Day 27 – Kubernetes Monitoring

Prometheus + Grafana

Introduction

- Monitoring is crucial in Kubernetes for:
- - Observability of system health
- - Performance tuning
- - Debugging and alerts

- Today, we will install Prometheus and Grafana.

What is Prometheus?

- - An open-source monitoring and alerting toolkit
- - Pull-based metrics collection
- - Stores time-series data
- - Exposes UI and APIs at port 9090

What is Grafana?

- - Visualization and analytics tool
- - Supports data sources like Prometheus, InfluxDB, etc.
- - Dashboards to visualize time-series data
- - Runs on port 3000

Folder Structure (monitoring/)

- - prometheus-config-map.yaml
- - prometheus-deployment.yaml
- - prometheus-service.yaml
- - grafana-deployment.yaml
- - grafana-service.yaml

Prometheus Setup

- 1. Create a ConfigMap for prometheus.yml
- 2. Deploy Prometheus using a Deployment
- 3. Expose it using a NodePort or LoadBalancer Service
- 4. Access it using EC2 Public IP and NodePort

Grafana Setup

- 1. Deploy Grafana using grafana/grafana image
- 2. Expose using NodePort or LoadBalancer
- 3. Default Login: admin / admin
- 4. Add Prometheus as a data source

Access & Port Forwarding

- - NodePort: http://<EC2-PUBLIC-IP>:<nodeport>
- - Or use SSH Tunnel:
- ssh -i your-key.pem -L 30090:localhost:30090 ubuntu@<EC2-IP>
- - Port-forwarding: kubectl port-forward svc/prometheus-service 30090:9090

Common Issues

- - ImagePullBackOff: Check disk space
- - Port not reachable: Ensure SG allows access
- - Empty dashboards: Check Prometheus targets
- - Prometheus not collecting metrics: Validate configMap

Summary

- - Prometheus scrapes and stores metrics
- - Grafana visualizes the metrics
- - Both were deployed without Helm
- - Covered deployment, configuration, and access