15 Minutes Introduction to ELK (Elastic Search,LogStash,Kibana)

KickStarter Series

Karun Subramanian

www.karunsubramanian.com

Contents

What is ELK and why is all the hoopla?

Elastic Search is a robust ***search and analytics* tool** that stores data in a document oriented data store. It is open source, meaning you can download, use and modify the program free of charge. The most popular usage of ElasticSearch today is **Log Management**. Similar products in the market today are Splunk and Solr.

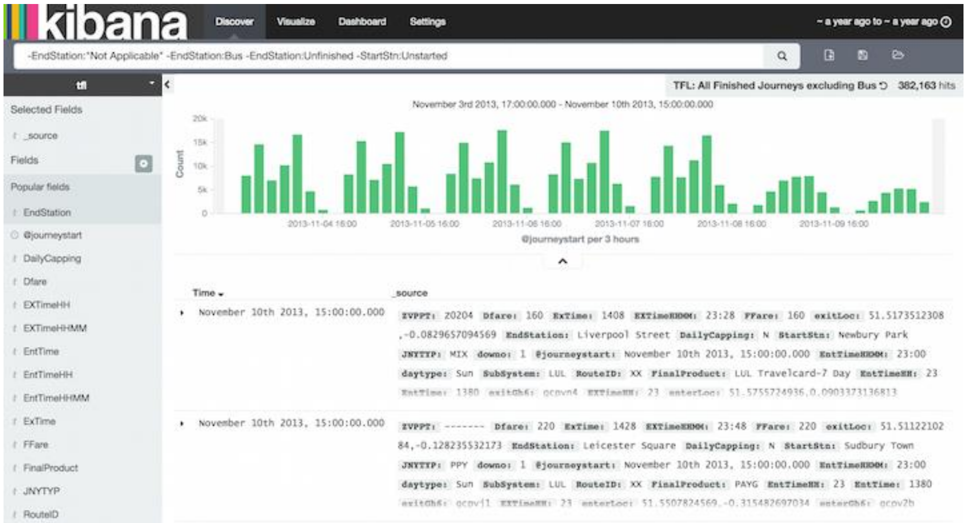
Elastic Search is built on top of high performance open source search engine **Apache Lucene**. The documented oriented storage differs sharply from traditional table oriented RDBMS (Such as Oracle, MS SQL Server). With document oriented data storage, data is stored as structured JSON (JavaScript Object Notation) documents. **Every field is indexed by default**. This is why the search speed is incredible.

The architecture of Elastic Search favors distribution, meaning you can scale your Elastic Search infrastructure massively and seamlessly. The infrastructure is resilient for failures – it automatically relocates and balances resources. Note that you can install and maintain an Elastic Search infrastructure **in your premise**, or you can simply purchase a **SaaS based solution** where the infrastructure is hosted and managed by Elastic Search for you.

The user interface to perform search and analytics is **Kibana**, an open source **data visualization** platform. With its intuitive, clean and responsive interface Kibana makes searching for data a joy. **Logstash**, another open source tool does the heavy lifting of consuming the logs from various systems and sends them to Elastic Search. Together, Elastic Search, LogStash and Kibana form the commonly known System **ELK**.

Finally a powerful **RESTFul API** armors developers to do every action imaginable using the data set. Client libraries are also available for many languages.

A sample screen from Kibana:



