@PrometheusIO

Prometheus

Shaping Metric Monitoring in 2020

Deep Dive

Goutham Veeramachaneni
@putadent

Bartłomiej Płotka
@bwplotka

Hello!

Goutham Veeramachaneni

Software Engineer @ Grafana Labs

Prometheus and **Cortex** maintainer

Co-author of **Loki**

@bwplotka
@putadent

Hello!

Bartłomiej Płotka

Principal Software Engineer @ Red Hat

Prometheus maintainer

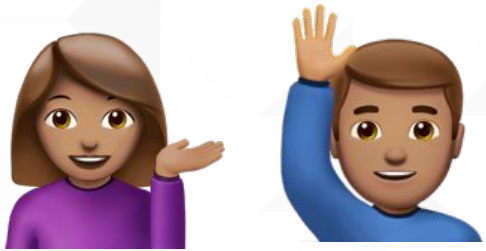
Co-founder of **Thanos**

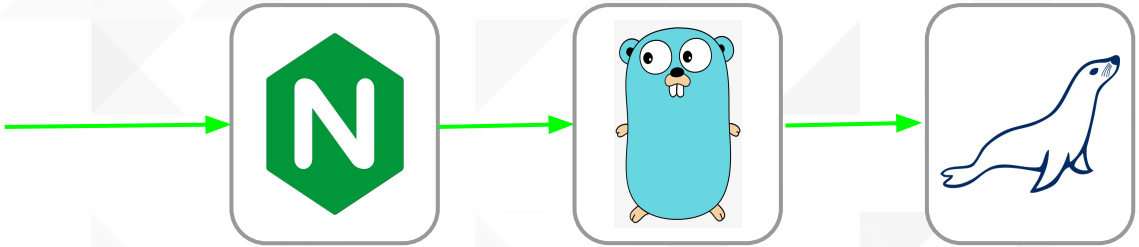
CNCF SIG Observability Tech Lead

https://github.com/cncf/sig-observability

@bwplotka
@putadent

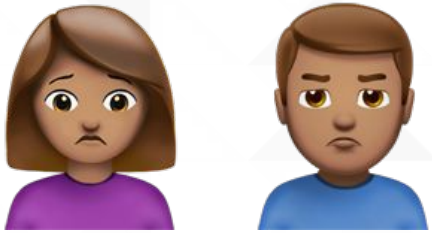
Let's start with a typical team

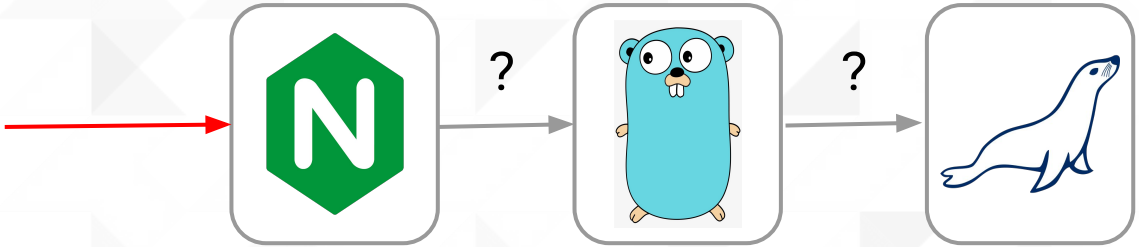




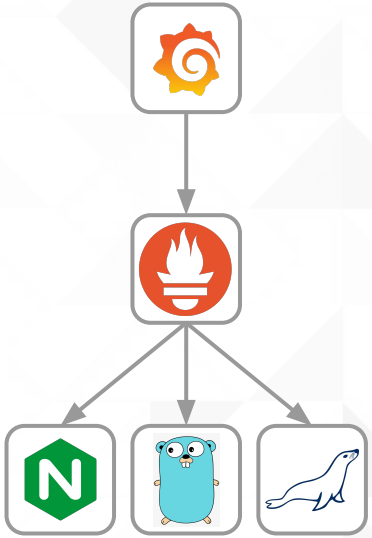
@bwplotka
@putadent

Things are great until they're not...



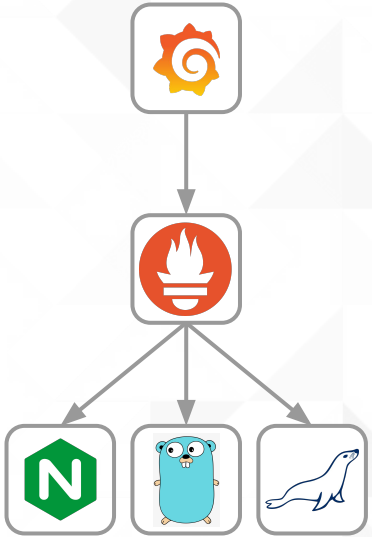


Add some Prometheus magic!



