

クラウドネイティブ時代に 進化し続けるFluentd

CloudNative Days Winter 2025
2025/11/19





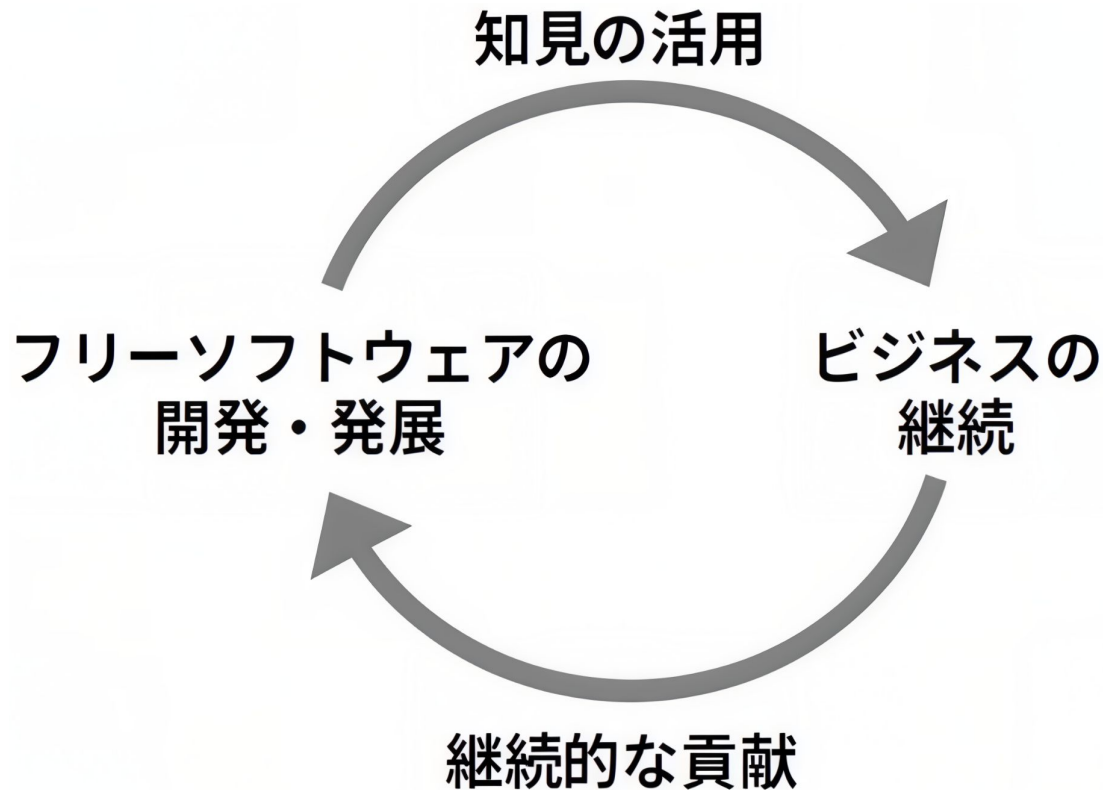
Shizuo Fujita (藤田 静男)

株式会社クリアコード
Fluentd メンテナー

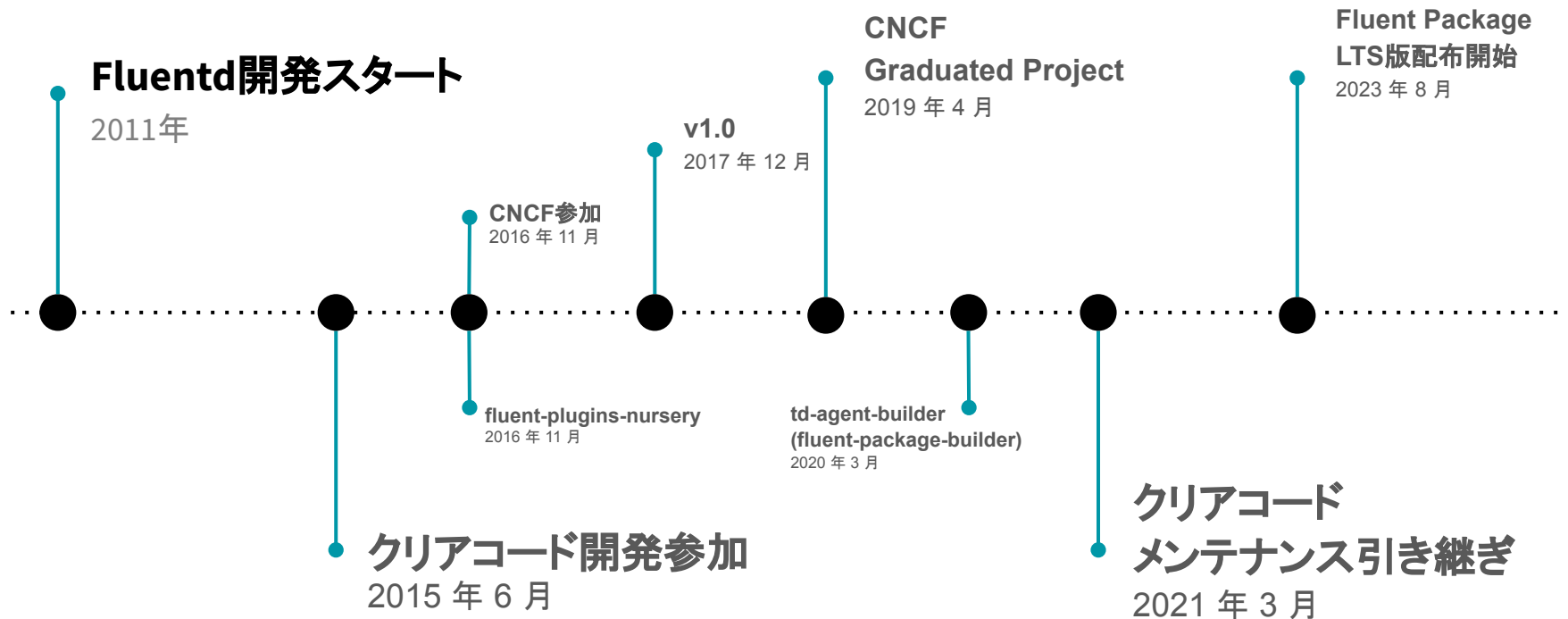


Daijiro Fukuda (福田 大司郎)

