Logstash

Logstash is an open-source, centralized, events and logging manager. It is a part of the ELK (ElasticSearch, Logstash, Kibana) stack.
expected to have a basic understanding of Ruby, JSON, and web technologies.
Logstash is a tool based on the filter/pipes patterns for gathering, processing and generating the logs or events. It helps in centralizing and making real time analysis of logs and events from different sources.

Logstash is written in the JRuby programming language that runs on the JVM, hence you can run Logstash on different platforms. It **collects different types of data like Logs, Packets, Events, Transactions, Timestamp Data, etc**., from almost every type of source. The data source can be Social data, E-commerce, News articles, CRM, Game data, Web trends, Financial data, Internet of Things, Mobile devices, etc.

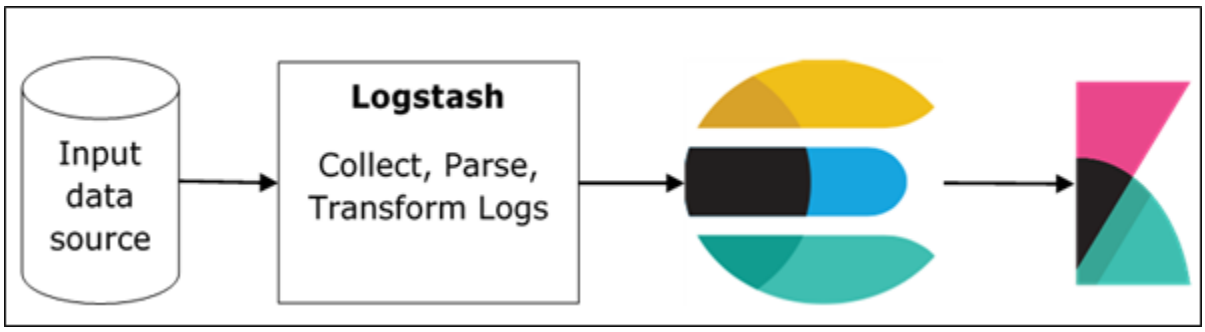
Logstash General Features

- Logstash can collect data from different sources and send to multiple destinations.
- Logstash can handle all types of logging data like Apache Logs, Windows Event Logs, Data over Network Protocols, Data from Standard Input and many more.
- Logstash can also handle http requests and response data.
- Logstash provides a variety of filters, which helps the user to find more meaning in the data by parsing and transforming it.
- Logstash can also be used for handling sensor data in iot.
- Logstash is open source and available under the Apache license version 2.0.

https://www.tutorialspoint.com/logstash/logstash_introduction.htm

ELK stands for Elasticsearch, Logstash, and Kibana. In the ELK stack, Logstash extracts the logging data or other events from different input sources. It processes the events

and later stores them in Elasticsearch. Kibana is a web interface, which accesses the logging data from Elasticsearch and visualizes it.



Logstash and Kibana

Kibana does not interact with Logstash directly but through a data source, which is Elasticsearch in the ELK stack. Logstash collects the data from every source and Elasticsearch analyzes it at a very fast speed, then Kibana provides the actionable insights on that data.

Kibana is a web based visualization tool, which helps developers and others to analyze the variations in large amounts of events collected by Logstash in the Elasticsearch engine. This visualization makes it easy to predict or to see the changes in trends of errors or other significant events of the input source.

Logstash

Some of the important features of logstash are as follows:

1. Pluggable data pipeline architecture
2. Extensibility
3. Centralized data processing
4. Variety and volume
5. Compatible

Installation of Logstash

Prerequisites

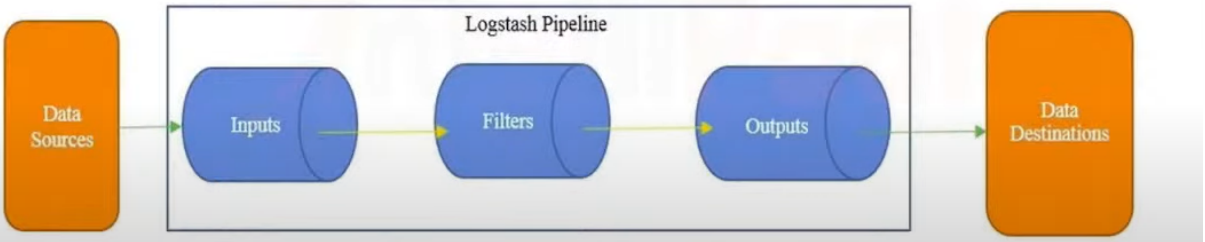
Java runtime is required to run Logstash. Logstash requires Java 8. Make sure that **JAVA_HOME** is set as an environment variable, and to check your Java version, run the following command:

java -version

You should expect java version output.

