import boot-elk-zipkin project. it has 4 modules in it.

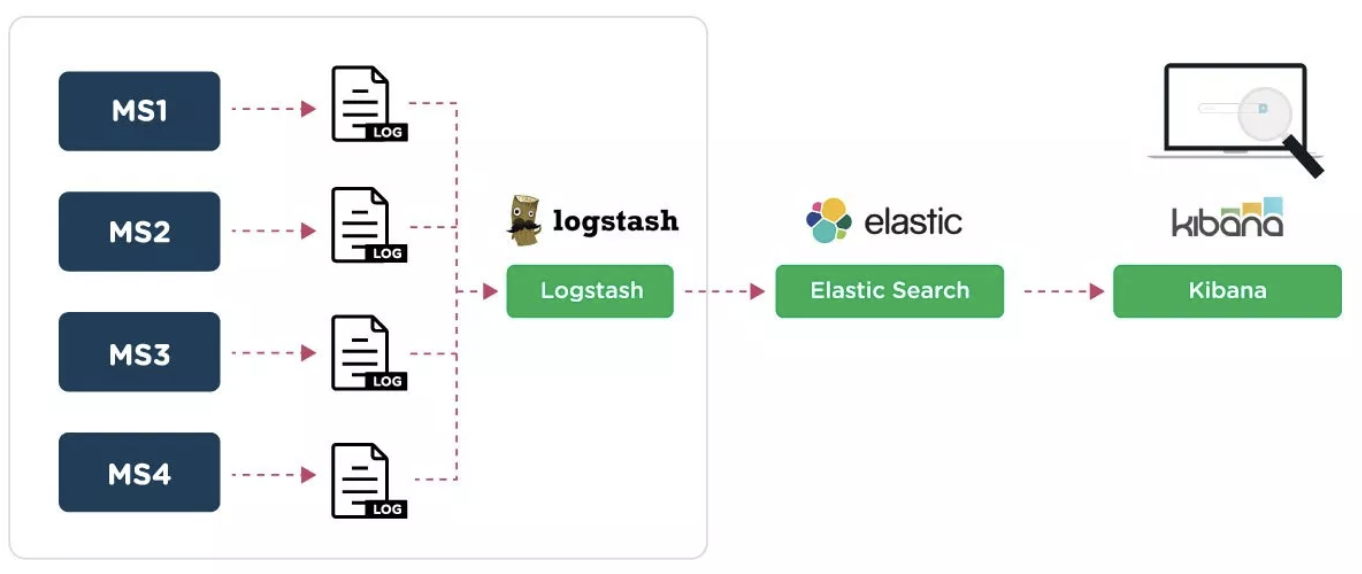
Each calling one another using rest API.

Repo location:

Build and start all 4 REST services.

Application REST call starts with: ‘http://localhost:6081/bootelkzipkin’ and then it calls intermediate REST services

First we will try to aggregate all 4 application logs and display in kibana.



Follow below steps to achieve as shown in above image:

1. Install logstash on the machine where all logs from various microservices are getting generated.
2. Put logstash.conf in logstash/bin folder. Also change the log file paths mentioned in it.

e.g. of logstash.conf:

input {

file {

type => "java"

path => "/Users/logs/app2.log"

}

file {

type => "java"

path => "/Users/lolgs/zip\*.log"

