****

Spring boot Microservices

Distributed Tracing with Jaeger, Elasticsearch and Kibana

&

Centralized Logging with ELK (Elasticsearch, Logstash and Kibana)

Author: IPTV BGC DEV Team

Email: IPTVBEPDEV@rsystems.com

Version: 1.0

Table of Contents

**[1.](#_Toc535318060)****[Introduction:](#_Toc535318060)** [3](#_Toc535318060)

[**2.** **Distributed tracing using Jaeger, Elasticsearch and Kibana** 3](#_Toc535318061)

[**3.** **Elasticsearch Installation** 3](#_Toc535318062)

[3.1. Windows: 3](#_Toc535318063)

[3.2. Linux: 3](#_Toc535318064)

[3.3. Elasticsearch URL and output: 3](#_Toc535318065)

[**4.** **Jaeger** 4](#_Toc535318066)

[4.1. Architecture: 4](#_Toc535318067)

[4.2. Jaeger Installation: 4](#_Toc535318068)

[4.2.1 Windows: 4](#_Toc535318069)

[4.2.2. Linux: 5](#_Toc535318070)

[4.3. Run Jaeger Components: 5](#_Toc535318071)

[4.3.1. Run Jaeger Collector: 5](#_Toc535318072)

[4.3.2. Run Jaeger Agent: 5](#_Toc535318073)

[4.3.3. Run Jaeger Query: 5](#_Toc535318074)

[4.4. Jaeger URL and Output: 5](#_Toc535318075)

[**5.** **Spring boot application and Jaeger Integration:** 6](#_Toc535318076)

[5.1. pom.xml Change: 6](#_Toc535318077)

[5.2 Java Code Change: 6](#_Toc535318078)

[5.2.1. Rest API tracing: 6](#_Toc535318079)

[5.2.2. HornetQ tracing: 7](#_Toc535318080)

[5.2.3. Kafka tracing: 9](#_Toc535318081)

[**6. Kibana Installation:** 9](#_Toc535318082)

[6.1. Kafka tracing: 9](#_Toc535318083)

[6.2 Jaeger service tracing log (jaeger-service-\*): 9](#_Toc535318084)

[6.3 Jaeger service tracing log (jaeger-span-\*): 10](#_Toc535318085)

[**7. Centralized logging using ELK (Elasticsearch, Logstash and Kibana):** 11](#_Toc535318086)

[**8. Logstash Installation:** 12](#_Toc535318087)

[**9. File Beat Installation:** 12](#_Toc535318088)

[**10. References:** 13](#_Toc535318089)

# **Introduction:**

Micro services is a distributed environment. A Single operation may involve multiple REST calls, messaging and other operations. If something goes wrong or analyzing the relation between the operations using distributed logs is difficult. In that case we need to use **Distributed tracing** and **Centralized logging** mechanism.

Jaeger, Elasticsearch and Kibana provides very efficient distributed tracing.

ELK (Elasticsearch, Logstash and Kibana) provides very efficient centralized logging.

# **Distributed tracing using Jaeger, Elasticsearch and Kibana**

Jaeger, Elasticsearch and Kibana provides very efficient distributed tracing.

# **Elasticsearch Installation**

Elastic Search Download URL: <https://www.elastic.co/downloads/elasticsearch>

## 3.1. Windows:

Step 1: Download elasticsearch-6.5.4.tar file.

Step 2: Extract the file.

Step 4: Run bin/elasticsearch (or bin\elasticsearch.bat on Windows)

## 3.2. Linux:

Step 1. Upload elasticsearch-6.5.4.tar.gz in /tmp

Step 2. Change installation directory

$cd /opt/local/java

Step 3. Extract .gz file

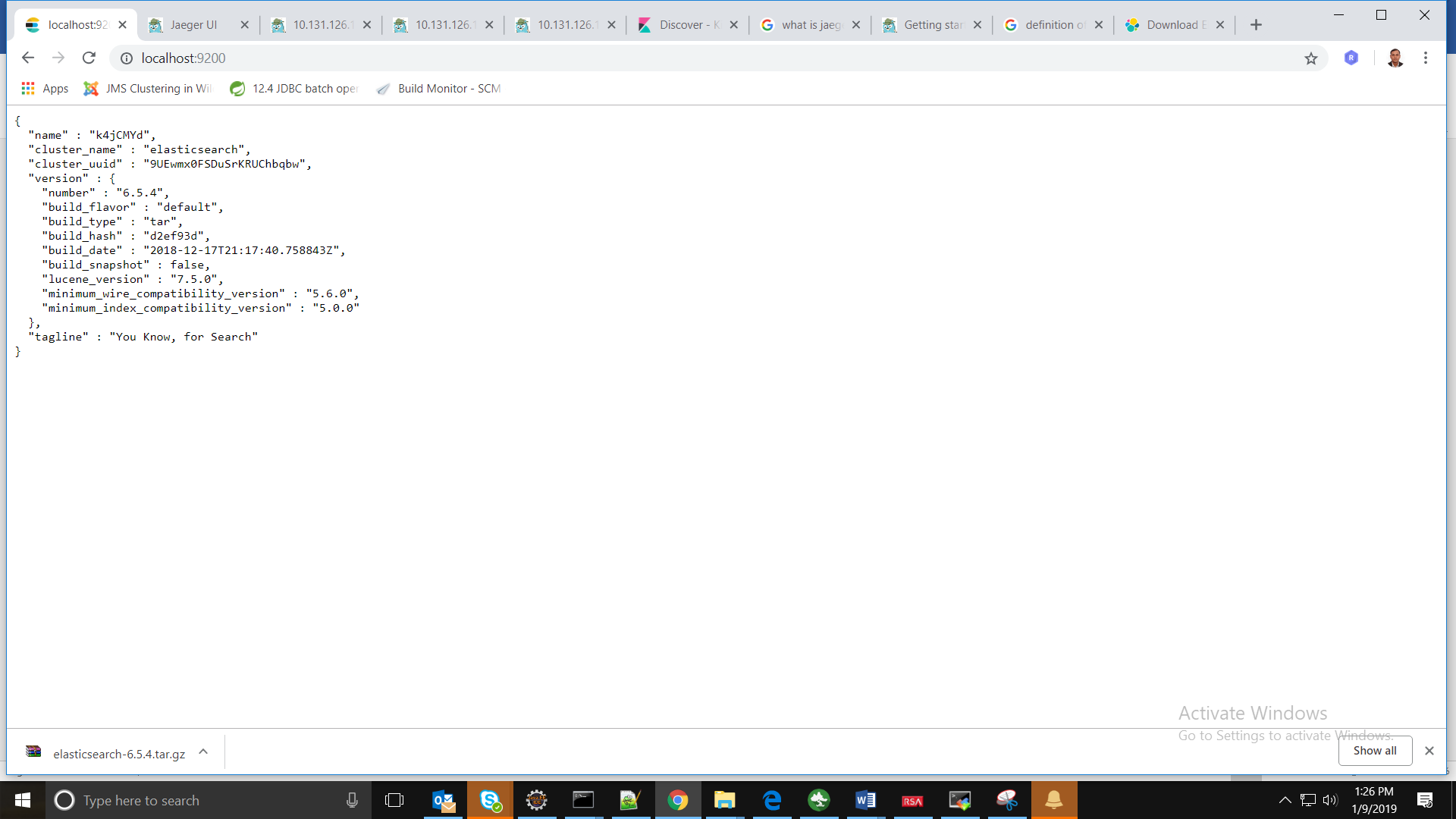
$tar -zxvf /tmp/elasticsearch-6.5.4.tar.gz

