

# ElasticSearch

Log everything !

<https://www.elastic.co/>  
<https://www.elastic.co/learn>

# ElasticSearch

Do you know your systems, applications, environment ?

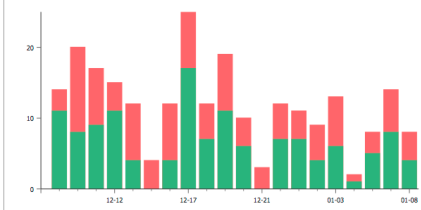
## BUILD DASHBOARD

Name	Last Build	Recent
master-alm-continuous-gul-test	FAILURE	10 / 20
backward-compatibility-of-migrations	FAILURE	9 / 20
appack	SUCCESS	11 / 20
master-faky-finder-continuous	SUCCESS	20 / 20
master-alm-continuous-js-chrome	SUCCESS	20 / 20
master-alm-continuous-js-firefox	SUCCESS	20 / 20
alm	FAILURE	9 / 20
master-alm-continuous-jav	FAILURE	17 / 20
on-demand-java	FAILURE	0 / 2
alm-continuous	SUCCESS	2 / 4
on-demand-alm-continuous-jav	SUCCESS	2 / 10
on-demand-deploy	FAILURE	0 / 1
533179-alm-continuous	FAILURE	No Recent Builds
532843-alm-continuous	SUCCESS	No Recent Builds

master-alm-continuous-gul-test 130 / 240

Last 30 days Last 90 days By Release

Success Failure



Build #	Date	Duration	Status
8822	2014-01-08T17:02:37.570Z	1479.412	FAILURE
8821	2014-01-08T16:30:31.583Z	1516.76	SUCCESS
8820	2014-01-08T14:34:14.255Z	1530.884	SUCCESS
8819	2014-01-08T14:08:04.419Z	1494.361	FAILURE
8818	2014-01-08T05:01:48.885Z	1515.575	SUCCESS

adfoc.us Link shortening service with a twist. Shrink, ul

## WHOOPSIE!

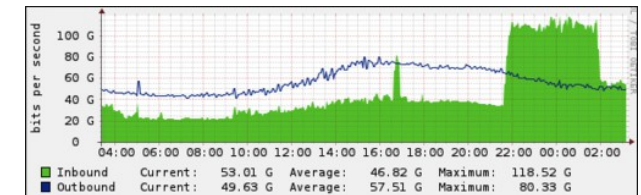
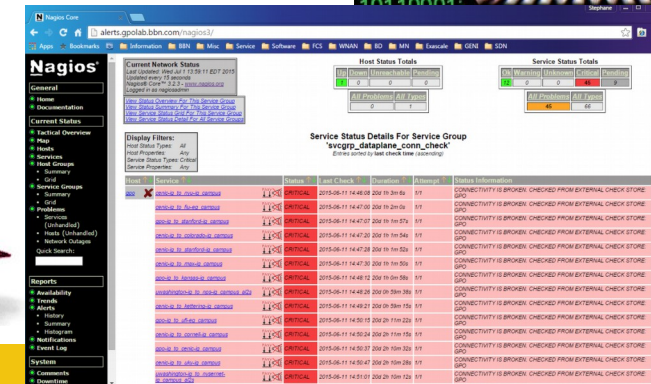
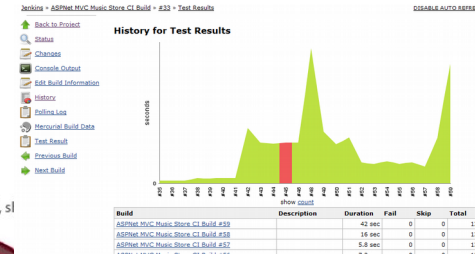
Our servers are experiencing an overload in connections!

[CLICK HERE TO REFRESH THE PAGE](#)  
(You might have to try a few times)

We apologize for this inconvenience.



Error establishing a database connection



# ELK Bio


- Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.
- Elasticsearch is developed in Java and is released as open source under the terms of the Apache License. Elasticsearch is the most popular enterprise search engine followed by Apache Solr, also based on Lucene.
- First release 2010-02-08
- Latest version 5.1

# Overview

Elasticsearch can be used to search all kinds of documents. It provides scalable search, has near real-time search, and supports multitenancy. Elasticsearch is **distributed**, which means that indices can be divided into shards and each shard can have zero or more replicas.

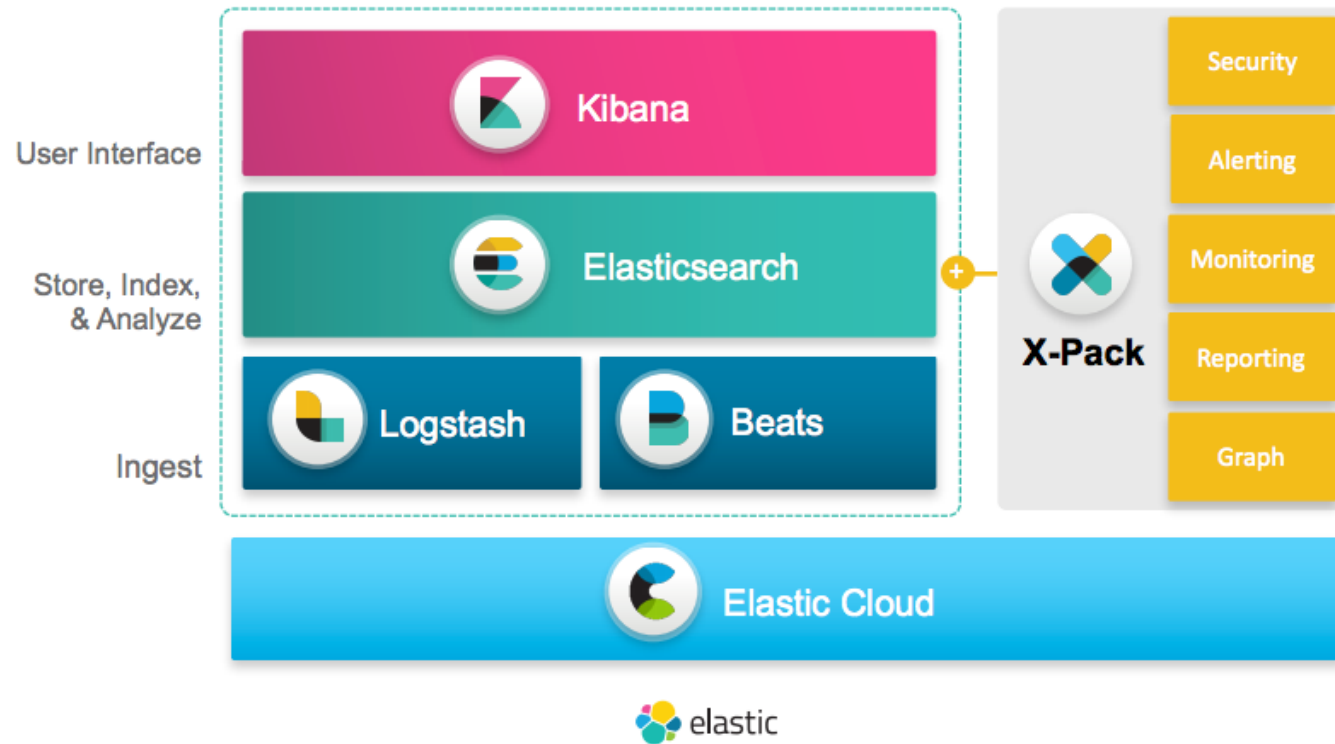
Each node hosts one or more shards, and acts as a coordinator to delegate operations to the correct shard(s). Rebalancing and routing are done **automatically**.

