
NEW ORLEANS
DRUPALCON 2016

Drupal and Logstash: centralised logging

Marji Cermak

Marji Cermak

Systems Engineer at Morpht

@cermakm

To get you an idea

Customer says they get randomly redirected while browsing their website…

The old school

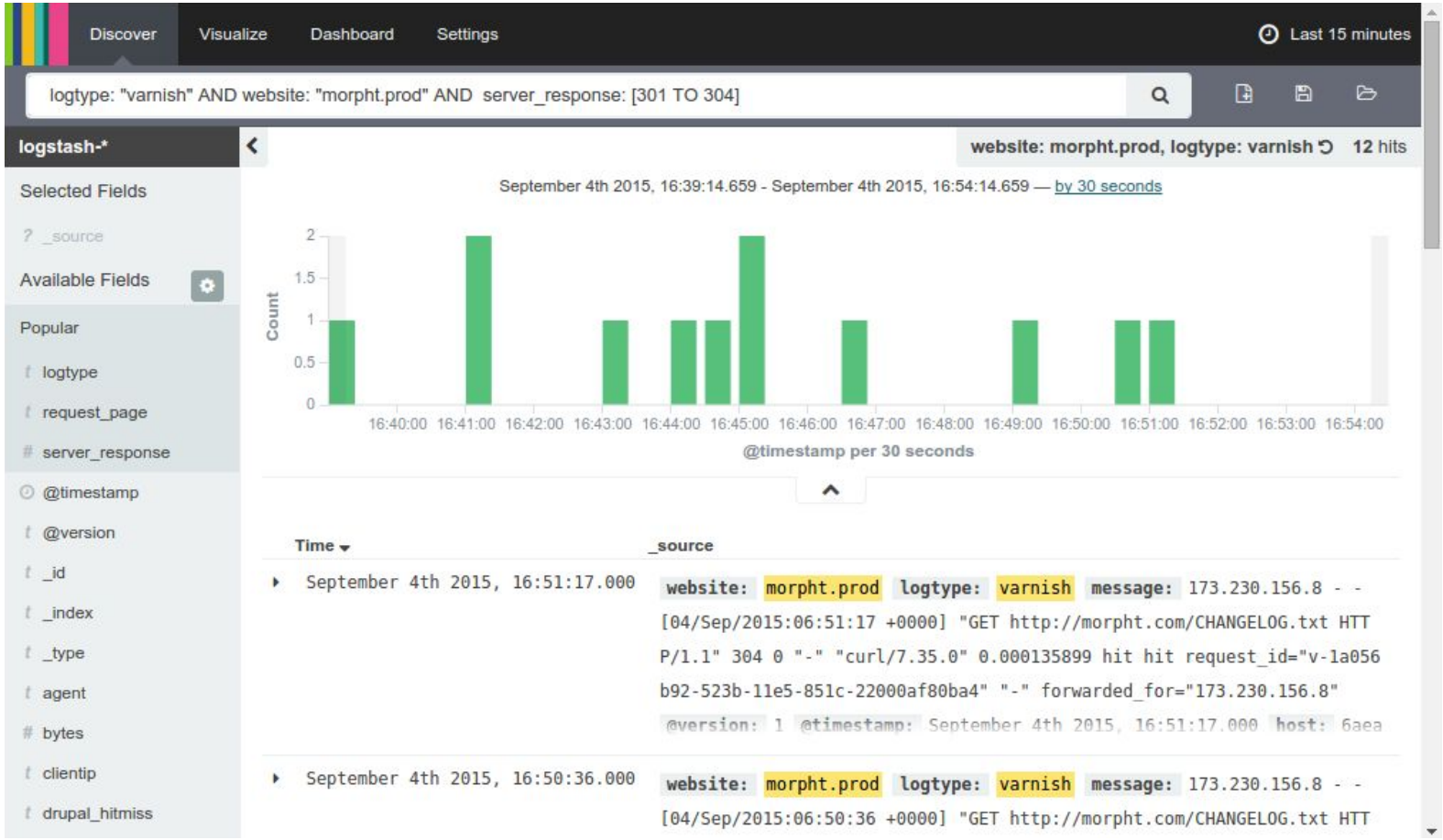
```
$ grep ' 30[1234] ' /var/logs/apache2/access.log | grep -v
baidu | grep -v Googlebot

173.230.156.8 - - [04/Sep/2015:06:10:10 +0000] "GET /morpht HTTP/1.0" 301 26
"-" "Mozilla/5.0 (pc-x86_64-linux-gnu)"
192.3.83.5 - - [04/Sep/2015:06:10:22 +0000] "GET /?q=node/add HTTP/1.0" 301
26 "http://morpht.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1)
AppleWebKit/600.2.5 (KHTML, like Gecko) Version/8.0.2 Safari/600.2.5"
192.3.83.5 - - [04/Sep/2015:06:10:23 +0000] "GET /?q=user/register HTTP/1.0"
301 26 "http://morpht.com/node/add" "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_10_1) AppleWebKit/600.2.5 (KHTML, like Gecko) Version/8.0.2 Safari/600.
2.5"
```

