Prometheus 2018 Dev Summit

Saturday August 11 2018
09:45 - 18:15

Google, Tower 1, 6th Floor, “Isar Valley”
33 Erika-Mann-Straße
München, Germany

Discussion Topics

Last Meetings

- https://docs.google.com/document/d/1DaHFao0saZ3MDt9yuuxLaCQg8WGadO8s44i3cxSARcM/edit#heading=h.bbvw42ndm0nb

Attendees

- Richard Hartmann - Organizer, Lead
- Krasi Georgiev < kgeorgie@redhat.com > < http://github.com/krasi-georgiev >
- Matthias Rampke <mr@soundcloud.com>
- Ben Kochie <bjk@gitlab.com>
- Simon Pasquier <pasquier.simon@gmail.com>
- Tom Wilkie <tom.wilkie@gmail.com>
- Brian Brazil
- Max Inden <IndenML@gmail.com>
- Paul Dix <paul@influxdata.com>
- Gianluca Arbezzano <gianluca@influxdb.com>
- Luc Perkins<lucperkins@gmail.com>
- Christian "BlueCmd" Svensson <blue@cmd.nu>
- Chris Marchbanks <chrism@freshtracks.io>

Meeting Notes

<Start at 0945>

Security audit by cure53 & our security model

- RichiH: Security audit by cure53 & our security model; not happy and we should improve here
- Brian Brazil: One of the issues: shutdown POST request can be pulled as e.g. image link
- bbrazil: Security headers incomplete, can add those
    - Explicit decision ~3 years ago to not require auth
    - RichiH: Old consensus can still be changed by new consensus
        i. eg API token
    - bbrazil: Issues with handling lifecycle of API tokens

        i. bbrazil: Any breaking changes -> would need to go into 3.x
    - Matthias Rampke: Make security optional in 2.x, default to secure in 3.x
    - TomWilkie: What are the attack vectors?
        i. Prom behind Grafana
        ii. Public internet, LAN, etc
    - Blue: Delete endpoint exists?
    - bbrazil: Yes, lifecycle endpoints optional now, but often needed
        i. Must turn them on on Windows
    - Matthias Rampke: pre-shared-key on CLI as 80/20?
    - jrv: CORS problem
        i. jrv: How do you set the valid origins?
        ii. bbrazil: Will be a flag to list the valid origins
    - bbrazil: Alertmanager is more vulnerable as some mutating endpoints are required for AM
        i. TomWilkie: CORS should be the same
        ii. bbrazil: AM new API could do this first, post vs put
    - SuperQ: there's no default way to do this.
    - BlueCmd: Option for "Yes I'm sure" button
    - TomWilkie: Yes, we should fix this
    - CONSENSUS: Yes, we plan to fix this - CORS header configurable, appropriate verbs

Scrape Security

- SuperQ: In the past, we said no to TLS, auth, etc
    - Propose that we support TLS & client certs as a minimum (for everything)
    - Add it as an option for the client libraries we use.
    - bbrazil: Please do Golang first, will follow with Java and Python with whatever we end up with
    - TomWilkie: Mark it experimental as this is a can of worms
    - SuperQ: Do exporters need to support HUP?
    - bbrazil: Not necessarily
        i. JMX exporter can't take a HUP so it watches mtimes
    - TomWilkie: What are we guarding against?
        i. SuperQ: Snooping on the line
    - TomWilkie: Why do we have transport sec, without auth?
    - <lotsa back and forth, re-focus on just transport security for now>
    - Matthias Rampke: Need to reload the certs, how?
        i. bbrazil: windows has no HUP
        ii. Matthias Rampke: Try it in node_exporter
        iii. bbrazil: Yes, but should be a library, and needs to be shared everywhere
    - bbrazil: Want to have good plan to make sure what we do won't break anything we might not even have control over (SIGHUP on Windows, etc)
    - RichiH: Need a decision tree, like if you're on X, do this. Best practices for the reloads.