Related data is often stored in the same index, which consists of one or more primary shards, and zero or more replica shards. Once an index has been created, the number of primary shards **cannot be changed**.

 **elastic** search uses Lucene and tries to make all its features available through the JSON and Java API. It supports facetting and

# Architecture

## Elastic Stack



# Log flow

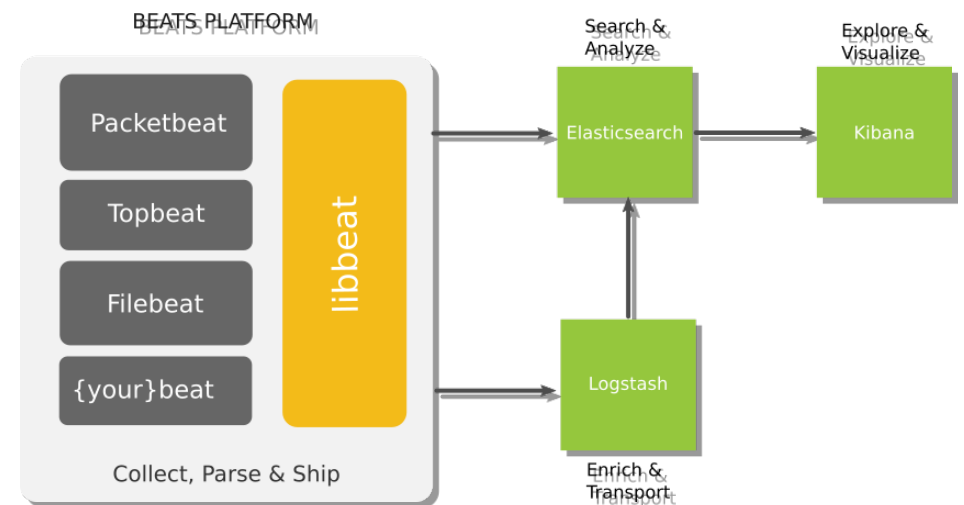
- Beats and Logstash can be replaced or enriched by variety of tools, like Fluent, RabbitMQ/Kafka, Flume, ...
- This presentation describe official ELK stack components



# Log flow

## Beats

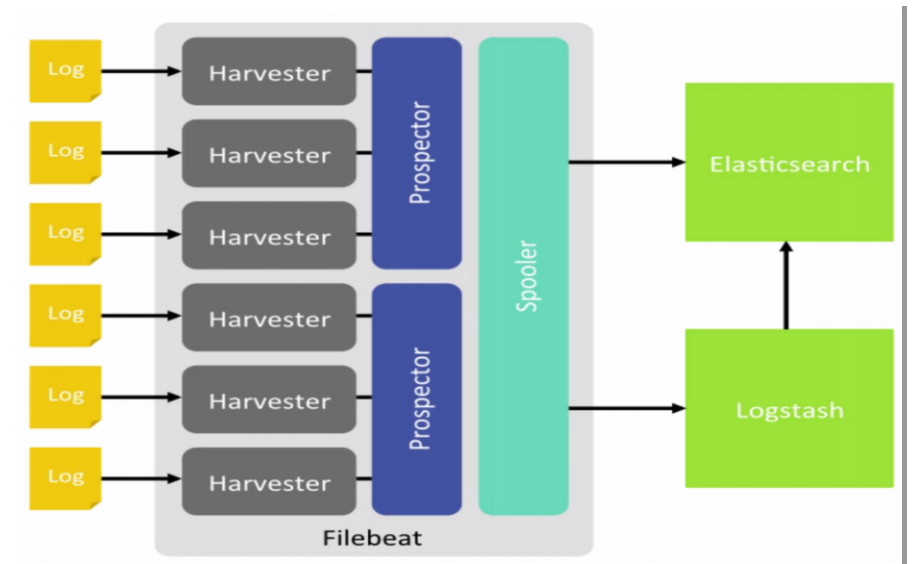
- Beats are lightweight shippers for (log) data
  - Packetbeats for analysing complex distributed applications and troubleshooting
  - Topbeats for shipping resource utilization metrics
  - Filebeats for shipping log files
  - Community beats – httpbeat, pingbeat, apachebeat, dockerbeat, nginxbeat, uwsgibeat, phpfpmbat, ...



# Log flow

FileBeat

- Filebeat properties
  - Send at least once by confirmation
  - Handles log rotation
  - Last reading state in case you restart your system of LogStash is not reachable  
=> upon revive it will send all missing logs
  - By default send new log lines every 10seconds





# Log flow

Logstash

- LogStash functional flow
  - Inputs: beats, syslog, stdin, S3, Redis, Kafka, ...
  - Filters: using GROK (regex templating)
  - Outputs: ElasticSearch, eMail, exec, Redis, Kafka, Zabbix, ...

```
input { stdin { } }

filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
  date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
}

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



# Log flow

Logstash, Why grok ?

- <http://grokconstructor.appspot.com/groklib/grok-patterns>

**Compare these two entries:**

**Grok**

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

vs

**Regexp**

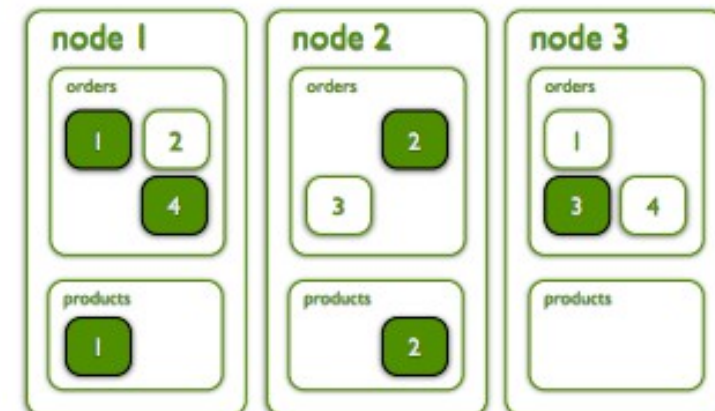
```
^([0-9.]+)s([w.-]+)s([w.-]+)s([^[ ]+))s"((?:[ ^"]|")+)s"(d{3})s(d+|-)s"((?:[ ^"]|")+)s"((?:[ ^"]|")+)s"((?:[ ^"]|")+)s"$
```

