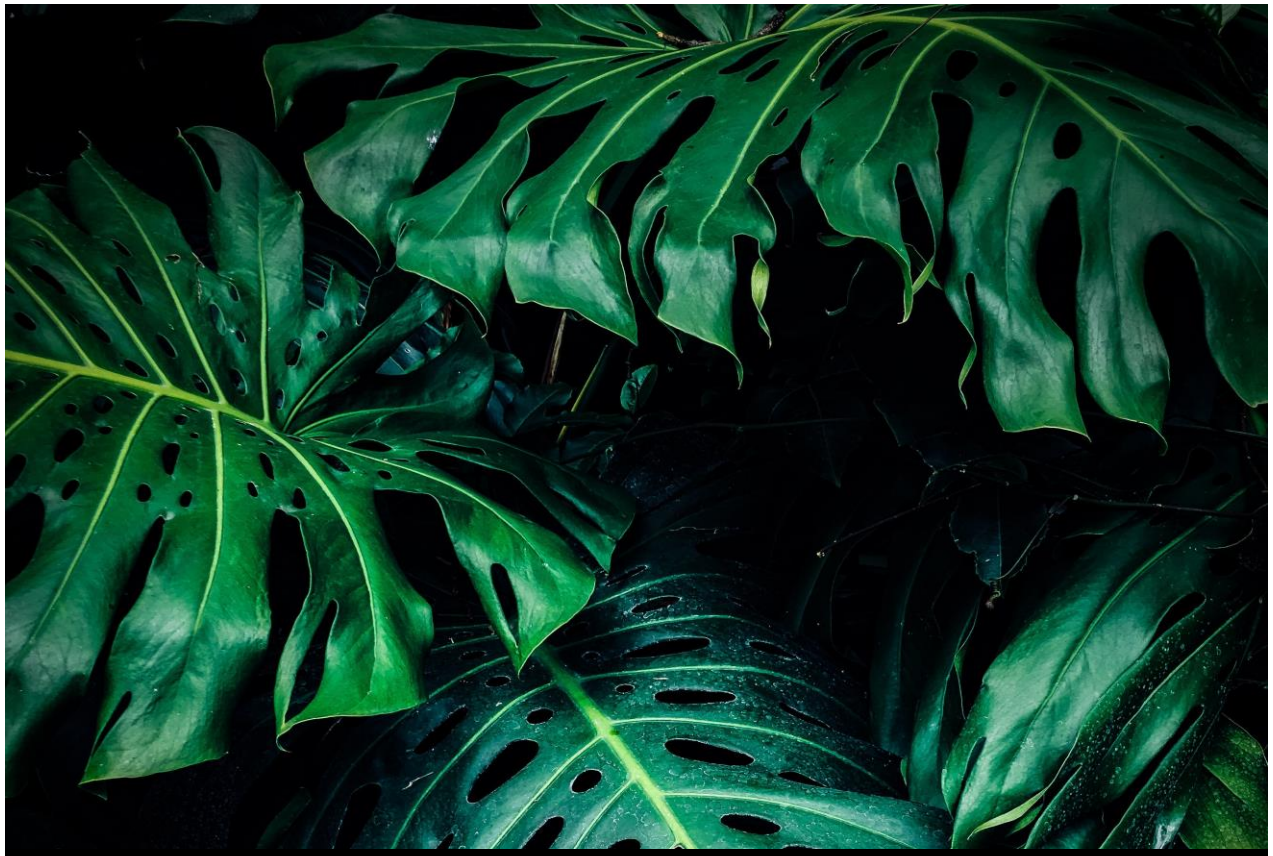
Elastic stack

Advanced course

ITFORVN

Course contents

- Centralized Logging Management
- Logstash advanced
- Kibana advanced
- APM – Application Performance Monitoring
- Alerting



02 – Logstash

Data transformation

- Input
- Filter
- Grok
- Geolocation
- Debug
- Pipeline

Logstash - inputs

- An input plugin enables a specific source of events to be read by Logstash.
- Các input hay dùng: file, tcp, udp

Ref:*https://www.elastic.co/guide/en/logstash/current/input-plugins.html*

- Cấu trúc chung Logstash configuration:

```
input {
  ...
}

filter {
  ...
}

output {
  ...
}
```

- Common option: hỗ trợ trên tất cả các input

Setting	Input type	Required
add_field	hash	No
codec	codec	No
enable_metric	boolean	No
id	string	No
tags	array	No
type	string	No

- Ref: *https://www.elastic.co/guide/en/logstash/current/plugins-inputs-file.html#plugins-inputs-file-common-options*

Logstash – inputs

- File input:

```
input {
  file {
    path => "/var/log/messages"
    type => "syslog"
  }

  file {
    path => "/var/log/apache/access.log"
    type => "apache"
  }
}
```

- TCP/UDP input:

```
input {
  tcp {
    port => 12345
    codec => json
  }

  udp {
    port => 23456
    codec => json
  }
}
```

- Codec: message input sẽ được parse theo codec được chỉ định. Nếu message "không match" với codec được chỉ định, toàn bộ message sẽ được lưu trong file "message" và dòng dữ liệu parse fail này sẽ được thêm tự động 1 tag đánh dấu "parse failed". Ví dụ với JSON parse failed sẽ là "_jsonparsefailure".
- Type: field này sẽ được add thêm vào dữ liệu, thường dung cho mục đích phân tách các luồng input và search.