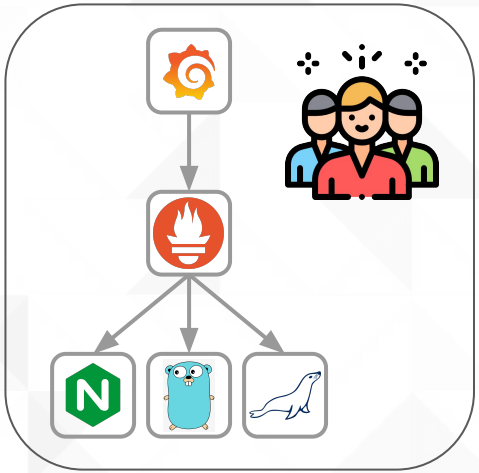
@bwplotka
@putadent

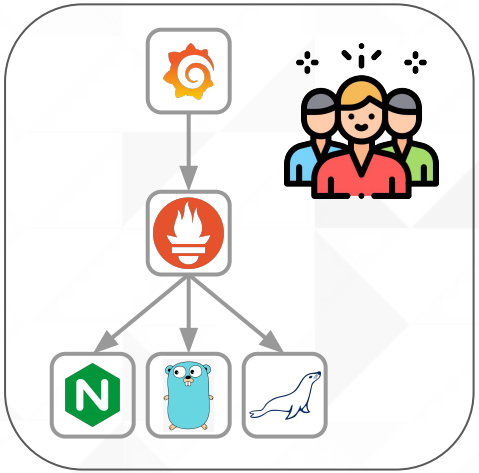
… with Dashboards!

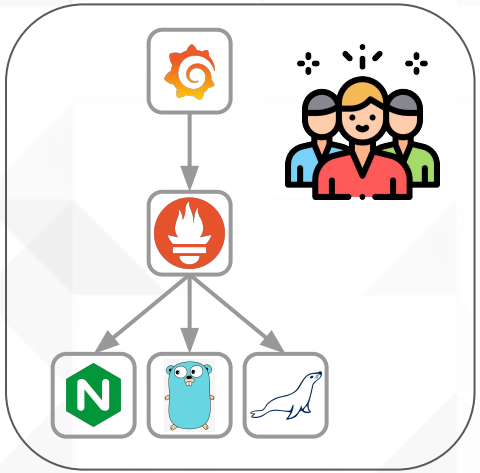


@bwplotka
@putadent

Slowly adoption grows!

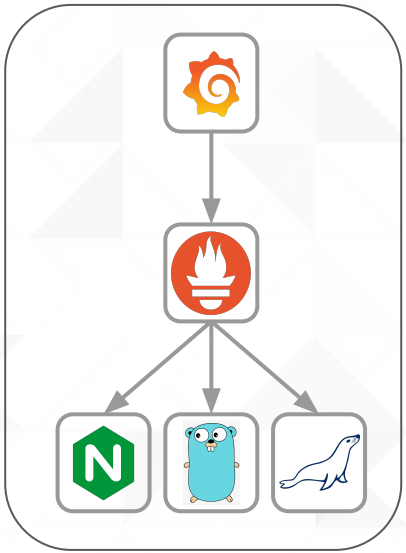




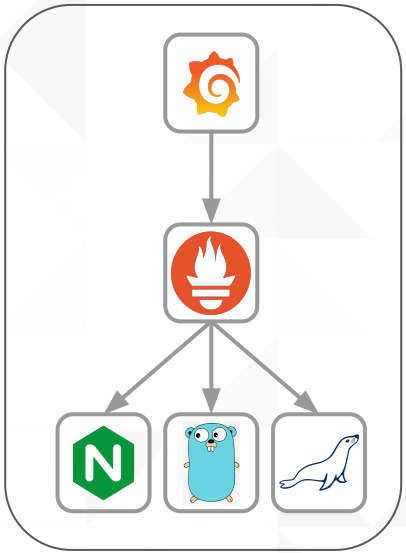


@bwplotka
@putadent

So does usage!



us-central-1



eu-west-2

@bwplotka
@putadent

Global View

How do I combine data from **two Prometheus servers**?

@bwplotka
@putadent

Global View

How do I combine data from **two Prometheus servers**?

sum by (handler) (rate(http_requests_total{code=~"5..", job=~"app1"}[5m]))

Execute - insert metric at cursor -

@bwplotka
@putadent

Global View via Query API?

sum by (handler) (rate(http_requests_total{code=~"5..", job=~"app1"}[5m]))

Execute - insert metric at cursor -

???

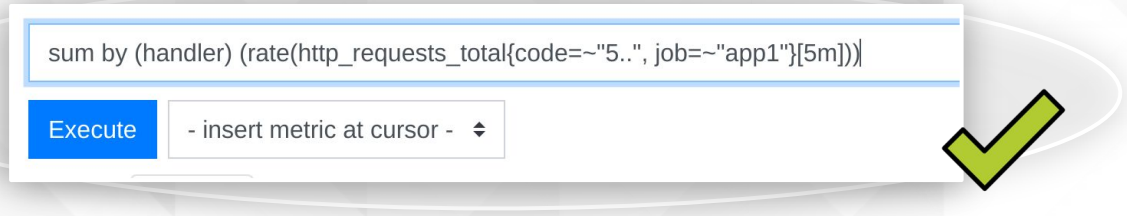
/query_range?**query='sum by (handler) (rate(....'**

