Prometheus from Zero

Metrics Federation Initiative

Hugo Prudente
@hugoprudente

Principal, Site Reliability Engineer (SRE)

Workday, Inc @ Dublin - Ireland

March 28, 2025

Agenda

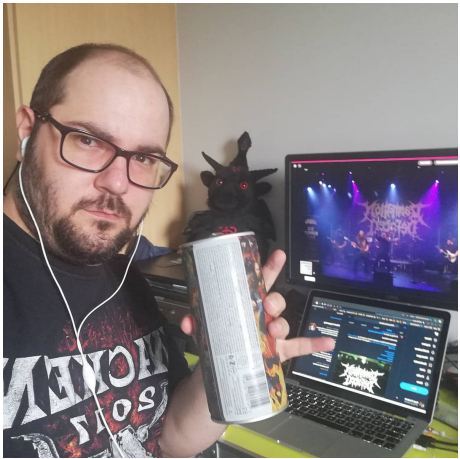
$ whoami

Prometheus
- What is Prometheus?
- What are metrics?
- PromQL
- Instrumenting
- Exporters and Integrations

Demo

References

workday.



Metalhead

Dev/Ops

Open Source and Free Software

Linux

Computer Vision

Cycling

CKA & CKAD & PCA

https://hugoprudente.github.io

https://nerdweek.com.br

Workday Inc | Private Cloud Team.

What is Prometheus?

What is Prometheus?

Prometheus is an open-source monitoring and alerting toolkit originally built at SoundCloud, now hosted by CNCF (since 2016).(Community, 2025a)

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.

In addition to stored time series, Prometheus may generate temporary derived time series as a result of queries.

Features

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

What are metrics?

What are metrics?

Metrics are numerical measurements. The term time series refers to the recording of changes over time, of the following types. (Community, 2025g)

Counter: A counter is a cumulative metric that represents a single monotonically increasing counter whose value can only increase or be reset to zero on restart.

Gauge: A gauge is a metric that represents a single numerical value that can arbitrarily go up and down, e.g.

Histogram: A histogram samples observations (usually things like request durations or response sizes) and counts them in configurable buckets. It also provides a sum of all observed values.

Summary: Similar to a histogram, a summary samples observations (usually things like request durations and response sizes). While it also provides a total count of observations and a sum of all observed values, it calculates configurable quantiles over a sliding time window.

Metrics Name, Labels, Jobs and Instances

Every time series is uniquely identified by its metric name and optional key-value pairs called labels. (Community, 2025c)

```
<metric name>{<label name>=<label value>, ...} <value>
# example
api_http_requests_total{method="POST", handler="/messages"} 100
```

Note: The colons are reserved for user defined recording rules (Community, 2025d). They should not be used by exporters or direct instrumentation.

```
# example
method:api_http_requests_total:sum{method="POST"}
```

Metrics Name, Labels, Jobs and Instances

When Prometheus scrapes a target, it attaches some labels automatically to the scraped time series which serve to identify the scraped target (Community, 2025f):

```
<metric name>{<label name>=<label value>, job=<job-=name>, instance=<instance-id>} <value>
# example
api_http_requests_total{method="POST", handler="/messages",
                        job="haproxy", instance="192.168.10.1:9000"} 100
```

job: The configured job name that the target belongs to.

instance: The <host>:<port> part of the target's URL that was scraped.

PromQL

PromQL: Basic

In Prometheus's expression language, an expression or sub-expression can evaluate to one of four types:(Community, 2025h)

Instant vector	a set of time series containing a single sample for each time series, all sharing the same timestamp
Range vector	a set of time series containing a range of data points over time for each time series
Scalar	a simple numeric floating point
String	a simple string value; currently unused

PromQL: Basic

ms – milliseconds

s – seconds – 1s equals 1000ms

m – minutes – 1m equals 60s (ignoring leap seconds)

h – hours – 1h equals 60m

d – days – 1d equals 24h (ignoring so-called daylight saving time)

w – weeks – 1w equals 7d

y – years – 1y equals 365d (ignoring leap days)

PromQL: Basic

=	Select labels that are exactly equal to the provided string.
!=	Select labels that are not equal to the provided string.
=~	Select labels that regex-match the provided string.
! ~	Select labels that do not regex-match the provided string.

Regex matches are fully anchored. A match of env =~ "foo" is treated as env =~ "^foo$"

PromQL: Basic

The offset modifier allows changing the time offset for individual instant and range vectors in a query.

Range vector literals work like instant vector literals, except that they select a range of samples back from the current instant. Syntactically, a float literal is appended in square brackets ([]) at the end of a vector selector

```
# offset
http_requests_total{} offset 5m
# range vector
http_requests_total{job="prometheus"}[5m]
```

PromQL: Operators

The following binary arithmetic operators and comparison binary exist in Prometheus.Community, 2025j

+	addition	==	equal
-	subtraction	!=	not-equal
*	multiplication	>	greater-than
/	division	<	less-than
%	modulo	>=	greater-or-equal
^	power/exponentiation	<=	less-or-equal

PromQL: Functions

There are 57 functions supported by Prometheus by March 2025, Community, 2025i, here are some of the heavily used

sum	abs
avg	ceil
max	floor
min	exp
count	ln
stddev	log2
stdvar	log10
label_values	sqrt
label_replace	year
match	month

Instrumenting

Instrumenting

Before you can monitor your services, you need to add instrumentation to their code via one of the Prometheus client libraries.(Community, 2025b)

- Go
- Java or Scala
- Python
- Ruby
- Rust

And over 15 unofficial third-party client libraries as C, C, Lua, Elixir, Erlang, Haskell, .NetC#, PHP, R and more..

Exporters and Integrations

Exporters and Integrations

There are a number of libraries and servers which help in exporting existing metrics from third-party systems as Prometheus metrics. (Community, 2025e)

- node-exporter
- MySQL Server Exporter
- HAProxy exporter
- Kafka exporter

with over 215 exporters native developed on tools or as external exporters that can be deployed as side-car ready to be scrapped by Prometheus.

Demo

References

Community, Prometheus (2025a). Prometheus. URL: https://prometheus.io/docs/introduction/overview/.

— (2025b). Prometheus Client Libs. URL: https://prometheus.io/docs/instrumenting/clientlibs/.

— (2025c). Prometheus Concepts. URL: https://prometheus.io/docs/concepts/data_model/.

— (2025d). Prometheus Concepts. URL: https://prometheus.io/docs/prometheus/latest/configuration/recording_rules/.

— (2025e). Prometheus Exporters, Integrations. URL: https://prometheus.io/docs/instrumenting/exporters/.

— (2025f). Prometheus Job Instances. URL: https://prometheus.io/docs/concepts/jobs_instances/.

— (2025g). Prometheus Metric Types. URL: https://prometheus.io/docs/concepts/metric_types/.

— (2025h). Prometheus Querying: Basic. URL: https://prometheus.io/docs/prometheus/latest/querying/basics/.

— (2025i). Prometheus Querying: Functions. URL: https://prometheus.io/docs/prometheus/latest/querying/functions/.

— (2025j). Prometheus Querying: Operators. URL: https://prometheus.io/docs/prometheus/latest/querying/operators/.

Thank You!

powered by LaTeX

Figure: Linkedin

Hugo Prudente
@hugoprudente

Principal, Site Reliability Engineer (SRE)
https://workday.wd5.myworkdayjobs.com/Workday

https://github.com/hugoprudente/presentations-pub

https://hugoprudente.github.io

https://nerdweek.com.br