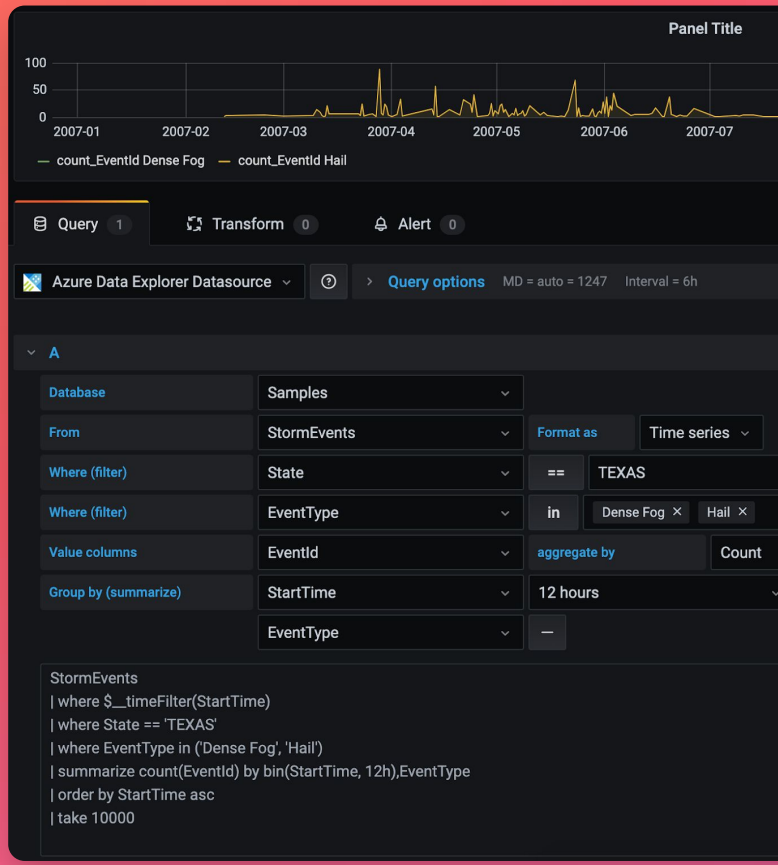
Grafana Labs

Driving organisational change using OpenTelemetry

Juraci Paixão Kröhling
Software Engineer



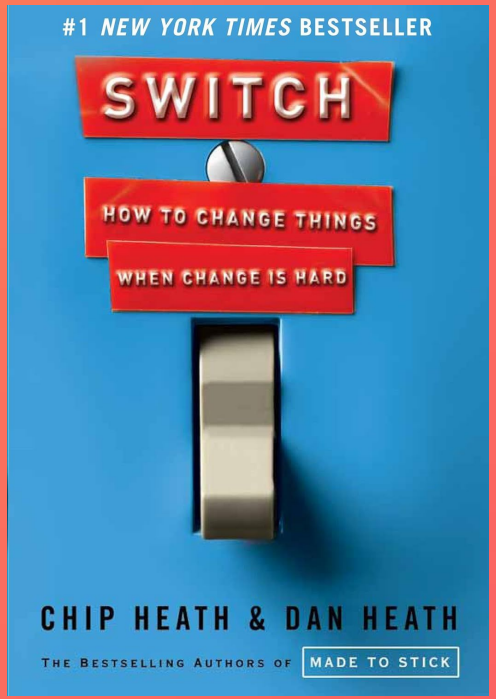
Agenda

- Organisational change?
- OpenTelemetry
- Implementing internally
- Leveraging the *Community*