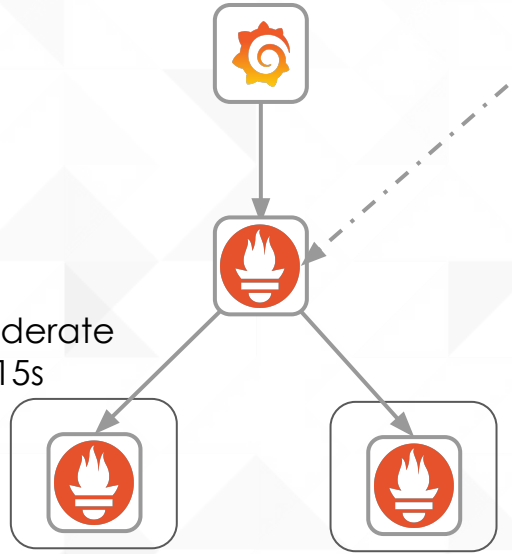
us-central-1

eu-west-2

@bwplotka
@putadent

Global View via Federation






```
scrape_configs:
  - job_name: 'federate'
    scrape_interval: 15s
    metrics_path: '/federate'

    static_configs:
      - targets:
        - 'us-central-1.prometheus:9090'
        - 'eu-west-2.prometheus:9090'
```

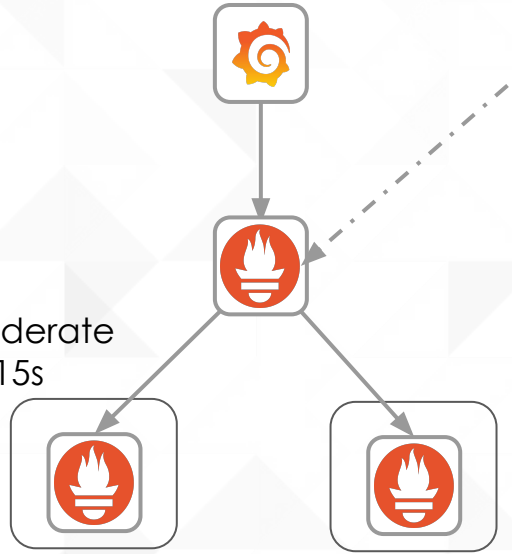
@bwplotka
@putadent

https://prometheus.io/docs/prometheus/latest/federation/

Global View via Federation

sum by (handler) (job:handler:http_requests_errors_total:rate5m:sum{job=~"app1"})

Execute - insert metric at cursor -




```
scrape_configs:
  - job_name: 'federate'
    scrape_interval: 15s
    metrics_path: '/federate'

    params:
      'match[]':
        - '{__name__=~".*:.*"}'

    static_configs:
      - targets:
        - 'us-central-1.prometheus:9090'
        - 'eu-west-2.prometheus:9090'
```

@bwplotka
@putadent

https://prometheus.io/docs/prometheus/latest/federation/

Global View via Query API?

sum by (handler) (rate(http_requests_total{code=~"5..", job=~"app1"}[5m]))

Execute - insert metric at cursor -

???

/query_range?**query='sum by (handler) (rate(....'**

us-central-1

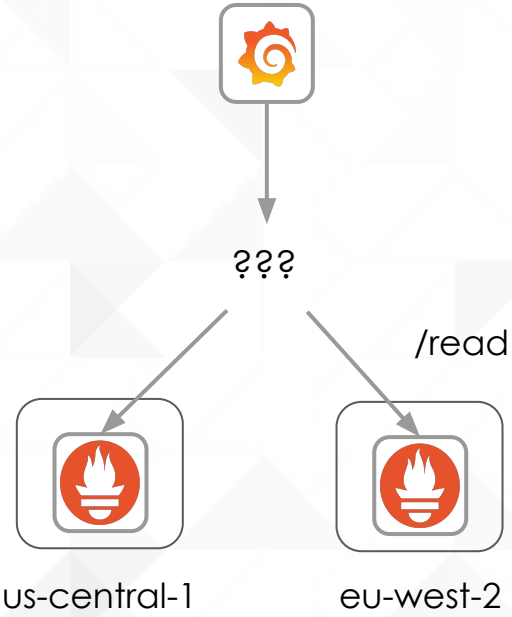
eu-west-2

@bwplotka
@putadent

Global View via Remote Read

sum by (handler) (rate(http_requests_total{code=~"5..", job=~"app1"}[5m]))

Execute - insert metric at cursor -




```
message ReadRequest {
  repeated Query queries = 1;
}

message Query {
  int64 start_timestamp_ms = 1;
  int64 end_timestamp_ms = 2;
  repeated prometheus.LabelMatcher matchers = 3;
  ...
}
```