```
input {  
  tcp {  
    port => 5000  
    codec => json  
  }  
}  
  
filter {  
  mutate {  
    add_field => [ "[geoip][coordinates]", "group.group_lon" ]  
    add_field => [ "[geoip][coordinates]", "group.group_lat" ]  
    convert => [ "[geoip][coordinates]", "float" ]  
  }  
}  
  
output {  
  elasticsearch {  
    hosts => "elasticsearch:9200"  
    user => elastic  
    password => changeme  
    template => "/etc/logstash/templates/geoip.json"  
  }  
}
```

# Log flow

## ElasticSearch

- **Cluster** – An Elasticsearch cluster consists of one or more nodes and is identifiable by its cluster name. *COLLECTION OF NODES*
- **Node** – A single Elasticsearch instance. In most environments, each node runs on a separate box or virtual machine/container
  - **node.data** - node take care about data storage (datas physicaly stored on them)
  - **node.master** - node is part of cluster logic, based on quorum these nodes take care about data routing, rebalancing, replicas placement, ...
  - **node.gateway** - not a part of cluster quorum, not store datas, but do routing of requests to data nodes and caches for results
  - **node.ingest** – for pre-process documents before the actual indexing takes place. Ingest node intercepts bulk and index requests, applies the transformations, and then passes the documents back to the index or bulk APIs.



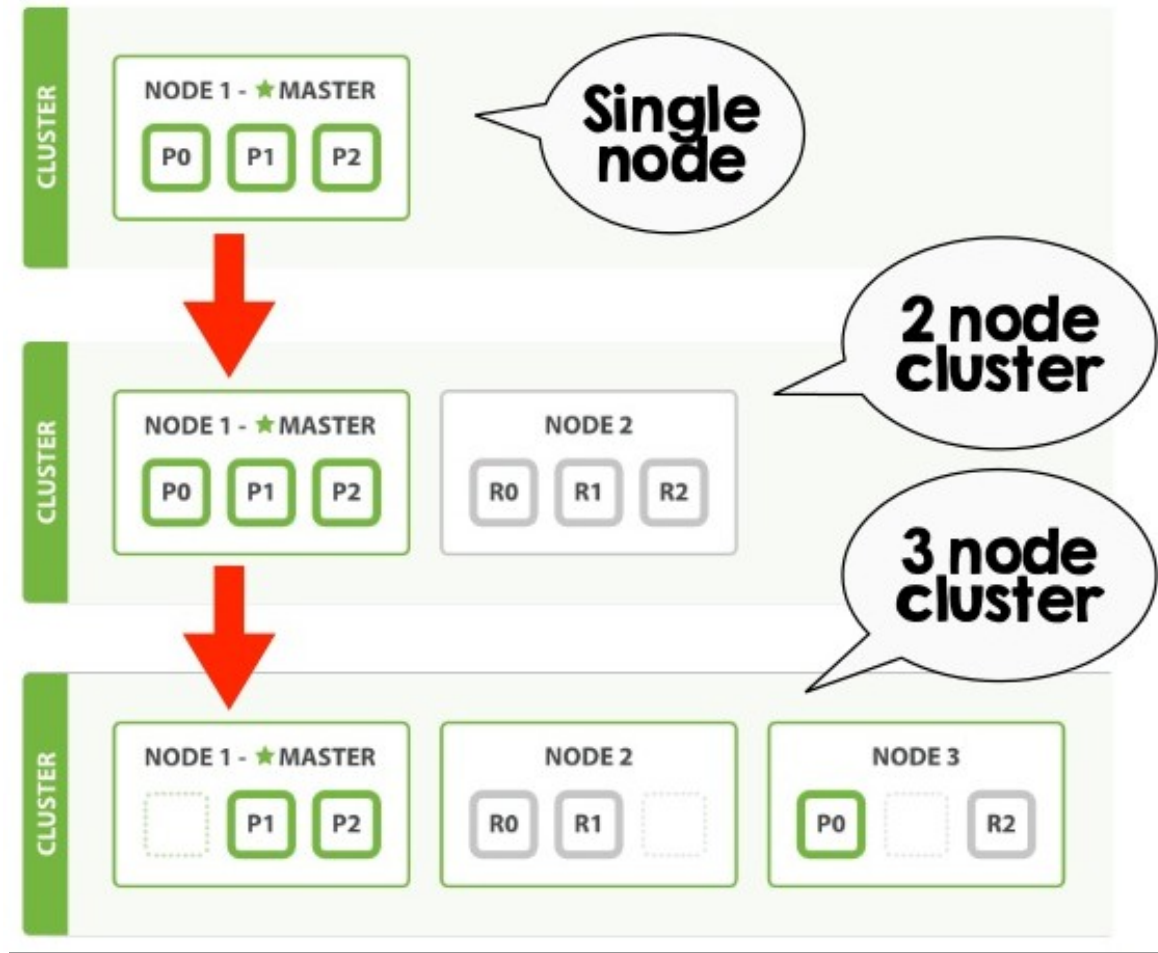
# ElasticSearch

## Data management

- **Index** – In Elasticsearch, an index is a collection of documents. Two possible types:
  - time-based - events
  - timeless
- **Shard** – Because Elasticsearch is a distributed search engine, an index is usually split into elements known as shards that are distributed across multiple nodes. Elasticsearch automatically manages the arrangement of these shards. It also rebalances the shards as necessary, so users need not worry about the details.
- **Replica** – By default, Elasticsearch creates five primary shards and one replica for each index. This means that each index will consist of five **primary** shards, and each shard will have one copy - **replica**.
- Allocating multiple shards and replicas is the essence of the design for distributed search capability, providing for high availability and quick access in searches against the documents within an index. The main difference between a primary and a replica shard is that only the primary shard can accept indexing requests. Both replica and primary shards can serve querying requests.

# ElasticSearch

Data management



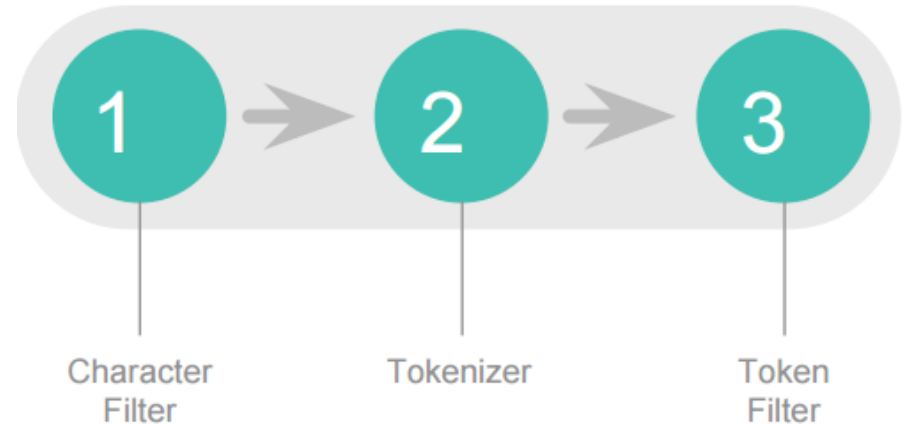
# ElasticSearch

## Retention policy

- Maintain by Kibana (for Marvel and other cluster data), for other indexes this can be done manually/scripted or use Curator tool.
- Example setup:
  - Current replicas - to speed up search on stronger boxes
  - Week old snapshot - keep only 1 replica
  - Month old - move to weaker boxes
  - Two months - close the indices
  - Three months - delete
- Backups/snapshots can be create on:
  - **Local** filesystem (accessible by all ELK servers) – glusterfs can be used
  - **S3** – experimental howto is included in workshop, using Riak CS instead AWS S3
  - **HDFS**
  - **Google** Cloud Storage
  - **Azure**

# Stuff a Search Engine Can Do

- Document Analysis
- Indexing
- Searching and Ranking



# Document Analysis

- <https://www.elastic.co/guide/en/elasticsearch/reference/current/analyzer-anatomy.html>
- Character filter
  - receives the original text as a stream of characters and can transform the stream by adding, removing, or changing characters. For instance, a character filter could be used to convert Hindu-Arabic numerals (٠١٢٣٤٥٦٧٨٩) into their Arabic-Latin equivalents (0123456789), or to strip HTML elements like <b> from the stream
- Tokenizer
  - receives a stream of characters, breaks it up into individual *tokens* (usually individual words), and outputs a stream of *tokens*
- Token filters
  - receives the token stream and may add, remove, or change tokens. For





# Indexing

- Elasticsearch terms:
  - An Index: data structure that houses documents (think RDBMS "table")
  - Index a document: insert into an Index
  - Document: a JSON object (hash map)

