

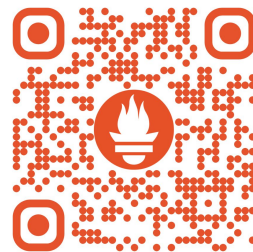
Prometheus 3.0

Bryan Boreham, Grafana

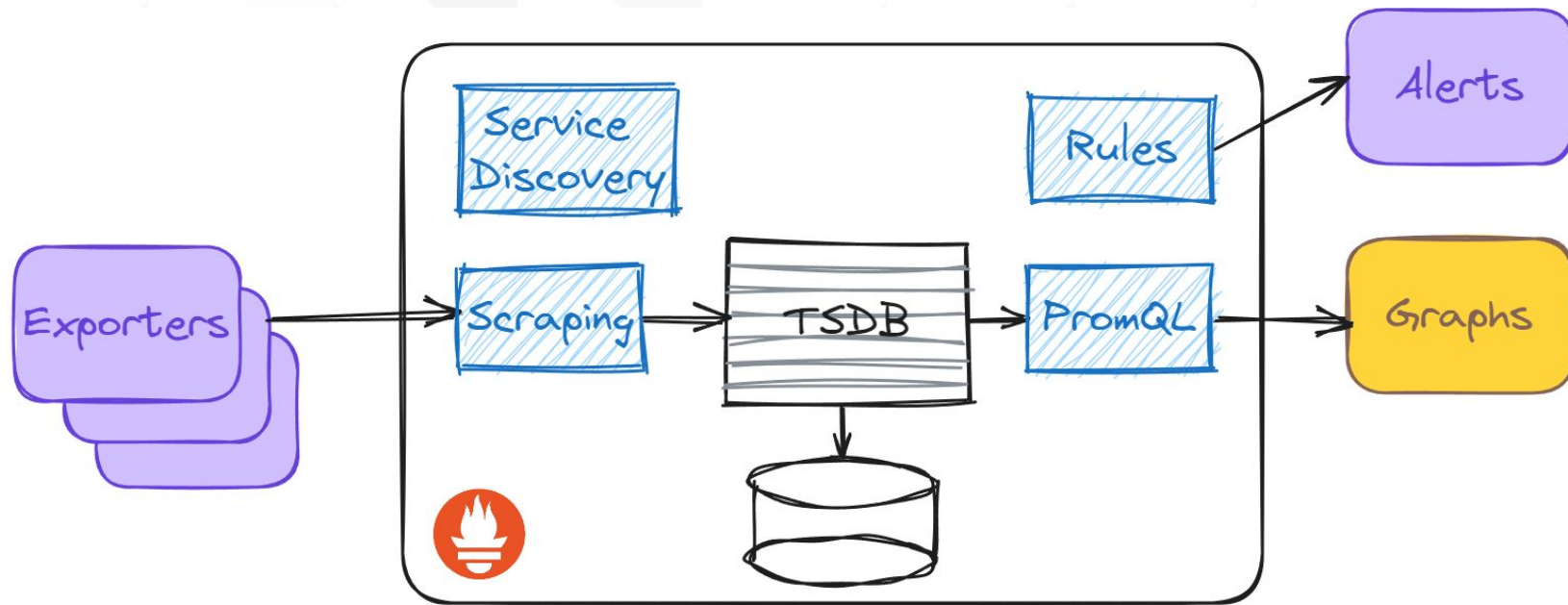
Jan Fajerski, Red Hat

Show of hands





<https://youtu.be/rT4fJNbf14>



History

Prometheus started at Soundcloud

Second project to join and graduate CNCF

Active exporter ecosystem, widely used

2012

Initial
commit

2016

Join
CNCF

2017

2.0
Release

2018

Graduate
CNCF

2022

First LTS
release

2024

3.0
Release

What is Prometheus 3.0 ?

First Major release in 7 years

Brand new UI

OTEL Compatibility

UTF-8 support

Remote Write 2.0

Native Histogram support

Breaking changes

Checkout the [Migration guide](#)