Logstash – filter

- Filter được áp dụng trên dữ liệu, thực hiện các tác vụ "transformation" dự vào đặc điểm của input và mong muốn có được ở output.
- Logstash hỗ trợ hầu hết filter trên các định dạng thông dụng. Với các định dạng "custom", sử dụng "grok".
- Các filter thông dụng: csv, json, geoip, kv, mutate

- Ref: *https://www.elastic.co/guide/en/logstash/7.13/filter-plugins.html*

- Common option: hỗ trợ trên tất cả các filter

Setting	Input type	Required
add_field	hash	No
add_tag	array	No
enable_metric	boolean	No
id	string	No
periodic_flush	boolean	No
remove_field	array	No
remove_tag	array	No

- Ref: *https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html#plugins-filters-grok-common-options*

Logstash – filter kv

- Bộ lọc áp dụng trên dữ liệu dạng “key=value”:

```
filter {
  kv {  }
}

filter {
  kv {
    field_split => "&?"
  }
}
```

- Ví dụ:
  - **Input**: ?pin=12345~0&d=123&e=foo@bar.com&oq=bobo&ss=12345
  - **Output**:
    - `pin: 12345~0`
    - `d: 123`
    - `e: foo@bar.com`
    - `oq: bobo`
    - `ss: 12345`

- Ref: https://www.elastic.co/guide/en/logstash/7.13/plugins-filters-kv.html

Logstash – filter json

- Bộ lọc áp dụng trên dữ liệu định dạng JSON:

```
filter {
  json {
    source => "message"
  }
}
```

(source: là field bắt buộc, chứa dữ liệu ở định dạng JSON)

- Nếu dữ liệu parse failed sẽ được đánh tag "_jsonparsefailure".
- Nếu dữ liệu parse thành công:
  - Ví dụ: Mặc định dữ liệu sau khi parse sẽ được ghi ở "top level" root document.
  - Sử dụng "target" nếu muốn ghi dữ liệu ở một root document khác.

- Ref: https://www.elastic.co/guide/en/logstash/7.13/plugins-filters-json.html

Logstash – output

- Output plugin sẽ gởi dữ liệu sau khi parse đến một destination cụ thể (stash).
- Logstash hỗ trợ sẵn nhiều stash output khác nhau, được sử dụng nhiều nhất là Elasticsearch.

```
output {
  elasticsearch {
    hosts => ["127.0.0.1:9200","127.0.0.2:9200"]
    index => "%{[some_field][sub_field]}-%{+YYYY.MM.dd}"
  }
}
```

- **Lưu ý**: nếu 1 Elasticsearch là "dedicated master node", thì không nên đưa vào list hosts của Elasticsearch output plugin, vì "dedicated master node" không chứa dữ liệu.
- Test logstash config: **logstash --config.test_and_exit -f <path_to_config_file>**
- Debug output:

```
output {
  stdout { codec => rubydebug }
}
```

- Ref: https://www.elastic.co/guide/en/logstash/7.13/output-plugins.html

Logstash – grok overview

- Parse unstructured log data into something structured and queryable. Grok sử dụng regular expresion.
- Logstash có khoảng 120 pattern được build sẵn: https://github.com/logstash-plugins/logstash-patterns-core/tree/master/patterns
- Thư mục chứa các pattern default của Logstash (version 7.13): /usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/logstash-patterns-core-4.3.1/patterns/
- Tools:
- http://grokdebug.herokuapp.com/
- http://grokconstructor.appspot.com/

```
USERNAME [a-zA-Z0-9._-]+
USER %{USERNAME}
EMAILLOCALPART [a-zA-Z0-9!#$%&'*+\-/=?^_`{|}~]{1,64}(?:\.[a-zA-Z0-9!#$
EMAILADDRESS %{EMAILLOCALPART}@%{HOSTNAME}
INT (?:[+-]?(?:[0-9]+))
BASE10NUM (?<![0-9.+-])(?>[+-]?(?:(?:[0-9]+(?:\.[0-9]+)?)|(?:\.[0-9]+)
NUMBER (?:%{BASE10NUM})
BASE16NUM (?<![0-9A-Fa-f])(?:[+-]?(?:0x)?(?:[0-9A-Fa-f]+))
BASE16FLOAT \b(?<![0-9A-Fa-f.])(?:[+-]?(?:0x)?(?:(?:[0-9A-Fa-f]+(?:\.[

POSINT \b(?:[1-9][0-9]*)\b
NONNEGINT \b(?:[0-9]+)\b
WORD \b\w+\b
NOTSPACE \S+
SPACE \s*
DATA .*?
GREEDYDATA .*
QUOTEDSTRING (?>(?<!\\)(?>"(?>\\.|[^\\"]+)+"|""|(?>'(?>\\.|[^\\']+)+')
UUID [A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}
# URN, allowing use of RFC 2141 section 2.3 reserved characters
URN urn:[0-9A-Za-z][0-9A-Za-z-]{0,31}:(?:%[0-9a-fA-F]{2}|[0-9A-Za-z()+
```