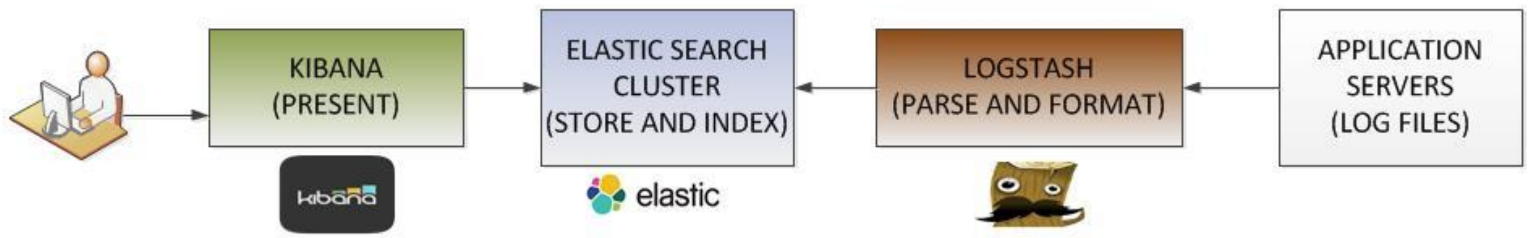
```
-EndStation:"Not Applicable" -EndStation:Bus -EndStation:Unfinished -StartStn:Unstarted
```

```
November 10th 2013, 15:00:00.000   ZVPPT: Z0204 Dfare: 160 ExTime: 1408 EXTimeHHMM: 23:28 FFare: 160 exitLoc: 51.5173512308
                                   ,-0.0829657094569 EndStation: Liverpool Street DailyCapping: N StartStn: Newbury Park
                                   JNYTYP: MIX downo: 1 @journeystart: November 10th 2013, 15:00:00.000 EntTimeHHMM: 23:00
                                   daytype: Sun SubSystem: LUL RouteID: XX FinalProduct: LUL Travelcard-7 Day EntTimeHH: 23
                                   EntTime: 1380 exitGh6: gcpvn4 EXTimeHH: 23 enterLoc: 51.5755724936,0.0903373136813

November 10th 2013, 15:00:00.000   ZVPPT: ------- Dfare: 220 ExTime: 1428 EXTimeHHMM: 23:48 FFare: 220 exitLoc: 51.51122102
                                   84,-0.128235532173 EndStation: Leicester Square DailyCapping: N StartStn: Sudbury Town
                                   JNYTYP: PPY downo: 1 @journeystart: November 10th 2013, 15:00:00.000 EntTimeHHMM: 23:00
                                   daytype: Sun SubSystem: LUL RouteID: XX FinalProduct: PAYG EntTimeHH: 23 EntTime: 1380
                                   exitGh6: gcpvj1 EXTimeHH: 23 enterLoc: 51.5507824569,-0.315482697034 enterGh6: gcpv2b
```

Image Source: www.elastic.co

The plumbing – How does it work?

ELK Logical Architecture



© Karun Subramanian

Logstash is the workhorse that collects the log files from Application Servers, parses them, formats them and sends them to Elastic Search. Elastic Search stores and indexes the data, which is presented by Kibana. The end users access Kibana Web Interface to view the data.

Let us explore the flow a bit more:

1. Application Servers to LogStash

   Typically, a Logstash Forwarder component is installed on the Application Servers. This piece of software is responsible for collecting the logs based on the configuration that you setup via a configuration file. Logstash Forwarder is an instance of Logstash whose input is Application Log files and output is Logstash Indexer (The 2nd block from the right in the above diagram).

Optionally you can introduce a Message queueing component between the Application Servers and the LogStash Indexer. Rabbit MQ is a popular choice in ELK implementations.

2. Logstash to Elastic Search Cluster

   Logstash (indexer) parses and formats the log (based on the log file content and the configuration of LogStash) and feeds Elastic Search Cluster. The protocol used is a **Native Elastic Search Transport.** Elastic Search consumes data from Logstash and creates Indexes (for log files, typically a date-based index is created by Elastic Search).
3. Kibana

   Kibana is a data visualization platform that is the presentation layer in the ELK stack. It consumes data from Elastic Search Indexes. A user accesses Kibana interface via a web browser.

How can I get started, really?

Getting started with Elastic Search is straight forward, but may require some time and effort. The basic software components are completely free and ready for you to download.

You will need to download the following three components from https://www.elastic.co/downloads

Elasticsearch

Logstash

Kibana

Installation process

Elasticsearch:

Installing Elasticsearch is simply downloading the zip/tar file into a directory of your choice and unzipping it. If you are planning on a cluster that spans more than one physical node, you will have to install the software on each physical node.

Almost everything in elastic search is preconfigured. For POC (Proof of concept) purposes, you should have no trouble simply starting up Elastic Search without any custom configuration. Use the following command to startup elastic search

```
<eshome>/bin/elasticsarch
```

(On windows platform, *elasticsearch.bat*)

The primary configuration file for elastic search is *elasticsearch.yml* under *conf* directory.

The log file <elastic search clustername.log> is found under logs directory.

By default Elastic Search will listen on port 9200 for http requests and port 9300 for requests from Elastic Search nodes (Log stash will be considered a node in elastic search).

A quick way to test your installation is to access Elasticsearch using curl

curl –XGET http://<localhost>:9200/

You should see a JSON output with the cluster name and few other metrics.

Logstash:

Installing Logstash is simply downloading the zip/tar file into a directory of your choice and unzipping it. Unlike ElasticSearch, you will have to minimally configure the logstash configuration file before starting up logstash.