- Matthias Rampke: qq, do we want this, where do we want this, and how do we do this
- Matthias Rampke: Let's just do this in one place and say, don't copy this!
- PaulD: Don't do this for direct instrumentation
    i. Do this only for the exporters.
- RichiH: We have consensus? Let's start with node_exporter with big warnings.
- FB: Let's just do this somewhere, client_golang/NE, whatever. Let's just do this.
- bbrazil: Weird TLS option requests. Do we just do defaults?
- SuperQ: Yes, defaults and ONLY defaults.
- RichiH: If defaults are wrong?
    i. SuperQ: we update the docs, and say here's the new thing.
- RichiH: Build time not runtime, hence we need to rebuild everything if defaults are wrong
- bbrazil: But the defaults are fine anyways, now.
- Matthias Rampke: If the defaults are not fine, hypothetically, what do we do?
- bbrazil: Cloudflare did come up with some weird request, but we'll just reject it.
    i. bbrazil: Ask them to carry a patch on their side to keep public code clean
- RichiH: What if we have some mismatch of the target and prometheus?
- RichiH: Will prometheus reject the ooooold targets?
- bbrazil: We just turn off the TLS if there is a mismatch? Or use this proxy
- Matthias Rampke: When do we make changes of the defaults then?
- bbrazil: Not when there is a valid use-case for someone, but when TLS is broken fundamentally
- SuperQ: We need TLS 1.2
    i. FB: But that's not Go default
    ii. SuperQ: We document and recommend 1.2
    iii. Matthias Rampke: We can say prometheus supports 1.2, but we cannot trust the exporter to support it.
    iv. Matthias Rampke: Direct instrumentation in an old enterprise service, cannot support it.
- Blue: What is the goal?
    i. SuperQ: Sane defaults
    ii. Blue: Right now we are doing this via proxies, and if we are talking sane defaults, we shouldn't be discussing the esoteric options.
    iii. SuperQ: Yep, sane defaults only, no esoteric options.
- <topic changes towards how quickly we can push security fixes to binaries if need be>
- bbrazil: What if TLS is broken. How do we release 18 binaries?
- SuperQ: Just like any go vulnerability, we just gang up and get it done.
- SuperQ: Just like any vuln, this doesn't change because it's security.
- bbrazil: What expectations do we set to users?
- SuperQ: "soon"
- TomWilkie: Security conscious people don't trust us anyways, CI is flawed.
- bbrazil: Yes, but higher expectations for security

    - Matthias Rampke: We can have a short deadline for prom, bb_expo, node_expo, for everything else it's "soon"
    - bbrazil: 24hrs is not realistic, cuz no one on-call, 7 days more realistic it
    - Blue: But it's OSS, just compile it might be the answer
    - ??: But we need to remove the vuln binaries,
        i. SuperQ: if we remove them, we break the world
    - RichiH: We need to be clear about expectations, and if someone is not happy, they are welcome to chip in.
    - BJK, bbrazil: 7 days, prom, AM, nE, bbE
    - Matthias Rampke: If they are not happy with 7 days, they can do it themselves
    - RichiH: Yes, but let's also be open about it.
    - TomWilkie: If you are security conscious then don't trust our binaries, just build it yourself. This advice shouldn't change.
    - bbrazil: Because of this, there might be breaking changes. We need the right messaging
        i. bbrazil: Needs to go into security model
    - Blue: Can CNCF help?
    - RichiH: Asking CNCF to have a full-time security team with full knowledge of the projects is not realistic unless they limit scope
        i. Cover just for graduated projects?
        ii. Sane cut-off, and real, tangible benefit of graduation
    - bbrazil: Can we do this to get in the front of the 90 days instead the end
    - bbrazil: Security disclosure, just email -team@
- CONSENSUS: Best effort binary releases after any vulnerability; user promise:
    - 7 days
    - Prometheus, Alertmanager, node_exporter, pushgateway, blackbox_exporter
    - Policy: only touch last minor release
- CONSENSUS: RH to poke Chris about CNCF security team
    - And ask if we can get on the the 90 day responsible disclosure lists, if possible

Authentication

- SuperQ: Recommendation for AUTH on the exporters.
- RichiH: We cannot do PSK unless we fail on un-encrypted links
- SuperQ: We should not do bearer tokens
    - Matthias Rampke: But we should, to have a simple way
    - RichiH: It sucks, but it works 80/20
- Blue: Cert revocation?
- bbrazil: Nope, go doesn't
- Blue: Can we trust the certs then? Shorter lifetimes would make sense.
- SuperQ: We ain't talking about that, what is the auth method? PSK is not my fave, but it's easy, I lean towards bearer tokens.
- TomWilkie: Browsers don't do bearer tokens, so what do we do?

- SuperQ: Is there a way to do bearer token in the browser
- Matthias Rampke: Yes, let's just basic-auth with the bearer token as the key
- bbrazil: Where do we put the token, CLI, nope?
- Matthias Rampke: Yes, machine compromise is a bigger deal.
- jrv: Secrets on CLI, meehhh
- Okay, let's do secrets from file
- SuperQ: JWT in the browser, they are the replacement for OAuth, non-cookie way of doing auth.
- bbrazil: Bearer token in JSON
- SuperQ: HMAC signed, PSK is signed, so it's nice. JWT can be scoped.
- TomWilkie: JWT? Why? What's better?
- TomWilkie: Replay attack protection? JWT as a PSK is just a PSK, more complex than bearer token.
- MR, TW, bbrazil: JWT is over-engineering the problem
- SuperQ: More research needed. JWT has an RFC, that's nice.
- bbrazil: client certs are too hard for users, but does anyone have objections to just client certs.
- Matthias Rampke: Better to start with bearer token, client certs are too steep
- bbrazil: Nobody has objection to do client certs.
- TomWilkie: We want to start with basic-auth, because client certs are too steep
    - Scope: Don't guard against replay attack for now
        i. Some would need more, but let's focus here now
- bbrazil: How do we manage PSK rotation? Reload.
- RichiH: Option to build it in; valid for entities which bake their own binaries and does away with file management
- SuperQ: Do we need to support standard htpasswd and reload on file change
- RichiH: Do we agree that we can have more than one token
    - All: yes
- Matthias Rampke: Ignore the file perms
    - RichiH: Fine if we document it
- Matthias Rampke: We need to do HTTPS + BA, and that needs to be documented
- SuperQ: We need to ensure that, otherwise no point
- bbrazil: Too much opinionated. Don't require TLS for BA and also support client certs
- bbrazil: Expiry cert time must be a metric
- RichiH: Ask CNCF to fund cure53 review before implementation?
    - ALL: Yes
- CONSENSUS:
    - HTTP Basic auth, as we need to be able to hit /metrics in the browser
    - .htpassword format, filename on command line
    - Must support multiple passwords in the file for rotation
    - Don't care about permissions/owners on the file
        i. Put verbiage into security considerations
    - Don't require TLS to use basic auth

