OpenSearchCon EUROPE

Fluent Bit & OpenSearch

Patrick Stephens, Fluent Bit maintainer

$ whoami

- 23+ years experience in software engineering
- Primarily in defence as an infra/platform engineer
- More recently containers, K8S and cloud native
- OSS maintainer for Fluent Bit since 2021
- Big companies to successful startups
- Co-founder observability consultancy telemetryforge.io

pat@telemetryforge.io

linkedin.com/in/patrickjkstephens

github.com/patrick-stephens

TelemetryForge

telemetryforge.io

Goals of this talk

Explain, explore and encourage use of Fluent Bit

Demonstrate native Opensearch integration

Talk about myself

- Come find me and chat - I’m terrified!

telemetryforge.io

fluentbit

fluentbit.io

TLDR; Fluent Bit

A lightweight, high-performance Telemetry Processor and Forwarder.

Collects telemetry data from diverse sources, transforms them, and routes them to multiple destinations (e.g., OpenSearch, CloudWatch, Kafka).

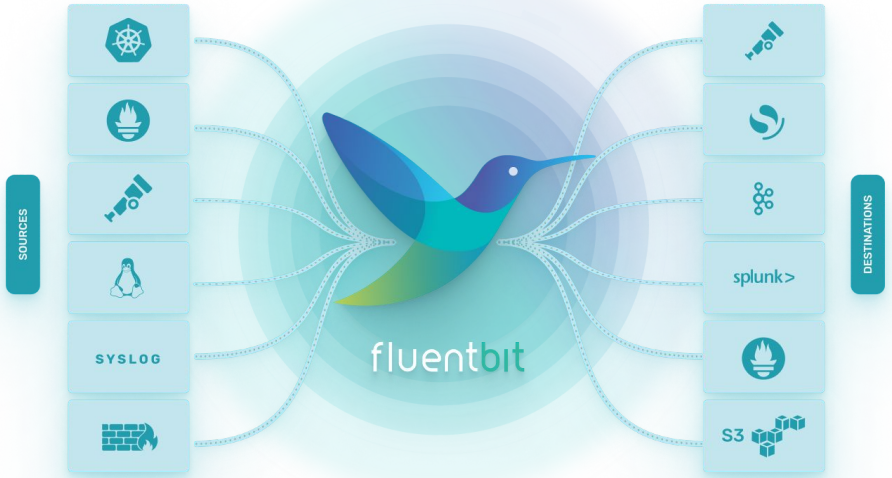
- **Ultra-Low Resource Footprint**: Optimised for CPU and memory efficiency.
- **Cloud-Native and Edge**: Ideal for Kubernetes, containerised environments, and Edge computing.
- **Flexible Pipelines**: Supports filtering, parsing, and masking data on the fly.
- **Vendor Agnostic**: Same pipeline, different outputs

Solves the "***Log Gravity***" problem: efficiently moving data from short-lived containers to persistent storage without exhausting host resources.

telemetryforge.io

Telemetry data is everywhere

Collect once, route anywhere!

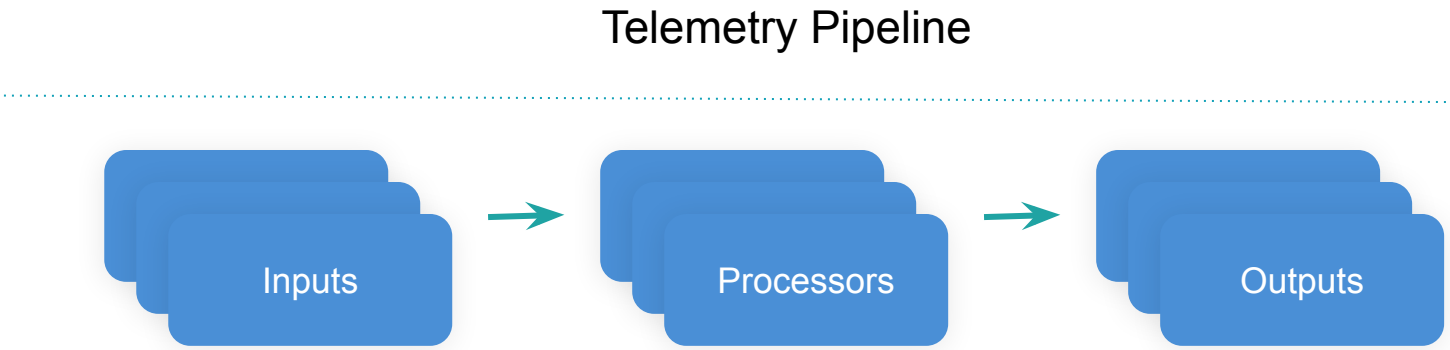


TelemetryForge

telemetryforge.io

How does Fluent Bit work?