株式会社クリアコード
Fluentd メンテナー



Fluentdの歴史とクリアコードの関わり



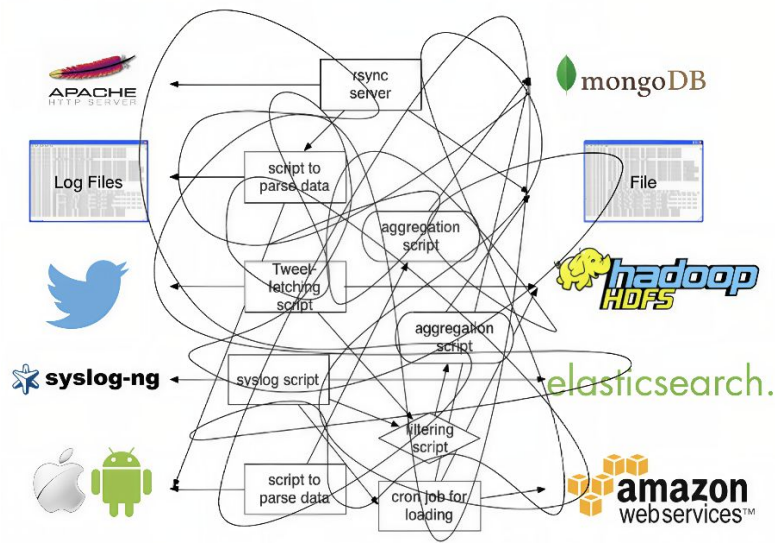
今日お話すこと

- Fluentdとは
- FluentdとCloud Native
- Fluentdの進化
- 最新版紹介
- OpenTelemetry対応

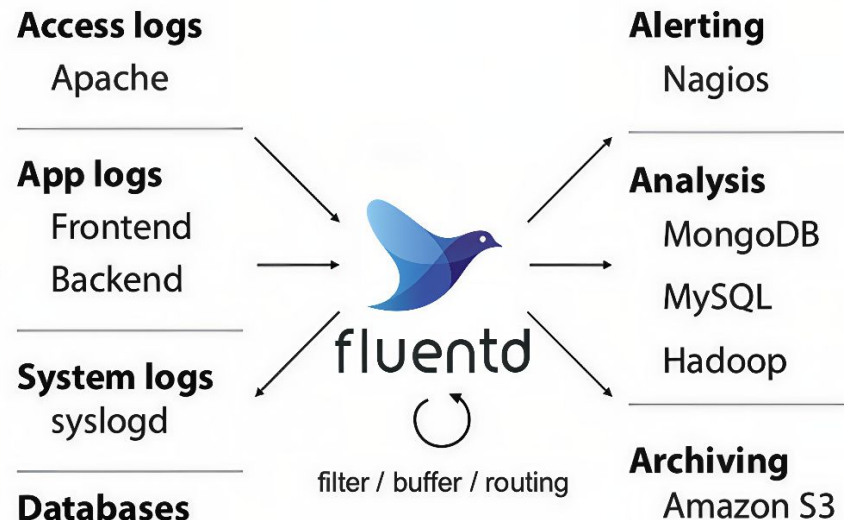
- Fluentdとは
- FluentdとCloud Native
- Fluentdの進化
- 最新版紹介
- OpenTelemetry対応

Fluentdが実現する世界

Without Fluentd



With Fluentd



Fluentd



Fluent Bit



Fluentdの配布形態

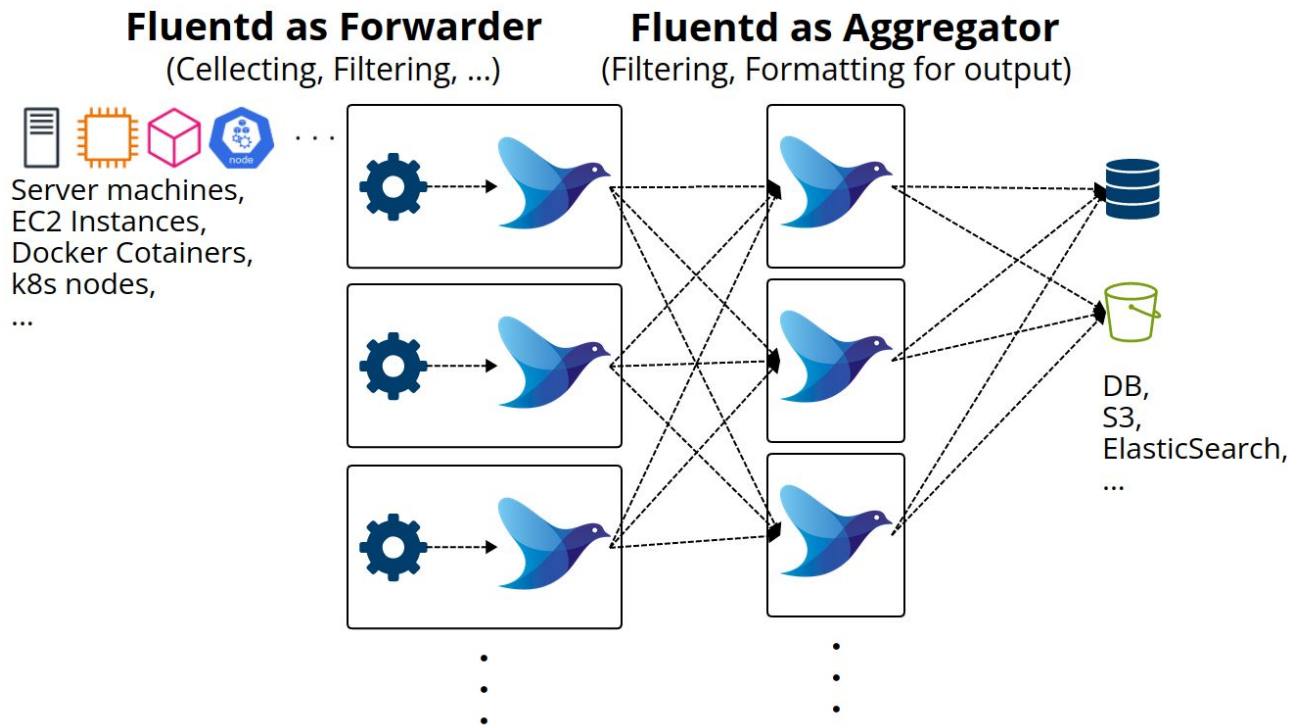
- Ruby gem
- Package
 - Fluent Package (fluent-package)
 - Treasure Agent (td-agent)
 - Calyptia Fluentd (calyptia-fluentd)
- Container
 - Docker images
 - Daemonset for Kubernetes