The logstash configuration file (you must create one) must have the following entries:

a. Input block (required)
b. Output block (required)
c. Filter block (optional)

For example:

```
input {

    file {

        path => "/path/to/mylog"

        start_position => beginning

    }

}
```

An output block might look like this for an elastic search cluster

```
output {
    elasticsearch {}
    stdout {}
}
```

You can start logstash using the following command *bin/logstash agent –f <logstash config file>*

Once you start logstash, you can check elastic search log file to make sure the logstash instance gest added as a Node.

Kibana:

Download kibana binaries from https://www.elastic.co/downloads/kibana. Unzip the zip/tar file and update config/kibana.yml to point to your elastic search instance. For example,

```
# The Elasticsearch instance to use for all your queries.

elasticsearch_url: http://localhost:9200
```

Then simply start kibana with the command bin/kibana (bin\kibana.bat in Windows). Kibana is now listening on 5601. You can access the Kibana interface via http://<hostname>:5601.

Note: In windows environment, you can also install Kiaban as Windows Service by using NSSM.

The Search

There are few ways you can interact with Elastic search.

CURL:

The simplest way is to use curl command. The format of the curl command is:

```
curl -X<VERB> '<PROTOCOL>://<HOST>:<PORT>/<PATH>?<QUERY_STRING>' -d '<BODY>'
```

VERB: Can be one of GET, POST, PUT, HEAD, or DELETE

HOST:PORT : Your Elastic Search Cluster. Default port is 9200

PATH: This is the API endpoint. The API endpoints generally begin with the letter _

QUERY STRING: Optional query string. For example *?pretty* will display output Json response in easily readable format

BODY: JSON body (if required).

For example, if you want to know the number of nodes in your elastic search cluster, you would use:

```
curl -XGET http://localhost:9200/_cat/nodes
```

For searching strings, you would use the API endpoint "_search"

Java API:

Elastic Search provides two built-in clients to use if you are a java shop.

A node client joins elastic search cluster as a non-data node and accesses the data in the cluster

A transport client accesses Elastic search by sending requests to a remote elastic search cluster.

Both node and transport client use port 9300 of Elastic search cluster with a native transport. Note that the Elastic Search cluster nodes talk among themselves using this port as well.

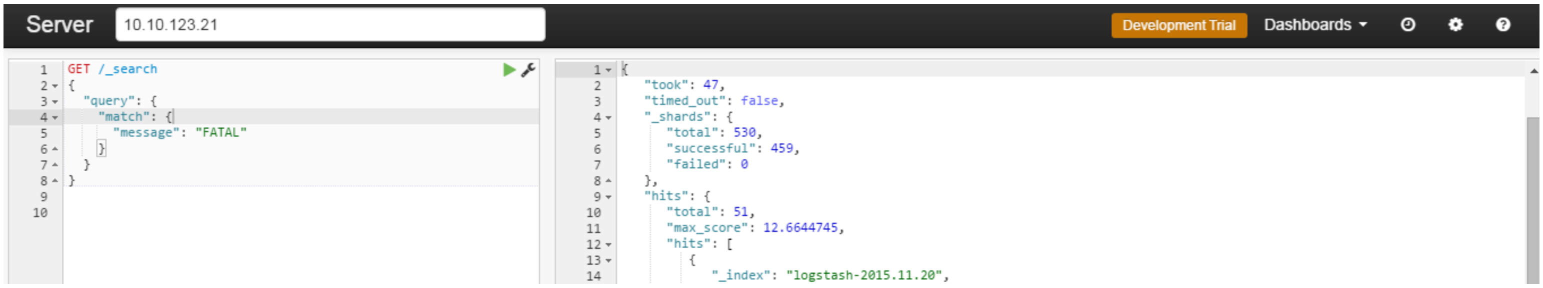
SENSE:

SENSE is a browser based tool that comes with the plugin MARVEL.

Note: Elastic search uses plugin architecture, meaning there are tons and tons of plugin available for you to simply download,install and start to use. Marvel is one such plugin that is extremely useful. It is used to monitor and troubleshoot the Elastic search cluster.

With Sense dashboard, you can send curl commands to elastic search cluster and obtain the response.

Note that sense uses a compact version of curl command (i.e it leaves out the host:port, etc)



```
GET /_search
{
  "query": {
    "match": {
      "message": "FATAL"
    }
  }
}
```

```
{
  "took": 47,
  "timed_out": false,
  "_shards": {
    "total": 530,
    "successful": 459,
    "failed": 0
  },
  "hits": {
    "total": 51,
    "max_score": 12.6644745,
    "hits": [
      {
        "_index": "logstash-2015.11.20",
```

KIBANA:

This is the obvious choice for querying elastic search. With the intuitive graphs and the powerful DSL querying facility there is nothing you can’t find with Kibana. Kibana will be covered in detail in another article.