Grafana Labs

Organisational Change

Grafana Labs



Organisational changes *at Grafana Labs*

- RED dashboards
- Adopting SLOs
- Cost sensitivity

Grafana Labs

RED Dashboards

RED Method

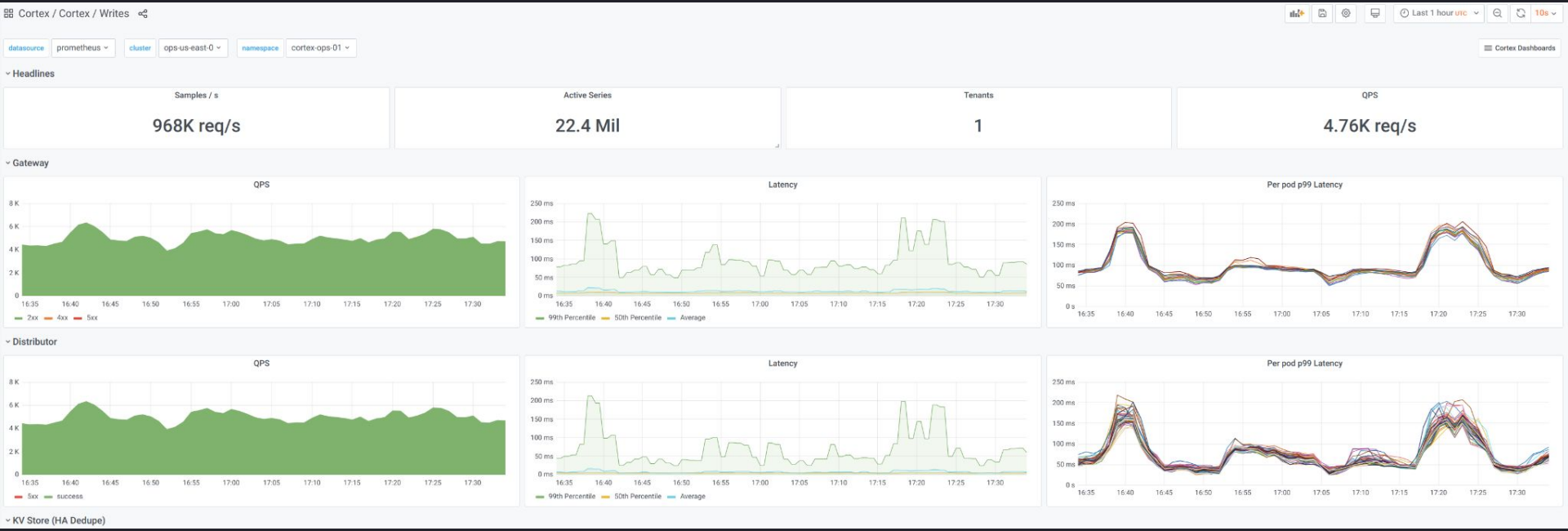
Requests - The number requests per seconds split out by status code

Errors - The number of errors, superimposed on the requests

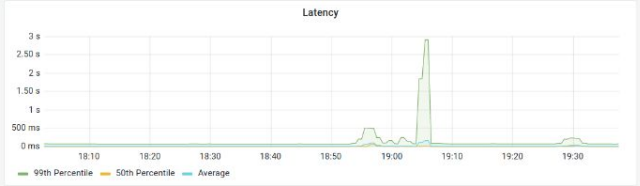
Duration - 50th and 95th percentile response times

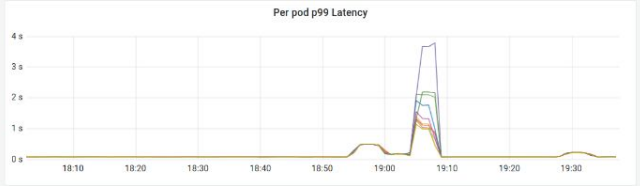
Following the *Critical Request Path*

RED Dashboards

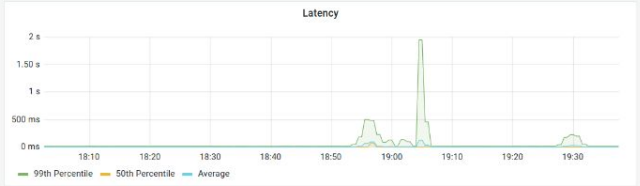


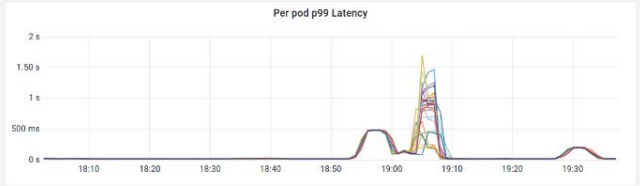
Gateway





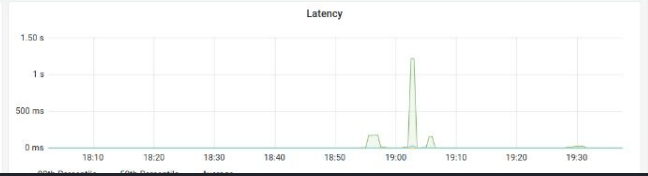
Distributor

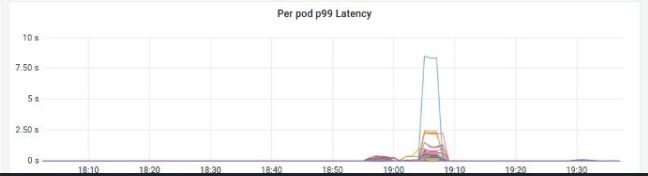




KV Store (HA Dedupe) *(2 panels)*

Ingester





Grafana Labs

SLOs

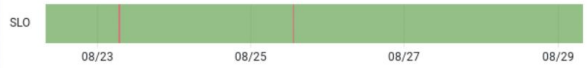
Grafana Cloud SLOs 1/3

Overall Performance (1d)	Overall Performance (7d)	Overall Performance (14d)	Overall Performance (30d)
99.984%	99.980%	99.983%	99.980%