    - Client Certs will also be supported
    - RH to ask CNCF for cure53 review

<Lunch 1230-1330>

Do we want to sign releases?

- MI: we should sign the git tags.
- MI: Could have travis enforce this.
- bbrazil: Builds are not trusted.
- TomWilkie: What do we sign with? Personal gpg?
- RichiH: If we do gpg, then two signatures from other maintainers before being allowed to sign; Debian model
- bbrazil: Can we put this into sec-doc, that the tags are signed.
- FB: Can we sign the release tarballs.
    - RichiH: Yes, but it's not secure as no trust circle
    - RichiH: Devs should not release it themselves until they meet two others.
- Matthias Rampke: If you cannot release then you are not a maintainer. Signing with anything is better?
    - RichiH: Signing with "anything" is pretending to have more security than we have
- jrv: Hangouts for ID verification works, right? What's better in person?
    - RichiH: Debian has this requirement, they don't have travel funding, we do.
- RichiH: Do we agree that we need to have signatures on that key? How do we get the signature there?
- TomWilkie: Do we now need to sign stuff? What are we stopping here?
    - bbrazil: To stop GH/Circle from putting out a release.
    - TomWilkie: We need this, but is it now?
    - bbrazil: We do it incrementally
    - TomWilkie: Yes, just a lot of work if we do it right.
- RichiH: If you care about security, then grab the source and build it yourself.
    - RichiH: We should only secure the source, can't make promises for our CI
- TomWilkie: We should make the CI bork when unsigned.
    - MI: Start with just signing our commits, and then iterate.
    - Matthias Rampke: Fail if not signed.
- RichiH: For key management, put it in Maintainers, that's how we get it into the CI.
    - Obvious issue if GH compromised, could ask CNCF to carry an external keyring file
- bbrazil: We should do it, open an issue. But then we cannot make it a policy, until I figure it out if Python and Java also support this.
- RichiH: We should have some policy around this. Just signing something is not good.
- bbrazil: Cross sign when we can, but let's not mandate it here.
- bbrazil: Add more verbiage to the security doc.
- CONSENSUS:

    - Sign every single release tag (need to check how java does releases exactly, there's some signing of the actual build)
        i. bbrazil: Update: Java artifacts are signed, tags are not
        ii. bbrazil: Update: Support for this in maven was only merged the day of the summit, so don't expect this to happen anytime soon on the Java side
    - (RichiH: Not blocking consensus, but this is worse than useless if no policy about handling which keys are allowed to sign)

OpenMetrics Adoption

- RichiH: Any concerns to adding OM to Prometheus?
- bbrazil: Client libraries with content-negotiation and add support for the parser them.
    - What point do we change default?
    - What point do we drop off the prometheus format?
- RichiH: As soon as we have a OM in client then we should do it in prom via content negotiation.
    - Put it into a design doc
    - Prom 4.0 is when we drop prom format
    - Prometheus 3.x should have OM in the server and all the client-libs.
- Matthias Rampke: We should do Prom ASAP just to weed out the bugs.
    - And NEVER remove a format
- bbrazil: OM should have additional time-series that shouldn't be there for prometheus
- TomWilkie: Do we care about others that are not prometheus
    - Paul Dix: I don't care ;)
- RichiH: We just push prometheus format if no CN for now. 6mo, 1yr later we should default to OM
- TomWilkie: What do we see in the browser?
    - It would be nice to have OM
- Matthias Rampke: Make prom default for now, but later we flip. But let's make it an option in client lib.
- bbrazil: We need to be nice to the others in the ecosystem who use our parsers, but give them enough headspace for them to update their stuff.
- Paul: We just need to get them an upgrade path.
- RichiH: If we ask for OM, and we get prometheus? What do we do?
    - bbrazil: Reject it.
- bbrazil: What if we don't get the header from the client, then prometheus expo.
    - Matthias Rampke: It'll be nice to just try every parser
- RichiH: Let's give an option to the client lib to default to either
    - bbrazil: I don't know if I want that
- bbrazil: Contact everyone to send the right header.
- CONSENSUS:
    - We should prefer negotiating OM
    - Up to the client lib for changing the default

        i. Won't block anyone wanting to add this
    - 1 year to switch over the defaults
    - Email to dev & user list, Twitter, ASAP that they MUST send the header to request Prom format if they want to keep using it
        i. Also a transition doc

