

Prometheus 2020 Dev Summit 1

2020-07-10, 14:00 - 18:00 UTC, Online

Last Meetings

- [2017](#)
- [2018](#)
- [2019.05](#)
- [2019.11](#)

Attendees

- Richard “RichiH” Hartmann
- Ben Kochie
- Chris Marchbanks
- Matt Layher
- Björn Rabenstein
- Frederic Branczyk
- Brian Brazil
- Matthias Rampke, partially
- Simon Pasquier, partially
- Goutham Veeramachaneni
- Bartłomiej Płotka
- Callum Styan
- Julius Volz, partially
- Julien Pivotto, partially

Editorial note: We captured less of the actual discussions in this document than we used to.
Incentivizing people to write notes online is harder than if we're all in the same room.

Meeting Notes

Master plan for metadata

- Beorn: <walking through the below>
 - Recap existing API, usage, problems, ...: [by target](#), [by metric](#).
 - Experience with [propagating metadata via remote write](#).

- Problem of [matching metadata for complex metrics](#), i.e. a histogram `foo` will have no timeseries/"flattened-metric" that is actually called `foo` (but rather `foo\_sum`, `foo\_count`, `foo\_bucket`).
- How to deal with HELP and TYPE being legitimately different for the same metric name (if coming from different targets).
- How to deal with HELP and TYPE being legitimately different over time (target changes instrumentation).
- What good is a "rich" metadata store (per "complex" series perhaps, changes over time)? Brainstorm starters: Mark first/last sample of a series, create-at for counters, sparse histogram bucketing schema, mark counter resets, intended scrape interval, ...
- CONSENSUS: We want to support more metadata. Work will be done in [LINK TO DESIGN DOC](#)
- CONSENSUS: <https://github.com/prometheus/prometheus/pull/6815> is acceptable as an EXPERIMENTAL stopgap. It is likely to change for 3.x

Bit more strict protection for main/release branches. Disable force-push, warnings for no-review, no-build merges.

- Bartek: Visual indicator if PR is ready for merge WRT approvals, CI tests, ...
 - It's just visual. Admin (i.e. all of -team) can override at any time.
 - We have this in some prometheus/... repos already. Should we have this for all prometheus/... repositories?
 - If we require formal approval, just a :+1: and click merge is not a valid workflow anymore (either do the formal approval or do the override).
 - Source: <https://groups.google.com/forum/#!topic/prometheus-team/OjrWeMCEUfQ>

<15:00 Julius drops out for a bit>

- CONSENSUS: Set "Require status checks to pass before merging" (see figure above) on all prometheus/... repos.
- Prevent direct pushes to main branch? Prevent force pushes to main branch?
 - Brian: Java release tools need to be able to push to main branch
 - CONSENSUS: Disable force-pushes to all main branches.
 - CONSENSUS: Default behavior is to allow pushes to the main branch, but it should be disabled for certain "important" repos, e.g. prometheus/prometheus, at the discretion of the maintainer.

Bartek: Backfill missing things, direction: [ticket importers](#)

- Work within Red Hat on backfilling important data through a backfilling tool. This started a discussion on how to allow backfilling in the generic case. Had a Twitter survey
 - 319 votes total
 - CSV 26.6%
 - OpenMetrics file 13.2%
 - Streaming OpenMetrics 15.7%

- Remote-Write API 44.5%
- Tool accepts modified Prometheus exposition format/OpenMetrics data carrying time series, put into storage, vertical compaction
- Might want to support
 - Remote-write to Prometheus, promtool at first
 - Continuous out-of-order data
 - Should we even support OM
 - RichiH: You can put this into OM format and have long lines with ts/value tuples
 - Brian: You can use line-per-timestamp, just like Prometheus format
- Bartek: VM supports CSV, we don't support it
 - RichiH: Why not also support CSV? Many users use Excel etc as their DB, CSV is horrible but well-supported
 - Brian: Well-formatted CSV or just any CSV?
 - <all>: Well-formatted
- Bartek: Should we support OM as well?
 - RichiH: Yes
 - Bartek: Yes, because of Metadata push? But maybe later
 - RichiH: Both because OpenMetrics / Prometheus Exposition is natural for the ecosystem and our users and custom tooling exists in a myriad of places today. CSV because it's the lingua franca in data transport even if we don't like it. TSV better than CSV, but implementation detail. Large target audience wanting to backfill, need to support a few common cases