- 柔軟性(カスタマイズ性)
- スケールする
- オープンソース

- 柔軟性の高いプラグインシステム
 - 収集、フィルタリング、バッファリング、および送信といった主要な機能がすべてプラグインによって提供される
- 1000以上のプラグイン
- Ruby言語で新規のプラグインを簡単に自作可能
- RubyGemsのしくみによってプラグインを配布・利用可能

Fluentdの特徴: スケールする



※ Fluent Bitと組み合わせ可能

Fluentdの特徴: オープンソース

- 誰でも開発に参加できる
- プラグインをコミュニティで管理するしくみ
 - <https://github.com/fluent-plugins-nursery>

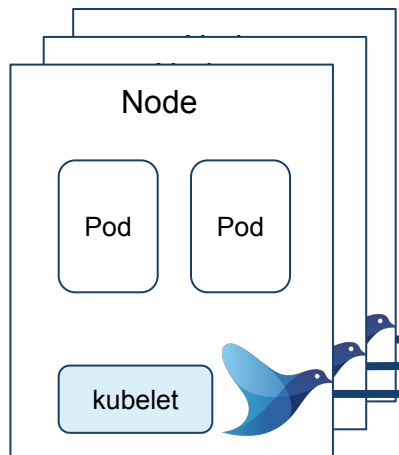
- Fluentdの特徴
 - 豊富なプラグイン、カスタマイズ性の高さ
- Fluent Bitの特徴
 - 軽量、高速
- 互いに転送可能
 - 例: エッジ処理をFluent Bit、集約・加工処理をFluentdで行う
- OpenTelemetry対応
 - Fluentdが未対応の状態が続いていたが、ついに部分的に対応

- Fluentdとは
- FluentdとCloud Native
- Fluentdの進化
- 最新版紹介
- OpenTelemetry対応

- 2019年にCNCFのGraduatedレベルに認定
- 分散システムの可観測性を実現する
- 近年、サブプロジェクトであるFluent Bitが、コンテナ環境に適したログ転送エージェントとして普及



kubernetes



elasticsearch

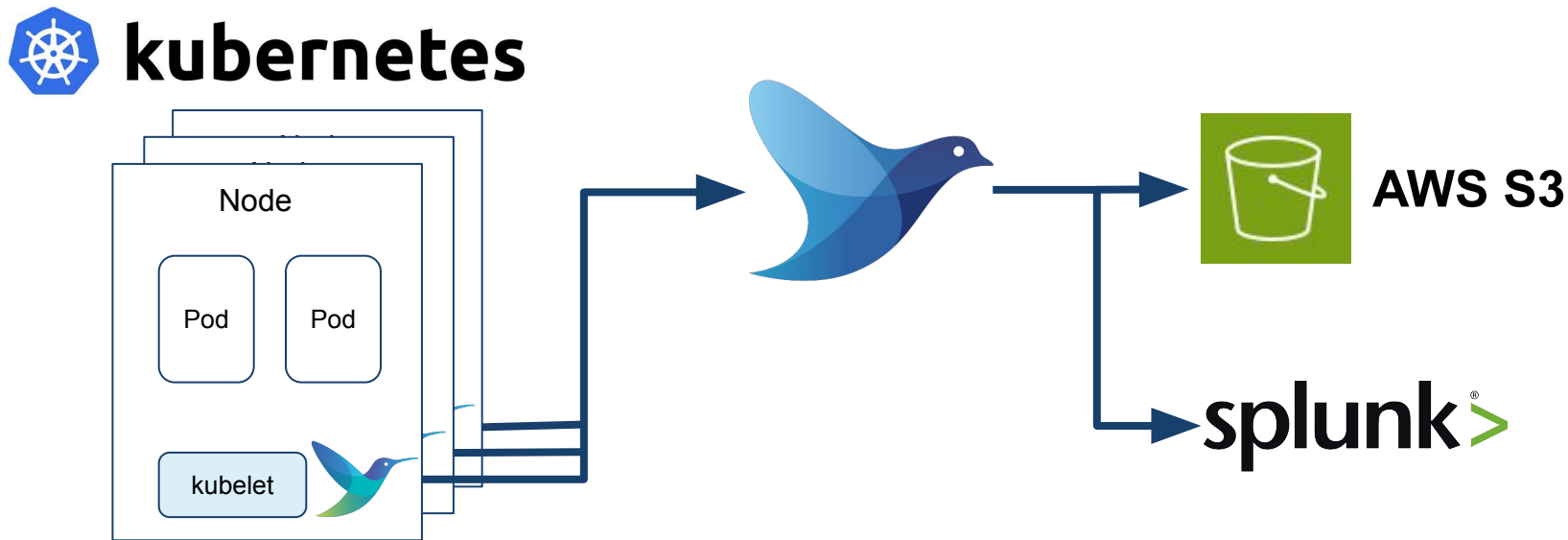
+



kibana

引用: Fluentd/Fluent Bit で実現する楽なKubernetesのログ運用
CloudNative Days Tokyo 2021

FluentdとFluent Bitの併用例



引用: Fluentd/Fluent Bit で実現する楽なKubernetesのログ運用
CloudNative Days Tokyo 2021

Fluentdはもう古い？



- Fluentdとは
- FluentdとCloud Native
- **Fluentdの進化**
- 最新版紹介
- OpenTelemetry対応

- 継続的な不具合修正による動作の安定化
- 継続的に便利な新機能を追加
- リリースサイクルの改善(LTS版の提供)