Step 4. cd elasticsearch-6.5.4/bin

Step 5. Run Jaeger

$. / elasticsearch.sh

## 3.3. Elasticsearch URL and output:

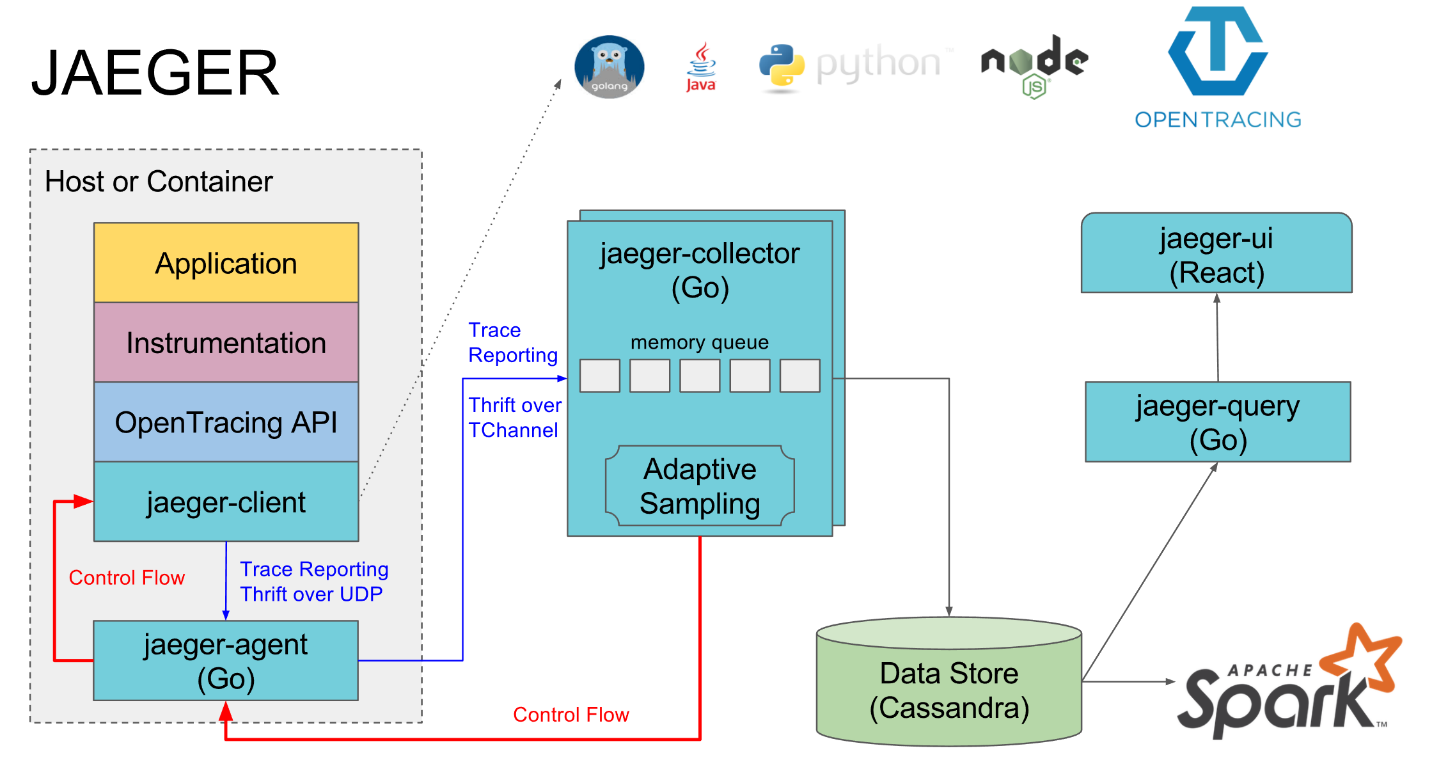
**Elastic After installation you are able to access the URL http://<Host IP>:9200**

# **Jaeger**

Jaeger is a distributed tracing system. It is released as open source [Uber Technologies](http://uber.github.io/). It is used for monitoring and troubleshooting micro services-based distributed systems, including:

* Distributed context propagation
* Distributed transaction monitoring
* Root cause analysis
* Service dependency analysis
* Performance / latency optimization

## 4.1. Architecture:



Supported storage back ends:

* [Cassandra 3.4+](https://www.jaegertracing.io/docs/1.8/deployment/#cassandra)
* [Elasticsearch 5.x, 6.x](https://www.jaegertracing.io/docs/1.8/deployment/#elasticsearch)
* [Kafka](https://www.jaegertracing.io/docs/1.8/deployment/#kafka)
* memory storage

## 4.2. Jaeger Installation:

Jaeger Download URL: <https://www.jaegertracing.io/download/>

## 4.2.1 Windows:

Step 1: Download jaeger-1.8.0-windows-amd64.tar file.

