Using elasticsearch, logstash and kibana to create realtime dashboards

Alexander Reelsen
@spinscale
alexander.reelsen@elasticsearch.com
Agenda

• The need, complexity and pain of logging
• Logstash basics
• Usage examples
• Scalability
• Tools
• Demo
about

• Me
  Interested in metrics, ops and the web
  Likes the JVM
  Working with elasticsearch since 2011

• Elasticsearch, founded in 2012
  Products: Elasticsearch, Logstash, Kibana, Marvel
  Professional services: Support & development subscriptions
  Trainings
Why collect & centralise data?

- Access log files without system access
- Shell scripting: Too limited or slow
- Using unique ids for errors aggregate it across your stack
- Reporting (everyone can create his/her own report)
  Don’t be your boss’ grep/charting library
Why collect & centralise data?

- Detect & correlate patterns
  Traffic, load, DDoS

- Scale out/down on-demand

- Bonus points: Unify your data to make it easily searchable
Unify data

- apache
- unix timestamp
- log4j
- postfix.log
- ISO 8601

[23/Jan/2014:17:11:55 +0000]

1390994740

[2014-01-29 12:28:25,470]

Feb 3 20:37:35

2009-01-01T12:00:00+01:00

2014-01-01
Enter logstash

- Managing events and logs
- Collect data
- Parse data
- Enrich data
- Store data (search and visualizing)
Enter logstash

- Managing events and logs
- Collect data
- Parse data
- Enrich data
- Store data (search and visualizing)
Logstash architecture

Input → Filter → Output

Logstash
Inputs

collectd  drupal_dblog  elasticsearch  eventlog  exec  file  ganglia  gelf  gemfire  generator  graphite  heroku  imap  irc  jmx  log4j  lumberjack  pipe  puppet_facter  rabbitmq  redis  relp  s3  snmptrap  sqlite  sqs  stdin  stomp  syslog  tcp  twitter  udp  unix  varnishlog  websocket  wmi  xmpp  zenoss  zeromq
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
Installation

• ruby application, but Java required (JRuby)

• Download tarball, deb, RPM (also repositories) no gem/dependency hell!

• Puppet module
Simple setup

- Download, create config and run

```
input {
    stdin {}
}

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

```
echo foo | logstash-1.4.0.rc1/bin/logstash -f simple.conf
{
    "message" => "foo"
    "@version" => "1"
    "@timestamp" => "2014-01-20T13:30:59.648Z"
    "host" => "kryptic.fritz.box"
}
```
Analyze the output

```json
{
    "message" => "foo",
    "@version" => "1",
    "@timestamp" => "2014-01-20T13:30:59.648Z",
    "host" => "kryptic.fritz.box"
}
```

- **message**: Original content
- **version**: internal
- **timestamp**: Current timestamp
- **host**: Logstash hostname
But what about filtering?

```json
input {
    stdin {}
}

filter {
    grok {
        match => [ "message" "{%WORD:firstname} {%WORD:lastname} {%NUMBER:age}" ]
    }
}

output {
    stdout { codec => rubydebug }
}
```
But what about filtering?

echo "Alexander Reelsen 30" | logstash-1.4.0.rc1/bin/logstash -f sample-2.conf
{
    "message" => "Alexander Reelsen 30"
    "@version" => "1"
    "@timestamp" => "2014-01-21T16:56:02.502Z"
    "host" => "kryptic"
    "firstname" => "Alexander"
    "lastname" => "Reelsen"
    "age" => "30"
}
Grok

• Maintaining regexes for mere mortals
  http://logstash.net/docs/1.3.3/filters/grok

• Default patterns
  ciscofw, haproxy, apache, syslog, cron, nagios, postfix, redis...

  https://github.com/logstash/logstash/tree/v1.3.3/patterns

• Grok Debugger
  https://grokdebug.herokuapp.com/
Syslog example with grok