Previous talks



https://www.youtube.com/results?search_query=Prometheus+3.0

Changes

What does this selector do? `my_series{id=~".*"}{`

A: No work, just fetches all **my_series**

B: Scans every possible value of the **id** label

Answer: B in Prometheus 1.x and 2.x, A in Prometheus 3.x

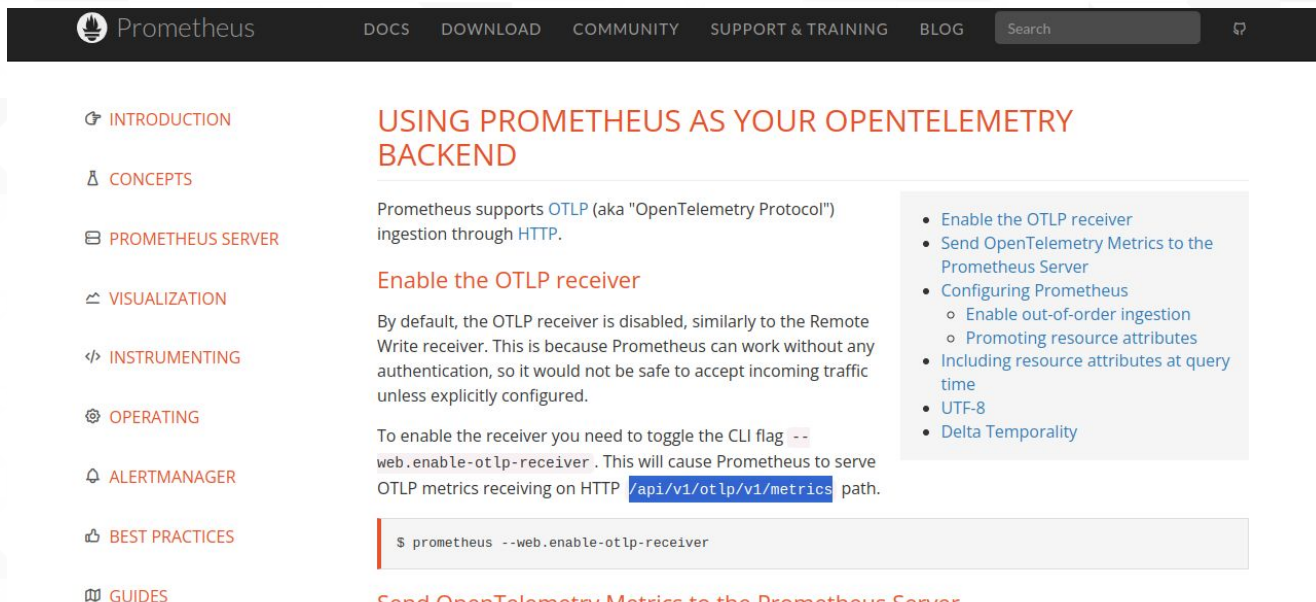
Once you've made one breaking change...

- [CHANGE] Set the GOMAXPROCS variable automatically to match the Linux CPU quota. Use `--no-auto-gomaxprocs` to disable it. The `auto-gomaxprocs` feature flag was removed. [#15376](#)
- [CHANGE] Set the GOMEMLIMIT variable automatically to match the Linux container memory limit. Use `--no-auto-gomemlimit` to disable it. The `auto-gomemlimit` feature flag was removed. [#15373](#)
- [CHANGE] Scraping: Remove implicit fallback to the Prometheus text format in case of invalid/missing Content-Type and fail the scrape instead. Add ability to specify a `fallback_scrape_protocol` in the scrape config. [#15136](#)
- [CHANGE] Remote-write: default `enable_http2` to false. [#15219](#)
- [CHANGE] Scraping: normalize "le" and "quantile" label values upon ingestion. [#15164](#)
- [CHANGE] Scraping: config `scrape_classic_histograms` was renamed to `always_scrape_classic_histograms`. [#15178](#)
- [CHANGE] Config: remove `expand-external-labels` flag, expand external labels env vars by default. [#14657](#)
- [CHANGE] Disallow configuring AM with the v1 api. [#13883](#)
- [CHANGE] `regexp` . now matches all characters (performance improvement). [#14505](#)
- [CHANGE] `holt_winters` is now called `double_exponential_smoothing` and moves behind the [experimental-promql-functions feature flag](#). [#14930](#)
- [CHANGE] API: The OTLP receiver endpoint can now be enabled using `--web.enable-otlp-receiver` instead of `--enable-feature=otlp-write-receiver`. [#14894](#)
- [CHANGE] Prometheus will not add or remove port numbers from the target address. `no-default-scrape-port` feature flag removed. [#14160](#)
- [CHANGE] Logging: the format of log lines has changed a little, along with the adoption of Go's Structured Logging package. [#14906](#)
- [CHANGE] Don't create extra `_created` timeseries if feature-flag `created-timestamp-zero-ingestion` is enabled. [#14738](#)
- [CHANGE] Float literals and time durations being the same is now a stable feature. [#15111](#)
- [CHANGE] UI: The old web UI has been replaced by a completely new one that is less cluttered and adds a few new features (PromLens-style tree view, better metrics explorer, "Explain" tab). However, it is still missing some features of the old UI (notably, exemplar display and heatmaps). To switch back to the old UI, you can use the feature flag `--enable-feature=old-ui` for the time being. [#14872](#)
- [CHANGE] PromQL: Range selectors and the lookback delta are now left-open, i.e. a sample coinciding with the lower time limit is excluded rather than included. [#13904](#)
- [CHANGE] Kubernetes SD: Remove support for `discovery.k8s.io/v1beta1` API version of `EndpointSlice`. This version is no longer served as of Kubernetes v1.25. [#14365](#)
- [CHANGE] Kubernetes SD: Remove support for `networking.k8s.io/v1beta1` API version of Ingress. This version is no longer served as of Kubernetes v1.22. [#14365](#)
- [CHANGE] UTF-8: Enable UTF-8 support by default. Prometheus now allows all UTF-8 characters in metric and label names. The corresponding `utf8-name` feature flag has been removed. [#14705](#)
- [CHANGE] Console: Remove example files for the console feature. Users can continue using the console feature by supplying their own JavaScript and templates. [#14807](#)
- [CHANGE] SD: Enable the new service discovery manager by default. This SD manager does not restart unchanged discoveries upon reloading. This makes reloads faster and reduces pressure on service discoveries' sources. The corresponding `new-service-discovery-manager` feature flag has been removed. [#14770](#)
- [CHANGE] Agent mode has been promoted to stable. The feature flag `agent` has been removed. To run Prometheus in Agent mode, use the new `--agent` cmdline arg instead. [#14747](#)
- [CHANGE] Remove deprecated `remote-write-receiver`, `promql-at-modifier`, and `promql-negative-offset` feature flags. [#13456](#), [#14526](#)
- [CHANGE] Remove deprecated `storage.tsdb.allow-overlapping-blocks`, `alertmanager.timeout`, and `storage.tsdb.retention` flags. [#14640](#), [#14643](#)