Access

GrafanaCom API	99.98%	
StackStateSe API	100.00%	

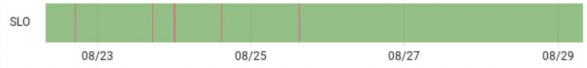
Alerting

Alertmanage Threshold is 99.5%, Runbook	99.90%	
Config Threshold is 99.5%, Runbook	99.98%	

Auth

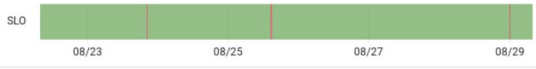
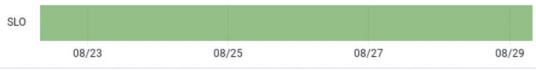
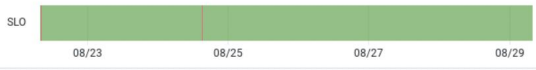
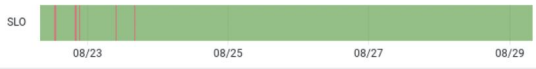
Auth API Threshold is 99.5%, Runbook	99.99%	

Cortex

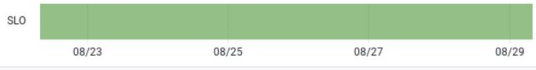
Writes Threshold is 99.9%, Runbook	99.99%	
Reads Threshold is 99.5%, Runbook	99.81%	

Grafana Cloud SLOs 3/3

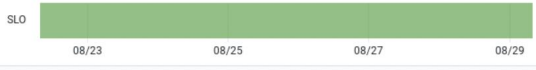
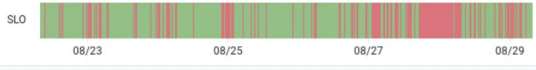
Middleware

Integrations API	99.93%	
Hosted Exporters API	98.45%	
CloudWatch Exporter	100.00%	
Confluent Exporter	99.96%	
SyntheticMor Threshold is 99.5%,	99.97%	

OnCall

API Threshold is 99.5%,	99.97%	

Tempo

Writes Threshold is 99.9%, Runbook	100.00%	
TraceByID Threshold is 99.5%, Runbook	99.61%	
Search Threshold is 99.5%, Runbook	99.89%	

```
// An SLO is defined as
// {
//   name:            Name of SLO; used in dashboard and rule names.  No spaces.
//   desc:            Free-form markdown for exposition.
//
//   gen_alert:       Set to true to generate alerts for this SLO.
//   alert_name:      Name of alert to use for this SLO.
//   alert_summary:   Summary for alert.
//   alert_labels:    Extra labels to add to a generated alert (optional).
//   runbook_template: Runbook template such as "https://deployment_tools/docs/runbook.md#%(alert_name)s" (optional)
//                    The %(alert_name)s will be used to substitute with the alert name (also optional)
//
//   gen_dashboard: Set to true to include in the SLO dashboard.
//
//   successful_requests_expr:
//     PromQL expression for the number of requests that succeeded wrt SLA
//     (ie succeeded quickly enough).  Should take a "%(period)s" substitution.
//     Final aggregate should only include cluster and namespace labels.
//
//   total_requests_expr:
//     PromQL expression for the total number of request served in a given
//     period.
//
//   threshold:
//     Proportion of total requests (0-100) that need to comply for us to be
//     within SLO.
//
//   min_failures:
//     Absolute number of failures that must be reached over an alert's evaluation
//     period before the alert fires. This protects against noisy alerts during
//     low-traffic periods.
//
//   weight: (optional)
//     A constant factor that both successful_requests_expr and total_requests_expr
//     is multiplied by; used to control the weight this SLO contributes to the
//     aggregate SLO.
// }
```