PromCon .5

- RichiH: 2 PromCons per year?
    - Locations
        i. Asia: non practical
        ii. North America:
            1. USA not a preferred choice due to visa issues
            2. Expensive in Canada, and to get there, but Vancouver is nice
            3. Mexico City?
    - RichiH: Need someone local (not necessarily Prom team) to keep the family style. No outsourcing.
    - TomWilkie: Grafana happy to help on the logistic side.
- ??: Canada is more expensive to get to than US
    - bbrazil: Current conference ticket prices are already sponsoring travel cost as they are so low
- RichiH: If no solution for a second PromCon, organize a 2nd dev summit?
    - All: Yes
- LucP: Envoy is attaching its first conference with Kubecon Seattle.
- CONSENSUS: We want to have a second PromCon if feasible
    - RH will try and come up with proposals

Piggy-bank

- CONSENSUS:
    - Order of preference
        i. Rainy day fund
        ii. Diversity funding
        iii. Dev summit (long weekend)

Promcon in 12 months:

- bbrazil: enough talks to do two tracks
    - bbrazil: we have to squeeze in talks due to single track, less time for breaks
    - TomWilkie: I very much like single track
- Matthias Rampke: talks very detailed and in depth this year, no beginner level ones

    - GV: Do a workshop a day before for beginners?
- jrv: would appreciate a community day, self organise with facilitation, segmented rooms, BoF etc
    - Tables in big room with topic etc
    - Constraints on venue
    - Extra day?
- Everyone: 20min talked worked very well.
- PD: dotGo etc in paris do 18mins then a speaker interview, like TED
- RichiH: My style of doing change overs to new speaker while still handling questions to previous one to maximize question time ok?
    - All: Didn't even notice, so good
- jrv: Workshop before / community day after?
    - Could do satellite event beforehand, perhaps not an official thing?
    - RichiH: Sysdig did this the evening before, got 80 people.
    - jrv: Would want introductory sessions next year.
- TomWilkie: Panels weren't very diverse. We can choose topic to make them more diverse. Prometheus user panel?
    - bbrazil: Few talk submissions from female speaker
    - Matthias Rampke: How as a community can we be more welcoming?
    - RichiH: Should we proactively invite a keynote speaker?
- bbrazil: Size of next promcon?
    - RH; 400-500
    - bbrazil: 350 to keep community
    - jrv: 500, don't want to keep people out
    - Pressure from wait list will lessen with PromCon NA
- Where are we going to do it?
- Consensus:
    - Stay single track.
    - Do satellite introductory event day before, semi-official, different location
    - Do community day after, official unconference etc
    - ~400 depending on venue

PromCon NA

- CONSENSUS:
    - Capacity? 300-450 depending
    - 4 months away from PromCon EU each way (Jan - March)
    - Want to try to do it in 2019, but its tight
- jrv: Lets do a more dedicated session on PromCon in the near future, online.

AlertManager v1.0.0

- Common user question: when can I use it?
- What needs doing before it “ready”:
    - New API (v2)
    - Memberlist library
- Want to set expectation.
- Matthias Rampke: Not a big fan of lining up features and then do 1.0, prefer if its the same as the previous version running for several months
- FB: We cannot break stuff after 1.0
- RichiH: Is 1.0 an issue? Can we tell them its stable but we need to be able to break things.
- SP: Don’t expect to not break
- TomWilkie: Can we call this 1.0 already?
    - MI: Nope, not stable enough
- MI: Let’s do semver so not delay 1.0 too much
- Matthias Rampke: If we can defer major breaking changes, what do we need to get 1.0 out, what stability issues
- MI: Clustering not stable
- TomWilkie: Mark clustering experimental, and call single node stable
- RichiH: Is it a problem if we have too many majors?
    - bbrazil: Yes, it means we are not stable enough
    - jrv: Yep, people needed maintained stable versions, so nope, don't release too many majors
- jrv: Get all breaking changes in ASAP and then stabilise.
    - 1.x should stay long
    - People should be able to depend on this
- Matthias Rampke: People should know that they need to not touch it for like 6 months
- SP: We broke flags, and peoples setups broke
- bbrazil: We shouldn’t bump major versions in less than an year
- FB: Is there feature work to do?
- TomWilkie: We should update the readme to say that it’s in production use at many companies.
- FB: Scalability issues. Need the details.
- CONSENSUS:
    - 1.0 is very important and we need more time to stabilise clustering and others
    - Only do 1.x when we are confident that we don’t need breaking changes any more
    - It’s ready when it’s ready

