



# ELK + Spring Boot: A Guide to Local Configuration

Setting up a centralized logging solution using the ELK stack on your local machine.



Denuwan Himanga Hettiarachchi · [Follow](#)

Published in Cloud Native Daily · 7 min read · May 16, 2023

152 2

Hi Folks,

In this tutorial, I will guide you through the process of setting up a centralized logging solution using the ELK stack on your local machine. While the production-ready configuration may be more complex and different, it is still valuable to familiarize yourself with the components and configurations by setting up the ELK stack locally.

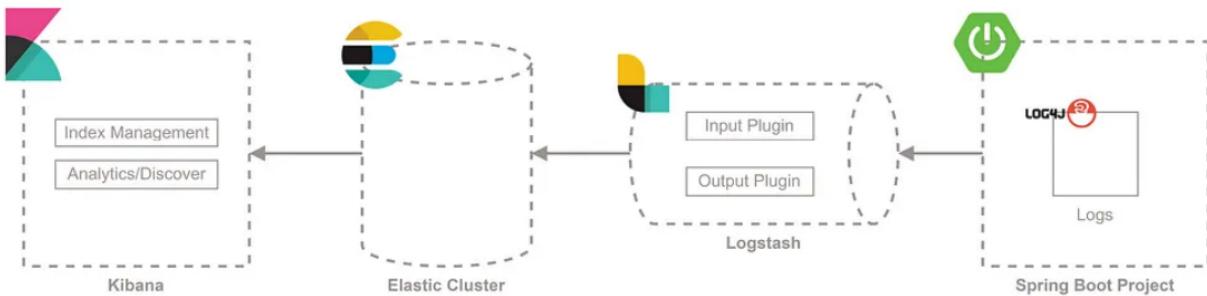
## What is ELK?\*

ELK is an acronym that stands for Elasticsearch, Logstash, and Kibana. It is a powerful open-source software stack used for centralized logging, log analysis, and visualization. Each component of ELK serves a specific purpose:

- **Elasticsearch:** *It is a distributed search and analytics engine that stores and indexes data. Elasticsearch allows for fast and efficient searching, aggregating, and analyzing of large volumes of data.*
- **Logstash:** *It is a data processing pipeline that ingests, filters, and transforms data from various sources before sending it to Elasticsearch for indexing. Logstash can handle different types of data, such as logs, metrics, and other event data.*
- **Kibana:** *It is a web-based data visualization and exploration tool that works in conjunction with Elasticsearch. Kibana provides a user-friendly interface for searching, analyzing, and visualizing data stored in Elasticsearch. It offers various visualizations, dashboards, and tools to help gain insights from the data.*

Together, these three components form the ELK stack, which is widely used for log management, monitoring, and troubleshooting in various applications and systems.

The following high-level architecture diagram illustrates the placement of components within a system.



ELK + Spring boot Component Diagram

## 1. Build Springboot Application

1. Go to [Spring initializr](#) and select your preferable Language, Project & Java version (> v1.8 is recommended).
2. Add Web dependence to your project.
3. Generate the project, in my demo the generated maven project POM file is mentioned below.

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/
modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.0.6</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.elk.demo</groupId>
<artifactId>com-elk-demo</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>elk-demo</name>
<description>Demo project for Spring Boot</description>
<properties>
<java.version>17</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>

```

4. Since you add the WEB dependency, loggers are pre-configured in your project.

But by default, spring boots are logs only to the console. So you have to add the following configuration in the `application.property` file to write logs to a file.

```
logging.file.path=.
```

5. Implement Controller class and create an endpoint to write a log on both the console & log file.

```
package com.elk.controller;

import org.apache.logging.log4j.Level;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class Controller {

    private static final Logger LOG = LogManager.getLogger(Controller.class);

    @GetMapping(value = "/echo")
    public String echoMessage() {
        LOG.log(Level.INFO, "Echo Triggered");
        return "Echo Triggered";
    }
}
```

6. spinning up the spring boot application by executing the below command.

```
mvn clean install spring-boot:run
```

## 2. Set up Elastic Search

1. Download and unzip Elasticsearch.
2. Go to `elasticsearch-<version>-<os>-<system-type>\elasticsearch-<version>\bin`
3. Open CMD and navigate to the above-mentioned directory then execute the following command.

