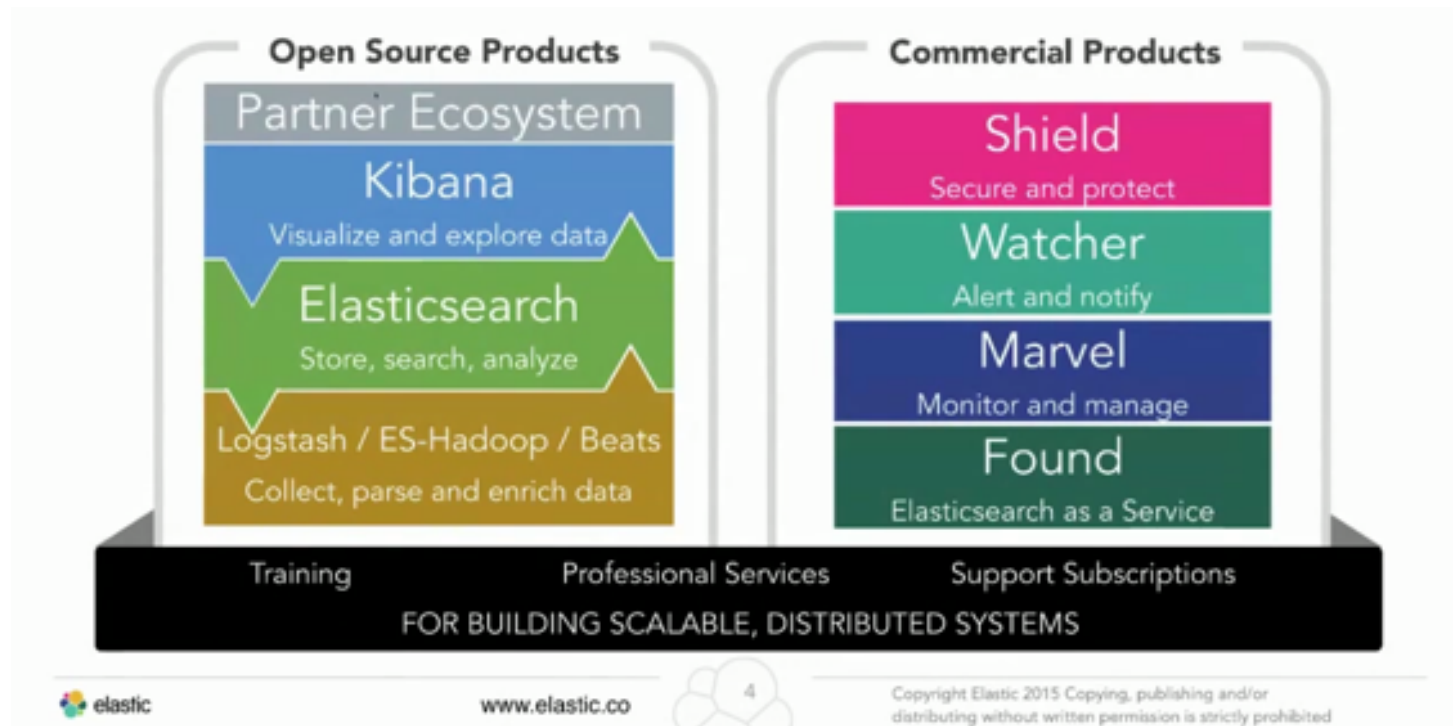




# Elasticsearch



# Elastic Products



# Elasticsearch

- ❑ Highly scalable open-source full-text search and analytics engine
- ❑ Based on Apache Lucene™
- ❑ JSON + RESTful
- ❑ Real-time, Distributed, Schema-Free, Conflict Management
- ❑ Per-Operation Persistence (Document changes are recorded in transaction logs on multiple nodes in the cluster to minimize the chance of any data loss)

# Who is using ES

- ❏ Wikipedia
- ❏ Stack Overflow
- ❏ GitHub
- ❏ SoundCloud
- ❏ The Guardian
- ❏ Mozilla
- ❏ ...