Step 2: Extract the file.

Step 3: Open the file location in Command prompt

## 4.2.2. Linux:

Step 1. Upload jaeger-1.8.0-linux-amd64.tar.gz in /tmp

Step 2. Change installation directory

$cd /opt/local/java

Step 3. Extract .gz file

$tar -zxvf /tmp/jaeger-1.8.0-linux-amd64.tar.gz

Step 4. cd jaeger-1.8.0-linux-amd64

## 4.3. Run Jaeger Components:

Go to Jaeger installed directory and run below command.

## 4.3.1. Run Jaeger Collector:

jaeger-collector.exe --span-storage.type=elasticsearch --es.server-urls=http://127.0.0.1:9200 –

log-level=debug --collector.http-port=14268 --collector.port=14267 --health-check-http-

port=14269

## 4.3.2. Run Jaeger Agent:

jaeger-agent.exe --collector.host-port=127.0.0.1:14267 --log-level=debug

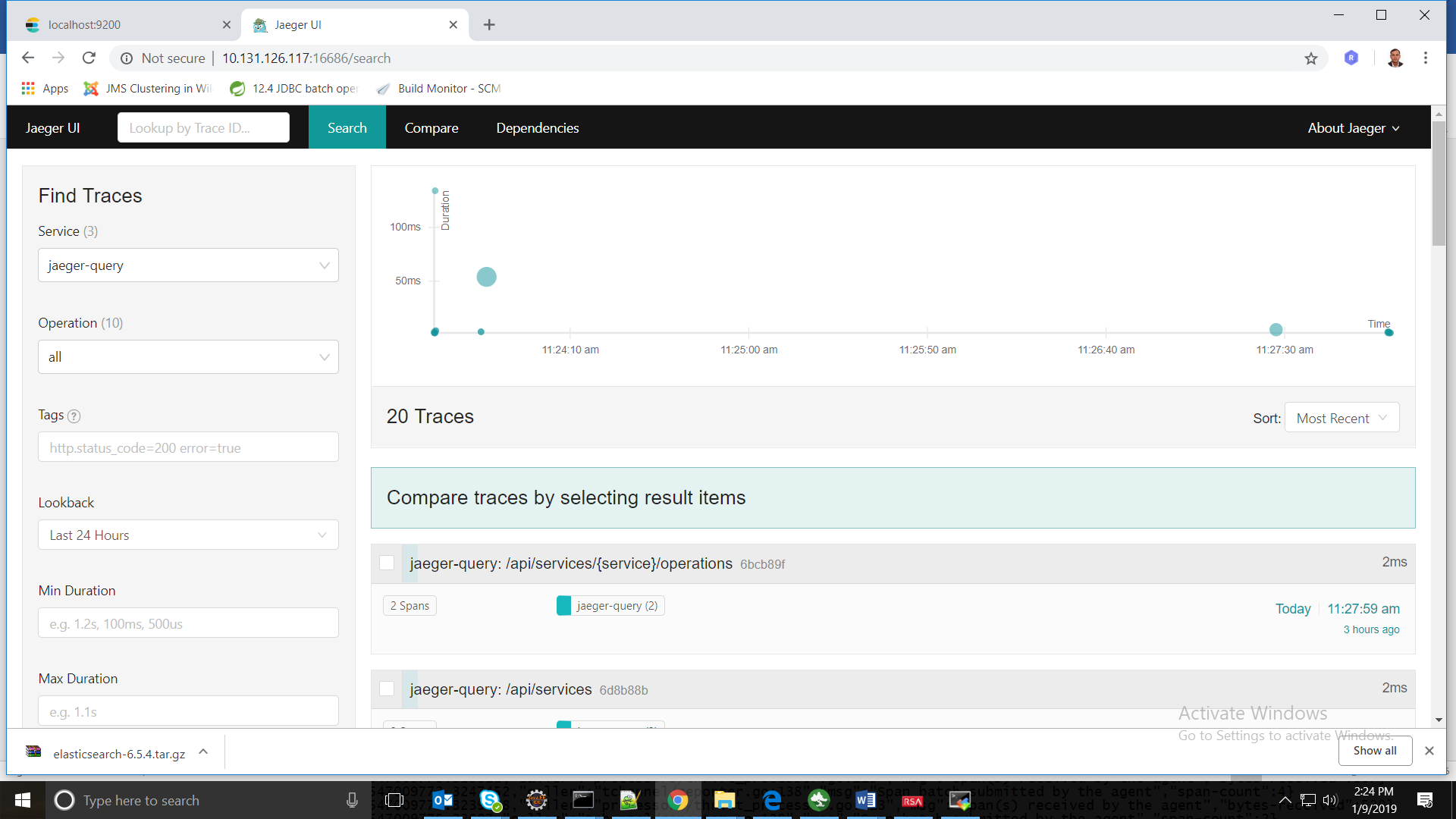
## 4.3.3. Run Jaeger Query:

jaeger-query.exe --span-storage.type=elasticsearch --es.server-urls=http://127.0.0.1:9200 --log-

level=debug --es.tags-as-fields.all=true

## 4.4. Jaeger URL and Output:

**After installation you are able to access the URL http://<Host IP>:16686**



# **Spring boot application and Jaeger Integration:**

## 5.1. pom.xml Change:

<dependency>

<groupId>io.opentracing.contrib</groupId>

<artifactId>

opentracing-spring-web-autoconfigure </artifactId>

<version>0.0.5</version>

</dependency>

<dependency>

<groupId>com.uber.jaeger</groupId>

<artifactId>jaeger-core</artifactId>

<version>0.20.0</version>

</dependency>

<dependency>

<groupId>io.opentracing.brave</groupId>

<artifactId>brave-opentracing</artifactId>

<version>0.20.0</version>

</dependency>

<dependency>

<groupId>io.zipkin.reporter</groupId>

<artifactId>zipkin-sender-okhttp3</artifactId>

<version>0.10.0</version>

</dependency>

<dependency>

<groupId>io.opentracing</groupId>

<artifactId>opentracing-api</artifactId>

<version>0.30.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/io.opentracing/opentracing-util -->

