Journey from Fluent Bit, Fluentd and Prometheus to OpenTelemetry Collector

Marcin "Perk" Stożek

Canonical
Collectors Zoo

collectd

fluentbit

telegraf

JAEGGER

Prometheus

OpenTelemetry

Beats

fluentd

Grafana Agent

VECTOR

BY DATADOG

logstash
Collectors Zoo

- telegraf
- fluentbit
- Prometheus
- OpenTelemetry
- fluentd
Collectors Zoo

OpenTelemetry
where it all started?
de facto k8s telemetry data collection standards in early 2023
logs

metrics

traces

Prometheus
logs

a lot of plugins

community, documentation

written in Ruby
logs

a lot of plugins

community, documentation

written in Ruby
Exhibit 1 - kubeclient gem

used by more than 1k projects

1 (one) active maintainer
Exhibit 2 - multithreading vs HPA

1 worker = 1 thread

resources.requests.cpu = 2000
autoscaling.targetCPUUtilizationPercentage = 50
crazy fast

great memory usage

written in C
metrics

community, documentation

memory usage

database, not a forwarder *
traces?
traces

OpenTelemetry
why?
OpenTelemetry 101
No OpenTelemetry backend
Vendor support

dynatrace

New Relic

Uptrace

elastic

sumo logic

Aspecto

Grafana

splunk

honeycomb.io

DATADOG

Lightstep from ServiceNow

APPDYNAMICS
Cumulative number of companies contributing by quarter Q2 2019-Q2 2023

www.cncf.io/reports/opentelemetry-project-journey-report/
OpenTelemetry

Specification

Language SDKs

Collector
Specification

Guidelines - cross language requirements and expectations for all implementations

Semantic conventions

API, SDK

OTLP

https://github.com/open-telemetry/opentelemetry-specification
Specification

Community driven industry standard

https://github.com/open-telemetry/opentelemetry-specification
OpenTelemetry

Specification

Language SDKs

Collector
OpenTelemetry

Specification

Language SDKs

Collector
Language SDKs
Separate Collection

Infra/Host/VM/Pod/Container

Application

Logging libraries
Tracing libraries
Metric libraries

Tracing Agent (e.g., Jaeger Agent)

Metric Agent (e.g., OpenCensus Agent)

Log Collection Agent (e.g., FluentBit)

Logging Backend

Tracing Backend

Metric Backend

OpenTelemetry Collection

Infra/Host/VM/Pod/Container

Application

Auto and Manual instrumentation

System Logs

App Logs

Traces

App metrics

Host metrics

Infra Attributes

Logs

Traces

Metrics

Enrichment Processor

Exporter

Correlated Telemetry

Backend(s)

Logs correlate

Traces correlate

Metrics correlate

Instrumentation

Separate Collection

Infra/Host/VM/Pod/Container

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logging libraries</td>
</tr>
<tr>
<td>Tracing libraries</td>
</tr>
<tr>
<td>Metric libraries</td>
</tr>
</tbody>
</table>

OpenTelemetry Collection

Infra/Host/VM/Pod/Container

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto and Manual instrumentation</td>
</tr>
</tbody>
</table>
## Language SDKs

<table>
<thead>
<tr>
<th>Feature</th>
<th>Optional</th>
<th>Go</th>
<th>Java</th>
<th>JS</th>
<th>Python</th>
<th>Ruby</th>
<th>Erlang</th>
<th>PHP</th>
<th>Rust</th>
<th>C++</th>
<th>.NET</th>
<th>Swift</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TracerProvider</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create TracerProvider</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Get a Tracer</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Get a Tracer with schema_url</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td><strong>Associate Tracer with InstrumentationScope</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safe for concurrent calls</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Shutdown (SDK only required)</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>ForceFlush (SDK only required)</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Trace / Context interaction</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get active Span</td>
<td>N/A</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Set active Span</td>
<td>N/A</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Tracer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create a new Span</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Documentation defines adding attributes at span creation as preferred</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Get active Span</td>
<td>N/A</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