Fluentdの進化: 過去の主な不具合修正

年月	Fluentd	修正した不具合
2018年3月	1.1.2	マルチスレッドで転送が失敗することがある
2019年8月	1.7.0	圧縮データを解凍するときにログ欠損が生じることがある
2022年7月	1.15.1	マルチワーカーでのファイル追記出力がデータ破損することがある
2023年11月	1.16.3	データ収集がエラーを出さずに停止してしまうことがある
2024年3月	1.16.4	大きいデータのバッファ書き込みに失敗してログ欠損が生じることがある
2025年7月	1.19.0	設定によっては日本語などの非 ASCII文字が収集時に破損することがある

※ 公式Wiki「[影響度の高い不具合の一覧](#)」

不具合診断ツール

- Fluentd v1.12.1 (td-agent v4.1.0) ~ v1.16.2 (fluent-package v5.0.1) にかけて、ログ収集(in_tailプラグイン)がエラーなく止まってしまうことがある深刻な不具合があった
- 診断ツールを開発・提供
 - https://github.com/clear-code/fluent-tail_checker
 - [診断ツールガイド](#)

Fluentdの進化: 過去の機能追加

年月	Fluentd	修正した不具合
2022年6月	1.15.0	YAML形式の設定ファイルをサポート
2023年3月	1.16.0	バッファファイル破損時の対応強化
2024年4月	1.17.0	より柔軟な収集対象ファイルパスの指定を可能に
2024年11月	1.18.0	組み込みRubyコードをArrayやHash表記内でも利用可能に
	1.18.0	ゼロダウンタイム・リスタート機能を追加
2025年7月	1.19.0	より高速なzstd圧縮形式に対応

※ <https://www.fluentd.org/blog/archive/>

- 長期サポート(LTS)版の提供を開始
 - サポート期間を事前アナウンス(約2年間)
 - セキュリティー・バグフィックスのみを提供
- 現在のLTS版
 - Fluentd v1.16.x == Fluent Package v5.0.x
 - 2025年いっぱいまで
 - Fluentd v1.19.x == Fluent Package v6.0.x
 - 2027年いっぱいまで

