

# **Self-Healing Infrastructure Project**

Using Prometheus, Alertmanager, and Ansible

**Done by: Mohamed Askhaf B.Tech.IT**

# Self-Healing Infrastructure using Prometheus, Alertmanager, and Ansible

## Introduction

In modern IT infrastructures, ensuring high availability and fault tolerance is critical. Even a few minutes of downtime can result in significant disruptions and revenue loss. To address this, organizations implement self-healing systems that automatically detect failures and recover without manual intervention. This project demonstrates a self-healing mechanism for an Nginx web server using Prometheus for monitoring, Alertmanager for alerting, and Ansible for automated remediation.

## Abstract

This project focuses on building an automated monitoring and healing solution. The setup continuously monitors the Nginx server's health using Prometheus and Blackbox Exporter. If the server goes down, Alertmanager triggers an alert, which is forwarded to a Flask-based webhook service. The webhook then executes an Ansible playbook that automatically restarts the Nginx service. This ensures that the system recovers immediately without manual intervention, thereby reducing downtime and improving system reliability.

## Tools Used

- Prometheus – For monitoring metrics and scraping data from exporters.
- Alertmanager – For managing alerts and routing them to the webhook service.
- Node Exporter – For exposing system-level metrics such as CPU, memory, and disk usage.
- Blackbox Exporter – For probing endpoints and monitoring service availability.
- Flask – A lightweight Python framework to build the webhook service that triggers automated actions.
- Ansible – For automating the remediation process (restarting Nginx service).
- Nginx – The target service monitored and restarted in case of failures.

## Steps Involved in Building the Project

- 1 Installed and configured **Prometheus** to collect metrics from system exporters and service endpoints.
- 2 Configured **Node Exporter** and **Blackbox Exporter** for system health and endpoint monitoring.

- 3 Defined custom alerting rules in Prometheus to detect Nginx downtime and high CPU usage.
- 4 Set up **Alertmanager** to receive alerts and forward them to a webhook service.
- 5 Developed a **Flask-based webhook** to receive alerts and trigger remediation actions.
- 6 Created an **Ansible playbook** to restart the Nginx service upon receiving an alert.
- 7 Tested the system by manually stopping the Nginx service and observing the automated recovery process.

## Conclusion

This project successfully demonstrates the concept of self-healing infrastructure using open-source monitoring and automation tools. The integration of Prometheus, Alertmanager, Flask, and Ansible ensures that service downtime is detected and resolved automatically. Such approaches are highly valuable for real-world IT operations, as they improve system reliability, reduce the need for manual intervention, and ensure business continuity. Future improvements could include integrating notification channels (Slack, Email) and expanding the system to manage multiple services beyond Nginx.