Prometheus Design and Philosophy
Why It Is the Way It Is

Julius Volz, August 2016

Opinionated Monitoring System

- Prometheus is different
- potentially surprising decisions
- disclaimer: our opinion

What is Prometheus?

Monitoring system and TSDB:

- instrumentation
- metrics collection and storage
- querying
- alerting
- dashboarding / graphing / trending

Focus on:

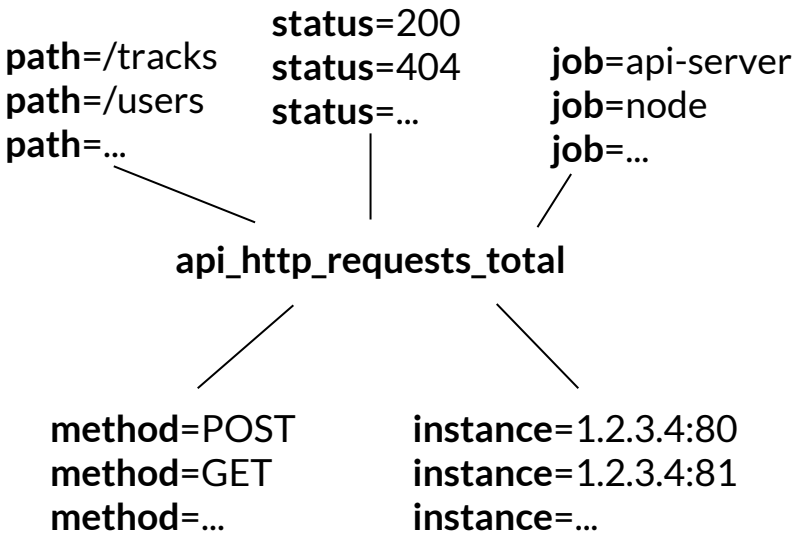
- operational systems monitoring
- dynamic cloud environments

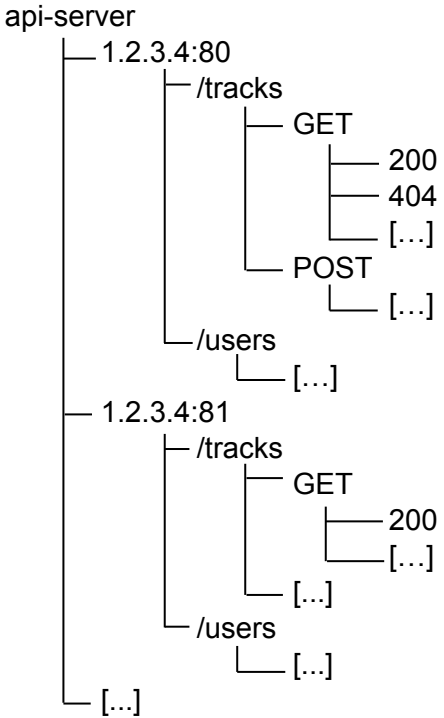
What does Prometheus NOT do?

- raw log / event collection
- request tracing
- “magic” anomaly detection
- durable long-term storage
- automatic horizontal scaling
- user / auth management

Data Model

Labels > Hierarchy





Labels > Hierarchy

```
api_http_requests_total{method="post"}
```

vs.

```
api-server.*.*.post.*
```

- more flexible
- more efficient
- explicit dimensions

Non-SQL Query Language

PromQL: `rate(api_http_requests_total[5m])`

SQL: `SELECT job, instance, method, status, path, rate(value, 5m) FROM api_http_requests_total`

PromQL: `avg by(city) (temperature_celsius{country="germany"})`

SQL: `SELECT city, AVG(value) FROM temperature_celsius WHERE country="germany" GROUP BY city`

PromQL: `errors{job="foo"} / total{job="foo"}`

SQL:

`SELECT errors.job, errors.instance, [...more labels...], errors.value / total.value FROM errors, total WHERE errors.job="foo" AND total.job="foo" JOIN [...some more complicated stuff here...]`

