

# Elasticsearch, Logstash & Kibana

Kevin Kluge

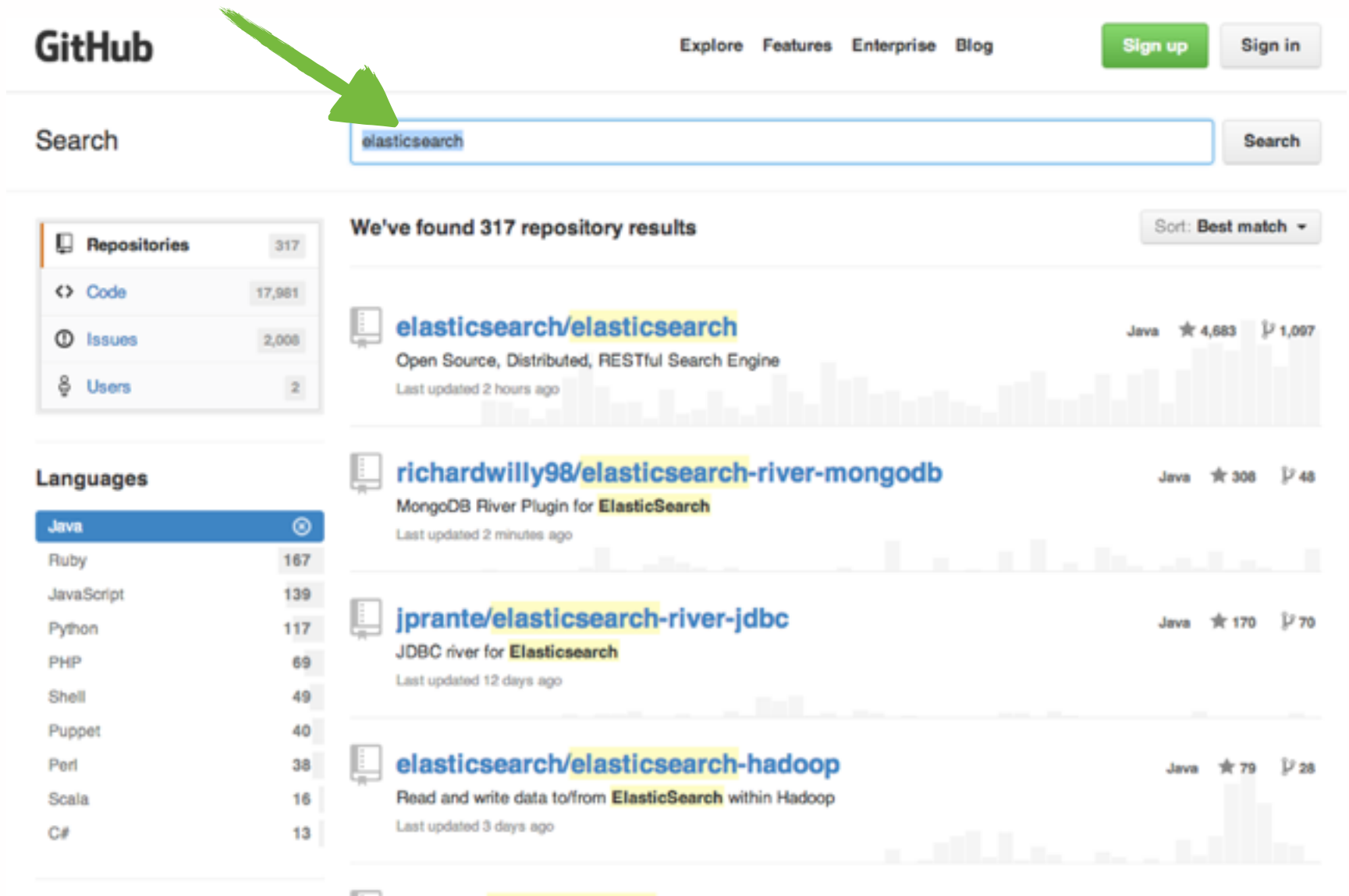
@kevinkluge

kevin.kluge@elasticsearch.com

# Elasticsearch in 10 seconds

- Schema-free, REST & JSON based document store
- Distributed and horizontally scalable
- Open Source: Apache License 2.0
- Zero configuration
- Written in Java, extensible

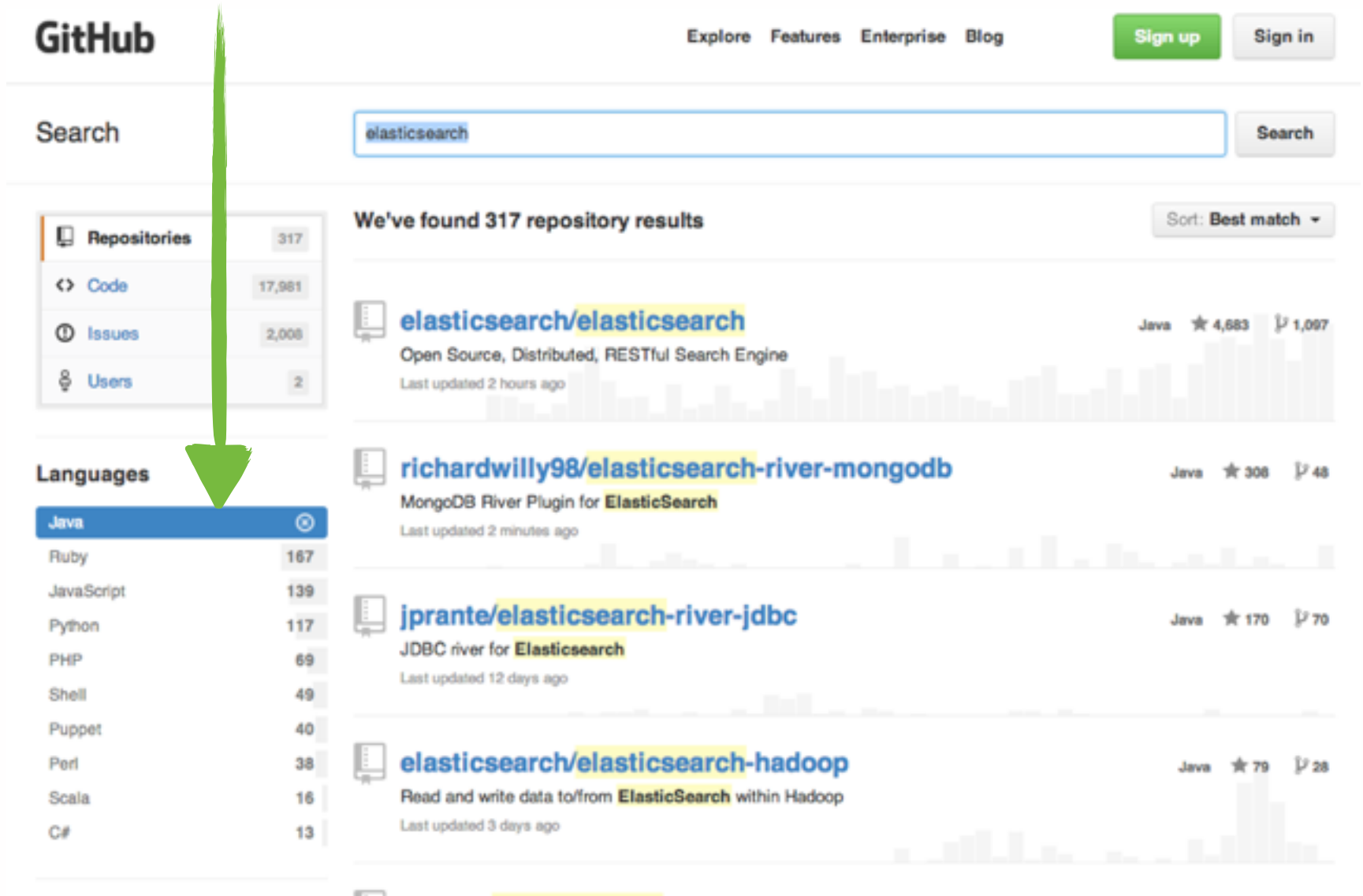
# Unstructured search



The screenshot shows the GitHub search interface. A green arrow points to the search bar which contains the text 'elasticsearch'. The search results are displayed on the right, showing 317 repository results. The left sidebar shows filters for Repositories (317), Code (17,981), Issues (2,008), and Users (2). Below the filters is a 'Languages' section with a list of languages and their counts: Java (167), Ruby (139), JavaScript (117), Python (69), PHP (49), Shell (40), Puppet (38), Perl (16), Scala (13), and C# (13). The search results list includes:

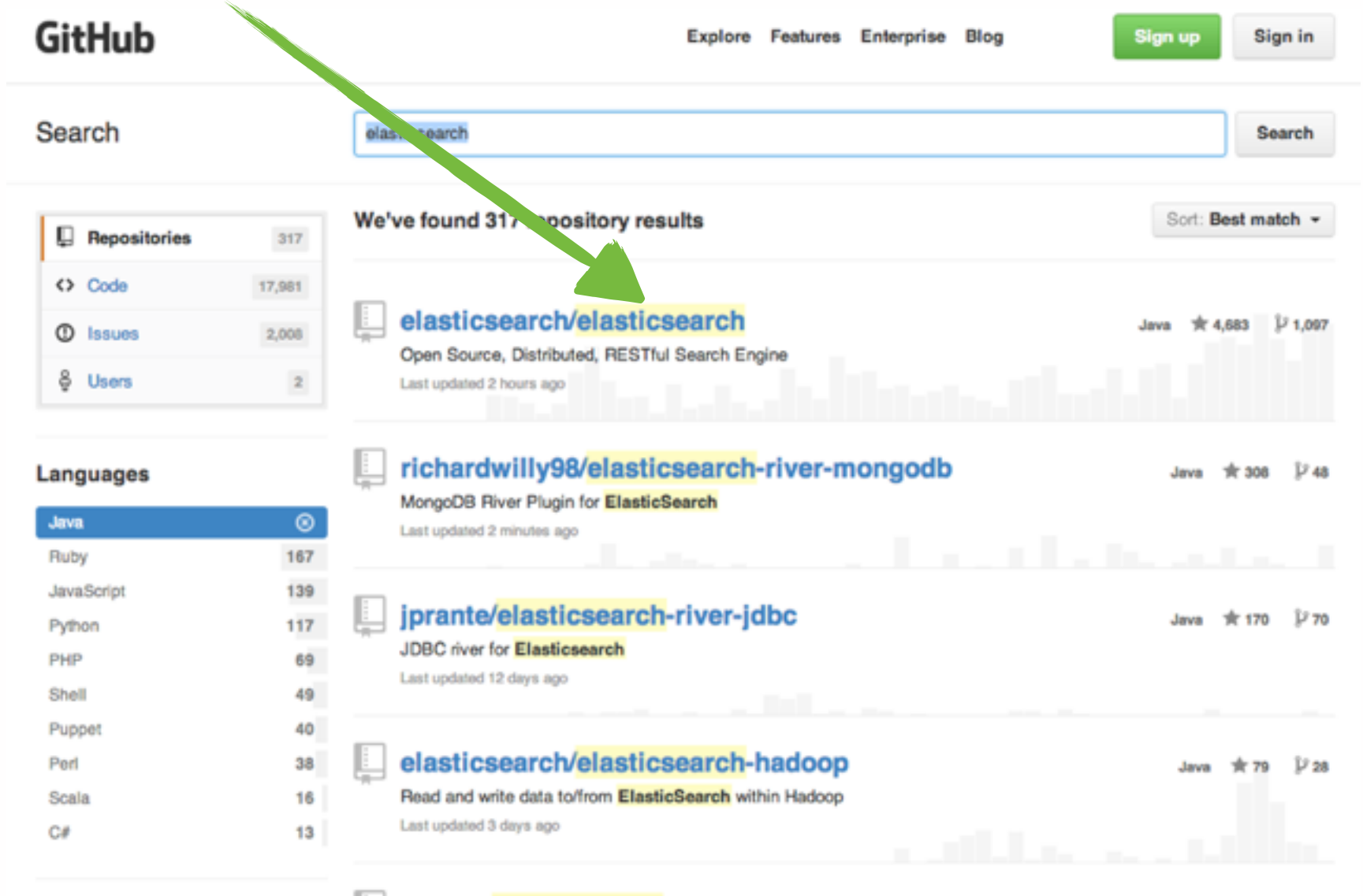
- elasticsearch/elasticsearch**: Open Source, Distributed, RESTful Search Engine. Last updated 2 hours ago. Java, 4,683 stars, 1,097 forks.
- richardwilly98/elasticsearch-river-mongodb**: MongoDB River Plugin for ElasticSearch. Last updated 2 minutes ago. Java, 308 stars, 48 forks.
- jprante/elasticsearch-river-jdbc**: JDBC river for Elasticsearch. Last updated 12 days ago. Java, 170 stars, 70 forks.
- elasticsearch/elasticsearch-hadoop**: Read and write data to/from ElasticSearch within Hadoop. Last updated 3 days ago. Java, 79 stars, 28 forks.

# Structured search



The screenshot shows the GitHub search interface. At the top, the GitHub logo is on the left, and navigation links for 'Explore', 'Features', 'Enterprise', and 'Blog' are on the right. Below the logo is a 'Search' section with a search bar containing 'elasticsearch' and a 'Search' button. To the left of the search results is a sidebar with filters. The 'Repositories' filter shows 317 results. The 'Code' filter shows 17,981 results. The 'Issues' filter shows 2,008 results. The 'Users' filter shows 2 results. Below these is a 'Languages' section with a list of languages and their counts: Java (167), Ruby (139), JavaScript (117), Python (69), PHP (49), Shell (40), Puppet (38), Perl (16), Scala (13), and C# (13). A large green arrow points from the top of the page down to the 'Java' language filter. The main search results area shows 'We've found 317 repository results' and a 'Sort: Best match' dropdown. The first four results are listed, each with a repository icon, the repository name, a description, the last updated time, and a bar chart showing the number of stars and forks. The results are: 1. **elasticsearch/elasticsearch** (Open Source, Distributed, RESTful Search Engine, Last updated 2 hours ago, 4,683 stars, 1,097 forks). 2. **richardwilly98/elasticsearch-river-mongodb** (MongoDB River Plugin for ElasticSearch, Last updated 2 minutes ago, 308 stars, 48 forks). 3. **jprante/elasticsearch-river-jdbc** (JDBC river for Elasticsearch, Last updated 12 days ago, 170 stars, 70 forks). 4. **elasticsearch/elasticsearch-hadoop** (Read and write data to/from ElasticSearch within Hadoop, Last updated 3 days ago, 79 stars, 28 forks).

# Enrichment



The screenshot shows the GitHub search interface. At the top, the GitHub logo is on the left, and navigation links for 'Explore', 'Features', 'Enterprise', and 'Blog' are on the right. Below the logo is a 'Search' section with a search bar containing 'elasticsearch' and a 'Search' button. To the left of the search results is a sidebar with filters: 'Repositories' (317), 'Code' (17,981), 'Issues' (2,008), and 'Users' (2). Below these is a 'Languages' section with a list of languages and their counts: Java (167), Ruby (139), JavaScript (117), Python (69), PHP (49), Shell (40), Puppet (38), Perl (16), Scala (13), and C# (13). The main search results area shows 'We've found 317 repository results' with a 'Sort: Best match' dropdown. The top result is 'elasticsearch/elasticsearch', described as 'Open Source, Distributed, RESTful Search Engine', with 4,683 stars and 1,097 forks. A green arrow points from the word 'Enrichment' in the title to this top result. Below it are two other results: 'richardwilly98/elasticsearch-river-mongodb' (MongoDB River Plugin for ElasticSearch) and 'jprante/elasticsearch-river-jdbc' (JDBC river for Elasticsearch). The bottom result is 'elasticsearch/elasticsearch-hadoop' (Read and write data to/from ElasticSearch within Hadoop).

GitHub

Explore Features Enterprise Blog

Sign up Sign in

Search

elasticsearch Search

We've found 317 repository results Sort: Best match

**elasticsearch/elasticsearch** Java ★ 4,683 1,097  
Open Source, Distributed, RESTful Search Engine  
Last updated 2 hours ago

**richardwilly98/elasticsearch-river-mongodb** Java ★ 308 48  
MongoDB River Plugin for ElasticSearch  
Last updated 2 minutes ago

**jprante/elasticsearch-river-jdbc** Java ★ 170 70  
JDBC river for Elasticsearch  
Last updated 12 days ago

**elasticsearch/elasticsearch-hadoop** Java ★ 79 28  
Read and write data to/from ElasticSearch within Hadoop  
Last updated 3 days ago

**Repositories** 317

**Code** 17,981

**Issues** 2,008

**Users** 2

**Languages**

Java 167

Ruby 139

JavaScript 117

Python 69

PHP 49

Shell 40

Puppet 38

Perl 16

Scala 13

C# 13

elasticsearch.

# Sorting

GitHub

Explore Features Enterprise Blog

Sign up

Sign in

Search

elasticsearch

Search

Sort: Best match ▾

Repositories 317

Code 17,981

Issues 2,008

Users 2

Languages

Java 167

Ruby 139

JavaScript 117

Python 69

PHP 49

Shell 40

Puppet 38

Perl 16

Scala 13

C#

We've found 317 repository results

**elasticsearch/elasticsearch** Java ★ 4,683 1,097  
Open Source, Distributed, RESTful Search Engine  
Last updated 2 hours ago

**richardwilly98/elasticsearch-river-mongodb** Java ★ 308 48  
MongoDB River Plugin for ElasticSearch  
Last updated 2 minutes ago

**jprante/elasticsearch-river-jdbc** Java ★ 170 70  
JDBC river for Elasticsearch  
Last updated 12 days ago

**elasticsearch/elasticsearch-hadoop** Java ★ 79 28  
Read and write data to/from ElasticSearch within Hadoop  
Last updated 3 days ago

elasticsearch.

# Pagination

The screenshot shows the GitHub search interface. At the top, the GitHub logo is on the left, and navigation links for 'Explore', 'Features', 'Enterprise', and 'Blog' are on the right. Below the navigation bar is a search bar containing the text 'elasticsearch' and a 'Search' button. To the left of the search results is a sidebar with filters: 'Repositories' (317), 'Code' (17,981), 'Issues' (2,008), and 'Users' (2). The main search results area displays 'We've found 317 repository results' and a 'Sort: Best match' dropdown. The first result is 'elasticsearch/elasticsearch', described as an 'Open Source, Distributed, RESTful Search Engine', with 4,683 stars and 1,097 forks. The second result is 'spinscale/elasticsearch-suggest-plugin', described as a 'Plugin for elasticsearch which uses the lucene FST Suggester', with 103 stars and 23 forks. At the bottom of the results, there is a pagination bar with page numbers 1 through 32, with '1' being the active page. A large green arrow points from the word 'Pagination' in the title to the pagination bar. To the right of the pagination bar is a link that says 'How are these search results? Tell us!'.

elasticsearch.

# Aggregation

GitHub

Explore Features Enterprise Blog

Sign up

Sign in

Search

elasticsearch

Search

Repositories

317

Code

7,981

Issues

1,008

Users

2

Languages

Java

Ruby

167

JavaScript

139

Python

117

PHP

69

Shell

49

Puppet

40

Perl

38

Scala

16

C#

13

We've found 317 repository results

Sort: Best match



**elasticsearch/elasticsearch**

Java ★ 4,683 1,097

Open Source, Distributed, RESTful Search Engine

Last updated 2 hours ago



**richardwilly98/elasticsearch-river-mongodb**

Java ★ 308 48

MongoDB River Plugin for ElasticSearch

Last updated 2 minutes ago



**jprante/elasticsearch-river-jdbc**

Java ★ 170 70

JDBC river for Elasticsearch

Last updated 12 days ago



**elasticsearch/elasticsearch-hadoop**

Java ★ 79 28

Read and write data to/from ElasticSearch within Hadoop

Last updated 3 days ago

elasticsearch.