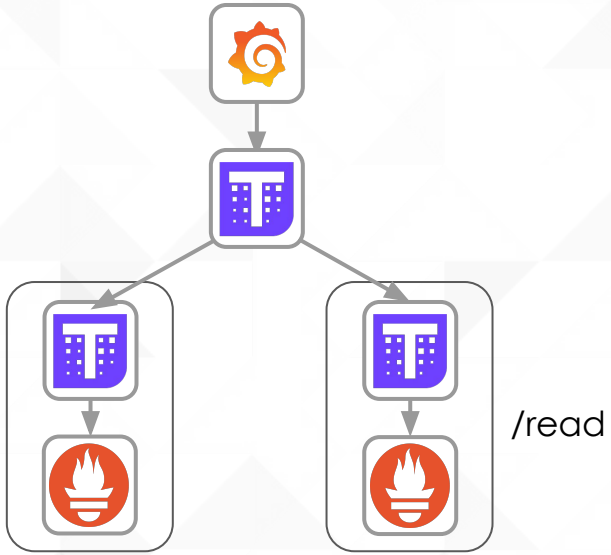
@bwplotka
@putadent

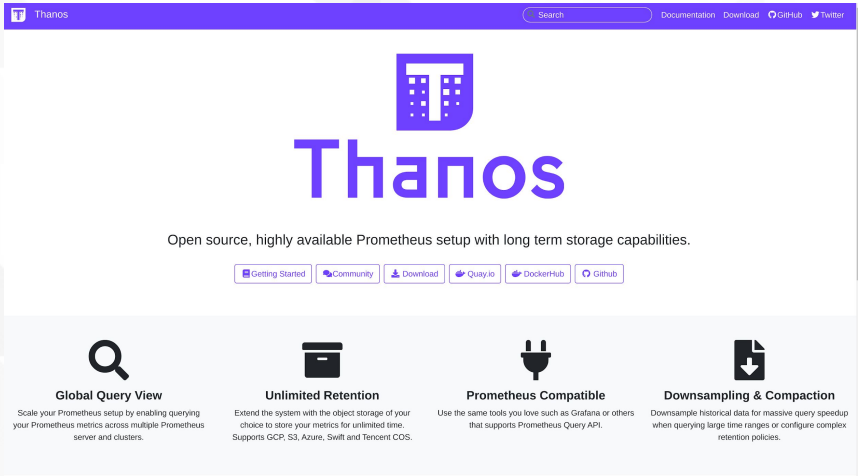
https://prometheus.io/blog/2019/10/10/remote-read-meets-streaming/

Global View via Remote Read (+ Thanos)

```
sum by (handler) (rate(http_requests_total{code=~"5..", job=~"app1"}[5m]))
```

Execute | - insert metric at cursor -





@bwplotka
@putadent

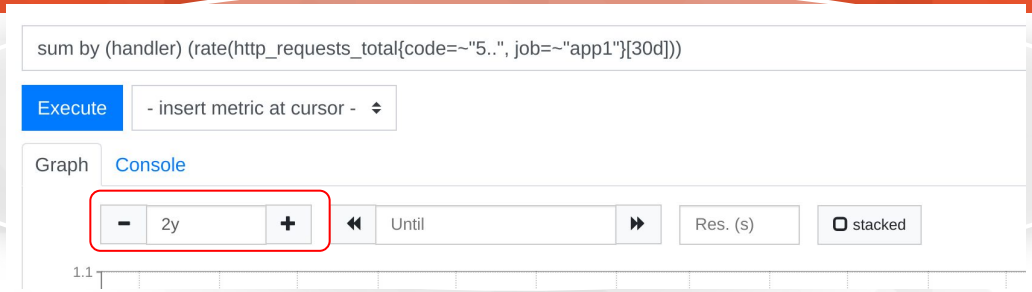
https://thanos.io

Long Term Metrics Retention

Nice, but how do I store **years** of our metrics?

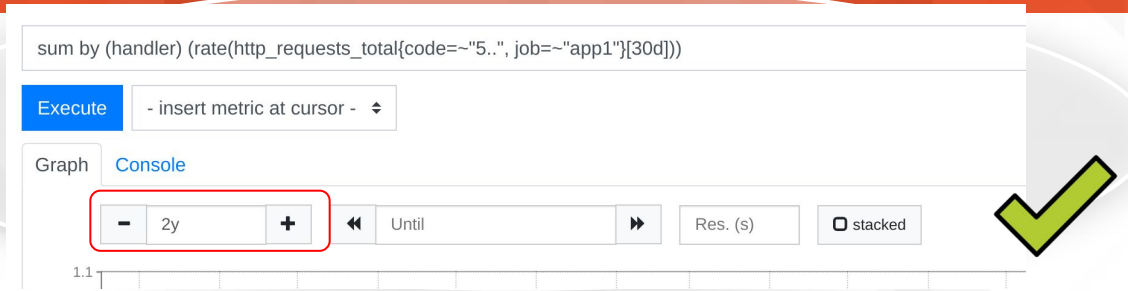
@bwplotka
@putadent

Long Term Retention with Prometheus?



@bwplotka
@putadent

Long Term Retention with Prometheus: **Yes**!



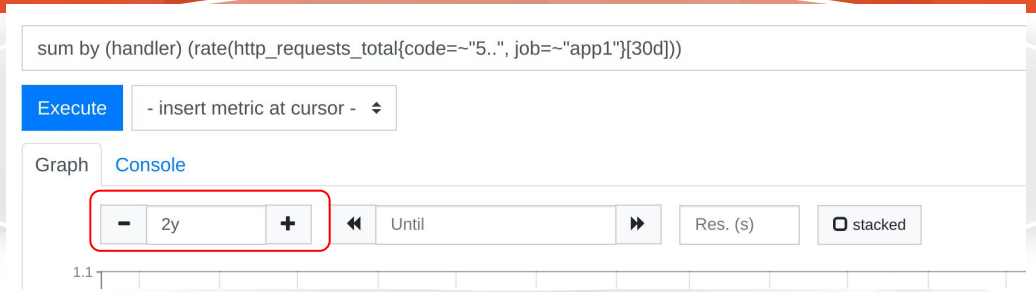
Prometheus **can** hold long metric retention (years!)