OpenTelemetry

Specification

Language SDKs

Collector
OpenTelemetry

Specification

Language SDKs

Collector
OpenTelemetry Collector 101
Collection
OpenTelemetry Collector

https://opentelemetry.io/docs/collector/
OpenTelemetry Collector

Receivers

https://opentelemetry.io/docs/collector/
OpenTelemetry Collector

Receivers

Processors

https://opentelemetry.io/docs/collector/
OpenTelemetry Collector
aka **otelcol**

[Diagram of OpenTelemetry Collector]

**OTEL COLLECTOR**

[Link to OpenTelemetry Collector documentation]
https://opentelemetry.io/docs/collector/
Host Metrics Receiver

<table>
<thead>
<tr>
<th>Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>beta</td>
</tr>
<tr>
<td>Supported pipeline types</td>
<td>metrics</td>
</tr>
<tr>
<td>Distributions</td>
<td>core, contrib, observiq, splunk, sumo</td>
</tr>
</tbody>
</table>

The Host Metrics receiver generates metrics about the host system scraped from various sources. This is intended to be used when the collector is deployed as an agent.

Getting Started

The collection interval, root path, and the categories of metrics to be scraped can be configured:

```yaml
hostmetrics:
  collection_interval: <duration> # default = 1m
  root_path: <string>
  scrapers:
    <scraper1>:
    <scraper2>:
    ...
```

The available scrapers are:

<table>
<thead>
<tr>
<th>Scraper</th>
<th>Supported OSs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu</td>
<td>All except Mac[1]</td>
<td>CPU utilization metrics</td>
</tr>
<tr>
<td>disk</td>
<td>All except Mac[1]</td>
<td>Disk I/O metrics</td>
</tr>
<tr>
<td>load</td>
<td>All</td>
<td>CPU load metrics</td>
</tr>
<tr>
<td>filesystem</td>
<td>All</td>
<td>File System utilization metrics</td>
</tr>
</tbody>
</table>
Resource Detection Processor

<table>
<thead>
<tr>
<th>Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>beta: traces, metrics, logs</td>
</tr>
<tr>
<td>Distributions</td>
<td>contrib, observiq, splunk, sumo</td>
</tr>
</tbody>
</table>

The resource detection processor can be used to detect resource information from the host, in a format that conforms to the OpenTelemetry resource semantic conventions, and append or override the resource value in telemetry data with this information.

Supported detectors

Environment Variable

Reads resource information from the OTEL_RESOURCE_ATTRIBUTES environment variable. This is expected to be in the format `<key1>=<value1>,<key2>=<value2>,...`, the details of which are currently pending confirmation in the OpenTelemetry specification.

Example:

```yaml
processors:
  resourcedetection/env:
    detectors: [env]
    timeout: 2s
    override: false
```

System metadata

Note: use the Docker detector (see below) if running the Collector as a Docker container.

Queries the host machine to retrieve the following resource attributes:

- host.name
- host.id
- os.type

https://github.com/open-telemetry/opentelemetry-collector-contrib/tree/v0.78.0/processor/resourcedetectionprocessor
OTLP gRPC Exporter

<table>
<thead>
<tr>
<th>Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>traces stable</td>
</tr>
<tr>
<td>metrics</td>
<td>stable</td>
</tr>
<tr>
<td>logs</td>
<td>beta</td>
</tr>
<tr>
<td>Supported pipeline types</td>
<td>traces, metrics, logs</td>
</tr>
<tr>
<td>Distributions</td>
<td>core, contrib</td>
</tr>
</tbody>
</table>

Export data via gRPC using OTLP format. By default, this exporter requires TLS and offers queued retry capabilities.

