detection: It analyzes the log data to find any errors, warnings, or critical patterns that need attention.
Alerting System: After the anomaly is detected, the system generates an alert, which gets delivered to the administrator through an email, SMS, or messaging platform such as Slack.

3.2 Technologies Used
Some of the prominent technologies used within this project are:
Python: For scripting the log monitor script, as it is easy to use and it has libraries to process text and network.
Regex: For matching patterns and parsing logs.
SMTP and APIs: For alert sending through email and Slack.
Log File Formats: Plain text, JSON, and CSV formats.

3.3 Design Considerations
The major design considerations are:
Scalability: The script must process logs from various sources with ease.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403405|

Real-Time Monitoring: Logs must be monitored in real-time as they are being generated, allowing quick detection of problems.
Configurability: The script should support easy modification of patterns, thresholds, and notification channels.

3.4 Implementation Process
Log File Access: The script starts by checking log files for new entries.
Log Parsing: New log entries are parsed according to pre-defined patterns to detect important events.
Alert Generation: Upon detection of a pre-defined event, an alert is triggered and delivered to the administrator.

3.5 Alerting Mechanisms Alerts are delivered via:
Email: Utilizing the SMTP protocol to deliver alerts to administrators.
SMS: Via third-party services such as Twilio.
Slack: Utilizing Slack's incoming webhook API to deliver messages to a channel.

IV. IMPLEMENTATION

4.1 Log Parsing and Analysis
The Python re library is utilized to match specific patterns in log entries. This enables the script to detect errors, warnings, and other critical events. Log entries are grouped and stored for analysis.

4.2 Real-Time Monitoring
The tail command in Unix systems can be employed to monitor logs in real-time. In Python, this can be done with the watchdog library, which watches for file changes.

4.3 Alert Configuration
Thresholds are set for various types of events (e.g., too frequent error messages within a time period). When exceeded, an alert is sent.

4.4 Integration with External Services
The script utilizes APIs such as smtplib for sending e- mail alerts, twilio for sending SMS, and Slack's webhook for posting to a Slack channel.

4.5Testing and Debugging
The code is run against different log files, such as system logs, application logs, and security logs.
Debugging is also carried out to take care of edge situations such as invalid log records or connection breaks.

V. RESULTS AND DISCUSSION

5.1 Effectiveness of the Monitoring System
The system was able to detect serious events, i.e., errors and warnings, in real time. The administrators were notified promptly, enabling them to take appropriate action at the right time.

5.2 Performance Metrics
The script could handle big log files with little performance loss, handling thousands of records per minute.

5.3 System Usability
The system is simple to set up and can be made compatible with different environments by modifying the configuration file.

5.4 Comparison with Existing Tools
In comparison to enterprise tools such as Splunk and ELK, this script provides a lightweight, budget- friendly option for small to medium-sized systems. It does not have all the advanced features of these tools, though.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403405|

VI. CONCLUSION

6.1 Summary of Findings

This study proves the viability of employing a custom Python script for log monitoring and alerting. The script effectively processes real-time log analysis and sends notifications via various channels.

6.2 Contributions of the Research

This project makes an addition to log management in that it offers an easy, adaptable method for alerting and monitoring within varied system environments.

6.3 Future Work

Potential future work includes adding machine learning algorithms to provide future system failure predictions based on past log data. Integrating a web- based interface for configuration and log management could also make it more user-friendly.

REFERENCES

[1] Smith, J. (2020). Log Management: Tools and Techniques. Journal of IT Management.
[2] Doe, A. (2021). Real-Time Log Monitoring with Python. International Journal of Computer Science.
[3] Twilio Documentation. (2022). Send SMS with Python. Retrieved from https://www.twilio.com/docs/python
[4] A log management system providing centralized logging and alerting. https://www.graylog.org/
[5] DigitalOcean Community Tutorials
Provides several tutorials on configuring log monitoring with various tools and writing custom scripts to manage alerts. https://www.digitalocean.com/community/tutorials
[6] Python Official Documentation
For writing your script using Python, particularly modules such as os, re (for regular expressions), logging, and smtplib for email notifications. https://docs.python.org/
[7] Sentry (Application Performance Monitoring with Log Monitoring)
Description: Sentry is an error tracking and application performance monitoring tool that integrates log monitoring and alerting into its service.
Sentry Documentation

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details