Fact: **Older data use marginal resources when not queried.**

How? **Get large SSD and plan data size head of time.**

@bwplotka
@putadent

Long Term Retention with Prometheus: **Caveats**



Be careful:

- Hard to plan disk space for future: Uncontrollable cardinality.
- Persistent disk backup are not always easy
- No downsampling

@bwplotka
@putadent

Remote Write

So what if I want to send to a different DB?

@bwplotka
@putadent

Remote Write

```
remote_write:
- name: data-lake
  url: https://remote-tsdb.net
  basic_auth:
  username: "10428"
  password: <secret>
```

@bwplotka
@putadent

Remote Write

```
remote_write:
- name: data-lake
  url: https://remote-tsdb.net
  basic_auth:
  username: "10428"
  password: <secret>

  write_relabel_configs:
  - source_labels: [__name__]
    regex: .*:.*
    action: keep
```

Remote Write

Integrations for every popular TSDB available.

Remote Endpoints and Storage

The remote write and remote read features of Prometheus allow transparently sending and receiving samples. This is primarily intended for long term storage. It is recommended that you perform careful evaluation of any solution in this space to confirm it can handle your data volumes.

- AppOptics: write
- Azure Data Explorer: read and write
- Azure Event Hubs: write
- Chronix: write
- Cortex: read and write
- CrateDB: read and write
- Elasticsearch: write
- Gnocchi: write
- Google Cloud Spanner: read and write
- Graphite: write
- InfluxDB: read and write
- IRONdb: read and write
- Kafka: write
- M3DB: read and write
- MetricFire: read and write
- OpenTSDB: write
- PostgreSQL/TimescaleDB: read and write
- QuasarDB: read and write
- SignalFx: write
- Splunk: read and write
- TiKV: read and write
- Thanos: read and write
- VictoriaMetrics: write
- Wavefront: write

@bwplotka
@putadent

Remote Write

```
message WriteRequest {
  repeated prometheus.TimeSeries timeseries = 1;
}
```

@bwplotka
@putadent

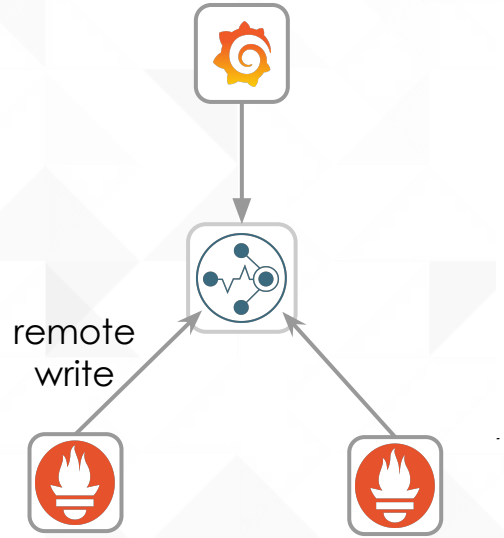
Remote Write

```
message WriteRequest {
  repeated prometheus.TimeSeries timeseries = 1;
}

message TimeSeries {
  repeated Label labels   = 1;
  repeated Sample samples = 2;
}
```

Global View via Remote Write (+ Cortex/M3/et.)

sum by (handler) (rate(http_requests_total{code=~"5..", job=~"app1"}[5m]))

Execute | - insert metric at cursor -



Documentation Twitter Github Search this site...

cortex

Horizontally scalable, highly available, multi-tenant, long term Prometheus.

Learn More | Releases

@bwplotka
@putadent

https://cortexmetrics.io

Metadata

Wait, what is this metric in this query??

I'm new to the team, how can I understand the alerts better?

@bwplotka
@putadent

Metadata

Follow naming best-practice

Sometimes, still not clear!

@bwplotka
@putadent

METRIC AND LABEL NAMING

The metric and label conventions presented in this document are not required for using Prometheus, but can serve as both a style-guide and a collection of best practices. Individual organizations may want to approach some of these practices, e.g. naming conventions, differently.

Metric names

A metric name...