The Logstash architecture

The Logstash event processing pipeline has three stages, that is, Inputs, Filters, and Outputs. A Logstash pipeline has two required elements, that is, input and output, and one option element known as filters:



The Logstash architecture

Inputs create events, Filters modify the input events, and Outputs ship them to the destination.

Inputs and outputs support codecs, which allow you to encode or decode the data as and when it enters or exits the pipeline, without having to use a separate filter.

Logstash uses in-memory bounded queues between pipeline stages by default (Input to Filter and Filter to Output) to buffer events.

If Logstash terminates unsafely, any events that are stored in memory will be lost. To prevent data loss, you can enable Logstash to persist in-flight events to the disk by making use of persistent queues.

Adding new fields
Removing existing unnecessary fields

LOGSTASH FILTERS

list of Logstash filters with a very brief description of each:

1. **Grok Filter**: Parse unstructured log data using predefined patterns or custom regular expressions and extract fields.

2. **Date Filter**: Parse timestamps from log data and convert them into a standardized format.

3. **JSON Filter**: Parse JSON-formatted log messages and extract specific fields from the JSON data.

4. **Mutate Filter**: Perform various operations on fields, such as renaming, removing, converting data types, or adding new fields.

5. **KV Filter**: Parse key-value pairs from log data and add them as separate fields.

6. **GeoIP Filter**: Enrich log data with geolocation information based on IP addresses.

7. **UserAgent Filter**: Extract browser and device information from user agent strings.

8. **Drop Filter**: Exclude events from further processing based on certain conditions.

9. **Conditional Filter**: Apply filters conditionally based on specific criteria.

10. **Translate Filter**: Enrich log data by adding fields based on predefined lookup tables.

11. **Fingerprint Filter**: Generate a unique hash for log events based on selected fields.

12. **Clone Filter**: Duplicate log events to send them to multiple outputs.

13. **URLDecode Filter**: Decode URL-encoded fields in log data.

14. **Prune Filter**: Remove fields from log events based on specified patterns.

15. **CSV Filter**: Parse log data in CSV format and extract fields.

16. **XML Filter**: Parse log data in XML format and extract fields.

17. **UUID Filter**: Generate UUIDs and add them as fields to log events.

18. **Aggregate Filter**: Combine multiple log events into a single event based on specified criteria.

These filters are just some of the many available in Logstash, and they can be combined and configured in various ways to handle specific data processing needs for your log data.

Parsing: Logstash filters can use patterns and regular expressions to parse log messages and extract specific fields, timestamps, and other relevant data. This is often done using the Grok filter.

Timestamp Extraction: Logstash filters can extract timestamps from log messages and convert them into a standardized format for easier analysis.

Enrichment: Logstash filters can enrich log data by adding additional fields, data, or metadata from external sources. For example, IP geolocation, user agent information, or lookup tables can be used for enrichment.

Conditional Processing: Logstash filters can apply different processing steps based on conditions. For example, different filters can be applied to log messages of different types or coming from specific sources.

JSON Parsing: Logstash filters can parse JSON-formatted log messages and extract specific fields from the JSON data.

Data Transformation: Logstash filters can transform data by renaming fields, converting data types, removing unwanted fields, or combining multiple fields into one.

Handling Multiline Logs: Logstash filters can handle multiline log entries and ensure that they are processed as single events.

Data Validation: Logstash filters can perform data validation to ensure that log messages meet certain criteria or match expected patterns.