Logstash – grok syntax

- Syntax: %{SYNTAX:SEMANTIC}
  - SYNTAX: pattern name
  - SEMANTIC: định danh gán cho "matched text". Mặc định "matched text" sẽ được lưu ở "string type"
- Ví dụ:
  - 123: match với pattern NUMBER
  - 208.67.222.222: match với pattern IP

- Ví dụ:
  - Text: `123 208.67.222.222`
  - Grok filter: `%{NUMBER:myid} %{IP:opendns}`
  - Grok filter: `%{NUMBER:myid:int} %{IP:opendns}` (myid sẽ có type là int thay vì mặc định là string)

Logstash – grok example

- Ví dụ:
  - Text: 55.3.244.1 GET /index.html 15824 0.043
  - Grok filter: %{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}

- Logstash config:

```
filter {
  grok {
    match => { "message" => "%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}" }
  }
}
```

```
- client: 55.3.244.1
- method: GET
- request: /index.html
- bytes: 15824
- duration: 0.043
```

Logstash – grok custom pattern

- Syntax: **(?<field_name>pattern)**
  - **field_name**: định danh gán cho "matched text"
  - **pattern**: regular expression pattern. Logstash sử dụng Oniguruma regex library: https://github.com/kkos/oniguruma/blob/master/doc/RE
- Ví dụ:
  - Text: 4e30b39b89564b9f29aeec09e96bf1ac
  - Grok filter: (?<md5hash>[a-z0-9]{32})
- "Custom pattern" có thể được khai báo ở dạng pattern và dùng như sau:

```
# cat ./patterns/myhash
MY_MD5HASH [a-z0-9]{32}


filter {
  grok {
    patterns_dir => ["./patterns"]
    match => { "message" => "%{SYSLOGBASE} %{MY_MD5HASH:md5hash}" }
  }
}
```

Logstash – mutate

- Thực hiện các tác vụ thay đổi thông thường trên field dữ liệu. Các tác vụ thường dùng:
  - **convert**: chuyển đổi định dạng dữ liệu. Ví dụ từ string sang int.
  - **rename**: rename field's name.
  - **replace**: replace the value of a field with a new value.
  - **gsub**: match a regular expression against a field value (string) and replace all matches with a replacement string.
  - **update**: update an existing field with a new value.
- Ví dụ:

```
filter {
  mutate {
    convert => {
      "fieldname" => "integer"
      "booleanfield" => "boolean"
    }
    gsub => [
      # replace all forward slashes with underscore
      "fieldname", "/", "_"
    ]
    rename => { "HOSTORIP" => "client_ip" }
    replace => { "message" => "%{source_host}: My new message" }
    replace => { "request" => "new request" }
  }
}
```

Logstash – geoip

- The GeoIP filter adds information about the geographical location of IP addresses, based on data from the MaxMind GeoLite2 databases.
- Logstash bundled hỗ trợ sẵn “GeoLite2-City” và “GeoLite2-ASN”.
- Configuration:

```
filter {
  grok {
    match => {
      "message" => "%{xxxxxxx}"
    }
  }

  geoip {
    source => "remote_addr"
  }
}
```

Logstash pipelines

Logstash – pipelines

- Mặc định logstash sử dụng 1 pipeline “main”.
- Pipeline được sử dụng khi có nhiều events flow, nhưng không chia sẻ cùng input/filter/output với nhau (có thể đang dung “tag” hoặc các điều kiện để phân biệt).
- Khai báo pipeline:

```
# /etc/logstash/pipelines.yml

- pipeline.id: main
  path.config: "/etc/logstash/conf.d/00-main.conf"

- pipeline.id: 01-nginx
  path.config: "/etc/logstash/conf.d/01-nginx.conf"
```

- Nếu không dùng pipeline, có thể sử dụng các điều kiện đầu vào như “tag”, “type” để phân tách từng events flow với nhau.
- Ví dụ (không **if** ở output): https://thachnuida.com/2016/12/04/push-nginx-log-to-elasticsearch-by-using-logstash/