```
elasticsearch.bat
```

4. If you're using Windows OS, you may need to allow access blocked due to Windows Defender Firewall.

```

[2023-05-15T23:08:00,111][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [file-history-ilm-policy]
[2023-05-15T23:08:00,388][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [fleet-file-data-ilm-policy]
[2023-05-15T23:08:00,491][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [fleet-files-ilm-policy]
[2023-05-15T23:08:00,694][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [deprecation-indexing-ilm-policy]
[2023-05-15T23:08:11,621][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [fleet-actions-results-ilm-policy]
[2023-05-15T23:08:15,833][INFO ][l.e.h.n.s.TransportPutLifecycleAction] [DENUMAN-LAPTOP] Node [[DENUMAN-LAPTOP][msv-nGwY1yLfgqasHTA]] is selected as the current health node.
[2023-05-15T23:08:18,013][INFO ][o.e.l.LicenseService] [DENUMAN-LAPTOP] license [0bb06580-H123-0723-9aa0-311b179782ae] mode [basic] - valid
[2023-05-15T23:08:18,015][INFO ][o.e.x.s.Realms] [DENUMAN-LAPTOP] license mode is [basic], currently licensed security realms are [reserved,reserved,file/default_file/native/default_native]

```

Allow access blocked due to Windows Defender Firewall

5. Scroll down a bit, you can see the `Elasticsearch security features have been automatically configured!` message on the CMD output.