}

}

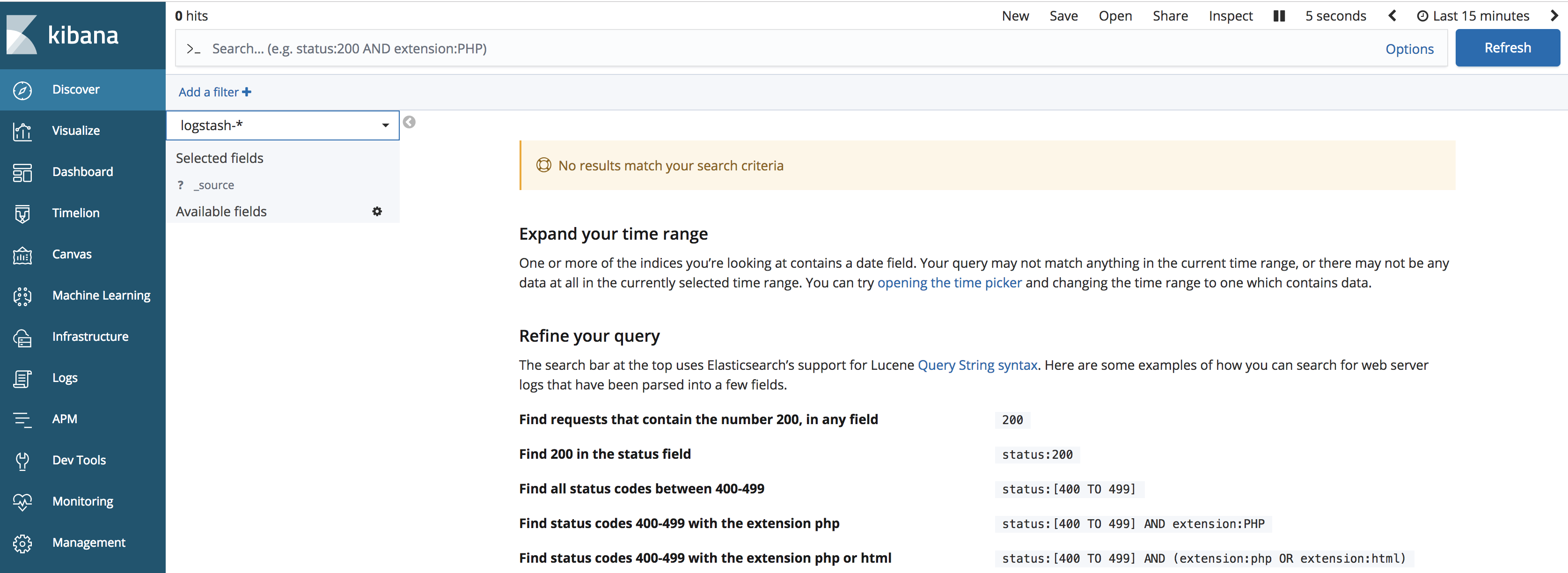
output {

elasticsearch { hosts => ["localhost:9200"] }

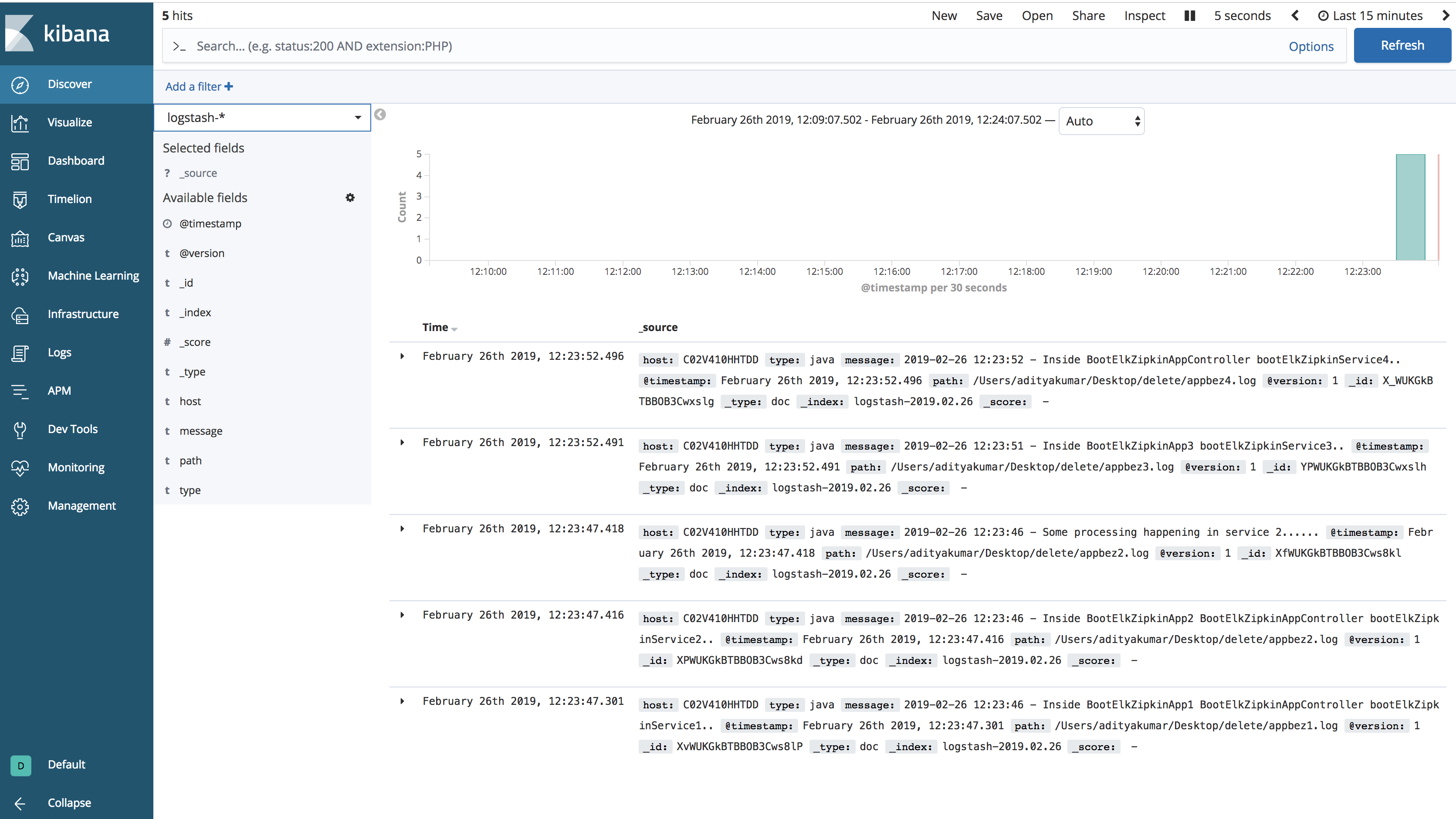
}

