PROMETHEUS (like InfluxDB) AND GRAFANA:

* Prometheus is a monitoring solution for storing time series data like metrics. Grafana allows to visualize the data stored in Prometheus (and other sources).
* (Prometheus collects rich metrics and provides a powerful querying language; Grafana transforms metrics into meaningful visualizations.
* Both are compatible with many, if not most, data source types. In fact, it is very common for DevOps teams to run Grafana on top of Prometheus.)

## (Prometheus and Grafana are both built for time-series data. Influx DB is a push-based system. It requires an application to actively push data into InfluxDB.

## The process of collecting metrics is called scraping.

* Prometheus is a pull-based system. An application publishes the metrics at a given endpoint, and Prometheus fetches them periodically.
* Prometheus and Grafana are both built for time-series data. Prometheus excels in metric data collection, whereas Grafana champions metric visualizations.
* Both tools are open source, free and have vibrant communities of open source developers supporting their development.

Click here to get k8s, Grafana, Prometheus scrip [click here](#_top)

PROMETHEUS:

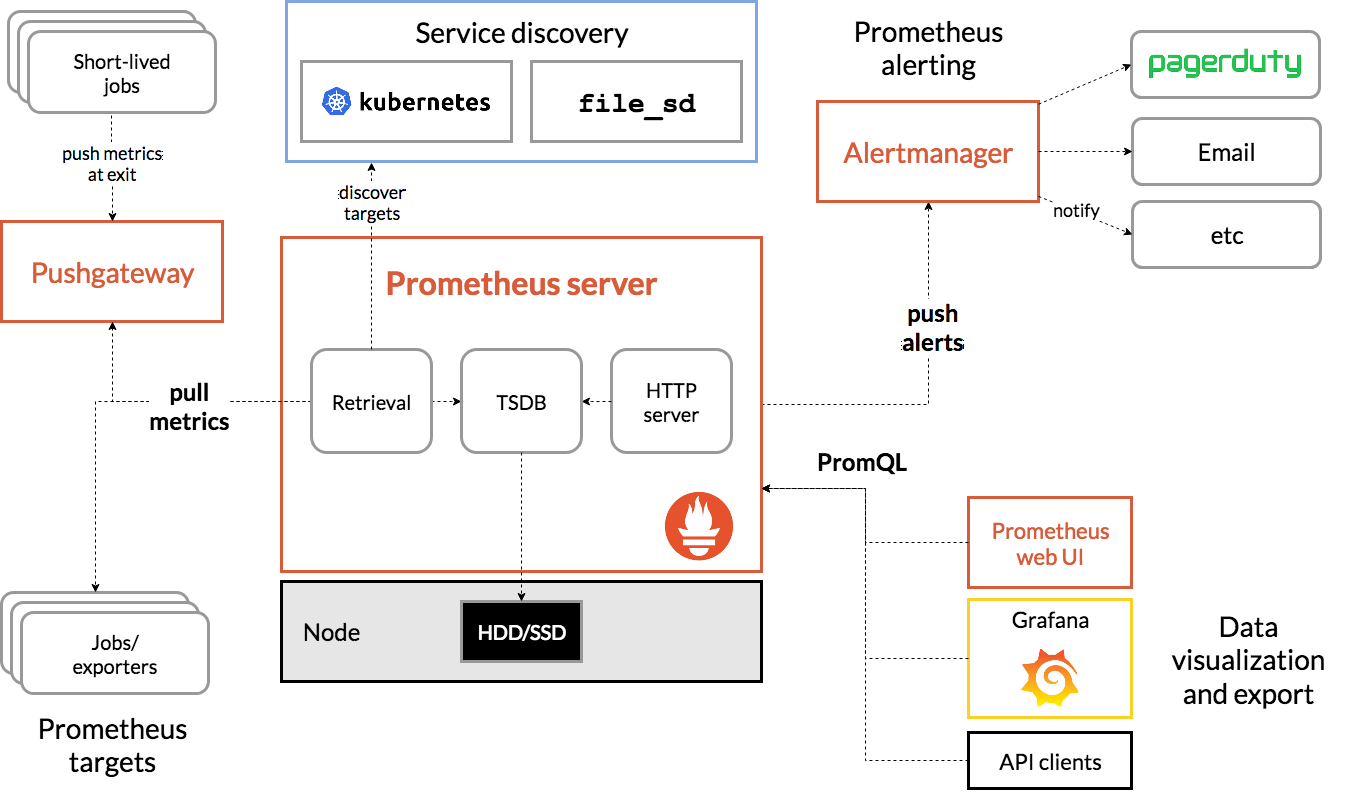
* Prometheus is an open-source technology designed to provide monitoring and alerting functionality for cloud-native environments, including Kubernetes. It can collect and store metrics as time-series data, recording information with a timestamp. It can also collect and record labels, which are optional key-value pairs.
* Prometheus is typically used to collect numeric metrics from services that run 24/7 and allow metric data to be accessed via HTTP endpoints.
* This can be done manually or with various client libraries. Prometheus exposes data using a simple format, with a new line for each metric, separated with line feed characters
* Prometheus is a free software application used for event monitoring and alerting. It records real-time metrics in a time series database (allowing for high dimensionality) built using a HTTP pull model, with flexible queries and real-time alerting