<dependency>

<groupId>io.opentracing</groupId>

<artifactId>opentracing-util</artifactId>

<version>0.30.0</version>

</dependency>

<dependency>

<groupId>io.opentracing</groupId>

<artifactId>opentracing-noop</artifactId>

<version>0.30.0</version>

</dependency>

## 5.2 Java Code Change:

## 5.2.1. Rest API tracing:

**Property file changes:**

jaeger.sampler.manager.host.port=10.131.126.117:5778

jaeger.agent=10.131.126.117

jaeger.agent.port=6831

**Java code changes:**

@Bean

**public** io.opentracing.Tracer jaegerTracer() {

io.opentracing.Tracer jaegerTracer **=** **new** Configuration("opentracingApp",

**new** Configuration.SamplerConfiguration(ProbabilisticSampler.***TYPE***, 1,

Integer.*parseInt*(env.getProperty("jaeger.sampler.manager.host.port"))),

**new** Configuration.ReporterConfiguration(**true**, env.getProperty("jaeger.agent.host"),

Integer.*parseInt*(env.getProperty("jaeger.agent.port")), 1000, 100)).getTracer();

GlobalTracer.register(tracer);

**return** jaegerTracer;

}



Note: you will get the Rest API tracing after adding the above bean.

## 5.2.2. HornetQ tracing:

@Bean

**public** JmsTemplate jmsTemplate(@Qualifier("jmsConnectionFactory") CachingConnectionFactory jmsConnectionFactory, @Qualifier("jaegerTracer") Tracer jaegerTracer,@Qualifier("tracingJmsMessageConverter") MessageConverter tracingJmsMessageConverter)

{

***LOGGER***.info("jmsTemplate start .....");

JmsTemplate jmsTemplate = **new** TracingJmsTemplate(jmsConnectionFactory,

jaegerTracer);

jmsTemplate.setMessageConverter(tracingJmsMessageConverter);

***LOGGER***.info("jmsTemplate end .....");

**return** jmsTemplate;

}

@Bean

**public** MessageConverter tracingJmsMessageConverter(@Qualifier("jaegerTracer") Tracer jaegerTracer,@Qualifier("jacksonJmsMessageConverter") MessageConverter jacksonJmsMessageConverter)

{

***LOGGER***.info("jacksonJmsMessageConverter start .....");

**return** **new** TracingMessageConverter(jacksonJmsMessageConverter(), jaegerTracer);

}

@Bean

**public** JmsListenerContainerFactory<?> myFactory(@Qualifier("jmsConnectionFactory") CachingConnectionFactory jmsConnectionFactory, DefaultJmsListenerContainerFactoryConfigurer configurer, MessageConverter tracingJmsMessageConverter)

{

DefaultJmsListenerContainerFactory factory = **new** DefaultJmsListenerContainerFactory();

factory.setMessageConverter(tracingJmsMessageConverter);

// This provides all boot's default to this factory, including the message

// converter

configurer.configure(factory, jmsConnectionFactory);

// You could still override some of Boot's default if necessary.

**return** factory;

}

@Bean // Serialize message content to json using TextMessage

**public** MessageConverter jacksonJmsMessageConverter()

{

MappingJackson2MessageConverter converter = **new** MappingJackson2MessageConverter();

converter.setTargetType(MessageType.***TEXT***);

converter.setTypeIdPropertyName("\_type");

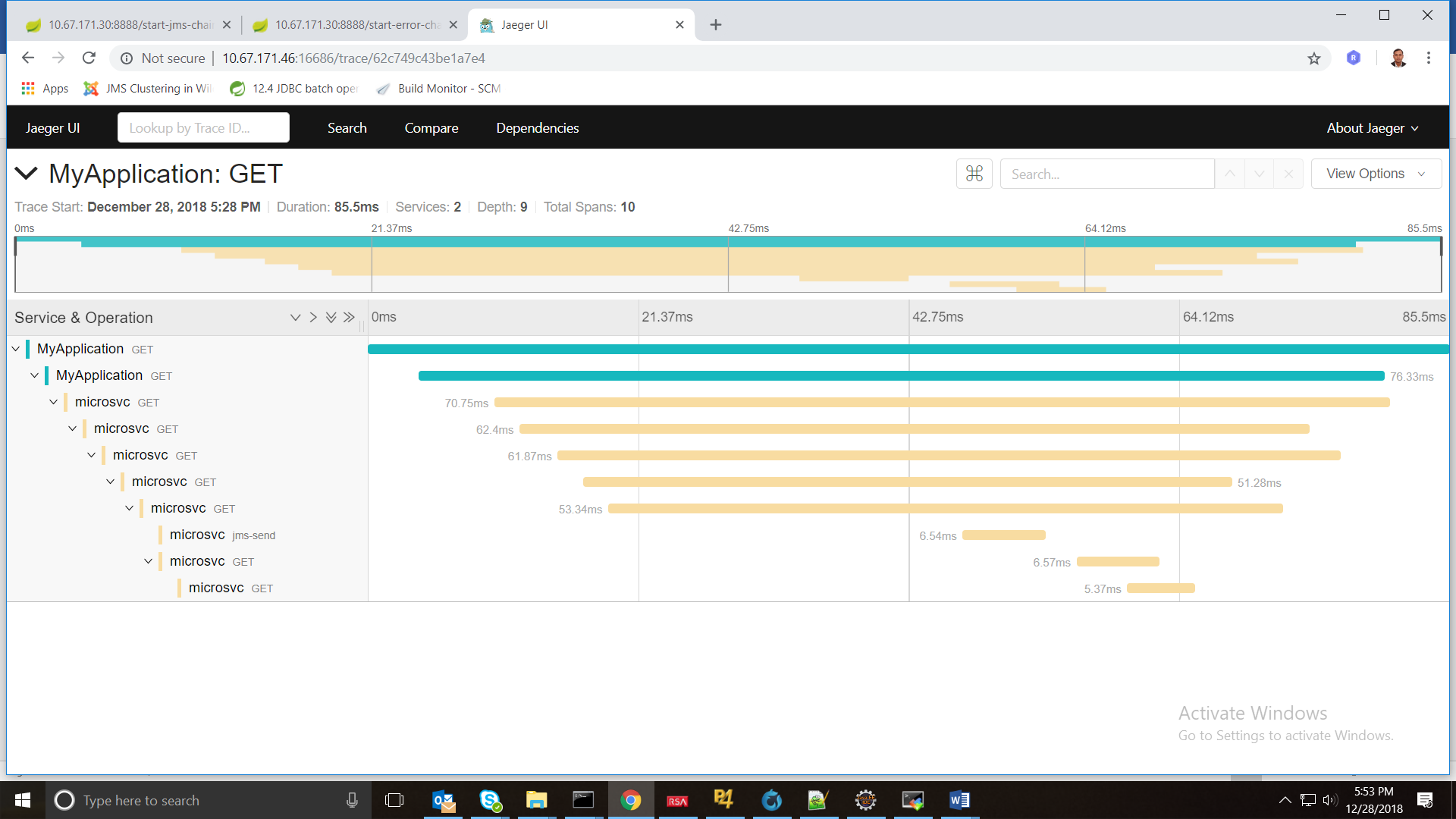
**return** converter;