Collect once, route anywhere!

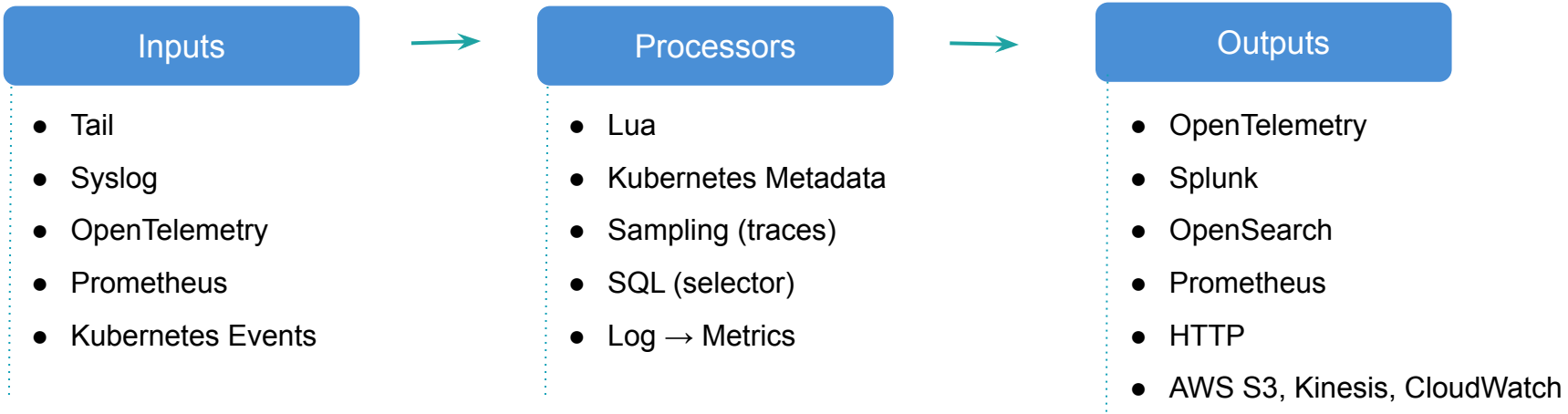


TelemetryForge

telemetryforge.io

Plugin Architecture

Extensible through a rich plugin ecosystem



TelemetryForge

telemetryforge.io

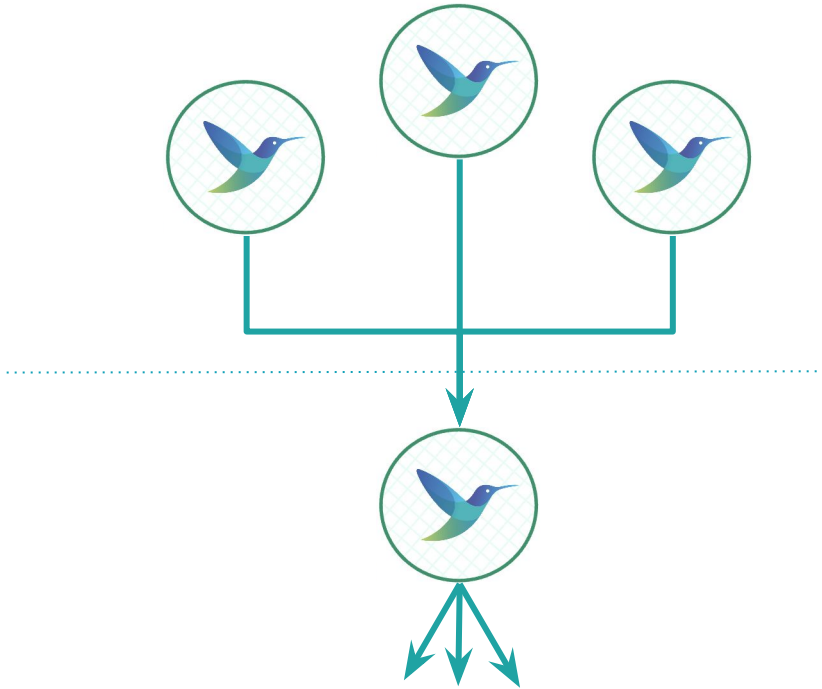
How to use Fluent Bit?

