                                                          High-level-Document

                                            Web-based network monitoring system

Table of Contents

1. Introduction

2. Objective

3. Technology Stack

4. Architectural Overview

5. Features

5.1. Real time Network monitoring

5.2. Performance Analysis

5.3. Network Configuration Management

5.4. Alerts and Notifications

5.5. Reporting and Analytics

6. Non Functional Requirements

7. User Acceptance Criteria

8. Risk and Mitigation Strategies

9. Conclusion

**1. Introduction**

   The Web-Based Network Monitoring System is a comprehensive tool designed to monitor and manage network infrastructure efficiently. It provides real-time insights into network performance, availability, and security, empowering administrators to make informed decisions and ensure optimal network operation.

**2. Objectives**

* Provide real-time monitoring of network devices, servers, and services.
* Track network performance metrics such as bandwidth utilization, latency, and packet loss.
* Alert administrators about critical issues and potential security threats.
* Generate reports and analytics to facilitate decision-making and troubleshooting.
* Support scalability to accommodate growing network infrastructure.
* Ensure ease of use through a user-friendly web interface.

**3. Technology Stack**

**3.1. Frontend Development:**

* + Framework: Angular or React for building responsive, interactive user interfaces.
  + Languages: HTML, CSS, JavaScript/TypeScript.
  + UI Libraries: Bootstrap or Material UI for styling and layout.
  + HTTP Client: Angular Http Client for making HTTP requests to backend services.

**3.2. Backend Development:**

* + Framework: Spring Boot for building RESTful APIs and microservices.
  + Language: Java for backend logic implementation.
  + Database: MySQL for storing user data, server details, performance, etc.
  + ORM: Spring Data JPA or Hibernate for object-relational mapping.
  + Authentication: Spring Security for handling user authentication and authorization.

**4. Architectural Overview:**

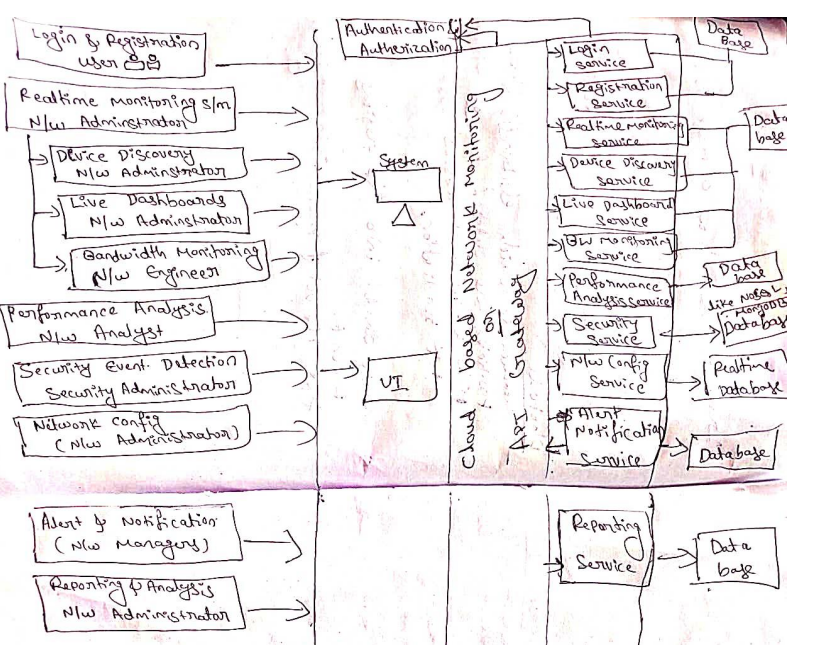


  Fig: Architecture

**4.1. System Architecture**

The Web-Based Network Monitoring System comprises the following components:

* Data Collection Agents: These agents are deployed across the network to collect data from devices, servers, and services. They gather information such as system metrics, network traffic, and device status.
* Data Processing Engine: This component processes the collected data, aggregates it, and stores it in a centralized database. It performs analysis to derive insights and detect anomalies.
* Web Application: The user interface accessed via web browsers. It provides dashboards, reports, and configuration options for administrators to monitor and manage the network.
* Alerting System: Monitors data in real-time and triggers alerts based on predefined thresholds or abnormal patterns. Alerts are delivered via email, SMS, or other notification channels.

**5. Features**

**5.1. Real-time Network Monitoring**

**5.1.1 Device Discovery**

* The network monitoring platform uses a combination of active and passive discovery techniques to automatically identify and map all devices connected to the network.
* Active discovery involves actively probing the network using protocols like SNMP or ICMP to detect devices.
* Passive discovery involves analysing network traffic to identify devices based on their communication patterns.
* The platform may also allow for manual device addition, enabling administrators to add devices that were not automatically discovered.

**5.1.2 Live Dashboards**

* The network monitoring platform provides real-time visualizations of network performance metrics through customizable dashboards.
* These dashboards display key performance indicators (KPIs), such as bandwidth utilization, latency, packet loss, and device status.
* Dashboards are customizable based on user roles, allowing different stakeholders (e.g., network administrators, IT managers) to view relevant information tailored to their responsibilities and interests.
* Platform continuously updates the dashboards with the latest data, providing users with up-to-date insights into network performance.

**5.1.3 Bandwidth Monitoring**

* The system should provide real-time monitoring of bandwidth usage across different network segments.
* Bandwidth usage data should be updated dynamically and displayed on the monitoring interface without delays.
* Users should be able to view bandwidth usage by device, application, and protocol in real-time.