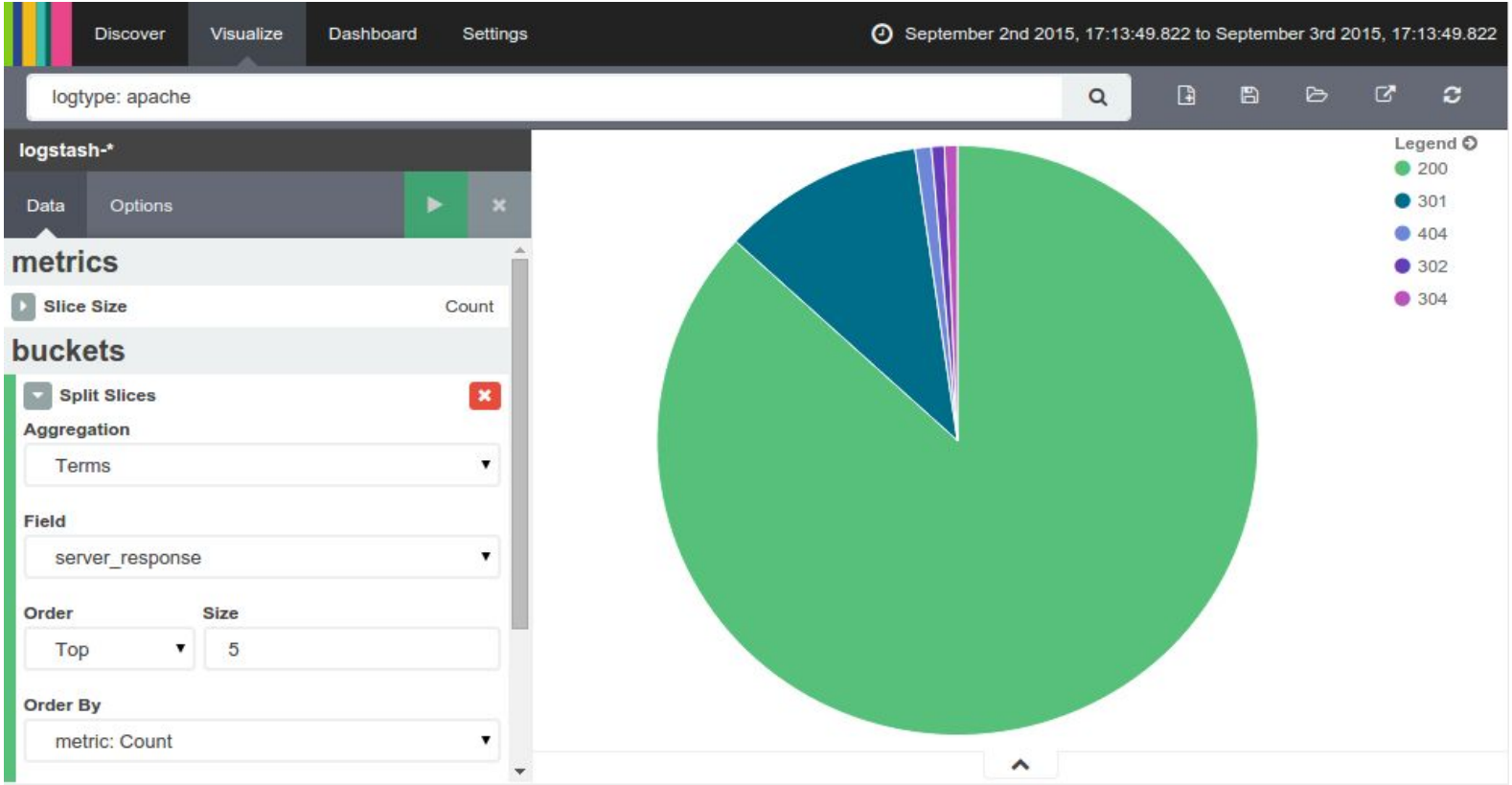

The new school

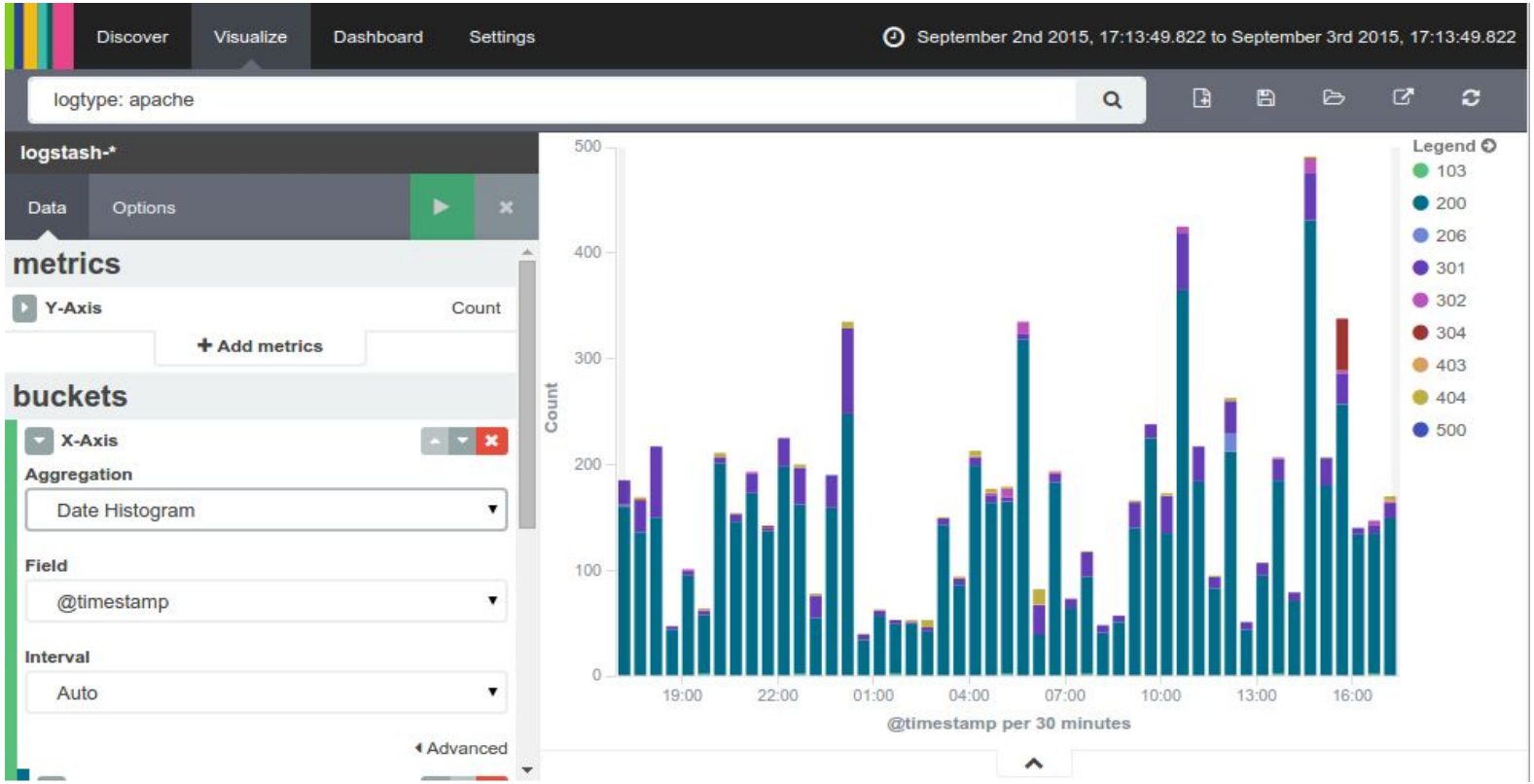
logtype: "apache" AND

website: "mysite" AND

server_response: [301 TO 304]


Discover
Visualize
Dashboard
Settings
Last 15 minutes
logtype: "varnish" AND website: "morpht.prod" AND server_response: [301 TO 304]
logstash-*
website: morpht.prod, logtype: varnish
12 hits
Selected Fields
_source
Available Fields
Popular
logtype
request_page
server_response
@timestamp
@version
_id
_index
_type
agent
bytes
clientip
drupal_hitmiss
September 4th 2015, 16:39:14.659 - September 4th 2015, 16:54:14.659 — by 30 seconds
Count
2
1.5
1
0.5
0
16:40:00 16:41:00 16:42:00 16:43:00 16:44:00 16:45:00 16:46:00 16:47:00 16:48:00 16:49:00 16:50:00 16:51:00 16:52:00 16:53:00 16:54:00
@timestamp per 30 seconds
Time
_source
September 4th 2015, 16:51:17.000
website: morpht.prod logtype: varnish message: 173.230.156.8 - - [04/Sep/2015:06:51:17 +0000] "GET http://morpht.com/CHANGELOG.txt HTTP/1.1" 304 0 "-" "curl/7.35.0" 0.000135899 hit hit request_id="v-1a056b92-523b-11e5-851c-22000af80ba4" "-" forwarded_for="173.230.156.8"
@version: 1 @timestamp: September 4th 2015, 16:51:17.000 host: 6aea
September 4th 2015, 16:50:36.000
website: morpht.prod logtype: varnish message: 173.230.156.8 - - [04/Sep/2015:06:50:36 +0000] "GET http://morpht.com/CHANGELOG.txt HTT


Discover
Visualize
Dashboard
Settings
September 2nd 2015, 17:13:49.822 to September 3rd 2015, 17:13:49.822
logtype: apache
logstash-*
Data
Options
metrics
Slice Size
Count
buckets
Split Slices
Aggregation
Terms
Field
server_response
Order
Top
Size
5
Order By
metric: Count
Legend
200
301
404
302
304


Discover
Visualize
Dashboard
Settings
September 2nd 2015, 17:13:49.822 to September 3rd 2015, 17:13:49.822
logtype: apache
logstash-*
Data
Options
metrics
Y-Axis
Count
+ Add metrics
buckets
X-Axis
Aggregation
Date Histogram
Field
@timestamp
Interval
Auto
Advanced
Count
500
400
300
200
100
0
19:00
22:00
01:00
04:00
07:00
10:00
13:00
16:00
@timestamp per 30 minutes
Legend
103
200
206
301
302
304
403
404
500

What have we just seen?