- Fluentdとは
- FluentdとCloud Native
- Fluentdの進化
- 最新版紹介
- OpenTelemetry対応

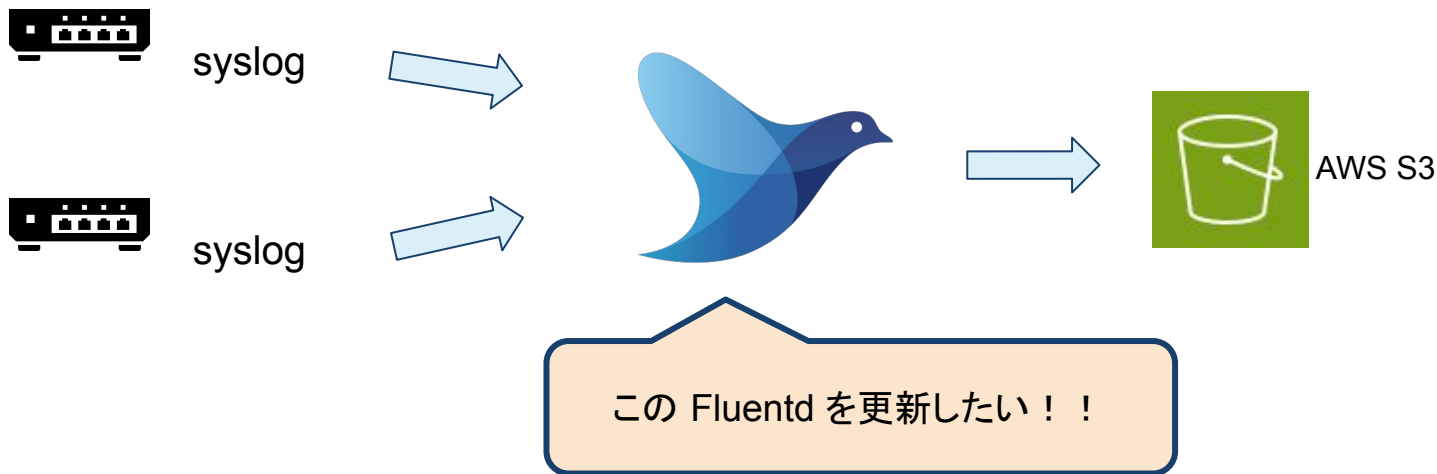
最新版: Fluentd v1.19 / Fluent Package v6



- ゼロダウンタイム・リスタート
- 耐障害性の強化
- パフォーマンス改善
- OpenTelemetryの一部対応

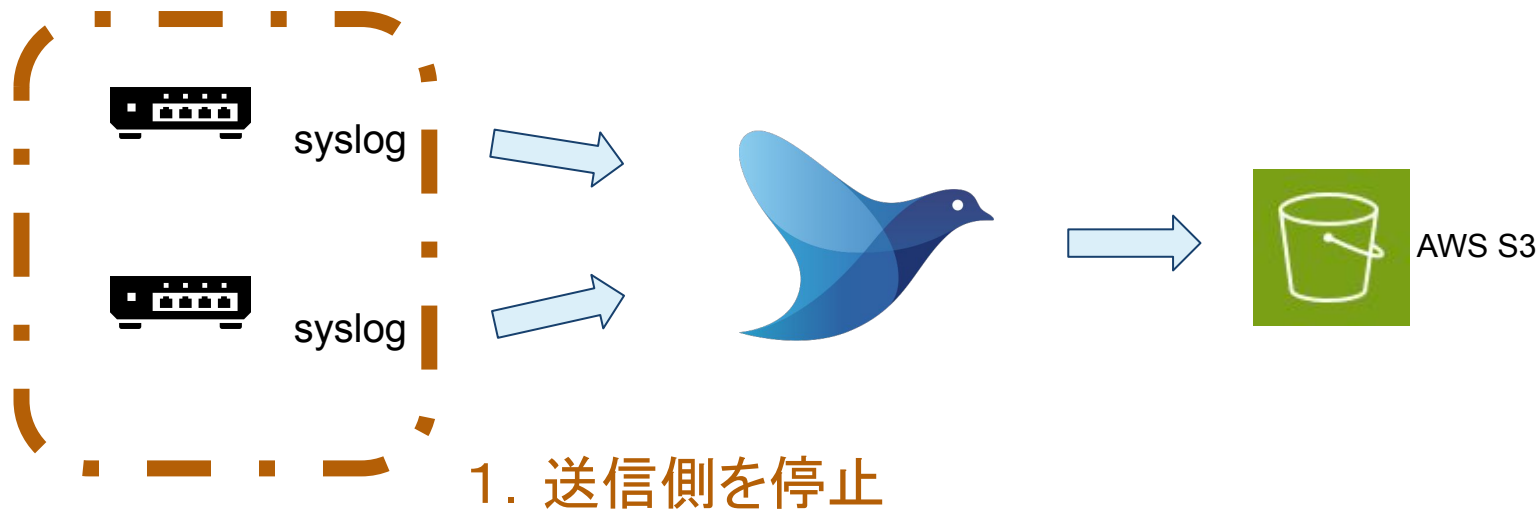
これまでの Fluentd 更新作業

- これまで、アプリケーションのログ欠損なしで稼働中のFluentd設定ファイル更新、Fluentdアップデートが煩雑



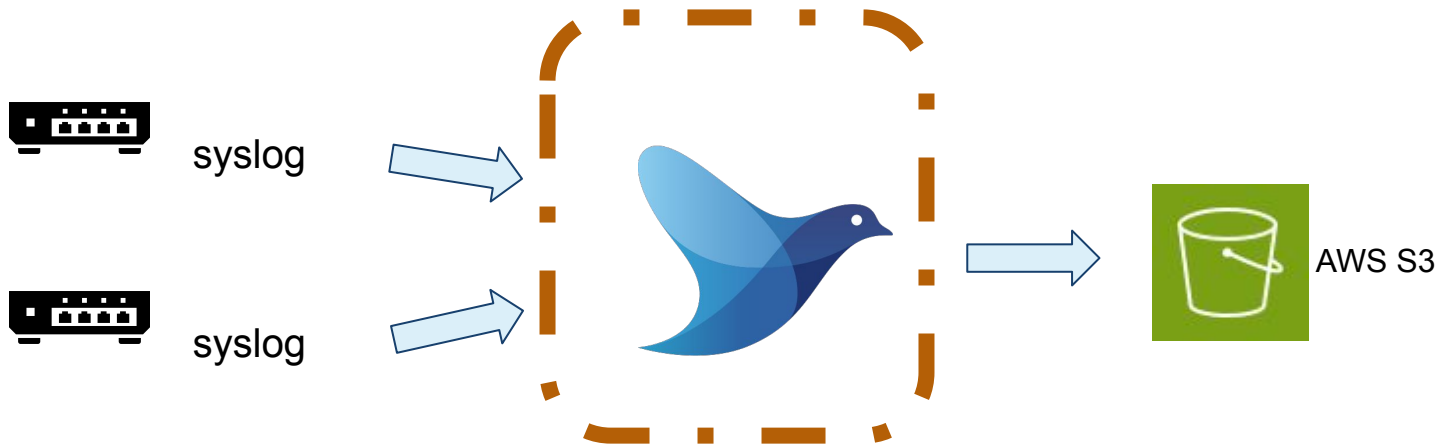
これまでの Fluentd 更新作業

- これまで、アプリケーションのログ欠損なしで稼働中のFluentd設定ファイル更新、Fluentdアップデートが煩雑



これまでの Fluentd 更新作業

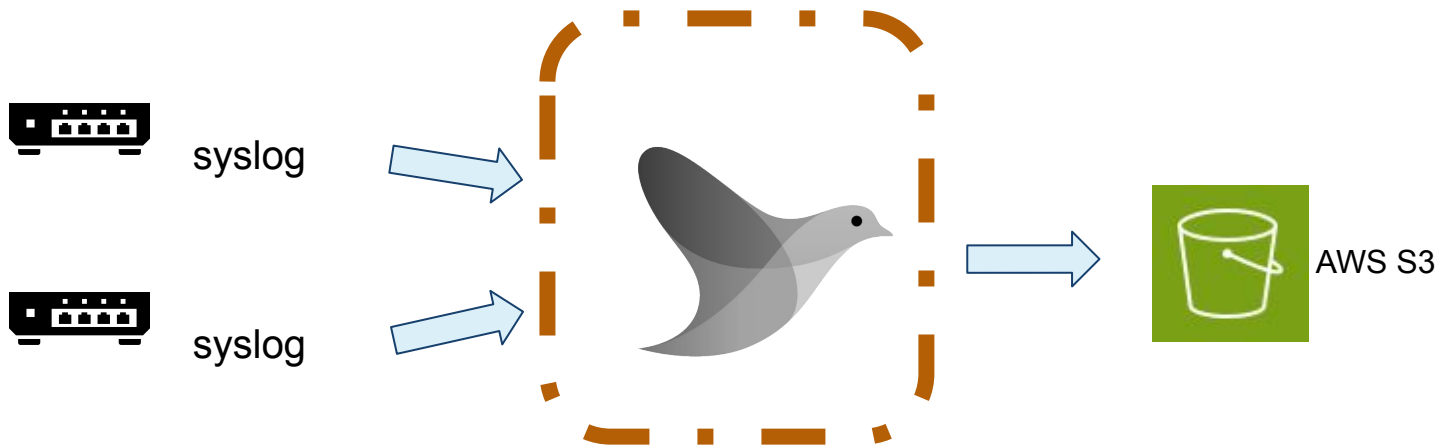
- これまで、アプリケーションのログ欠損なしで稼働中のFluentd設定ファイル更新、Fluentdアップデートが煩雑