Getting Started

The following settings are required:

- **endpoint** (no default): host:port to which the exporter is going to send OTLP trace data, using the gRPC protocol. The valid syntax is described [here](https://github.com/open-telemetry/opentelemetry-collector/tree/v0.78.0/exporter/otlpexporter). If a scheme of `https` is used then client transport security is enabled and overrides the insecure setting.
- **tls**: see [TLS Configuration Settings](https://github.com/open-telemetry/opentelemetry-collector/tree/v0.78.0/exporter/otlpexporter) for the full set of available options.

Example:

```yaml
exporters:
  otlp:
    endpoint: otelcol2:4317
    tls:
      cert_file: file.cert
      key_file: file.key
  otlp/2:
    endpoint: otelcol2:4317
    tls:
      insecure: true
```
Put it all together

```yaml
receivers:
  hostmetrics:
    scrapers:
      memory:

processors:
  resourcedetection/detect-host-name:
    detectors:
      system:
        hostname_sources:
          - os

exporters:
  otlp:
    endpoint: otelcol2:4317

service:
  pipelines:
    metrics:
      receivers:
        - hostmetrics
      processors:
        - resourcedetection/detect-host-name
      exporters:
        - otlp
```
Data Pipeline

```yaml
receivers:
  hostmetrics:
    scrapers:
      memory:

processors:
  resourcedetection/detect-host-name:
    detectors:
      system:
        hostname_sources:
          os:

exporters:
  otlp:
    endpoint: otelcol2:4317

service:
  pipelines:
    metrics:
      receivers:
        - hostmetrics
      processors:
        - resourcedetection/detect-host-name
      exporters:
        - otlp
```
More Data **Pipelines**

```yaml
1 service:
2   pipelines:
3     metrics:
4       receivers: [hostmetrics]
5       processors: [resourcedetection/detect-host-name]
6       exporters: [otlp]
7
8     metrics/kafka:
9       receivers: [kafka]
10      exporters: [otlp/kafka]
11
12    logs:
13      receivers: [filelog]
14      exporters: [otlp]
15
16    logs/kafka:
17      receivers: [kafka]
18      exporters: [otlp, otlp/kafka]
19
20   traces:
21      receivers: [otlp, kafka]
22      processors: [resourcedetection/detect-host-name]
23      exporters: [otlp, otlp/kafka]
```
OTel Collector distros
OTel Collector

distros

Core distribution

github.com/open-telemetry/opentelemetry-collector - components
OTel Collector distros

Core distribution

github.com/open-telemetry/opentelemetry-collector - components

Contrib distribution

https://github.com/open-telemetry/opentelemetry-collector-contrib - components
OTel Collector **distros**

**Core** distribution

github.com/open-telemetry/opentelemetry-collector - components

**Contrib** distribution

https://github.com/open-telemetry/opentelemetry-collector-contrib - components

**Custom** distributions

- Grafana Agent
- Sumo Logic
- ADOT (AWS)
- ...

...
K8s observability

- Logs
- Metrics
- Traces

Metadata

sumo logic
K8s observability

Traces

Metadata

Logs

Metrics
K8s observability

Traces

Metadata

Logs

Metrics
Traces

~March 2020
data flood
K8s observability

Traces

Metadata

Logs

Metrics
K8s observability

Traces

Metadata

Logs

Metrics
Metadata

- Battle tested
- Single threaded
- Weak performance
- Ruby magic

February 2022
Metadata

- Battle tested
- Single threaded
- Weak performance
- Ruby magic

February 2022

- K8s Attributes in beta
- Go-lang performance
- Removed backpressure from Prometheus' remote-write
- Lowered Prometheus' memory
~ 38 CPUs
~ 38 CPUs
~ 13 CPUs
~ 38 CPUs

~ 13 CPUs
~ 220G of RAM
~ 220G of RAM

~ 75G of RAM
~ 220G of RAM

~ 75G of RAM
... memory after more tweaks?