`/api/v1/otlp/v1/metrics` Endpoint, disabled by default

Accepts POST request via OTLP/HTTP protocol

Requires additional configuration to work well



The screenshot shows the Prometheus website with a dark navigation bar at the top containing the Prometheus logo, the name 'Prometheus', and links for 'DOCS', 'DOWNLOAD', 'COMMUNITY', 'SUPPORT & TRAINING', and 'BLOG'. A search bar is also present. The left sidebar lists various documentation sections: 'INTRODUCTION', 'CONCEPTS', 'PROMETHEUS SERVER', 'VISUALIZATION', 'INSTRUMENTING', 'OPERATING', 'ALERTMANAGER', 'BEST PRACTICES', and 'GUIDES'. The main content area is titled 'USING PROMETHEUS AS YOUR OPENTELEMETRY BACKEND'. It explains that Prometheus supports OTLP (OpenTelemetry Protocol) ingestion through HTTP. A section titled 'Enable the OTLP receiver' states that by default, the OTLP receiver is disabled. To enable it, one must toggle the CLI flag `--web.enable-otlp-receiver`, which will cause Prometheus to serve OTLP metrics on the `/api/v1/otlp/v1/metrics` path. A code block shows the command: `$ prometheus --web.enable-otlp-receiver`. A list of configuration options for the OTLP receiver is provided in a box on the right, including enabling the receiver, sending metrics to the Prometheus Server, configuring Prometheus (with sub-points for out-of-order ingestion and resource attributes), including resource attributes at query time, UTF-8 encoding, and Delta Temporality.

Prometheus

DOCS DOWNLOAD COMMUNITY SUPPORT & TRAINING BLOG Search

INTRODUCTION

CONCEPTS

PROMETHEUS SERVER

VISUALIZATION

INSTRUMENTING

OPERATING

ALERTMANAGER

BEST PRACTICES

GUIDES

USING PROMETHEUS AS YOUR OPENTELEMETRY BACKEND

Prometheus supports [OTLP](#) (aka "OpenTelemetry Protocol") ingestion through [HTTP](#).

Enable the OTLP receiver

By default, the OTLP receiver is disabled, similarly to the Remote Write receiver. This is because Prometheus can work without any authentication, so it would not be safe to accept incoming traffic unless explicitly configured.

To enable the receiver you need to toggle the CLI flag `--web.enable-otlp-receiver`. This will cause Prometheus to serve OTLP metrics receiving on HTTP `/api/v1/otlp/v1/metrics` path.

```
$ prometheus --web.enable-otlp-receiver
```

- [Enable the OTLP receiver](#)
- [Send OpenTelemetry Metrics to the Prometheus Server](#)
- [Configuring Prometheus](#)
 - [Enable out-of-order ingestion](#)
 - [Promoting resource attributes](#)
- [Including resource attributes at query time](#)
- [UTF-8](#)
- [Delta Temporality](#)

Send OpenTelemetry Metrics to the Prometheus Server

Support all characters ("UTF-8")

This change allows Prometheus to accept OpenTelemetry Semantic Conventions.

Example: **service.name**

Characters outside of traditional Prometheus label names need quoting.

```
{"service.name"="nginx"}
```

<https://prometheus.io/docs/guides/utf8/>

promql: Make range selections left-open and right-closed #13213

✓ Closed

#13904

“This is, however, a breaking change, although the **impact is mostly academic**. Therefore, we should implement this change with the upcoming 3.0.0 release.”

“This change has likely few effects for everyday use, except for **some** subquery use cases.

Query front-ends that **align queries** usually align subqueries to multiples of the step size. These subqueries will likely be affected.

Tests are more likely to be affected. To fix those either adjust the expected number of samples or extend the range by less than one sample interval.”

```
rate(prometheus_http_requests_total[1m:1m])
```

New UI

>_ `rate(prometheus_http_requests_total[1m:1m])`

Execute

Table Graph Explain

< 2025-01-29 13:00:00 x >

Load time: 46ms Result series: 0

Empty query result

This query returned no data.