```
$ curl -XPUT 'http://localhost:9200/twitter/tweet/1' -d '{  
  "user" : "kimchy",  
  "post_date" : "2009-11-15T14:12:12",  
  "message" : "trying out Elasticsearch"  
}'
```

# Indexing

# document id 1 {"text": "He who controls the spice, controls the universe."}

token	document_id	frequency
He	1	1
who	1	1
controls	1	1
the	1	1
spice	1	1
universe	1	1

# Indexing

- # document id 1 {"text": "He who controls the spice, controls the universe."}
- # document id 2 {"text": "A mad man sees what he sees."}

token	document_id	frequency
He	1	1
who	1	1
controls	1	1
the	1	1
spice	1	1
universe	1	1
A	2	1
mad	2	1
man	2	1
sees	2	1
what	2	1
he	2	1

# Indexing

# document id 1 {"text": "He who controls the spice, controls the universe."}

# document id 2 {"text": "A mad man sees what he sees."}

# document id 3 {"text": "What if a mad man controlled the universe?"}

token	document_id	frequency
He	1	1
who	1	1
controls	1	1
<b>the</b>	<b>1,3</b>	<b>2</b>
spice	1	1
<b>universe</b>	<b>1,3</b>	<b>2</b>
A	2	1
<b>mad</b>	<b>2,3</b>	<b>2</b>
<b>man</b>	<b>2,3</b>	<b>2</b>
sees	2	1
what	2	1
he	2	1
What	3	1
if	3	1
a	3	1
controlled	3	1

# Indexing

Lower case token filter

# document id 1 {"text": "He who controls the spice, controls the universe."}

# document id 2 {"text": "A mad man sees what he sees."}

# document id 3 {"text": "What if a mad man controlled the universe?"}

token	document_id	frequency
he	1,2	2
who	1	1
controls	1	1
the	1,3	2
spice	1	1
universe	1,3	2
a	2,3	2
mad	2,3	2
man	2,3	2
sees	2	1
what	2,3	2
if	3	1
controlled	3	1

# Indexing

## Stopwords

# document id 1 {"text": "He who controls the spice, controls the universe."}

# document id 2 {"text": "A mad man sees what he sees."}

# document id 3 {"text": "What if a mad man controlled the universe?"}

token	document_id	frequency
he	1,2	2
who	1	1
control	1,3	2
the	1,3	2
spice	1	1
univers	1,3	2
a	2,3	2
mad	2,3	2
man	2,3	2
see	2	1
what	2,3	2
if	3	1

# Searching and Ranking

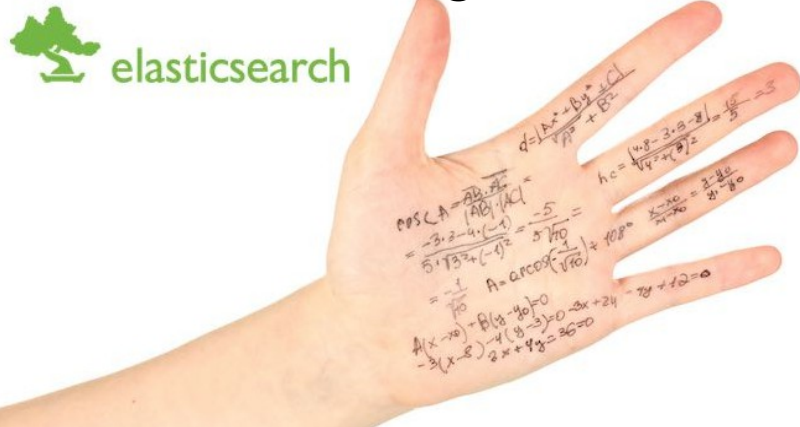
```
GET my_index/_search
{
  "query": {
    "match": {
      "text": {
        "query": "control spice"
      }
    }
  }
}
```

token
control
spice