<15:27 Julien Pivotto joins>

- **CONSENSUS: We want to support Prometheus text exposition format and OpenMetrics, and one well-defined CSV format for importing data**

<15:32 Julius returns>

<15:32-15:42 break>

<Backfilling, cont>

- Bartek: Out of order is a common ask from the community. Maybe even through remote read-write. Storage engine has issues supporting this
- Brian: Let's split, talk about remote-write for backfilling first. Don't see use case as libraries exist - if you can write Go to do remote write, you can write Go to create a tsdb block.
- Frederic: GH/Twitter questions were biased in favor of backfilling/remote
- Matthias R: Copying to disk is possible, but people want an API endpoint; remote read-write is what we use for this
 - RichiH: 100% agreed. It might be possible, but it's not what users want and remote-write is the default for data transport between Prometheus-like systems now

- RichiH: We need to acknowledge that backfilling continuously in out-of-order is a footgun, but many users want it and we should not block advanced use cases without good reason. We can provide guidance, prioritize certain operations, have default alerts telling the users if they are starving Prometheus of resources, other mechanisms to support and enable, not to prevent
 - Matthias R: Remote read-write has backpressure already
 - Callum: No backpressure at the moment
 - Beorn: Sender will wait for a response before sending more data, receiver can use this to slow down writing
- Brian: This would be a TSDB, not a monitoring system and it makes us a push system
 - RichiH: Disagree; we can support use cases and if we do not, someone else will. They are valid use cases and I would much rather enable them in a safe manner based on our shared experience than have others implement something that “works” and might break
 - Matthias R: Nothing bad about being a TSDB
 - Frederic: Bulk mechanism makes sense, streaming not so much. There are other projects which solve this, e.g. Cortex & Thanos.
 - RichiH: Cortex & Thanos are also proving grounds for new tech which can be fed back into Prometheus
 - Matthias: Monitoring probes/caches which get disconnected can sync their data
 - Bartek: What about block upload API?
 - Frederic: Block format API becomes standard.
 - Julien: Remote read-write could replace federation
 - Frederic/Brian: Bad idea, worst of both worlds
 - Matthias: If I don't use one component, there's no extra cost to having it around
 - <more back and forth I couldn't capture>

<16:01 Simon leaves>

- CONSENSUS: We want to support backfilling over the network at least via blocks which do not overlap with the head block
- NOT CONSENSUS: We want to support backfilling over the network at least via streams which do not overlap with the head block
- CONSENSUS: We would like to support backfilling over the network via blocks which do overlap with the head block given proper design

<16:14 Matthias leaves>

RichiH: OpenMetrics status

- Prose version being condensed into IETF tightness
 - Will share the working doc with prometheus-team, still not consistent

- Wordsmithing is exposing many old or hidden disagreements, needed for eventual consensus
- People involved & still doing work:
 - Ben, Brian, Rob Skillington, RichiH
- Started navigating IANA ports assignments etc
 - Plan is to see if we can get ports for Prometheus, Alertmanager re-assigned to us. One generic port for reverse proxying all integrations and exporters
- Working with OTEL SIG Metrics to ensure interoperability, ideally make OM a first-class citizen within OTEL
- Some concerns about OTEL's everything-and-the-kitchensink approach, in particular the three-stage metric emitting pipeline and requiring counters to be stored as deltas which are then rebuilt into counters as we know them. Performance, timing artifacts, etc. Am in the OTEL calls and am trying to find a balance between suggesting changes and not waltzing over ALL the things.
- Agreed that OTEL Metrics & OM Working Groups will meet

<16:20-16:45 various discussion, break, etc>

Beorn: Keep or remove the **vendor** directory?