2. Fluentd を停止

これまでの Fluentd 更新作業

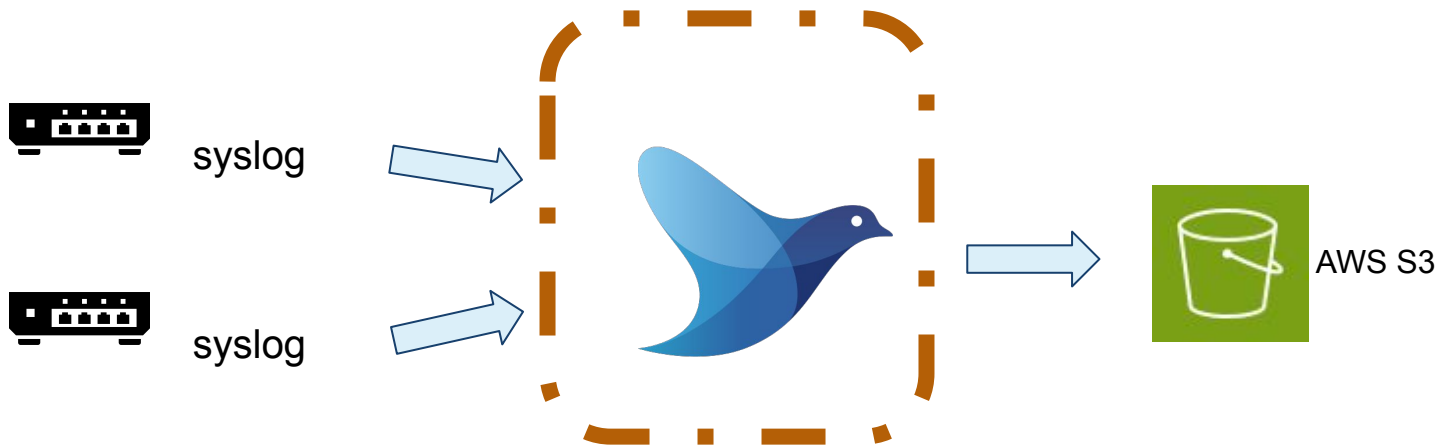
- これまで、アプリケーションのログ欠損なしで稼働中のFluentd設定ファイル更新、Fluentdアップデートが煩雑



3. Fluentd を更新

これまでの Fluentd 更新作業

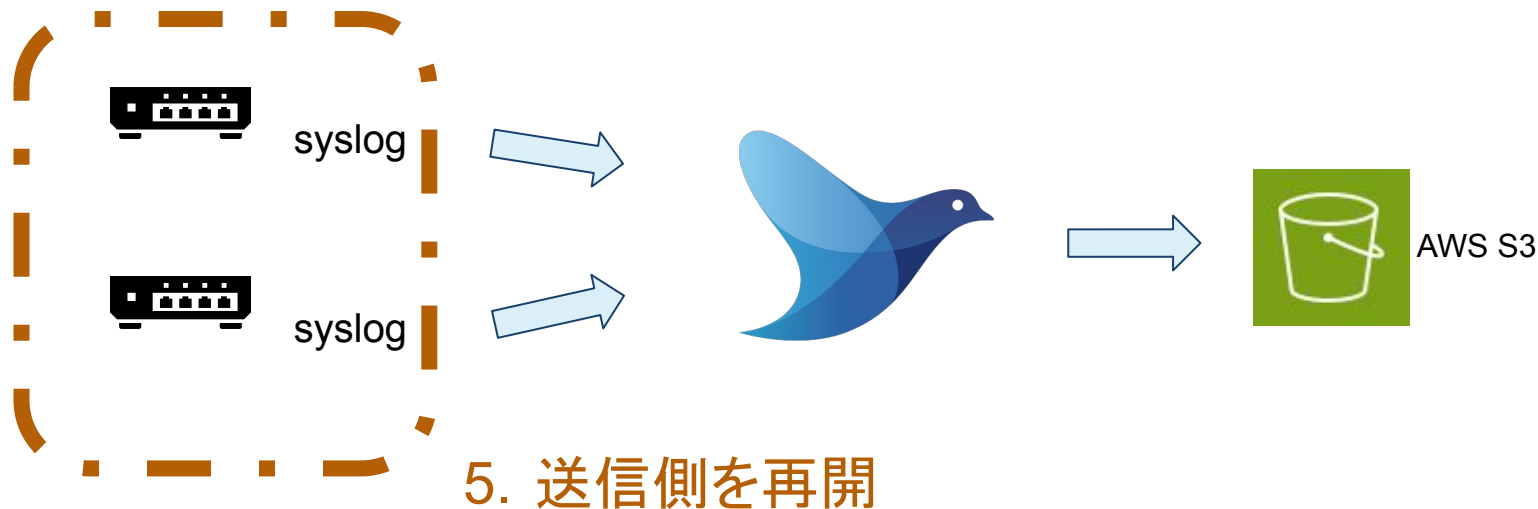
- これまで、アプリケーションのログ欠損なしで稼働中のFluentd設定ファイル更新、Fluentdアップデートが煩雑



4. Fluentd を再開

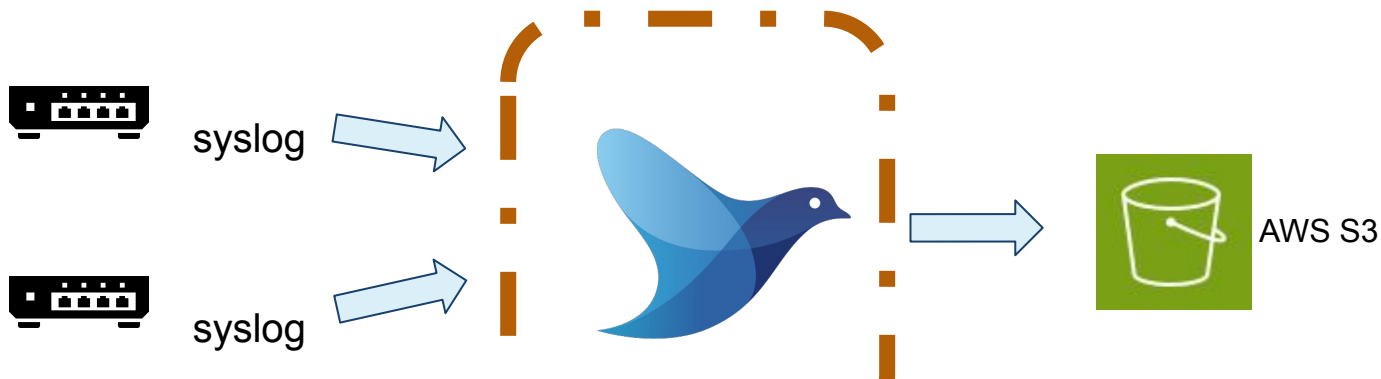
これまでの Fluentd 更新作業

- これまで、アプリケーションのログ欠損なしで稼働中のFluentd設定ファイル更新、Fluentdアップデートが煩雑



ゼロダウンタイム・リスタート

- UDP/TCP通信を止めずにFluentdをリスタート可能に
 - 設定リロードによる受信取りこぼしを気にせずに済む！
- ゼロダウンタイムでの Fluentd アップデートもサポート

