

OpenMetrics

RichiH

 Grafana Labs



Prometheus

- Inspired by Google's Borgmon
- Time series database
- Prometheus exposition format 0.0.4, stable since 2014
- Cortex & Thanos for long-term storage
- Kubernetes metrics explicitly designed for Prometheus
- Cloud-native default
- Instrumentation library 29th global GitHub Go rank
- 700 publicly listed exporters, many more closed & integrations



Wide, organic adoption since 2012

Millions of installations that we know of

OpenMetrics

- Initial goals in 2016
 - Neutral name to ease adoption
 - RFC to use in tenders with network vendors
 - Careful & compatible evolution of Prometheus exposition 0.0.4
- Input from dozens of people, projects, and companies over the years
- Production usage in Prometheus since 2018
- Re-implemented from reference code by DataDog, without our help



**Prometheus ecosystem is moving to OpenMetrics
100%**

Ranked 5th place in 2020 CNCF End User Survey

OpenMetrics

Text format MUST be supported:

```
vendor_interface_in_bytes_total{interface="TenGigE0/0/2/0"} 12345678  
vendor_interface_out_bytes_total{interface="TenGigE0/0/2/0"} 34567890  
vendor_psu_status{module="psu1"} 1  
vendor_temperature_celsius{module="psu1",location="inlet"} 35
```

Equivalent protobuf MAY be used

Thank you!

richih@richih.org

<https://twitter.com/TwitchiH>

FAQs

- Why not SNMP?
 - Personally respect it, but vendors losing interest
 - snmp_exporter maps SNMP -> OpenMetrics
- Why not YANG, et al?
 - Too narrow in scope
 - One standard from GenSet, to router, to DB, to microservice
- Scale?
 - Single Prometheus can ingest 100 million metrics at 10-120s cadence

Breaking changes, relative to Prom 0.0.4

- Counters now require `_total` on the time series
 - Common convention, but now enforced, not really breaking
 - If your metric was `cpu_seconds`, our libraries will migrate you to `cpu_seconds_total`
- Timestamps are in seconds, for consistency
 - We use base units everywhere else
 - Used to be milliseconds
 - Exposing an explicit timestamp is possible, but usually an antipattern

Improvements & interop, relative to Prom 0.0.4

- Cleaner and tighter specification, e.g. spacing, escaping
- Explicit EOF to detect incomplete scrapes
- Allowing for nanosecond resolution timestamps
- 64-bit integer values
- Unit as new metadata
- `_created` for metric creation & resets
- Considerations for both pull and push
- Text format still mandatory, reintroduce optional protobuf

What's New: Exemplars

Exemplars allow linking certain metrics to example traces:

```
# TYPE foo histogram
foo_bucket{le="0.01"} 0
foo_bucket{le="0.1"} 8 # {id="abc"} 0.043
foo_bucket{le="1"} 10 # {id="def"} 0.29
foo_bucket{le="10"} 17 # {id="ghi"} 7.73
foo_bucket{le="+Inf"} 18
foo_count 18
foo_sum 219.3
foo_created 1520430000.123
```

Further reading

https://docs.google.com/presentation/d/1zD8tcStqqAGeOh2IKEOLQpyEJjU2RcSvEbm5FqQ4Z_I

<https://github.com/OpenObservability/OpenMetrics/>

<https://www.cncf.io/blog/2020/09/11/cncf-end-user-technology-radar-observability-september-2020/>

<https://github.com/cncf/enduser-public/blob/master/end-user-tech-radar/cncf-end-user-tech-radar-2020-09-observability.pdf>