[Open]

The document provides a comprehensive overview of multiple monitoring systems designed to ensure system health, performance, and security across various environments including servers, containers, APIs, and network devices.

- **Self-Healing Engine Capabilities:** A Bash-based system continuously monitors system health with stateful alerting, automatic log rotation, modular checks, and tiered healing strategies for resources, Docker containers, network connectivity, file permissions, and services. It includes monthly audits and intelligent cleanup with webhook notifications.
- **Docker and Container Management:** Features include container health monitoring, automatic restarts with failure tracking, special Redis handling, Docker daemon status checks, and resource usage tracking with restart frequency limits and cooldowns to prevent service flapping.
- **Network and Connectivity Monitoring:** Monitors SSH tunnels with fallback hosts, port and internet connectivity, and network configuration including route and interface status.
- **Security and Permission Auditing:** Tracks login attempts, sudo usage, file permission auditing, process security, and log analysis for critical system errors and patterns.
- **API Monitoring System:** Real-time health checks for over 30 API endpoints with multi-method support, response time and status code tracking, authentication management, and feature state monitoring. Supports Prometheus metrics integration and persistent state storage in Elasticsearch.
- **Database and Message Queue Monitoring:** Covers Elasticsearch, CouchDB, MySQL, RabbitMQ, and Redis monitoring for health, size, replication, queue metrics, and performance indicators.
- **Job Handler and File System Monitoring:** Tracks multiple job handlers, monitors file existence and modification, and analyzes log files for job execution status, running in a reliable separate daemon thread.
- **Alerting and Notification Systems:** Provides SMTP email alerts, HMAC-signed webhook notifications with retry and backoff logic, and various alert types including stale activity, reminders, and critical issue notifications.
- **Vessel and Network Device Monitoring:** Grafana dashboards monitor SSH connectivity of vessels and CloudRouter devices, tracking uptime, bandwidth, route changes, peer status, latency, failovers, and alerts with historical data analysis.
- **System and Docker Monitoring with Prometheus and Grafana:** Node Exporter on Ubuntu systems exposes CPU, memory, disk, network, and hardware metrics for real-time visualization and alerting. A Telegraf, InfluxDB, and Grafana stack provides detailed Docker container and host metrics with scalable dashboards and proactive alerts.

[Open]

Self-Healing Engine v8.6

Overview

The Self-Healing Engine is a robust Bash-based monitoring system that continuously monitors system health and automatically resolves common issues. It features intelligent webhook notifications, comprehensive system auditing, and tiered healing approaches for maximum system reliability.

Key Features

Core Functionality

- Continuous Monitoring: Runs as a foreground process ideal for system service management
- Stateful Alerting: Prevents alert spam by tracking issue states and sending notifications only on state changes
- Graceful Shutdown: Signal trapping for clean exits and proper cleanup
- Automatic Log Rotation: Prevents disk exhaustion with configurable log size limits (5MB default, 3 backups)
- Modular Architecture: Per-check scheduling with customizable intervals

Health Monitoring & Healing

- System Resource Monitoring
- CPU Monitoring: Tracks CPU usage with 90% threshold alerting
- Memory Monitoring: Monitors memory consumption with 90% threshold alerting
- Disk Space Monitoring: 80% threshold with tiered cleanup strategies
- Filesystem Health: Writability checks and read-only filesystem detection

Docker Container Management

- Container Health Checks: Monitors running containers with exclusion lists
- Automatic Container Restart: Intelligent restart logic with failure tracking
- Redis-Specific Healing: Special handling for Redis restart loops (clears RDB/AOF files)
- Docker Daemon Monitoring: Service status checks with protected restart mechanisms
- Resource Usage Tracking: Container CPU and memory monitoring

Network & Connectivity

- SSH Tunnel Monitoring: Monitors with fallback host support
- Port Monitoring: Tracks listening services and network connectivity

[Open]

