

Prometheus & Grafana Setup In K8's Cluster Using Helm

About Prometheus

Prometheus is an opensource **monitoring and alerting toolkit**. Prometheus consists of several components some of which are listed below:

- The Prometheus server which scrapes(collects) and stores time series data based on a pull mechanism.
- A rules engine which allows generation of Alerts based on the scraped metrics.
- An alertmanager for handling alerts.
- Multiple integrations for graphing and dashboarding.




Enterprises looking **to decrease downtime and optimize resources** can implement server monitoring using tools like Prometheus and Grafana.

What Is Server Monitoring?

Server monitoring is a way to look into what your servers are doing in real time.

It can provide you with actionable data, and is most often used for troubleshooting and capacity planning. It's typically done at three levels:

- 
- Network (e.g., traffic, bandwidth, latency).
 - Machine (e.g., CPU and memory utilization and storage).
 - Application (e.g., rate of user commands, locks, large syncs, commits/submits, etc.).

Why Monitor Your Servers?



Server monitoring has a lot of benefits. The biggest benefit is avoiding reactive panics. You can get a head start on issues occurring in your servers or applications before your users are impacted. Along with this benefit, you can also use monitoring to figure out how to:

- Increase uptime.
- Improve hardware and software performance.
- Plan for the future by making the best use of your resources.

Implementing monitoring can give you an almost immediate return on investment. You can observe trends, spikes, and anomalies that may indicate a problem. And then you can drill down to discover the root cause of the issue.

What Is Prometheus Monitoring?

Prometheus is an open source system that collects and manages server and application metrics. It can also be configured to notify your team when an issue arises.

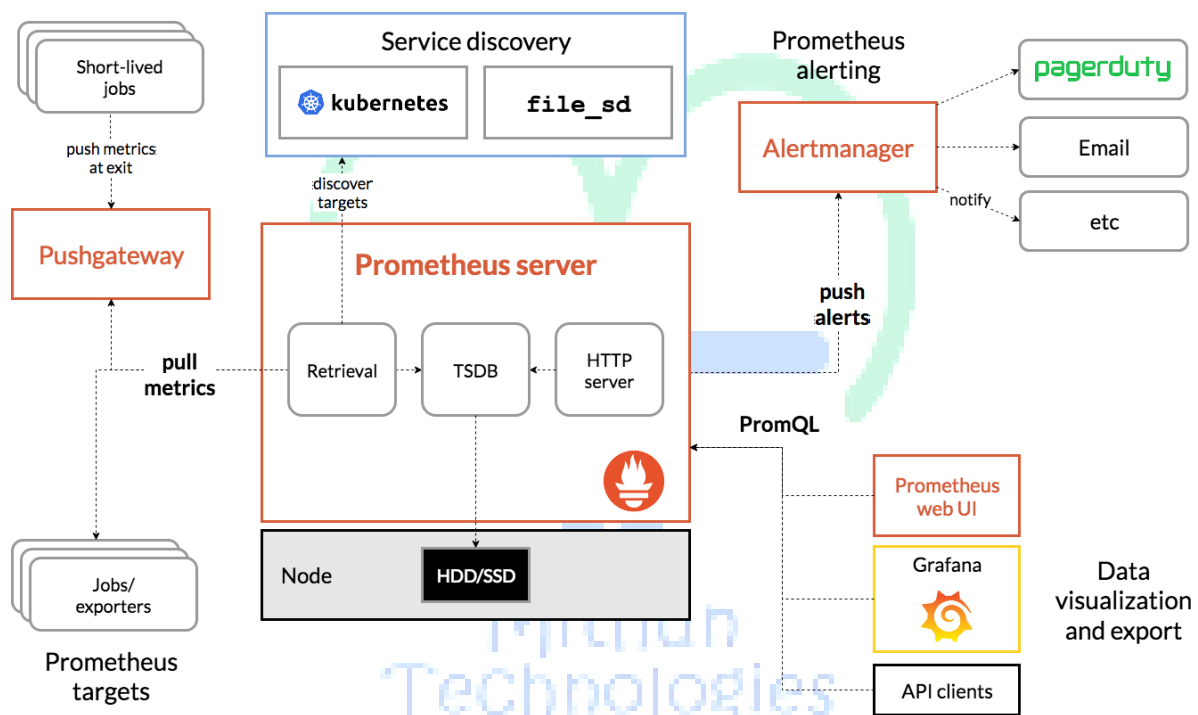
What Is Grafana?