```
input { stdin {} }

filter {
  grok {
    match => { "message" => "\%{SYSLOGTIMESTAMP:syslog_timestamp} \%{SYSLOGHOST:syslog_hostname} \%{DATA:syslog_program}(?:\%{POSINT:syslog_pid}\])?: \%{GREEDYDATA:syslog_message}" }
    date {
      match => [ "syslog_timestamp",
        "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}

output { stdout { codec => rubydebug } }
```
Syslog example with grok

cat sample-syslog.txt | logstash-1.4.0.rc1/bin/logstash -f sample-syslog.conf
{
    "message" => "Jun 10 04:04:01 lvps109-104-93-171 postfix/smtpd[11105]: connect from mail-we0-f196.google.com[74.125.82.196]"
    "@version" => "1"
    "@timestamp" => "2014-06-10T04:04:01.000+02:00"
    "host" => "kryptic.local"
    "syslog_timestamp" => "Jun 10 04:04:01"
    "syslog_hostname" => "lvps109-104-93-171"
    "syslog_program" => "postfix/smtpd"
    "syslog_pid" => "11105"
    "syslog_message" => "connect from mail-we0-f196.google.com[74.125.82.196]"
}
Syslog example with grok

```
cat sample-syslog.txt | java -jar logstash-1.3.3-flatjar.jar agent -f sample-syslog.conf
{
    "message" => "Jun 10 04:04:01 lvps109-104-93-171 postfix/smtpd[11105]: connect from mail-we0-f196.google.com[74.125.82.196]"
    "@version" => "1"
    "@timestamp" => "2014-06-10T04:04:01.000+02:00"
    "host" => "kryptic.local"
    "syslog_timestamp" => "Jun 10 04:04:01"
    "syslog_hostname" => "lvps109-104-93-171"
    "syslog_program" => "postfix/smtpd"
    "syslog_pid" => "11105"
    "syslog_message" => "connect from mail-we0-f196.google.com[74.125.82.196]"
}
```
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
Codecs

cloudtrail  compress_spooler  dots  edn  edn_lines  fluent  graphite  \textbf{json}  \textbf{json_lines}  json_spooler  line  \textbf{msgpack}  multiline  netflow  noop  oldlogstashjson  plain  rubydebug  spool
JSON codec

```plaintext
input {
    stdin {
        codec => json
    }
}

output {
    stdout { codec => rubydebug }
}

(echo -e '{"foo":"bar", "spam" : "eggs"\n} ' ) | logstash-1.4.0.rc1/bin/logstash -f sample-json-codec.conf
{
    "foo" => "bar"
    "spam" => "eggs"
    "@version" => "1"
    "@timestamp" => "2014-01-23T13:12:17.325Z"
    "host" => "kryptic.local"
}
```
JSON lines codec

```sh
(input { stdin { codec => json_lines } }
output { stdout { debug => true } })

( echo -e '{"foo":"bar", "spam" : "eggs" }' ;
  echo '{ "c":"d", "e": "f" }') | logstash-1.4.0.rc1/bin/logstash -f sample-json-multi-codec.conf

{
  "foo" => "bar"
  "spam" => "eggs"
  "@version" => "1"
  "@timestamp" => "2014-01-23T13:17:47.582Z"
  "host" => "kryptic.local"
}

{
  "c" => "d"
  "e" => "f"
  "@version" => "1"
  "@timestamp" => "2014-01-23T13:17:47.584Z"
  "host" => "kryptic.local"
}
CLF log files

193.99.144.85 -- [23/Jan/2014:17:11:55 +0000] "GET / HTTP/1.1" 200 140 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.5 Safari/535.19"


input { stdin {} }

filter {
  grok {
    match => { message "%%{COMBINEDAPACHELOG}" }
  }
}

output { stdout { codec => rubydebug } }

CLF log files

```json
{
    "message" => "193.99.144.85 -- [23/Jan/2014:17:11:55 +0000] "GET / HTTP/1.1" 200 140 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.5 Safari/535.19"
    "@version" => "1"
    "@timestamp" => "2014-01-24T07:56:02.460Z"
    "host" => "kryptic.local"
    "clientip" => "193.99.144.85"
    "ident" => "-"
    "auth" => "-"
    "timestamp" => "23/Jan/2014:17:11:55 +0000"
    "verb" => "GET"
    "request" => "/
    "httpversion" => "1.1"
    "response" => "200"
    "bytes" => "140"
    "referrer" => "-"
    "agent" => "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.5 Safari/535.19"
}
Write to elasticsearch

