Event Aggregation Techniques (Logstash Vs Fluentd)

Amit Sengupta (Independent Research)

Email – amits2913@gmail.com

Independent Researcher

Abstract

Event Aggregators are the unsung heroes behind every smart analytics. They are the hard-working daemons that run on servers to pull telemetry data and transport them to AI/ML engines to enable enterprise Event-Driven capabilities. While visualization tools like Kibana or Redash or Grafana bask in the glory, data collectors' s routing makes, it all possible. Here, we will put the two of the most popular data collectors in the open-source world.

When it comes to collecting and shipping logs to Elastic stack, we usually hear about ELK — Elastic, Logstash, and Kibana. It has almost become a synonym for Elastic stack. While Fluentd is gaining popularity as far as logging for microservices (in Docker /Kubernetes environment) is concerned. This is due to the fact that Fluentd is built by Treasure Data and is part of CNCF. So Fluentd has much better integration with CNCF hosted projects like Kubernetes, Prometheus, OpenTracing, etc.

Recently, I got a chance to evaluate these two log collectors for an initiative, I was working on, and it gave me an opportunity to deep dive and look at the pros and cons for each of them. The purpose of this comparison is not to choose a winner but to find an appropriate use case fitment.

Keywords: - *Logstash, Fluentd, Data Aggregation, Event Correlation, Single Pane of Glass Observability, Operational Optimization, Open Telemetry*

Logstash Vs Fluentd - Capability Based Comparison: -

Category	Logstash	FluentD
Platform	Linux and Windows	Linux and Windows
Code Language	Jruby - Requires JRE in host machine	Cruby - No Runtime environment required
Event Routing	Algorithmic Statements	Tags
Plugin Ecosystem	Centralized	Decentalized
Transport	Deploy with Redis for Reliability	Built in reliability but tedious to configure
Performance	Uses more memory. Use Elastic beats for Leafs	Uses less memory. Use Fluent Bit and FluentD Forwarder for Leafs
Plugins	About 200 Plugins. All of them are availaible in official Git Repo	About 500 Plugins, however no centralized Repo for plugins.

Logstash Vs Fluentd — Use Case-Based Comparison: -

1. **Data Collection**

Fluentd — Docker has a built-in logging driver for Fluentd. This means no additional agent is required on the container to push logs to Fluentd. Logs are directly shipped to Fluentd service from STDOUT and no additional logs file or persistent storage is required.

Example- Add below section to the service in a docker compose file and you are done -

logging: driver: "fluentd" options: fluentd-address: <fluentd IP>:<fluentd service port> tag: PilotAppID.Logs

Logstash — The application logs from STDOUT are logged in docker logs and written to file. The logs from the file then have to be read through a plugin such as file beats or elastic beats and sent to Logstash.

2. **Data Parsing**

Fluentd has standard built-in parsers such as json, regex, csv, syslog, apache, nginx etc as well as third-party parsers like grok to parse the logs.

Logstash has out-of-the-box plugins for filtering and parsing like aggregate, geoip etc in addition to the standard formats.

3. **Metric Data Collection**

Fluentd doesn't have the out-of-the-box capability to collect system/container metrics. It can however scrape metrics from a Prometheus exporter.

Logstash uses **Metricbeat** which has the inherent capability to collect system/container metrics and forward them to Logstash. Additionally, Logstash can also scrape metrics from Prometheus exporter.

4. **Scraping**

Fluentd has a plugin **http_pull** which provides the capability to pull data from HTTP endpoints like metrics, health checks, etc.

Logstash has **http plugin** (supported by elastic) which provides the capability to pull data from HTTP endpoints.

Logstash Vs Fluentd — Architecture Fitment: -

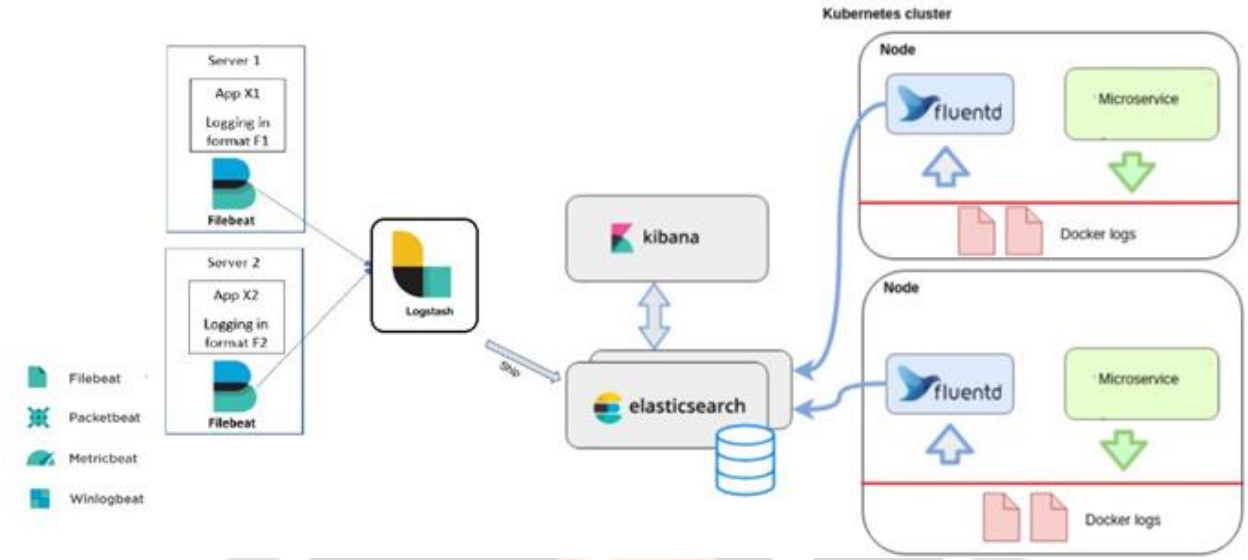
Looking at the above use cases, it should be clear that both Fluentd and Logstash are suitable for certain requirements. The best part is both can co-exist in the same environment and can be used for specific use cases.

For monolithic applications on traditional VMs, Logstash looks like a clear choice and way to proceed as it supports multiple agents for the collection of logs, metrics, health, etc.

For microservices hosted on Docker/Kubernetes/OCP, Fluentd looks like a great choice considering a built-in logging driver and seamless integration. It supports all commonly used parsers like json, nginx, grok etc.

While there are several differences, the similarities between Logstash and Fluentd are greater than their differences. In a **hybrid environment**, both can coexist and support their use cases.

ELK-EFK Hybrid Reference Architecture



Reference:

1. https://docs.docker.com/config/containers/logging/fluentd/
2. https://docs.docker.com/config/containers/logging/fluentd/
3. https://www.elastic.co/guide/en/logstash/current/filter-plugins.html
4. https://stackshare.io/stackups/fluentd-vs-graylog-vs-logstash
5. https://docs.dapr.io/operations/observability/logging/fluentd/