Project Report: Self-Healing Infrastructure with Prometheus, Alertmanager & Ansible

Introduction

This project demonstrates the design and implementation of a self-healing infrastructure that can automatically detect and recover from service failures using monitoring and automation tools. The goal is to reduce downtime and ensure system reliability without manual intervention.

Abstract

The self-healing system integrates Prometheus for monitoring, Alertmanager for alert handling, and Ansible for automation. When a failure or threshold breach (e.g., service down, high CPU usage) is detected, an automated recovery action is triggered. This ensures continuous availability of services such as NGINX by restarting them or executing corrective actions.

Tools Used

- Prometheus: Monitors system metrics and service availability.
- Alertmanager: Handles alerts and triggers actions.
- Ansible: Executes automation playbooks for recovery.
- Shell Scripting: Supports integration and custom commands.
- Docker/Ubuntu VM: Provides isolated environments for services.

Steps Involved in Building the Project

1. Deploy a sample service (NGINX) on Docker/VM.
2. Configure Prometheus to monitor system metrics and service uptime.
3. Set thresholds for alerts (e.g., service down, CPU usage > 90%).
4. Configure Alertmanager to trigger actions on alerts.
5. Create Ansible playbooks to restart failed services automatically.
6. Validate self-healing by simulating failures and observing recovery.

Conclusion

The self-healing infrastructure project highlights how automation and monitoring can reduce downtime, increase reliability, and minimize the need for manual interventions. By integrating Prometheus, Alertmanager, and Ansible, organizations can achieve resilient and fault-tolerant systems effectively.