- Internet Connectivity: Automated connectivity tests (ping 8.8.8.8)

File System & Permissions

- Permission Healing: Automatic correction of file permissions for critical paths
- File Permission Monitoring: Checks for world-writable system files
- Directory Cleanup: Automated cleanup of old files and directories based on patterns
- Specific Path Monitoring:

Service Management

- vnStat Management: Automatic installation/upgrade to version 1.18 with local/remote package support
- InfluxDB Monitoring: Database connectivity and health checks
- Additional Services: Configurable monitoring for custom services
- Hardware Monitoring: System hardware status checks

Advanced Monitoring Features

- Monthly System Audit
- Comprehensive System Analysis: Detailed monthly reports on the 8th of each month
- System Overview: OS info, kernel version, uptime, load averages, CPU cores, memory
- Disk Usage Analysis: Detailed disk usage with top 5 largest directories
- Memory Analysis: Usage patterns and top 5 memory-consuming processes
- Docker Analysis: Container health, resource usage, volume analysis, network status

Security Analysis:

- Failed/successful login tracking
- Sudo usage monitoring
- SSH connection analysis
- Process security assessment
- File permission auditing

Log Analysis: Error/warning counts across system logs

Critical Pattern Detection: OOM kills, disk errors, segmentation faults

Network Configuration: Route analysis and interface status

Service Status: Critical service health monitoring

[Open]

Intelligent Cleanup System

- Tiered Cleanup Strategy: Progressive cleanup based on disk usage severity
- Pattern-Based Cleanup: Regex-based file and directory removal
- Configurable Retention: 15-day default retention with customizable periods
- Aggressive Cleanup Mode: Optional intensive cleanup for critical situations

Notification System

- Webhook Integration
- Primary Notification Method: HMAC-authenticated webhook notifications
- Retry Mechanism: 3 retries with 5-second delays
- Comprehensive Payloads: Detailed system status and error information
- State-Based Notifications: Only sends on state changes or 24-hour reminders
- Monthly Audit Reports: Automated delivery of comprehensive system reports

Notification Types

- NEW_PROBLEM: First occurrence of an issue
- RESOLVED: Issue resolution notifications (configurable)
- REMINDER: 24-hour reminders for persistent issues
- MONTHLY_AUDIT_COMPLETED: Monthly system audit reports
- CRITICAL: System-critical issues requiring immediate attention

Advanced Features

- Docker Service Protection
- Restart Frequency Limits: Maximum 2 Docker service restarts per hour
- Cooldown Periods: 30-minute minimum between restarts
- Escalation Handling: Manual intervention required after 3 consecutive failures
- State Tracking: Persistent restart history with automatic cleanup

Error Handling & Recovery

- Comprehensive Error Trapping: Script-level error handling with line number reporting
- Dependency Checking: Validates all required system commands
- Edge Case Handling: Read-only filesystems, Docker daemon failures
- Graceful Degradation: Continues operation even when individual checks fail

Security Features

- Root Privilege Requirement: Ensures proper system access
- File Permission Management: Secure log file permissions (640)
- State File Protection: Secure temporary file handling

[Open]

- Authentication Monitoring: Tracks login attempts and sudo usage

API Monitoring System

Features

API Endpoint Monitoring

- Real-time API Health Checks: Monitors 30+ API endpoints with configurable intervals
- Multi-Method Support: GET and POST endpoint monitoring with custom request bodies
- Response Time Tracking: Measures and reports API response times in seconds
- Status Code Monitoring: Tracks HTTP status codes and failure messages
- Authentication Management: Automatic token refresh and management for secured endpoints
- Version Compatibility Checks: Compares versions across different endpoints for consistency
- Feature State Tracking: Monitors individual feature flags and their states