Most important Plugins

Elastic Search is built with plugin architecture. This opens door for numerous plugins that are available for you to download and use. While some plugins are not free, lot of them are.

Three important plugins are:

1. Marvel – Used for monitoring Elastic Search
2. Head – Used for managing Elastic Search indexes
3. Shield – Security for Elastic Search

Installing plugins is a snap with Elastic Search. For installing Marvel, use the following command

```
bin/plugin -i elasticsearch/marvel/latest
```

To install Head

```
bin/plugin -i mobz/elasticsearch-head
```

To install Shield: (Note: Shield is not a free plugin)

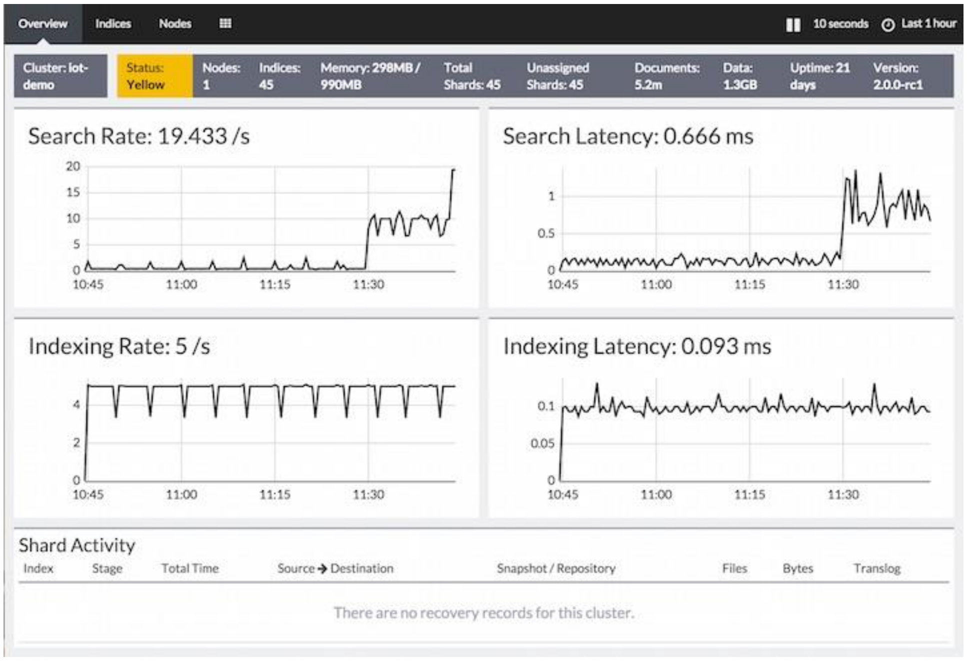
```
bin/plugin -i license

bin/plugin -i shield
```

To see the list of plugins installed in your system

```
bin/plugin -l
```

Marvel:

You can monitor the health of the Elasticsearch cluster using Marvel. For example, you can find out how much disk space is left in the volume where indexes are stored. You can also check the JVM heap utilization of Elasticsearch JVM



Cluster: lot-demo | Status: Yellow | Nodes: 1 | Indices: 45 | Memory: 298MB / 990MB | Total Shards: 45 | Unassigned Shards: 45 | Documents: 5.2m | Data: 1.3GB | Uptime: 21 days | Version: 2.0.0-rc1