# Basic Concepts

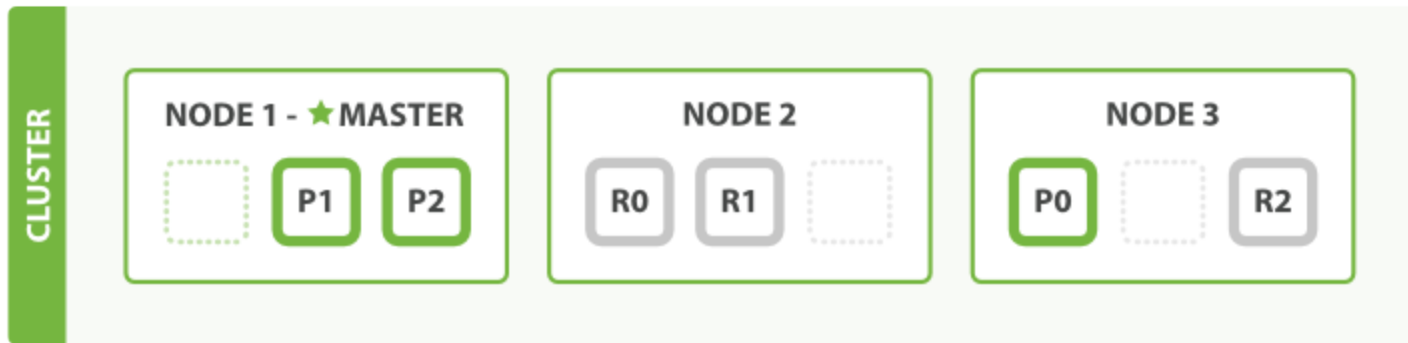
Cluster, Nodes

Index, Type, Document, Field

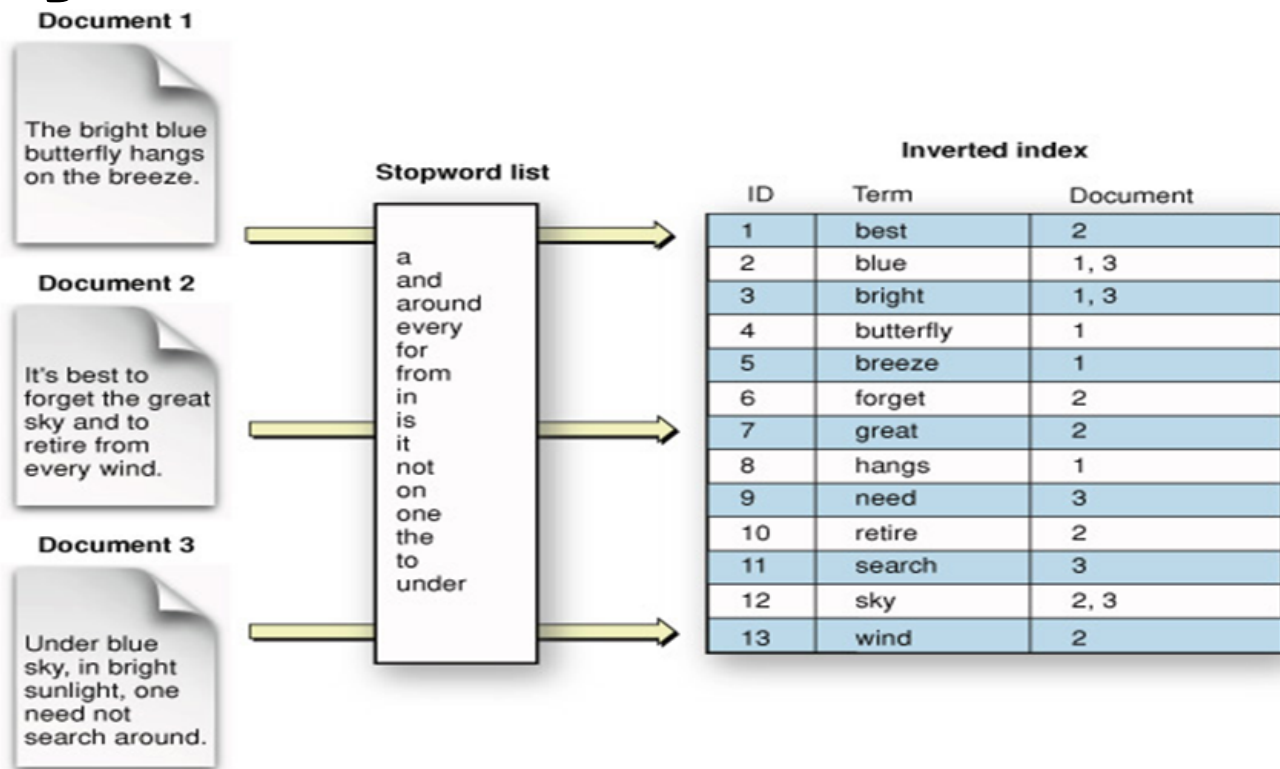
Shards & Replicas

Mapping

# Cluster



# Indexing



# Terminology

Relational DB	ElasticSearch
Database	Index
Table	Type
Row	Document
Column	Field
Schema	Mapping
Index	*Everything
SQL	query DSL



# Installing and running

```
curl -L -O http://download.elasticsearch.org/PATH/TO/VERSION.zip
```

```
unzip elasticsearch-$VERSION.zip
```

```
cd elasticsearch-$VERSION
```

Settings: config/elasticsearch.yml (cluster.name, node.name)

```
./bin/elasticsearch
```

# Info

```
curl 'http://localhost:9200/?pretty'
```

```
{
  "status" : 200,
  "name" : "katarina_dev",
  "cluster_name" : "elasticsearch_kat",
  "version" : {
    "number" : "1.7.1",
    "build_timestamp" : "2015-07-29T09:54:16Z",
    "build_snapshot" : false,
    "lucene_version" : "4.10.4"
  },
  "tagline" : "You Know, for Search"
}
```

# Cluster health

GET \_cluster/health

```
{  
  "cluster_name": "elasticsearch_kat",  
  "status": "yellow",  
  "timed_out": false,  
  "number_of_nodes": 1,  
  "number_of_data_nodes": 1,  
  "active_primary_shards": 16,  
  "active_shards": 16,  
  "relocating_shards": 0,  
  "initializing_shards": 0,  
  "unassigned_shards": 16,  
  "delayed_unassigned_shards": 0,  
  "number_of_pending_tasks": 0,  
  "number_of_in_flight_fetch": 0  
}
```

# Sense

The screenshot displays the Elasticsearch Sense web interface. At the top, the 'Server' dropdown is set to 'localhost:9200'. The interface is divided into two main panels. The left panel shows a REST client with a `GET _search` query: 

```
{
  "query": {
    "match_all": {}
  },
  "aggs": {}
}
```

