

Kubernetes Monitoring on Minikube using Prometheus, Grafana & New Relic

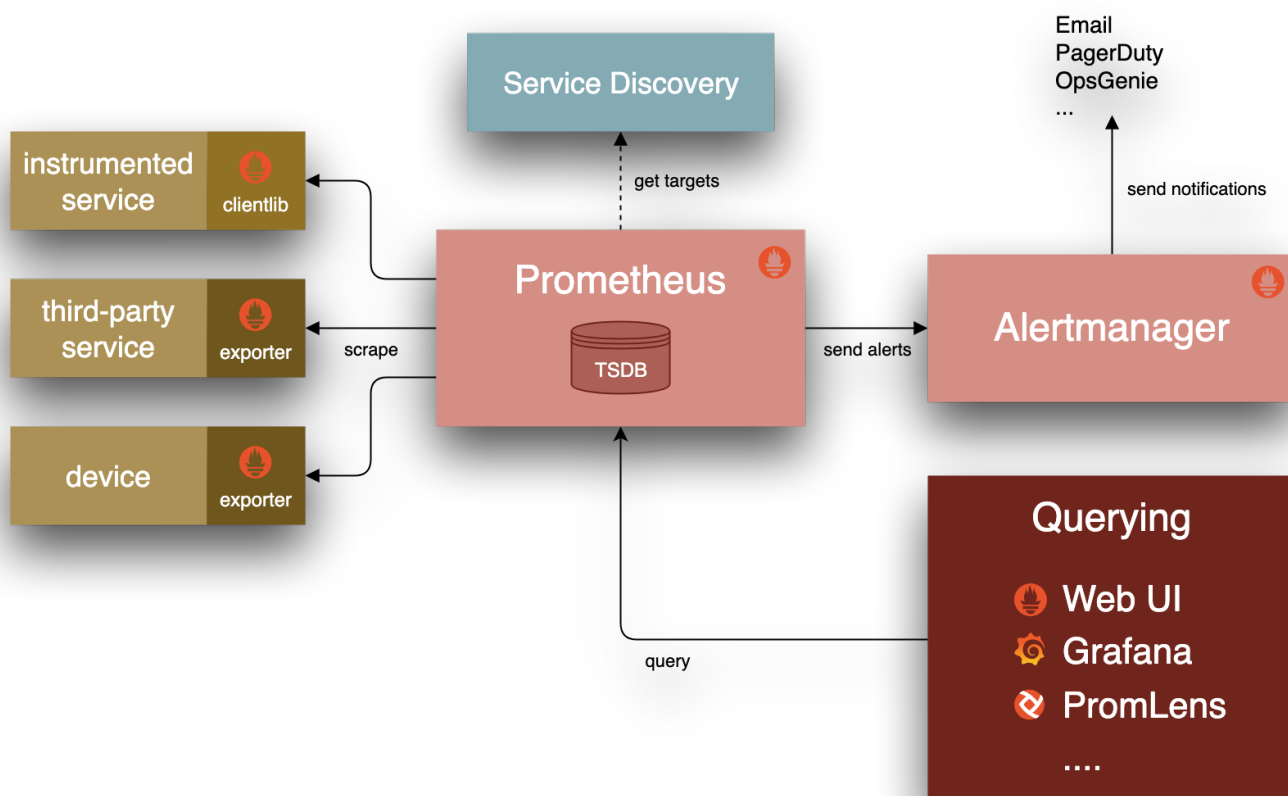
Project Overview

This project showcases how to set up monitoring stack using Newrelic, Prometheus and Grafana on a local Kubernetes cluster created with Minikube(using Docker Desktop), Installing monitoring tools using Helm.