# Suggestions

The screenshot shows the GitHub interface for the `elasticsearch/elasticsearch` repository. A search bar at the top contains the text `debian`. A dropdown menu is open, displaying several suggestions:

- ① `elasticsearch/elasticsearch#1726` `debian` package violates naming convention
- ① `elasticsearch/elasticsearch#3571` `debian` package init-script: start-stop-daemon ne
- 📦 `elasticsearch/elasticsearch#1681` `Debian` pkg
- ① `elasticsearch/elasticsearch#3286` There is no official `debian/ubuntu` repository
- 🔄 `elasticsearch/elasticsearch#3500` Elasticsearch should include `debian`'s standard j
- 📦 `elasticsearch/elasticsearch#1526` Moving `debian` package to maven
- Search `elasticsearch/elasticsearch` for '`debian`'
- Search GitHub for '`debian`'

The repository page also shows a sidebar with labels (Lucene 4.5 Upgrade, breaking, bug, enhancement, feature, non-issue) and a list of issues. The main content area displays a list of issues, including:

- NoShardAvailableActionException in ES 0.90.3 on startup** (#3700)
- Feature Request: Don't reindex the document when updating non-indexed fields** (#3696)

# Installation & first steps

# 2 minutes to live

```
$ wget https://download.elasticsearch.org/...  
$ tar -xf elasticsearch-1.0.0.tar.gz  
$ ./elasticsearch-1.0.0/bin/elasticsearch  
  
...  
[2014-01-19 14:53:11,508][INFO ][node] [Scanner] started  
  
...
```

Also puppet modules and RPM/DEB

# Is it alive?

```
» curl localhost:9200
{
  "status" : 200,
  "name" : "Scanner",
  "version" : {
    "number" : "1.0.0",
    "build_hash" : "e018cda7e7a32643d59e0ac3cdb412ccc239af04",
    "build_timestamp" : "2014-01-17T15:11:47Z",
    "build_snapshot" : true,
    "lucene_version" : "4.6.1"
  },
  "tagline" : "You Know, for Search"
}
```

# Create...

```
» curl -XPUT localhost:9200/books/book/1 -d '{
  "title" : "Elasticsearch – The definitive guide",
  "authors" : "Clinton Gormley",
  "started" : "2013-02-04",
  "pages" : 230
}'
```

# Update...

```
» curl -XPUT localhost:9200/books/book/1 -d '{
  "title" : "Elasticsearch – The definitive guide",
  "authors" : [ "Clinton Gormley", "Zachary Tong" ],
  "started" : "2013-02-04",
  "pages" : 230
}'
```

# Delete...

```
» curl -X DELETE localhost:9200/books/book/1
```

# Realtime GET...

```
» curl -X GET localhost:9200/books/book/1  
» curl -X GET localhost:9200/books/book/1/_source
```

# Search

```
» curl -XGET localhost:9200/books/_search?q=elasticsearch
```

```
{
  "took" : 2, "timed_out" : false,
  "_shards" : { "total" : 5, "successful" : 5, "failed" : 0 },
  "hits" : {
    "total" : 1, "max_score" : 0.076713204,
    "hits" : [ {
      "_index" : "books", "_type" : "book", "_id" : "1",
      "_score" : 0.076713204, "_source" : {
        "title" : "Elasticsearch - The definitive guide",
        "authors" : [ "Clinton Gormley", "Zachary Tong" ],
        "started" : "2013-02-04", "pages" : 230
      }
    } ]
  }
}
```



# Search - Query DSL

```
» curl -XGET 'localhost:9200/books/book/_search' -d '{
  "query": {
    "filtered" : {
      "query" : {
        "match": {
          "text" : {
            "query" : "To Be Or Not To Be",
            "cutoff_frequency" : 0.01
          }
        }
      },
      "filter" : {
        "range": {
          "price": {
            "gte": 20.0
            "lte": 50.0
          }
        }
      }
    }
  }
}
```

# Distributed and scalable

# Basic terms

- Index

Logical collection of data; might be time based  
Analogous to a database

- Replication

Read scalability  
Removing SPOF

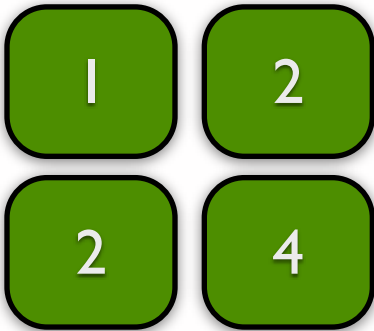
- Sharding

Split logical data over several machines  
Write scalability  
Control data flows

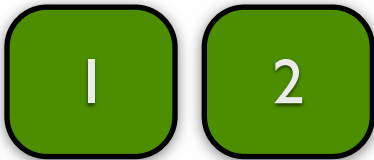
# Shards and replicas

## node 1

### orders



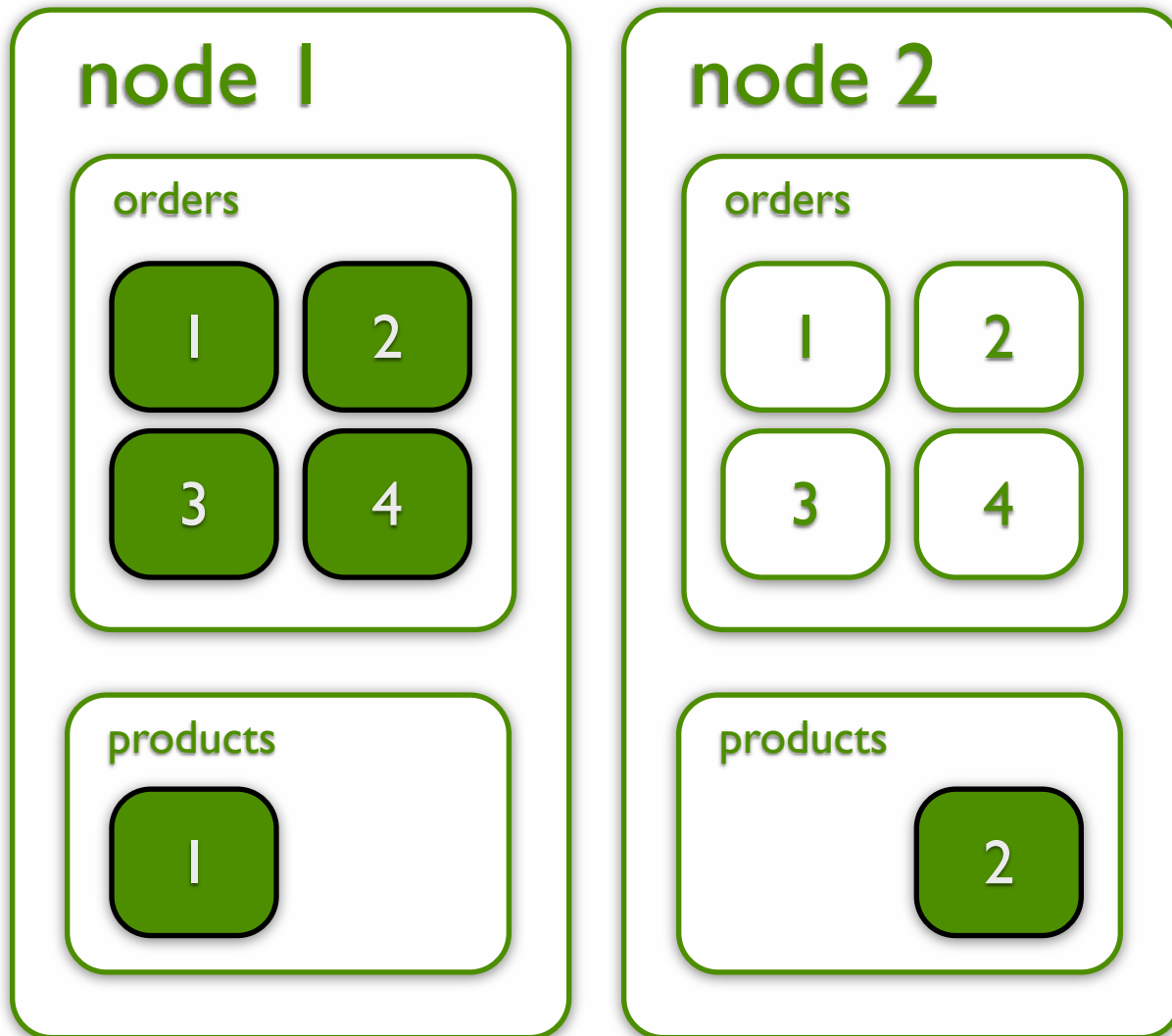
### products



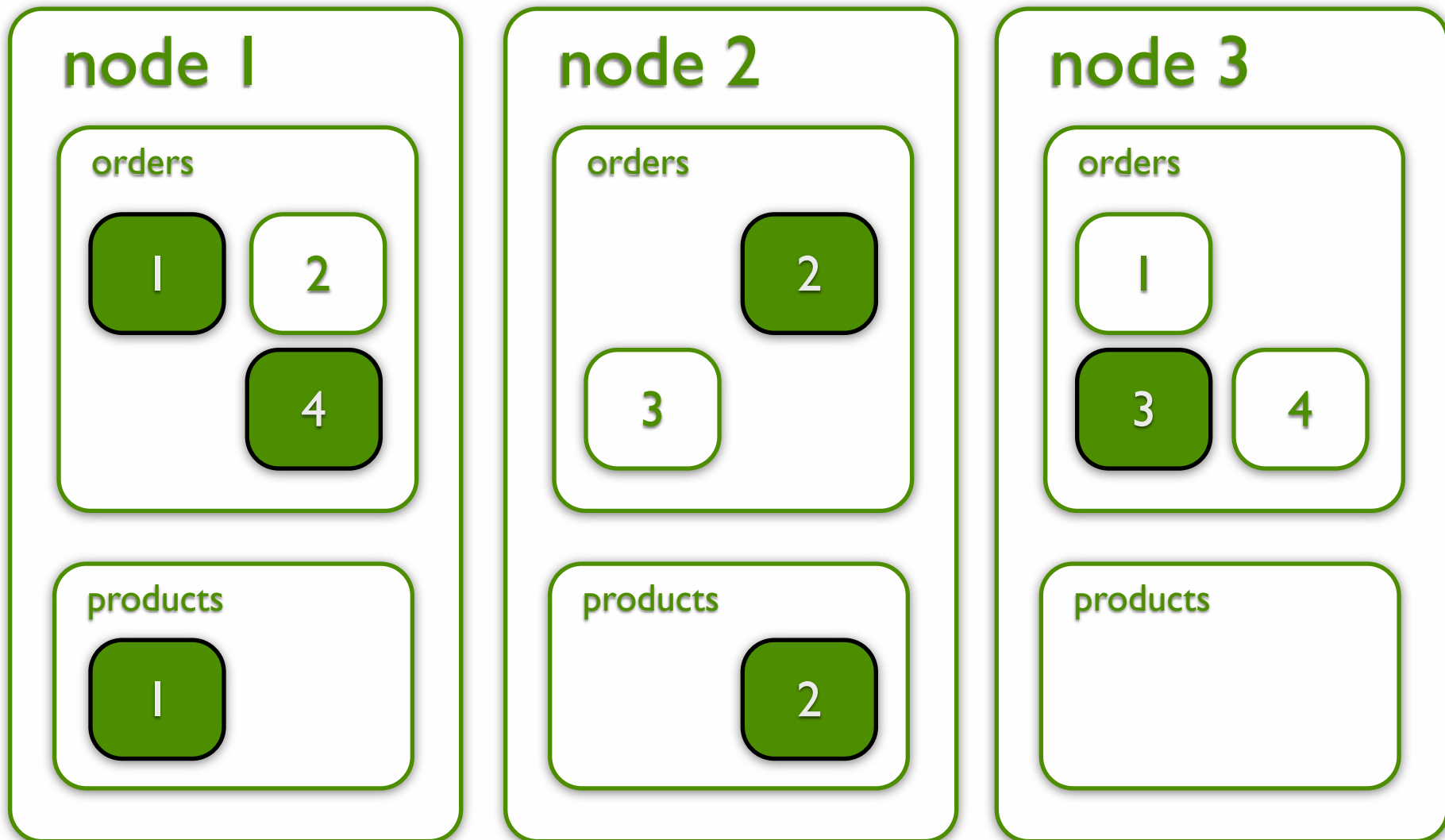
```
curl -X PUT localhost:9200/orders -d '{  
  "settings.index.number_of_shards" : 4  
  "settings.index.number_of_replicas" : 1  
}'
```

```
curl -X PUT localhost:9200/products -d '{  
  "settings.index.number_of_shards" : 2  
  "settings.index.number_of_replicas" : 0  
}'
```

# Shards and replicas



# Automatic leveling



# Cluster management

- Single master at any point in time  
Responsible for cluster state (node entry, mappings)
- Multicast based discovery (optionally unicast)
- Configuration is required here  
Tell each node the name of the cluster to join  
Set minimum master nodes
- Tip: reserve 3 nodes for master role and do not put data on them

# Sizing a cluster or node

- Data and operation dependent

How big are your documents? How many fields in them?