```
"hits": [
  {
    "_index": "my_index",
    "_type": "my_type",
    "_id": "AVlPsErba2LPPIT1i",
    "_score": 0.6434033,
    "_source": {
      "text": "He who controls the spice, controls the universe."
    }
  },
  {
    "_index": "my_index",
    "_type": "my_type",
    "_id": "AVlPsFsRa2LPPIT1k",
    "_score": 0.2824934,
    "_source": {
      "text": "What if a mad man controlled the universe?"
    }
  }
]
```

# Searching and Ranking

- There are three main factors of a document's score:
  - **TF** (term frequency): The more a token appears in a doc, the more important it is
  - **IDF** (inverse document frequency): The more documents containing the term, the less important it is
  - **Field length**: shorter docs are more likely to be relevant than longer docs



$$\text{tfidf}_{i,j} = \text{tf}_{i,j} \times \log \left( \frac{N}{\text{df}_i} \right)$$

$\text{tf}_{i,j}$  = total number of occurrences of  $i$  in  $j$   
 $\text{df}_i$  = total number of documents (speeches) containing  $i$   
 $N$  = total number of documents (speeches)

$$\text{bm25}(D, Q) = -1 \sum_{i=1}^{n\text{Phrase}} \text{IDF}(q_i) \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \frac{|D|}{\text{avgdl}})}$$



# ElasticSearch

## Mapping

**Mapping** is the process of defining how a document, and the fields it contains, are stored and indexed. For instance, use mappings to define:

- which string fields should be treated as full text fields.
- which fields contain numbers, dates, or geolocations.
- whether the values of all fields in the document should be indexed into the catch-all `_all` field.
- the format of date values.
- whether you want analyzed strings or not (default is yes) – split string to indices/facets
- custom rules to control the mapping for dynamically added fields.

### **Dynamic** mapping

Fields and mapping types do not need to be defined before being used. Thanks to *dynamic mapping*, new mapping types and new field names will be added automatically, just by indexing a document.

### **Explicit** mappings

You know more about your data than Elasticsearch can guess, so while dynamic mapping can be useful to get started, at some point you will want to specify your own explicit mappings.

# ElasticSearch

## Mapping

**Mapping** is setup on index creation, cannot be easily changed (data reimport/remapping) on existing indexes/fields (you can add scripted fields).

- Create an index called my\_index.
- Add mapping types called user and blogpost.
- Disable the \_all meta field for the user mapping type.
- Specify fields or properties in each mapping type.
- Specify the data type and mapping for each field.

```
PUT my_index ❶
{
  "mappings": {
    "user": { ❷
      "_all": { "enabled": false }, ❸
      "properties": { ❹
        "title": { "type": "text" }, ❺
        "name": { "type": "text" }, ❻
        "age": { "type": "integer" } ❼
      }
    },
    "blogpost": { ❸
      "_all": { "enabled": false }, ❹
      "properties": { ❺
        "title": { "type": "text" }, ❻
        "body": { "type": "text" }, ❼
        "user_id": {
          "type": "keyword" ❺
        },
        "created": {
          "type": "date", ❻
          "format": "strict_date_optional_time||epoch_millis"
        }
      }
    }
  }
}
```

# ElasticSearch

## Mapping - Templates

- Add raw field which won't be analyze from any string field, do this only for indexes which match logstash\* regexp
- Order 100 (lower order being applied first, and higher orders overriding them)