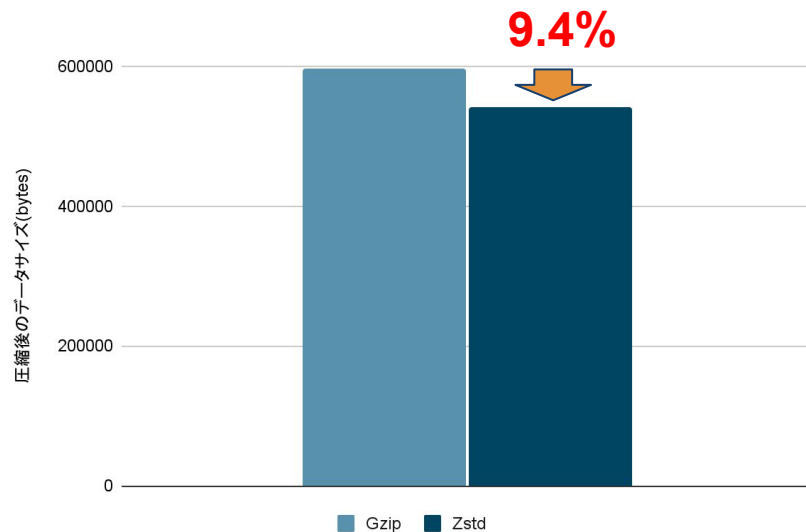
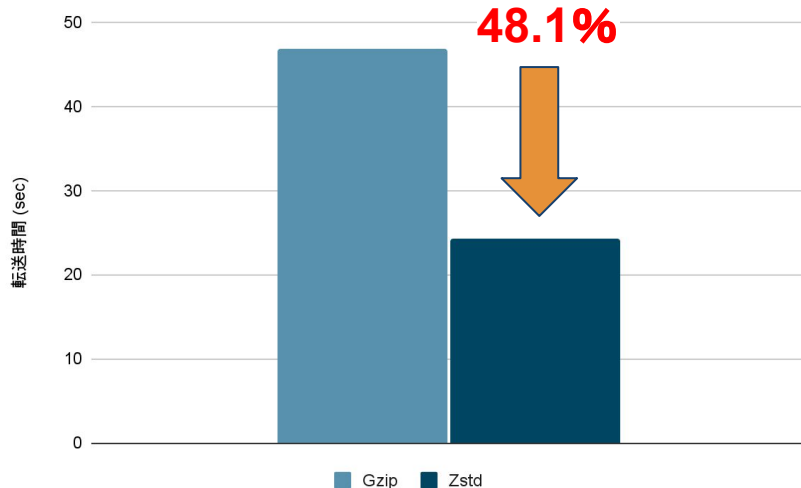


1. ゼロダウンタイム機能で更新、以上。

- リカバリーの簡易化
 - バッファファイル破損の検出機能の強化
- メトリクスの強化
 - input プラグインのメトリクスをデフォルトで取得
 - メトリクスを新規に追加

パフォーマンス改善(1)

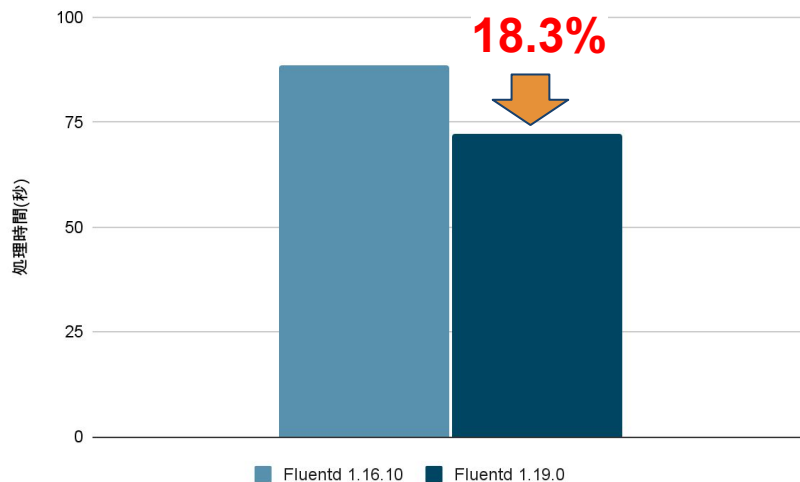
- Zstandard (zstd)圧縮形式をサポート
- 低負荷で高効率な圧縮



(ランダムな 10 GBのデータを読み込み、それぞれの形式で圧縮し転送)

パフォーマンス改善(2)

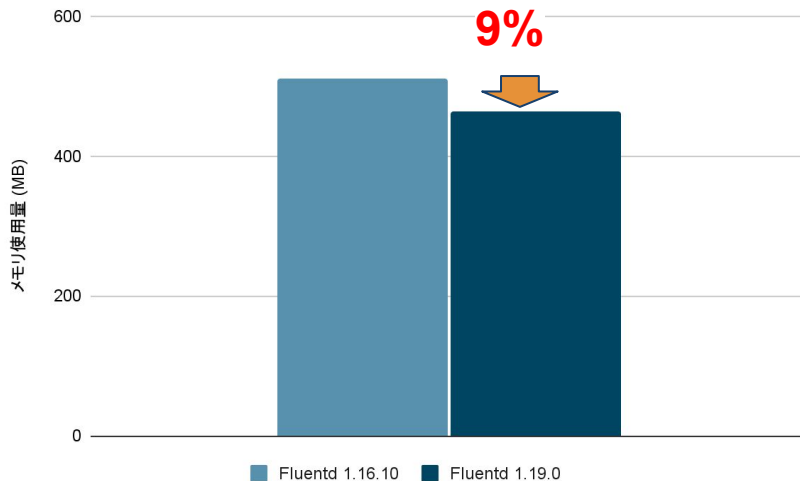
- ログファイルを読み込み、集約した結果をファイル出力する途中処理のJSONパフォーマンス改善