What is your query rate?

Do you do facets/aggregations, sorting, custom scoring?

What is your write rate?

Do you delete documents? Update them?

Is the data time-based?

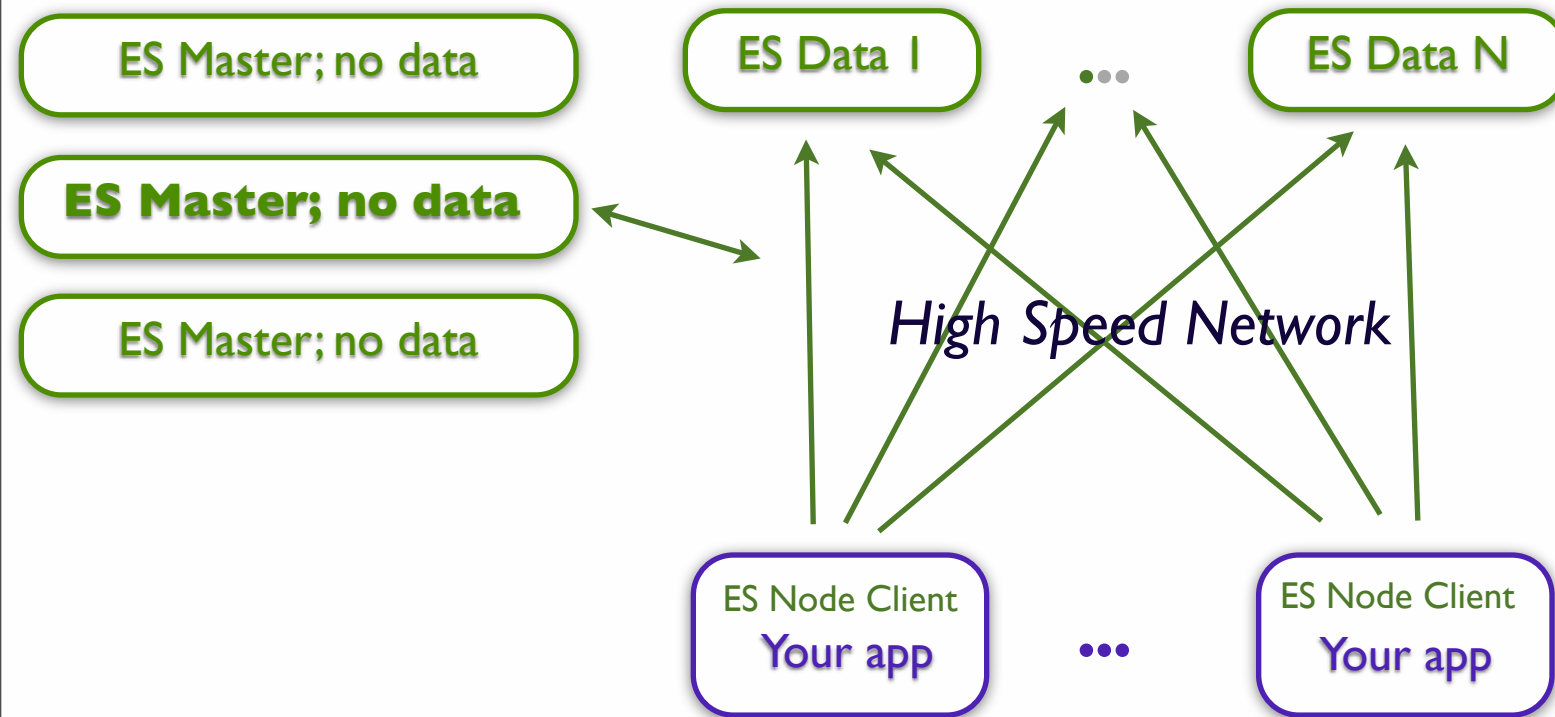
- Test on one node, no replicas

Look at shard size, JVM heap usage and GC frequency, number of shards/node, docs per shard, CPU util, disk util, index pattern

- Tip: 30 GB heap



# Deployment architecture



- Above shows local disk; SAN OK
- Tip: clusters spanning high latency WANs are not recommended. Cross-zone in EC2 is OK.

# Elasticsearch use-cases

# What is data?

- Whatever provides value for your business
- Domain data
  - Internal: Orders, products
  - External: Social media streams, email
- Application data
  - Log files
  - Metrics

# Use case: Product search engine

# Product search engine

- Just index all your products and be happy?  
Search is not that easy
- Synonyms, Suggestions, Faceting, Custom scoring, Analytics, Decomponding, Query optimization, beyond search
- User your domain knowledge

# Scoring

- Is full-text search relevancy really your preferred scoring algorithm?

- Possible influential factors

Age of the product, been ordered in last 24h

In Stock?

No shipping costs

Special offer

Rating (product or seller)

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/query-dsl-function-score-query.html>

# Faceting & user exploration

- Products grouped by
  - Category
  - Material
  - Brand
- Allowing to filter
  - All of the facets
  - Price range
  - Color
  - Seller
  - Ratings (hard!)

# Notification with percolation

- Customer: If a product matches name  $X$  and costs below price  $Y$ , is color  $Z$ , then I want to get a mail  
More likely: Notify customer, when it is back in stock

- Enter percolation!

Not: Index a document and fire a query

But: Index a query and check a document for a match

<https://speakerdeck.com/javanna/whats-new-in-percolator>



# Use-case: Analytics

# Analytics

- Aggregation of information
- Facets are one dimensional  
Categories/brands/material of all results of this query
- Questions are multidimensional  
Average revenue per category id per day
- Elasticsearch 1.0 has aggregations  
Nested faceting

# Create knowledge from data

- Orders

How many orders were created every day in the last month?

How many orders were created per state in the last month?

- Money

What is the average revenue per shopping cart?

What is the average shopping cart size per order per hour?

- Product portfolio

Take the location of people into account for special offers?

Analyse page views: Premium or low budget ecommerce site?

# Ecosystem

- Plugins

Many third party plugins available

- Clients for many languages

Ruby, python, php, perl, javascript, (.NET coming)  
Scala, clojure, go

- Kibana

- Logstash

- Hadoop integration

XING 



elasticsearch.

# Tools for sys admins

# REST-based management

- Elasticsearch is full of monitoring APIs  
Everything is returned as JSON
- Humans are not the world's best JSON parsers
- What if elasticsearch had an easy to use interface from the commandline?

# Which node is the master?

```
$ curl "localhost:9200/_cluster/state?pretty&filter_metadata=true&filter_routing_table=true"
{
  "cluster_name" : "elasticsearch",
  "master_node" : "GNf0hEXlTfaBvQXKBF300A",
  "blocks" : { },
  "nodes" : {
    "ObdRqLHGQ6CMI5rOEstA5A" : {
      "name" : "Triton",
      "transport_address" : "inet[/10.0.1.11:9300]",
      "attributes" : { }
    },
    "4C7pKbfhTvu0slcSy_G4_w" : {
      "name" : "Kid Colt",
      "transport_address" : "inet[/10.0.1.12:9300]",
      "attributes" : { }
    },
    "GNf0hEXlTfaBvQXKBF300A" : {
      "name" : "Lang, Steven",
      "transport_address" : "inet[/10.0.1.13:9300]",
      "attributes" : { }
    }
  }
}
```



# Which one is the master? (v1.0)

```
$ curl localhost:9200/_cat/master  
GNf0hEXlTfaBvQXKBF300A 10.0.1.13 Lang, Steven
```