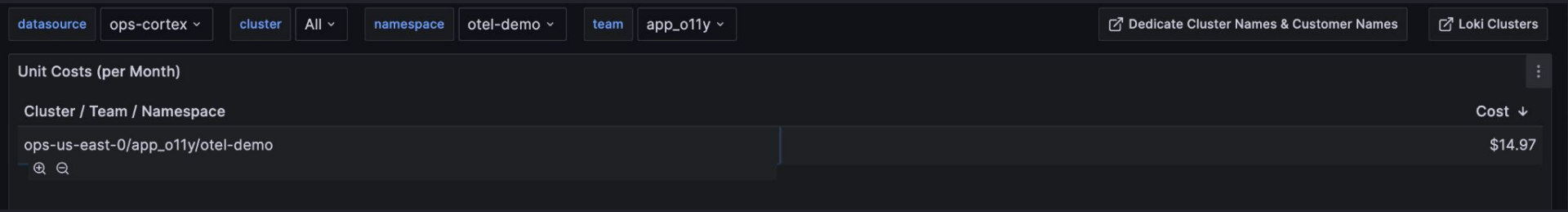
```
{
  name: 'TraceByID',

  gen_alert: true,
  alert_name: 'TempoTraceByIDErrorBudgetBurn',
  alert_summary: 'Tempo burns its trace by id error budget too fast.',
  gen_dashboard: true,

  successful_requests_expr: 'sum by (cluster, namespace) (rate(tempo_request_duration_seconds_bucket{status_code!~'
  total_requests_expr: 'sum by (cluster, namespace) (rate(tempo_request_duration_seconds_count{route=~"tempo_api_tr
  threshold: 99.5,
  min_failures: 10,
},
```

Grafana Labs

Cost sensitivity


datasource
ops-cortex
cluster
All
namespace
otel-demo
team
app_o11y
Dedicate Cluster Names & Customer Names
Loki Clusters
Unit Costs (per Month)
Cluster / Team / Namespace
Cost
ops-us-east-0/app_o11y/otel-demo
$14.97

cluster All | namespace All | team app_o11y

Dedicate Cluster Names & Customer Names | Loki Clusters

Team Growth

Team	Previous Month	This Month	Time	Growth
networking	$		2023-10-25 11:14:34.054	
app_o11y			2023-10-25 11:14:34.054	
loki	$6	$8	2023-10-25 11:14:34.054	$
grafana_com	$		2023-10-25 11:14:34.054	
hosted_grafana	$1	$	2023-10-25 11:14:34.054	
incident			2023-10-25 11:14:34.054	
tempo	$1	$1	2023-10-25 11:14:34.054	$
Total				$2

Namespace Growth

Cluster / Team / Namespace	Previous Month	This Month	Growth ↓
prod-us-west-0/app_o11y/faro	$	$	$
prod-ca-east-0/app_o11y/faro	$	$	$
prod-ap-southeast-1/app_o11y/faro	$	$	$
ops-us-east-0/app_o11y/otel-demo			$
prod-au-southeast-1/app_o11y/faro	$	$	$
us-central2/app_o11y/faro	$	$	$
prod-us-central-0/app_o11y/faro	$	$	$

Merged **Set Beyla CPU resources and limits** #98979

grcevski merged 1 commit into `master` from `set_beyla_resources` 2 days ago

Set Beyla CPU resources and limits ✓ 8582a97

grcevski requested a review from **grafana/ebpf** as a code owner 2 days ago

kube-manifests-exporter bot commented 2 days ago

2 files changed in kube-manifests. permalink (relative)

▶ Changes to manifests in dev clusters, **deployed on merge** (0 added, 0 deleted, 2 modified)

kube-manifests-exporter bot commented 2 days ago

💵 Cost Estimation Report 📈

Monthly cost for the affected resources will increase by $46.97 (37.98%)

▶ Details for `dev-us-central-0`

Legend: previous cost on top, expected cost below.

See the FAQ for any questions!

Still need help? Then join us in the `#platform-capacity-chat` channel.

ebpf

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications Customize

Unsubscribe

You're receiving notifications because your review was requested.

3 participants

Lock conversation

How?

- Near Zero effort from engineers
  - Engineers want to focus on product velocity
- Make it a priority for the team
  - But make it easy, visible and trackable
  - Weekly reports sent automatically
- Automate *EVERYTHING*

How???

- Key is standardization
  - We use Prometheus metrics
  - We use Prometheus histograms with the same buckets
  - We use the same libraries company wide
- Doing it in a 1000 person company is *easy*
- How do you do it in a 10000 person company?
  - Standardization is ~impossible

What is OpenTelemetry

OpenTelemetry is a collection of tools, APIs, and SDKs. Use it to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help you analyze your software's performance and behavior.