```json
input { stdin {} }

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

output {
    elasticsearch {
        protocol => 'http'
    }
}
```
Use case: Log files

Shipper → Logstash → Store/Search → Visualize
Use case: Log files with broker

Shipper -> Broker -> Logstash -> Store/Search

Visualize
kibana

elasticsearch.
Use case: Log files with broker

Shipper → Broker → Logstash → Store/Search → Kibana → Visualize

Copyright Elasticsearch 2014. Copying, publishing and/or distributing without written permission is strictly prohibited.
Scale out any component

Shipper

Broker

Logstash

Store/Search

Visualize

elasticsearch.
Scale out any component

Shipper

Broker

Logstash

Visualize

Shipper

Broker

Logstash

Store/Search

elasticsearch
Scale any component

Shipper  
Broker  
Logstash  
Visualize

Shipper  
Broker  
Logstash  
Store/Search

Shipper  
Broker  
Logstash  
Store/Search

Copyright Elasticsearch 2014. Copying, publishing and/or distributing without written permission is strictly prohibited
Logstash scaling

- Events get passed via ruby SizedQueue
- input/worker/output threads, can be configured
- each input is one thread, unless explicitly configurable
- one worker thread by default, use -w to change
- output is a single thread (some outputs have their own queueing thread)

http://logstash.net/docs/1.3.3/life-of-an-event
Data growth & capacity planning

![Graph showing linear growth of data over time.](image-url)
Data growth & capacity planning

No!
Data growth
Data growth & capacity planning

Graph showing data growth over time with arrows indicating increases in data.
Data growth & capacity planning

• Added a new forwarder/shipper
• Added new type of logs
• Increased traffic/usage

• Capacity planning?
Capacity management

- capacity of one node

data

time
Scale data to your needs!

- Small dataset
- Fits on one machine, cannot be divided
Scale data to your needs!

- More data gets indexed
- Can be scaled on up to eight machines
Scale data to your needs!

- Safety: Data available twice in cluster
- Can be scaled on up to 62 machines

per day
Scale data to your needs!

- **logs-2014-01**
  - 1

- **logs-2014-02-w01**
  - 1
  - 2

- **logs-2014-03-01**
  - 1
  - 1

- **logs-2014-02-w04**
  - 1
  - 2

- **logs-2014-03-31**
  - 1
  - 1

- per month
- per week
- per day
Kibana
Kibana

Micro Analysis of extension

525 events in the table set

```plaintext
<table>
<thead>
<tr>
<th>field</th>
<th>action</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>extension</td>
<td>html</td>
<td>230</td>
</tr>
<tr>
<td></td>
<td>php</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td>png</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>gif</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>css</td>
<td>28</td>
</tr>
<tr>
<td>@message</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>@tags</td>
<td></td>
</tr>
<tr>
<td></td>
<td>@timestamp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>agent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>clientip</td>
<td></td>
</tr>
<tr>
<td></td>
<td>country</td>
<td></td>
</tr>
<tr>
<td></td>
<td>extension</td>
<td></td>
</tr>
<tr>
<td></td>
<td>geo_coordinates</td>
<td></td>
</tr>
<tr>
<td></td>
<td>geo_country</td>
<td></td>
</tr>
<tr>
<td></td>
<td>id</td>
<td></td>
</tr>
<tr>
<td></td>
<td>index</td>
<td></td>
</tr>
<tr>
<td></td>
<td>phpmemory</td>
<td></td>
</tr>
<tr>
<td></td>
<td>request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>response</td>
<td></td>
</tr>
<tr>
<td></td>
<td>type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>utc_time</td>
<td></td>
</tr>
</tbody>
</table>
```

@message (100%), @tags (100%), @timestamp (100%), agent (100%), bytes (100%), clientip (100%), geo_coordinates (100%), geo_country (100%), id (100%), index (100%), phpmemory (25%), request (100%), response (100%), type (100%), utc_time (100%).

Copyright Elasticsearch 2014. Copying, publishing and/or distributing without written permission is strictly prohibited
Kibana
Kibana
Tools
Useful helpers

- Curator
  
  http://www.elasticsearch.org/blog/curator-tending-your-time-series-indices/

- Puppet module
  
  https://github.com/elasticsearch/puppet-logstash

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

- Logstash cookbook
  
  http://cookbook.logstash.net/
Demo - Meetup RSVP stream
Soon... 1.4

• tons of documentation updates
• puppet module love
• tests to ensure backwards compatibility
• new packaging (less startup time)
Thanks for listening
Q & A

P.S. We’re hiring
http://elasticsearch.com/about/jobs
http://elasticsearch.com/support

Alexander Reelsen
@spinscale
alexander.reelsen@elasticsearch.com