Search Rate: 19.433 /s

Search Latency: 0.666 ms

Indexing Rate: 5 /s

Indexing Latency: 0.093 ms

Shard Activity

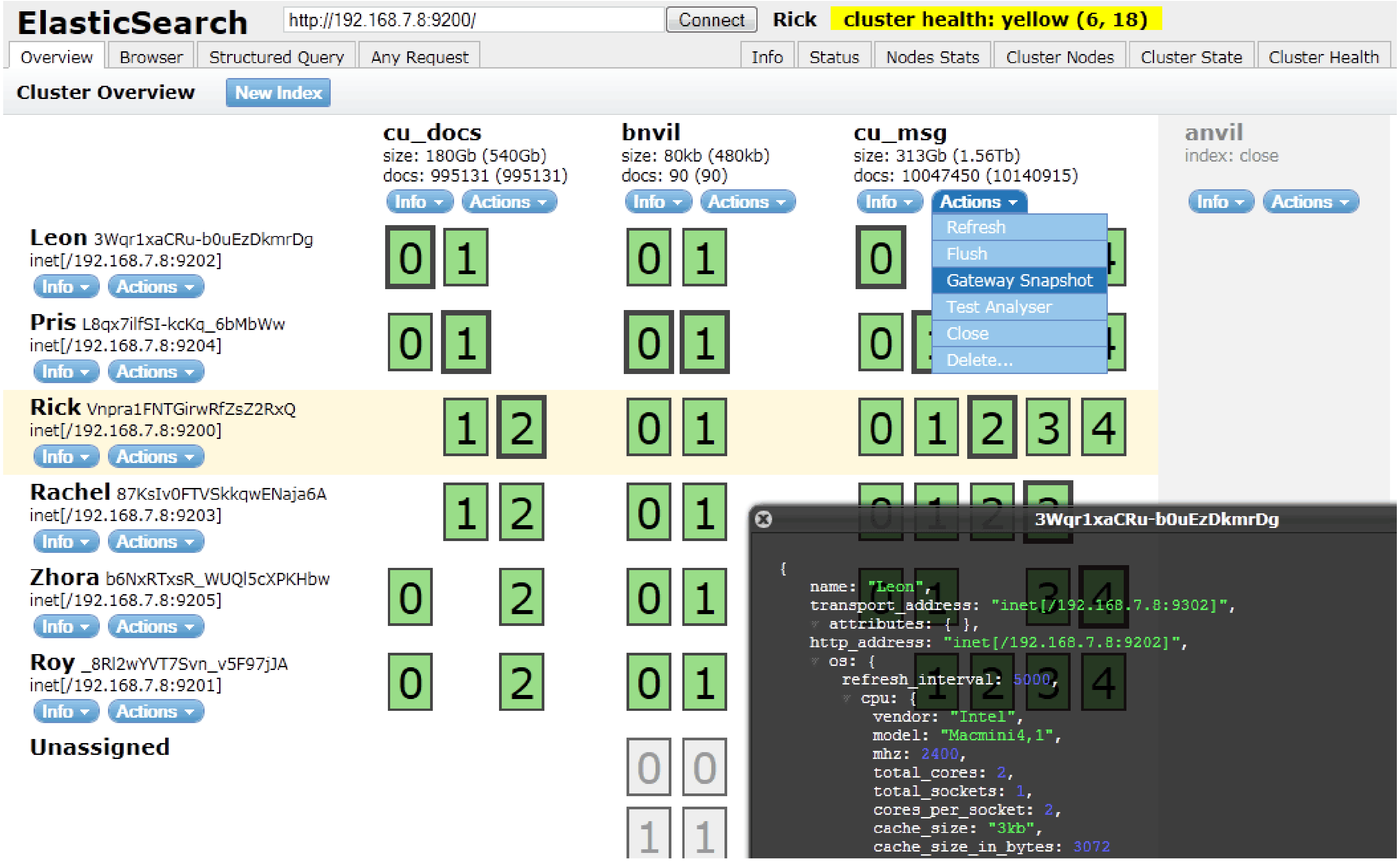
Index	Stage	Total Time	Source → Destination	Snapshot / Repository	Files	Bytes	Translog

There are no recovery records for this cluster.

Image Source: https://www.elastic.co/guide/en/marvel/current/introduction.html

Head:

Most common use case of Head is to manage the indexes. You can view and delete indexes as you wish.



```
{
  name: "Leon",
  transport_address: "inet[/192.168.7.8:9302]",
  attributes: { },
  http_address: "inet[/192.168.7.8:9202]",
  os: {
    refresh_interval: 5000,
    cpu: {
      vendor: "Intel",
      model: "Macmini4,1",
      mhz: 2400,
      total_cores: 2,
      total_sockets: 1,
      cores_per_socket: 2,
      cache_size: "3kb",
      cache_size_in_bytes: 3072
```

Image source: https://mobz.github.io/elasticsearch-head/

Shield

Shield protects the data stored by Elastic Search. A basic username/password security is deployed. But if/when needed, advanced security configurations such as encryption, ip filtering, RBAC (Role based access control) and auditing can be deployed as well.

Where to go next?

Undoubtedly, the most useful resource for learning Elastic Search is https://www.elastic.co/learn. There you will find tons of tutorials, videos, articles and even the full content of the book 'Elastic Search – the definitive guide'.

There are also meet-up groups available for you to join. For a local meeting, visit meetup.com

Elastic Search organizes frequent seminars and webminars. And then there is the Elastic ON yearly conference which hosts numerous training sessions. For training schedule, you can visit http://training.elastic.co/

But by far the most effective way to learn Elastic Search is to simply download, install and start collecting logs. It IS easy. Give it a shot and you will be glad you did.

Happy Monitoring !!!