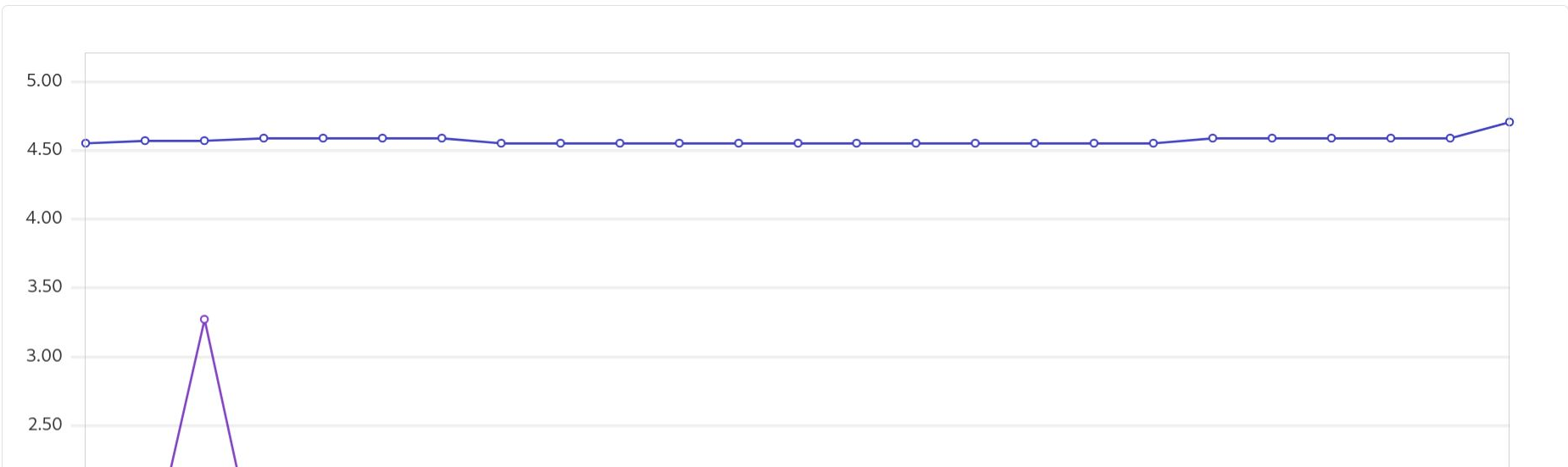
+ Add query

> `rate(prometheus_http_requests_total[1m1ms:1m])` Execute

Table Graph Explain

- 1d + < 2025-01-29 13:00:00 x > 1h

Unstacked Stacked





> `rate(prometheus_http_requests_total[1m:1m])`



Execute

`rate` 0 results — 46ms

`[1m:1m]` 73 results — 112ms — handler: 56 code: 9 instance: 1 job: 1

`prometheus_http_requests_total` 73 results — 52ms — handler: 56 code: 9 instance: 1 job: 1



Table



Graph



Explain

< 2025-01-29 13:00:00 × >

Load time: 44ms Result series: 71

<code>prometheus_http_requests_total{code="200", handler="/", instance="demo.do.prometheus.io:9090", job="prometheus"}</code>	0 @ 1738155600
<code>prometheus_http_requests_total{code="200", handler="/-/healthy", instance="demo.do.prometheus.io:9090", job="prometheus"}</code>	3063 @ 1738155600
<code>prometheus_http_requests_total{code="200", handler="/-/quit", instance="demo.do.prometheus.io:9090", job="prometheus"}</code>	0 @ 1738155600

>_ `rate(prometheus_http_requests_total[1m1ms:1m])` ⋮ Execute



Table Graph Explain

< 2025-01-29 13:00:00 x >

Load time: 47ms Result series: 71

prometheus_http_requests_total{code="200", handler="/", instance="demo.do.prometheus.io:9090", job="prometheus"}	0 @ 1738155540
	0 @ 1738155600
prometheus_http_requests_total{code="200", handler="/-/healthy", instance="demo.do.prometheus.io:9090", job="prometheus"}	3061 @ 1738155540
	3063 @ 1738155600



Query



Alerts



Status

▼



Too many match groups to display, only showing 100 out of 109 groups.

[Show all groups](#)

no matching series

1

```
{code="200", handler="/metrics", instance="demo.do.prometheus.io:9090",  
  job="prometheus"}
```

prometheus_http_requests_total{}

二

dropped

no matching series

1

```
{code="200", handler="/-/healthy", instance="demo.do.prometheus.io:9090",  
  job="prometheus"}
```

prometheus_http_requests_total{

$$=$$

dropped

no matching series

1

```
{code="200", handler="/api/v1/query_range",  
instance="demo.do.prometheus.io:9090", job="prometheus"}
```

prometheus_http_requests_total{

==

dropped

```
{code="200", handler="/api/v1/query", instance="demo.do.prometheus.io:9090",
```

>_ prometheus_http_request_duration_seconds_sum / ignoring (code) prometheus_http_requests_total

Execute

prometheus_http_request_duration_seconds_sum 39 results - 209ms - handler: 39 instance: 1 job: 1

/ ignoring(code) ● Error executing query: found duplicate series for the match group {handler="/api/v1/query", instance="demo.do.prometheus.io:9090", job="prometheus"} on t

prometheus_http_requests_total 70 results - 209ms - handler: 56 code: 7 instance: 1 job: 1

Table Graph Explain

Vector-to-vector binary operation

This node calculates the result of applying the "/" operator between the sample values of matching series from two sets of time series.

- `ignoring(code)`: series on both sides are matched on all of their labels, except `code`.
- One-to-one match. Each series from the left-hand side is allowed to match with at most one series on the right-hand side, and vice versa.

⚠️ Error in match group below

Binary operators only allow **one-to-one** matching by default, but we found **multiple series on the right side** for this match group.

Possible fixes:

- **Allow one-to-many matching:** If you want to allow one-to-many matching, you need to explicitly request it by adding a `group_right()` modifier to the operator:

... / `ignoring(code)` `group_right()` ...

- **Update your matching parameters:** Consider including more differentiating labels in your matching modifiers (via `on()` / `ignoring()`) to split multiple series into distinct match groups.
- **Aggregate the input:** Consider aggregating away the extra labels that create multiple series per group before applying the binary operation.

```
{handler="/api/v1/query_range",
instance="demo.do.prometheus.io:9090",
job="prometheus"}

prometheus_http_request_duration_seconds_sum{
```

/

```
{handler="/api/v1/query_range",
instance="demo.do.prometheus.io:9090",
job="prometheus"}

prometheus_http_requests_total{code="200"}
prometheus_http_requests_total{code="503"}
prometheus_http_requests_total{code="400"}
```

=

error, result omitted

What went wrong?

