Filebeat

1. 介绍、原理

Filebeat是Beat成员之一，基于Go语言; Filebeat基于 `libbeat` 框架；

> libbeat, A Go framework for creating Beats; More products based on libbeat is here

1.1 工作原理

1.1.1 组成

> Filebeat consists of two main components: inputs and harvesters.

1.1.1.1 harvester

> A harvester is responsible for opening, reading and closing a single file.

- `harvester` 逐行读取文件，并发送到输出端；
- filebeat为每个文件启动一个harvester，当文件移除或重命名，harvester会继续读取文件；

> This has the side effect that the space on your disk is reserved until the harvester closes.
>
> 这会产生副作用，即在 `harvester` 关闭之前，磁盘上的空间是保留的
>
> By default, Filebeat keeps the file open until close_inactive is reached.
>
> Closing a harvester has the following consequences（影响/后果）:
>
> - The file handler is closed, freeing up the underlying resources if the file was deleted while the harvester was still reading the file.
> - The harvesting of the file will only be started again after `scan_frequency` has elapsed.
> - If the file is moved or removed while the harvester is closed, harvesting of the file will not continue.
>
> To control when a harvester is closed, use the `close_*` configuration options.

1.1.1.2 input

> An input is responsible for managing the harvesters and finding all sources to read from.
>
> If the input type is `log`, the input finds all files on the drive that match the defined glob paths and starts a harvester for each file. `Each input runs in its own Go routine`.
>
> New lines are only picked up if the size of the file has changed since the harvester was closed.

- **input_type**支持多种类型，每种类型可配置多次；
- 如果output不可到达，filebeat会一直记录最后一行的send事件，直到output连通再继续发送文件；
- 当filebeat运行时，filebeat会为每个input在内存中也创建state(状态)信息；
- filebeat重启时，使用registry_file(硬盘中保存state的文件)重建state信息；
- 由于文件可以移动和删除，filebeat使用UUID来判断文件是否传递过；

- 如果频繁地创建文件，导致registry_file太庞大，怎么办？

> Registry file is too large?

1.1.2 ensure at-least-once

- 每个发送事件都记录在registry_file中；
- 如果一个log event发送失败，filebeat会再次发送，知道接收到ack（acknowledge 确认信息）；
- 如果filebeat被shut down， 是不会等待ack的，而是直接关闭；重启时，那些未收到ack的event会再次发送；
  - 但是这样可能存在一个event发送了2份，可以通过设置 `shutdown_timeout` ，使filebeat关闭时等待一段时间；
- 遗漏日志：
  1. 日志文件回旋（rotated）太快，filebeat来不及处理；
  2. Linux系统下，重用inode可能导致filebeat读取日志文件跳行；

2. 配置

2.1 配套工具

- Elasticsearch for storing and indexing the data.
- Kibana for the UI.
- Logstash (optional) for parsing and enhancing the data.

> Logstash：分析处理数据

配套工具安装完成后，filebeat的安装、配置、运行教程请看这里

2.2 配置项

> Filebeat modules provide the fastest getting started experience for common log formats. If you want use Filebeat modules, go directly to Quick start: modules for common log formats.

filebeat的安装包各种文件目录结构看这里

更全面的filebeat的配置指导，请看这里

2.2.1 简单配置示例

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*
```

2.2.2 shutdown_timeout

问题：**Filebeat如何确保至少投递一次？**

> Filebeat保证事件将被投递到配置的输出中至少一次，并且不会丢失数据。Filebeat能够实现这种行为，因为它将每个事件的投递状态存储在注册表文件中。
>
> 在定义的输出被阻塞且没有确认所有事件的情况下，Filebeat将继续尝试发送事件，直到输出确认收到事件为止。
>
> 如果Filebeat在发送事件的过程中关闭了，则在关闭之前它不会等待输出确认所有事件。当Filebeat重新启动时，发送到输出（但在Filebeat关闭前未确认）的任何事件将再次发送。这确保每个事件至少被发送一次，但是你最终可

> 能会将重复的事件发送到输出。你可以通过设置 `shutdown_timeout` 选项，将Filebeat配置为在关闭之前等待特定的时间。

2.2.3 clean_remove

清理被移除文件的状态缓存：

2.2.4 clean_inactive

清理不活跃文件的状态缓存：

2.2.5 close_inactive

此项设置，以及clean_remove、clean_inactive等在这篇文章中都有涉及到：

2.2.5 paths

pattern形式支持go Glob

2.2.6 outputs

outputs支持类型

2.2.7 Elasticsearch/Kibana账号配置

```
output.elasticsearch:
  hosts: ["myEShost:9200"]
  username: "filebeat_internal"
  password: "YOUR_PASSWORD"
setup.kibana:
  host: "mykibanahost:5601"
  # 如果这里没有指定用户名和命名，kibana将使用output.elasticsearch中指定的密码
  username: "my_kibana_user"
  password: "YOUR_PASSWORD"
```

2.2.8 logstash

如果配置 `logstash` ，就需要将output.elasticsearch注释掉