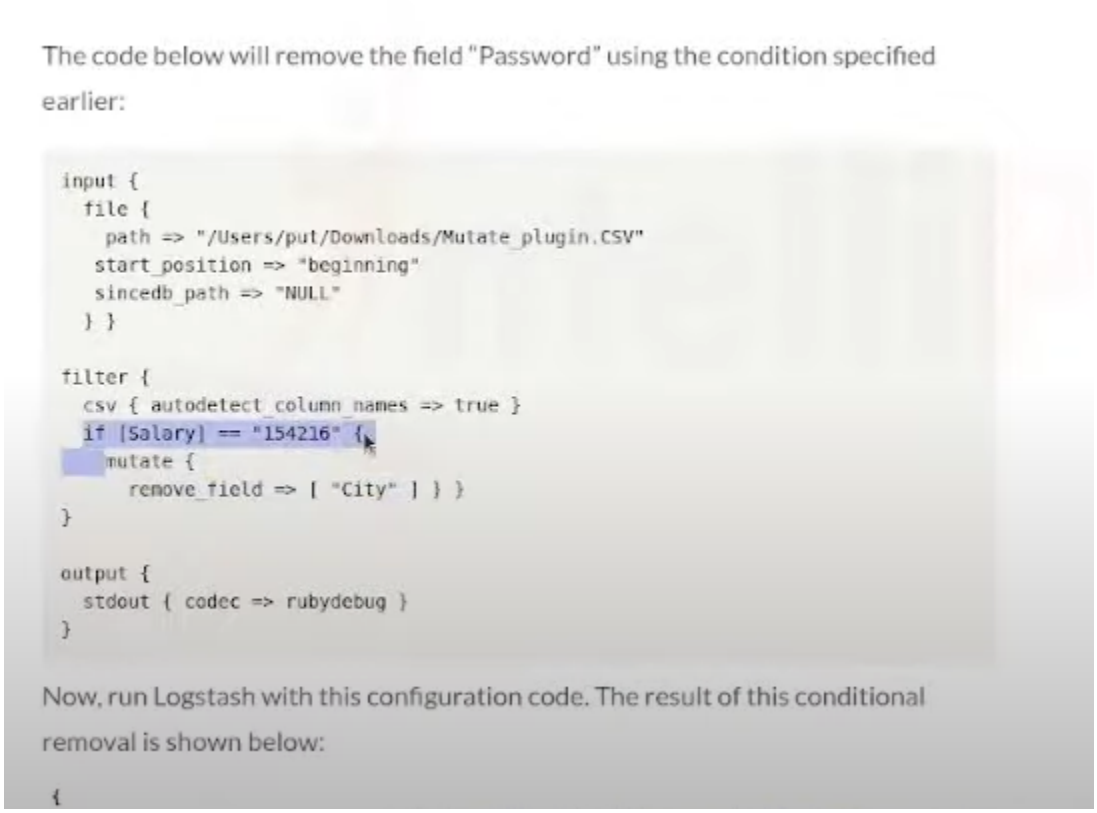
Deduplication: Logstash filters can identify and remove duplicate log entries from the data.

Conditional Filtering: Logstash filters can apply conditional filters to exclude or include certain log messages based on specific conditions.

Merge
Copy

Join
Strip
multiple filters can be performed

Conditional based mutate filter allows to remove fields based on the condition



grok	**Parses unstructured event data into fields**

kv	**Parses key-value pairs**

split	**Splits multi-line messages, strings, or arrays into distinct events**

truncate	**Truncates fields longer than a given length**

Mutate : The mutate filter allows you to perform general mutations on fields. You can rename, replace, and modify fields in your events.

GROK Filter

Regular expressions , custom patterns

https://www.elastic.co/guide/en/logstash/current/introduction.html

Logstash Introduction

github

Logstash is an open source data collection engine with real-time pipelining capabilities. Logstash can dynamically unify data from disparate sources and normalize the data into

destinations of your choice. Cleanse and democratize all your data for diverse advanced downstream analytics and visualization use cases.

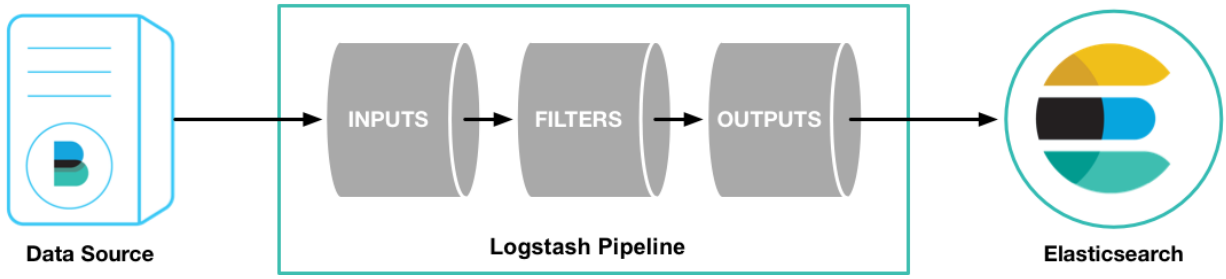
While Logstash originally drove innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process. Logstash accelerates your insights by harnessing a greater volume and variety of data.

Stashing Your First Event

edit

First, let's test your Logstash installation by running the most basic *Logstash pipeline*.

A Logstash pipeline has two required elements, `input` and `output`, and one optional element, `filter`. The input plugins consume data from a source, the filter plugins modify the data as you specify, and the output plugins write the data to a destination.



Parsing Logs with Logstash

edit

In Stashing Your First Event, you created a basic Logstash pipeline to test your Logstash setup. In the real world, a Logstash pipeline is a bit more complex: it typically has one or more input, filter, and output plugins.

In this section, you create a Logstash pipeline that uses Filebeat to take Apache web logs as input, parses those logs to create specific, named fields from the logs, and

writes the parsed data to an Elasticsearch cluster. Rather than defining the pipeline configuration at the command line, you'll define the pipeline in a config file.

To get started, go here to download the sample data set used in this example. Unpack the file.