The following are some of Prometheus' key features:

([https://www.atatus.com/glossary/prometheus-monitoring/](https://www.atatus.com/glossary/prometheus-monitoring/)))

* **Multidimensional Data Model**  
  using time-series data with metric names and key-value pairs as identifiers.
* **PromQL**  
  A querying language with a multi-dimensional data model can be used.
* **Pull Model**  
  By actively "grabbing" data through HTTP, Prometheus may collect time-series data.
* **No Reliance on Distributed Storage**  
  All single server nodes are self-contained.
* **Pushing Time-series Data**  
  Through the usage of an intermediary gateway, this service is available.
* **Monitoring Target Discovery**  
  Static configuration or service discovery are both options.
* **Visualization**  
  Prometheus has a variety of graphs and dashboards to choose from.\*\*

ARCHITECTURE:



Main Components in Prometheus architecture:

* Prometheus Server
* Push Gateway
* Alert Manager
* Prometheus Targets
* Client Libraries
* Prometheus Exporters
* Service Discovery
* Key Takeaways

# Is Prometheus a pull based system

Does Prometheus delete old data?

**Data will be automatically deleted after the storage retention time has passed**. By default it is 15 days. If you want to delete specific data earlier, then you are able. You need to enable the admin api in Prometheus before you can.

Does Prometheus use Kafka?

Kafka is one of the most widely used streaming platforms, and **Prometheus is a popular way to monitor Kafka**. We will use Prometheus to pull metrics from Kafka and then visualize the important metrics on a Grafana dashboard.

* Create a deployment file for Prometheus

What is metric?

Metrics tell you how much of something exists, such as **how much memory a computer system has available or how many centimeters long a desktop is**. In the case of Grafana, metrics are most useful when they are recorded repeatedly over time.

What metrics does Prometheus collect Kubernetes?

Prometheus also collects **application metrics and metrics related to Kubernetes services, orchestration status, and nodes**. Then, a node exporter exposes host-related metrics, like CPU usage, memory, disk space, network, and bandwidth

How does Prometheus collect data Kubernetes?

Prometheus **uses Kubernetes APIs to read all the available metrics from Nodes, Pods, Deployments, etc**. For this reason, we need to create an RBAC policy with read access to required API groups and bind the policy to the monitoring namespace.

Where does Prometheus store data in Kubernetes?

Prometheus stores its on-disk time series data **under the directory specified by the flag storage.** **local.** **path** . The default path is ./data (relative to the working directory), which is good to try something out quickly but most likely not what you want for actual operations.

How does Prometheus scrape Kubernetes metrics?

Prometheus **uses service discovery** to discover targets to scrape. Kubernetes clusters are equipped with labels, annotations, and a mechanism for tracking status and changes for different elements. To discover targets, Prometheus needs to use the Kubernetes API.

What is meant by service discovery?

Service discovery is **the automatic detection of devices and offered services over a network**. Discovery, which minimizes configuration efforts for administrators, is commonly found in microservice architectures and containerization platforms.

How does Prometheus store metric data?

Prometheus fundamentally stores all data as **time series**: streams of timestamped values belonging to the same metric and the same set of labeled dimensions. Besides stored time series, Prometheus may generate temporary derived time series as the result of queries

Can Prometheus monitor multiple Kubernetes clusters?

**Yes, with the Prometheus/Grafana combination, you can monitor multiple Kubernetes clusters and multiple providers**.

Can I use Grafana without Prometheus?

**Grafana can only use data from external systems like Prometheus, MySQL, Azure Monitor and Amazon CloudWatch**. That means Grafana has no means to collect data on its own, through agents or other data pipelines, and is thus dependent on other systems to provide data.

INTIGRATING K8S WITH PROMETHEOUS:

git clone https://github.com/bibinwilson/kubernetes-grafana.git

vi grafana-datasource-config.yaml

kubectl create -f grafana-datasource-config.yaml

vi deployment.yaml

kubectl create -f deployment.yaml

vi service.yaml

kubectl create -f service.yaml

http://<your-node-ip>:32000

kubectl port-forward -n monitoring <Grafana-pod-name> 3000 &

kubectl get po -n monitoring

http://localhost:3000