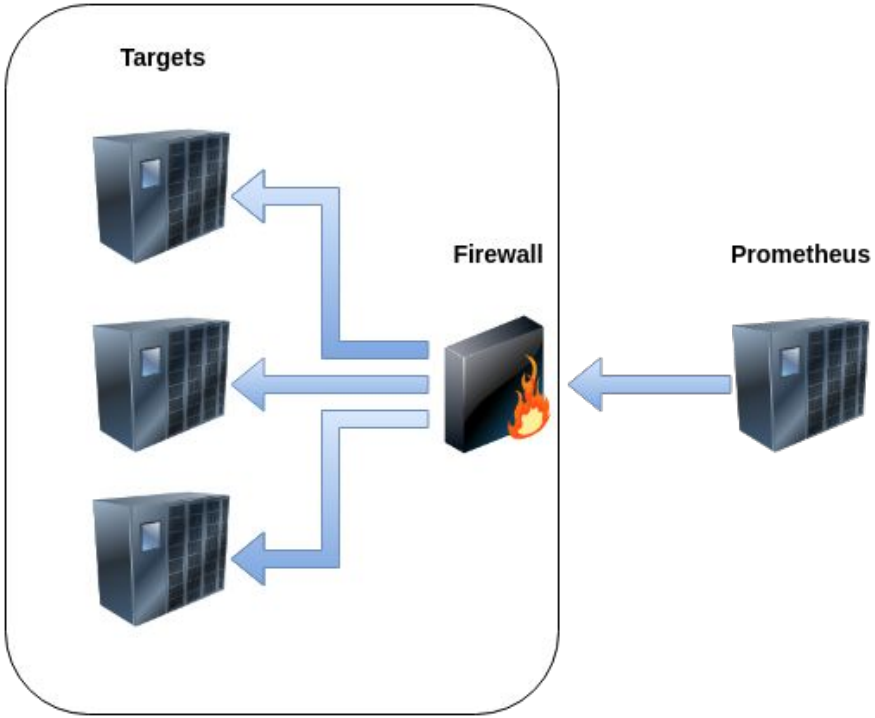
PromQL

- better for metrics computation
- only does reads

Pull vs. Push

- automatic upness monitoring
- horizontal monitoring
- more flexible
- simpler HA
- less configuration
- yes, it scales!

But but...



Alternatives and Workarounds

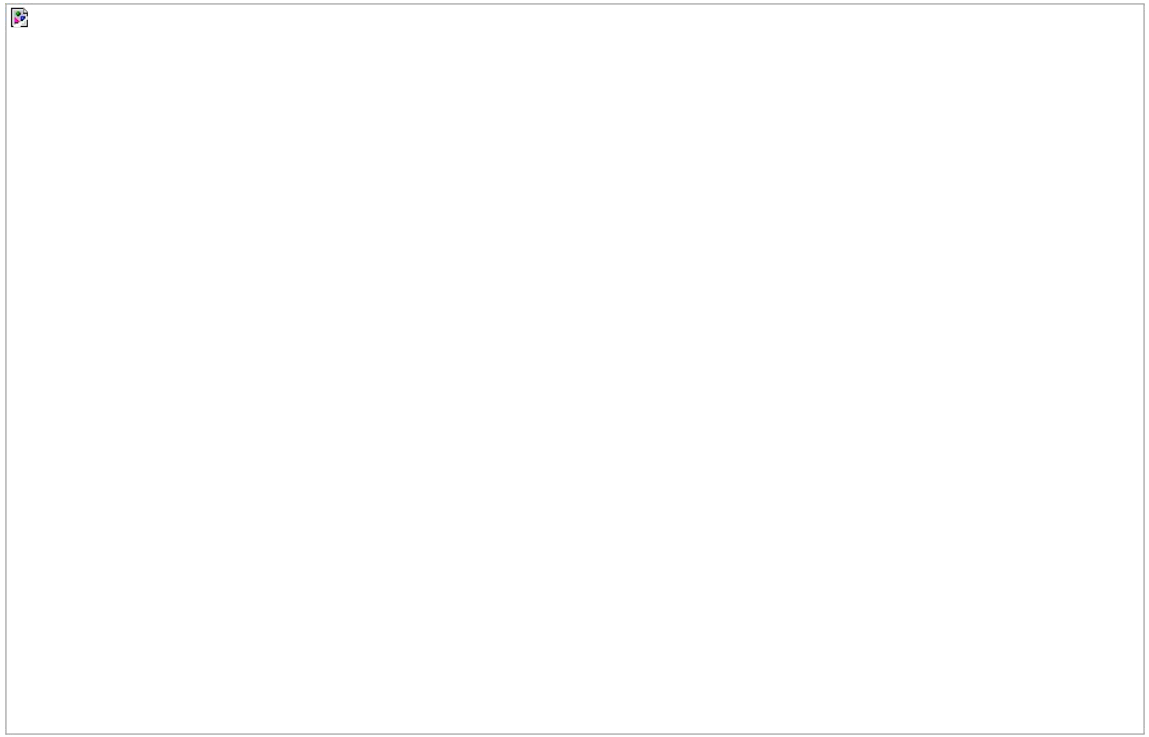
- run Prometheus on same network segment
- open port(s) in firewall / router
- open tunnel / VPN

Uber-Exporters

or...

Per-Process Exporters?