Monitoring Intervals

- Regular Monitoring: 5-minute intervals for critical endpoints (configurable via MONITORING_INTERVAL_SECONDS)
- Extended Monitoring: 10-minute intervals for infrastructure components (configurable via TEN_MINUTE_INTERVAL_SECONDS)
- Hourly Monitoring: 1-hour intervals for resource-intensive endpoints (configurable via HOURLY_INTERVAL_SECONDS)

Prometheus Metrics Integration

- Custom Registry: Isolated metrics registry for clean metric collection
- Comprehensive Metrics: 40+ different metric types including:
- API response times and status codes
- Database sizes and document counts
- Queue depths and consumer counts
- Job handler execution status
- Feature toggle states
- Application version information

Database Monitoring

- Elasticsearch Monitoring:

[Open]

- Index health, status, and size tracking
- Document count and deletion monitoring
- Primary and replica shard information
- Two different metric approaches for flexibility

CouchDB Monitoring:

- Database size and document count tracking
- Active size monitoring
- Replication status between local and remote instances
- Replication statistics (docs read/written, failures)

MySQL Integration:

- Job handler database connectivity
- Query execution monitoring
- Connection health checks

Message Queue & Cache Monitoring

- RabbitMQ Monitoring:
- Queue message counts (total, ready, unacknowledged)
- Consumer count tracking
- Memory usage per queue
- Queue-specific metrics with labels

Redis Monitoring:

- Memory usage tracking
- Key count monitoring
- Connection health checks
- Performance metrics

Job Handler Monitoring

- Multi-Service Monitoring: Tracks 9 different job handlers including:
- NSEDataImportService
- ErpPriorityListImportHandlerService
- ErpPositionListImportHandlerService
- ErpUserListImportHandlerService
- WorkOrderImportService
- ErpPurchaseOrderImportHandlerService
- ROBUpdateService
- WorkOrderImportCompletedListHandlerService

[Open]

- WorkOrderImportClosedListHandlerService

File System Monitoring: Tracks file existence and modification times

Log File Analysis: Monitors log files for job execution status

Independent Thread: Runs in a separate daemon thread for reliability

Alerting System

- Stale Activity Detection: Monitors for inactive systems based on configurable thresholds
- Email Notifications: SMTP-based email alerts with customizable templates
- Webhook Integration: HTTP webhook notifications with:
- Configurable authentication headers
- Request signing with HMAC-SHA256
- Retry logic with exponential backoff
- Timeout configuration

Alert Types:

- Initial stale activity alerts
- Periodic reminder alerts
- Recovery notifications
- Job handler failure alerts

Security Features

- Token Management: Automatic OAuth token refresh and validation
- Client Authentication: Separate client credentials for different endpoints
- Webhook Signing: HMAC-SHA256 signature verification for webhook security
- Environment-based Configuration: All sensitive data configurable via environment variables

State Persistence

- Elasticsearch State Storage: Persistent state management using Elasticsearch indices
- Alert State Tracking: Prevents duplicate alerts through state persistence
- Recovery Detection: Robust recovery detection to prevent false alerts on restart
- In-Memory Caching: Efficient in-memory state management for performance

Configuration Management

- Environment Variables: 50+ configurable environment variables
- Default Values: Sensible defaults for all configuration options

[Open]

- Runtime Configuration: Dynamic configuration loading without restart
- Multi-Environment Support: Easy deployment across different environments

Monitoring Endpoints Coverage

- Application Metadata: Version info, feature toggles, app size
- Business Logic: Purchase orders, tags, products, work orders
- System Health: Door sensors, metadata, configurations
- Search Functionality: Multi-criteria search endpoints
- Location Services: Location validation and search
- Stock Management: Stock-take operations and history

Advanced Features

- Concurrent Processing: Thread pool executor for parallel monitoring
- Error Resilience: Comprehensive error handling and recovery
- Logging: Detailed logging with configurable levels
- Health Checks: Built-in health check endpoints
- Metric Labeling: Rich metric labeling for detailed analysis
- Time Zone Handling: UTC-based timestamp management