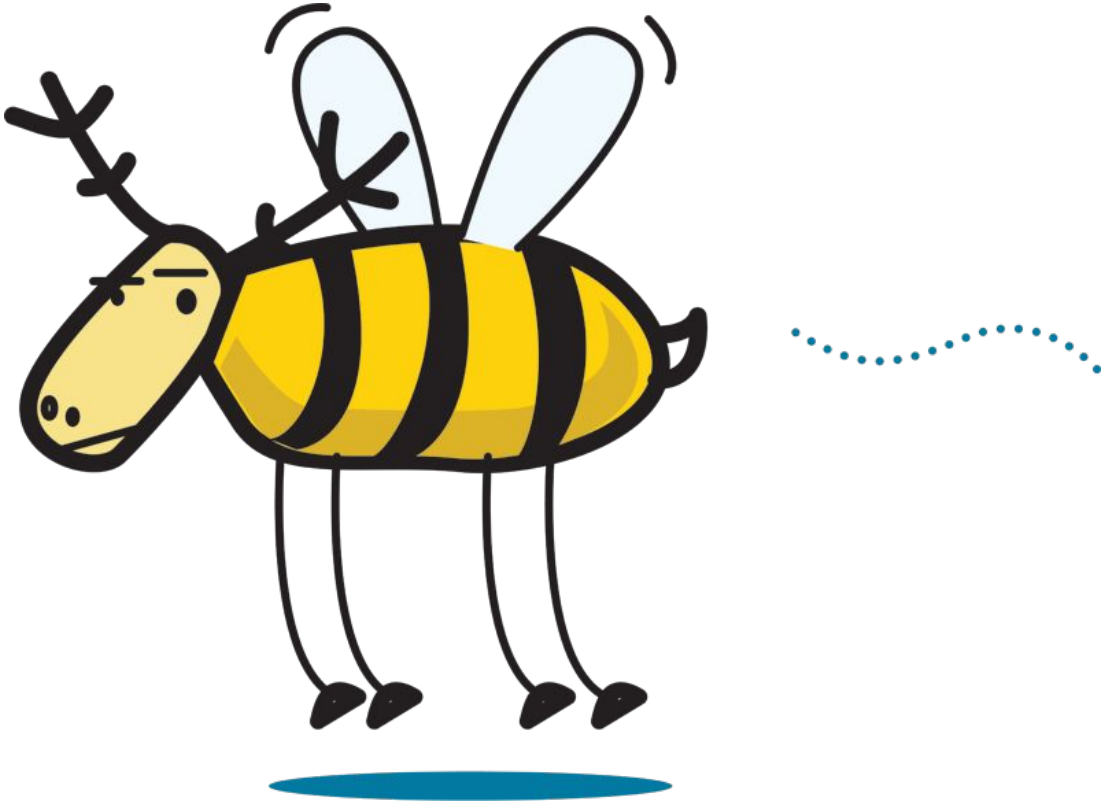
- These were interactions with Kibana.
- We executed a query, created several visualisations.
- But what else is under the hood?
- And where is logstash?

Source: "Family of Elk on Grassland" (CC BY-NC-ND 2.0) by Princess-Lodges

The ELK stack

Elasticsearch **L**ogstash **K**ibana

Marji Cermak @cermakm



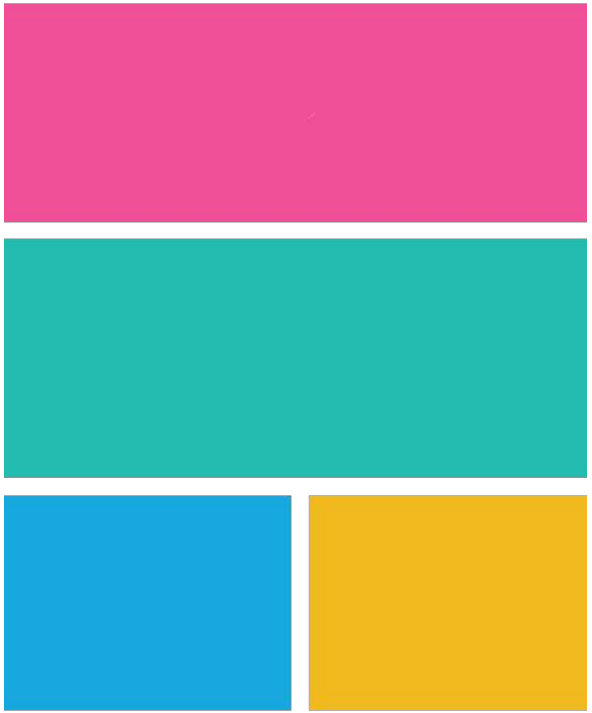
Source: https://www.elastic.co/blog/heya-elastic-stack-and-x-pack

The BELK stack

Beats **E**lasticsearch **L**ogstash **K**ibana

Marji Cermak @cermakm

The elastic stack



The elastic stack

The stack's goal

- Take data from any source, any format,

The stack's goal

- Take data from any source, any format,
- process, transform and enrich it,

The stack's goal

- Take data from any source, any format,
- process, transform and enrich it,
- store it,

The stack's goal

- Take data from any source, any format,
- process, transform and enrich it,
- store it,
- so you can search, analyse and visualise it in real time.