}

****

**Note: Other Used Bean configuration you can see from microsvc\_framework\_template**

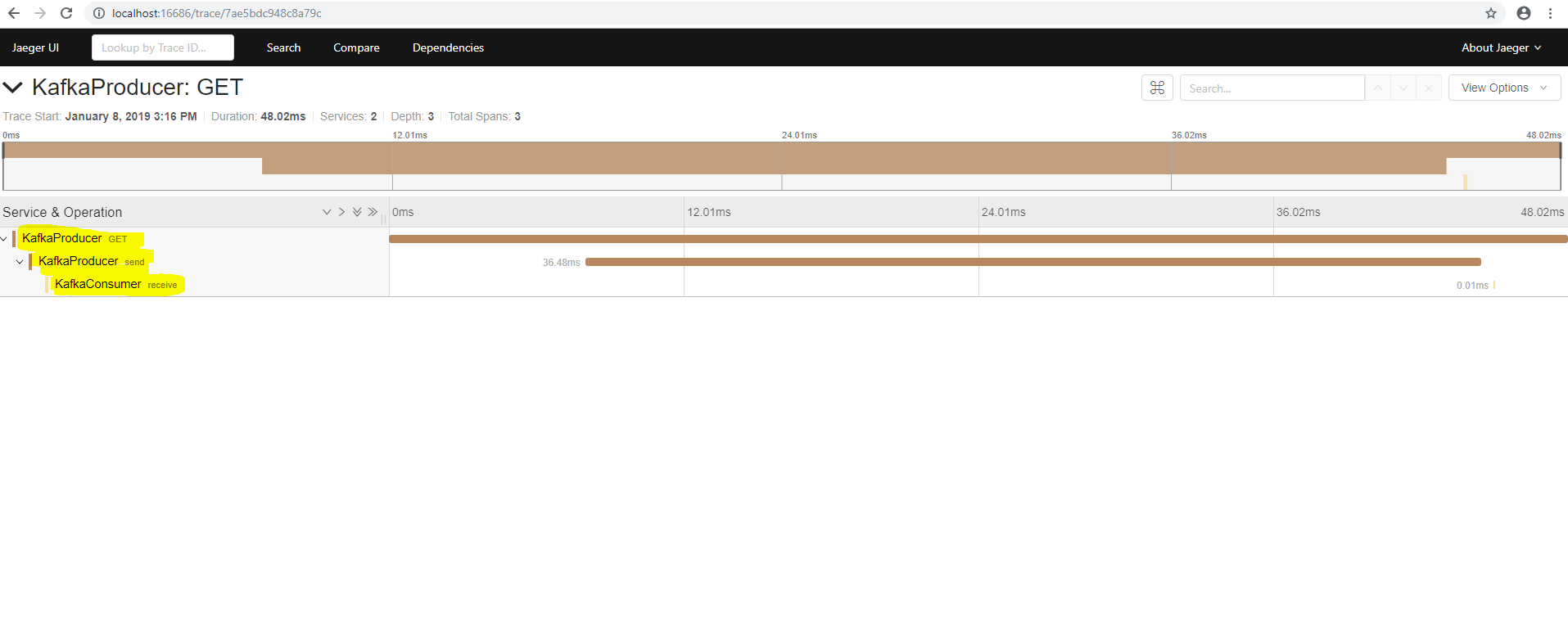
**Component**



## 5.2.3. Kafka tracing:

Configure Kadka Producer and Consumer using below URL.

[**https://github.com/opentracing-contrib/java-kafka-client**](https://github.com/opentracing-contrib/java-kafka-client)

