



elasticsearch
for developer

ELK Stack

elk.txt



Logstash

Problem

ปกติเก็บ log file ไว้ทำอะไร ?

ต้องการค้นหาข้อมูลต้องใช้ grep หรือ shell script ?

สรุปผลจากข้อมูลอย่างไร ?

สร้างรายงานอย่างไร ?

จัดการข้อมูลอย่างไร ?

Source	Date
Apache	[23/Jan/2014:17:11:55 +0000]
Unix Timestamp	1390994740
Log4J	[2014/01/23 17:11:55, 000]
Postfix Log	Jan 23 17:11:55
ISO 8601	2014-01-23T17:11:55+01:00 2014-01-23

Logstash

- จัดการพวก Event และ log
- รวบรวมข้อมูล
- วิเคราะห์ข้อมูล
- ทำข้อมูลให้ขึ้น
- จัดเก็บข้อมูล

Logstash

- จัดการพวก Event และ log
 - รวบรวมข้อมูล
 - วิเคราะห์ข้อมูล
 - ทำข้อมูลให้ขึ้น
 - จัดเก็บข้อมูล
- } input
- } filter
- } output

Architecture

Input

Filter

Output

?



?

Outputs

boundary circonus cloudwatch csv datadog
elasticsearch exec **email** file ganglia gelf
gemfire google_bigquery google_cloud_storage
graphite graphtastic **hipchat** http irc jira
juggernaut librato loggly lumberjack
metriccatcher mongodb **nagios** null opentsdb
pagerduty pipe **rabbitmq** redis riak riemann s3
sns solr_http sqs statsd stdout stomp syslog
tcp udp websocket xmpp zabbix **zeromq**

Hello Logstash

```
# simple.conf
```

```
input {
```

```
    stdin{}
```

```
}
```

```
output {
```

```
    stdout {
```

```
        codec => rubydebug
```

```
    }
```

```
}
```

Hello Logstash

```
$echo hello | bin/logstash -f simple.conf
```

```
{  
  "message" => "hello",  
  "@version" => "1",  
  "@timestamp" => "2015-01-29T13:59:20.288Z",  
  "host" => "somkiatpui.local"  
}
```

Filter ?

#filter.conf

```
input {  
    stdin{}  
}  
  
filter {  
    grok {  
        match => ["message", "%{WORD:firstname} %{WORD:lastname} %{NUMBER:age}"]  
    }  
}  
  
output {  
    stdout {  
        codec => rubydebug  
    }  
}
```

Filter !

```
$echo "somkiat pui 30" | bin/logstash -f filter.conf
```

```
{  
    "message" => "somkiat pui 30",  
    "@version" => "1",  
    "@timestamp" => "2015-01-29T14:36:04.528Z",  
    "host" => "somkiatpui.local",  
    "firstname" => "somkiat",  
    "lastname" => "pui",  
    "age" => "30"  
}
```

Grok

จัดการข้อมูลด้วย Regular expression

<http://logstash.net/docs/1.4.2/filters/grok>

Default patterns

<https://github.com/elasticsearch/logstash/tree/v1.4.2/patterns>

Grok debugger

<https://grokdebug.herokuapp.com/>

Filters

advisor alter **anonymize** checksum cidr cipher
clone collate **csv** **date** **dns** **drop** elapsed
elasticsearch environment extractnumbers
fingerprint gelfify **geoip** grep **grok** grokdiscovery
i18n json json_encode kv metaevent **metrics**
multiline **mutate** noop prune punct
railsparallelrequest range ruby sleep split
sumnumbers syslog_pri throttle translate unique
urldecode **useragent** uuid wms wmts xml
zeromq

Write to elasticsearch

```
#elasticsearch.conf
```

```
input {
```

```
    stdin{}
```

```
}
```

```
output {
```

```
    elasticsearch_http {
```

```
        host => localhost
```

```
    }
```

```
}
```