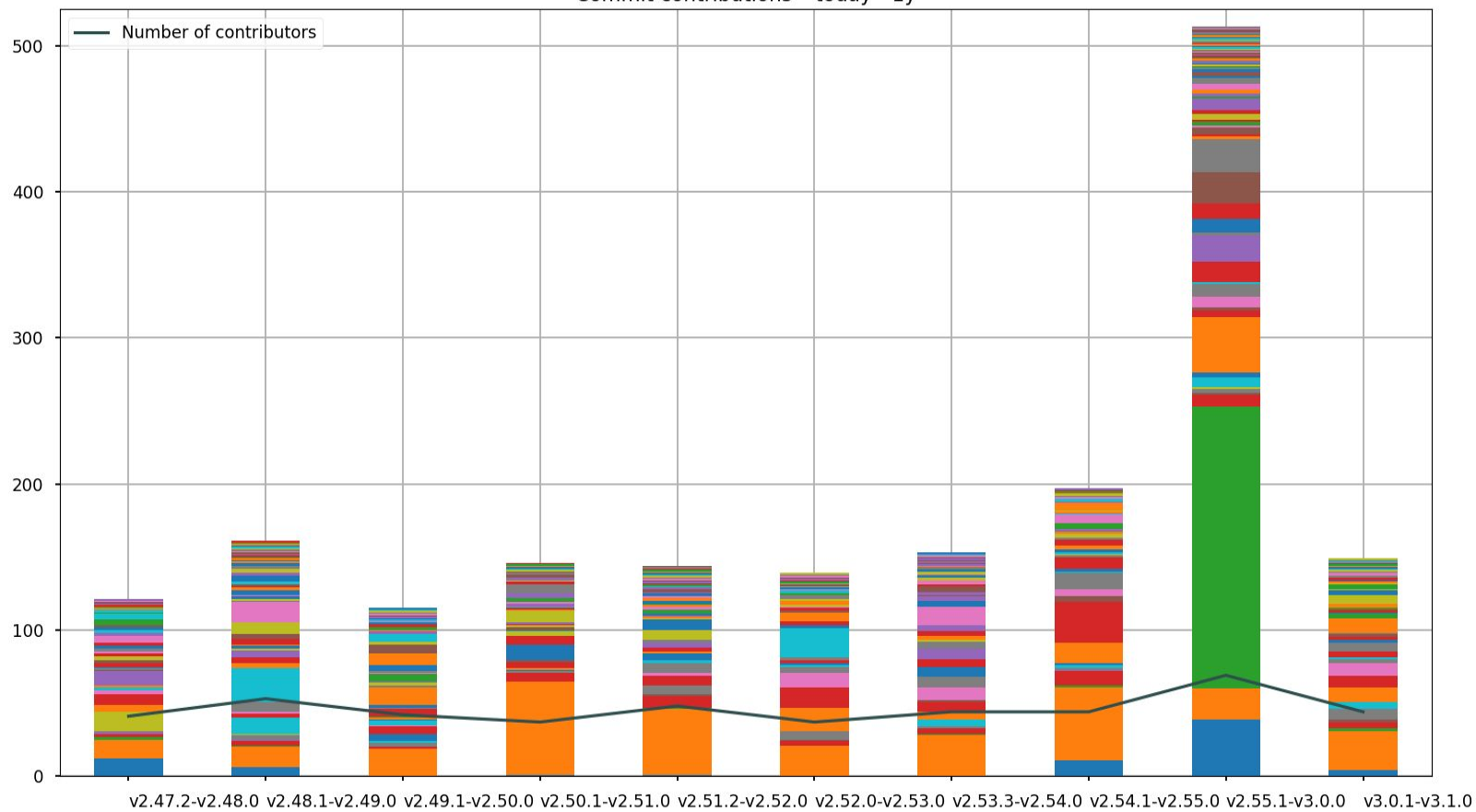
What went wrong in 3.0?

Just three things needed fixing quickly:

