Resource Insight Document

Nieben van Sint Annaland
06-19-2024

Overview

This document provides an in-depth look into the resource usage and monitoring of my application hosted on Azure. This insight will highlight the current state of the infrastructure, resource utilization, and the proof of active monitoring to ensure optimal performance and reliability

Resource Overview

The application is structured as a collection of microserivces, each deployed in a cluster of Kubernetes (AKS) to ensure scalability, maintainability, and resilience. In combination, I have created an Azure Function to handle notifications within my application.

Resource Utilization

Azure Kubernetes Service (AKS)

- **Cluster Name:** tuneturtle-aks
- **Node Count:** 1 node pool, Standard_DS2_v2
- **Kubernetes Version:** 1.28.9
- **CPU Utilization (Max):** 25%
- **Node pool network in (max):** 6.9 MB
- **Node pool network out (max):** 411.9MB
- **Node pool disk bytes (max):** 27 GB
- **Node pool memory working set (max):** 88%

Function App

- **Name:** notification-function-20240610094131599
- **Service Plan:** ASP-tuneturtleaksgroup-94c5 (Y1: 0)
- **Functions:**
    - **addNotification:** (HTTP Trigger)
    - **notifications:** (HTTP Trigger)
- **Memory working set:** 0B
- **Function execution count:** 0
- **MB Milliseconds:** 0

Monitoring and Alerts

Not only have I implemented logging and alerts systems like Application Insights, I have also implemented Grafana and Prometheus to ensure proper logging and monitoring of my application.

Application Insights

- **Tracked Metrics**
    - Response times of each microservice
    - Dependency call durations
    - Exception rates
    - Custom events for business logic tracking

Grafana + Prometheus

- **Tracked Metrics**
    - CPU Usage
    - Memory Usage
    - Disk Usage
    - Network Traffic
    - Response Times
    - Error Rates
    - Throughput
    - Custom Business Metrics
    - Pod Status
    - Node Health

Conclusion

The applications infrastructure and resource usage are being actively monitored through Azure's comprehensive monitoring tools combined with my own implementation of Grafana and Prometheus. The metrics, alerts, and logs collected provide us with valuable insights into the performance and health of the microservices, ensuring that I can proactively address any issues and maintain a high level of service availalblilty and performance.

By continuously monitoring these resources, we can optimize the application's performance, scale effectively, and quickly respond to any anomalies, thereby delivering reliable and robust service to my users