# **Spring Boot Microservices + ELK(Elasticsearch, Logstash, and Kibana)**

The ELK Stack consists of three open-source products –

* Elasticsearch,
* Logstash, and
* Kibana from Elastic.

Elasticsearch is a NoSQL database that is based on the **Lucene search engine**.

Logstash is a log pipeline tool that accepts inputs from various sources, executes different transformations, and exports the data to various targets. It is a dynamic data collection pipeline with an extensible plugin ecosystem and strong Elasticsearch synergy

Kibana is a visualization UI layer that works on top of Elasticsearch.

These three projects are used together for log analysis in various environments. So Logstash collects and parses logs, Elastic search indexes and store this information while Kibana provides a UI layer that provide actionable insights.

The ELK Stack consists of three open-source products - Elasticsearch, Logstash, and Kibana from Elastic.

* Elasticsearch is a NoSQL database that is based on the Lucene search engine.
* Logstash is a log pipeline tool that accepts inputs from various sources, executes different transformations, and exports the data to various targets. It is a dynamic data collection pipeline with an extensible plugin ecosystem and strong Elasticsearch synergy
* Kibana is a visualization UI layer that works on top of Elasticsearch.
* Diagram

  Description automatically generated

**Use Cases-**

Consider you have a single application running and it produces logs. Now suppose you want analyze the logs generated. One option is to manually analyze them. But suppose these logs are large, then manually analyzing them is not feasible.

Suppose we have multiple Application running and all these applications produce logs. If we have to analyze the logs manually, we will need to go through all the log files. These may run into hundreds.

We can use ELK here to analyze the logs more efficiently and also using more complex search criterias. It provides log aggregation and efficient searching.

Diagram

Description automatically generated

Step to setup:

**Step1:** First We need to download “Elastic Search”, “Kibana” and “Logstash” bundles as a zip file.

**Step2**: Need to extract in any location (Recommended is C: drive)

**Step-3:** First goto extracted location of Elastic search Elastic\_Search\_Home\bin folder, Here we need to run “elasticsearch.bat” command from command prompt.

**Step-4:** After successful run the elasticsearch server will start on default port 9200.

Graphical user interface, text

Description automatically generated

**Step-5:** We need to setup kibana first for that we need to modify kibana.yml file which will be under “Kibana\_Home/config” folder. Here we need to uncomment elasticsearch.url: <http://localhost:9200>

**Step-6:** Now we need to run kibana using “kibana.bat” file from command prompt and after successful the kibana will start on default port: 5601

Graphical user interface, text, website

Description automatically generated

Here we need to provide elastic token or, credentials.

**Step-7:** Now we need to configure logstash to retrieve the logfiles from our application and that will pass to elastic search. So for this we need to create one configuration file as: “logstash.conf”

**Step-8**: Since we are working on java application and we need to configure log file to a particular location so that logstash can read it successfully. So we will keep our application ready with configure log file in a particular location. Eg: logging.file=D:/logs/elk-spring-app.log

**Step-9**: Now we need to configure logstash.conf file for logstash setup and provide the log file location in yml file.

**Eg:**

|  |
| --- |
| input {  file {  type => "java"  path => "D:/logs/elk-spring-app.log"  codec => multiline {  pattern => "^%{YEAR}-%{MONTHNUM}-%{MONTHDAY} %{TIME}.\*"  negate => "true"  what => "previous"  }  }  }    filter {  #If log line contains tab character followed by 'at' then we will tag that entry as stacktrace  if [message] =~ "\tat" {  grok {  match => ["message", "^(\tat)"]  add\_tag => ["stacktrace"]  }  }    }    output {    stdout {  codec => rubydebug  }    # Sending properly parsed log events to elasticsearch  elasticsearch {  hosts => ["localhost:9200"]  }  } |

**Step-10:** Now in springboot application I have created two apis, one for /info and second for /exception to print info and error messages respectively.

**Step-11:** Now we will start our application and hit these two apis and will see the console and logs respectively for our application.

Graphical user interface, text, application, email

Description automatically generated

**Step-12:** Now our application is ready with kibana configuration and all related logs we can see in Kibana UI Dashboard.

Graphical user interface, text, application, Teams

Description automatically generated