The four components

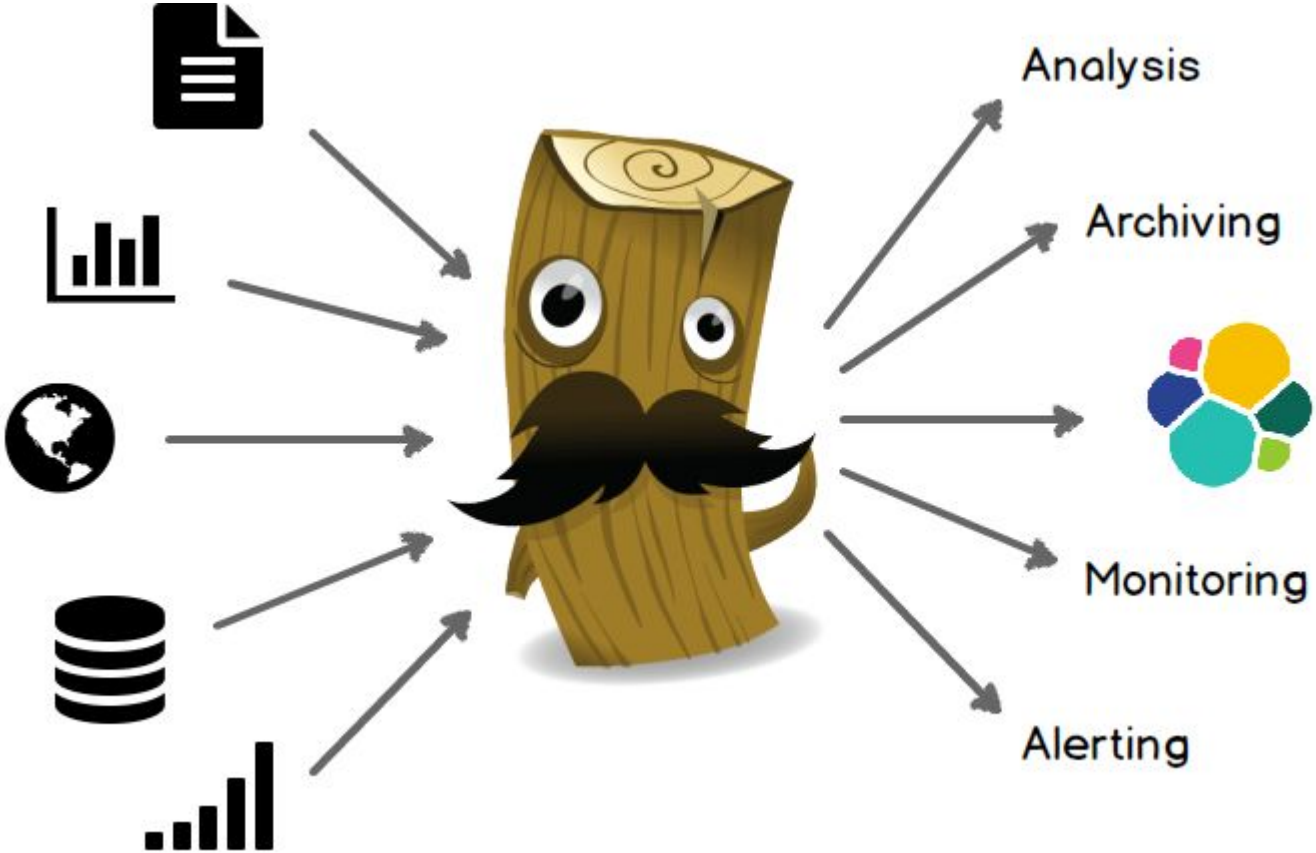
Elasticsearch

- open source, full-text search analytic engine
- distributed, High Availability
- designed for horizontal scalability and reliability
- based on Apache Lucene (like Apache solr)
- written in Java

elasticsearch

Logstash

- tool to collect, process, and forward events and log messages
- data collection, enrichment and transformation pipeline
- configurable input and output plugins
- e.g. logfile, MS windows eventlog, socket, Syslog, redis, salesforce, Drupal DBLog



Source: https://www.elastic.co/guide/en/logstash/current/introduction.html

Logstash

dozens of **input** plugins

- Beats
- **file**
- TCP, UDP, websocket
- syslog
- redis
- MS windows eventlog
- drupal_dblog

logstash

Logstash

dozens of **input** plugins

dozens of **output** plugins

- file
- TCP, UDP, websocket
- syslog
- redis, SQS
- graphite, influxdb
- nagios, zabbix
- jira, redmine
- s3
- **elasticsearch**

Logstash

dozens of **input** plugins

dozens of **output** plugins

dozens of **filter** plugins

- grok
- mutate
- drop
- date
- geoip

logstash

Kibana

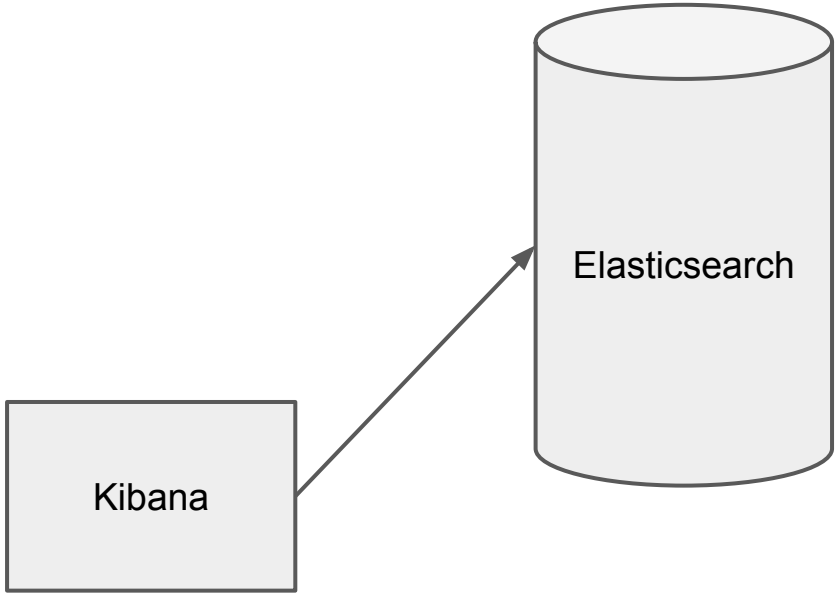
- open source data visualisation platform
- allows to interact with data through powerful graphics
- brings data to life with visuals

