The Elastic Stack
Log management and analysis
About this presentation

- What is log analysis?
- Why do I need it?
- The Elastic stack
- Practical example
- Alternatives
About me

• SUSE veteran
• Open source veteran
  (Google knows it all)
• Father, Maker, Hacker
• Agilist
• Currently: Product Owner for SUSE Container platform offering
Log Analysis
Problem statement

- Many daemons
- Large stacks
- Distributed
- Huge amounts of data
- Hard to read
Problem statement (2)

- Central logging
- Safe, Tamper-resistant
- Dependency, Causality
- One-offs vs. Trends
- Text vs. Graphic
The Elastic stack
(formerly: ELK stack)
Beats
Beats

- Formerly ‘logstash-forwarder’
- Unobtrusive (log) file forwarder
  - ‘tail -f | tee’
- Written in Go, fast
- Simple configuration
Beats

- Filebeat
- Metricbeat
- Packetbeat
- Heartbeat
- Auditbeat
- Winlogbeat
- Functionbeat
Example: Apache + Filebeat

# /usr/filebeat/filebeat.yml

filebeat:
  prospectors:
    -
      paths:
        - /var/log/apache2/access_log
      encoding: utf-8
      input_type: log
      **document_type**: access_log
      ...

output:
  logstash:
    hosts: ["logstash.mgr.suse.de:5045"]
Elasticsearch
Elasticsearch

- Fulltext database (Apache Lucene)
- Key-Value pairs
- Scalable

Terminology
- Index: Database
- Mapping: Schema
- Document: Record
- Field: key-value pair
Elasticsearch – raw data

```
message: 2016/06/27 10:35:00 +02:00 16079 0.0.0.0: osad/jabber_lib._orig_dispatch(<jabber.xmlstream.Node instance at 0x16c95a8>,) @version: 1 @timestamp: June 27th 2016, 10:35:00.000 type: osa-dispatcher tags: elasticsupport, osa-dispatcher pid: 16,079 clientip: 0.0.0.0 module: osad function: jabber_lib._orig_dispatch _id: AVYDVrc9PQbCb400Xg0k _type: osa-dispatcher _index: laszis100_160627_1033 _score:
```

```
message: 10.10.191.100 -- [27/Jun/2016:10:35:00 +0200] "POST /cobbler_api HTTP/1.1" 200 129 "-" "Java/1.7.0" @version: 1 @timestamp: June 27th 2016, 10:35:00.000 type: access_log tags: elasticsupport, access_log clientip: 10.10.191.100 ident: - auth: - verb: POST request: /cobbler_api httpversion: 1.1 response: 200 bytes: 129 referer: "-" agent: "Java/1.7.0" _id VYDNckpP0bCb4000wXX _type: access_log _index: laszis100_160627_1033 _score:
```
Elasticsearch – Kibana fields

message: 2016/06/27 10:35:00 +02:00 16079 0.0.0.0: osad/jabber_lib._orig_dispatch(<jabber.xml stream.Node instance at 0x16c95a8>,) @version: 1 @timestamp: June 27th 2016, 10:35:00.000

message: 10.10.191.100 -- [27/Jun/2016:10:35:00 +0200] "POST /cobbler_api HTTP/1.1" 200 129
"-" "Java/1.7.0" @version: 1 @timestamp: June 27th 2016, 10:35:00.000
type: access_log tags: lasticsupport, access_log clientip: 10.10.191.100 ident: - auth: - verb: POST request: /cobbler_api httpversion: 1.1 response: 200 bytes: 129 referer: "-" agent: "Java/1.7.0" _id VYDNckpPbCb4000wX _type: access_log _index: laszis100_160627_1033 _score:
Elasticsearch – internal fields

message: 2016/06/27 10:35:00 +02:00 16079 0.0.0.0: osad/jabber_lib._orig_dispatch(<jabber.xmlstream.Node instance at 0x16c95a8>,) @version: 1 @timestamp: June 27th 2016, 10:35:00.000
type: osa-dispatcher tags: elasticsupport, osa-dispatcher pid: 16,079 clientip: 0.0.0.0 module: osad function: jabber_lib._orig_dispatch _id: AVYDVrc9PQbCb400Xg0k _type: osa-dispatcher index: laszis100_160627_1033 score:

message: 10.10.191.100 -- [27/Jun/2016:10:35:00 +0200] "POST /cobbler_api HTTP/1.1" 200 129 "-" "Java/1.7.0" @version: 1 @timestamp: June 27th 2016, 10:35:00.000 type: access_log tags: elasticsupport, access_log clientip: 10.10.191.100 ident: - auth: - verb: POST request: /cobbler_api httpversion: 1.1 response: 200 bytes: 129 referer: "-" agent: "Java/1.7.0" _id VYDNckpP0bCb4000wxX _type: access_log index: laszis100_160627_1033 score:
Logstash
Logstash - Overview