~11G - 20x less
Metadata: instances

~85
Metadata: instances

Pods count otelcol metrics

A

collector=elasticsearch namespace=default statefulset=sumologic-fluentd-metrics metric=kube_statefulset_status_replicas | quantize using max

B

 collector=elasticsearch namespace=default statefulset=sumologic-otelcol-metrics metric=kube_statefulset_status_replicas | quantize using max

Chart

~85

~20

This Month
... instances after more tweaks?

~11 instances - 8x less
... instances after more tweaks?

Pods count otelcol metrics

Chart

Time Series

7 matching time series
K8s observability

- Traces
- Metadata
- Logs
- Metrics
K8s observability

Traces

Metadata

Logs

Metrics
Logs
Great CPU usage

Great memory usage

Hard to debug issues

Didn't support metrics and traces at the time (it does now)

Logs

June 2022
Great CPU usage

Great memory usage

Hard to debug issues

Didn't support metrics and traces at the time (it does now)

Filelog Receiver in beta

Great CPU usage

Reasonable memory usage

No major feature missing

June 2022
K8s observability

Traces

Metadata

Logs

Metrics
K8s observability

Traces

Metadata

Logs

Metrics
Metrics
Metrics

Remote write

Load balancing

Memory usage

September 2023
Metrics

- Remote write
- Load balancing
- Memory usage

September 2023

- Prometheus receiver in beta
- Metric names quirks
- Small resource usage
Metrics: CPU sum

~ 0.7 CPUs
Metrics: CPU sum

~ 0.7 CPUs

~ 0.15 CPUs
Metrics: memory sum

~ 7.5G of RAM
Metrics: memory sum

- Approximately 7.5G of RAM
- Approximately 1.6G of RAM
K8s observability - OTel edition

Logs

Metrics

Traces

Metadata

sumo logic
K8s observability - OTel edition

Logs → Metadata → sumo logic

Metrics → sumo logic

Traces → sumo logic
K8s observability - OTel edition

Logs

Metrics

Traces

Metadata

Logos of Sumo Logic and New Relic
OTC issues
Low hanging fruit bugs in 2022

- [receiver/filelog/receiver] Filelog/receiver doesn't read from file in case there are two or more empty lines at the beginning
  
  #10128 by perk was closed on Jul 18, 2022

- [receiver/filelog/receiver] Filelog/receiver reads the same file twice in case there is an empty line at the beginning

  #10127 by perk was closed on Jul 18, 2022

- [receiver/filelog/receiver] Filelog/receiver reads the same file in the loop in case there is an empty line between logs

  #10126 by perk was closed on Jul 18, 2022
State as of 2023

306 Open  1,011 Closed

- **Flaky test: TestRabbitmqIntegration** [bug, flaky test, receiver/rabbitmq]
  #22134 opened yesterday by atoulme

- **The awsccloudwatch receiver isn’t able to filter cloudwatch log streams reliably using a stream prefix** [bug]
  needs triage, receiver/awsccloudwatch
  #22123 opened yesterday by mishtala-sumo

- **replace_all_patterns function for attribute keys doesn’t not expand regex** [bug, processor/transform]
  #22094 opened 2 days ago by vainsisud

- **[exporter/f5cloud] The link https://portal.cloudservices.f5.com is invalid** [bug, exporter/f5cloud]
  #22077 opened 3 days ago by astencel-sumo

- **Metric export is failing due to buffer full error on Open telemetry** [bug, needs triage]
  #22071 opened 3 days ago by vermaprateek695

- **[receiver/prometheus] Histograms without buckets are dropped** [bug, receiver/prometheus]
  #22070 opened 3 days ago by swiatekm-sumo
Some more #OpenTelemetry

opentelemetry.io

opentelemetry.io/docs/collector

opentelemetry.io/community
Community

- OpenTelemetry org at GitHub
- Mailing lists
- GitHub discussions or Slack channel
- Calendar
- Community page
Thank you!

Marcin "Perk" Stożek

@marcinstozek / perk.pl

slides.com/perk/journey-to-opentelemetry-collector