kibana

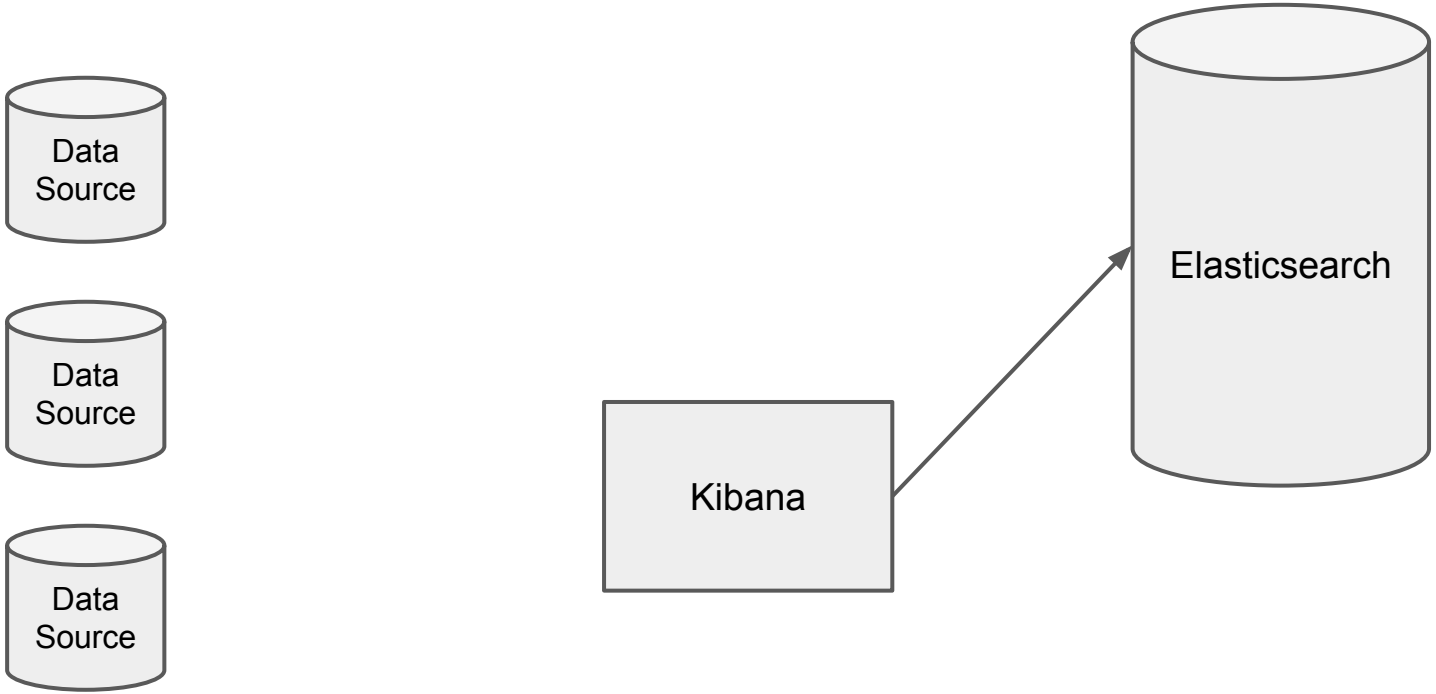
Beats

- Open source data shippers
- Lightweight
- e.g. network packets, log files

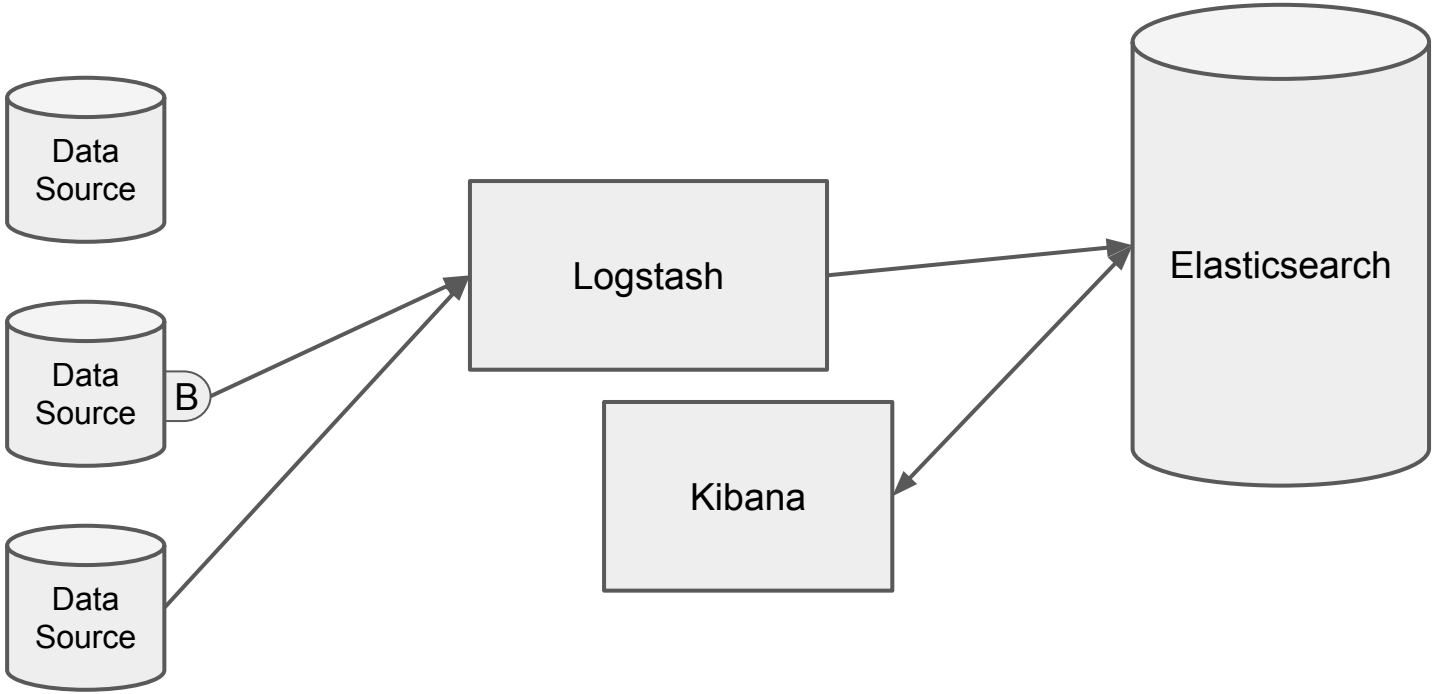
The BELK flow



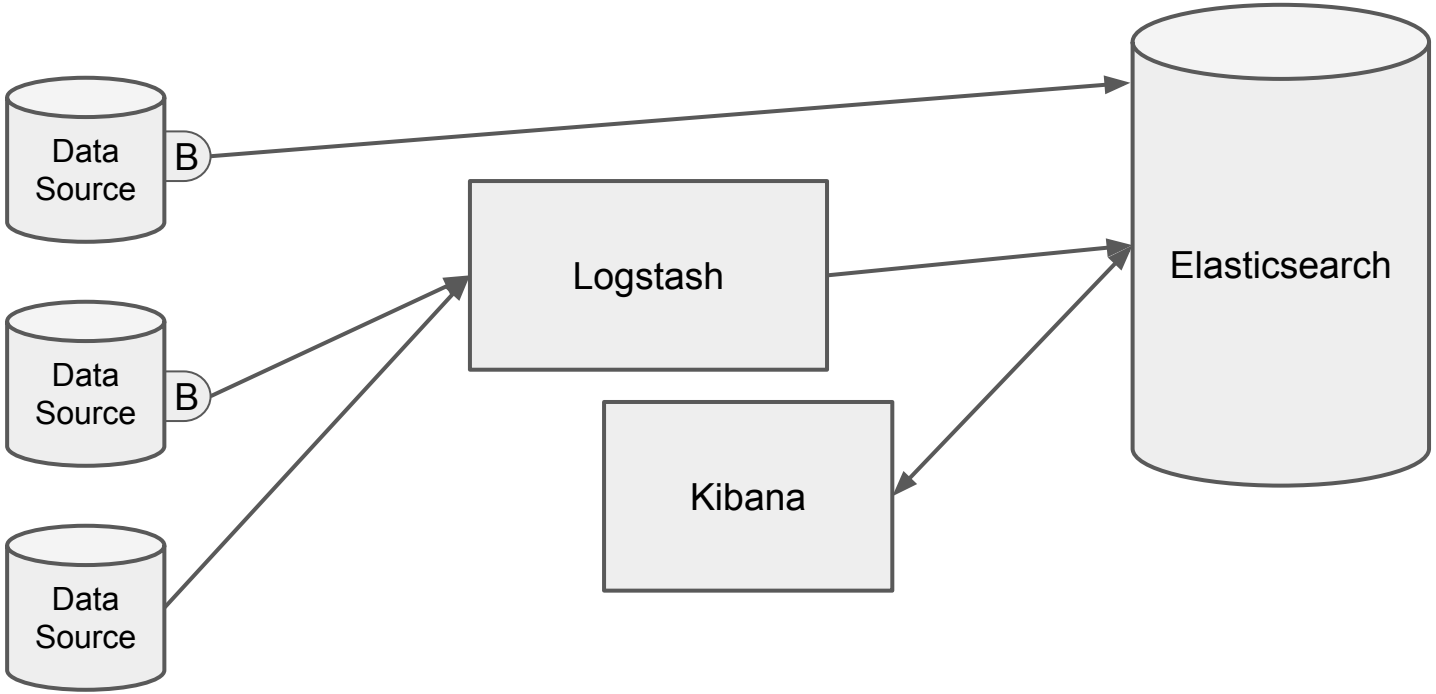
The BELK flow



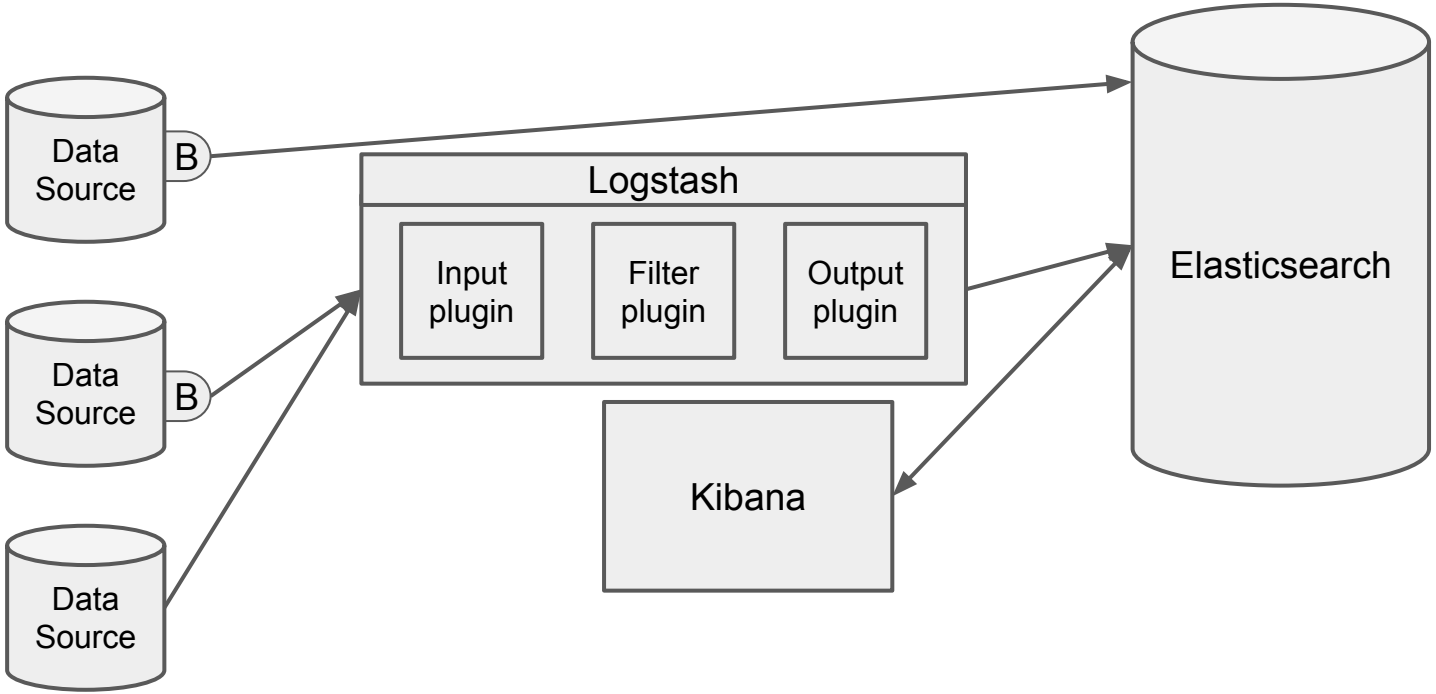
The BELK flow



The BELK flow



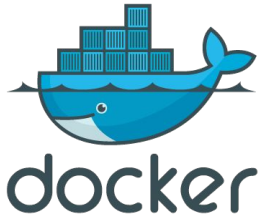
The BELK flow




docker

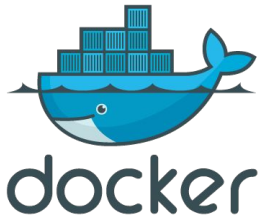
Docker

Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications.



Docker

Docker is a tool that can package an application and its dependencies in a virtual container that can run on any Linux server.



Docker Logstash Hello World!

```
docker run -it --rm logstash:2.3 logstash -e '
  input  { stdin { } }
  output { stdout { codec => rubydebug} }'
```

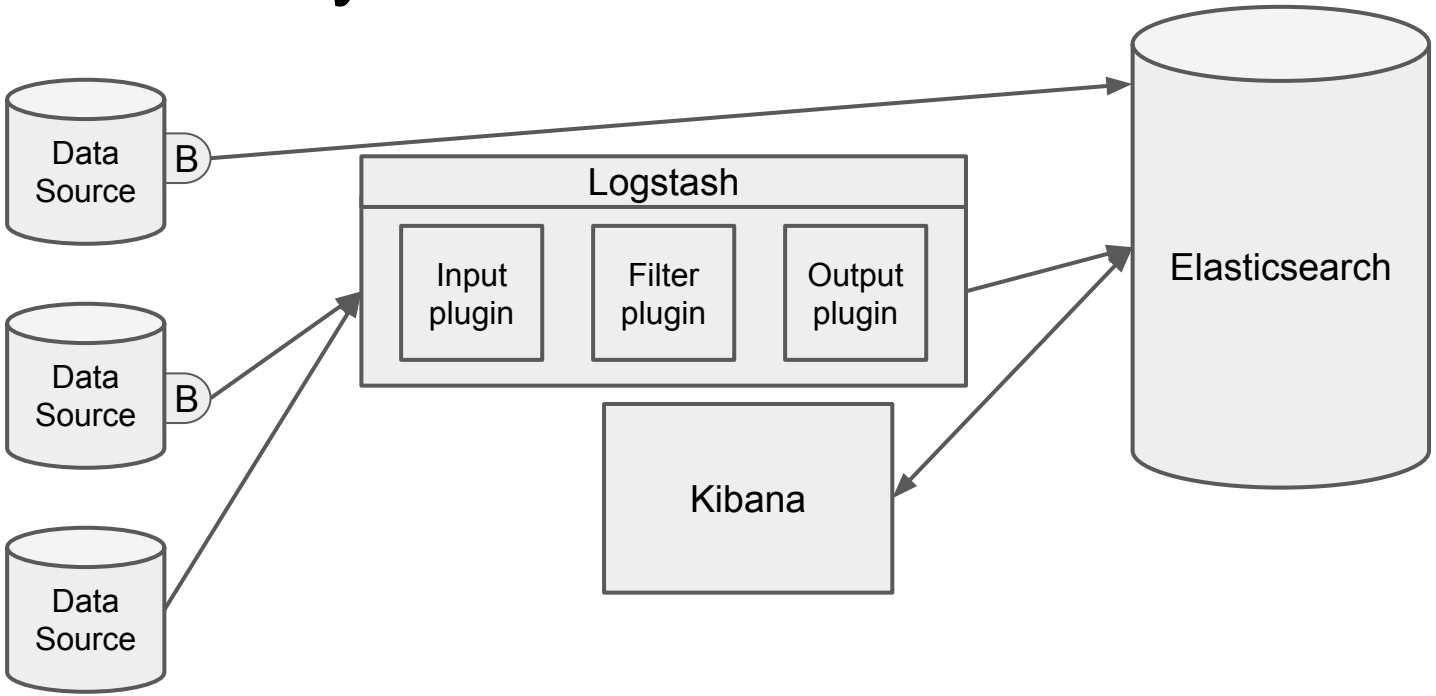
```
107.187.90.29 - - [05/Sep/2015:01:14:02 +0000] "GET / HTTP/1.1" 200 453 "-"
"curl/7.21.0"
```


Docker Logstash Hello World, apache!

```
docker run -it --rm logstash:2.3 logstash -e '
  input  { stdin { } }
  filter { grok { match => [ "message",
           "%{COMBINEDAPACHELOG}"]}}
  output { stdout { codec => rubydebug } }'
```

```