```
{~
  .."template" : "logstash*",~
  .."order" : 100,~
  .."settings" : {~
    .."index.refresh_interval" : "5s"~
    ..},~
  .."mappings" : {~
    .."default" : {~
      .."_all" : {"enabled" : true, "omit_norms" : true},~
      .."dynamic_templates" : [ {~
        .."string_fields" : {~
          .."match" : "*",~
          .."match_mapping_type" : "string",~
          .."mapping" : {~
            .."type" : "string", "index" : "not_analyzed", "omit_norms" : true,~
            .."fields" : {~
              .."raw" : {"type": "string", "index" : "not_analyzed", "ignore_above" : true}~
            }~
          }~
        }~
      }~
    }~
  }~
}
```

# ElasticSearch

## Mapping - Templates

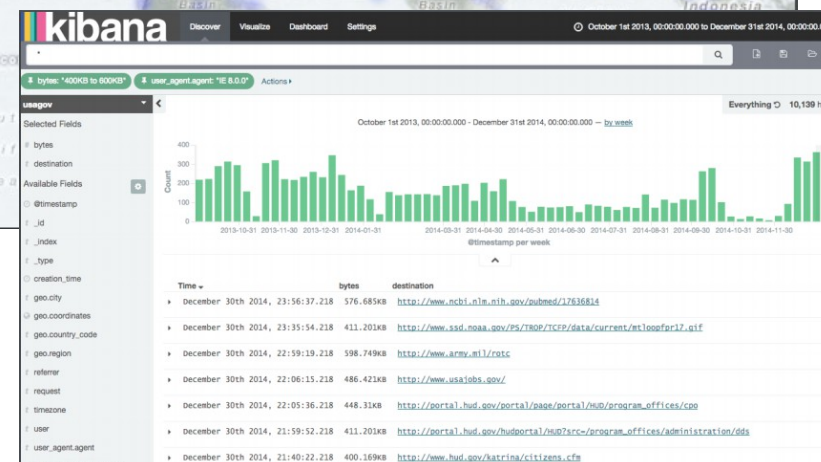
- Mappings/templates can be handled:
  - manually (curl,...) - XPUT
  - logstash/fluent/... - can also include mapping jsons and apply them on the fly (on index init)
  - config management (salt,puppet,ansible,...) - runtime or as a file (need process reload)
  - from application which directly write to Elastic

# ElasticSearch

## Challenges

- Setup and architecture complexity
- Mapping and indexing
  - Conflicts with naming
  - Log types and integration
- Capacity issues
  - Disk usage over time
  - Latency on log parsing
  - Issues with overburdened log servers
- Logging cluster health
- Cost of infrastructure and upkeep

# Kibana



# Visualization

## Kibana

Kibana lets you visualize your Elasticsearch data and navigate the Elastic Stack, so you can do anything from learning why you're getting paged at 2:00 a.m. to understanding the impact rain might have on your quarterly numbers. Kibana gives you the freedom to select the way you give shape to your data. And you don't always have to know what you're looking for. With its interactive visualizations, start with one question and see where it leads you.

### Components:

- Index management
- Discovery
- Visualization
- Dashboards
- Marvel single cluster monitoring

### Premium components\*

- ~~Timeline~~ - Time series
- ~~Graph~~ - Relationships exploration
- Generate PDF reports
- ~~Multicloud~~ monitoring
- ~~Security~~ (ACL, user management, tls, ...)

# XPack

Extra functionality

Built and maintained by Elastic engineers, X-Pack is a single extension that integrates handy features you can trust across the Elastic Stack.



## Security

*(formerly Shield)*

Protect your data across the Elastic Stack.

[Learn More](#)



## Alerting

*(via Watcher)*

Get notifications about changes in your data.

[Learn More](#)

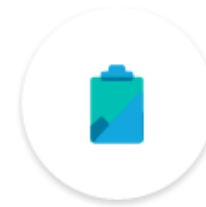


## Monitoring

*(formerly Marvel)*

Keep a pulse on the health of the Elastic Stack.

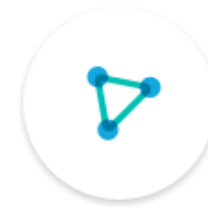
[Learn More](#)



## Reporting

Generate, schedule, and email reports.

[Learn More](#)



## Graph

Explore meaningful relationships in your data.

[Learn More](#)

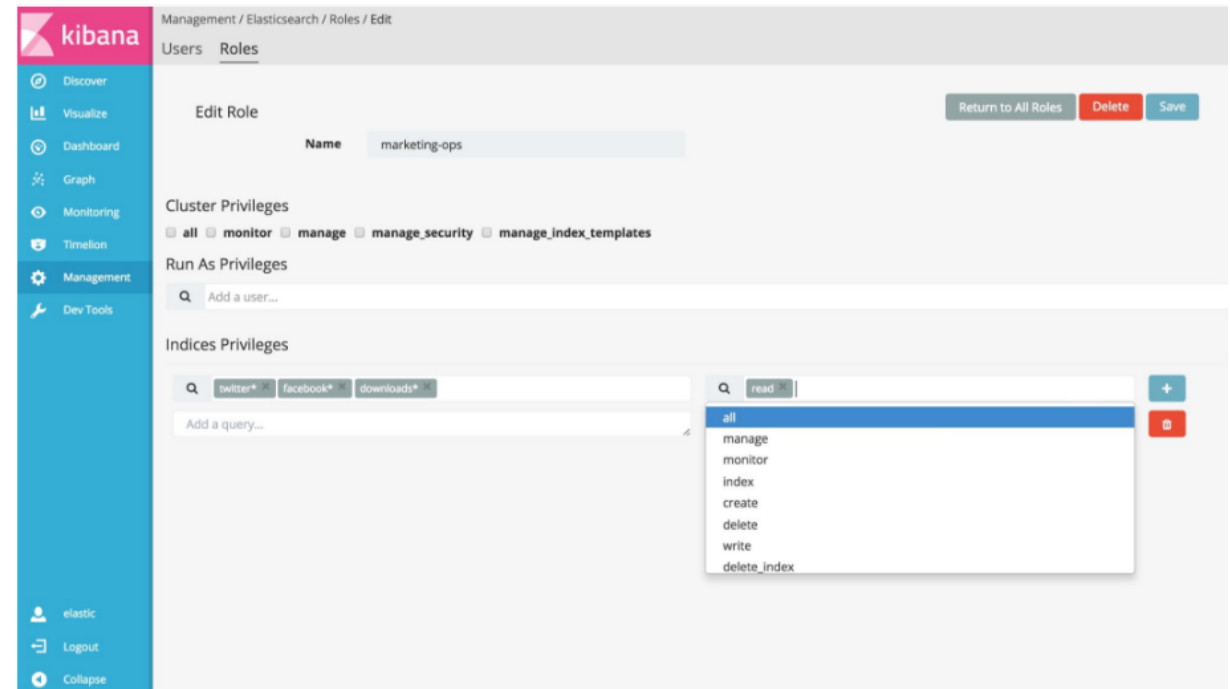
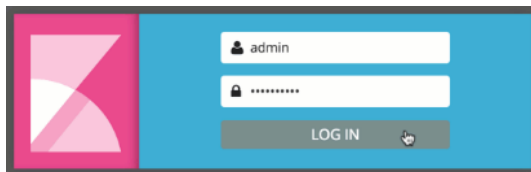


# XPack

## Security

X-Pack security features give the right access to the right people. IT, operations, and application teams rely on X-Pack to manage well-intentioned users and keep nefarious actors at bay, while executives and customers can rest easy knowing data stored in the Elastic Stack is safe and secure.

- Manage Users and Roles
- Prevent Snooping, Tampering, and Sniffing
- Secure All the Way Down to the Field Level
- Have a Record of Who Did What and When



# XPack

## Security

### Preventing unauthorized access

- To prevent unauthorized access to your Elasticsearch cluster, you must have a way to *authenticate* users.
