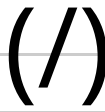


[Docs \(/guide\)](#)[Docs \(/guide\)](#)

You are looking at documentation for an older release. Not what you want? See the current release documentation ([../current/index.html](#)).

[Logstash Reference \[5.3\] \(index.html\)](#) » [How Logstash Works](#)

« [Stitching Together Multiple Input and Output](#)

[Execution Model](#) » ([execution-model.html](#))

[Plugins \(multiple-input-output-plugins.html\)](#)

How Logstash Works [edit \(https://github.com/elastic/logstash/edit/5.3/docs/static/life-of-an-event.asciidoc\)](#)

The Logstash event processing pipeline has three stages: inputs → filters → outputs. Inputs generate events, filters modify them, and outputs ship them elsewhere. Inputs and outputs support codecs that enable you to encode or decode the data as it enters or exits the pipeline without having to use a separate filter.

Inputs [edit \(https://github.com/elastic/logstash/edit/5.3/docs/static/life-of-an-event.asciidoc\)](#)

You use inputs to get data into Logstash. Some of the more commonly-used inputs are:

- **file:** reads from a file on the filesystem, much like the UNIX command `tail -0F`
- **syslog:** listens on the well-known port 514 for syslog messages and parses according to the RFC3164 format
- **redis:** reads from a redis server, using both redis channels and redis lists. Redis is often used as a "broker" in a centralized Logstash installation, which queues Logstash events from remote Logstash "shippers".
- **beats:** processes events sent by Filebeat (<https://www.elastic.co/downloads/beats/filebeat>).

For more information about the available inputs, see [Input Plugins \(input-plugins.html\)](#).

Filters [edit \(https://github.com/elastic/logstash/edit/5.3/docs/static/life-of-an-event.asciidoc\)](#)

Filters are intermediary processing devices in the Logstash pipeline. You can combine filters with conditionals to perform an action on an event if it meets certain criteria. Some useful filters include:

- **grok**: parse and structure arbitrary text. Grok is currently the best way in Logstash to parse unstructured log data into something structured and queryable. With 120 patterns built-in to Logstash, it's more than likely you'll find one that meets your needs!
- **mutate**: perform general transformations on event fields. You can rename, remove, replace, and modify fields in your events.
- **drop**: drop an event completely, for example, *debug* events.
- **clone**: make a copy of an event, possibly adding or removing fields.
- **geoip**: add information about geographical location of IP addresses (also displays amazing charts in Kibana!)

For more information about the available filters, see [Filter Plugins \(filter-plugins.html\)](#).

Outputs

[edit \(https://github.com/elastic/logstash/edit/5.3/docs/static/life-of-an-event.asciidoc\)](https://github.com/elastic/logstash/edit/5.3/docs/static/life-of-an-event.asciidoc)

Outputs are the final phase of the Logstash pipeline. An event can pass through multiple outputs, but once all output processing is complete, the event has finished its execution. Some commonly used outputs include:

- **elasticsearch**: send event data to Elasticsearch. If you're planning to save your data in an efficient, convenient, and easily queryable format... Elasticsearch is the way to go. Period. Yes, we're biased :)
- **file**: write event data to a file on disk.
- **graphite**: send event data to graphite, a popular open source tool for storing and graphing metrics. <http://graphite.readthedocs.io/en/latest/> (<http://graphite.readthedocs.io/en/latest/>)
- **statsd**: send event data to statsd, a service that "listens for statistics, like counters and timers, sent over UDP and sends aggregates to one or more pluggable backend services". If you're already using statsd, this could be useful for you!

For more information about the available outputs, see [Output Plugins \(output-plugins.html\)](#).

Codecs

[edit \(https://github.com/elastic/logstash/edit/5.3/docs/static/life-of-an-event.asciidoc\)](https://github.com/elastic/logstash/edit/5.3/docs/static/life-of-an-event.asciidoc)

Codecs are basically stream filters that can operate as part of an input or output. Codecs enable you to easily separate the transport of your messages from the serialization process. Popular codecs include `json`, `msgpack`, and `plain` (text).

- **json**: encode or decode data in the JSON format.

- **multiline:** merge multiple-line text events such as java exception and stacktrace messages into a single event.

For more information about the available codecs, see [Codec Plugins \(codec-plugins.html\)](#).

« [Stitching Together Multiple Input and Output Plugins \(multiple-input-output-plugins.html\)](#)

[Execution Model](#) » ([execution-model.html](#))

Top Videos

- [Elasticsearch Demo \(https://www.elastic.co/webinars/getting-started-elasticsearch?baymax=default&elektra=docs&storm=top-video\)](https://www.elastic.co/webinars/getting-started-elasticsearch?baymax=default&elektra=docs&storm=top-video)
- [Kibana 101 \(https://www.elastic.co/webinars/getting-started-kibana?baymax=default&elektra=docs&storm=top-video\)](https://www.elastic.co/webinars/getting-started-kibana?baymax=default&elektra=docs&storm=top-video)
- [Logstash Primer \(https://www.elastic.co/webinars/getting-started-logstash?baymax=default&elektra=docs&storm=top-video\)](https://www.elastic.co/webinars/getting-started-logstash?baymax=default&elektra=docs&storm=top-video)

On this page

[Inputs](#)

[Filters](#)

[Outputs](#)

[Codecs](#)

Logstash Reference: 5.3	▼
Logstash Introduction (introduction.html)	
Getting Started with Logstash (getting-started-with-logstash.html)	
How Logstash Works (pipeline.html)	
Execution Model (execution-model.html)	
Setting Up and Running Logstash (setup-logstash.html)	
Breaking changes (breaking-changes.html)	
Upgrading Logstash (upgrading-logstash.html)	
Configuring Logstash (configuration.html)	
Working with Filebeat Modules (filebeat-modules.html)	
Deploying and Scaling Logstash (deploying-and-scaling.html)	