

OpenTelemetry Collector の Connectors って何者？

オブザービリティ再入門 @Mackerel - Tomonori Hayashi

Tomonori Hayashi

- NTT コミュニケーションズ イノベーションセンター所属
 - ノーコード時系列分析ツール「**Node-AI**」の開発/運用
 - ソフトウェアエンジニア
 - Front: TypeScript - React/Next.js
 - Infra: **Google Cloud**
- Google Cloud Partner Top Engineer 2024
- Google Cloud All Certifications
- Favorite Word
 - **class SRE implements DevOps**



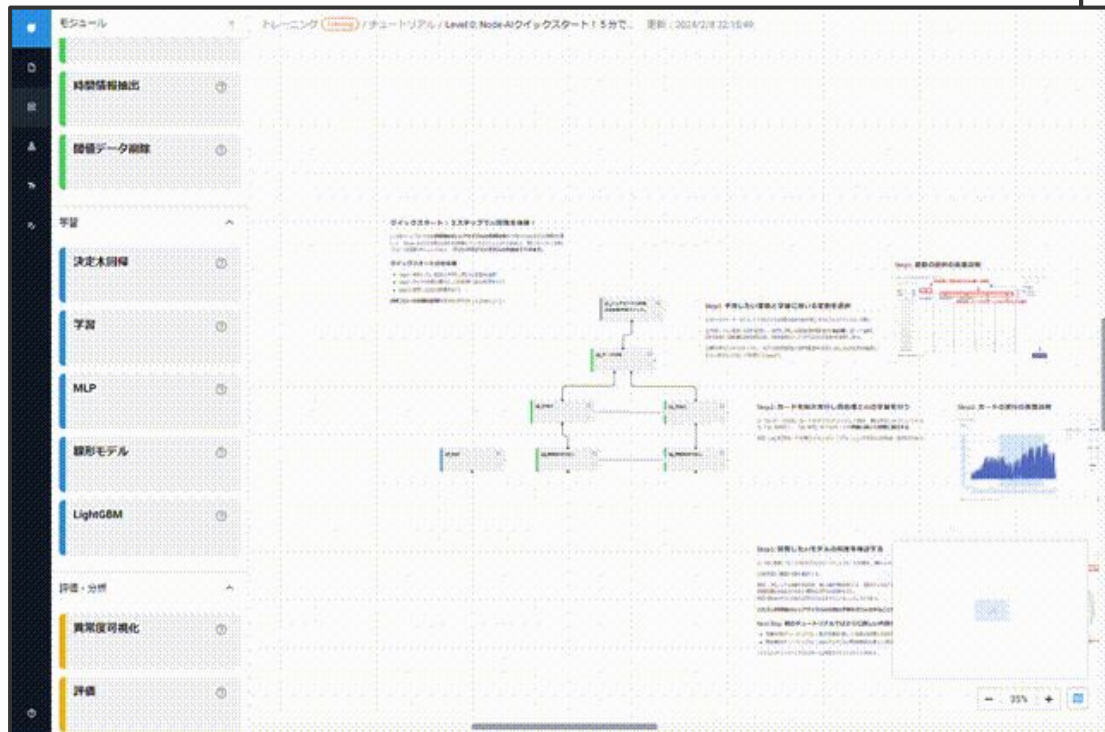
@pHaya72



@t_hayashi

Node-AI の紹介

- ノーコードで AI モデルを作成できる WEB アプリケーション
- カードを直感的につなげるだけで時系列データの前処理から AI モデルの学習・評価までのパイプラインを作成・実行できる
- 技術スタック
 - TypeScript + React / Next
 - Python + Django
 - C# + ASP.NET Core
 - Kubernetes
 - **Google Cloud**
 - Scikit-learn / Tensorflow / Pytorch



オブザーバビリティ再入門ということで OpenTelemetry Collector を思い出す

OpenTelemetry Collector の概要

計装されたサービスとバックエンドを繋ぐプロキシ

“The OpenTelemetry Collector offers a vendor-agnostic implementation of how to receive, process and export telemetry data. It removes the need to run, operate, and maintain multiple agents/collectors. This works with improved scalability and supports open source observability data formats (e.g. Jaeger, Prometheus, Fluent Bit, etc.)”