From: https://opentelemetry.io/

Standards, specifications and conventions

Client-side libraries (API, SDKs, instrumentation)

Middleware

Standards, specifications and conventions

OpenTelemetry API / SDK

- For those who want to implement an API or SDK

Semantic conventions

- What I should include in my telemetry data

OpenTelemetry Line Protocol

- Specification for data transport

Client-side libraries

OpenTelemetry API

- Enables developers to specify “what” to generate in terms of telemetry data

OpenTelemetry SDK

- Configuration of “how” to collect and transport instrumented application data

Instrumentation libraries

- Libraries that instrument parts of your stack automatically

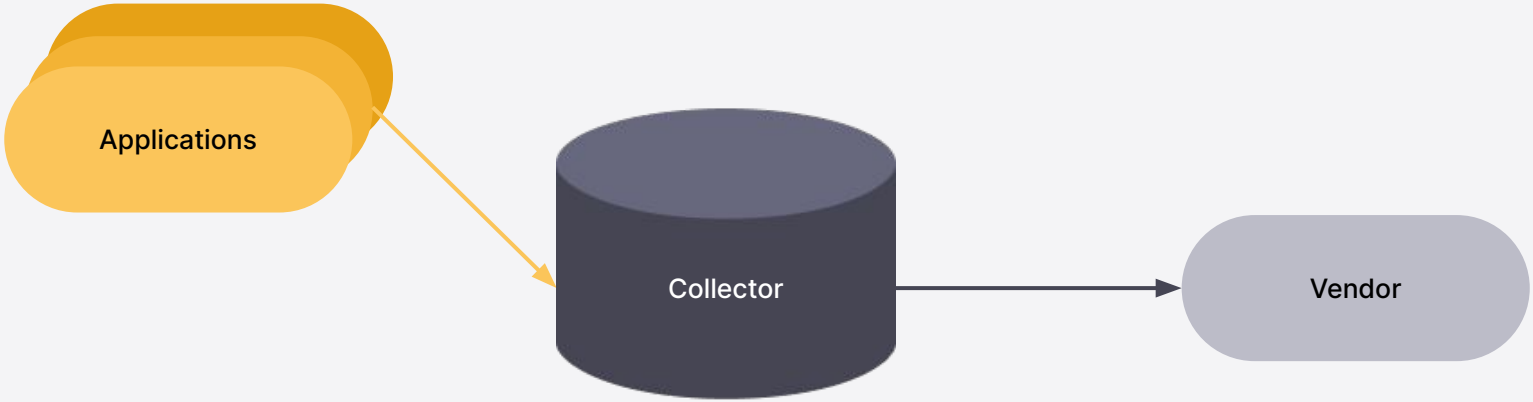
Middleware

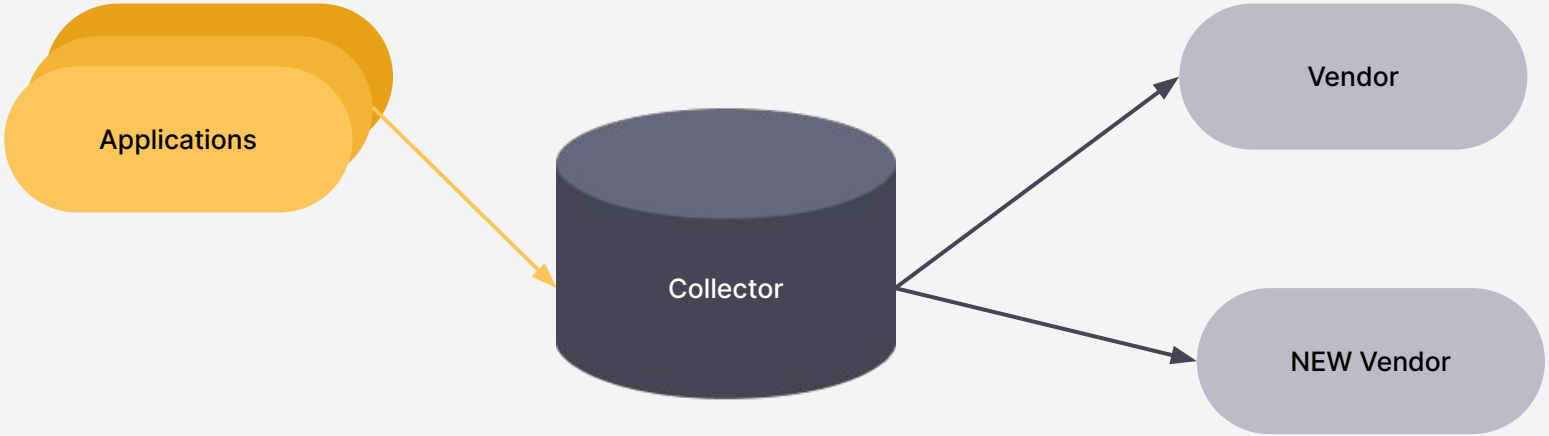
OpenTelemetry Collector

- Conduit that allows connecting sources to destinations

OpenTelemetry Operator

- Operator for Kubernetes, able to manage Collectors and self-instrument services