(ランダムな 10 GBのデータを読み込み出力)

メモリ使用量の改善

- ログファイルを読み込み、集約した結果をファイル出力する際のメモリ使用量を改善



(1 GBのログファイル読込・出力)

- Fluentdとは
- FluentdとCloud Native
- Fluentdの進化
- 最新版紹介
- OpenTelemetry対応

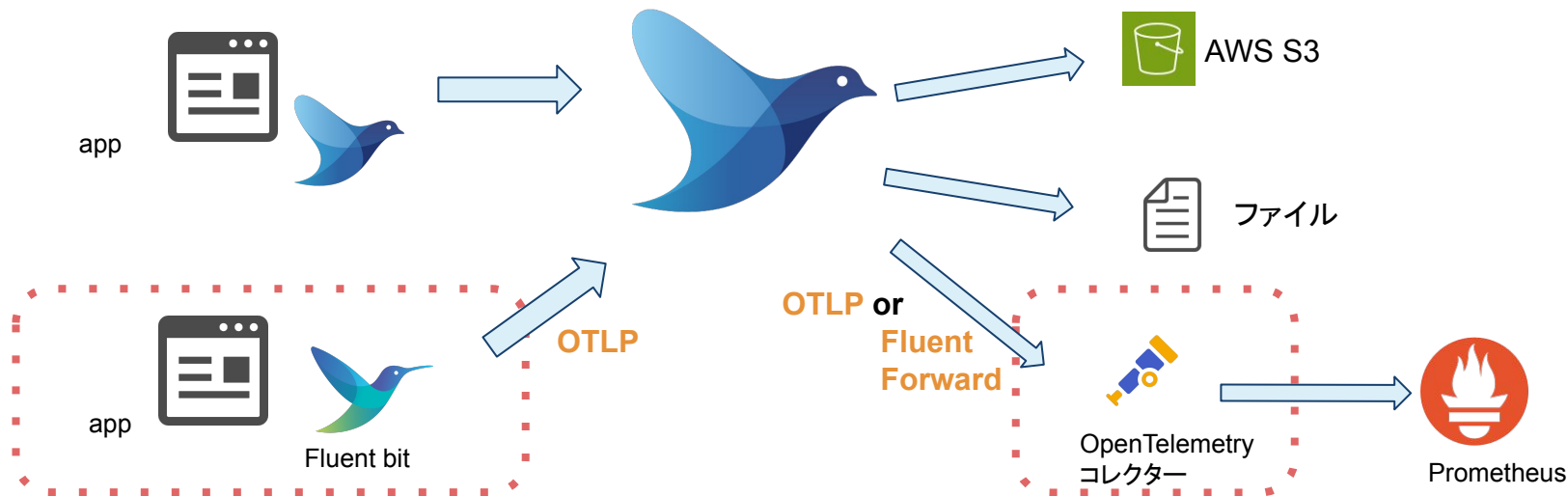
- CNCF主導プロジェクト
- ログ・メトリクス・トレース統合の標準仕様
 - ベンダーに依存しないオブザーバビリティの共通言語
 - ログ: テキストや構造化イベント
 - メトリクス: リクエスト数、レイテンシなど
 - トレース: リクエストや処理の経路情報

- ✓ アプリケーションのテレメトリデータを転送可能
 - Fluentdでテレメトリデータを受信
 - OpenTelemetry Collector へ OTLP 形式で転送
- ✓ FluentdのメトリクスデータをOTLP 形式で送信可能
- ⚠ OTLP 形式でないテキストのログデータの転送は今後対応予定
 - Fluent Forward 形式でログを Collector に転送可能

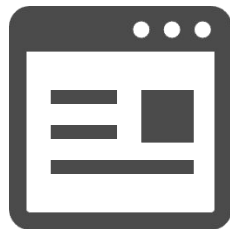
既存のログ収集システムをベースに...



OTLP形式のデータ経路を、既存のシステムに追加



設定例: OTLPデータの受信



アプリケーションなど



```
<source>
  @type opentelemetry
  tag opentelemetry
<http>
  bind 0.0.0.0
  port 4318
</http>
</source>
```

設定例: OTLPデータの送信

```
<match tag>
  @type opentelemetry
  <http>
    endpoint "http://collector:4318"
  </http>
```

```
...
</match>
```



OpenTelemetryコレクター

```
receivers:
  otlp:
    protocols:
      http:
        endpoint: 0.0.0.0:4318

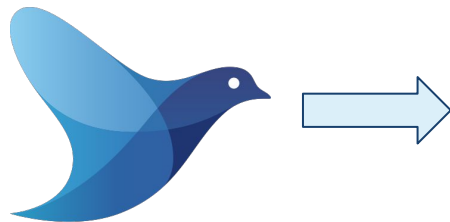
service:
  pipelines:
    traces:
      receivers: [otlp]
    metrics:
      receivers: [otlp]
    logs:
      receivers: [otlp]
```

設定例: ログ送信

```
<match tag>
  @type forward
```

```
<server>
  host collector
  port 24224
</server>
```

```
...
</match>
```



OpenTelemetryコレクター

```
receivers:
  fluentforward:
    endpoint: 0.0.0.0:24224

service:
  pipelines:
    logs:
      receivers: [fluentforward]
```

OTLP 形式でないテキストのログデータの転送は今後対応予定
Fluent Forward 形式でログを Collector に転送可能

- Fluentdは今も進化を続けています
 - ゼロダウンタイムで運用を止めずに更新
 - 耐障害性の向上で安定稼働
 - パフォーマンス改善で高速化
 - OpenTelemetry対応

Discussionに日本語Q&Aあります！

<https://github.com/fluent/fluentd/discussions/categories/q-a-japanese>

日本語での問い合わせも、
お待ちしております！！

Issue報告やPull Requestもお待ちしております！