CLOUD NATIVE
COMPUTING FOUNDATION

BLOG / MEMBER POST

Logstash, Fluentd, Fluent Bit, or Vector? How to choose the right open-source log collector

By Stela Udovicic

February 10, 2022

*Guest post originally published on **ERA Software's blog** by Stela Udovicic*

In this blog, we'll discuss the most popular log collectors, including Logstash, Fluentd, Fluent Bit, and Vector.

Logstash, Fluentd, Fluent Bit, and Vector logos

Whether you already use an open-source log collector or are about to choose one or more for your environment, it's important to understand the key requirements of a log collector critical for your day-to-day operations. These requirements include high data throughput, reliability, scalability, flexibility, security, and resource consumption (CPU, memory). In this blog, we'll discuss the most popular log collectors, including Logstash, Fluentd, Fluent Bit, and Vector.

Key requirements to consider when evaluating log collectors

Before diving into specific open-source log collector implementations, here are important requirements to consider when evaluating log collectors.

High data throughput

To debug issues successfully, engineering teams need a high count of logs per second and low-latency log processing. To avoid business-disrupting outages or failures, engineers need to get critical log data quickly, and this is where log collectors with high data throughput are preferable.

Reliability

Log collectors should ensure the high integrity of processed data. Even under increased data throughput, the integrity of data should be preserved. The frequency and amount of data loss by log collectors should be bounded and ideally avoided under any conditions.

Scalability

There are several strategies that enable log collectors to handle large volumes of data. Filtering the logs that aren't a high priority, parsing, or compressing complex logs are just a few of them. Also, it's important to consider that such data processing may incur increased CPU and memory resource utilization as the price to pay for the capability to scale. Also, when the data rate increases, there is a higher resource consumption and possible backpressure which also affects the scaling capability of log collectors.

Handling a variety of data formats

Logs come in many different formats from various elements of cloud applications and infrastructure. Containers and microservices are developed in different programming languages or frameworks, with varying approaches to logging formats. Avoiding the need to use more than one log collector to handle the number of formats reduces the overall complexity.

Support for various data sources and destinations

Log collectors source data from a variety of cloud environments. The log data sources may also include message queueing and streaming platforms such as Kafka, Redis, and RabbitMQ. The log collectors send the data to different destinations, such as log management tools and storage archives. The ability of log collectors to handle various sources and destinations adds to their flexibility and usability.

Security

The capacity to handle sensitive information, such as anonymizing or excluding confidential fields and sending logs to storage backends in a secure manner, should be considered when assessing any log collector.

Now, let’s discuss specific open-source collectors and their key characteristics.

Logstash

Logstash is among the most popular log collectors, and it’s a part of the ELK (Elasticsearch, Logstash, Kibana) stack.

Key Logstash strengths:

- Handles structured and unstructured data.

- Supports increased data security with the ability to anonymize or exclude sensitive fields.

- Supports hundreds of plugins which include input, filter, and output plugins. The filter plugins perform log processing such as aggregation and parsing.

Although Logstash is a reliable log collector with many options for processing log data, other log collectors described in this blog may be better if a small memory footprint is a critical requirement. Because Logstash is written in Java, it requires JVM support. If you’re planning to collect logs from embedded devices and IoT applications, it‘s not the best choice.

Fluentd

Fluentd is a log collector with a small memory footprint that handles various log sources and destinations. Many supported plugins allow connections to multiple types of sources and destinations. As with the other log collectors, the typical sources for Fluentd include applications, infrastructure, and message-queueing platforms, while the usual destinations are log management tools and storage archives.

Key Fluentd advantages:

- Supports many log sources and destinations
- Flexible and extensible parsing options, supporting a wide array of input formats
- Has a large ecosystem surrounding it, including hundreds of plugins plus the ability to write your own in Ruby
- Supports the Apache license, version 2.0
- Vendor neutrality (a CNCF project)

Fluentd is a good choice if you're looking for vendor neutrality. It's also very popular to use with Kubernetes and containerized environments.