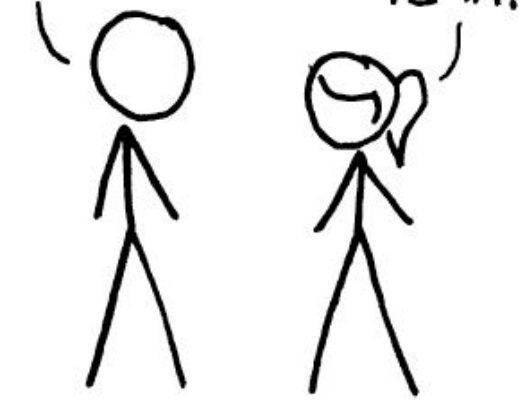

Applications
Collector
Vendor


Applications
Collector
Vendor
NEW Vendor

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

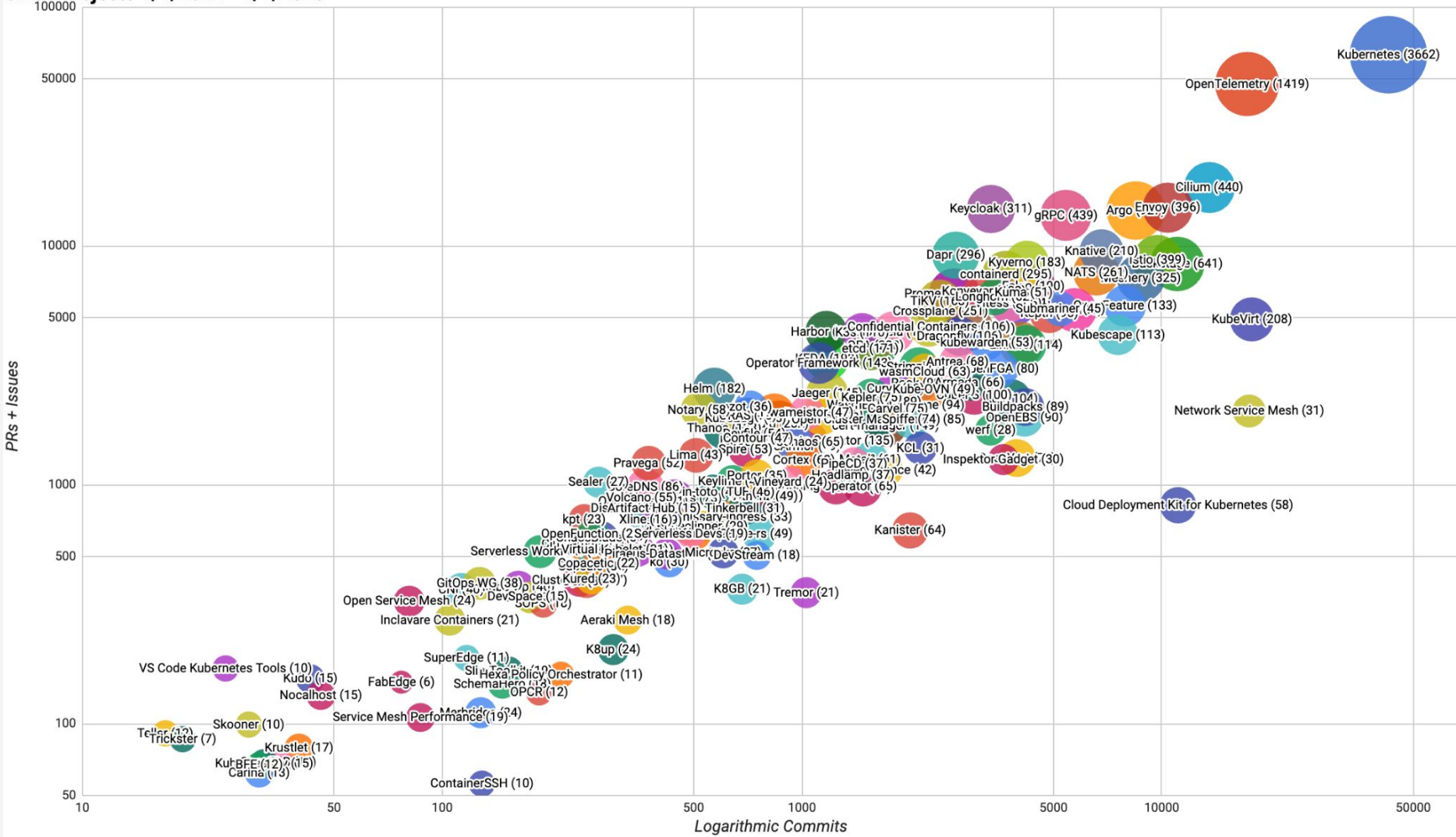
SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