Deployment Features

- Docker Ready: Containerized deployment support
- Resource Efficient: Optimized for minimal resource usage
- Scalable Architecture: Designed for horizontal scaling
- Monitoring Integration: Ready for Grafana/Prometheus stack integration

Monitored Services

- Primary APIs
- Main API (Port 4004): Core business logic endpoints
- Device API (Port 4002): Device management endpoints

Infrastructure Services

- Elasticsearch: Document storage and search
- CouchDB: Document database with replication
- Redis: In-memory cache and session storage
- RabbitMQ: Message queue system
- MySQL: Relational database for job handlers

Job Processing

- NSE Data Import: Stock exchange data processing

[Open]

- ERP Integration: Enterprise resource planning sync
- Work Order Management: Work order lifecycle management
- ROB Updates: Remaining on Board inventory updates

Database Connections

- Elasticsearch, CouchDB, Redis, RabbitMQ, and MySQL connection parameters
- Timeout and retry configurations

Alerting Configuration

- SMTP settings for email notifications
- Webhook configuration for HTTP notifications
- Alert thresholds and intervals

Vessel Connection Monitoring

A Grafana-based system monitors SSH connectivity of 66+ vessels in real time, providing live status indicators and historical charts. The system automates connectivity checks, reduces manual effort, enables faster issue detection, and supports long-term analysis.

CloudRouter Monitoring

Developed a comprehensive Grafana dashboard for real-time monitoring of CloudRouter devices across the network.

- Uptime: Continuous monitoring to detect offline routers or connectivity drops.
- Bandwidth usage: Real-time and historical tracking of inbound and outbound traffic per interface.
- Route changes: Detection of dynamic routing updates and anomalies in routing tables.
- Peer status: Monitoring the health and connectivity of BGP/OSPF peers.
- Latency: Round-trip time measurement to critical network destinations for performance analysis.
- Failovers: Automatic detection of primary/backup route failovers and path changes.
- Alerts and Notifications: Configured threshold-based alerts using Grafana and integrated webhook notifications for critical issues.
- Benefits: Provides network administrators with immediate visibility into network performance, reduces manual monitoring effort, and enables rapid troubleshooting of network incidents.
- Historical Analysis: Stores and visualizes historical data to identify trends, recurring issues, and plan capacity upgrades.

[Open]

VNStat Monitoring

Developed a containerized VNStat-based application for Linux systems to collect and monitor network traffic statistics reliably.

- Hourly, daily, monthly, and total data usage tracking across all network interfaces.
- Per-interface monitoring: Differentiates traffic for physical interfaces, VLANs, and virtual interfaces.
- Containerization: Runs inside Docker for easy deployment, scalability, and isolated resource usage.
- Data Storage: Maintains historical traffic logs to support long-term usage analysis and reporting.
- Custom Reports: Generates detailed network usage reports, enabling administrators to track usage patterns and identify bandwidth hogs.
- Integration: Can feed metrics into Grafana dashboards or Prometheus for centralized monitoring.
- Benefits: Enables accurate and automated tracking of bandwidth usage, simplifies network monitoring, supports capacity planning, and helps detect abnormal traffic behavior in real time.

Prometheus & Node Exporter Monitoring for Ubuntu Systems

Prometheus Overview

- Open-source monitoring and alerting toolkit.
- Scrapes metrics from configured targets over HTTP at regular intervals.
- Stores metrics as time-series data with labels for identification.
- Integrates seamlessly with Grafana for real-time dashboards and alerts.

Node Exporter Overview

- Installed on Ubuntu systems to expose OS-level and hardware metrics.
- Runs an HTTP endpoint (default port 9100) that Prometheus scrapes.
- Provides comprehensive insight into system performance, resource usage, and health.

Key Metrics Provided by Node Exporter

- CPU Metrics
- CPU usage by mode: user, system, idle, I/O wait, etc.
- Load averages for 1, 5, and 15 minutes.