```

[2023-05-15T23:08:04,111][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [file-history-ilm-policy]
[2023-05-15T23:08:04,388][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [fleet-file-data-ilm-policy]
[2023-05-15T23:08:04,491][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [fleet-files-ilm-policy]
[2023-05-15T23:08:04,694][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [deprecation-indexing-ilm-policy]
[2023-05-15T23:08:11,621][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DENUMAN-LAPTOP] adding index lifecycle policy [fleet-actions-results-ilm-policy]
[2023-05-15T23:08:15,833][INFO ][l.e.h.n.s.TransportPutLifecycleAction] [DENUMAN-LAPTOP] Node [[DENUMAN-LAPTOP][msv-nGwY1yLfgqasHTA]] is selected as the current health node.
[2023-05-15T23:08:18,013][INFO ][o.e.l.LicenseService] [DENUMAN-LAPTOP] license [0bb06580-H123-0723-9aa0-311b179782ae] mode [basic] - valid
[2023-05-15T23:08:18,015][INFO ][o.e.x.s.Realms] [DENUMAN-LAPTOP] license mode is [basic], currently licensed security realms are [reserved,reserved,file/default_file/native/default_native]

```

Elasticsearch security features have been automatically configured

6. Copy the following values and save them.

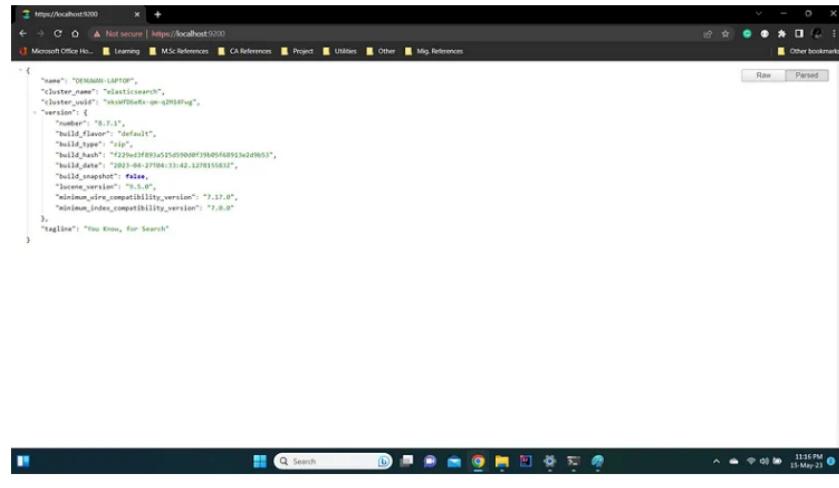
- Password for the elastic user
- Enrollment token



7. To check the status of the Elasticsearch cluster connection, hit

<https://localhost:9200> on your browser and provide the Username as "elastic" and use the password you saved in step #6.

If your cluster is working properly you should get the below output.



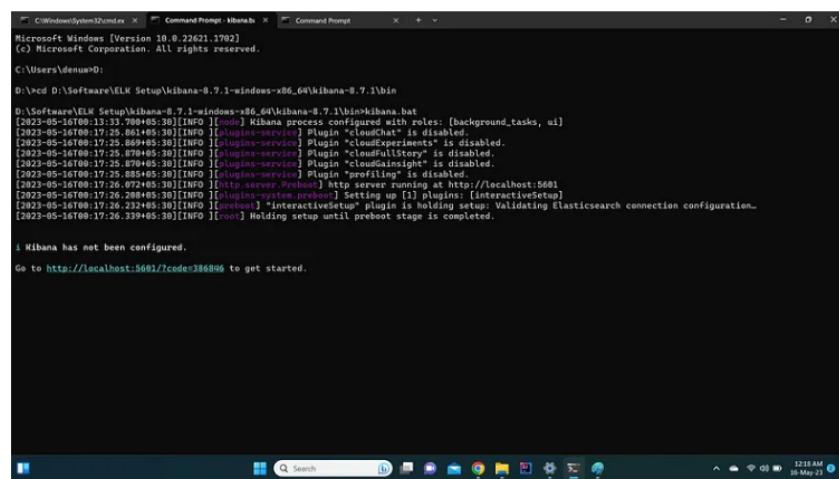
Check the ElasticSearch Cluster

### 3. Set up Kibana

1. Download and unzip Kibana.
2. Go to `elasticsearch-<version>-<os>-<system-type>\elasticsearch-<version>\bin`.
3. Open CMD and navigate to step #2 mentioned directory, then execute the following command.

kibana.bat

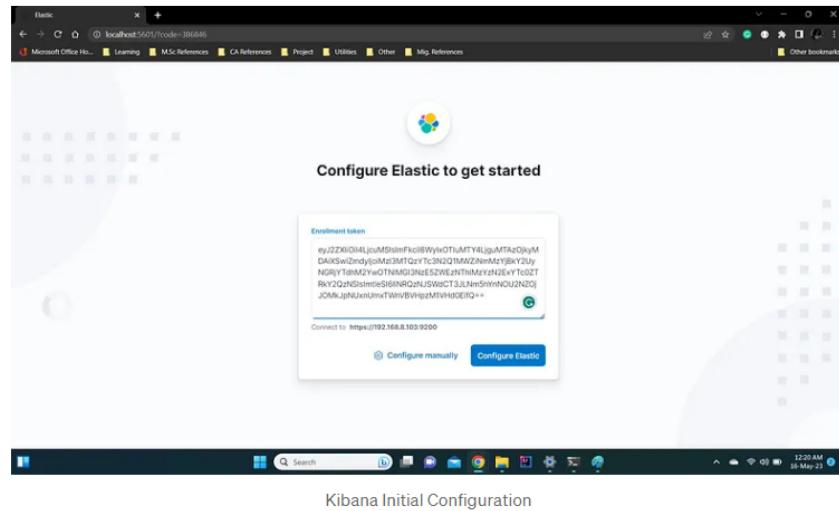
4. Once the Kibana spinning up correctly, the URL should be available on CMD, copy it.



Kibana Terminal with URL

5. Go to the URL copied in step #4.

6. Provide the Enrollment token saved in “Set up Elastic Search” step #6.



7. Click Configure Elastic.

## 4. Set up Logstash

1. Download and unzip Logstash.
2. Go to `logstash-<version>-<os>-<system-type>\logstash-<version>\config`.
3. Open the `logstash-sample.conf` & replace the input & output plugins as mentioned below.

```
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
    file {
        path => "<Log File Full Path>"
        start_position => "beginning"
    }
}

output {
    stdout {
        codec => rubydebug
    }
    elasticsearch {
        hosts => ["https://localhost:9200"]
        ssl_certificate_verification => false
        ssl => true
        index => "elkdemoindex"
        user => "elastic"
        password => "<Elastic Search Password>"
    }
}
```

4. Replace the `<Log File Full Path>` with your log file location. (Make sure to use forward slashes).

5. Replace `<Elastic Search Password>` with the password, you saved in “Set up Elastic Search” step #6.

6. Go to the `logstash-<version>-<os>-<system-type>\logstash-<version>\bin` directory and spinning up the Logstash using the below command.