. The right panel displays the JSON response from the search. The response includes metadata like 'took' (5ms) and 'timed\_out' (false), and a 'hits' section containing one document. This document is a 'node\_stats' entry for a node named 'Mister Hyde' with IP '192.168.2.100'. It also includes an 'indices' section with document counts and an 'indexing' section with various statistics.

```
1 GET _search
2 {
3   "query": {
4     "match_all": {}
5   },
6   "aggs": {}
7 }

1 {
2   "took": 5,
3   "timed_out": false,
4   "shards": {
5     "total": 10,
6     "successful": 6,
7     "failed": 0
8   },
9   "hits": {
10    "total": 204,
11    "max_score": 1,
12    "hits": [
13      {
14        "_index": "marvel-2014-06-12",
15        "_type": "node_stats",
16        "_id": "-EjvuX0L7i1gb55yIqX8Hg",
17        "_score": 1,
18        "_source": {
19          "@timestamp": "2014-06-12T05:34:01.347Z",
20          "cluster_name": "elasticsearch",
21          "node": {
22            "id": "McRwsW76Rpy5d_6A071WZA",
23            "name": "Mister Hyde",
24            "transport_address": "inet[/192.168.2.100:9300]",
25            "ip": "192.168.2.100",
26            "host": "Flo-ThinkPad-T420.local",
27            "ip_port": "192.168.2.100:9300",
28            "master_node": true,
29            "data_node": true,
30            "master": true
31          },
32          "indices": {
33            "docs": {
34              "count": 1,
35              "deleted": 0
36            },
37            "store": {
38              "size_in_bytes": 3565,
39              "throttle_time_in_millis": 0
40            },
41            "indexing": {
42              "index_total": 0,
43              "index_time_in_millis": 0,
44              "index_current": 0,
45              "delete_total": 0,
46              "delete_time_in_millis": 0,
47              "delete_current": 0
48            },
49            "get": {
50              "total": 0,
51              "time_in_millis": 0,
52              "exists_total": 0,
53              "exists_time_in_millis": 0,
54              "missing_total": 0
55            }
56          }
57        }
58      ]
59    }
60  }
```

# Analyzer

```
curl -X POST "localhost:9200/3a2ilati" -d '
```

[illegible]

# Insert data

```
PUT /website/blog/123
```

```
{  
  "title": "My first blog entry",  
  "text": "Just trying this out...",  
  "date": "2015/09/09"  
}
```

```
response:
```

```
{  
  "_index": "website",  
  "_type": "blog",  
  "_id": "123",  
  "_version": 1,  
  "created": true  
}
```

# Get data

GET /website/blog/123?pretty

```
{
  "_index" :   "website",
  "_type" :    "blog",
  "_id" :      "123",
  "_version" : 1,
  "found" :    true,
  "_source" :  {
    "title": "My first blog entry",
    "text":  "Just trying this out...",
    "date":  "2015/09/09"
  }
}
```

part of a document: GET /website/blog/123?\_source=title,text

just source: GET /website/blog/123/\_source

# Multi-index, Multitype

`/_search`

Search all types in all indices

`/gb/_search`

Search all types in the `gb` index

`/gb,us/_search`

Search all types in the `gb` and `us` indices

`/g*,u*/_search`

Search all types in any indices beginning with `g` or beginning with `u`

`/gb/user/_search`

Search type `user` in the `gb` index

`/gb,us/user,tweet/_search`

Search types `user` and `tweet` in the `gb` and `us` indices

`/_all/user,tweet/_search`

Search types `user` and `tweet` in all indices



# Search Lite

*query-string* version that expects all its parameters to be passed in the query string

```
GET /website/blog/_search
```

```
GET /website/blog/_search?q=title:elasticsearch
```

# Search with Query DSL

The domain-specific language (DSL) is specified using a JSON request body

```
GET /website/blog/_search
{
  "match" : {
    "title" : "elasticsearch"
  }
}
```

# Filters

GET /megacorp/employee/\_search

```
{
  "query": {
    "filtered": {
      "filter": {
        "range": {
          "age": { "gt": 30 }
        }
      },
      "query": {
        "match": {
          "last_name": "smith"
        }
      }
    }
  }
}
```

```
{
  ...
  "hits": {
    "total": 1,
    "max_score": 0.30685282,
    "hits": [
      {
        ...
        "_source": {
          "first_name": "Jane",
          "last_name": "Smith",
          "age": 32,
          "about": "I like to collect rock albums",
          "interests": [ "music" ]
        }
      }
    ]
  }
}
```

# Full-Text Search

GET /megacorp/employee/\_search

```
{
  "query": {
    "match": {
      "about": "rock climbing"
    }
  }
}
```

```
{
  ...
  "hits": {
    "total": 2,
    "max_score": 0.16273327,
    "hits": [
      {
        ...
        "_score": 0.16273327, (1)
        "_source": {
          "first_name": "John",
          "last_name": "Smith",
          "age": 25,
          "about": "I love to go rock climbing",
          "interests": [ "sports", "music" ]
        }
      },
      {
        ...
        "_score": 0.016878016, (1)
        "_source": {
          "first_name": "Jane",
          "last_name": "Smith",
          "age": 32,
          "about": "I like to collect rock albums",
          "interests": [ "music" ]
        }
      }
    ]
  }
}
```