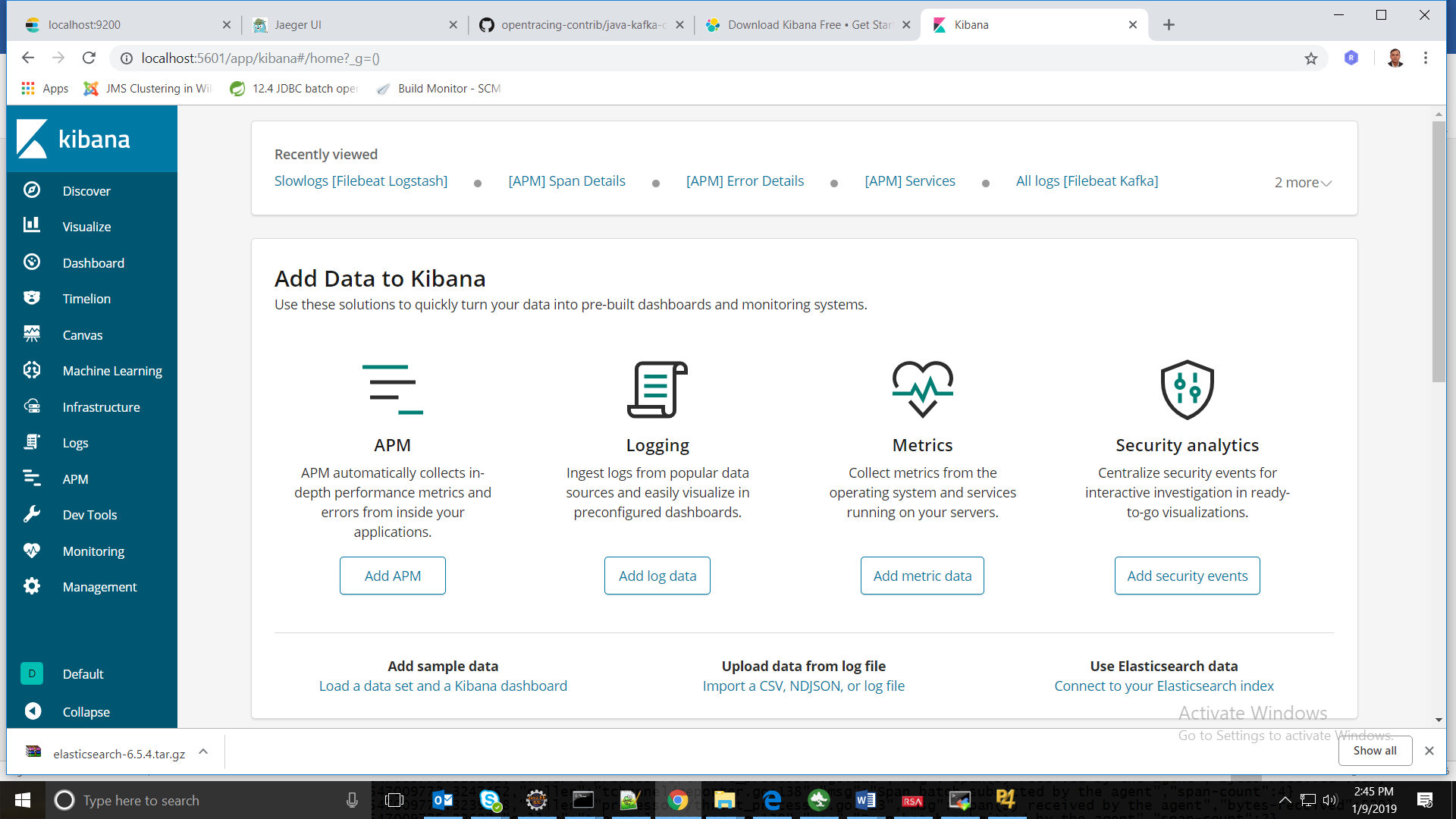


# **6. Kibana Installation:**

Download kibana from <https://www.elastic.co/downloads/kibana> url and follow the instruction to install the same.

## 6.1. Kafka tracing:

**After installation you are able to access the URL http://<Host IP>:5601**



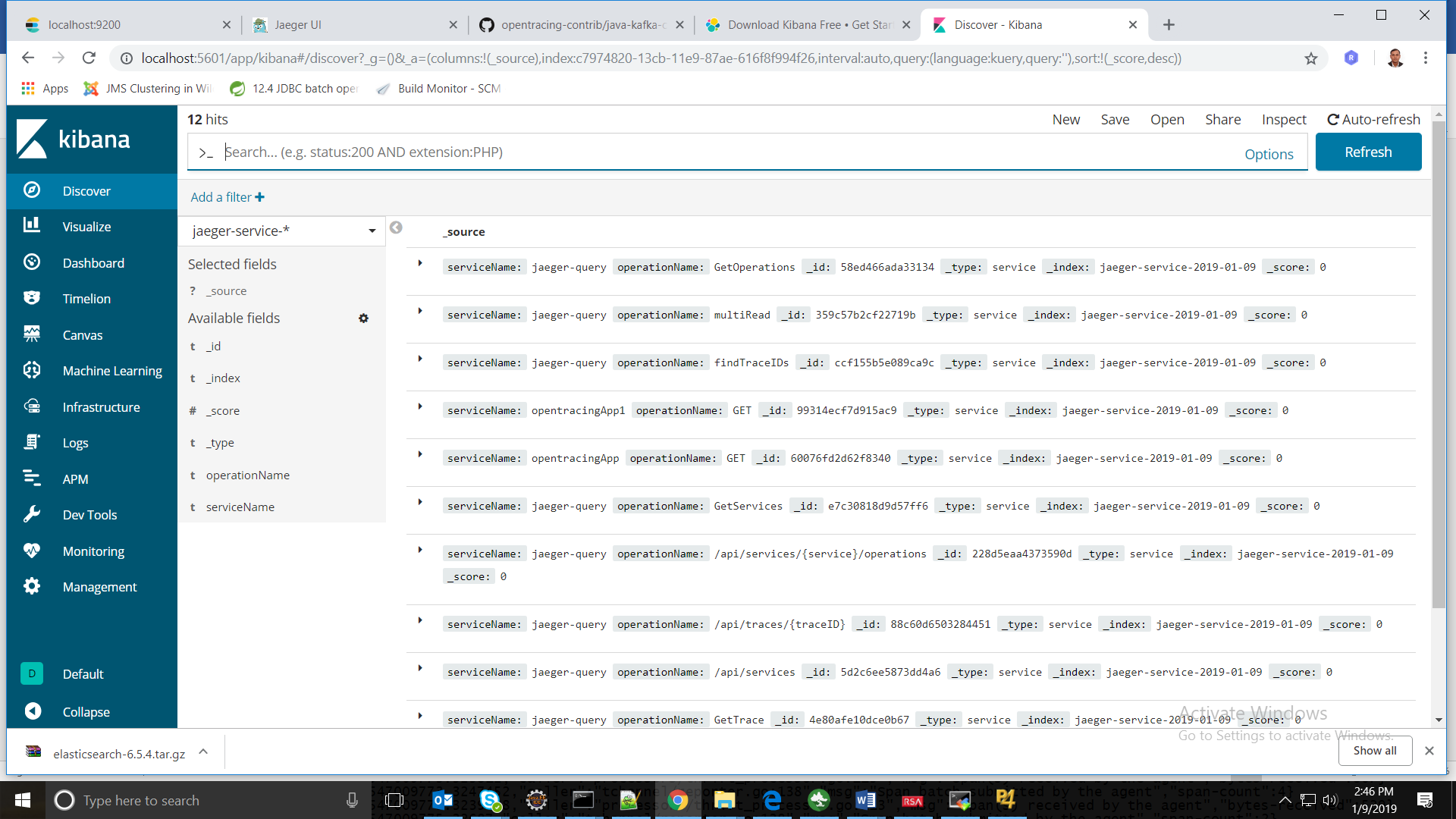
## 6.2 Jaeger service tracing log ([jaeger-service-\*](http://10.131.126.117:5601/app/kibana#/management/kibana/indices/c7974820-13cb-11e9-87ae-616f8f994f26)):