```
logstash.bat -f ./config/logstash-sample.conf
```

## 5. Verify the ELK + Spring boot Flow

### 1. Check the Logstash to Spring boot Integration

In “Set up Logstash” step #3, we set parameters in the output plugin, `stdout { codec => rubydebug }`, this will allow us to check the integration between the Springboot application & Logstash.

If both parties are connected as expected, you can see spring boot application logs in JSON format on the Logstash terminal like this.

```
{
    "@version" => "1",
    "log" => {
        "file" => {
            "path" => "D:/Projects/com-elk-demo/com-elk-demo/spring.log"
        }
    },
    "host" => {
        "name" => "Denuwan-Laptop"
    },
    "event" => {
        "original" => "2023-05-16T01:34:54.503+05:30  INFO 19908 --- [main] o.s.
    },
    "@timestamp" => 2023-05-16T05:22:37.881555300Z,
    "message" => "2023-05-16T01:34:54.503+05:30  INFO 19908 --- [main] o.s.b.
}
```

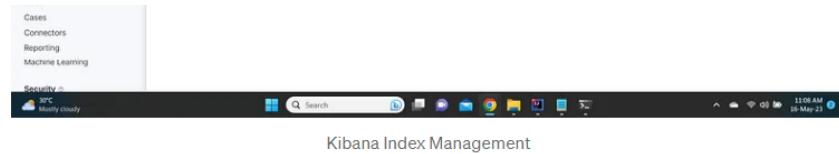
### 2. Configure Kibana Dashboard

#### 1. Go to <http://localhost:5601/>

2. Click the burger button on the left side and go to `Management > Stack Management`.

3. Inside the Management portal, go to `Data > Index Management`, if the Logstash to Elastic search connection working properly you should get the `elkdemoindex` value on the indices list.

The screenshot shows the Elasticsearch Index Management interface. On the left, there's a navigation sidebar with sections for Management, Ingest, Data, Index Management, and Alerts and Insights. The main area is titled "Index Management" and has tabs for Indices, Data Streams, Index Templates, and Component Templates. Under the Indices tab, it says "Update your Elasticsearch indices individually or in bulk. Learn more." There are checkboxes for "Include rollup indices" and "Include hidden indices". Below that is a search bar and a table with columns: Name, Health, Status, Primaries, Replicas, Docs count, Storage size, and Data stream. Two indices are listed: "elkdemo" (Health: yellow, Status: open, Primaries: 1, Replicas: 1, Docs count: 11509, Storage size: 2.26mb) and "elkdemoindex" (Health: yellow, Status: open, Primaries: 1, Replicas: 1, Docs count: 16, Storage size: 87.53kb). At the bottom right of the table, there's a "Reload indices" button.



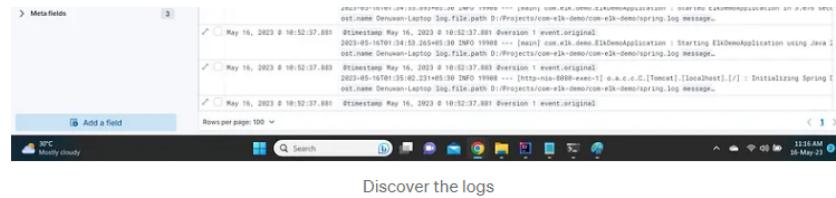
4. Again click the burger button on the left side and go to `Analytics > Discover`.

5. You can create a new data view as mentioned below.

Create a new data view in Kibana > Discover

Set data view to Kibana in Kibana > Discover

6. Once you create a new data view, navigate to the newly created data view and check the logs.



Congratulations! Now you are successfully configured ELK + Spring boot stack on your local.

Happy coding!

\* "What is ELK?" the theoretical part of the content is generated by ChatGPT and

## Further Reading:

### Deploying OpenTelemetry with Java: Your Guide

Learn to deploy OpenTelemetry in Java to collect services data. Use Jaeger Helios and other tools and troubleshoot with...

[gethelios.dev](https://gethelios.dev)



### Testing Microservices - Trace Based Integration Testing Example

Microservices architectures require a new type of testing. Here's why traditional testing fail and the new automated...

[gethelios.dev](https://gethelios.dev)



### OpenTelemetry Tracing: Everything you need to know

OpenTelemetry tracing is filling the gaps of traditional observability methods in microservices apps. Here's how it's...