Collector (“Edge”)

- Embedded within the application environment (e.g., as a DaemonSet).
- **Focus**: Low overhead, immediate parsing, and forwarding.

Aggregator (“Hub”)

- Sits between Collectors and the backend (e.g., OpenSearch, S3).
- **Focus**: Load balancing, backpressure management, and data enrichment.



TelemetryForge

telemetryforge.io

What drives Fluent Bit?

Core Design Principles

High Performance & Low Resource Usage

- Minimal CPU and memory footprint
- Optimized for speed and efficiency

Broad Ecosystem Support

- Rich plugin ecosystem
- Runs across cloud, on-prem, and edge

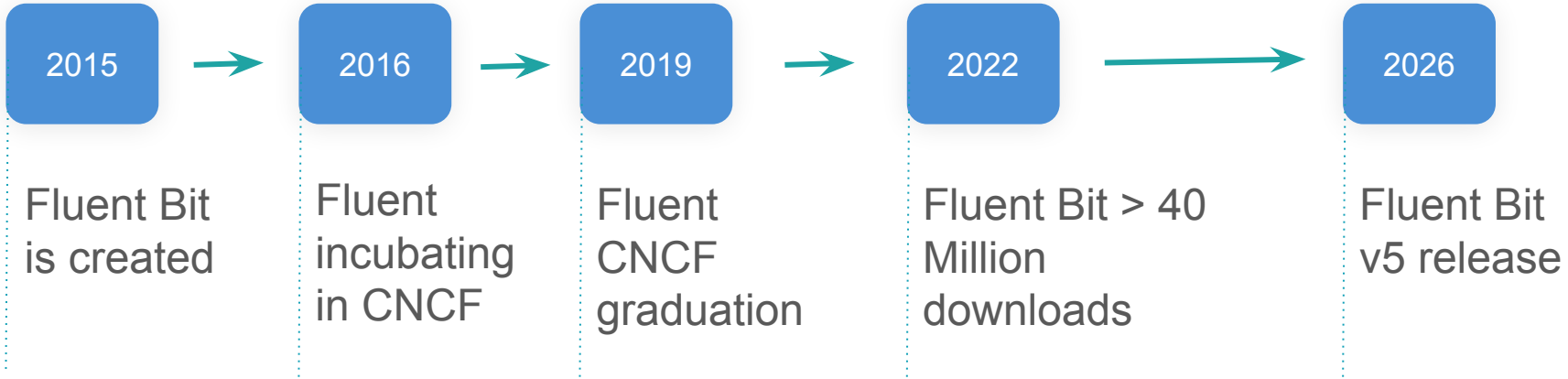
Vendor Neutral & Flexible Integration

- Open Source & Vendor Agnostic
- Integrates with Prometheus, OpenTelemetry, OpenSearch and more!

telemetryforge.io

Maturity

Over a decade of battle testing



TelemetryForge

telemetryforge.io

Fluent Bit & OpenSearch

First-class native integration with Opensearch

Contribution by AWS and Opensearch maintainers

```
pipeline:

  outputs:
    - name: opensearch
      match: '*'
      host: vpc-test-domain-ke7thhzoo7jawsrhmm6mb7ite7y.us-west-2.es.amazonaws.com
      port: 443
      index: my_index
      type: my_type
      aws_auth: on
      aws_region: us-west-2
      tls: on
```

telemetryforge.io

Demo time

https://o11y-workshops.gitlab.io/workshop-fluentbit/#/11

Demo details

Local container deployment (compose/podman stack)

