Hello World Template version: 1.0

[[doc-title]]

[[doc-type]]

Version: [[doc-version]]
Release Date: [[doc-release-date]]

1 Filebeat

1.1 Filebeat

Filebeat is a lightweight shipper for forwarding and centralizing log data. Filebeat monitors the log files or locations that you specify, collects log events, and forwards them either to Elasticsearch or Logstash for indexing.

1.2 1.0 Installing filebeat

To download and install Filebeat, use the commands that work with your system.

```
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.5.0-amd64.deb
sudo dpkg -i filebeat-7.5.0-amd64.deb
```

for other download options visit this page.

1.3 2.0 How filebeat works

Filebeat consists of two main components: inputs and harvesters.

1.3.1 2.1 What is a harvester

A harvester is responsible for reading the content of a single file. One harvester is started for each file and it reads the file, line by line, then sends the content to output. The harvester is responsible for opening and closing the file, which means that the file descriptor remains open while the harvester is running. If a file is removed or renamed while it's being harvested, Filebeat continues to read the file.

1.3.2 2.2 What is an input

An input is responsible for managing the harvesters and finding all sources to read from. Filebeat currently supports several input types. If the input type is log, input finds all files in the provided paths and starts a harvester for each file.

```
filebeat.inputs:
- type: log
paths:
- /var/log/*.log
- /var/path2/*.log
```

For more informations on how filebeat works visit this page.

1.4 3.0 Filebeat configuration

To configure Filebeat, you need to edit the configuration file. The default configuration file is called filebeat.yml and should be located in /etc/filebeat/. There is also example configuration file in the same folder called filebeat.reference.yml.

1.4.1 3.1 Specifying modules

Using filebeat modules is optional.

Modules provide quick way to get started with processing log formats. They contain default configurations, elasticsearch pipeline definitions and kibana dashboards for a quick implementation. Modules can be enable in a few ways:

- *Enable modules configs in the modules.d directory*
- *Enable modules when you run filebeat*
- *Enable module configs in the filebeat.yml file*

For more information on each way visit this page.

1.4.2 3.2 Inputs configuration

Another way of configuration is to manually configure inputs as oppose to enabling modules. To do so you specify list of inputs in the filebeat.inputs section of filebeat.yml file. Inputs specify how filebeat locates and processes input data.

```
filebeat.inputs:
- type: log
paths:
- /var/log/system.log
- /var/log/wifi.log
- type: log
paths:
- "/var/log/apache2/*"
fields:
```

```
apache: true
fields_under_root: true
```

Multiple inputs can be used even if they are the same type. Filebeat supports many inputs. See all available here.

1.4.3 3.3 Output configuration

Output can be configured in an output section of a filebeat config file. Only one output can be defined. Filebeat supports a variety of outputs. Some of them are logstash and elasticsearch. Example of logstash output:

```
output.logstash:
hosts: ["0.0.0.0:5044"]
ssl.enabled: true
ssl.certificate_authorities: ["/etc/logstash/logstash.crt"]
```

For more information about logstash output go here or for more information on other output options go here

1.4.4 3.4 Multiline messages management

Log files that are processed can contain messages that span multiple lines of text such as java stack traces.

1.4.4.1 3.4.1 Configuration options

To handle these multiline events you need to configure multiline settings in the filebeat.yml config file. You can specify these options in the filebeat.inputs section of the config file. Here is the example:

```
multiline.pattern: '^\['
multiline.negate: true
multiline.match: after
```

In multiline.pattern is specified regexp pattern to match. Multiline.negates defines whether the pattern is negated and multiline.match specifies how filebeat combines matching lines into an event. The settings are "before" and "after" and behaviour depends on the value in negate option. For more info on multiline management visit this page or for more multiline example this page

1.4.5 3.5 Filter and enhance the exported data

In some cases there might be some sufficient data or some additional information needs to be included. Filebeat provides a couple of options for data modification. For each input can be specified to include or exclude certain files or lines. This is done by adding corresponding options to the filebeat.inputs section

```
filebeat.inputs:
- type: log
...
exclude_files: ['\.gz$']
include_lines: ['^ERR', '^WARN']
```

The disadvantage of this approach is that you need to implement a configuration option for each filtering criteria that you need.

Another approach is to define processors to configure global processing across all data.

1.4.5.1 3.5.1 Processors

Processors are defined to process events before they are sent to the configure output. Processors can be used for:

- *reducing the number of exported fields*
- *enhancing events with additional metadata*
- *performing additional processing and decoding*

Each processor receives an event, applies action and returns the event. If there is multiple processors defined they are executed in order they are defined. Some of the processors that can be used are drop_event and json decode. Examples:

```
processors:
- drop_event:
when:
regexp:
message: "^DBG:"
```

Full list of processors and their definitions can be found here.

1.4.6 3.6 Other configurations

There is a lot more configuration that can be done. Some of it include specifying SSL or general settings, loading external configuration files and much more. See full list here.

1.5 4.0 Running filebeat

1.5.1 4.1 Command line

Filebeat can be run from command line with following command

```
filebeat run -e
```

or

```
filebeat -e
```

It could also be run as a service

```
service filebeat start
```

For all filebeat commands visit this page.

1.5.2 4.2 Running filebeat on docker

Docker images for Filebeat are available from the Elastic Docker registry.

1.5.2.1 4.2.1 Pulling the image

Filebeat docker image can be obtained with docker pull command:

```
docker pull docker.elastic.co/beats/filebeat:7.5.0
```

1.5.2.2 4.2.2 Configure filebeat on docker

Configuration can be done in multiple ways, by providing config file via a volume mount or by creating custom image with user defined configuration.

1.5.2.2.1 4.2.2.1

One way to configure Filebeat on Docker is to provide filebeat.docker.yml via a volume mount. With docker run, the volume mount can be specified like this.

```
docker run -d \
--name=filebeat \
--user=root \
--volume="$(pwd)/filebeat.docker.yml:/usr/share/filebeat/filebeat.yml:ro" \
--volume="/var/lib/docker/containers:/var/lib/docker/containers:ro" \
--volume="/var/run/docker.sock:/var/run/docker.sock:ro" \
docker.elastic.co/beats/filebeat:7.5.0 filebeat -e -strict.perms=false \
-E output.elasticsearch.hosts=["elasticsearch:9200"]
```

1.5.2.2.2 4.2.2.2 Custom image configuration

It's possible to embed your Filebeat configuration in a custom image. Here is an example Dockerfile to achieve this:

```
FROM docker.elastic.co/beats/filebeat:7.5.0
COPY filebeat.yml /usr/share/filebeat/filebeat.yml
USER root
RUN chown root:filebeat /usr/share/filebeat/filebeat.yml
USER filebeat
```


PUZZLES
CLOUD