107.187.90.29 - - [05/Sep/2015:01:14:02 +0000] "GET / HTTP/1.1" 200 453 "-"
"curl/7.21.0"
```


Now let's try this



Docker ELK

Let's run 3 docker images:

```
$ docker run --name myes -d elasticsearch:2.3

$ docker run --name mykibana --link myes:elasticsearch
  -p 5601:5601 -d kibana:4.5

$ docker run --rm --link myes:elasticsearch
  -v ${PWD}/config-dir:/config-dir
  -v ${PWD}/source:/source
  logstash:2.3 logstash -f /config-dir
```


Is it going to work this time? :)

Local demo

What we have just seen

(In case it worked :)

- **Logstash** input reading lines from apache logfile

What we have just seen

(In case it worked :)

- **Logstash** input reading lines from apache logfile
- **Logstash** filter matching them with `COMBINEDAPACHELOG` pattern

What we have just seen

(In case it worked :)

- **Logstash** input reading lines from apache logfile
- **Logstash** filter matching them with `COMBINEDAPACHELOG` pattern
- **Logstash** output storing parsed lines to **Elasticsearch**

What we have just seen

(In case it worked :)

- **Logstash** input reading lines from apache logfile
- **Logstash** filter matching them with `COMBINEDAPACHELOG` pattern
- **Logstash** output storing parsed lines to **Elasticsearch**