Memory Metrics

- Total system memory.

[Open]

- Available memory and memory in use.
- Buffers and cached memory details.
- Swap usage and swap availability.

Disk Metrics

- Total and available space per filesystem.
- Disk usage percentage and filesystem utilization.
- Disk I/O statistics: read/write operations, I/O time, and wait time.

Network Metrics

- Bytes transmitted and received per network interface.
- Packet errors and drops for each interface.
- Interface-level traffic monitoring for inbound/outbound data.

System Metrics

- System uptime and boot time.
- Running and blocked processes.
- Context switches and interrupts.

Hardware Metrics (if supported)

- CPU temperature and fan speed.
- Other sensor-based hardware statistics (voltage, power, etc.).

Grafana Visualization

- Dashboards display Node Exporter metrics in real-time.
- Common panels include:
- CPU usage per core and mode.
- Memory and swap usage over time.
- Disk I/O and filesystem usage trends.
- Network traffic per interface with historical comparison.
- System load averages and uptime metrics.

Supports alerts for threshold violations, e.g., high CPU, low memory, or high disk usage.

Historical data helps in trend analysis, capacity planning, and long-term system monitoring.

Benefits

- Complete visibility into Ubuntu system performance.
- Real-time monitoring and automated alerting.
- Historical analysis for resource usage trends.

[Open]

- Easily extendable to monitor additional services like Docker, databases, and network components.
- Lightweight, efficient, and scalable solution for system health monitoring.

Docker Monitoring with Grafana, InfluxDB, and Telegraf

Overview

A monitoring stack using Telegraf, InfluxDB, and Grafana provides real-time visibility into Docker container performance.

Telegraf acts as the metrics collector and agent inside Docker hosts, gathering container, system, and application metrics.

InfluxDB stores the collected metrics as a time-series database.

Grafana visualizes metrics, trends, and alerts in interactive dashboards.

Metrics Collected

- Container Performance Metrics
- CPU usage per container (user/system/total).
- Memory usage and limits for each container.
- Network traffic (bytes sent/received) per container interface.
- Disk I/O: read/write bytes and operations per container.
- Container uptime and restart count.

Docker Daemon Metrics

- Total number of running, paused, and stopped containers.
- Number of images, volumes, and networks.
- Event logs and errors from Docker daemon.

System Metrics (via Telegraf Docker plugin)

- Host CPU and memory usage.
- Host disk usage and filesystem statistics.
- Network interface statistics for the host.

Resource Limits & Health

- Memory limits exceeded alerts.
- CPU throttling due to container limits.
- Container health status if health checks are configured.

[Open]

Grafana Visualization

- Dashboard Panels
- Container Overview Panel: Lists all containers with real-time CPU, memory, and network usage.
- CPU Usage Panel: Displays CPU usage per container and overall host CPU utilization.
- Memory Usage Panel: Tracks memory usage against container limits and host usage.
- Disk I/O Panel: Monitors read/write bytes and operations per container.
- Network Panel: Monitors per-container network throughput and errors.
- Container Health Panel: Shows container uptime, restart count, and health status.
- Host Overview Panel: Summarizes host CPU, memory, disk, and network usage.
- Historical Trends: Line graphs showing resource utilization over hours, days, or weeks.
- Alerts & Notifications: Configurable alerts for high CPU/memory usage, container down events, or exceeding resource limits, integrated with email, Slack, or webhook channels.

Benefits

- Real-Time Monitoring: Detect performance bottlenecks and container failures immediately.
- Historical Analysis: Evaluate trends in container resource usage to optimise deployments.
- Proactive Alerts: Automatic notifications for resource threshold breaches or container crashes.
- Scalability: Easily monitor dozens of containers across multiple hosts.
- Integration Ready: Works alongside system monitoring stacks like Node Exporter for complete host + container visibility.