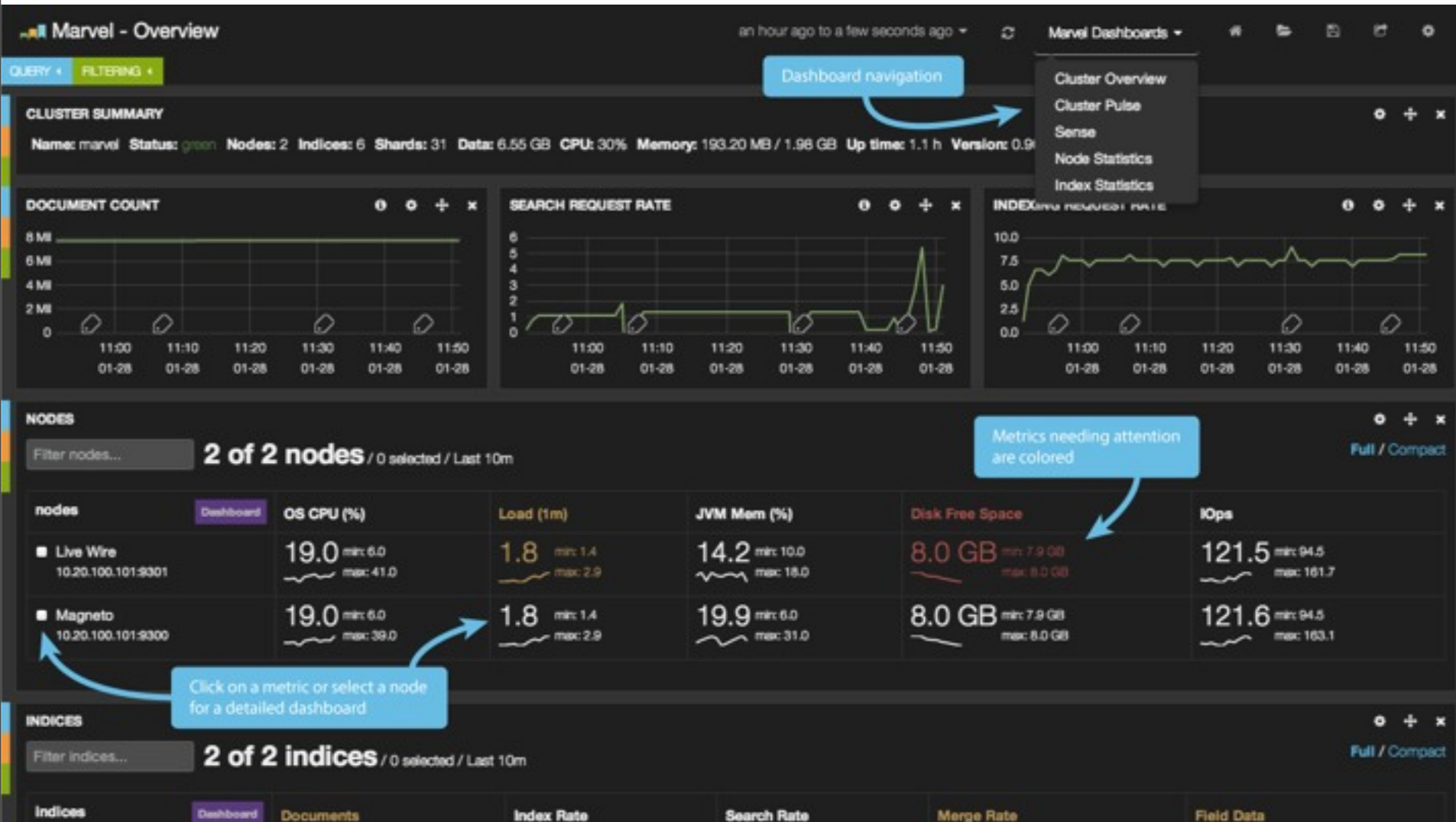
# `_cat/*` api

- `/_cat/allocation`
- `/_cat/count`
- `/_cat/health`
- `/_cat/master`
- `/_cat/aliases`
- `/_cat/nodes`
- `/_cat/recovery`
- `/_cat/shards`
- `/_cat/indices`
- `/_cat/thread_pool`

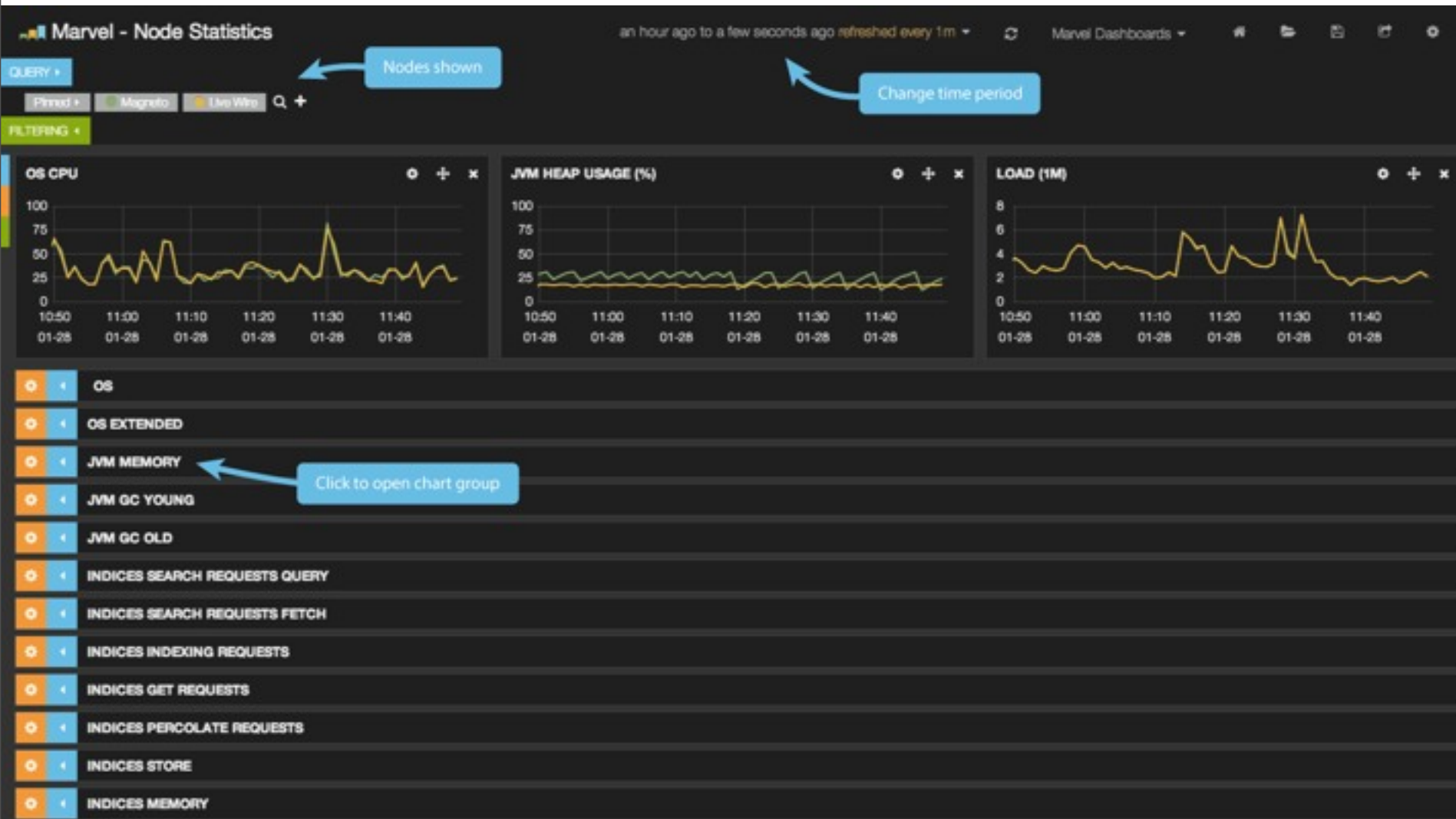
# Monitor your cluster with Marvel

- Point in time views are a start
- Marvel shows historical trends
- Visualize cluster behavior, act before problems
- Free for development, \$500/year for up to 5 nodes

# Overview



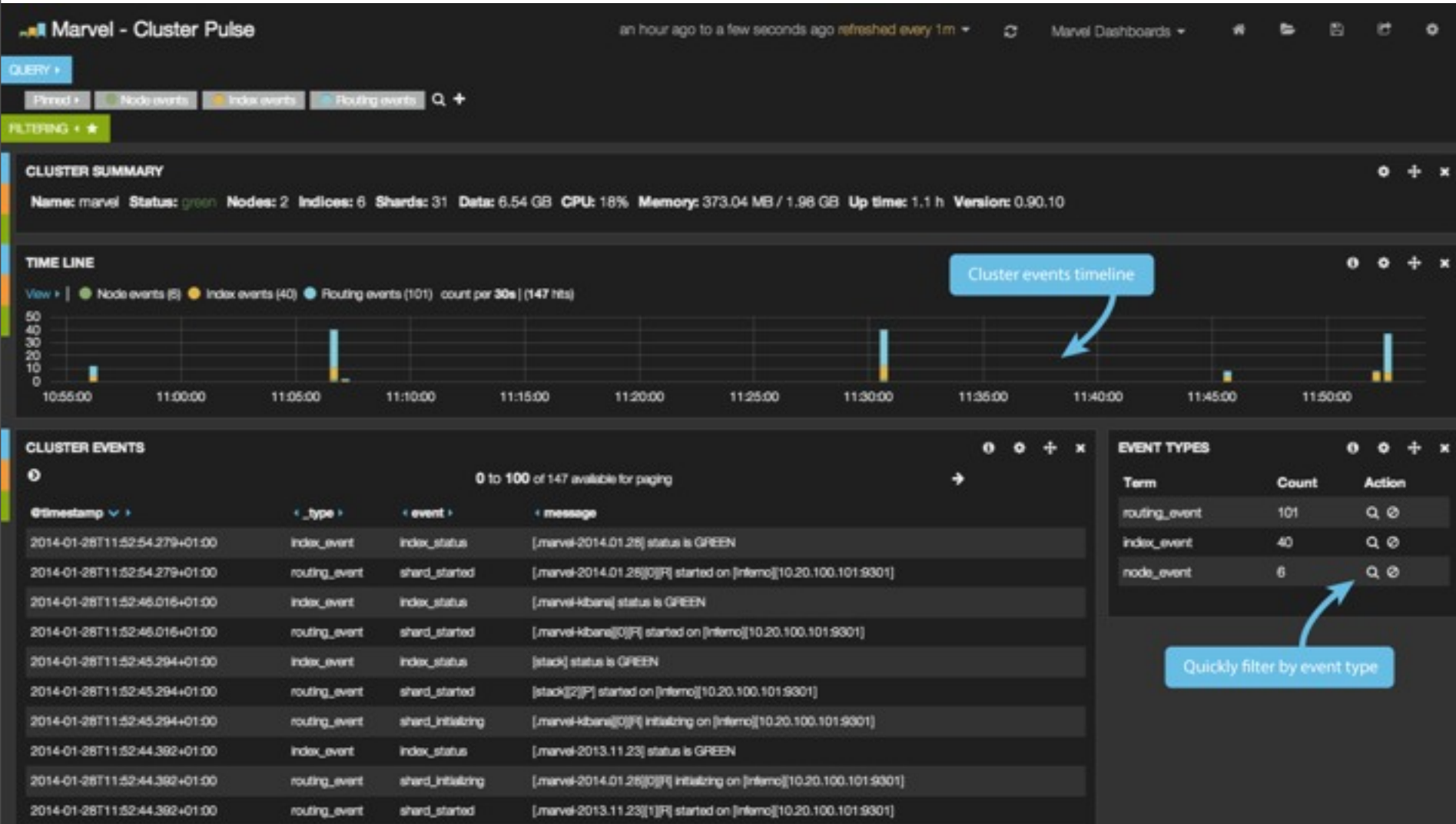
# Node statistics



# Index statistics



# Cluster Pulse





# Sense

Server: localhost:9200

Submit request to Elasticsearch

1- # search for a super hero  
2- GET marvel/superhero/\_search  
3- {  
4- "query": {  
5- "match": {  
6- "name": "spiderman"  
7- }  
8- }  
9- "powers": string  
10- "enemies": string  
11- "rating": long  
12- }  
13- # index a doc  
14- PUT marvel/superhero/spiderman  
15- {  
16- "name": "Spiderman",  
17- "powers": ["webbing", "climbing", "night vision"],  
18- "enemies": ["the green goblin", "venom"]  
19- }  
20-  
21- # create an index  
22- PUT marvel  
23- {  
24- "settings": {  
25- "number\_of\_shards": 2,  
26- "number\_of\_replicas": 1  
27- },  
28- "mappings": {  
29- "superhero": {  
30- "properties": {  
31- "name": { "type": "string" },  
32- "powers": {  
33- "type": "string",  
34- "index": "not\_analyzed"  
35- }  
36- }  
37- }  
38- }  
39- }  
40-  
41- PUT marvel/superhero/venom  
42- {  
43- "name": "Venom",  
44- "rating": 5  
45- }  
46-  
47- PUT marvel/superhero/green Goblin

Suggestions as you type

1- {  
2- "took": 6,  
3- "timed\_out": false,  
4- "\_shards": {  
5- "total": 2,  
6- "successful": 2,  
7- "failed": 0  
8- },  
9- "hits": {  
10- "total": 1,  
11- "max\_score": 1,  
12- "hits": [  
13- {  
14- "\_index": "marvel",  
15- "\_type": "superhero",  
16- "\_id": "spiderman",  
17- "\_score": 1,  
18- "\_source": {  
19- "name": "Spiderman",  
20- "powers": [  
21- "webbing",  
22- "climbing",  
23- "night vision"  
24- ],  
25- "enemies": [  
26- "the green goblin",  
27- "venom"  
28- ]  
29- }  
30- }  
31- ]  
32- }  
33- }

API response





# Log analysis with Logstash and Kibana

# Logstash in 10 seconds

- Managing events and logs
- Collect, parse, enrich, store data
- Modular: many, many inputs and outputs
- Apache License 2.0
- Ruby app (JRuby)
- Part of Elasticsearch family

# What is a log?

- Time-based data
- This data is everywhere!
  - Server logs
  - Twitter stream
  - Financial transactions
  - Metric / monitoring data
  - ...
- Log all things

# Why collect & centralize logs?

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

# Logstash architecture

## Input

*collect and split*

## Filter

*alter and enrich*

## Output

*store and visualize*

?