Alertmanager API v2

- MI: See lightning talk

    - Swagger
- Inconsistent current API with code as docs
- Client side API is autogenerated also with OpenAPISpec
- Matthias Rampke: What's part of the API
    - MI: Ingesting alerts, CURD on silences, status, others.
- MI: Implemented, needs instrumentation
- MI: We'll have to buy into openAPI eco-system and then generator
    - We can switch out the generator though
- TomWilkie: Quality of the generator?
- MI: It's quite good, am quite happy
- MI: No scalability tests on the generated code
- FB: Read through it, it seems similar.
- bbrazil: Can we shim in the json iterator library
- Luc: Docs are free, and we can add it to our prometheus docs.
- Matthias Rampke: Can we hand-write the code? Instead of the generator?
    - Yes
- MI: When do we need to deprecate v1?
- bbrazil: Prometheus 3+
- Matthias Rampke: How long do we want to keep 1.8 working with AM?
- bbrazil: We need consistency across the ecosystem
- bbrazil: Prometheus would need to have the same openAPI stuff
    - Which would mean it's v3
- Matthias Rampke: This is like experimenting with openAPI and if and only if this works, then expand.
- KG: We can test the API from the browser, that's really good.
- MI: We can have swagger UI being served by AM itself. (not prefered)
- MI: Webhook also is openAPI?
- MI: Stuart is very positive about the API.
- bbrazil: Webhook and templating is the same data structure, that's interesting.
- Tincho: Will this generated ALWAYS?
- MI: Nope, checked in, but will fail if spec is different.
- Tincho: Problem still exists because Swagger needs to be packaged.
- bbrazil: How about docs?
- Luc: Can easily do that once there is JSON source.
- Matthias Rampke: Can we add context around the APIs?
    - MI: Yes, you can add custom text
- KG: Can we do gRPC?
- MI: Nope, but it's super hacky to get both gRPC and HTTP at the same time as seen by Fabian in Prometheus
- CONSENSUS:
    - Everyone should go and review the PR
        - https://github.com/prometheus/alertmanager/pull/1352
    - No need to rush into this for Prometheus; lets see how it works out

<pause 10m>

Default alerting priorities

- bbrazil: If we have an example alert what should be the severity label be? This ties into mixins and we need to specify the severity "best-practices".
- TomWilkie: Completely customisable but we need a default
- TomWilkie: Options: page/ticket AND critical/warning
- Matthias Rampke: No way to ticket it. Everything is a notification, so I'm leaning to "c/w"
- TomWilkie: I can have critical alerts in dev, that'll never page.
- RichiH: We are putting two things in the same label: importance and urgency, should split them up
    - RichiH: Important things can be dealt with next business day, urgent ones call out
    - RichiH: e.g. an alert from dev env can never be urgent, so never paging people at night
    - TomWilkie: I don't want two sets of alerts
- <https://docs.google.com/document/d/1fpIdsAzS2c45TqSXjFHl0r5DY5yrcDEP5jmja46zmO4/edit is shown>
- Matthias Rampke: You mean have two different labels: importance, action
- TomWilkie: Have you ever needed 4 states?
    - All: Nope
- RichiH: I need info alerts, because I need system state in my overview as context when fixing things and AM for me is a good status-pagey.
    - Info alerts are pre-made hints on what to fix to make the big alerts go away, or indications of undesirable system state I want to look at before things go boom
    - This is my workbook. Start at the top, work towards bottom, then I know everything is healthy
    - Matthias Rampke: I find the 2 dimensions really interesting. But if I write an alert, I don't want to explain it.
    - bbrazil: I agree, these are largely examples, but need good ones to subtly also tell people they can do it whichever way they wanted.
    - RichiH: We should also help educate people how to do ops properly? 2 dimensions is proven to be better.
    - TomWilkie: Yes, but I don't think it's an 80% usecase. We should not put it in the defaults.
        - RichiH: Not convinced, also changed the world with n-dimensional label sets on data
    - RichiH: 2 dimensions should go into the reference, not push them into best-practices.
    - bbrazil: If I look at an alert, I want to see if it'll page me. Reading "page" helps!
    - Matthias Rampke: I look at how bad, so I like critical.
- TomWilkie: I don't know how to have alert that says page but doesn't page me.