# Phrase Search

```
GET /megacorp/employee/_search
```

```
{
```

```
  "query" : {
```

```
    "match_phrase" : {
```

```
      "about" : "rock climbing"
```

```
    }
```

```
  }
```

```
}
```

# Highlighting

GET /megacorp/employee/\_search

```
{
  "query" : {
    "match_phrase" : {
      "about" : "rock climbing"
    }
  },
  "highlight" : {
    "fields" : {
      "about" : {}
    }
  }
}
```

```
{
  ...
  "hits": {
    "total": 1,
    "max_score": 0.23013961,
    "hits": [
      {
        ...
        "_score": 0.23013961,
        "_source": {
          "first_name": "John",
          "last_name": "Smith",
          "age": 25,
          "about": "I love to go rock climbing",
          "interests": [ "sports", "music" ]
        },
        "highlight": {
          "about": [
            "I love to go <em>rock</em> <em>climbing</em>" (1)
          ]
        }
      }
    ]
  }
}
```

# Bool Queries

Occurrence types:

- `must`
- `should`
- `must_not`

GET `/my_index/my_type/_search`

```
{
  "query": {
    "bool": {
      "should": [
        { "match": { "title": "brown" } },
        { "match": { "title": "fox" } },
        { "match": { "title": "dog" } }
      ],
      "minimum_should_match": 2
    }
  }
}
```

# Combining Queries

GET /my\_index/my\_type/\_search

```
{
  "query": {
    "bool": {
      "must":      { "match": { "title": "quick" }},
      "must_not": { "match": { "title": "lazy" }},
      "should": [
        { "match": { "title": "brown" }},
        { "match": { "title": "dog" }}
      ]
    }
  }
}
```

```
{
  "hits": [
    {
      "_id":      "3",
      "_score":   0.70134366,
      "_source": {
        "title": "The quick brown fox jumps over the quick dog"
      }
    },
    {
      "_id":      "1",
      "_score":   0.3312608,
      "_source": {
        "title": "The quick brown fox"
      }
    }
  ]
}
```



# From, Size

GET /megacorp/employee/\_search

```
{
  "from" : 0,
  "size" : 15,
  "query" : {
    "match" : {
      "about" : "rock climbing"
    }
  }
}
```

# Validate query

GET /website/blog/\_validate/query?explain

```
1 {  
2   "valid": false,  
3   "_shards": {  
4     "total": 1,  
5     "successful": 1,  
6     "failed": 0  
7   },  
8   "explanations": [  
9     {  
10      "index": "shakespeare",  
11      "valid": false,  
12      "error": "org.elasticsearch.index.query.QueryParseException: [shakespeare] request does not support [script_fields]"  
13    }  
14  ]  
15 }
```

# Bulk API

```
{action_and_meta_data}\n{optional_data_source}\n{action_and_meta_data}\n{optional_data_source}\n...
```

## Actions:

POST /\_bulk

```
{ "create": { "_index": "website", "_type": "blog", "_id": "123" }}\n{ "title":    "Cannot create - it already exists" }\n{ "index":   { "_index": "website", "_type": "blog", "_id": "123" }}\n{ "title":    "But we can update it" }
```

- **create** (Create a document only if the document does not already exist)
- **index** (Create a new document or replace an existing document)
- **update** (Do a partial update on a document)
- **delete** (Delete a document)

# Import some data

<https://www.elastic.co/guide/en/kibana/3.0/import-some-data.html>

<https://www.elastic.co/guide/en/kibana/3.0/snippets/shakespeare.json>

```
curl -XPUT localhost:9200/_bulk --data-binary @shakespeare.json
```