Click Infrastructure 🡪 Setup Instruction🡪All->Elasticsearch logs🡪 Select OS and follow the given instruction.

Now Go To Management 🡪 Kibana-> Index pattern🡪 create index pattern

**ja**eger-service-\* next and finish.

Now go Logs or Discover option you will be able to see the log and perform different filters here.



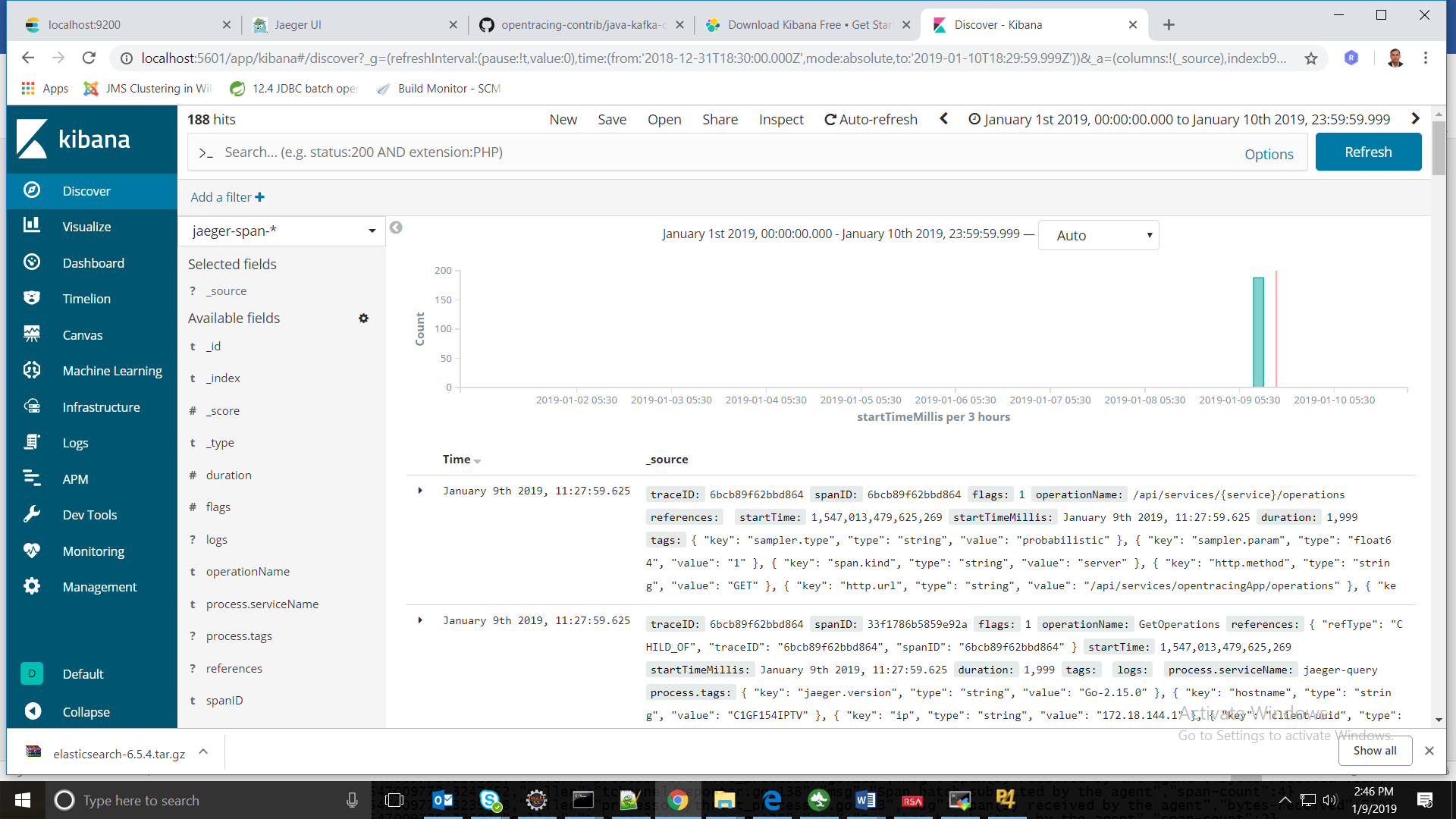
## 6.3 Jaeger service tracing log ([jaeger-span-\*](http://10.131.126.117:5601/app/kibana#/management/kibana/indices/c7974820-13cb-11e9-87ae-616f8f994f26)):

Click Infrastructure 🡪 Setup Instruction🡪All->Elasticsearch logs🡪 Select OS and follow the given instruction.

Now Go To Management 🡪 Kibana-> Index pattern🡪 create index pattern

**ja**eger-span-\* next and finish.

Now go Logs or Discover option you will be able to see the log and perform different filters here.

