

Self-Healing Infrastructure with Prometheus, Alertmanager & Ansible

Introduction

In modern IT environments, system availability and reliability are critical. Self-healing infrastructure automatically detects failures and recovers services without manual intervention. This project implements such an infrastructure using open-source tools to monitor, alert, and automate recovery.

Abstract

The project demonstrates an automated system that monitors a service's health (like NGINX) using Prometheus, triggers alerts via Alertmanager when predefined thresholds are breached, and automatically executes Ansible playbooks to recover failed services. This integration reduces downtime and manual troubleshooting.

Tools Used

- **Prometheus:** Collects metrics and monitors service uptime and resource usage.
- **Alertmanager:** Processes alerts from Prometheus and routes them to a webhook.
- **Flask (Webhook server):** Receives alerts and triggers Ansible automation.

- **Ansible:** Automates recovery steps such as restarting services or rebooting systems.
- **Ubuntu VM / Docker:** Environment for deploying services and tools.
- **Shell Scripting:** Used in automation workflows.

Steps Involved in Building the Project

1.
Deploy Sample Service: Installed and ran NGINX as the monitored service.
2.
Configure Prometheus: Setup to scrape metrics from NGINX and system resources. Defined alert rules (e.g., service down, CPU > 90%).
3.
Setup Alertmanager: Configured to send alerts to a custom Flask webhook when thresholds are breached.
4.
Develop Webhook Server: Built a Flask app to receive alerts and invoke Ansible playbooks.

5.

Write Ansible Playbooks: Created playbooks to restart services or reboot the server based on alert type.

6.

Testing & Validation: Simulated failures to verify alerts were sent and automated recovery executed successfully. Captured logs and screenshots as evidence.

Conclusion

This project successfully implemented a self-healing infrastructure using Prometheus, Alertmanager, and Ansible. It enables real-time detection and automated recovery from service failures, significantly improving system resilience and reducing downtime.