- ...must comply with the data model for valid characters.
- ...should have a (single-word) application prefix relevant to the domain the metric belongs to. The prefix is sometimes referred to as `namespace` by client libraries. For metrics specific to an application, the prefix is usually the application name itself. Sometimes, however, metrics are more generic, like standardized metrics exported by client libraries. Examples:
  - `prometheus_notifications_total` (specific to the Prometheus server)
  - `process_cpu_seconds_total` (exported by many client libraries)
  - `http_request_duration_seconds` (for all HTTP requests)
- ...must have a single unit (i.e. do not mix seconds with milliseconds, or seconds with bytes).
- ...should use base units (e.g. seconds, bytes, meters - not milliseconds, megabytes, kilometers). See below for a list of base units.
- ...should have a suffix describing the unit, in plural form. Note that an accumulating count has `total` as a suffix, in addition to the unit if applicable.
  - `http_request_duration_seconds`
  - `node_memory_usage_bytes`
  - `http_requests_total` (for a unit-less accumulating count)

Metadata

```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0.000670276
go_gc_duration_seconds{quantile="0.25"} 0.001622397
go_gc_duration_seconds{quantile="0.5"} 0.002350074
go_gc_duration_seconds{quantile="0.75"} 0.003350546
go_gc_duration_seconds{quantile="1"} 0.015834097
go_gc_duration_seconds_sum 142.036233996
go_gc_duration_seconds_count 46744
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 5218
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.14.4"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.7939151784e+10
```

@bwplotka
@putadent

Metadata: API

Querying metric metadata

It returns metadata about metrics currently scrapped from targets. However, it does not provide any target information. This is considered **experimental** and might change in the future.

```
GET /api/v1/metadata
```

URL query parameters:

- `limit=<number>`: Maximum number of metrics to return.
- `metric=<string>`: A metric name to filter metadata for. All metric metadata is retrieved if left empty.

The `data` section of the query result consists of an object where each key is a metric name and each value is a list of unique metadata objects, as exposed for that metric name across all targets.

The following example returns two metrics. Note that the metric `http_requests_total` has more than one object in the list. At least one target has a value for `HELP` that do not match with the rest.

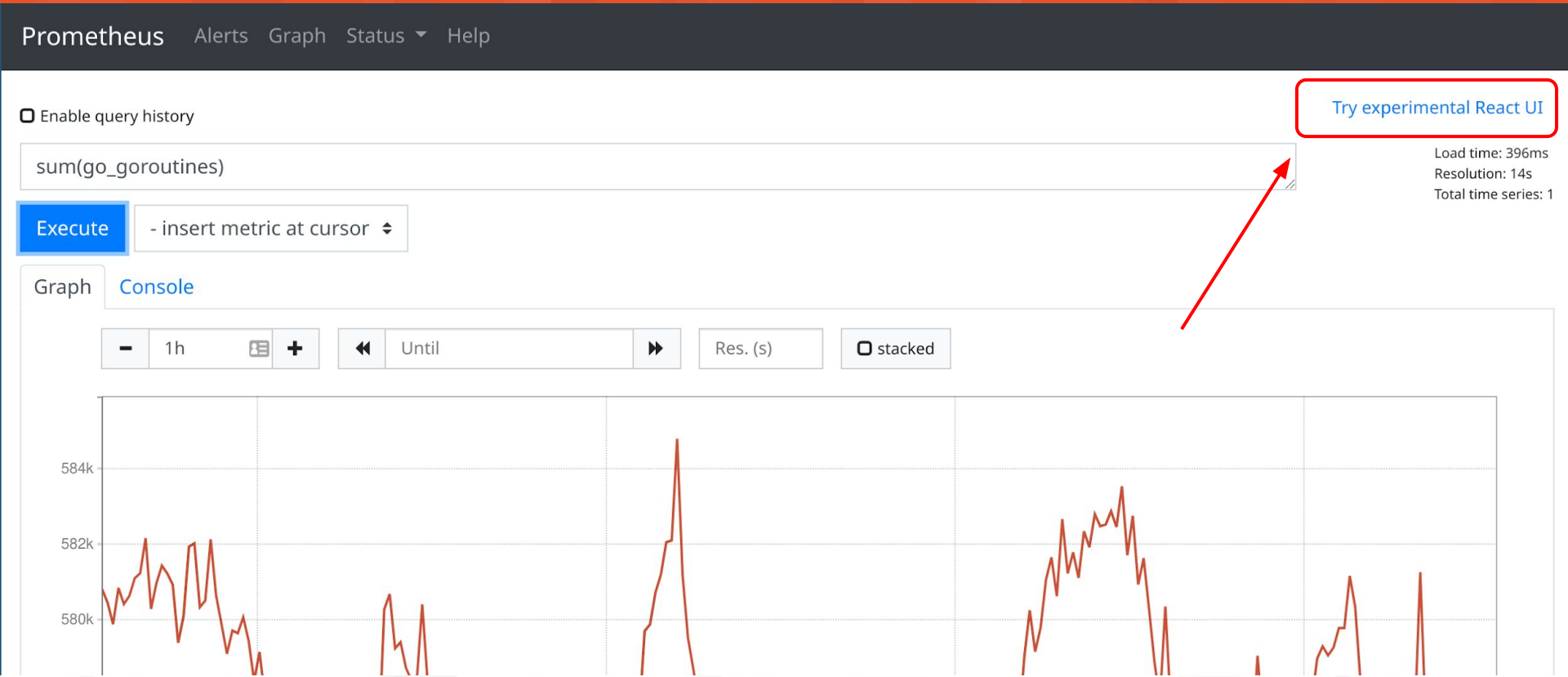
```
curl -G http://localhost:9090/api/v1/metadata?limit=2

{
  "status": "success",
  "data": {
    "cortex_ring_tokens": [
      {
        "type": "gauge",
        "help": "Number of tokens in the ring",
        "unit": ""
      }
    ],
    "http_requests_total": [
      {
```