Logstash



?

# Inputs

- Monitoring: collectd, graphite, ganglia, snmptrap, zenoss
- Datastores: elasticsearch, redis, sqlite, s3
- Queues: rabbitmq, zeromq
- Logging: eventlog, lumberjack, gelf, log4j, rdp, syslog, varnish log
- Platforms: drupal\_dblog, gemfire, heroku, sqs, s3, twitter
- Local: exec, generator, file, stdin, pipe, unix
- Protocol: imap, irc, stomp, tcp, udp, websocket, wmi, xmpp

# Filters

- alter, anonymize, checksum, csv, drop, multiline
- dns, date, extractnumbers, geoip, i18n, kv, noop, ruby, range
- json, urldecode, useragent
- metrics, sleep
- ... many, many more ...

# Outputs

- Store: elasticsearch, gemfire, mongodb, redis, riak, rabbitmq
- Monitoring: ganglia, graphite, graphtastic, nagios, opentsdb, statsd, zabbix
- Notification: email, hipchat, irc, pagerduty, sns
- Protocol: gelf, http, lumberjack, metriccatcher, stomp, tcp, udp, websocket, xmpp
- External Monitoring: boundary, circonus, cloudwatch, datadog, librato
- External service: google big query, google cloud storage, jira, loggly, riemann, s3, sqs, syslog, zeromq
- Local: csv, exec, file, pipe, stdout, null



# Installation

- Ruby application, but Java required (JRuby)
- Download single tgz, deb, RPM (also repositories)  
No gem/dependency nightmares!
- Puppet module

# Simple example

- Download, create config and run

```
input {  
  stdin {}  
}  
  
output {  
  stdout { debug => true }  
}
```

← simple.conf



```
echo foo | java -jar logstash-1.3.3-flatjar.jar agent -f simple.conf  
{  
  "message" => "foo",  
  "@version" => "1",  
  "@timestamp" => "2014-01-20T13:30:59.648Z",  
  "host" => "kryptic.fritz.box"  
}
```

# Simple filter with grok

```
input {
  stdin {}
}

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

output {
  stdout { debug => true }
}
```

# Simple filter with grok

```
echo "Alexander Reelsen 30" | java -jar
logstash-1.3.3-flatjar.jar agent -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"
}
```

# 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 { debug => true } }
```

# Syslog example with grok

```
Jun 10 04:04:01 lvps109-104-93-171 postfix/smtpd[11105]:  
connect from mail-we0-f196.google.com[74.125.82.196]  
{  
    "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]"  
}
```

# CLF log files

```
{
  "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

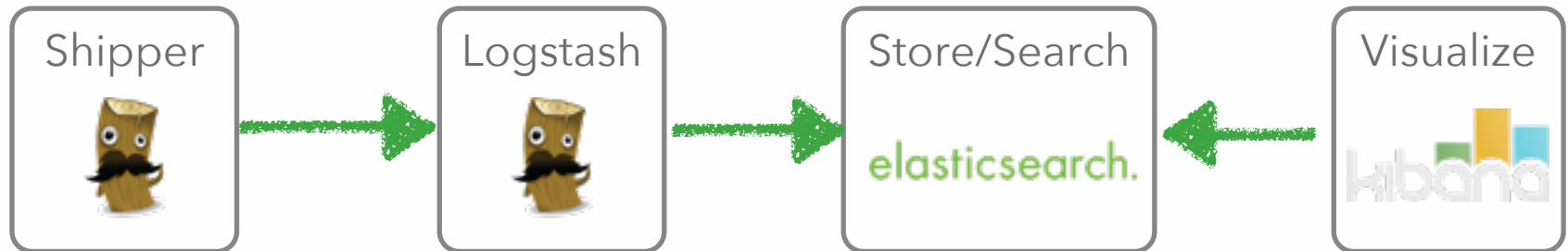
```
input { stdin {} }

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

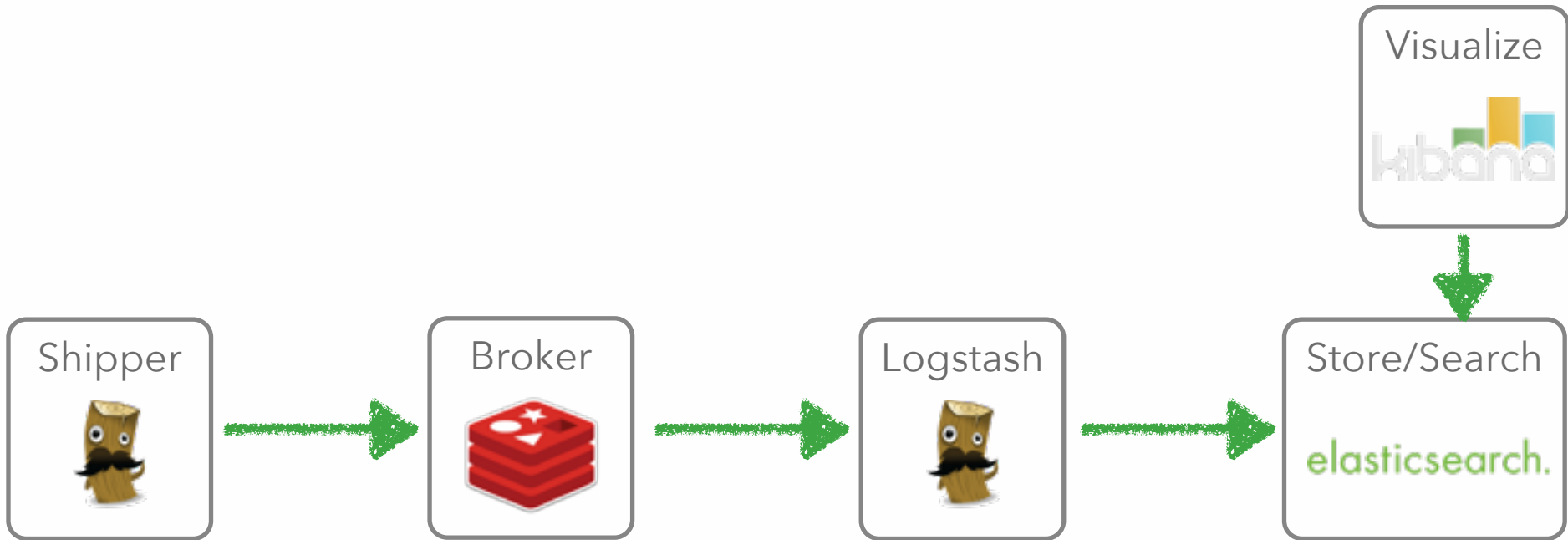
output {
  elasticsearch_http {}
}
```



