Frequently asked questions

- What is Prometheus?
- How does Prometheus compare against other monitoring systems?
- What dependencies does Prometheus have?
  - Is this cloud native?
- Can Prometheus be made highly available?
- I was told Prometheus “doesn't scale”.
- What language is Prometheus written in?
- How stable are Prometheus features, storage formats, and APIs?
- Why do you pull rather than push?
- How to feed logs into Prometheus?
- Who wrote Prometheus?
- What license is Prometheus released under?
- What is the plural of Prometheus?
- Can I reload Prometheus's configuration?
- Can I send alerts?
- Can I create dashboards?
- Can I change the timezone? Why is everything in UTC?

- Instrumentation
  - Which languages have instrumentation libraries?
  - Can I monitor machines?
  - Can I monitor network devices?
  - Can I monitor batch jobs?
  - What applications can Prometheus monitor out of the box?
  - Can I monitor JVM applications via JMX?
  - What is the performance impact of instrumentation?
- Implementation
  - Why are all sample values 64-bit floats?

What is Prometheus?

Prometheus is an open-source system monitoring and alerting toolkit with an active ecosystem. It is the only system directly supported by Kubernetes and the de facto standard across the cloud native ecosystem. See the overview.

How does Prometheus compare against other monitoring systems?

See the comparison page.

What dependencies does Prometheus have?

The main Prometheus server runs standalone as a single monolithic binary and has no external dependencies.

Is this cloud native?

Yes.

Cloud native is a flexible operating model, breaking up old service boundaries to allow for more flexible and scalable deployments.

Prometheus's service discovery integrates with most tools and clouds. Its dimensional data model and scale into the tens of millions of active series allows it to monitor large cloud-native deployments. There are always trade-offs to make when running services, and Prometheus values reliably getting alerts out to humans above all else.

Can Prometheus be made highly available?

Yes, run identical Prometheus servers on two or more separate machines.

Systems like Thanos can deduplicate the data. Many of the external systems that support remote write can also deduplicate the data.

Identical alerts will be deduplicated by the Alertmanager.

Alertmanager supports high availability by interconnecting multiple Alertmanager instances to build an Alertmanager cluster.

I was told Prometheus “doesn't scale”.

This is often more of a marketing claim than anything else.

A single instance of Prometheus can be more performant than some systems positioning themselves as long term storage solution for Prometheus. You can run Prometheus reliably with tens of millions of active series.

If you need more than that, there are several options. Scaling and Federating Prometheus on the Robust Perception blog is a good starting point, as are the long storage systems listed on our integrations page.

What language is Prometheus written in?

Most Prometheus components are written in Go. Some are also written in Java, Python, and Ruby.

How stable are Prometheus features, storage formats, and APIs?

All repositories in the Prometheus GitHub organization that have reached version 1.0.0 broadly follow semantic versioning. Breaking changes are indicated by increments of the major version. Exceptions are possible for experimental components, which are clearly marked as such in announcements.

Even repositories that have not yet reached version 1.0.0 are, in general, quite stable. We aim for a proper release process and an eventual 1.0.0 release for each repository. In any case, breaking changes will be pointed out in release notes (marked by `[CHANGE]`) or communicated clearly for components that do not have formal releases yet.

Why do you pull rather than push?

Pulling over HTTP offers a number of advantages:

- You can start extra monitoring instances as needed, e.g. on your laptop when developing changes.
- You can more easily and reliably tell if a target is down.
- You can manually go to a target and inspect its health with a web browser.

Overall, we believe that pulling is slightly better than pushing, but it should not be considered a major point when considering a monitoring system.

For cases where you must push, we offer the Pushgateway.

How to feed logs into Prometheus?

Short answer: Don't! Use something like Grafana Loki or OpenSearch instead.

Longer answer: Prometheus is a system to collect and process metrics, not an event logging system. The Grafana blog post Logs and Metrics and Graphs, Oh My! provides more details about the differences between logs and metrics.

If you want to extract Prometheus metrics from application logs, Grafana Loki is designed for just that. See Loki's metric queries documentation.

Who wrote Prometheus?

Prometheus was initially started privately by Matt T. Proud and Julius Volz. The majority of its initial development was sponsored by SoundCloud.

It's now maintained and extended by a wide range of companies and individuals.

What license is Prometheus released under?

Prometheus is released under the Apache 2.0 license.

What is the plural of Prometheus?

After extensive research, it has been determined that the correct plural of 'Prometheus' is 'Prometheis'.

If you cannot remember this, "Prometheus instances" is a good workaround.

Can I reload Prometheus's configuration?

Yes, sending `SIGHUP` to the Prometheus process or an HTTP POST request to the `/-/reload` endpoint will reload and apply the configuration file. The various components attempt to handle failing changes gracefully.

Can I send alerts?

Yes, with the Alertmanager.

We support sending alerts through email, various native integrations, and a webhook system anyone can add integrations to.

Can I create dashboards?

Yes, we recommend Grafana for production usage. There are also Console templates.

Can I change the timezone? Why is everything in UTC?

To avoid any kind of timezone confusion, especially when the so-called daylight saving time is involved, we decided to exclusively use Unix time internally and UTC for display purposes in all components of Prometheus. A carefully done timezone selection could be introduced into the UI. Contributions are welcome. See issue #500 for the current state of this effort.

Instrumentation

Which languages have instrumentation libraries?

There are a number of client libraries for instrumenting your services with Prometheus metrics. See the client libraries documentation for details.

If you are interested in contributing a client library for a new language, see the exposition formats.

Can I monitor machines?

Yes, the Node Exporter exposes an extensive set of machine-level metrics on Linux and other Unix systems such as CPU usage, memory, disk utilization, filesystem fullness, and network bandwidth.

Can I monitor network devices?

Yes, the SNMP Exporter allows monitoring of devices that support SNMP. For industrial networks, there's also a Modbus exporter.

Can I monitor batch jobs?

Yes, using the Pushgateway. See also the best practices for monitoring batch jobs.

What applications can Prometheus monitor out of the box?

See the list of exporters and integrations.

Can I monitor JVM applications via JMX?

Yes, for applications that you cannot instrument directly with the Java client, you can use the JMX Exporter either standalone or as a Java Agent.

What is the performance impact of instrumentation?

Performance across client libraries and languages may vary. For Java, benchmarks indicate that incrementing a counter/gauge with the Java client will take 12-17ns, depending on contention. This is negligible for all but the most latency-critical code.

Implementation

Why are all sample values 64-bit floats?

We restrained ourselves to 64-bit floats to simplify the design. The IEEE 754 double-precision binary floating-point format supports integer precision for values up to 2^{53}. Supporting native 64 bit integers would (only) help if you need integer precision above 2^{53} but below 2^{63}. In principle, support for different sample value types (including some kind of big integer, supporting even more than 64 bit) could be implemented, but it is not a priority right now. A counter, even if incremented one million times per second, will only run into precision issues after over 285 years.

□□□ #2
□ □□□ □□□ 2026-10-31 13:19:52 UTC
□ □□□ □□□ 2026-10-31 13:19:53 UTC