```
output.logstash:
  hosts: ["127.0.0.1:5044"]
```

> For this configuration, you must load the index template into Elasticsearch manually because the options for auto loading the template are only available for the Elasticsearch output.

2.2.9 敏感信息保存

> Filebeat keystore, the syntax for environment variables: ${key}
>
> For example, imagine that the keystore contains a key called `ES_PWD` with the value `yourelasticsearchpassword` :
>
> - In the configuration file, use `output.elasticsearch.password: "${ES_PWD}"`
> - On the command line, use: `-E "output.elasticsearch.password=\${ES_PWD}"`

创建和管理keys：

> To create and manage keys, use the `keystore` command. See the command reference for the full command

> syntax, including optional flags.
>
> **Note**: The `keystore` command must be run by the same user who will run Filebeat.

- create a keystore

```
filebeat keystore create
```

- add keys

```
filebeat keystore add ES_PWD  # 然后会提示输入值
```

强制覆盖：

```
filebeat keystore add ES_PWD --force
```

通过stdin：

```
cat /file/containing/setting/value | filebeat keystore add ES_PWD --stdin --force
```

- list keys

```
filebeat keystore list
```

- remove keys

```
filebeat keystore remove ES_PWD
```

2.3 使用logstash

如果使用其他的output而不是Elasticsearch，那么需要 load the template manually。filebeat也提供了一个参考template：

> The recommended index template file for Filebeat is installed by the Filebeat packages. If you accept the default configuration in the `filebeat.yml` config file, Filebeat loads the template automatically after successfully connecting to Elasticsearch. If the template already exists, it's not overwritten unless you configure Filebeat to do so.

2.3.1 load-dashboards-logstash

2.4 inputs子配置项

2.4.1 scan_frequency

`harvester` 扫描（文件中）新行的频率；默认10s, 一般不建议设置太小，最好不要小于1s；

> 如果您需要近乎实时发送日志行，请不要使用非常低的 `scan_frequency` ，可以调整close_inactive以使文件处理程序保持打开状态并不断轮询您的文件。

2.4.2 close_inactive

表示最近一次 `harvester` 扫描到行开始，一段时间（close_inactive）内，如果 `harvester` 没有扫描到新行，就说明文件为 `inactive` 的，就会关闭这个文件对应的 `harvester` ，关闭了 `harvester` ，就表示关闭了 `file handler` ；

- 不建议将此项的值设置的太小，因为太小会导致频繁地关闭文件；
- `5m` - 表示5分钟；| `2h` - 表示2小时；

2.4.3 type

`type` 支持多种类型

2.4.4 路径符号 `**` 与 `recursive_glob.enabled`

表示多级目录，最多扩展到8级子目录；详情请参考官方文档

2.4.5 encoding

日志文件编码格式设置，常用格式有：

> plain, latin1, utf-8, utf-16be-bom, utf-16be, utf-16le, big5, gb18030, gbk, hz-gb-2312, euc-kr, euc-jp, iso-2022-jp, shift-jis, and so on

2.4.6 harvester_buffer_size

`harvester` 抓取文件的缓存大小，默认16k

2.4.7 max_bytes

单独一条 `message` 的最大字节数，超过部分会被抛弃，不发送到elasticsearch，默认10M;

2.4.8 json

3. 实战

3.1 安装Elasticsearch

直接官网下载解压，然后运行 `./bin/elasticsearch`；后台运行 `./bin/elasticsearch -d`；

可访问"localhost:9200?pretty"测试安装结果；

3.2 安装kibana

直接在官网下载，然后执行命令：

```
# uri默认是"http://localhost:9200"要包含schema，如http
nohup ./bin/kibana serve -e <elasticSearch-uri> > kibana-runtime.log 2>&1 &
```

可访问"localhost:5601"测试安装结果；

3.3 安装logstash

1. 下载安装包，直接解压
2. 在 `安装主目录/config` 目录下新建 `demo-metrics-pipeline.conf` (命名随意)：

```
input {
  beats {
    port => 5044
  }
}

# The filter part of this file is commented out to indicate that it
# is optional.
# filter {
#
# }

output {
  elasticsearch {
    hosts => "localhost:9200"
```

```
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
}
```

3. 启动logstash

```
nohup ./bin/logstash -f config/demo-metrics-pipeline.conf > logstash-runtime.log 2>&1 &
```

4. filebeat的配置文件filebeat.yml中，关闭elasticsearch输出，打开logstash：

```
#------------------------- Elasticsearch output ------------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
  #hosts: ["localhost:9200"]
.
.
.
#------------------------------ Logstash output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["localhost:5044"]
```

3.4 安装filebeat

1. 下载并解压二进制包
2. 更改配置文件
3. 启动命令：

```
nohup ./filebeat > runtime.log 2>&1 &
```

4. 使用

4.1 搜索

4.1.1 字符串

- 精确查询字符串

  双引号包裹

  ```
  "word"
  ```

- 搜索词组

  双引号包裹词组

  ```
  "Phrases and expressions"
  ```

- 范围查询

  闭区间

  ```
  responsetime: [10 TO *]
  ```

  开区间：

```
responsetime: {10 TO *}
```

- 逻辑操作

```
NOT type: mysql
```

```
(method: INSERT OR method: UPDATE) AND responsetime: [30 TO *]
```

More

filebeat命令大全

configuration-central-management 中心配置

设置index(索引)模板

所有modules