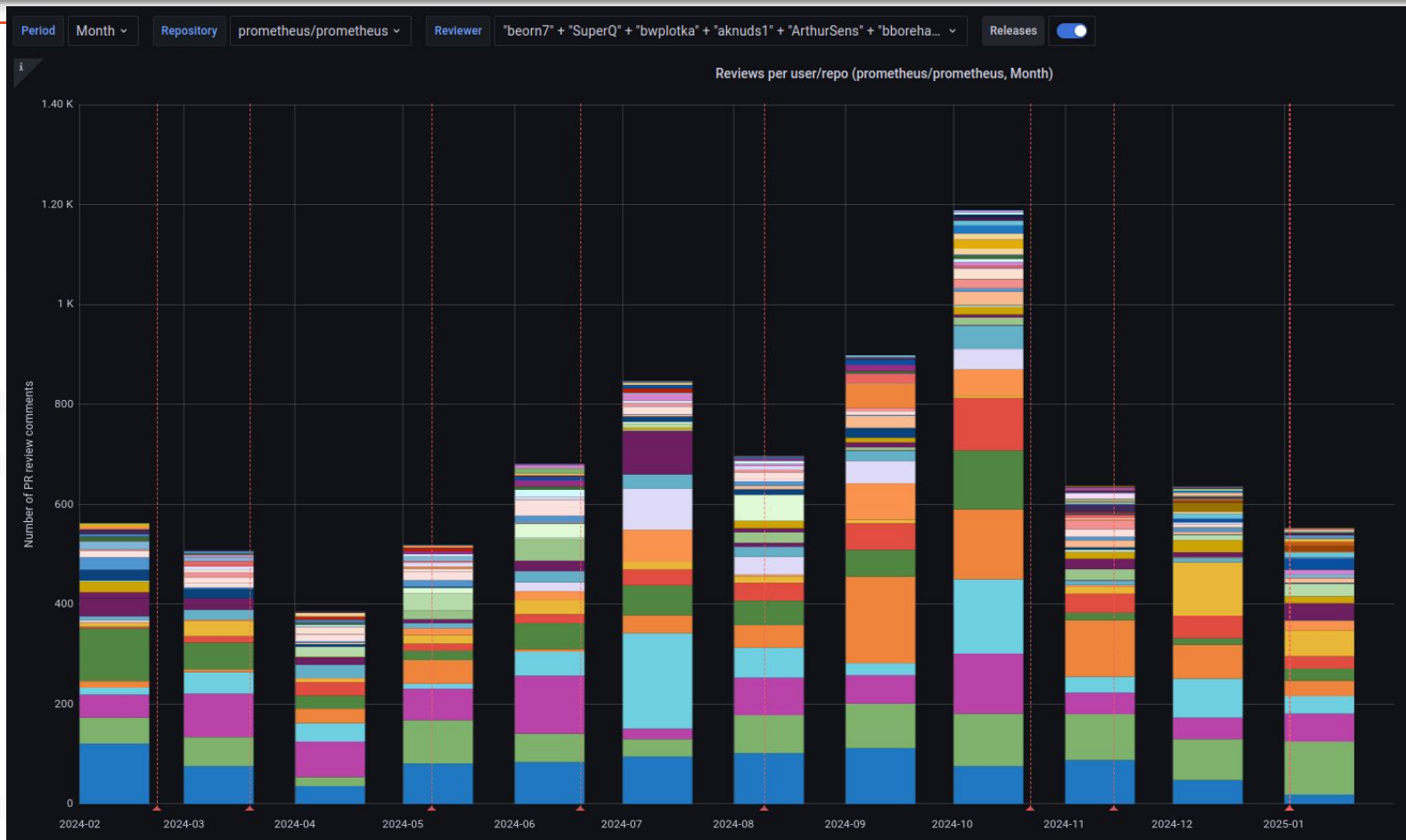
- [BUGFIX] Promql: Make subqueries left open. #15431
- [BUGFIX] Fix memory leak when query log is enabled. #15434
- [BUGFIX] Support utf8 names on /v1/label/:name/values endpoint. #15399

Contribution development

Commit contributions ~today - 1y



Review contributions per month



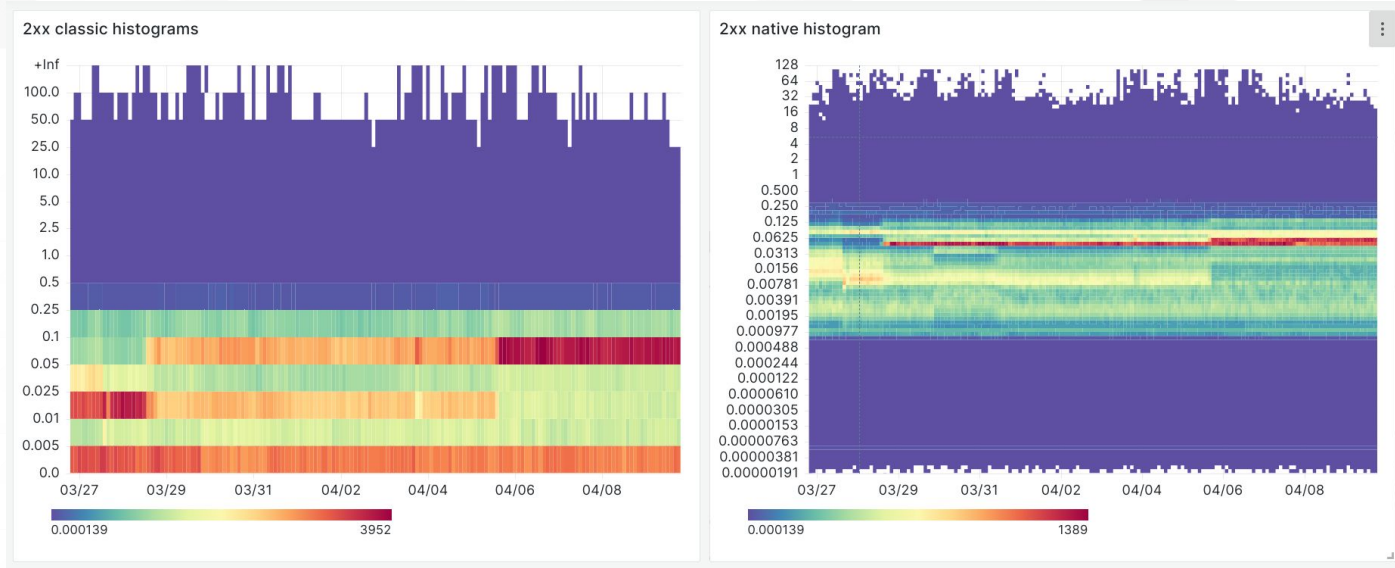
Join Us!



