



☰ Show nav

Overview

What is Prometheus?

[Prometheus](#) is an open-source systems monitoring and alerting toolkit originally built at [SoundCloud](#). Since its inception in 2012, many companies and organizations have adopted Prometheus, and the project has a very active developer and user [community](#). It is now a standalone open source project and maintained independently of any company. To emphasize this, and to clarify the project's [governance](#) structure, Prometheus joined the [Cloud Native Computing Foundation](#) in 2016 as the second hosted project, after [Kubernetes](#).

Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.

For more elaborate overviews of Prometheus, see the resources linked from the [media](#) section.

Features

Prometheus's main features are:

- a multi-dimensional [data model](#) with time series data identified by metric name and key/value pairs
- PromQL, a [flexible query language](#) to leverage this dimensionality
- no reliance 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
- multiple modes of graphing and dashboarding support



Prometheus



Metrics are numerical measurements in layperson terms. The term time series refers to the recording of changes over time. What users want to measure differs from application to application. For a web server, it could be request times; for a database, it could be the number of active connections or active queries, and so on.

Metrics play an important role in understanding why your application is working in a certain way. Let's assume you are running a web application and discover that it is slow. To learn what is happening with your application, you will need some information. For example, when the number of requests is high, the application may become slow. If you have the request count metric, you can determine the cause and increase the number of servers to handle the load.

Components

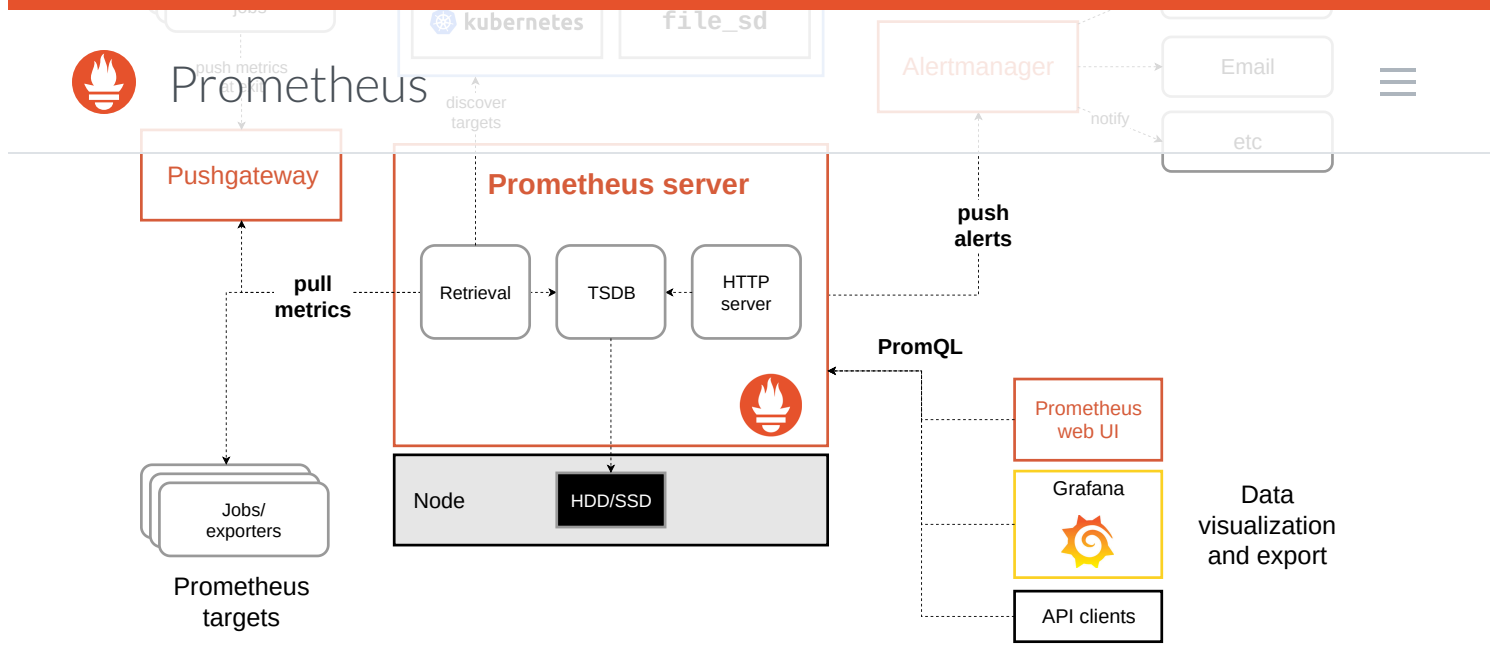
The Prometheus ecosystem consists of multiple components, many of which are optional:

- the main [Prometheus server](#) which scrapes and stores time series data
- [client libraries](#) for instrumenting application code
- a [push gateway](#) for supporting short-lived jobs
- special-purpose [exporters](#) for services like HAProxy, StatsD, Graphite, etc.
- an [alertmanager](#) to handle alerts
- various support tools

Most Prometheus components are written in [Go](#), making them easy to build and deploy as static binaries.

Architecture

This diagram illustrates the architecture of Prometheus and some of its ecosystem components:



Prometheus scrapes metrics from instrumented jobs, either directly or via an intermediary push gateway for short-lived jobs. It stores all scraped samples locally and runs rules over this data to either aggregate and record new time series from existing data or generate alerts. [Grafana](#) or other API consumers can be used to visualize the collected data.

When does it fit?

Prometheus works well for recording any purely numeric time series. It fits both machine-centric monitoring as well as monitoring of highly dynamic service-oriented architectures. In a world of microservices, its support for multi-dimensional data collection and querying is a particular strength.

Prometheus is designed for reliability, to be the system you go to during an outage to allow you to quickly diagnose problems. Each Prometheus server is standalone, not depending on network storage or other remote services. You can rely on it when other parts of your infrastructure are broken, and you do not need to set up extensive infrastructure to use it.

When does it not fit?

Prometheus values reliability. You can always view what statistics are available about your system, even under failure conditions. If you need 100% accuracy, such as for per-request billing, Prometheus is not a good choice, as the collected data will likely not be detailed and



Edit



Issue

Next

First steps



© Prometheus Authors 2014-2026 | All components are available under the [Apache 2 License](#) on [GitHub](#).

© 2026 The Linux Foundation. All rights reserved. The Linux Foundation has registered trademarks and uses trademarks. For a list of trademarks of The Linux Foundation, please see our [Trademark Usage](#) page.