- Logserver
- Scalable
- Time-based events
- JRuby

- Input: Text or JSON
- Filter: Parse and manipulate
- Output: Elasticsearch or other
input {
    stdin {}
}

Logstash - input.conf
input {
  tcp {
    port => 9000
    type => "access_log"
  }
  tcp {
    port => 9001
    type => "error_log"
    tags => ["tag1", "tag2"]
  }
}
Logstash Grok Patterns
# osa-dispatcher
# 2015/06/12 11:39:04 +02:00 14117 0.0.0.0: osad/jabber_lib.main('ERROR',...)

filter {
  if ([type] == "osa-dispatcher") {
    grok {
      match => {
        "message" => "\d\d\d\d/\d\d/\d\d \d\d:\d\d:\d\d \[+-\]\d\d:\d\d:timestamp ..."
      }
    }
  }
}
Logstash - filter.pattern

# osa-dispatcher
# 2015/06/12 11:39:04 +02:00 14117 0.0.0.0: osad/jabber_lib.main('ERROR',...)

TIMESTAMP \d\d\d\d/\d\d/\d\d/\d\d/\d\d/\d\d/\d\d/s[]\d\d/\d\d/s[+-]/\d\d/\d\d
PID [\d]+
FUNCTION [\w_]+
ARGS \([^\]\)]+\)
Logstash - filter.conf

```ruby
filter {
    if ([type] == "osa-dispatcher") {
        grok {
            match => {
                "message" => "\%{TIMESTAMP:timestamp} \%{PID:pid:int} \%{IPV4:clientip}: ..."
            }
        }
    }
}
```
Logstash - output.conf

output {
  stdout { codec => rubydebug }
}

```ruby
output {
  stdout { codec => rubydebug }
}
```
output { 
  elasticsearch { 
    hosts => ["localhost:9200"]
  }
}

Logstash - output.conf
Logstash - Start

# ls
filter.conf  input.conf  osa-dispatcher.pattern  output.conf  rhn_web_api.pattern

# logstash -f ./* .conf --auto-reload
Kibana
Kibana - Overview

- Web based visualization frontend for Elasticsearch
- Time-based events
- Comfortable query interface
- Dashboard management

- Settings
- Discover
- Visualize
- Dashboard
Kibana - Settings

• Select index pattern
  • wildcards possible

• Time based ?
  • Time-field name

Loads mapping
• field names
• field types
• analyzed ?
Kibana - Discover

• No results found?
  • Expand your time range
• Explore fields
  • Include/Exclude
• Create query
• Save search
• Visualize!
Kibana - Visualize

- Create new
  - Select visualization type
- New/Saved search
- Graph-specific parameters
Kibana - Dashboard

- Visualize Elasticsearch fields
- Collection of visualization tiles
- Table, Graph, Map, ...
- Can be saved/shared
Help !
Help!

- https://github.com/elastic is very active
- Packaging is complex
- Java, JRuby, Go, JavaScript … oh my!

security:logging on build.opensuse.org
Other resources

- https://github.com/SUSE/log-analysis
- Dockerfiles
- Salt states
- Grok patterns

Contributions welcome!
Join Us at www.opensuse.org
License
This slide deck is licensed under the Creative Commons Attribution-ShareAlike 4.0 International license. It can be shared and adapted for any purpose (even commercially) as long as Attribution is given and any derivative work is distributed under the same license.

Details can be found at https://creativecommons.org/licenses/by-sa/4.0/

General Disclaimer
This document is not to be construed as a promise by any participating organisation to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. openSUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for openSUSE products remains at the sole discretion of openSUSE. Further, openSUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All openSUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC, in the United States and other countries. All third-party trademarks are the property of their respective owners.

Credits
Template
Richard Brown
rbrown@opensuse.org

Design & Inspiration
openSUSE Design Team
http://opensuse.github.io/branding-guidelines/