

# Metrics and Insight of your Cluster with Prometheus

Prateek Sahu

[prateeks@utexas.edu](mailto:prateeks@utexas.edu)

# What is Prometheus?

Prometheus is a systems and service monitoring system

- ▶ Collects metrics from configured targets at given intervals
- ▶ Evaluates rule expressions
- ▶ Displays the results as charts
- ▶ Can trigger alerts if some condition is observed to be true
- ▶ Written entirely in Go

# Prometheus Architecture

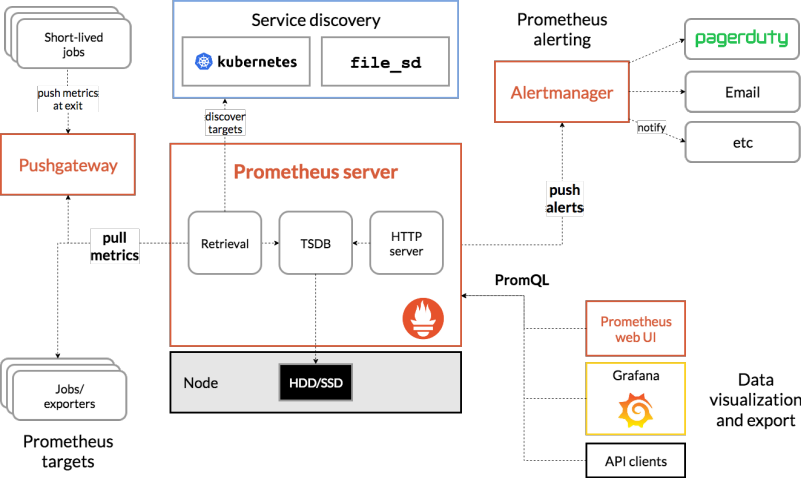


Figure 1: Prometheus Architecture

# Prometheus Features

Prometheus' main distinguishing features as compared to other monitoring systems are:

- ▶ A **multi-dimensional** data model (time-series defined by metric name and set of key/value dimensions).
- ▶ PromQL, a **flexible query language** to leverage this dimensionality.
- ▶ No dependency on distributed storage; **single server nodes are autonomous**.
- ▶ Time-series collection happens via a **pull model** over HTTP. Pushing time-series is supported via an intermediary gateway.
- ▶ Targets are discovered via **service discovery** or **static configuration**.
- ▶ Support for hierarchical and horizontal **federation**.

# Containers' Implications for Monitoring

- ▶ Designed for reliability rather than accuracy
  - ▶ Go-to system during an outage to allow you to quickly diagnose problems.
  - ▶ Standalone and does not depend on rest of the network/remote services.
  - ▶ Can always view what statistics are available about your system, even under failure conditions.

# Prometheus and Grafana

- ▶ Drawbacks of Prometheus
  - ▶ Since it is designed for reliability rather than accuracy it is not a good tool to be used for per-request billings
  - ▶ Charting tools are sub-par and openly adopt Grafana services for charting and dashboard
- ▶ Grafana
  - ▶ Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored
  - ▶ Supports multi-tenancy and integrates well with auth services like Google Auth,LDAP, github.
  - ▶ Boasts of a plethora of visualizations: heatmaps, histograms, geomaps

# Demo

- ▶ Enable: `microk8s.enable prometheus`
- ▶ Check: `microk8s.kubectl get pods -n monitoring`
- ▶ Port-forward services:
  - ▶ `microk8s.kubectl port-forward -n monitoring prometheus-k8s-0 9090`
  - ▶ `microk8s.kubectl port-forward -n monitoring (microk8s.kubectl get pods --selector=app=grafana -n monitoring --output=jsonpath="{.items..metadata.name}") 3000`
  - ▶ `microk8s.kubectl port-forward -n monitoring alertmanager-main-0 9093`

## Demo commands

- ▶ `microk8s.kubectl -n monitoring edit service prometheus-k8s`
  - ▶ grafana and alertmanager-main
- ▶ Service Monitors
  - ▶ Examples at </snap/microk8s/920/actions/prometheus/>