— 「**OpenTelemetry Document**」
Collector



ベンダーニュートラルな処理

テレメトリデータの収集 / 処理 / エクスポートについて、ベンダー依存しない実装を提供

複数のエージェントの不要化

複数のエージェントやコレクターを個別に管理する必要性を解消

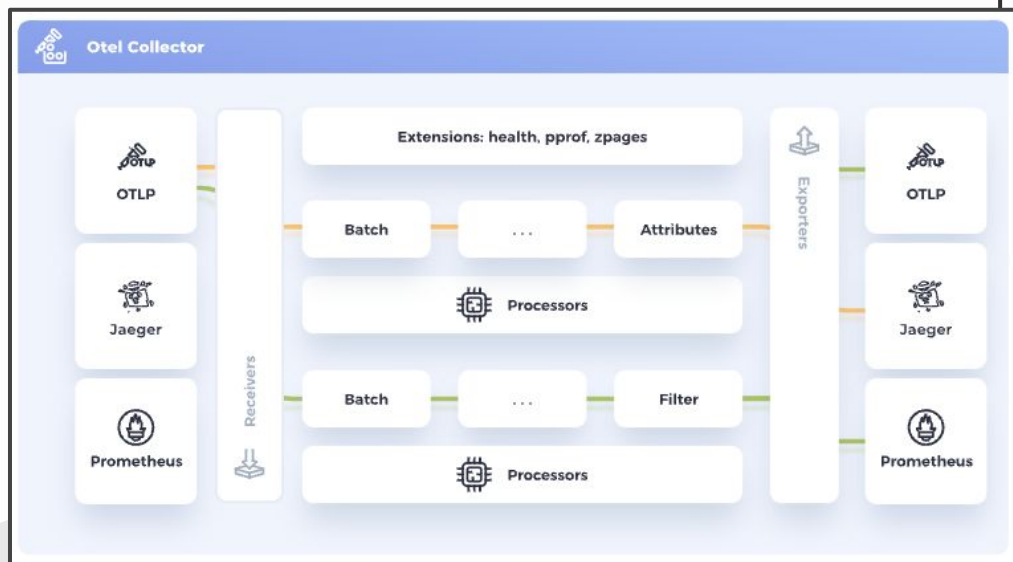
バックエンドとの互換性

Prometheus などのオープンソースのオブザーバビリティデータ形式をサポート、それらを各種バックエンドに送信

OpenTelemetry Collector の概要

構成要素

- Receivers / Exporters
- Processors : エクスポート前に変更または変換する
- Extensions : データ処理に直接関係しないタスクを実行する
- Connectors : 2つのパイプラインを接続してエクスポートとレシーバーの両方の役割を果たす