# Deploying ELK for scale



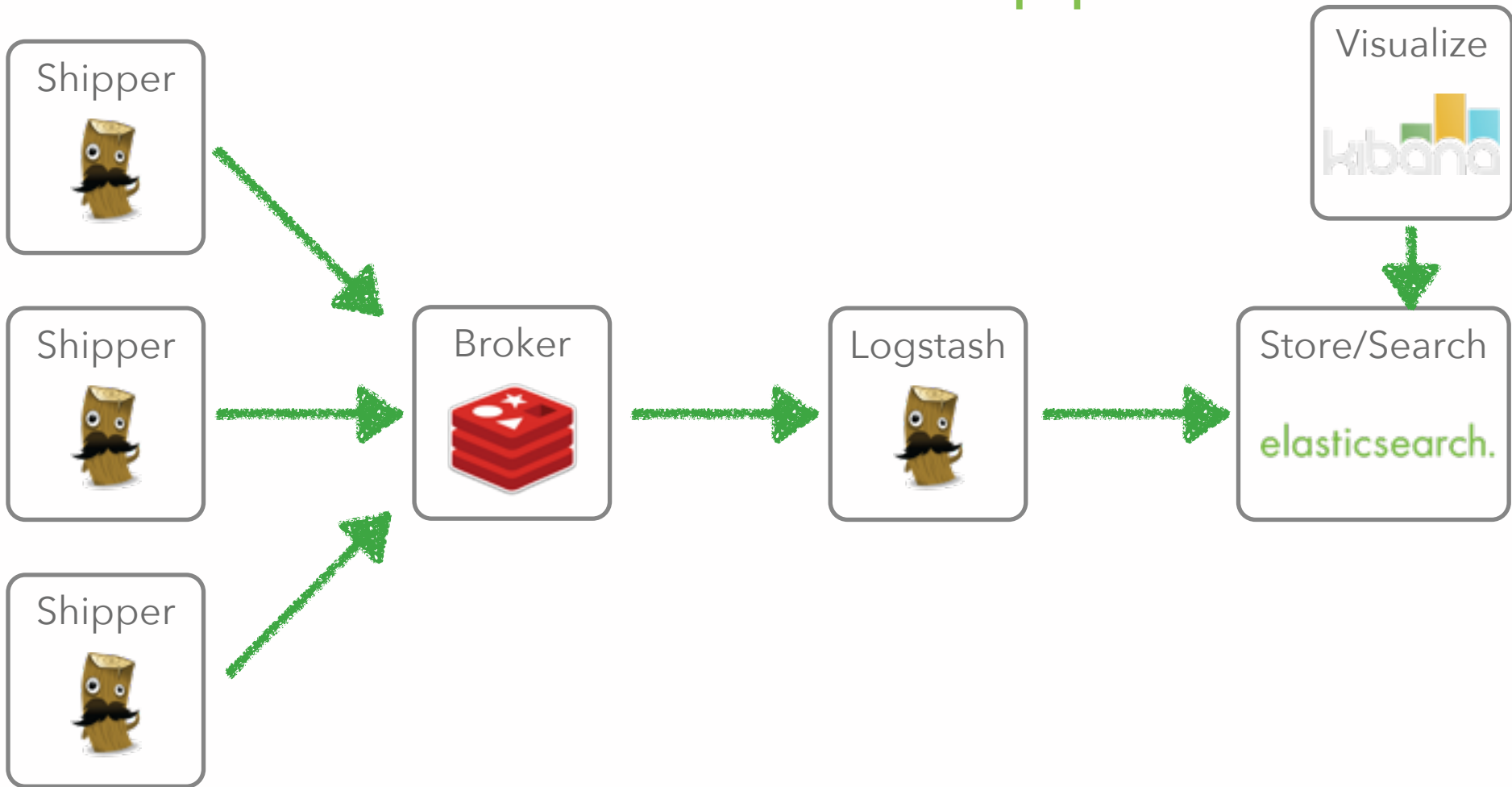
# Add a broker



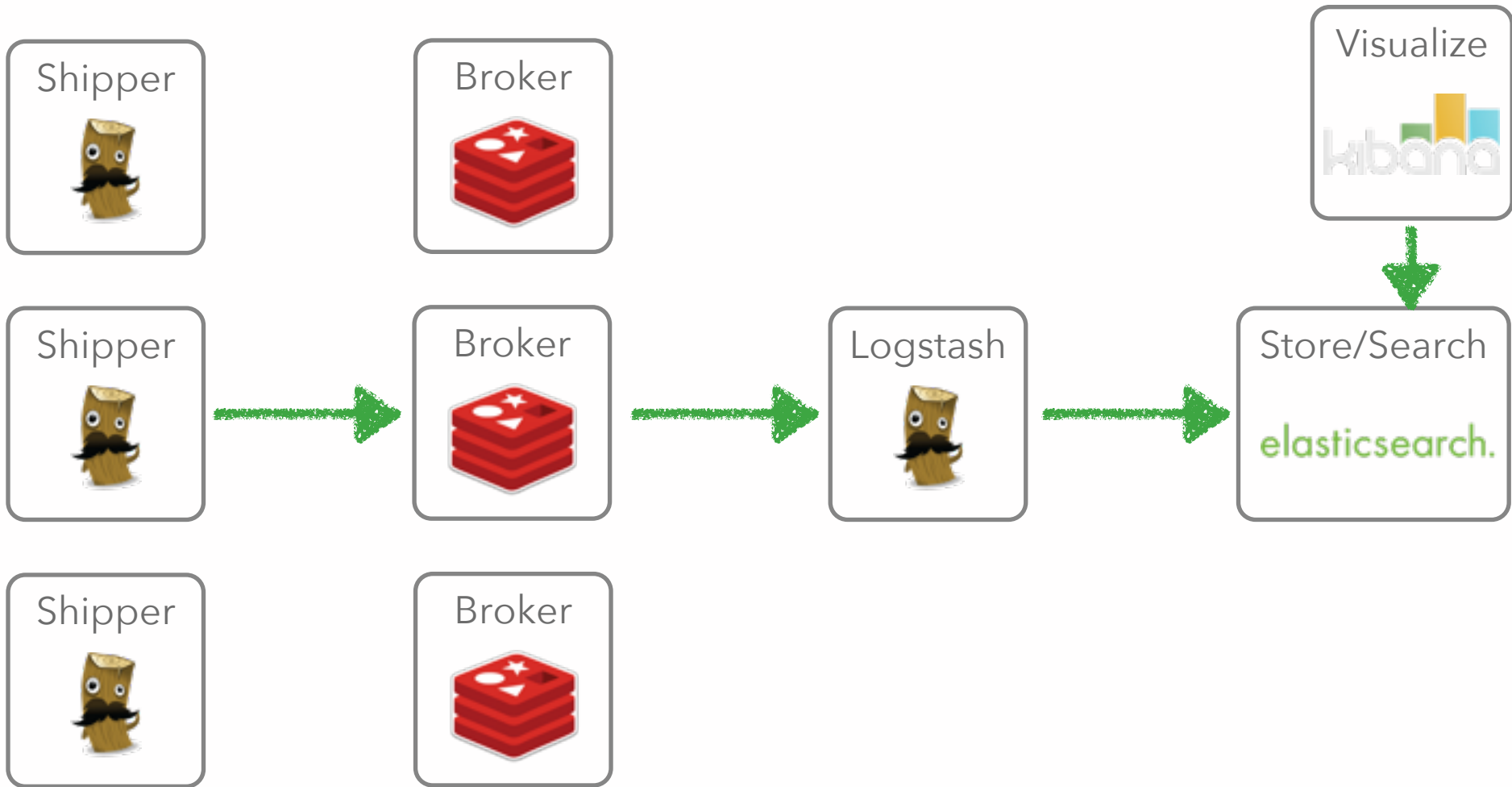
Brokers help with scale and stability by buffering the input and protecting against output downtime.

Tip: set limits on broker queue to push back on source as well.