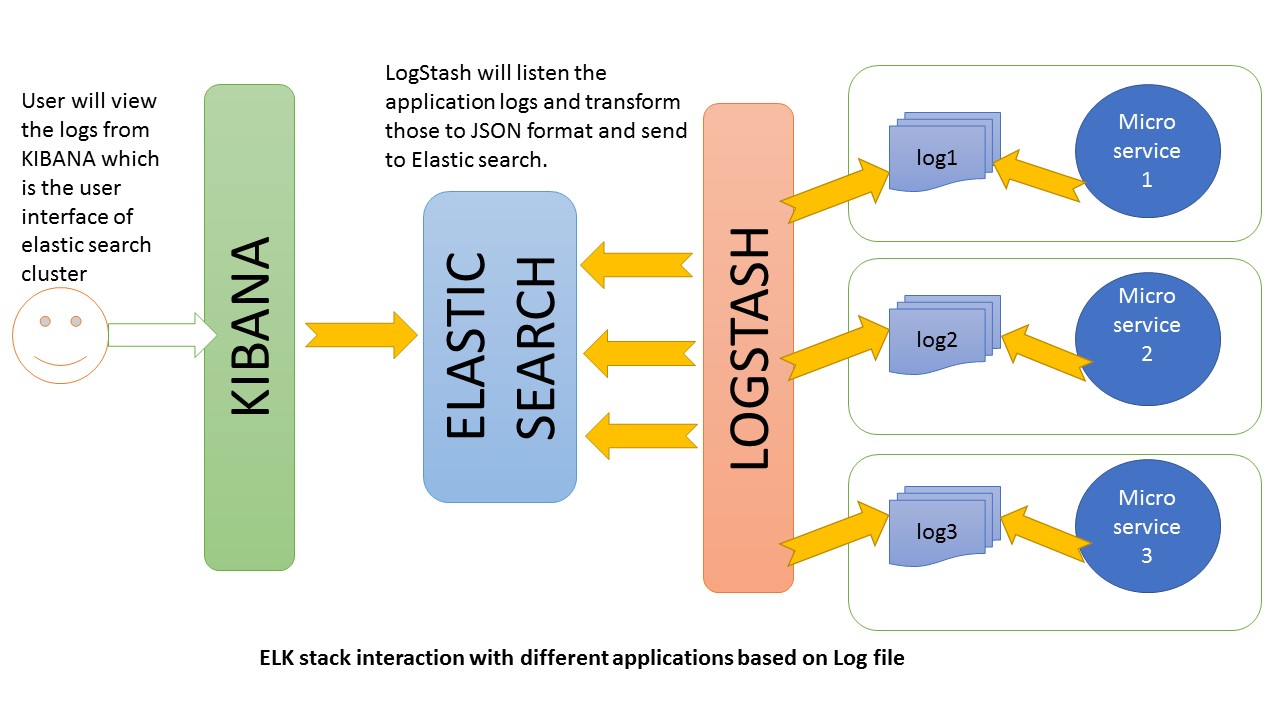


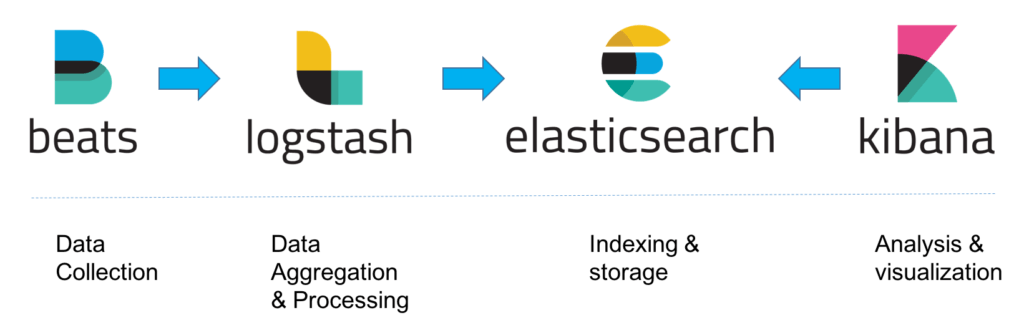
# **7. Centralized logging using ELK (Elasticsearch, Logstash and Kibana):**

ELK (Elasticsearch, Logstash and Kibana) provides very efficient centralized logging.

Elasticsearch and Kibana we have already installed in above sections.

The Elastic Stack (ELK) has four main components:





**Elasticsearch**: Elasticsearch is a distributed RESTful search engine which stores all of the collected data.

**Logstash**: Logstash is a data processing component of the Elastic Stack which sends incoming data to Elasticsearch.

**Kibana**: Kibana is a web interface for searching and visualizing logs.

**Beats**: Beats are the lightweight, single-purpose data shippers that can send data from hundreds or thousands of machines to either Logstash or Elasticsearch.

**Below are the Beats:**

**Filebeat:** Filebeat collects and ships log files to Logstash or Elasticsearch.

**Metricbeat:** Metricbeat collects metrics from your systems and services.

**Packetbeat**: Packetbeat collects and analyzes network data.

**Winlogbeat**: Winlogbeat collects Windows event logs.

**Auditbeat**: Auditbeat collects Linux audit framework data and monitors file integrity.

**Heartbeat**: monitors services for their availability with active probing.

# **8. Logstash Installation:**

Using logstash you can access the external logs and perform the analysis.

Download URL: <https://www.elastic.co/downloads>



Step1. Extract the downloaded file.

Step2. Create logstash.conf file and update input, filter and output tag.

Step3. Run below command

logstash.bat -f logstash.conf

Access below URL:

<http://10.131.126.117:5601>

Click Infrastructure 🡪 Setup Instruction🡪All->Elasticsearch logs🡪 Select OS and follow the given instruction.

Now Go To Management 🡪 Kibana-> Index pattern🡪 create index pattern

logstash-\* next and finish.

Now go Logs or Discover option you will be able to see the log and perform different filters here.

# **9. File Beat Installation:**

Using Filebeat you can access the external logs and perform the analysis.

Access below URL:

<http://10.131.126.117:5601>

Click Infrastructure 🡪 Setup Instruction🡪All->Elasticsearch logs🡪 Select OS and follow the given instruction.

Now Go To Management 🡪 Kibana-> Index pattern🡪 create index pattern

filebeat-\* next and finish.

Now go Logs or Discover option you will be able to see the log and perform different filters here.

# **10. References:**

<https://www.elastic.co/>

<https://www.jaegertracing.io/>