- **Kibana** querying the data from **Elasticsearch**, visualising them

Logstash

dozens of **input** plugins

dozens of **output** plugins

```
input {
   file {
      path => "/source/access.log"
      type => "apache"
      start_position => "beginning"
   }
}

output {
   elasticsearch {
      hosts => ["myes"]
   }
   stdout { codec => rubydebug }
}
```


Logstash

dozens of **input** plugins

dozens of **output** plugins

dozens of **filter** plugins

```
filter {
  if [type] == "apache" {
    grok {
      match => [
        "message", "%{COMBINEDAPACHELOG}"
       ]
    }
  }
}
```


Logstash

dozens of **input** plugins

dozens of **output** plugins

dozens of **filter** plugins

```
filter {
  if [type] == "apache" {
    grok {
      match => [
        "message", "%{COMBINEDAPACHELOG}"
       ]
    }
    geoip { source => "clientip" }
  }
}
```


Logstash

dozens of **input** plugins

dozens of **output** plugins

dozens of **filter** plugins

```
filter {
  if [type] == "apache" {
    grok {
      match => [
        "message", "%{COMBINEDAPACHELOG}"
       ]
    }
    geoip { source => "clientip" }
    date {
      locale => "en"
      match => [ "timestamp",
                 "dd/MMM/yyyy:HH:mm:ss Z" ]
    }
  }
}
```


Logstash grok filter

```
filter { grok { match => [ "message",  "%{COMBINEDAPACHELOG}"]}}
```

There are many pre-defined grok patterns, e.g.

- `GREEDYDATA .*`
- `USERNAME [a-zA-Z0-9._-]+`
- `POSINT \b(?:[1-9][0-9]*)\b`
- `COMMONAPACHELOG, COMBINEDAPACHELOG`
- `SYSLOGBASE`

Logstash grok filter

```
COMMONAPACHELOG %{IPORHOST:clientip} %{USER:ident} %{USER:auth}
\[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}
(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:
response} (?:%{NUMBER:bytes}|-)

COMBINEDAPACHELOG %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent}

127.0.0.1 - - [05/Sep/2015:01:10:04 +0000] "GET / HTTP/1.1" 200
490 "-" "Wget/1.13.4 (linux-gnu)"
```


Where are we at?

- We have described the elastic stack components

Where are we at?

- We have described the elastic stack components
- We have run a local instance of the stack

Where are we at?

- We have described the elastic stack components
- We have run a local instance of the stack
- We processed, stored and analysed apache log file.

Where are we at?

- We have described the elastic stack components
- We have run a local instance of the stack
- We processed, stored and analysed apache log file.
- Each of you could do the same (you need just two things: docker and a log file)