    - Matthias Rampke: Yes, now need to know that dev-page doesn't page :)
- bbrazil: critical can mean a lot of things, like critical, super-critical, so on.
    - Have a customer that has several hundred critical going on all the time.
    - TomWilkie: Thats doing it wrong
- TomWilkie: Still can't get around how page doesn't sometimes not page.
- Matthias Rampke: If we don't have ticketing, it's wrong to have ticket. Also think "w/c" has better separation of concerns. 2 dimensional is interesting but not 80%
- bbrazil: How do we make sure users do it right?
    - All: moar docs!
- Matthias Rampke: If we document this is how to do alerts, it doesn't need to be unique, maybe explain the customisations better.
- RichiH: If mixins have something with 2 dimensions, it'll be easy to just use it with one dimension
- RichiH: IMP-URGENT matrix is better.
- Matthias Rampke: This is the first time I'm hearing all this. This is not general knowledge. I like this, but we need to talk to SRE as a wider community before we put this into reference docs.
- MI: Its interesting, multiple critical alerts and I as a person say how critical they are.
- Chris: This is the default, it should be easy to consume.
- RichiH: ref docs are simple, but mixins should have 2 dimensions
- TomWilkie: What are the defaults then?
- RichiH: If I can use 2 dimensions in mixins, then it's good. I need time to play around with the mixins.
- RichiH: If mixins can do both approaches, then everything is fine.
- Tincho: Suggestion - Example of overriding mixins with 2-dimension alerting scale
- CONSENSUS:
    - Reference docs as simple as possible
    - warning/critical levels will be used in docs and mixins as the defaults
    - Mixins will be capable of overriding with one or two dimensional models, and an example will be provided.
    - Richi will write up and educate us on 2d model.

Streaming APIs

- Krasi: do we need it?
- bbrazil: where?
    - remote_read or query_range?
    - Remote_read endpoint
- TomWilkie: Remote read is weak and not a lot of people picked up.
- TomWilkie: Love the streaming API that thanos does, adopting it in Cortex.
- TomWilkie: I'd love to implement the streaming but can't commit to it.
- TomWilkie: gRPC is a bit heavy for prometheus IMO.
- bbrazil: I don't see the need to replace it yet.

- Krasi: what about promQL?
- bbrazil: Unclear about any wins, could increase or decrease memory usage
- TW/bbrazil: Implementation is going to be complicated.
- bbrazil: We cannot return partial requests, we need to error out early. Hence we need to marshal the full-response.
- Matthias Rampke: Streaming API could be a full new API, which could behave differently.
- TomWilkie: Yeah, but we don't need to do it now because it's complex.
- bbrazil: OOMs will be fixed by protections.
- TomWilkie: RR endpoint is cool, but no use-case.
- bbrazil: Paul wants it
- TomWilkie: Yeah, it's a toy, I don't think we need to worry too much about it.
- bbrazil: We can be smarter about the RR and stream.
- FB: Thanos uses it, hence it's good to do streaming in RR
- Matthias Rampke: Looks like while there are technical issues, we want to do it
    - All: yes
- TomWilkie: Can we add the thanos Store API?
    - bbrazil: Maybe, not sure, yet another API
- CONSENSUS:
    - Nobody is keen on this for PromQL
    - For remote_read endpoint, sure we want to improve that.

Prometheus Release management

- GV: I messed one up.
- GV: want a consistent release management process. Didn't do one for 6 month, changelog was too big. Lots broke.
- GV: informally enforce release timeline would be good
- FB: some agreed cadence would be good
- FB: lets have pre-releases
- Matthias Rampke: I like actual prerelease binaries that I can test without having to buld my own
- bbrazil: what kind of cadence should we have?
    - 6 weeks, monthly, 4-8?
- What would pre-release look like?
    - Prerelease every 6 weeks, give it a week
    - The prerelease becomes release
    - If it breaks, fix and put out another RC
- FB: would like to have someone on releases duty every cycle
    - To do patch releases
- TomWilkie: if we branch for release management, and master continues to diverge, stabilizing a release could be hard
- CONSENSUS

    - RC every 6 weeks
    - While in RC no new features in master until stable
    - RC phase is however long, if we've not seen issues after 3 days of testing, it becomes a release.
    - Beginning of each cycle ask for volunteer to do release.
    - Goutham to document all this

Moratoriums

CONSENSUS:

- Want to avoid them
- Feature freeze part of release process now.

Alertmanager Clustering

- MI: Issues with mesh library
    - Switched to memberlist
    - Seems not to fit, size of data we're gossiping is big.
    - Users asked for SD on this
    - Should we implement our own?
- TomWilkie: What were the issues with mesh?
    - FB: Couldn't have different ports
        - Used TCP connection being up for health
    - fabxc: Hard to debug
        - WW not responsive on issues
    - Matthias Rampke: found it confusing & mental overhead
        - Discover of alertmanagers between themselves is weird
        - Vendoring prometheus SD reduces cognitive load
- TomWilkie: this is a hard problem, think it is naive to think we can do a better job ourselves
- TomWilkie: If someone is willing to do it, we shouldn't stand in their way.
- FB: why do you want to use SD?
    - Its mostly useful for metadata
    - DNS works
- Matthias Rampke: what are our expectations?
    - 100 nodes? Global latencies?
    - What is the alertmanager clustering story?