Questions?

What did we not quite get to?

Native histograms



What did we not quite get to?

Better OTEL support

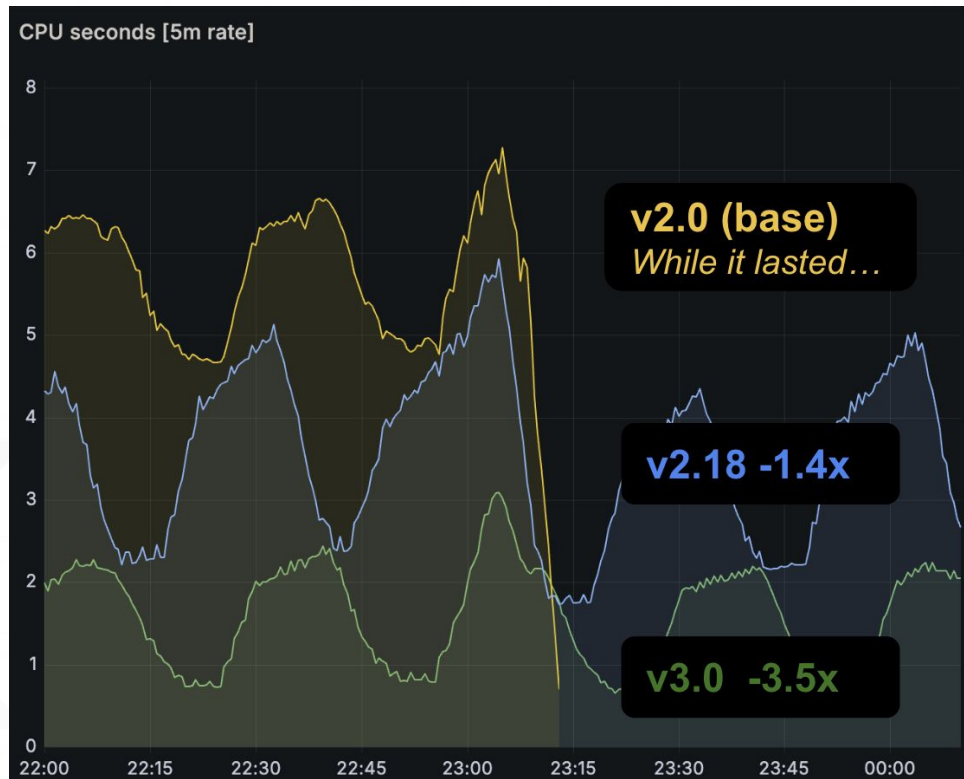
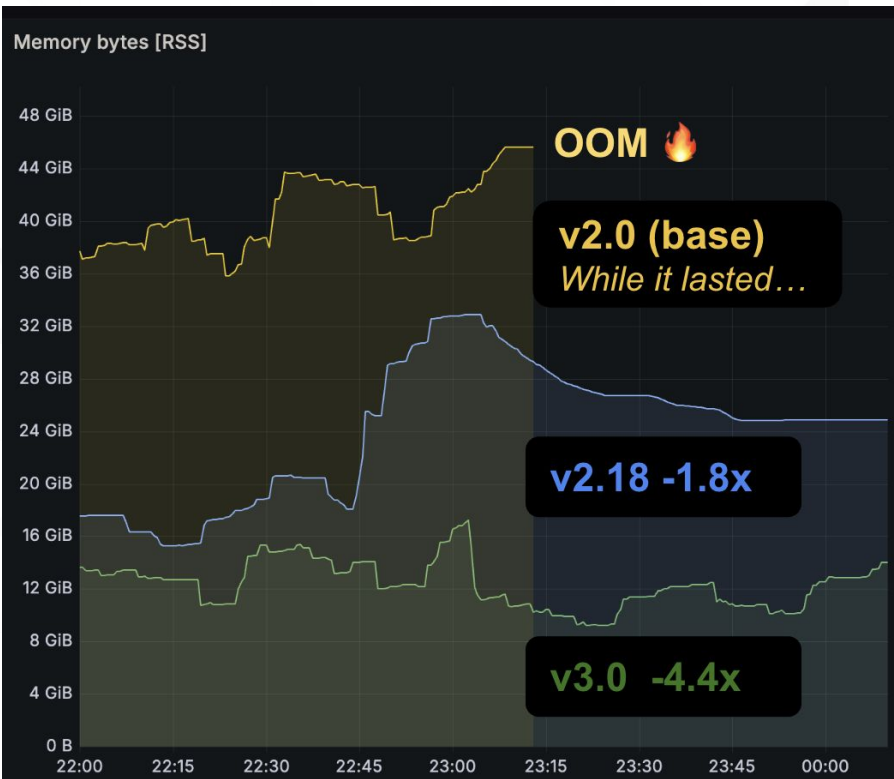
Delta temporality

Resource attributes

The background of the slide is a solid orange color with a subtle, repeating geometric pattern of triangles and squares in varying shades of orange, creating a textured effect.

Performance

Prometheus Performance



PROMETHEUS REMOTE-WRITE SPECIFICATION

- Version: 2.0-rc.3
- Status: **Experimental**
- Date: May 2024

The Remote-Write specification, in general, is intended to document the standard for how Prometheus and Prometheus Remote-Write compatible senders send data to Prometheus or Prometheus Remote-Write compatible receivers.