Prerequisites

1. Minikube
2. Docker Desktop
3. Minikube
4. Kubectl (kubernetes CLI)
5. Helm
6. Newrelic account

Prometheus & Grafana:



source: web

Prometheus

Step-by-Step Setup

1. Create a Minikube Cluster (linux/macOS - Docker Driver)

minikube start --driver=docker

```
[akhilrao@Akhils-MacBook-Air prom % minikube start
🐳 minikube v1.35.0 on Darwin 15.5 (arm64)
🌟 Automatically selected the docker driver
🔧 Using Docker Desktop driver with root privileges
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🚚 Pulling base image v0.0.46 ...
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
🐳 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  ▪ Generating certificates and keys ...
  ▪ Booting up control plane ...
  ▪ Configuring RBAC rules ...
🔗 Configuring bridge CNI (Container Networking Interface) ...
🔍 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🏠 Done! kubectl is now configured to use "minikube" cluster and "default" name
space by default
[akhilrao@Akhils-MacBook-Air prom % kubectl get pods
No resources found in default namespace.
[akhilrao@Akhils-MacBook-Air prom % kubectl get pods -a
error: unknown shorthand flag: 'a' in -a
See 'kubectl get --help' for usage.
[akhilrao@Akhils-MacBook-Air prom % kubectl get pods -A
NAMESPACE      NAME                                     READY   STATUS    RESTARTS   AG
E
kube-system     coredns-668d6bf9bc-gks4w              1/1     Running   0           88
s
kube-system     etcd-minikube                          1/1     Running   0           94
s
kube-system     kube-apiserver-minikube                1/1     Running   0           94
s
kube-system     kube-controller-manager-minikube       1/1     Running   0           94
s
kube-system     kube-proxy-hn427                       1/1     Running   0           88
s
kube-system     kube-scheduler-minikube                1/1     Running   0           94
s
kube-system     storage-provisioner                    1/1     Running   0           92
s
```

2. Add the Prometheus Helm Chart & Install

```
helm repo add prometheus-community https://prometheus-  
community.github.io/helm-charts  
helm repo update
```

```
helm install prometheus prometheus-community/prometheus
```

```
akhilrao@Akhils-MacBook-Air prom % helm repo add prometheus-community https://prometheus-community.github.io/helm-charts  
helm repo update  
"prometheus-community" already exists with the same configuration, skipping  
Hang tight while we grab the latest from your chart repositories...  
...Successfully got an update from the "prometheus-community" chart repository  
Update Complete. ★Happy Helming!★
```

```
akhilrao@Akhils-MacBook-Air prom % helm repo update  
Hang tight while we grab the latest from your chart repositories...  
...Successfully got an update from the "prometheus-community" chart repository  
Update Complete. ★Happy Helming!★  
akhilrao@Akhils-MacBook-Air prom % helm install prometheus prometheus-community/prometheus  
NAME: prometheus  
LAST DEPLOYED: Sun Jun  8 15:04:50 2025  
NAMESPACE: default  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None  
NOTES:  
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:  
prometheus-server.default.svc.cluster.local  
  
Get the Prometheus server URL by running these commands in the same shell:  
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=prometheus,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")  
kubectl --namespace default port-forward $POD_NAME 9090  
  
The Prometheus alertmanager can be accessed via port 9093 on the following DNS name from within your cluster:  
prometheus-alertmanager.default.svc.cluster.local  
  
Get the Alertmanager URL by running these commands in the same shell:  
export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=alertmanager,app.kubernetes.io/instance=prometheus" -o jsonpath="{.items[0].metadata.name}")  
kubectl --namespace default port-forward $POD_NAME 9093  
#####  
##### WARNING: Pod Security Policy has been disabled by default since #####  
##### it deprecated after k8s 1.25+. use #####  
##### (index .Values "prometheus-node-exporter" "rbac" #####  
##### "pspEnabled") with (index .Values #####  
##### "prometheus-node-exporter" "rbac" "pspAnnotations") #####  
##### in case you still need it. #####  
#####  
[  
  
The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:  
prometheus-prometheus-pushgateway.default.svc.cluster.local  
  
Get the PushGateway URL by running these commands in the same shell:  
export POD_NAME=$(kubectl get pods --namespace default -l "app=prometheus-pushgateway,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")  
kubectl --namespace default port-forward $POD_NAME 9091
```

3. Expose Prometheus via NodePort to Access in Browser

```
kubectl expose service prometheus-server --type=NodePort --target-  
port=9090 --name=prometheus-server-ext
```

Changes made:

type: ClusterIP

To:

type: NodePort

Then:

minikube ip