OpenAPI or Go Kit for the Prometheus API

- Lets not do anything until we're happy with alertmanager.

Subqueries

- RichiH: We use recording rules to hack in subqueries
    - No lookback, need to prepare data well in advance
        - Bad user story
    - Also, we can't backfill
- TomWilkie: eg rate(sum(foo)[1m]))?
- bbrazil: Composing range extra functions?
    - Can be done via recording rules, but this sucks
- Matthias Rampke: why don't we have them?
    - bbrazil: we should have backfill
        - It would have been to slow, but now it might work better
    - Subqueries need deeper consideration
        - What is the step etc?
- TomWilkie: don't want to block this feature on other issues
- JH: https://github.com/prometheus/prometheus/issues/1227
    - bbrazil: prefer option 2
    - Matthias Rampke: prefer option 1
- TomWilkie: don't see why we need step
    - RichiH: TSDB has info about when to switch over to new datapoint, thus arbitrary resolutions work, etc. Subqueries needs to know when to switch over as data generated on the fly
- Matthias Rampke: make explicit step optional, but have default
    - RH/bbrazil: use global evaluation interval
- CONSENSUS:
    - We want them
    - We should do it
    - https://github.com/prometheus/prometheus/issues/1227 Option 1
        - BB will think it through to see if there are issues
    - Default stepping to global evaluation interval

Data analysis

- RichiH: prometheus is for alerts, but with thanos it's becoming useful for data analysis.
    - We tell people to go to R or python, but that sucks
    - Do we want to enable more of this?
- bbrazil: such as?
    - RichiH: statistics functions
- bbrazil: they need huge ranges of data, OOMs and timeouts
    - RichiH: also the 11k problems
    - jrv: You can get data out without limits
        - FB: make this better known

- FB: make it easier to get the data out
    - Matthias Rampke: Do libraries exist?
    - We should link and promote it, include a section in the docs. Do a blog.
    - We should involve people who do this in the community who do this.
- CONSENSUS:
    - Preferred way: make it easier to get data out and write best practices
    - Blog post about Prometheus & data science as a call to action
        - Maybe also with feedback on what people would like to see

Annotations, i.e. non-identity-giving label set

- RichiH: want annotations on time series
    - attach data to a time series without changing the identity of the timeseries
    - ie build version, type, help string
- TomWilkie: what if the annotation changes?
    - Scrape at time X and Y - value in time series would be different
- bbrazil: do you want first class metrics and targets?
- SuperQ: Last wins for annotations is probably good enough
- <lots>
- RichiH: If we don't put build info etc in there, overhead is too much for just HELP & TYPE
- RichiH: Was also intended for namespaces
- CONSENSUS: Skip annotations

Namespaces

- RichiH: Do we want them, and how?
    - E.g. NAMESPACE metricname{} == metricname{__namespace__="NAMESPACE"}
- Slash for namespaces clashes with division operator
- <lots>
- RichiH: As history:
    - I started OpenMetrics to force more traditional vendors to support Prometheus
    - Second intention came later: support more use cases and to extend usefulness even if Prometheus does things differently
        - E.g. exemplars
    - Not an argument in any direction, just context
- Fabian: Namespacing is a big difficulty in prometheus and we need to fix it.
- bbrazil: No real problem
- Fabian: Stackdriver, but I personally care, even as I was an intern :)
- Matthias Rampke: Statsd and Graphite exporters will become super nice
- Matthias Rampke: dots might work
- Fabian: Yes, underscores are not enough
- Matthias Rampke: Yep, dots will work and nothing changes in the storage and anything else.
- SuperQ: We need to document it right.

- bbrazil: That'll break OM and our current data model
- TomWilkie: If OM is asked and then old format is asked, then what is returned? How will show namespace?
- RichiH: Empty namespace
    - TomWilkie: Not a good idea, as it should return `_`
- SuperQ: We could make it so that if we add a namespace to the the name, we break the old format.
- RichiH: How should it look? If we add namespaces
    - Io.prometheus.node_exporter.node_exporter_foo << RichiH likes this one
    - Io.prometheus.node_exporter_foo
    - Io.prometheus.node_exporter.foo
    - TomWilkie: It'll be a real pain to type namespaces in queries.
    - Fabian: We can fix the UX maybe.
- Fabian: Works great for Prometheus, but then maybe are we holding people back?
    - TomWilkie: Maybe, but it's okay?
- RichiH: We need to have ns but optional in the queries
- jrv: Can the ns be job?
- Matthias Rampke: All my exporters also expose http_reqs_total
- fabxc: Namespaces allow the same metric name to come several times from an exporter
    - RichiH: I have this issue at work with Ruby people, this would solve it
    - Matthias Rampke: I would like to expose http_latency, but the client library hogs that name
- TomWilkie: Can we just use dot?
- SuperQ: UTF-8 metric names
- GV: Just add a namespace label and add UX for the namespace selection.
- RichiH: No need of namespace UX, just add it in front of the metric.
- TomWilkie: Can make it ambiguous as what is selected is not clear. So to make it work, always include the namespace.
- Fabian: Not opposed to dots. Doesn't totally fix it, but it's a step forward.
- TomWilkie: UX issue, the exposition and the promql being the same metric works very well.
- fabxc: OM is prometheus heavy and other parties took a step back, but we need to accommodate everyone.
- RichiH: Yes, true, but then that is because of the adoption already.
    - How to namespace, io.prometheus or "kubernetes.io/", with a slash
    - bbrazil: But slash breaks promql query and the exposition is not good enough.
        - RichiH: Would we be willing to change promql too much?
    - bbrazil: That is borgmon style, and that is a very bad feature causes confusion.
- Matthias Rampke: Would purely dot based namespaces be enough for other people?
    - / based ns are nice, but also has other implication
    - Good argument that / is a div operator and makes sense in k8s labels but metrics are more mathematical
- Tincho: Currently ns is the name, but how much of a first class do we want to make ns. Is it just dots?