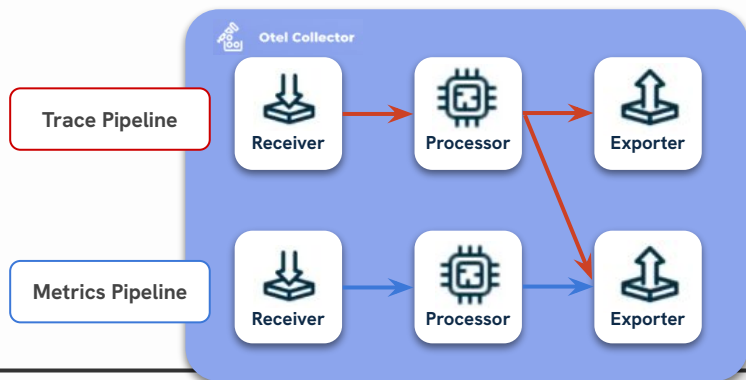


Connectors に注目する

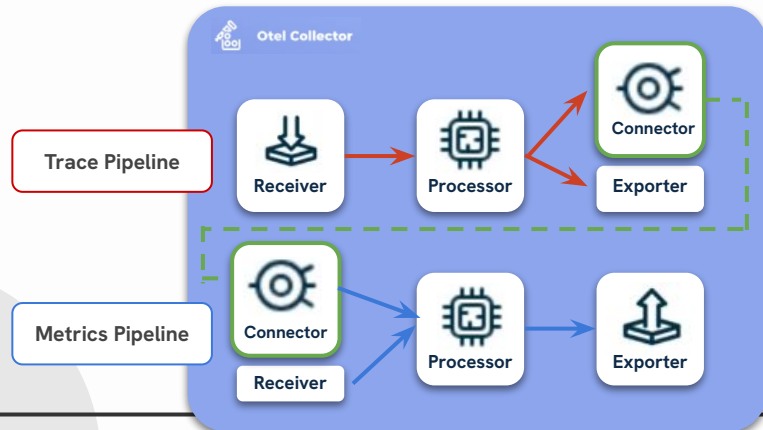
Connectors の概要

- ❑ 異なるパイプラインを接続することで、**テレメトリデータをパイプライン間で送る手段**として機能する
- ❑ 従来はトレースやスパン情報からメトリクスに変換するのはプロセッサーで行われていた(左図)
 - 一部のプロセッサーは処理後に異なるパイプラインからエクスポートするワークアラウンドが利用されていた
- ❑ 新しいアーキテクチャとしてコネクタが登場してこのようなワークアラウンドが解消された(右図)
 - 上記のプロセッサーはコネクタ登場により非推奨となった

従来のアーキテクチャ



新しいアーキテクチャ



Connectors の種類

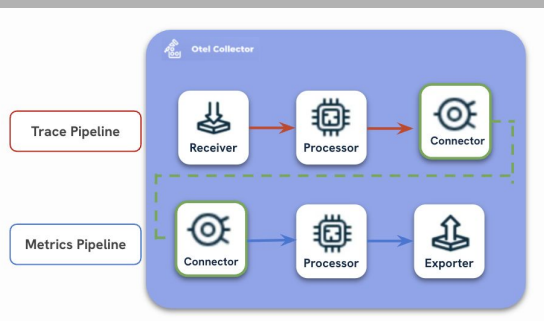
OpenTelemetry Collector のリポジトリ

- <https://github.com/open-telemetry/opentelemetry-collector> (core)
高い安定性を重視していて、一般的なユースケースをカバーする機能を提供
- <https://github.com/open-telemetry/opentelemetry-collector-contrib> (contrib)
Core に比べて安定性は若干低い、Core では提供されていない特殊な機能や拡張機能を提供

Connectors

- opentelemetry-collector-contrib リポジトリに存在する
- 9 種類のコネクタが存在する

サポートされている Pipeline Type



Span Metrics Connector

Status	
Distributions	contrib
Issues	open 14 closed 48
Code Owners	@portertech , @Frapschen Seeking more code owners!
Emeritus	@alberteoh

Supported Pipeline Types

Exporter Pipeline Type	Receiver Pipeline Type	Stability Level
traces	metrics	alpha

opentelemetry-collector-contrib / connector /

[ankitpatel96](#) [chore] add myself as a codeowner to datadogconnector

Name

..