@bwplotka
@putadent

Metadata

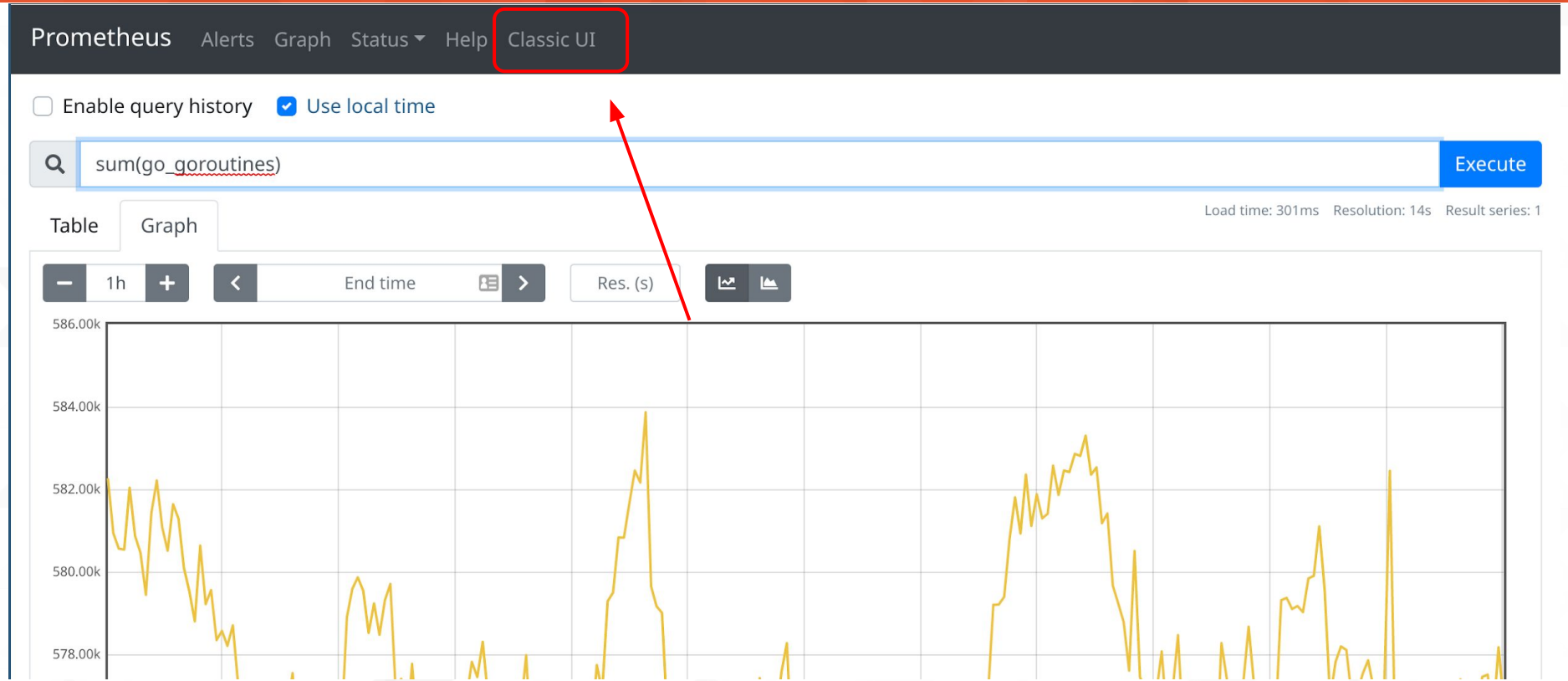
@bwplotka
@putadent

Sneak peek: React UI



@bwplotka
@putadent

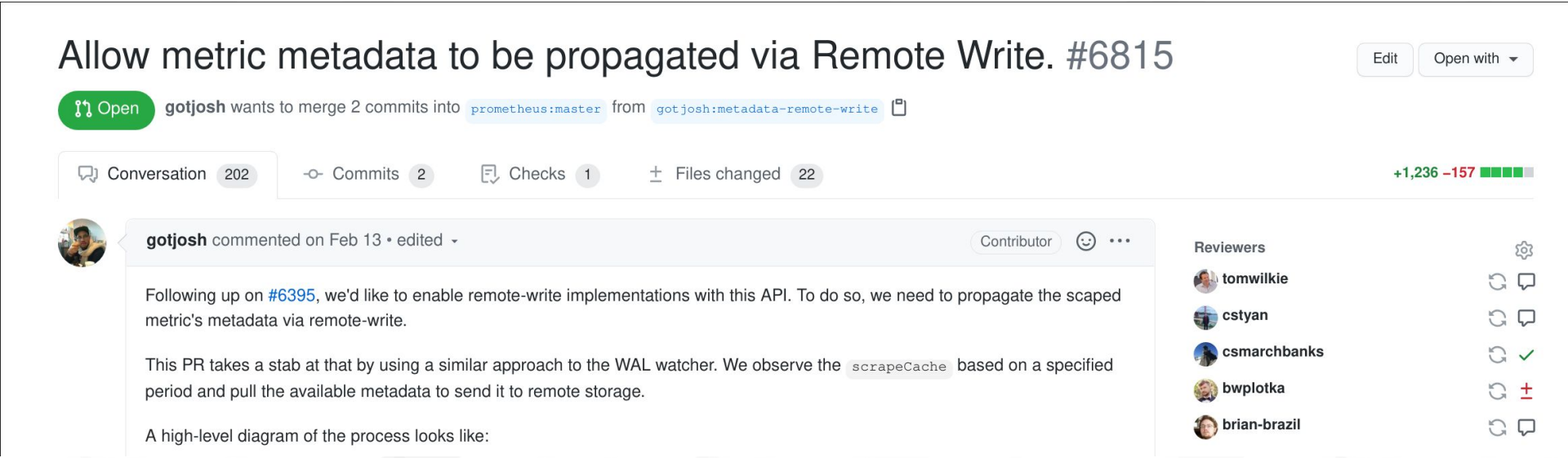
Sneak peek: React UI



@bwplotka
@putadent

Metadata: **Future**

- **Persist** the metadata to disk.
  - See it historically
- **Remote write** metadata to other systems



@bwplotka
@putadent

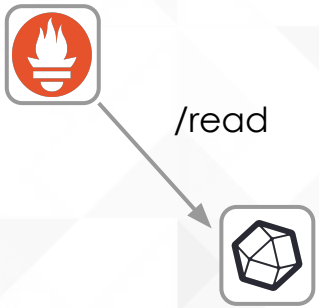
Backfilling

How do I **import my metrics** from different system to Prometheus?

@bwplotka
@putadent

Prometheus reading from another DB

- via Remote read




```
remote_read:
- name: remote-db
  url: https://my-influxdb
  basic_auth:
  username: "10428"
  password: <secret>
```


https://prometheus.io/docs/prometheus/latest/configuration/configuration/#remote_read

File based import to Prometheus?

Nice, but can I import some data from CSV file?

@bwplotka
@putadent

File based import to Prometheus: **Yes & DEMO**

Nice, but can I import some data from CSV file?

Added TSDB import with OpenMetrics and CSV file support. #7586

Edit | Open with

Open **bwplotka** wants to merge 2 commits into `master` from `tsdb-import`

Conversation 10 | Commits 2 | Checks 1 | Files changed 11 | +945 −44

bwplotka commented 4 hours ago • edited — Member

Based on #5887 Thanks for your work so far **@dipack95**, it helped a lot!

Changes on top of @dipack95 and above PR:

- Addressed all reviews components
- Used subcommands for different formats
- Simplified block creation, no need to be such complex for the first iteration.
- Simplified OpenMetrics parsing from io.Reader, the previous implementation was over-allocating a lot, I think this might be simple start. **@dipack95** please help to finish `openmetrics` package and tests 🤗
- Simplified and separate concerns. No need to have access to DB. Block
