When Prometheus met OpenTelemetry

Fosdem 2024

Pavol Loffay
Principal Software Engineer
About

- Pavol Loffay
- Principal Software Engineer @ Red Hat
- OpenTelemetry operator, Tempo operator and Jaeger maintainer
  
  @ploffay

  CNCF Slack
Agenda

- OSS metrics ecosystem
- Data model differences
- Prometheus in the OpenTelemetry ecosystem
  - API/SDK & Collector
- Collecting Prometheus metrics on Kubernetes
  - OpenTelemetry target allocator
- Wrap up
Why? (OSS metrics ecosystem)

Shall we create another “standard”? 
Data model differences
Prometheus data model vs OpenTelemetry
Protocol and Data model differences

- Pull vs Push
- Cumulative vs delta temporality #12763
- Native vs exponential histograms (with min/max)
- target_info vs resource attributes #1262
- float vs int
- Utf8 support for label and metric names #13095
Temporality

- OTEL_EXPORTER_OTLP_METRICS_TEMPORALITY_PREFERENCE
  - Cumulative, Delta, LowMemory
- Prometheus exporter
  - Collector does delta2cumulative
  - SDKs might not support delta2cumulative
Prometheus in OpenTelemetry ecosystem

Prometheus receiver/exporter, target allocator
OpenTelemetry SDKs

- Prometheus exporter
  - delta temporality not well supported
OpenTelemetry Collector

- Prometheus receiver
- Prometheus exporter
  - should support delta2cumulative
- Prometheus Remote Write Exporter
  - delta temporality not well supported
Demo

OpenTelemetry collector with Prometheus
example-prometheus-app

Prometheus client

/metRICS

OpenTelemetry

debug exporter
Prometheus receiver
Manually configure prometheus receiver
apiVersion: opentelemetry.io/v1alpha1
class: OpenTelemetryCollector
metadata:
  name: otel
spec:
  mode: deployment
  config:
    receivers:
      prometheus/static:
        config:
          scrape_configs:
            - job_name: 'app'
              scrape_interval: 10s
              static_configs:
                - targets: [ 'prometheus-example-app:80' ]
Prometheus receiver service discovery

Via OpenTelemetry collector
Prometheus receiver with service discovery

apiVersion: opentelemetry.io/v1alpha1
kind: OpenTelemetryCollector
metadata:
  name: otel
spec:
  mode: deployment
  config: |
    receivers:
      prometheus/k8s:
        config:
          scrape_configs:
            - job_name: k8s
              scrape_interval: 10s
              kubernetes_sd_configs:
                - role: pod
                  selectors:
                    - role: pod
                      label: "app.kubernetes.io/name=prometheus-example-app"
Prometheus receiver and
Service/Pod monitors

OpenTelemetry collector and target allocator
OpenTelemetry collector with target allocator

```yaml
apiVersion: monitoring.coreos.com/v1
kind: PodMonitor

apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
```

Platform
Prometheus, Thanos, Splunk...

Pod 2

Collector-1

OpenTelemetry operator

reconciles

Collector-2

Pod 3

Shards scrape targets

TA
Prometheus receiver with target allocator

```yaml
apiVersion: opentelemetry.io/v1alpha1
class: OpenTelemetryCollector
metadata:
  name: otel
spec:
  mode: statefulset
targetAllocator:
    enabled: true
  prometheusCR:
    enabled: true
    scrapeInterval: 5s
    serviceMonitorSelector:
      name: prometheus-example-app
config:
  receivers:
    prometheus:
      config:
        scrape_configs: []
```
Wrap-up

We have learned some OpenTelemetry magic tricks
Wrap-up

- Be careful with OTEL instrumented services (delta temporality)
- Prometheus receiver
  - escape $ with ($$)
- Probe and ScrapeConfig CRDs are not supported
- Target allocator
  - TLS
  - Other options from the Prometheus spec are not supported [1934](#)
Prometheus call to action

- Prometheus 3.0 OTLP
- prometheus/projects/9/views/8
OpenTelemetry call to action

- OpenTelemetry collector v1alpha2 CRD
- Target Allocator CRD
- Prometheus receiver/exporter in the collector
Thank you

Pavol Loffay
@ploffay