Workshop

Input

Elasticsearch log file => logs/elasticsearch.log

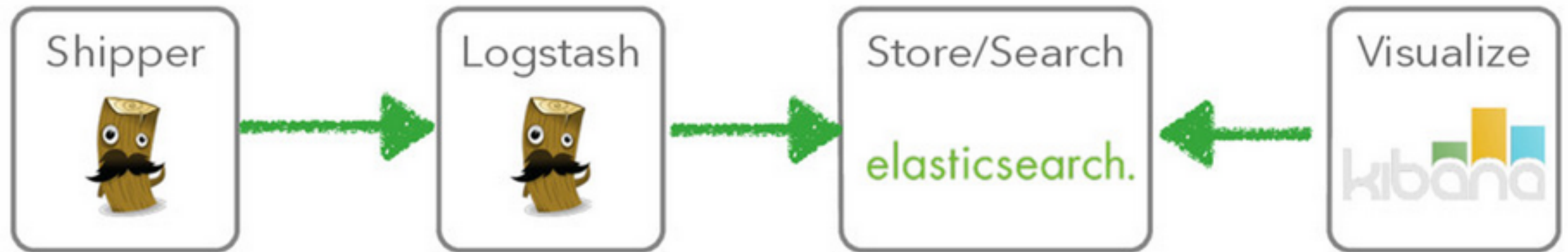
Output

Need level of log => INFO, ERROR, WARN

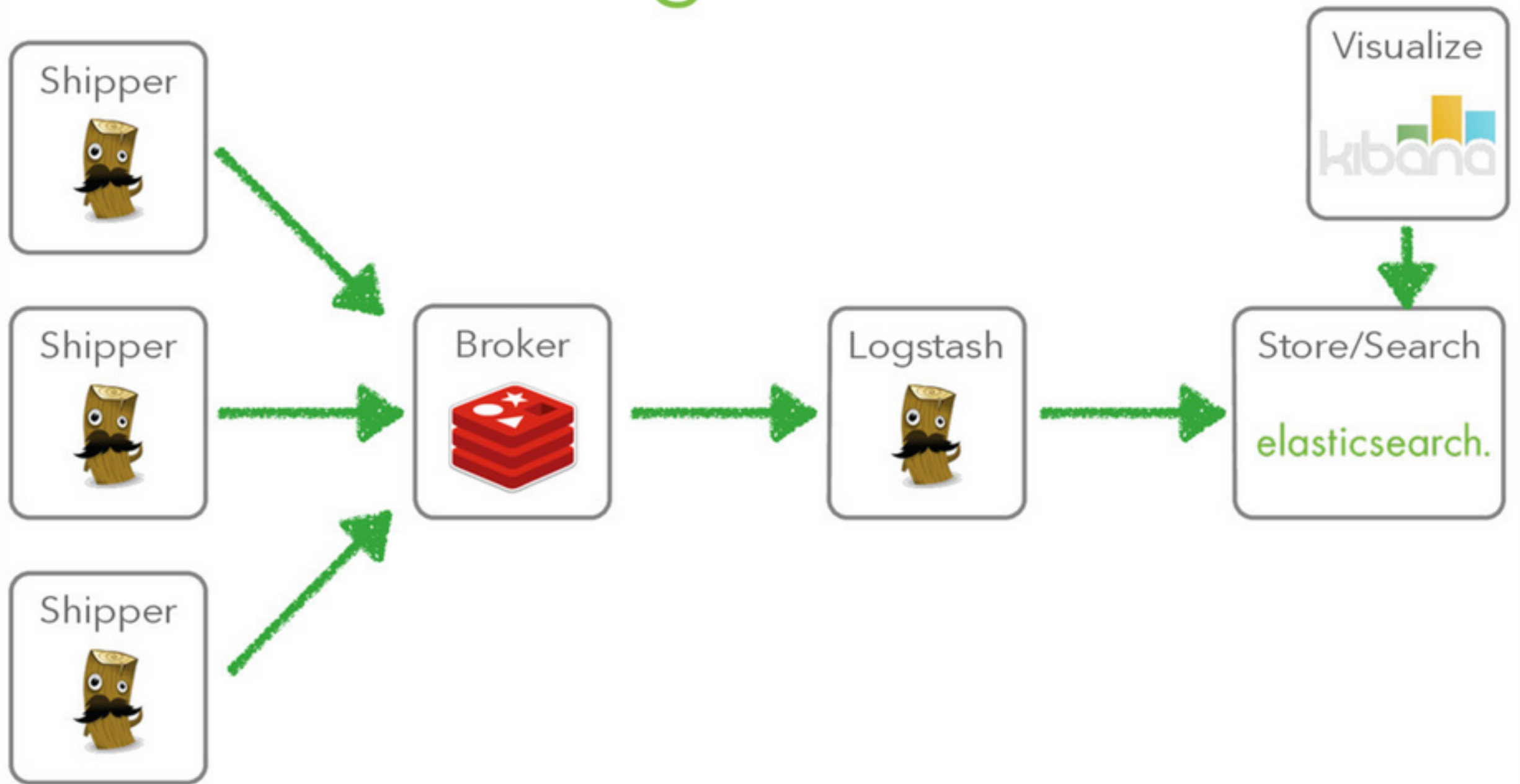
Write to elastic search

Visualize in Kibana !!

Use case :: log file



Use case :: scale log file



Scale data to your need !

Resources

Curator

<https://github.com/elasticsearch/curator>

Logstash cookbook

<https://github.com/logstash/cookbook>

Logstash forwarder

<https://github.com/elasticsearch/logstash-forwarder>