**5.2 Performance Analysis**:

* The system should be able to automatically discover and map the network topology, identifying devices, connections, and their relationships without manual intervention.
* Allow for manual input or import of network information for devices that may not be discoverable automatically.
* Provide real-time updates to reflect changes in the network, such as new devices, connections, or network failures.
* Users should be able to interact with the map, zooming in/out, panning, and selecting specific elements for more detailed information.
* Display detailed information about each device, including IP addresses, MAC addresses, device type, and status.
* Store performance data with an appropriate level of granularity to capture detailed information about network devices, applications, and services.
* Implement compression techniques to optimize storage efficiency without sacrificing data integrity.
* Display trend analysis through graphical representations, such as charts or graphs.

**5.3 Network Configuration Management**

**5.3.1 Configuration Backup**

* Implement an automated backup system for network device configurations, utilizing protocols.
* Integrate version control to track changes, treating each backup as a commit for easy management.
* Compare new configurations with previous ones, identifying modifications.
* This ensures data integrity, simplifies configuration management, and aids in troubleshooting network issues efficiently.

**5.3.2 Compliance Checks**

* Develop automated checks to verify network configurations against industry standards and internal policies.
* Set up alerts for non-compliance, notifying administrators via email, SMS.
* Ensure configurations adhere to best practices for security, performance, and compliance benchmarks.
* This proactive approach ensures network security, minimizes risks, and maintains regulatory compliance effortlessly.

**5.4 Alerts and Notifications**

**5.4.1. Customizable Alerts**

* Provide a feature allowing users to set thresholds for parameters such as bandwidth usage, latency, or device status.
* Users can opt for email notifications to receive detailed updates, SMS for immediate attention, or in-app notifications for seamless tracking within the monitoring application.
* Users might want to customize when alerts are active. Setting time windows allows them to receive alerts only during specific periods, such as business hours.
* Design an intuitive dashboard for users to easily view and manage their alert settings. Data Accuracy and Precision Ensure that all data presented on dashboards and reports is accurate and precise. Conduct extensive testing with various network configurations to validate data accuracy under different scenarios.

**5.5 Reporting and Analytics**

**5.5.1. Pre-built Reports**

* Allow users to drill down into specific data points within standard reports for deeper analysis.
* Provide customizable visualization options for custom reports, allowing users to choose the most suitable chart types (e.g., line charts, bar charts, pie charts) for their data.
* Integrate pre-built and custom reports seamlessly with other modules of the network monitoring system, such as real-time monitoring, performance analysis, and security event detection, to provide comprehensive insights.

**5.5.2. Trend Analysis**

* Designed a scalable architecture capable of handling large volumes of historical data and complex predictive analytics computations.
* Continuously monitor key performance metrics such as network bandwidth, latency, and device utilization.
* Storing historical performance data securely for analysis and trend identification.
* Develop machine learning models to forecast future performance trends and capacity requirements based on historical data.
* Generate reports summarizing capacity forecasts, trend analysis findings, and actionable recommendations for capacity planning.

**6. Non-Functional Requirements**

**6.1 Performance**

6.1.1 Response Time

Ensure real-time updates on dashboards with a response time under 3 seconds. Historical performance data retrieval within 5 seconds.

6.1.2 Scalability

The system should handle a 50% increase in monitored devices and network traffic over the next year.

**6.2 Security**

6.2.1 Data Encryption

Implement encryption for data transmission and storage. Role-based access control for user permissions.

6.2.2 Compliance

Adhere to data protection regulations and industry security standards.

**6.3 Usability**

6.3.1 Intuitive Interface

User testing for interface usability.

Personalized dashboards for different user roles.

6.3.2 Mobile Responsiveness

Ensure a seamless experience on various devices, including smartphones and tablets.

**6.4 Disaster Recovery**

6.4.1 Redundancy and Failover

Implement redundant systems and failover mechanisms to ensure continuous monitoring even in the event of hardware or software failures. Conduct regular failover testing to verify the effectiveness of the recovery process.

**7. User Acceptance Criteria**

7.1 Real-time Network Monitoring

Devices are accurately discovered and displayed on the network map. Live dashboards provide real-time updates on network performance.

7.2 Performance Analysis

Historical performance data is accessible and accurate. Network topology mapping is dynamically updated.

7.3 Security Event Detection

The IDS accurately detects and alerts on security anomalies. Log analysis identifies and correlates security incidents.

7.4 Network Configuration Management

Configurations are regularly backed up and version-controlled. Automated compliance checks generate accurate results.

7.5 Alerts and Notifications

Custom alerts are easily configured and trigger as expected. Users receive timely alerts through preferred channels.

7.6 Reporting and Analytics

Pre-built reports provide comprehensive insights into network health.

Custom reports can be generated with accurate data.

7.7 Data Accuracy and Precision

Ensure that all data presented on dashboards and reports is accurate and precise. Conduct extensive testing with various network configurations to validate data accuracy under different scenarios.

7.8 Integration Testing

Conduct thorough integration testing with third-party tools and systems, including SIEM solutions. Verify seamless data flow and synchronization between the network monitoring software and integrated systems. .

**8. Risks and Mitigation Strategies**

8.1 Risks Device Compatibility Issues: Compatibility challenges with diverse network devices. Security Vulnerabilities: Potential vulnerabilities that could be exploited.

8.2 Mitigation Strategies Device Compatibility Issues: Conduct thorough testing with various network devices. Security Vulnerabilities: Engage in regular security audits and implement patch.

**9. Conclusion**

The Web-Based Network Monitoring System is a powerful tool for ensuring the stability, performance,

and security of network infrastructure. By providing real-time insights and proactive alerts, it enables administrators to effectively manage and optimize their networks.

With continuous development and enhancements, it remains at the forefront of network monitoring technology.