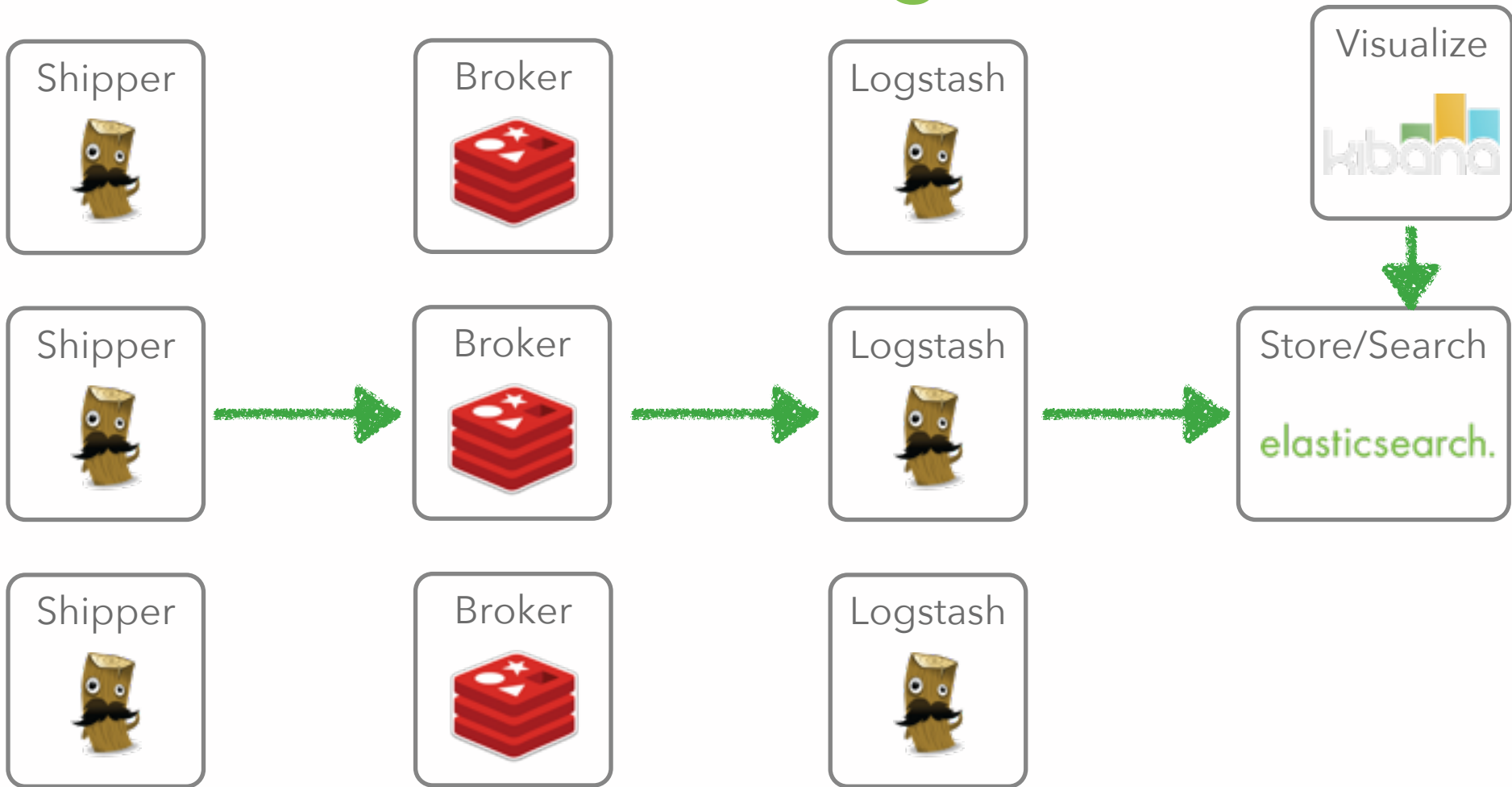
# Scale out the shipper



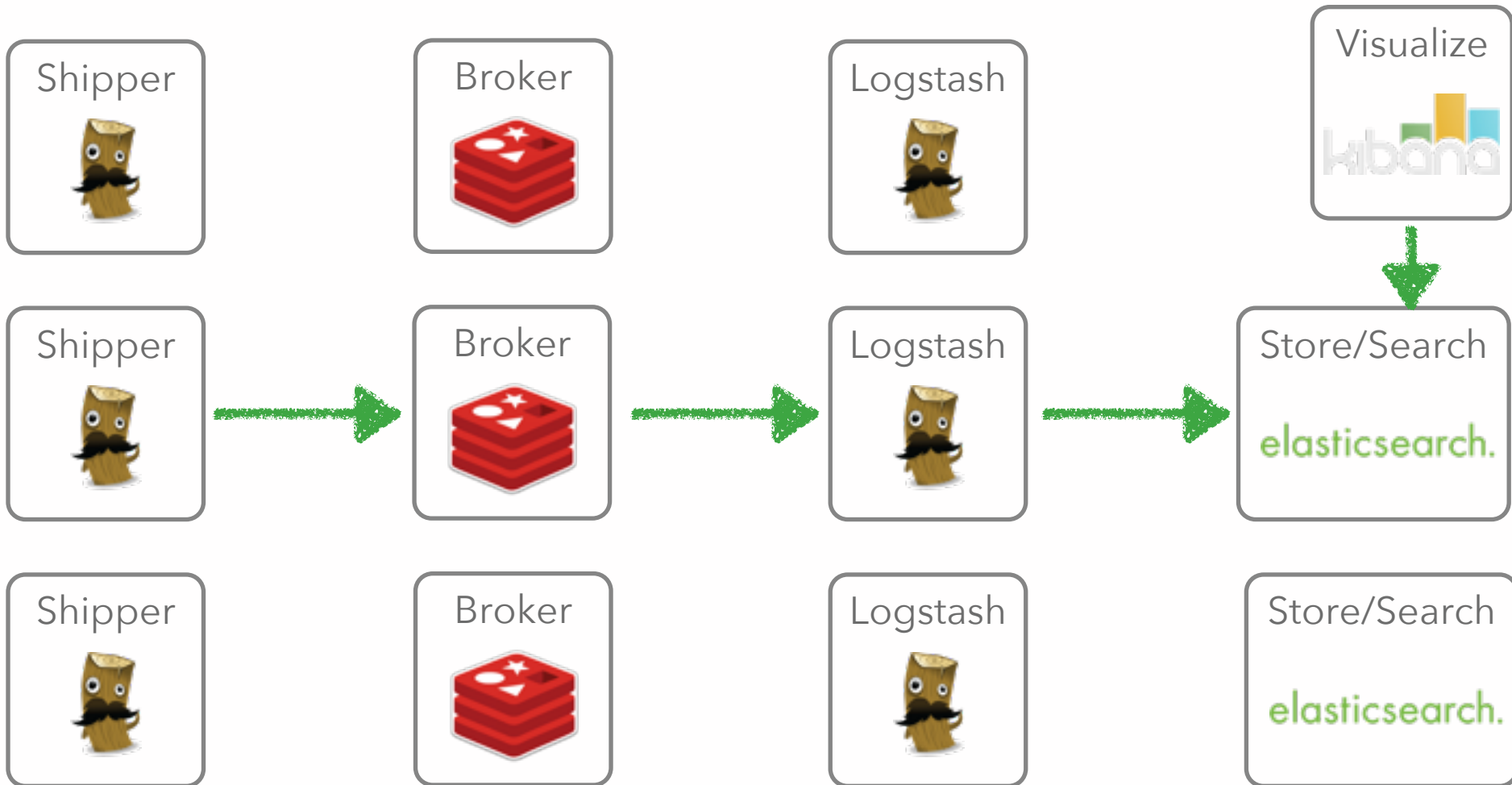
# Scale out the broker



# Scale out Logstash



# Scale out Elasticsearch



# Logstash scaling

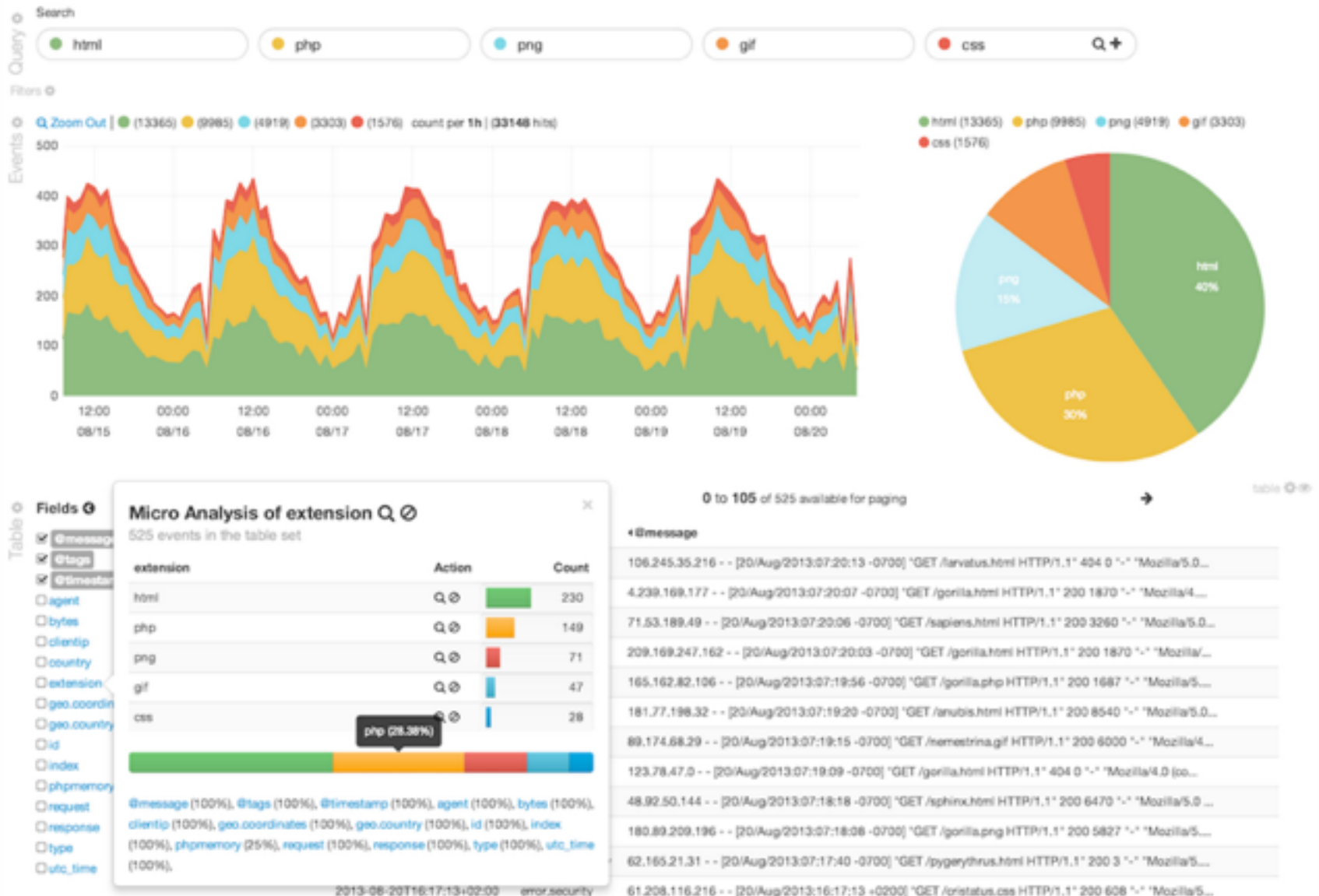
- Events get passed via Ruby SizedQueue
- input/worker/output threads, can be configured
- Each input is one thread, unless explicitly configured
- One worker thread by default, use -w to change
- Output is a single thread (some outputs have their own queueing thread)

<http://logstash.net/docs/1.3.3/life-of-an-event>

# Visualize with Kibana



# Kibana



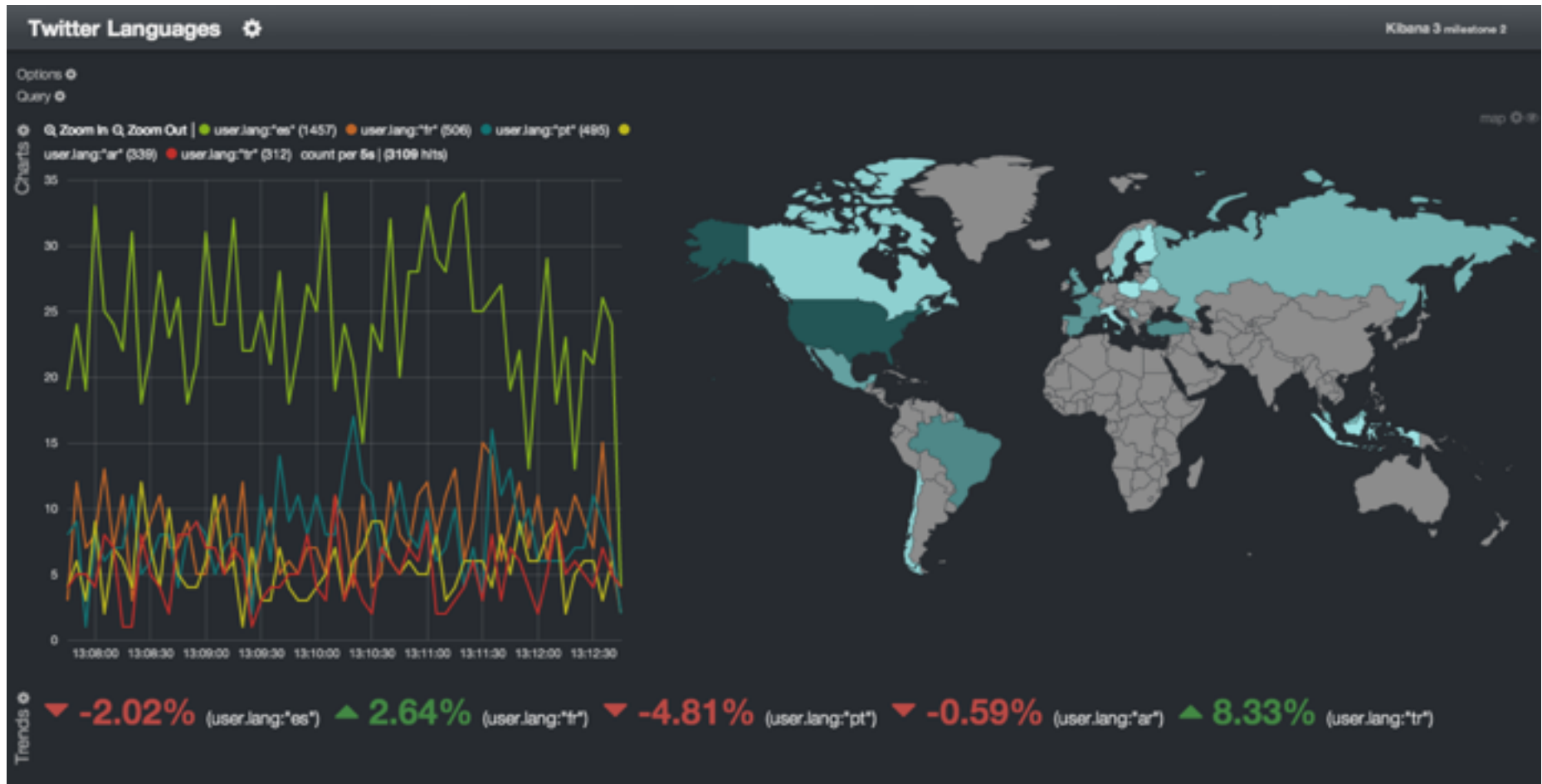
elasticsearch.

# Kibana



elasticsearch.

# Kibana



# Useful helpers

- Curator: index management

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

- Puppet module

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

- logstash forwarder: low overhead collector

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

- Logstash cookbook

<http://cookbook.logstash.net/>

# More info

- Github: <https://github.com/elasticsearch>  
Code, issues there  
Except Logstash issues at <https://logstash.jira.com>
- Mailing lists  
Google groups, logstash-users and elasticsearch
- IRC channels  
#logstash and #elasticsearch on freenode
- We're hiring!  
[jobs@elasticsearch.com](mailto:jobs@elasticsearch.com)