This document is intended to define a second version of the [Prometheus Remote-Write API](#) with minor changes to protocol and semantics. This second version adds a new Protobuf Message with new features enabling more use cases and wider adoption on top of performance and cost savings. The second version also deprecates the previous Protobuf Message from a [1.0 Remote-Write specification](#) and adds mandatory `X-Prometheus-Remote-Write-*`-Written HTTP response headers for reliability purposes.

Finally, this spec outlines how to implement backwards-compatible senders and receivers (even under a single endpoint) using existing basic content negotiation request headers. More advanced, automatic content negotiation mechanisms might come in a future minor version if needed. For the rationales behind the 2.0 specification, see [the formal proposal](#).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

NOTE: This is a release candidate for Remote-Write 2.0 specification. This means that this specification is currently in an experimental state--no major changes are expected, but we reserve the right to break the compatibility if it's necessary, based on the early adopters' feedback. The potential feedback, questions and suggestions should be added as comments to the PR with the open proposal.

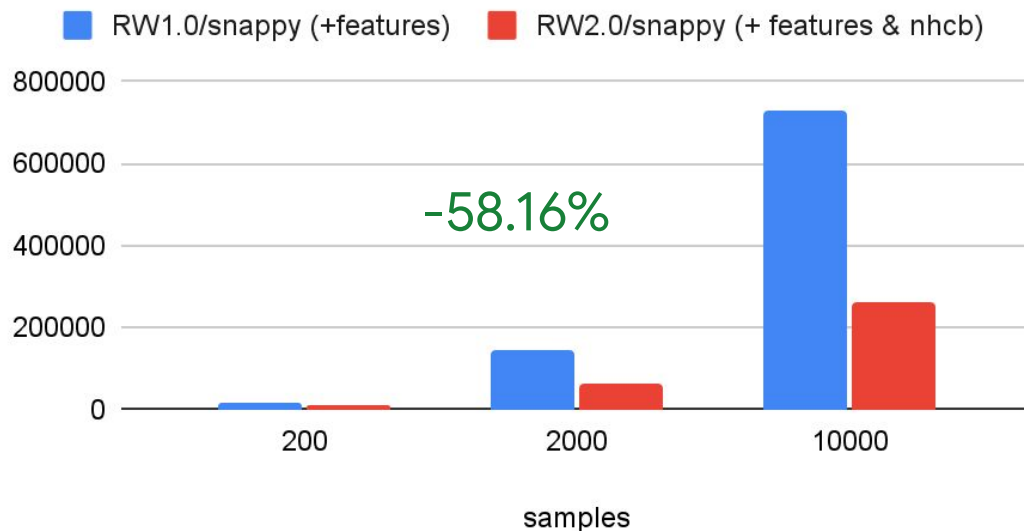
- [Introduction](#)
 - [Background](#)
 - [Glossary](#)
- [Definitions](#)
 - [Protocol](#)
 - [Response](#)
 - [Retries & Backoff](#)
 - [Backward and Forward Compatibility](#)
- [Protobuf Message](#)
 - `io.prometheus.write.v2.Request`
- [Out of Scope](#)
- [Future Plans](#)
- [Related](#)
 - [FAQ](#)

- PRW 2.0 Spec has native support for
 - Native Histograms
 - Created Timestamp
 - Exemplars
 - UTF-8 support
- Structure and Transactionality
- Enables partial write stats

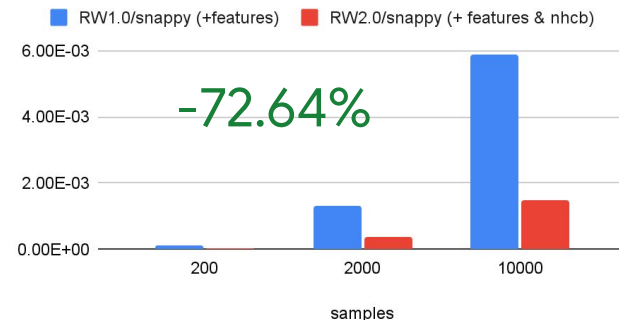
<https://www.youtube.com/watch?v=o5HpeMtpsTg&p=ygUYa3ViZWNvbiByZW1vdGUgd3JpdGUgMi4w>

Remote Write 2.0 Performance

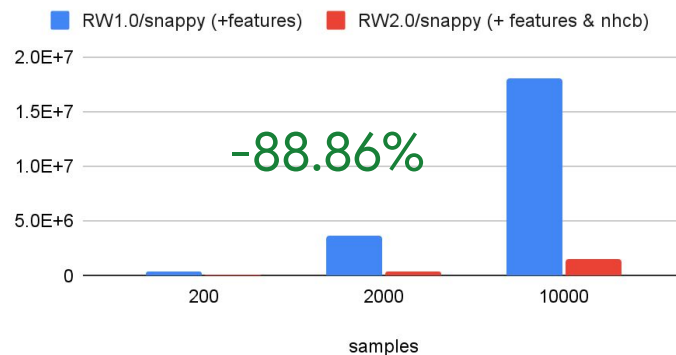
Message bytes on the wire (lower is better)



Serialization CPU nanoseconds (lower is better)



Serialization bytes allocated (lower is better)



Thanks to cstyan and bwplotka <https://sched.co/1i7kO>