belk.site-showcase.com

Pick your poison

Centralised logging

Centralised logging

Get logs to one (secure) place

It is not a new thing: Rsyslog / syslog-ng

The more servers you have, the more important it is

A must have for clusters with auto scaling

Centralised logging

There are many options

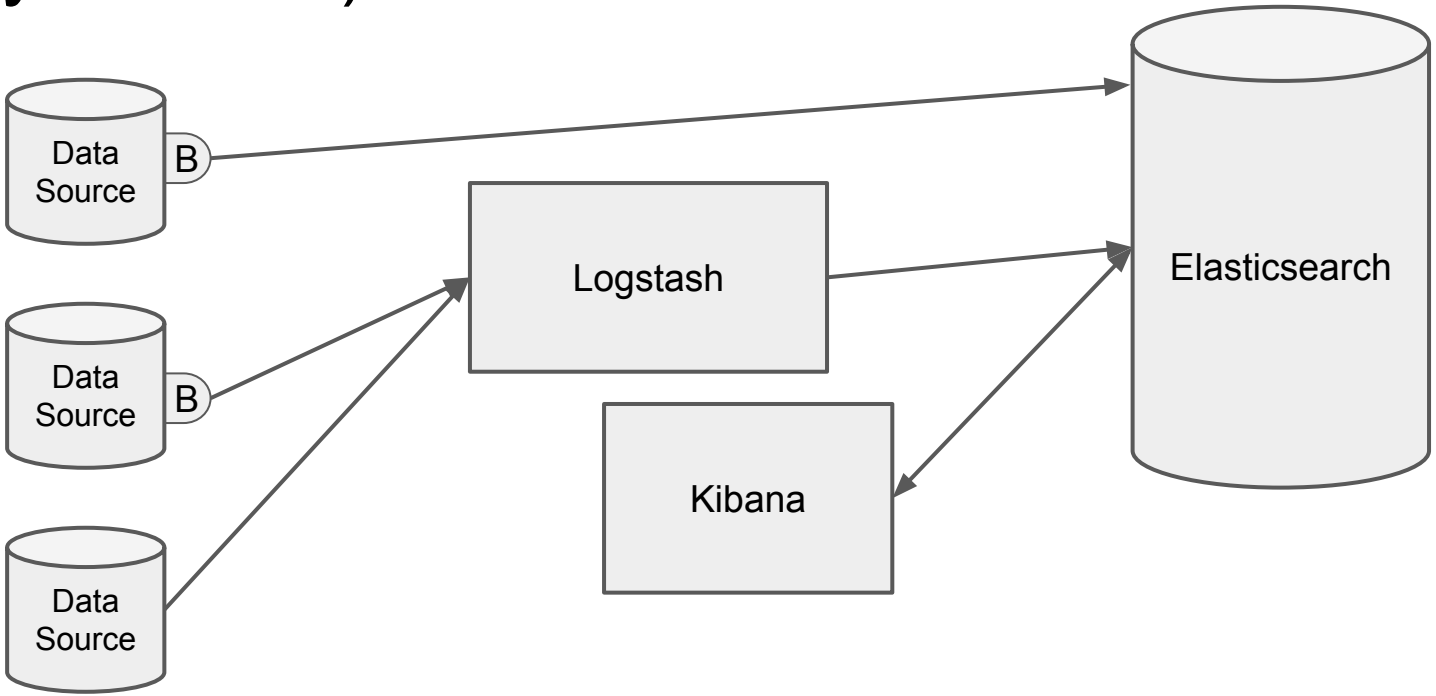
- Graylog
- Splunk
- Elastic stack

Centralised logging

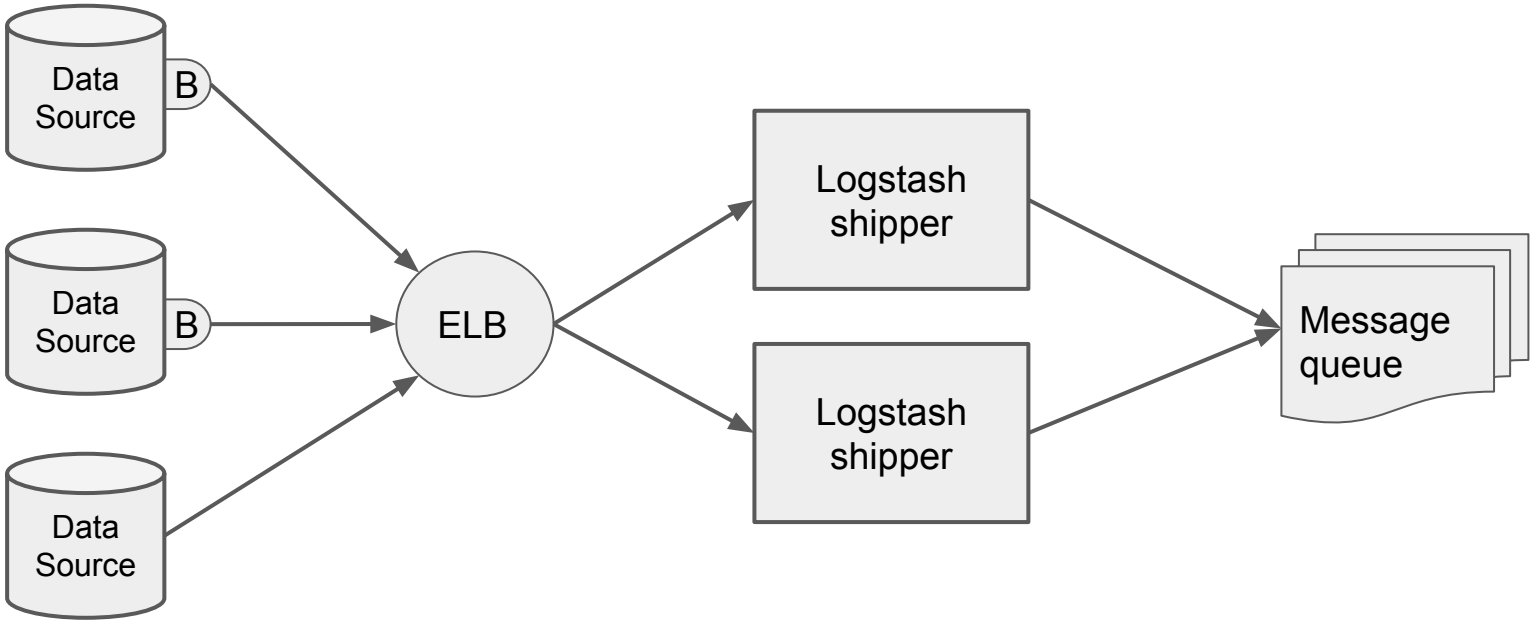
There are many SaaS options

- Datadog
- Loggly
- New Relic
- Sumo Logic
- Splunk
- Elastic Cloud

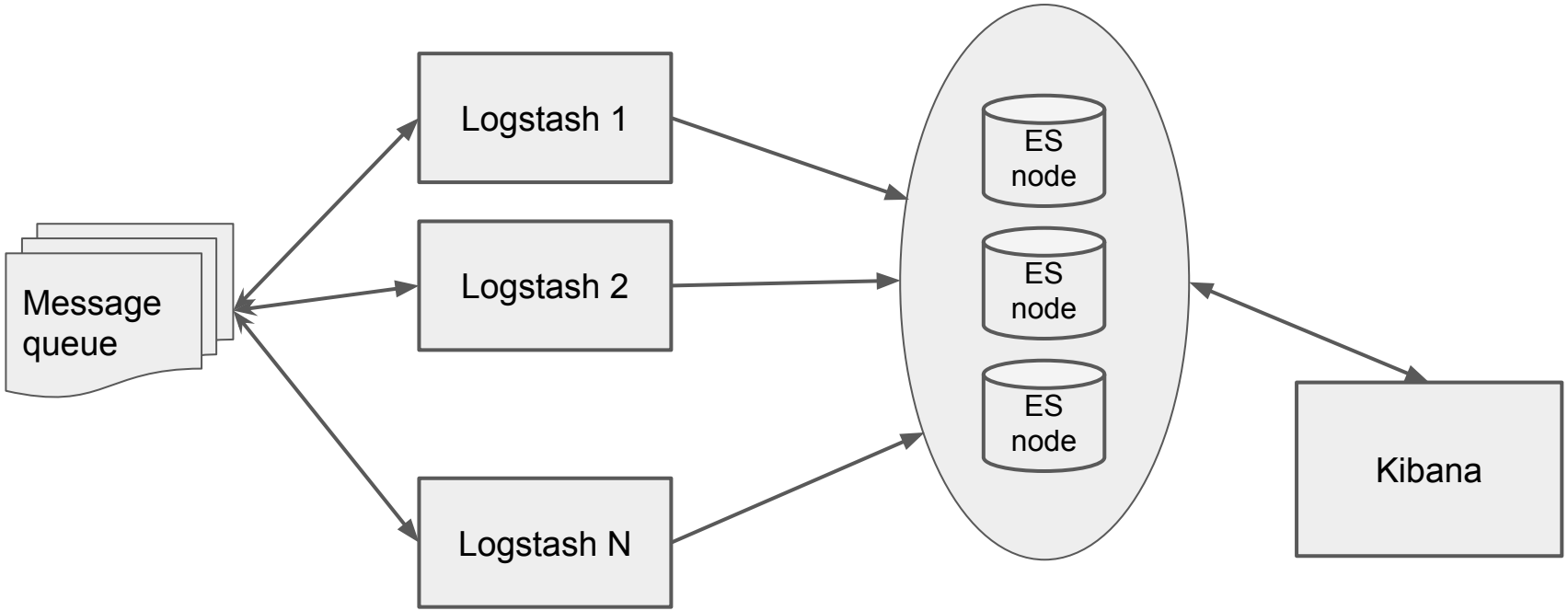
My choice :)


Data Source
B
Data Source
B
Data Source
Logstash
Kibana
Elasticsearch

High Available detour (1 of 2)



High Available detour (2 of 2)



Central ELK server demo

Similar ELK setup we tried locally, this time on a US hosted Linode.

Central ELK server demo

Similar ELK setup we tried locally, this time on a US hosted Linode.

Receiving logs from several sources:

- Japan based Linode LEMP via beats
- Germany based Linode LAMP via beats
- Australia based AWS instance via beats
- Australia based Acquia subscriptions

Central ELK server demo

If it works, we will have a look at:

- Drupal / watchdog logs
- Varnish logs
- Server metrics dashboard (teaser)
- and ...

Server logs using beats

Install **filebeat** package on the server with the logs.

Configure /etc/filebeat/filebeat.yml

```
filebeat:
  prospectors:
    -
      paths:
         - /var/log/apache/access.log
         - /var/log/nginx/access.log
         - /var/log/drupal.log


output:
  logstash:
    hosts: ["logstash.example.com:9876"]
```


Drupal logs

- Drupal syslog module, then get syslog log to ELK

```
create e.g. /etc/rsyslog.d/60-drupal.conf:

local0.* /var/log/drupal.log
```


Drupal logs

- Drupal syslog module, then get syslog log to ELK
- Logstash drupal_dblog input plugin (for dev)

```
input {
  drupal_dblog {
     databases =>
       ["site1", "mysql://usr:pass@host/db"]
     interval => "1"
   }
}
```


Acquia subscription logs

- Logstream gem
- wrapped in a docker container
- saving received logs to a local file

```
SUBS=test

logstream tail devcloud:${SUBS}
   prod --no-color  >> /opt/logs/${SUBS}.log
```


searching for the belk clicks

demo

Wrapping up

- We set up a local ELK stack.

Wrapping up

- We set up a local ELK stack.
- Processed an apache logfile, stored it.

Wrapping up

- We set up a local ELK stack.
- Processed an apache logfile, stored it.
- Hopefully it was very easy.

Wrapping up

- We set up a local ELK stack.
- Processed an apache logfile, stored it.
- Hopefully it was very easy.
- We examined the stored data, visualised it.

Wrapping up

- We set up a local ELK stack.
- Processed an apache logfile, stored it.
- Hopefully it was very easy.
- We examined the stored data, visualised it.

- We looked at a central logging solution, receiving logs from different sources.

Links

Main docs area for the ELK stack:
https://www.elastic.co/guide/index.html

The logstash book from James Turnbull
http://www.logstashbook.com/

Follow up blog post:
http://morpht.com/posts/drupal-and-logstash

Links

Docker
https://www.docker.com/

Official Docker images:

- https://hub.docker.com/_/logstash/
- https://hub.docker.com/_/elasticsearch/
- https://hub.docker.com/_/kibana/

Questions?

Thank you!

@cermakm

marji@morpht.com

So How Was It? - Tell Us What You Think

Evaluate this session - https://events.drupal.org/node/10096

Thanks!