Grafana is an open source tool that allows you to easily visualize information.

What Are Grafana Dashboards?

Grafana dashboards take information from server monitoring tool like Prometheus to display this information.

Architecture



alteringFiles.yml

rules

Prometheus:

Prometheus® is an open source monitoring system developed by engineers at SoundCloud in 2012. In 2016, Prometheus was the second project accepted into the Cloud Native Computing Foundation after Kubernetes.

The Prometheus monitoring system includes a rich, multidimensional data model, a concise and powerful query language called PromQL, an efficient embedded timeseries database.

Alert Manager:

The Alertmanager handles alerts sent by client applications such as the Prometheus server. It takes care of deduplicating, grouping, and routing them to the correct receiver integration such as email, PagerDuty, or OpsGenie. It also takes care of silencing and inhibition of alerts.

Node Exporter:

The Prometheus Node Exporter exposes a wide variety of hardware- and kernel-related metrics. That will periodically (every 1 second) gather all the metrics of your system. It will monitor your filesystems, disks, CPUs, memory but also your network statistics,

Kube State Metrics:

kube-state-metrics is a service that listens to the Kubernetes API server and generates metrics about the state of the objects, including deployments, nodes, and pods.

Grafana:

Grafana is designed for analyzing and visualizing metrics such as system CPU, memory, disk and I/O utilization.

Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored. Create, explore, and share dashboards with your team and foster a data driven culture:

Prerequisite:

- 1) Kubernetes Cluster With Storage Class Configured(Dynamic Provisioning).

If Storage Class is not Configured then make persistence enabled as false in helm values files # Which is not suggestable as if pods are terminated and recreated we will lost the data.

- 2) Kubernetes nodes with minimum 4GB RAM With 2 Core Processor.
- 3) Server which has kubectl & Helm Configured.

Deploy Prometheus ,Exporters ,Alert Manager & Grafana

- 1) Add helm repo which contains Prometheus and Grafana charts.

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

```
helm repo add grafana https://grafana.github.io/helm-charts
```

- 2) Create a namespace in which will deploy Prometheus & it's components.

```
kubectl create ns monitoring
```

- 3) Create prometheusvalues.yml file from below link. **And update this file to define alter rules & smtp details.**

<https://raw.githubusercontent.com/MithunTechnologiesDevOps/Kubernetes-Manifests/master/prometheusvalues.yml>

- 4) Deploy using helm with prometheusvalues.yml values file which u create in step 3.

Deploy helm chart with default values . And Create Ingress rules to access.

```
helm install prometheus prometheus-community/prometheus -n monitoring --f prometheusvalues.yml
```

If you have ingress deployed deploy by change service type as node port as below.
And access using NODEIP:NODEPORT

```
helm install --set alertmanager.service.type=NodePort --set server.service.type=NodePort prometheus prometheus-community/prometheus -n monitoring -f prometheusvalues.yml
```

Get NodePorts for alertmanager & Prometheus

```
kubectl get svc -n monitoring
```

Access Using Prometheus & Alert Manager using NODEIP:NODEPORT

- 5) Deploy Grafana using Helm

Deploy helm chart with default values . And Create Ingress rules to access.

```
helm install grafana grafana/grafana -n monitoring
```

If you have ingress deployed deploy by change service type as node port as below.
And access using NODEIP:NODEPORT

```
helm install --set service.type=NodePort grafana grafana/grafana -n monitoring
```

Access Grafana Using NodeIP & NodePort If Service type is Node Port or accessing using Load Balancer if service type is load balancer.

- 6) Get Grafana default password to login. Default user name is admin.

```
kubectl get secret --namespace monitoring grafana -o  
jsonpath="{.data.admin-password}" | base64 --decode ; echo
```

- 7) Configure Data Source Service URL(Below URL) in Grafana so that it use this data source to fetch metrics.

```
http://prometheus-server
```

- 8) import some sample dashboards from community find few ids below.

Sample Dash Board IDS: 3119,7249 8919,6417 ,11074, 1860, 13332