- Frederic: Offline builds were convenient with vendor, but tooling has improved.
- Brian: Mostly a call for Ben & Simon as maintainers of build
- Matt: General community moving towards removing vendor
 - gohack tool makes it easy to override dependencies without modifying vendored code: <https://github.com/rogppe/gohack>
 - Some annoying eventual consistency issues, occasional need for GOPROXY=direct to bypass the proxy if you push a repo and immediately need to update it in another module
- **CONSENSUS: Remove it in node_exporter, see if we like it**

<16:52-16:57 Goutham drops off and rejoins>

Prometheus 3.0: state of affairs

- RichiH: Initially proposed this to enable discussions about breaking changes and about how to break stuff, not if to break stuff; the intention is unblock us mentally on a team level
- [2019-05-24 26 Prometheus 2019 Dev Summit 1 - Prometheus 3.x](#)
- RichiH: Please read & comment on "Prometheus is not feature complete" (non-public brainstorming document) before next dev summit

Julien: Alertmanager custom webhooks / body template: discussions before casting a vote on the ML

- Brian: making the webhook sender work with arbitrary endpoints (and error handling etc.) would be difficult
- Julius: some kind of customizability of the contents would be nice (templating features in AM are useful), but not sure how exactly

- Brian: Existing AM templating shouldn't be used to generate JSON, gets messy, but there's a PR somewhere that adds sane-ish templating support for the webhook receiver
- Stalling the point as Simon is not in the call any more

Review the community communication mechanisms. As of today, we have:

- Three mailing lists
- Two IRC channels
- One Slack channel (unofficial)
- One subreddit (unofficial)
- IRC/Matrix issues? Nickserv identification is an issue, can we turn it off?
 - Was from a spam attack 18months ago
- Some people blocked from mailing lists
- Need google account for mailing lists, unintended effect of avoiding having to manually accept multiple false-positives per day
- <Lots of bikeshedding about everyone's favourite messenger>
- **CONSENSUS:**
 - See if it is possible to subscribe to our MLs without a Google account & enable this; still require subscription for posting
 - Link to public ML archive in our docs/community section, and explain how to subscribe without Google account
 - Disable identification requirements in IRC channels to make using Matrix easier; revert if there are any attacks

RichiH: Better documentation; split into reference, documentation, and howtos/tutorials

- The first thing I disliked about Prometheus in 2015 was the documentation, in 2020 I despise it. It's not human-friendly and almost useless if you don't already know what you're doing or at least have all the concepts down.
- RichiH: I propose three sections
 - Reference, what we have today
 - Human-friendly documentation (verbose reference & examples)
 - How-to/tutorials; a choose-your-own-adventure
- RichiH: I think we are setting the bar too high for documentation changes. This drives away contributors.
 - Frederic: Level of rigor for reference is appropriate, new things could have different bars
 - RichiH: Agreed.
- Brian: Need to make sure to keep the sections apart to not make a mess
- Julien: Link from manual back to reference
- Brian: What about highly specific use cases in PRs for docs?
 - RichiH: Will need 1-2 examples per thing anyway, so we would accept the first 1-2 examples and then stop. Otherwise nothing I am concerned about as we can always reject PRs if they don't add value.
- Frederic: Suggest less strict link stability for user manual and tutorials

- Julien: Redirects work
- RichiH: Reference links should “never” change, manuals for major versions, tutorials are somewhat malleable
- Brian: Don’t enforce that links can only break links on major versions
- RichiH: Agreed. Not intended as a hard rule, but as a guidance, more like a sniff test
- Brian: Who will own user manual?
 - RichiH: Willing to own coordination, ask Grafana & CNCF for tech writer time; suggest we also ask within Red Hat, GitLab, etc.
- **CONSENSUS:**
 - We will restructure the documentation into three sections
 - User manual, default, low threshold for contribution
 - Reference, what we currently have with PromQL examples of in/output
 - Guides, preferably on a modular basis
 - We will ask Diana Payton for a style guide, overall concept, etc. for all documentation
 - We will try not to break links unless it really makes sense

<18:03 End>