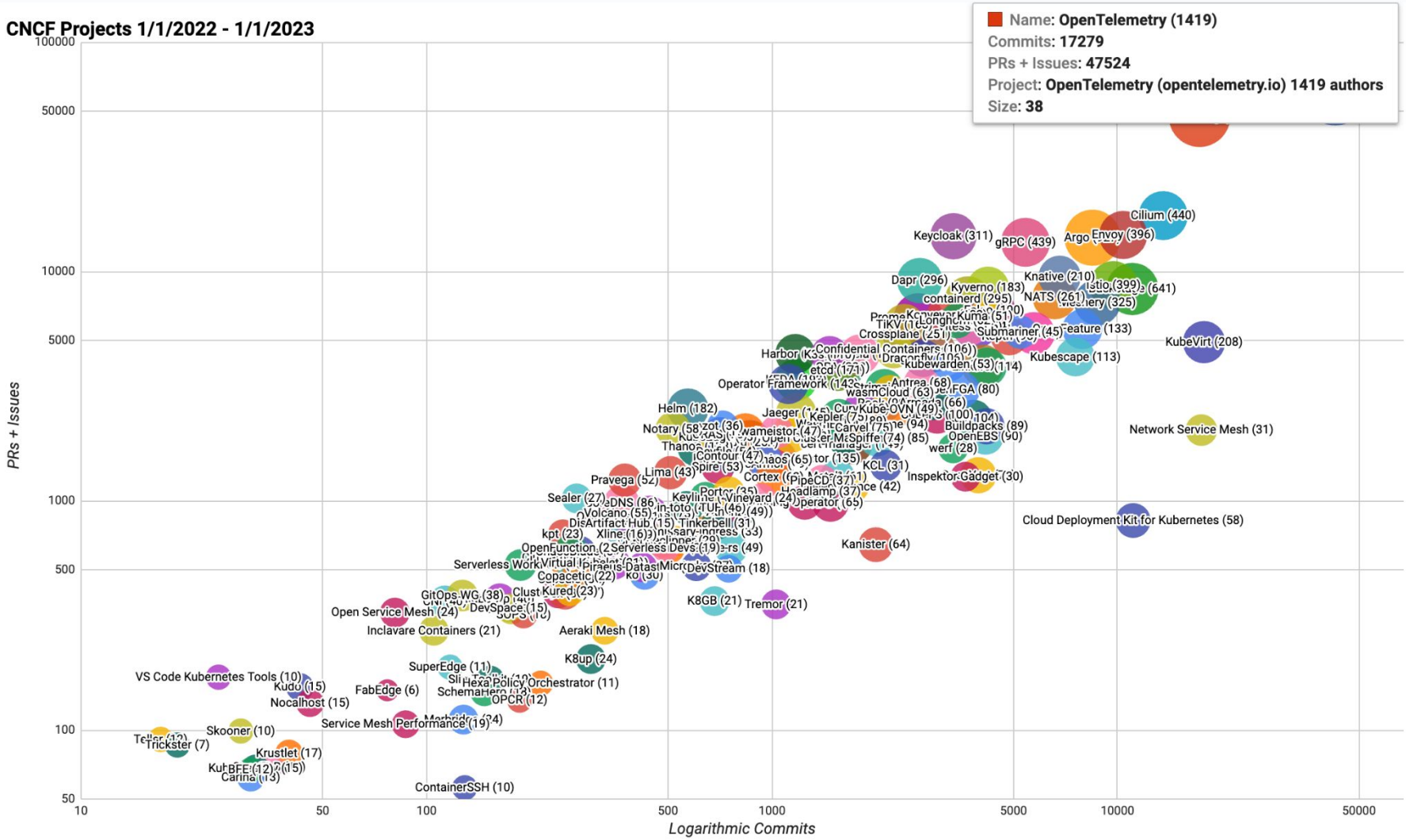
14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.


