|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [alert\_automation.py]** | | |
| **Summary***:*  This file automates the monitoring of various log files (error, security, performance) and triggers alerts based on new log entries. Additionally, it allows setting alert thresholds and scheduling log cleanups to prevent file overflow, enhancing the system's robustness and alert responsiveness. | | |
| ***Processes*** | | |
| * **Log Monitoring** | **Monitors the error, security, and performance log files for new entries, triggering alerts when specific criteria are met.** | |
| * **Alert Dispatching** | **Calls specific alert functions in the AlertManager based on log types (e.g., send\_error\_alert for errors).** | |
| * **Threshold Setting** | **Allows setting custom alert thresholds for each log type, enhancing control over alert sensitivity.** | |
| * **Scheduled Log Cleanup** | **Periodically clears log files to prevent overflow, based on a configurable interval (default: 24 hours).** | |
| * **Shutdown Management** | **Manages a graceful shutdown of monitoring processes, ensuring no unintended monitoring continues after termination.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Environment Variables** (dotenv): For thresholds, file paths, cleanup interval. | **Alert Manager** (alert\_manager): Sends alerts for error, security, and performance log triggers. | |
| * **Error Log** (logs/error.log): Monitors for new error entries. | **Log Files** (logs/error.log, logs/security.log, logs/performance.log): Writes log cleanup status. | |
| * **Security Log** (logs/security.log): Monitors for new security events. | **Console/Log File** (alert\_automation.log): Logs alert triggering and cleanup activities. | |
| * **Performance Log** (logs/performance.log): Monitors for new performance issues. |  | |
| **Expected input into file:** | | **Expected output from file:** |
| Log Entries: New entries in error, security, or performance logs.  Environment Variables: Error, security, and performance thresholds; log file paths; cleanup interval.  Threshold Values: Custom threshold settings provided through set\_alert\_thresholds method. | | Alert Triggering: Calls specific alert methods in AlertManager for detected issues (e.g., send\_error\_alert).  Log Cleanup Status: Periodically clears log contents and records this in the alert\_automation.log.  Monitoring Status Logs: Writes monitoring start, stop, and error details in alert\_automation.log. |
| **Things that need to be taking place:** | | |
| |  | | --- | | **● Continuous Monitoring: Actively monitor the designated log files for new entries, and trigger alerts as soon as they appear.** |  |  | | --- | | **● Threshold Configuration: Enable administrators to configure alert thresholds to avoid alert fatigue or excessive notifications.** |  |  | | --- | | **● Scheduled Cleanup Execution: Ensure log cleanup occurs regularly, retaining only recent entries to prevent overflow.** |  |  | | --- | | **● Graceful Shutdown: Monitor shutdown events to stop all monitoring threads smoothly, ensuring no unintentional process continuation.** |  |  | | --- | | **● Error Handling and Logging: Log any issues during monitoring or cleanup in alert\_automation.log for audit and debugging purposes.** |  |  | | --- | | **● Responsive Alert Dispatching: Ensure immediate alert dispatch through AlertManager whenever a new log entry matches criteria.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| Contribution to the Entire Project:   The **alert\_automation.py** file is critical for automating real-time monitoring and alerting in Razda Market. By automating log surveillance, it ensures prompt response to errors, security issues, and performance concerns. The scheduled cleanup function maintains log files at a manageable size, preventing potential storage issues. Together, these functions enhance the system's resilience and responsiveness to operational anomalies. | | |
| * Oliver Smith (Razda Admin) Nov 8, 2024: Developed functionality for monitoring, alert triggering, log cleanup scheduling, and shutdown handling in alert\_automation.py. | | |