Fluent Bit

Fluent Bit works well in containerized environments such as Kubernetes clusters. Also, Fluent Bit can scale and still conserve resources because it has a small footprint. Although Fluent Bit is frequently used in Kubernetes environments, it can also be deployed on bare-metal servers, virtual machines, and embedded devices.

Key Fluent Bit advantages:

- Lightweight design with minimal memory footprint (typically less than 1MB)
- Easy-to-scale architecture
- A pluggable architecture with a number of input, filter, and output plugins
- Supports metric-based as well as log-based payloads
- Supports sending logs to storage backends over a secure connection
- Support for stream processing using SQL
- Supports the Apache license, version 2.0
- Vendor neutrality (a CNCF project)

Fluent Bit collects logs and metrics from various sources and sends them to different destinations as do other log collectors. Where Fluent Bit really shines is in embedded, edge, and other resource-constrained environments where a lean runtime paired with a wide array of input/output options is critical. Fluent Bit is not only a log collector but can also be used as a stream processor as well as a shipper for forwarding log data to Fluentd.

Vector

Vector is designed to be a high-performance log collector. It's a relatively new product compared to other log collectors discussed in this blog.

Key Vector advantages:

- Efficient memory/CPU consumption and high data throughput
- Good reliability with correctness and delivery guarantees
- Includes custom DSL for transforming data on the fly in a safe and performant way
- Support for metrics-based as well as log-based payloads
- Large number of input and output integrations
- Can be deployed as an agent or aggregator

Vector is a great, flexible choice due to its wide array of deployment options, support for both metrics and logs, as well as the number integrations available. Also, because Vector is written in Rust, it provides memory safety and efficiency guarantees that make it unique among the other incumbents. Whether you're working with a new or old environment, it's certainly worth a close look.

Vector introduces a unit test framework that makes it easier to maintain complex log collector topologies. Also, Vector's software components attempt to provide delivery guarantees for logs and events delivered to destinations. Guarantees for the general code stability of Vector's components are also available to Vector's users.

Summary

It's hard to find a single collector that dominates the space. Choosing the right log collector depends on your specific needs and requirements. For instance, if you're looking for log collectors for IoT applications that require small resource consumption, then you're better off with Vector or Fluent Bit rather than Logstash. If you're looking for vendor neutrality, CNCF-backed projects such as Fluentd and Fluent Bit are good choices.

Careful examination of performance, resource consumption, flexibility to support a variety of input and output formats, scalability, reliability, vendor lock-in, and security requirements can help you find the log collector that work for you.

Luckily, no matter which log collector you choose, it's possible to have the ability to search and analyze logs at an affordable cost and in real time.

To learn how EraSearch integrates with Fluentd, read the **Connecting Fluentd to EraSearch** blog. Learn about EraSearch and Vector integrations by reading the following blogs: **Shipping Kubernetes Logs to EraSearch Using Vector** and **Collect All Cloudflare Logs Cost-Effectively**.

SHARE

Other posts to check out

Cloud Native Computing Foundation Announces Cilium Graduation

Kubernetes secure development best practices in action
BY SAYLOR BERMAN + CIARA STACKE
CLOUD NATIVE
COMPUTING FOUNDATION

Kubernetes secure development best practices in action

October 11, 2023

October 11, 2023

Exploring Kepler's potentials: unveiling cloud application power consumption

BY MARCELO AMARAL, SUNYANAN CHOOCHOTKAEW, EUN KYUNG LEE, HUAMIN CHEN, AND TAMAR EILAM

CLOUD NATIVE COMPUTING FOUNDATION

Exploring Kepler’s potentials: unveiling cloud application power consumption

October 11, 2023

Subscribe for updates, event info, webinars, and the latest community news

First name*

Last name*

Email address*

Select language*

Subscribe

JOIN NOW

All CNCF Sites

Accessibility Statement

Submit an issue with this page