Automated IT Infrastructure Monitoring with Log Scraping Using Ansible, Prometheus, Grafana, Python, and MySQL

1. Project Overview

Objective

The objective of this project is to implement an automated IT infrastructure monitoring and log management system using Ansible, Prometheus, Grafana, Python, and MySQL. The system will ensure proactive monitoring, real-time visualization of server metrics, log collection, storage, and automated alerting for system anomalies.

Scope

- Automate deployment of Prometheus, Node Exporter, and Grafana using Ansible.
- Collect system metrics such as CPU, memory, disk usage, and network activity.
- Use Grafana for real-time visualization of metrics.
- Implement Prometheus Alertmanager to send notifications based on predefined thresholds.
- Install Python and MySQL using Ansible for log scraping and storage.
- Use a Python script to scrape logs and store them in an SQL database.
- Install Node Exporter on target servers using Ansible.

2. Project Architecture

Components & Workflow

1. **Ansible** – Automates the installation of Prometheus, Node Exporter, Grafana, Python, and MySQL.
2. **Prometheus** – Collects real-time system metrics from monitored servers.
3. **Node Exporter** – Runs on target servers and exposes system metrics.
4. **Grafana** – Connects to Prometheus to visualize monitoring data.
5. **Python Script** – Scrapes logs and stores them in an SQL database.
6. **MySQL Database** – Stores logs collected by the Python script.
7. **Alerting System** – Prometheus Alertmanager sends alerts for critical events.

Architecture Diagram: *(Attach the updated architecture image here)*

3. Step-by-Step Implementation

Step 1: Setup Ansible for Automation

1. Install Ansible on the control node:

2. Define the inventory file (`inventory.ini`):

Step 2: Install Python and MySQL Using Ansible

1. Create the Ansible Playbook (`install_python_mysql.yml`) to install Python and MySQL.

2. Run the playbook:

Step 3: Deploy Prometheus with Ansible

1. Create the Ansible Playbook (`prometheus.yml`) to install and configure Prometheus.

2. Run the playbook:

Step 4: Install Node Exporter on Target Servers Using Ansible

1. Create the Ansible Playbook (`node_exporter.yml`) to install Node Exporter.

2. Run the playbook:

Step 5: Configure Prometheus to Scrape Node Exporter

1. Modify the Prometheus configuration file (`prometheus.yml`) to add target servers.

2. Restart Prometheus:

Step 6: Install and Configure Grafana

1. Create the Ansible Playbook (`grafana.yml`) to install Grafana.

2. Run the playbook:

3. Access Grafana.

Step 7: Implement Log Scraping Using Python

1. Write a Python script (`log_scraper.py`) to collect logs and store them in MySQL.

2. Schedule the script using cron:

Step 8: Configure Alerting in Prometheus

1. Define alert rules in `alert.rules.yml`.

2. Update Prometheus config and restart the service.

4. Expected Outcome

✅ Automated deployment of monitoring tools.
✅ Live visualization of system metrics.
✅ Real-time alerting for system issues.
✅ Scalability to monitor additional servers effortlessly.
✅ Automated log collection and storage in MySQL.
✅ Log analysis and troubleshooting support.

5. Conclusion

This project successfully automates IT infrastructure monitoring and log management using Ansible, Prometheus, Grafana, Python, and MySQL, ensuring real-time visibility, proactive alerting, efficient log storage, and improved operational efficiency