[gethelios.dev](https://gethelios.dev)



### Microservices Monitoring: Cutting Engineering Costs and Saving Time

A few ways for leveraging Helios to save on engineering costs and dev time for a more resource-efficient organization...

[gethelios.dev](https://gethelios.dev)



### Serverless observability, monitoring, and debugging explained

Serverless troubleshooting requires E2E observability, through collecting trace data on top of logs and metrics- Here's...

[gethelios.dev](https://gethelios.dev)



## OpenTelemetry: A full guide

Learn all about OpenTelemetry OpenSource and how it transforms microservices observability and troubleshooting

gethelios.dev



## Kubernetes Monitoring with OpenTelemetry

Learn how to monitor Kubernetes using OpenTelemetry with real-time visibility and granular error data - Reduce MTTR by...

gethelios.dev



Microservices

Distributed Systems

Spring Boot

Elk Stack

Observability

152

2

W+

U



## Written by Denuwan Himanga Hettiarachchi

138 Followers · Writer for Cloud Native Daily

Blogger, Freelance Writer, and Tech Lead based in Colombo, SL.

Follow

✉+

## More from Denuwan Himanga Hettiarachchi and Cloud Native Daily



Denuwan Himanga Hettiarachchi

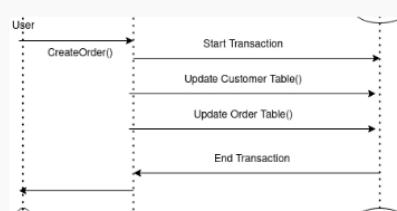
### Build Gen-AI LLM RAG API Ecosystem on a Node Server

Build your own Generative AI Large Language Model API ecosystem to get PDF document...

May 23

3

W+



Crishantha Nanayakkara in Cloud Native Daily

### Microservices Patterns: The Saga Pattern

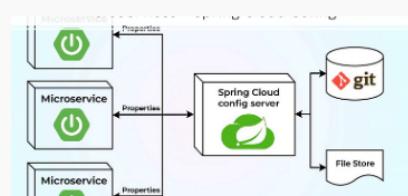
Microservices Pattern Series — Part 04: Some insights into ACID Transactions, 2PC...

Jun 15, 2023

400

3

W+



 Anji... in Cloud Native Daily

## Spring Cloud Config: Externalizing the Configurations From Your...

A deep dive into the Spring Cloud Config and how we can leverage it to externalize...

May 7, 2023  19



 Denuwan Himanga Hettiarachchi

## Maintain the data versioning info with Spring Data—Envers

The smart way to maintain the history of data in your Spring Boot project.

Feb 8, 2020  71



[See all from Denuwan Himanga Hettiarachchi](#)

[See all from Cloud Native Daily](#)

## Recommended from Medium

# elasticsearch



 Nithidol Vacharotayan

## Spring boot 3 Logback and Logstash integrated with...

Elasticsearch monitors, logs, collects, and analyses text data. The developer can use...

 Jul 5  5



 Abhinav Sonkar in ITNEXT

## Distributed Tracing with Spring Boot 3—Micrometer vs...

Spring Boot 3 recommends Micrometer for Distributed Tracing. But can OpenTelemetry...

 Jan 20  238  1



## Lists



### Staff Picks

734 stories • 1302 saves



### Natural Language Processing

1707 stories • 1274 saves



Szilárd Mátis in Level Up Coding

## Monitor Java metrics with Prometheus and Grafana

Use predefined Grafana dashboards for monitoring your application



aneesh kumar

## Mastering API Gateway with Spring Boot and Spring Cloud...

Created using Canva

Mar 29 169



6d ago 7



Eric Cabrel Tiogo

## Implement JWT authentication in a Spring Boot 3 application

This blog post will show you how to implement a JWT authentication on Spring...

Mar 19 842 23



Debra Kerman in Level Up Coding

## Sanitize Textfields with Substitution

Replace unwanted user input with predetermined strings

May 28 46



See more recommendations

[Help](#) [Status](#) [About](#) [Careers](#) [Press](#) [Blog](#) [Privacy](#) [Terms](#) [Text to speech](#) [Teams](#)