countconnector

datadogconnector

exceptionsconnector

failoverconnector

grafanacloudconnector

roundrobinconnector

routingconnector

servicegraphconnector

spanmetricsconnector

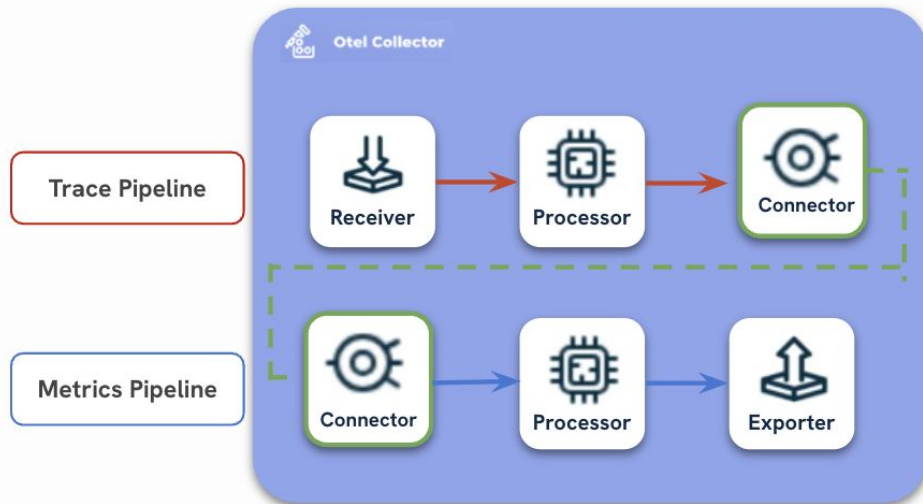
Span Metrics Connector

❏ スパンデータからリクエスト、エラー、期間(R.E.D)を集計してメトリクスに

- リクエスト数 (Request)
- エラー数 (Error)
- 期間 (Duration)

```
receivers:  
  otp:  
  
exporters:  
  googlecloud:  
  
connectors:  
  spanmetrics:  
    histogram:  
      explicit:  
        buckets: [10ms, 100ms, 500ms, 750ms, 1s, 2s, 5s, 10s]  
        metrics_flush_interval: 15s  
        namespace: "spanconnector"  
    exemplars:  
      enabled: true  
  
service:  
  pipelines:  
    traces:  
      receivers: [otp]  
      exporters: [spanmetrics]  
    metrics:  
      receivers: [spanmetrics]  
      exporters: [googlecloud]
```

OpenTelemetry Collector
のコンフィグファイル



Span Metrics Connector

- receivers や exporters と同様に **connectors** を記述
- namespace** を指定するとメトリクスのカテゴリ名に

Google Cloud の Cloud Monitoring
メトリクスを可視化できる

```
receivers:  
  otp:  
    namespace: "spanconnector"  
exporters:  
  googlecloud:  
    namespace: "spanconnector"  
connectors:  
  spanmetrics:  
    histogram:  
      explicit:  
        buckets: [10ms, 100ms, 500ms, 750ms, 1s, 2s, 5s, 10s]  
        metrics_flush_interval: 15s  
        namespace: "spanconnector"  
    exemplars:  
      enabled: true
```

otel-collector.yml

```
service:  
  pipelines:  
    traces:  
      receivers: [otlp]  
      exporters: [spanmetrics]  
      metrics:  
        receivers: [spanmetrics]  
        exporters: [googlecloud]
```

OpenTelemetry Collector
のコンフィグファイル

The screenshot shows the Google Cloud Metrics Explorer interface. The main view displays a list of metric categories under the heading '有効な指標カテゴリ' (Valid Metric Categories). The 'Spanconnector' category is highlighted, showing it contains 2 metrics. A modal window is open over the 'Spanconnector' category, listing the metrics: 'Workload/spanconnector.calls' and 'Workload/spanconnector.duration'. The 'Workload/spanconnector.duration' metric is selected. On the right side of the interface, a detailed view for the selected metric is shown, including its Name, Description, Metric, Resource types, Unit, Kind, and Value type.

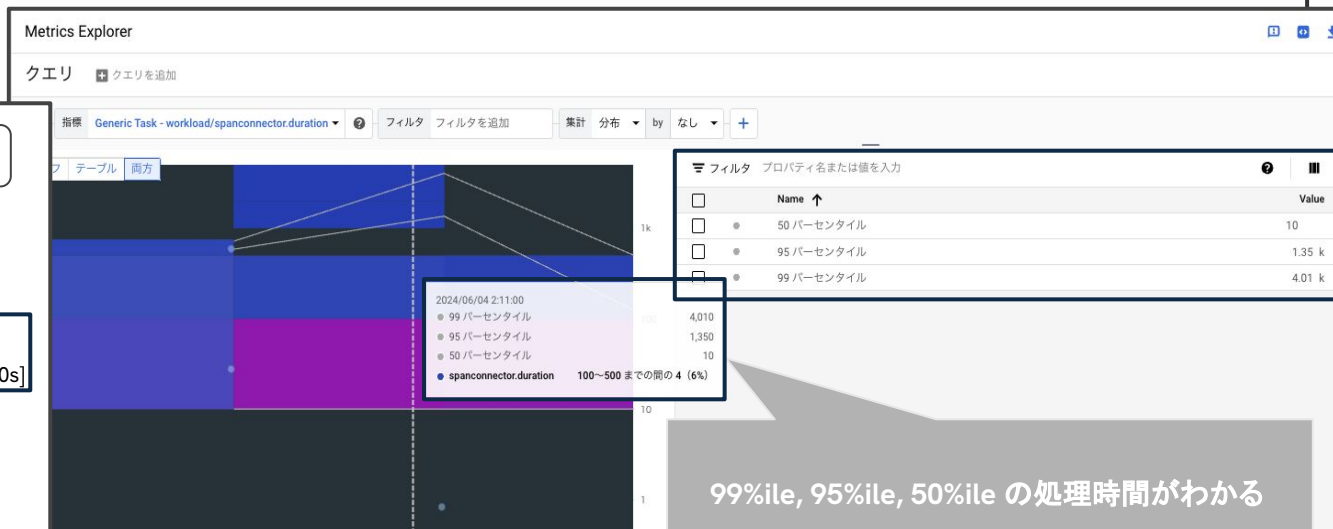
Name	workload/spanconnector.duration
Description	
Metric	workload.googleapis.com/spanconnector.duration
Resource types	aws_ec2_instance, aws_lambda_function, aws_sqs_queue, baremetalsolution.googleapis.com/instance, cloud_composer_environment, その他 15 個
Unit	1
Kind	CUMULATIVE
Value type	DISTRIBUTION