- RBAC - also need a way to control what data users have access to and what tasks they can perform
- X-Pack security also supports IP-based authorization. You can whitelist and blacklist specific IP addresses or subnets to control network-level access to a server.

### Preserving Data Integrity

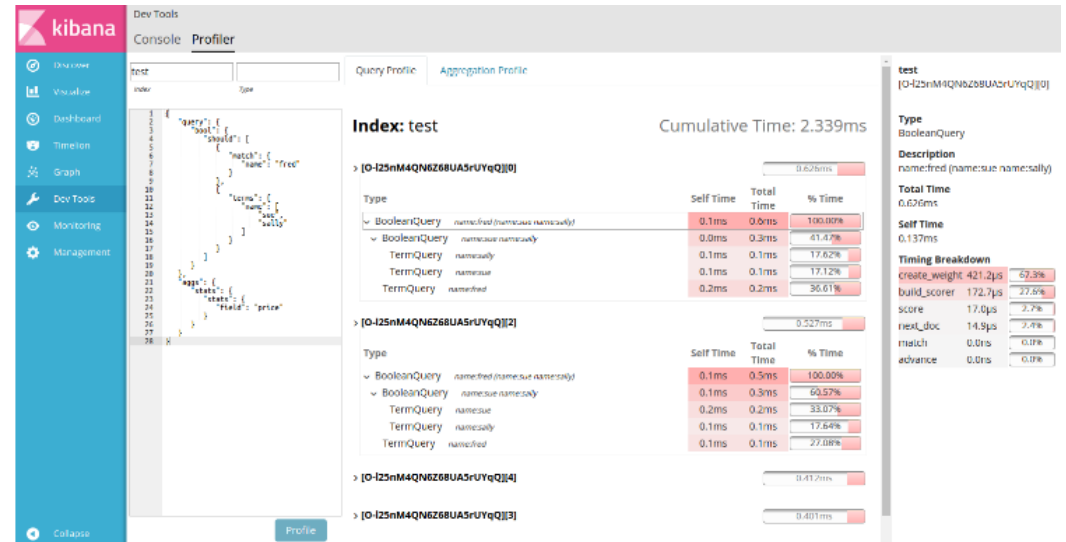
- A critical part of security is keeping confidential data confidential. Elasticsearch has built-in protections against accidental data loss and corruption. However, there's nothing to stop deliberate tampering or data interception. X-Pack security preserves the integrity of your data by encrypting communications to and from nodes and authenticating messages to verify that they have not been tampered with or corrupted in transit during node-to-node communication (via symmetric key).

# XPack

## DevTools - Profiling queries and aggregations

Elasticsearch has a powerful profiler API which can be used to inspect and analyze your search queries. The response, however, is a very large JSON blob which is difficult to analyze by hand.

**X-Pack** includes the Search Profiler tool which can transform this JSON output into a visualization that is easy to navigate, allowing you to diagnose and debug poorly performing queries much faster.



# XPack

## Watcher - Alerting

### **Detect** Changes in Your Data

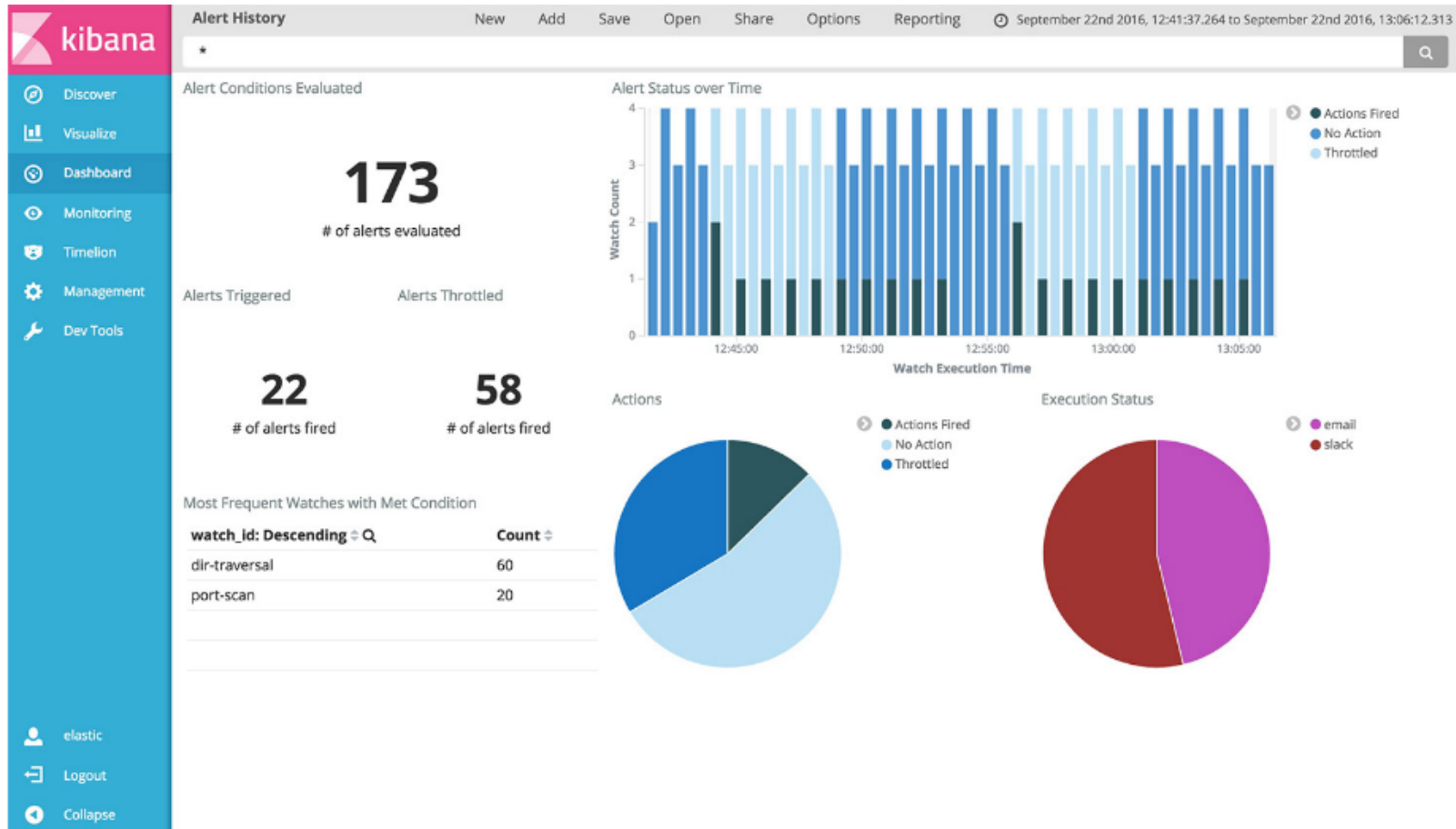
The alerting features in X-Pack give you the full power of the Elasticsearch query language to identify changes in your data that are interesting to you. In other words, if you can query something in Elasticsearch, you can alert (mail, slack, webhook call,...) on it.

For instance, you can be **notified** when:

- Intrusion - the same user logged in from 3 different locations within an hour, so you can proactively address possible intrusion attempts.
- Trending #YourProduct is trending on social media, and you need to prepare to meet the demand.
- Credit Card - card numbers are visible in your application logs and that's a compliance nightmare. It's time to talk with the application team.
- Indexing Rate - Your Elasticsearch indexing rate has plummeted due to changes in your web server log file location, so you know to update your Filebeat configuration.

# XPack

## Watcher - Alerting



# XPack

## Monitoring (Marvel)

Nodes  3 of 3

Name	Status	CPU Usage	JVM Memory	Load Average	Disk Free Space	Shards
<a href="#">81f3a447f552</a> 172.22.0.30:9300		0.67% ↓ 7.67% max 0.33% min	53% ↓ 77% max 31% min	6.84 ↑ 14.84 max 2.75 min	40.4 GB ↓ 41.1 GB max 40.2 GB min	31
<a href="#">bc1c287663b7</a> 172.22.0.20:9300		5.33% ↓ 8.33% max 0.33% min	40% ↓ 77% max 34% min	6.84 ↓ 14.37 max 2.66 min	40.4 GB ↓ 40.9 GB max 40.2 GB min	31
<a href="#">f65ef1a20dac</a> 172.22.0.31:9300		2.67% ↓ 4.33% max 2% min	64% ↓ 78% max 39% min	5.61 ↓ 14.84 max 2.66 min	40.4 GB ↓ 41.2 GB max 40.2 GB min	30

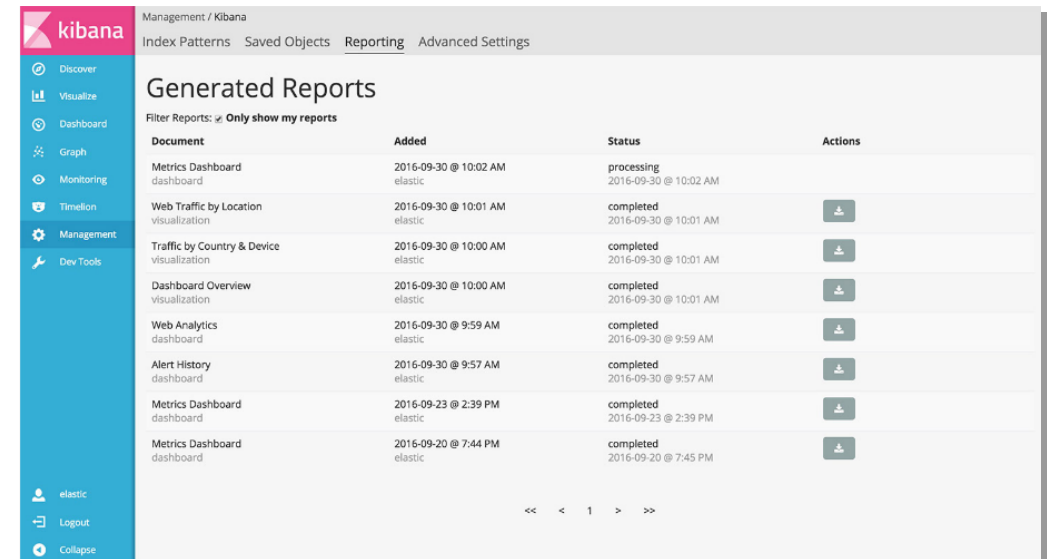


# XPack

## Report

### Generate, Schedule & Email Reports

Quickly generate reports of any Kibana visualization or dashboard. Get a report on demand, schedule it for later, trigger it based on specified conditions, and automatically share it with others — managers, customers, compliance officers. It's architected to scale and travel well, letting you take a piece of Kibana anywhere you like.