- Deploy OpenSearch & OpenSearch Dashboard
- Deploy Fluent Bit

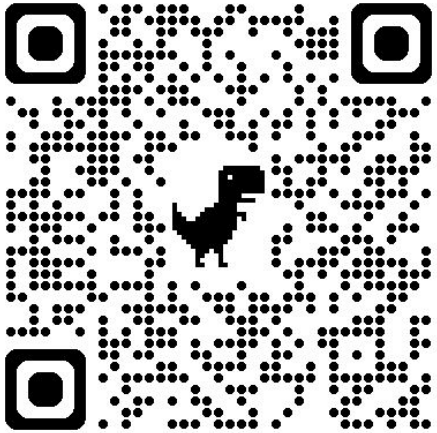
Fluent Bit sends data to OpenSearch

- And to terminal

Visualise live telemetry with OpenSearch Dashboard

Github repository: https://github.com/patrick-stephens/opensearch-install-demo

telemetryforge.io



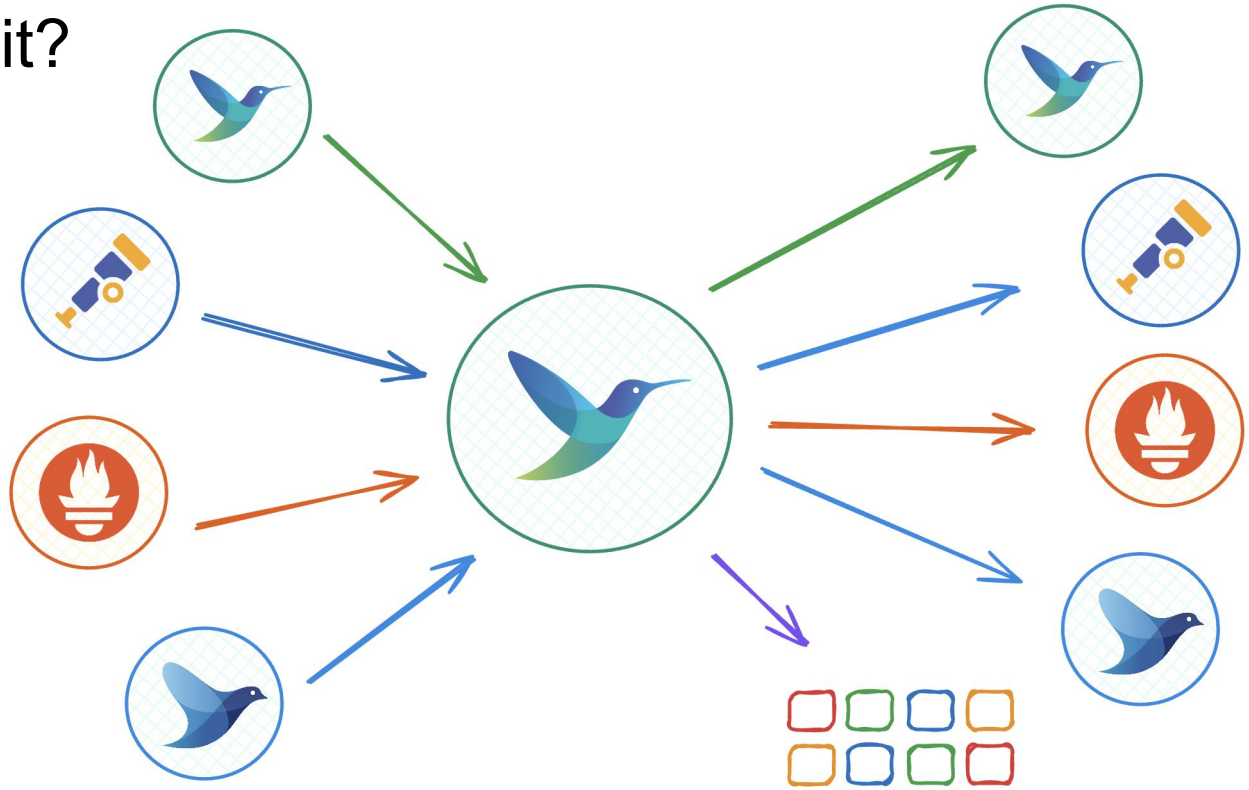
Questions?

info@telemetryforge.io

TelemetryForge

Why use Fluent Bit?

- Performance
- Maturity
- Vendor agnostic
- Open source



telemetryforge.io

TelemetryForge

Resources

Demo:

- o11y-workshops.gitlab.io/workshop-fluentbit
- github.com/patrick-stephens/opensearch-install-demo

Fluent Bit documentation: docs.fluentbit.io

- Opensearch output: docs.fluentbit.io/manual/data-pipeline/outputs/opensearch

Fluent Bit Slack: slack.fluentd.org/

telemetryforge.io