Span Metrics Connector

📄 Workload/spanconnector.duration を選択するとヒートマップで可視化される

```
receivers:  
  otp:  
  
exporters:  
  googlecloud:  
  
connectors:  
  spanmetrics:  
    histogram:  
      explicit:  
        buckets: [10ms, 100ms, 500ms, 750ms, 1s, 2s, 5s, 10s]  
    metrics_flush_interval: 15s  
    namespace: "spanconnector"  
    exemplars:  
      enabled: true  
  
service:  
  pipelines:  
    traces:  
      receivers: [otp]  
      exporters: [spanmetrics]  
    metrics:  
      receivers: [spanmetrics]  
      exporters: [googlecloud]
```

otel-collector.yml



Span Metrics Connector

- スパンからメトリクスを生成しているため、**スパン単位での duration が時系列で見れる**

receivers:
otlp:

otel-collector.yml

exporters:
googlecloud:

connectors:
spanmetrics:

```
histogram:  
  explicit:  
    buckets: [10ms, 100ms, 500ms, 750ms, 1s, 2s, 5s, 10s]
```

```
metrics_flush_interval: 15s  
namespace: "spanconnector"
```

```
exemplars:  
  enabled: true
```

```
service:  
  pipelines:  
    traces:  
      receivers: [otlp]  
      exporters: [spanmetrics]  
  metrics:  
    receivers: [spanmetrics]  
    exporters: [googlecloud]
```

Metrics Explorer

クエリ [クエリを追加](#) → [比率を作成](#)

PromQL で少し変換

```
1 histogram_quantile(0.95,rate(workload_googleapis_com:spanconnector_duration_bucket{monitored_resource="generic_task"}[${__interval}])))
```