The screenshot shows the Kibana Reporting interface. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Graph, Monitoring, Timeline, Management, and Dev Tools. The main content area is titled 'Generated Reports' and includes a filter 'Only show my reports'. Below this is a table with columns: Document, Added, Status, and Actions. The table lists several reports, including 'Metrics Dashboard dashboard', 'Web Traffic by Location visualization', 'Traffic by Country & Device visualization', 'Dashboard Overview visualization', 'Web Analytics dashboard', 'Alert History dashboard', and two more 'Metrics Dashboard dashboard' entries. Each report entry shows the date and time it was generated, its status (processing or completed), and a download icon.

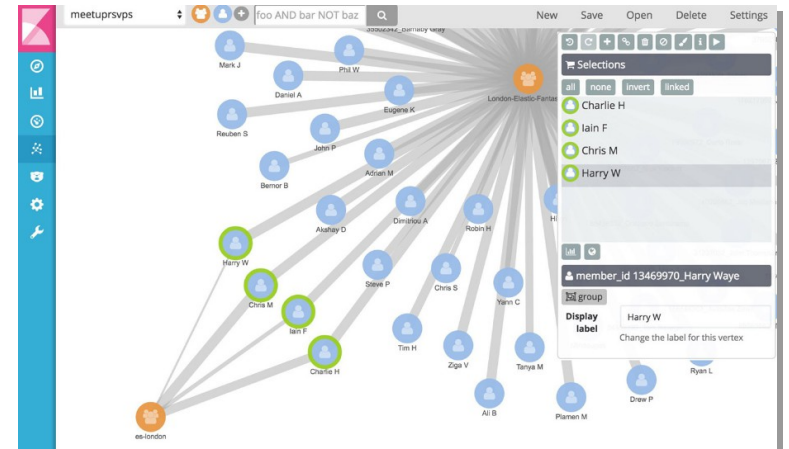
Document	Added	Status	Actions
Metrics Dashboard dashboard	2016-09-30 @ 10:02 AM elastic	processing 2016-09-30 @ 10:02 AM	
Web Traffic by Location visualization	2016-09-30 @ 10:01 AM elastic	completed 2016-09-30 @ 10:01 AM	
Traffic by Country & Device visualization	2016-09-30 @ 10:00 AM elastic	completed 2016-09-30 @ 10:01 AM	
Dashboard Overview visualization	2016-09-30 @ 10:00 AM elastic	completed 2016-09-30 @ 10:01 AM	
Web Analytics dashboard	2016-09-30 @ 9:59 AM elastic	completed 2016-09-30 @ 9:59 AM	
Alert History dashboard	2016-09-30 @ 9:57 AM elastic	completed 2016-09-30 @ 9:57 AM	
Metrics Dashboard dashboard	2016-09-23 @ 2:39 PM elastic	completed 2016-09-23 @ 2:39 PM	
Metrics Dashboard dashboard	2016-09-20 @ 7:44 PM elastic	completed 2016-09-20 @ 7:45 PM	

# XPack

## Graph

Graph is an API- and UI-driven tool that helps you surface relevant relationships in your data while leveraging Elasticsearch features like distributed query execution, real-time data availability, and indexing at any scale.

- Credit Card Fraud: Discover which vendor is responsible for a group of compromised credit cards by exploring the shops where purchases were made.
- Recommendations: Suggest the next best song for a listener who digs Mozart based on their preferences to and keep them engaged and happy.
- Intrusion Security: Identify potential bad actors and other unexpected associates by looking at external IPs that machines on your network are talking to.

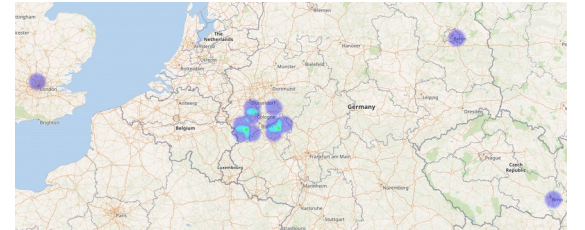







# Workshop

Contains:

- Demo stack setup via Docker Composer (local or on remote server created via Terraform)
- ELK basics (indexes, mapping, backups on fs or AWS S3) + working with API and Kibana
- Visualization (from simple graphs to geoip lon/lat tile map/heat map)
- Watcher examples (with communication to Mattermost and Jenkins)
- Crash testing
- Monitoring
- Auditing of ELK stack (Kibana discovery + dashboard)



Nodes <input type="text" value="Filter Nodes"/> 3 of 3		Status	CPU Usage		JVM Memory	Load Average
Name	IP		Usage	Max	Used	
 3dd948d614d7	172.22.0.26:9300	✓	0.33%	8% max 0.33% min	53% ↑ 78% max 32% min	1.5 ↓ 7.99 max 0.65 min
 6f8245265356	172.22.0.30:9300	✓	1.33%	7.67% max 0.33% min	73% ↑ 78% max 39% min	1.5 ↓ 7.99 max 0.65 min
 b185bbb9f54e	172.22.0.31:9300	✓	0.67%	4% max 0.33% min	69% ↑ 75% max 36% min	1.5 ↓ 7.99 max 0.65 min

<https://github.com/VAdamec/elk-stack-v5-xpack>  
and follow READMEs ...

# Credits

- <https://en.wikipedia.org/wiki/Elasticsearch>
- <https://www.elastic.co>
- <https://speakerd.s3.amazonaws.com/presentations/6eb860a1b2534deb8cc8d39cdd949898/elk.pdf>  
(Honza Kral)
- <https://speakerdeck.com/monicasarbu/unifying-logs-and-metrics-with-elastic-beats>  
(Monica Sarbu)
- <https://speakerdeck.com/monicasarbu/monitor-your-infrastructure-with-the-elastic-beats>
- <https://github.com/ianbytchek/docker-riak-cs>
- and many others ...