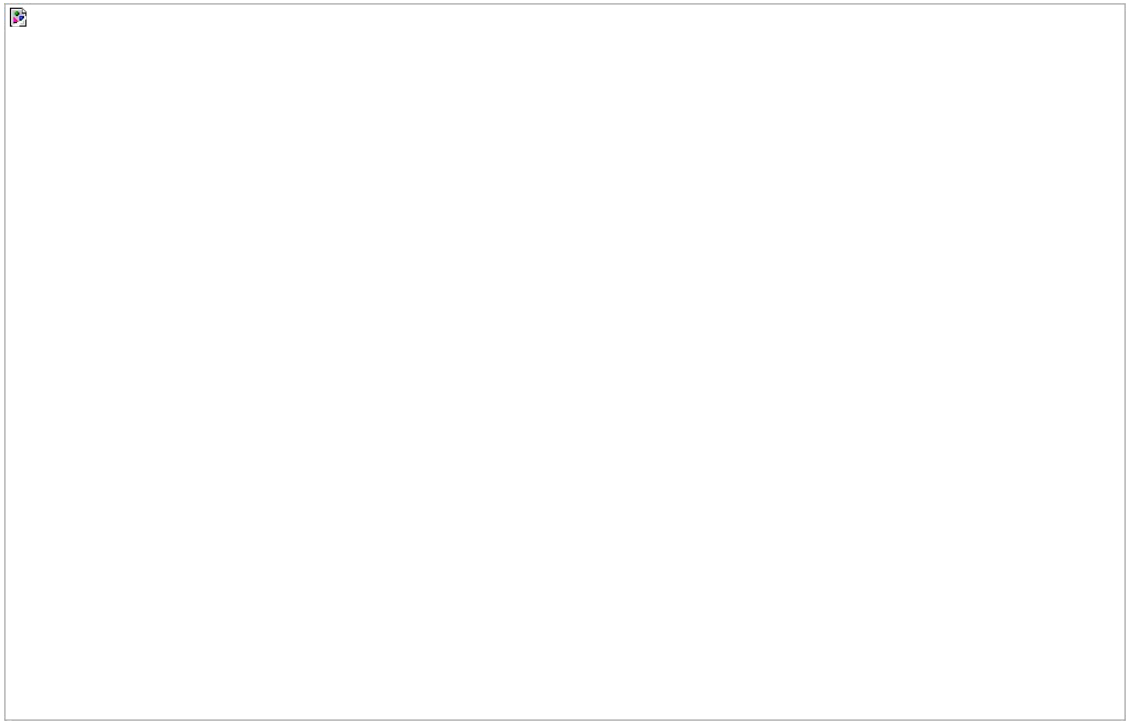
Per-Machine Uber-Exporters



Drawbacks

- operational bottleneck
- SPOF, no isolation
- can’t scrape selectively
- harder up-ness monitoring
- harder to associate metadata

One Exporter per Process



Why not JSON?

We optimized for two extremes:

Text format

- easy to construct
- relatively efficient
- readable
- streamable

Protobuf format

- very efficient
- robust
- streamable

JSON? Worse in all categories.

No Clustering?

- really hard to get right
- first thing to fail when you need it (e.g. during network outage)
- keep it simple, focus on operational monitoring
- HA for alerting still easy

Relabeling - WTF?

```
relabel_configs:
- source_labels: [__meta_kubernetes_service_annotation_prometheus_io_scrape]
  action: keep
  regex: true
- source_labels: [__meta_kubernetes_service_annotation_prometheus_io_scheme]
  action: replace
  target_label: __scheme__
  regex: (https?)
- source_labels: [__meta_kubernetes_service_annotation_prometheus_io_path]
  action: replace
  target_label: __metrics_path__
  regex: (.+)
- source_labels: [__address__, __meta_kubernetes_service_annotation_prometheus_io_port]
  action: replace
  target_label: __address__
  regex: (.+)(?::\d+);(\d+)
  replacement: $1:$2
- action: labelmap
  regex: __meta_kubernetes_service_label_(.+)
- source_labels: [__meta_kubernetes_service_namespace]
  action: replace
  target_label: kubernetes_namespace
- source_labels: [__meta_kubernetes_service_name]
  action: replace
  target_label: kubernetes_name
```

Relabeling - OK...

- a new DSL
- steep learning curve
- ...but very flexible

The alternative:
many special config options.

Stateful Client Libraries

- client libs keep state
- but not much
- manage metrics for you
- pre-aggregation is more efficient

Everything is a float64...

This is crazy! I only need integers!
What about precision?

Everything is a float64...

- it's simpler
- we compress it incredibly well
- float64 integer precision until 2^53

To run into trouble:

Increment counter 1 million times per second for over 285 years

Auth or Multi-User?

- too many different ways
- focus on great monitoring
- solve auth externally
- multitenancy is more difficult

Conclusion

- it’s not about Prometheus
- it’s about monitoring best practices
- it’s our opinion

Thanks!