クエリの例

テーブル 両方



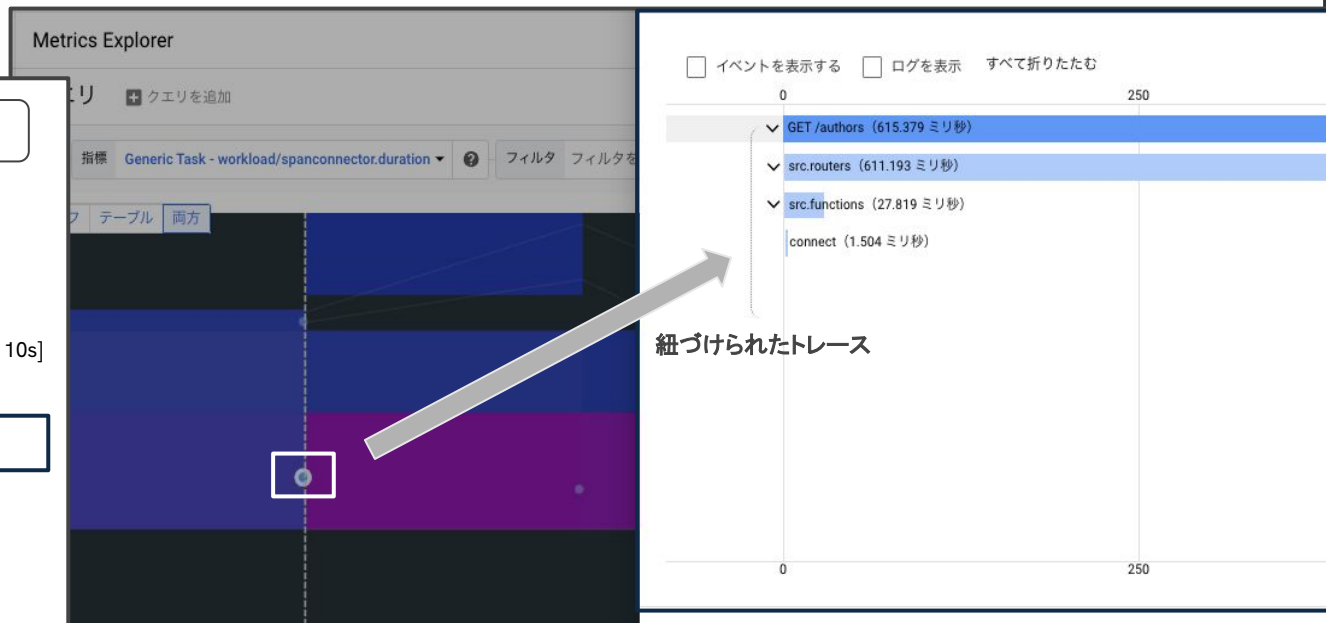
フィルタ プロパティ名または値を入力

	span_kind ↓	span_name	Value	
<input type="checkbox"/>	★	SPAN_KIND_SERVER	GET /authors	3.8 k
<input type="checkbox"/>	●	SPAN_KIND_SERVER	GET /books	95.5
<input type="checkbox"/>	●	SPAN_KIND_SERVER	POST /books	10
<input type="checkbox"/>	◆	SPAN_KIND_INTERNAL	GET /authors http send	10
<input type="checkbox"/>	▼	SPAN_KIND_INTERNAL	GET /books http send	10
<input type="checkbox"/>	●	SPAN_KIND_INTERNAL	POST /books http send	10
<input type="checkbox"/>	✚	SPAN_KIND_INTERNAL	src.functions	91
<input type="checkbox"/>	✱	SPAN_KIND_INTERNAL	src.routers	3.5 k
<input type="checkbox"/>	●	SPAN_KIND_CLIENT	connect	10

Span Metrics Connector

- ❑ エクザンプラーをオンにするとメトリクスにトレース情報が入る
- ❑ メトリクスの可視化の中でトレースを参照できるようになる

```
receivers:  
  otlp: otel-collector.yml  
  
exporters:  
  googlecloud:  
  
connectors:  
  spanmetrics:  
    histogram:  
      explicit:  
        buckets: [10ms, 100ms, 500ms, 750ms, 1s, 2s, 5s, 10s]  
        metrics_flush_interval: 15s  
        namespace: "spanconnector"  
    exemplars:  
      enabled: true  
  
service:  
  pipelines:  
    traces:  
      receivers: [otlp]  
      exporters: [spanmetrics]  
  metrics:  
    receivers: [spanmetrics]  
    exporters: [googlecloud]
```



紐づけられたトレース

まとめ

OpenTelemetry Collector の Connectors

- ❑ 異なるパイプラインを接続することで、テレメトリデータをパイプライン間で送る手段として機能する

Span Metrics Connector

- ❑ スパンデータからリクエスト、エラー、期間(R.E.D)を集計してメトリクスに
→ バックエンドで集計しなくてもメトリクスとしてバックエンドに送れる
- ❑ エグザンプラーをオンにすることでメトリクスとトレースを紐づけられる
→ 分析する際に可視化されたメトリクスからトレースへの遷移がスムーズに

ユースケースに合うコネクターを選択して分析に活かせるテレメトリーを集めよう！

※ 各コネクターの Stability Level は要確認

Thanks!

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

参考文献

- ★ オブザーバビリティ・エンジニアリング
- ★ 入門 監視

- ★ OpenTelemetry ドキュメント
 - <https://opentelemetry.io/docs/>
- ★ OpenTelemetry Collector の Span Metrics Connector を使ってメトリクスを生成してみる
 - <https://zenn.dev/k6s4i53rx/articles/2023-advent-calender-otel>
- ★ OpenTelemetry Collectorにログを集計させるために filelog receiverとcount connectorを使う
 - <https://zenn.dev/katzchang/articles/8ef357a35f0496>