- bbrazil: How is _ better than ., because io_prometheus_ also works?
- RichiH: __, double underscore is where the ns ends and metric starts. How about that
- RichiH: ns with / is useful but . is not really all that useful to make the effort.
- CONSENSUS: Need more discussion on implications and good way forward

Event sink / TAC / Outalator <> Grafana plans

- TomWilkie: Outalator in Grafana. Grafana alerting is also not that great.
- TomWilkie: Share design doc and work on it over the next few months
- TomWilkie: Also have WYSIWYG alert editor against cortex
- FB: Can you make the alert editor generic?
    - Yes
- RichiH: Has this TAC thing.
- RichiH: Need an alert sink somewhere, but prefer it if in AM
    - MI: Not keen on adding state to AM
    - RichiH: Put it in between
- fabxc: Are you going to deprecate the default UIs of prometheus to TW?
- TomWilkie: Yes, we think we can do better, but we also want to contribute back. And there should also be these UIs existing.
- RichiH: Where is state stored? Not grafana
- TomWilkie: Yes, I think that might what will end up happening.
- TomWilkie: Grafana <3 Prometheus, and we will make sure to be not evil.
- RichiH: Like all other components with a simple UI, standalone, as a baseline within Prometheus
- TomWilkie: Grafana engineers might not make the prometheus UIs better. But it's all OSS.
- RichiH: Would backend live in Prometheus org / stay with Alertmanager?
    - TomWilkie: Probably, but can't commit
- RichiH: Think it's fair that Grafana will not write UI for anything other than Grafana
- fabxc: Want to understand if Grafana will deprecate all Prometheus UIs
    - TomWilkie: Would like to see exploration to be as components
        - Can make components reusable
        - Lots of stuff to do, so would have hard time selling this internally
    - TomWilkie: I would veto any of this becoming part of paid Grafana
    - fabxc: Still not clear what the five year plan is
    - Matthias Rampke: What's the alternative
    - RichiH: Would we get time from anyone else to do basic UI?
        - Chris Marchbanks: Will take it up internally with Bob, etc
- CONSENSUS:
    - We always want to have a first-class Prometheus.* UI
    - No one should hold anyone else back on improving things

Prometheus UI

- CONSENSUS:
  - We should unblock on the UI discussion, but not right now because ELM faction is not here

<End at 18:16>

Discussion Topics

- RichiH: Security audit by cure53 & our security model
  - Securing prometheus/prometheus at least?
  - https://github.com/prometheus/docs/blob/42bb621a0d8bdfb12da7af4ee260fda25e0c072e/static/downloads/2018-06-11--cure53_security_audit.pdf
- RichiH: OpenMetrics adoption
- RichiH: PromCon .5?
- SimonP: AlertManager v1.0.0?
  - List missing features/development that prevent us from releasing a stable version.
- Mxinden: Alertmanager API v2 design
  - https://github.com/prometheus/alertmanager/pull/1352
- Default alerting priorities
  - Motivated by mixins
  - warning/critical vs ticket/page
  - See https://github.com/prometheus/prometheus/pull/4474#discussion_r208197596
  - https://docs.google.com/document/d/1fpIdsAzS2c45TqSXjFHl0r5DY5yrcDEP5jmja46zmO4/edit
- Krasi: Streaming APIs?
  - https://github.com/prometheus/prometheus/issues/3690
- Release management
- RichiH: Event sink / TAC / Outalator <> Grafana plans
- Moratoriums: Let's avoid them (bjk)
- Mxinden: Alertmanager cluster library
  - Retrospective on cluster library change and future steps
- Krasi : OpenAPI or Go Kit for the Prometheus API
  - Allows generating client on any language
  - Auto generating doc for the api
  - https://github.com/prometheus/prometheus/issues/3416
- RichiH: Subqueries & data analysis (see Paul's talk)
- RichiH: Annotations, i.e. non-identity-giving label set