  - Block Writing is separated from parsing for ease of benchmarking and test. This will be also needed by **@JessicaGreben**
- Added import support for different formats.
- Removed all tests - those had to be pulled over and adjusted):

I wanted to create a demo of CSV file import for KubeCon import, so went ahead and tried to help you **@dipack95**. Hope we can together finish this and add two formats soon 🤗 What do you think of using this (if you think it's ok, please give your feedback!). Feel free to propose changes (PRs) to this branch `tsdb-import` so we can have this merged together. Help wanted around tests and benchmarks!

Reviewers: brian-brazil

Still in progress? Convert to draft

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

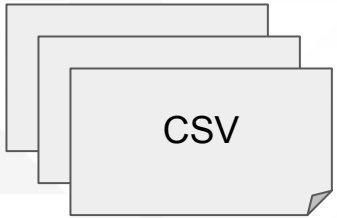
Linked issues

@bwplotka
@putadent

File based import to Prometheus?

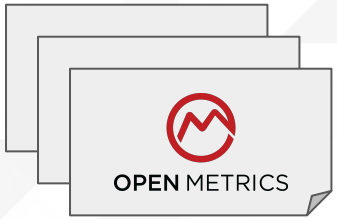
Nice, but can I import some data from CSV file?

cat my-file.csv | tsdb import



TSDB Block

cat my-file.om | tsdb import



@bwplotka
@putadent

Summary

@bwplotka
@putadent

Summary

- Global View
  - Query API
  - Federation
  - Remote Read

@bwplotka
@putadent

Summary

- Global View
  - Query API
  - Federation
  - Remote Read
- Long Term Storage
  - Just Prometheus!
  - Replication via Remote Write

@bwplotka
@putadent

Summary

- Global View
  - Query API
  - Federation
  - Remote Read
- Long Term Storage
  - Just Prometheus!
  - Replication via Remote Write
- Metadata

@bwplotka
@putadent

Summary

- Global View
  - Query API
  - Federation
  - Remote Read
- Long Term Storage
  - Just Prometheus!
  - Replication via Remote Write
- Metadata
- Future: Backfilling
  - CSV, OpenMetrics Import

@bwplotka
@putadent

Thank You!

Prometheus

You can reach us via:

- https://prometheus.io/community/
- Goutham: https://github.com/gouthamve @gouthamve
- Bartek: https://bwplotka.dev @bwplotka