PRs + Issues
Logarithmic Commits
100000
50000
10000
5000
1000
500
100
50
10
50
100
500
1000
5000
10000
50000
Kubernetes (3662)
OpenTelemetry (1419)
Cilium (440)
Keycloak (311)
gRPC (439)
Argo
Envoy (396)
Dapr (296)
Kyverno (183)
Knative (210)
containerd (295)
NATS (261)
Kuma (51)
Crossplane (251)
Submariner (45)
KubeVirt (208)
Kubescape (113)
Confidential Containers (106)
kubewarden (53)
Operator Framework (143)
Antrea (68)
OpenFGA (80)
wasmCloud (63)
Helm (182)
Jaeger
Kube-OVN (49)
Notary (58)
Buildpacks (89)
Network Service Mesh (31)
OpenEBS (90)
Spiffe (74) (85)
Thanos
Contour (47)
KCL (31)
Inspektor Gadget (30)
Pravega (52)
Lima (43)
Cortex
PipeCD (37)
Headlamp (37)
Sealer (27)
Keylime
Vineyard (24)
Volcano (55)
Artifact Hub (15)
Tinkerbell (31)
Cloud Deployment Kit for Kubernetes (58)
kpt (23)
Xline (16)
Kanister (64)
OpenFunction (2
Serverless Devs (19)
DevStream (18)
Copacetic (22)
K8GB (21)
Tremor (21)
GitOps WG (38)
Kured (23)
DevSpace (15)
Open Service Mesh (24)
Inclavare Containers (21)
Aeraki Mesh (18)
K8up (24)
SuperEdge (11)
VS Code Kubernetes Tools (10)
Hexa
Policy Orchestrator (11)
Kudo (15)
FabEdge (6)
SchemaHero
OPCR (12)
Nocalhost (15)
Service Mesh Performance (19)
Skooner (10)
Trickster (7)
Krustlet (17)
KubeEdge (12)
Carina (13)
ContainerSSH (10)


CNCF Projects 1/1/2022 - 1/1/2023
Name: OpenTelemetry (1419)
Commits: 17279
PRs + Issues: 47524
Project: OpenTelemetry (opentelemetry.io) 1419 authors
Size: 38
PRs + Issues
Logarithmic Commits
100000
50000
10000
5000
1000
500
100
50
10
50
100
500
1000
5000
10000
50000
Cilium (440)
Keycloak (311)
gRPC (439)
Argo (396)
Envoy (396)
Dapr (296)
Knative (210)
Kyverno (183)
containerd (295)
NATS (261)
Crossplane (251)
Submariner (45)
KubeVirt (208)
Kubescape (113)
Harbor (K8s
Confidential Containers (106)
kubewarden (53)
(114)
Operator Framework (143)
Antrea (68)
wasmCloud (63)
FGA (80)
Helm (182)
Jaeger (
Kube-OVN (49)
Notary (58
zot (36)
Buildpacks (89)
Network Service Mesh (31)
OpenEBS (90)
werf (28)
Contour (47)
KCL (31)
Inspektor Gadget (30)
Lima (43)
Pravega (52)
Cortex (
PipeCD (37)
Sealer (27)
Keylime
Vineyard (24)
Volcano (55)
Cloud Deployment Kit for Kubernetes (58)
Kanister (64)
Serverless Work
Virtual
DevStream (18)
Copacetic (22)
Kured (23)
GitOps WG (38)
DevSpace (15)
K8GB (21)
Tremor (21)
Open Service Mesh (24)
Inclavare Containers (21)
Aeraki Mesh (18)
K8up (24)
SuperEdge (11)
VS Code Kubernetes Tools (10)
Kudo (15)
FabEdge (6)
Hexa
Policy Orchestrator (11)
Nocalhost (15)
OPCR (12)
Service Mesh Performance (19)
Skooner (10)
Trickster (7)
Krustlet (17)
Carina (13)
ContainerSSH (10)

OTel super power: Semantic Conventions

Semantic Conventions | OpenTelemetry

Connection pools

The following metric instruments describe database client connection pool operations.

Metric: `db.client.connections.usage`

This metric is required.

Name	Instrument Type	Unit (UCUM)	Description
`db.client.connections.usage`	UpDownCounter	`{connection}`	The number of connections that are currently in state described by the `state` attribute

Attribute	Type	Description	Examples	Requirement Level

Create project issue

Grafana Labs

Demo

Grafana Labs

Implementing change internally

Implementing change

- Make change easy & obvious
  - Measure and showcase at every step.
  - We had **many** changes which failed because it wasn't easy or obvious why.
- Find the emotional argument to compliment the logical ones.
  - *Sleep better at night*.
- What is measured is what changes
- Find areas where things are working
  - Find the reproducible parts and leverage people to spread change

Open Standards

- Others are trying to do the same thing
  - Avoid duplication and increase collaboration
  - OpenTelemetry, Prometheus, etc.

- Build on open efforts
  - There will be lots of OSS projects building on top of OTel semantic conventions

Grafana Labs

Thanks for participating!

Have more questions?

Join us at community.grafana.com

Grafana public slack

#application-observability

Get involved:

#application-observability

grafana/

community.grafana.com

More meetups in Berlin? Yes, please!

We’re ready to give you the stage! Submit this form and we’ll get in touch about your presentation. Have a cool story but would rather skip the public speaking? Write a blog!

https://grafana-labs.typeform.com/community

Interested in working with us?

Scan a QR code to apply directly to our career page.
A Grafana team member will review your resume and get back to you shortly!

Take Grafana Labs Observability Survey