

Project Report

Project Name

Local DevOps Sandbox for Monitoring & Alerting (All-in-One VM)

Abstract

This project aims to deliver a fully functional sandbox environment that brings together monitoring and alerting capabilities in a single portable Virtual Machine (VM). In DevOps practices, monitoring forms the foundation of reliability, availability, and performance management.

However, setting up and configuring monitoring tools individually can be time-consuming and error-prone. To address this, the sandbox integrates widely used open-source monitoring technologies such as Prometheus, Grafana, Node Exporter, and Alert manager, pre-configured to provide system-level metrics and proactive alerts. By automating the provisioning process through Virtualmachine and shell scripts, this sandbox reduces manual setup overhead and accelerates the learning curve for DevOps practitioners. The environment helps users understand real-world monitoring workflows such as metric collection, visualization, and alerting, making it an ideal tool for training and experimentation.

Introduction

DevOps is not just about continuous integration and deployment but also about ensuring that deployed systems remain healthy and reliable. Monitoring tools play a crucial role in this ecosystem, enabling teams to gain insights into system performance, detect anomalies, and trigger alerts for potential issues before they impact end-users. The Local DevOps Sandbox project is designed to provide a hands-on environment that replicates a realistic monitoring stack inside a single VM. Learners, professionals, and researchers can use this sandbox to practice configurations, explore monitoring dashboards, and simulate scenarios such as CPU overload, disk failures, or service downtime. By combining virtualization and automation, the sandbox is lightweight, reproducible, and easy to distribute. This project also encourages self-paced learning by providing a safe testing ground without impacting production systems.

Tools Used

1. **Virtual machine** – Automates VM creation and provisioning. It ensures the environment can be rebuilt consistently with minimal effort.
2. **Shell Scripts** – Used in conjunction with Virtual machine to install and configure the monitoring tools automatically.
3. **Prometheus** – Responsible for scraping metrics from Node Exporter and storing time-series data. It also evaluates alert rules defined for system health.
4. **Grafana** – A visualization platform that connects to Prometheus and provides intuitive dashboards for CPU usage, memory consumption, disk activity, and service availability.
5. **Node Exporter** – A Prometheus exporter specifically designed to expose machine metrics such as CPU load, memory usage, disk statistics, and network performance.

6. **Alert manager** – Works alongside Prometheus to handle alert notifications. It supports grouping, inhibition, and routing of alerts to email, Slack, or other channels.

Steps Involved in Building the Project

1. **Set up prerequisites:** Install VirtualBox and Virtual machine on the host machine to enable virtualization and automation.
2. **Virtual machine file creation:** Write a configuration file (Virtual machine file) specifying VM resources such as CPU, RAM, and networking.
3. **Provisioning with automation:** Use Virtualmachine provisioning scripts (bash or shell) to install Prometheus, Grafana, Node Exporter, and Alert manager automatically when the VM boots.
4. **Prometheus configuration:** Define scrape targets for Node Exporter and configure jobs to collect metrics at regular intervals.
5. **Grafana setup:** Import Prometheus as a data source in Grafana and create dashboards for CPU, memory, disk, and system-level metrics.
6. **Alert rules:** Write Prometheus alerting rules for conditions such as high CPU utilization (e.g., above 80%), low disk space (e.g., less than 10% free), and service health failures.
7. **Alert manager integration:** Configure Alert manager to send alerts via email or chat applications. Ensure alerts are routed and formatted appropriately.
8. **Validation and testing:** Stress test the VM (e.g., simulate CPU spikes) to verify that Prometheus captures metrics, Grafana displays them, and Alert manager triggers alerts.
9. **Documentation and reusability:** Package the setup with a README guide, allowing others to reproduce the sandbox easily for training or testing.

Conclusion

The Local DevOps Sandbox for Monitoring & Alerting successfully brings together virtualization, automation, and open-source monitoring tools to deliver a powerful learning and experimentation platform. By automating the installation and configuration process, the sandbox minimizes setup complexity and ensures reproducibility. It provides learners with practical exposure to Prometheus, Grafana, Node Exporter, and Alert manager while simulating real-world scenarios of monitoring and alerting. This environment empowers DevOps practitioners to explore concepts such as proactive monitoring, dashboard creation, and incident response in a safe and controlled setting. Ultimately, the sandbox acts as a bridge between theoretical knowledge and practical implementation, making it an invaluable resource for DevOps training and self-paced learning.