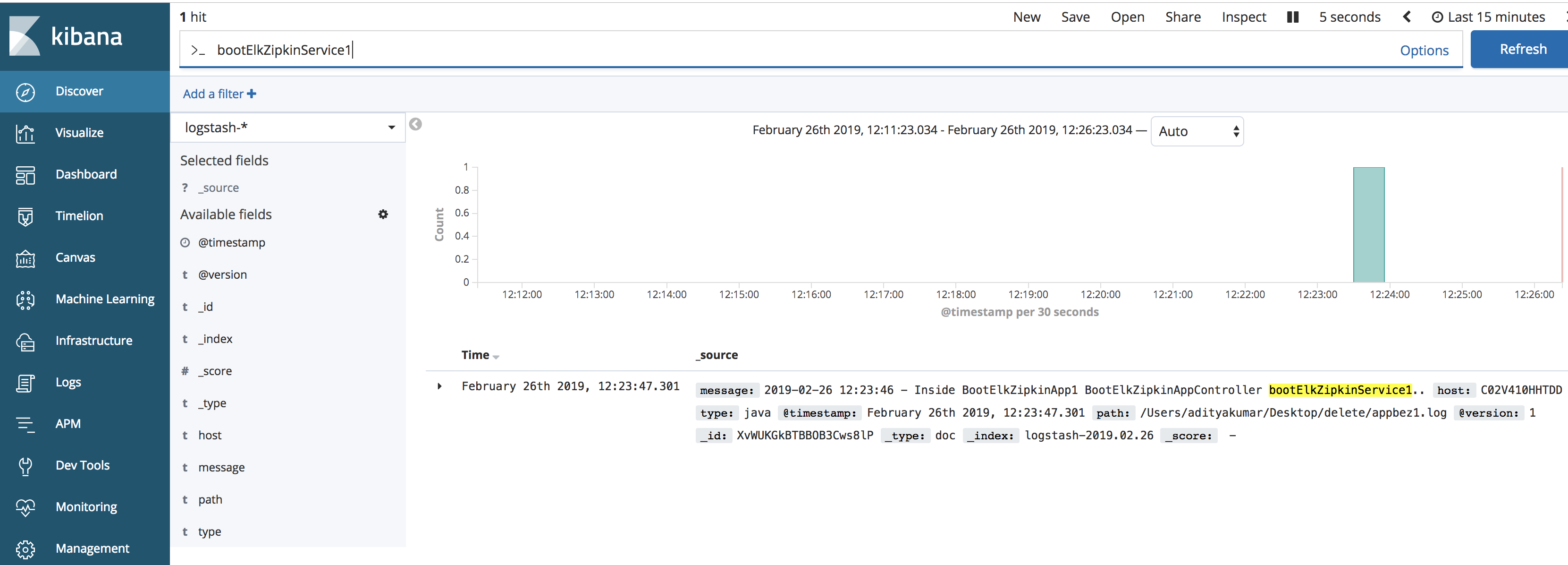
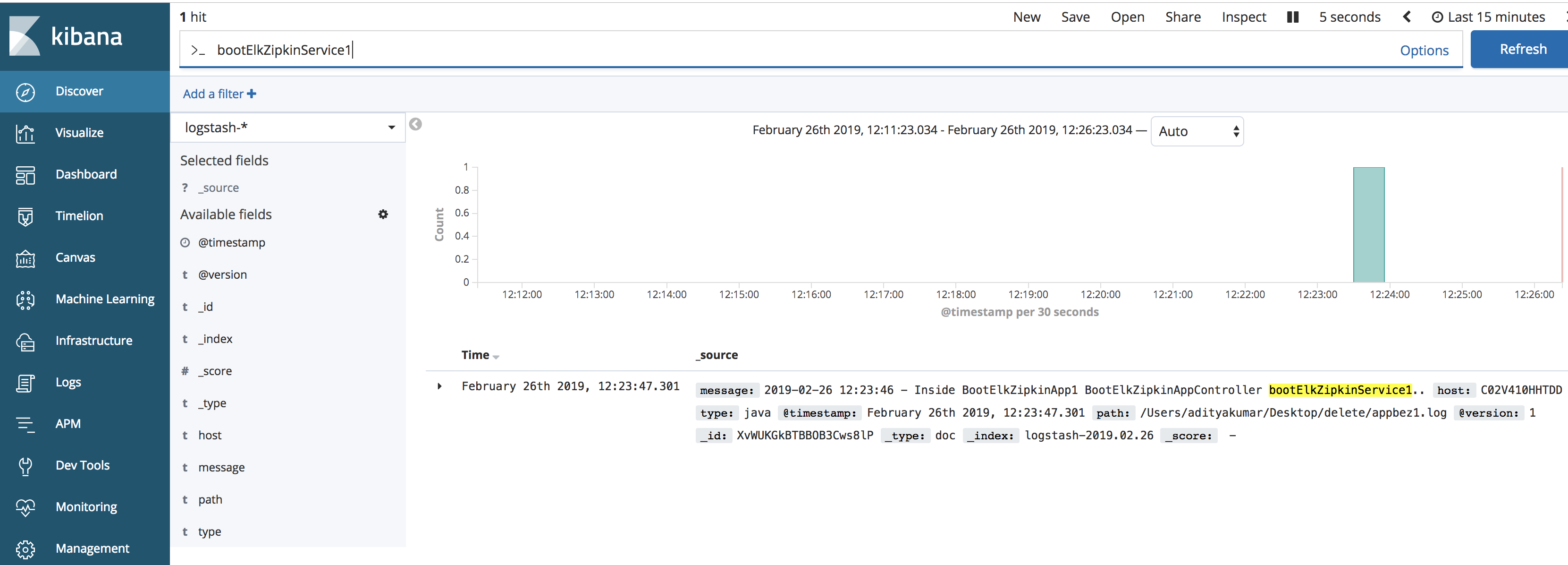
You can add filter in the above file if you want.

1. The output in logstash.conf is the location where elasticsearch is running. I am installing elasticsearch on my local system itself.
2. Install elasticsearch
3. Install kibana
4. Uncomment ‘elasticsearch.hosts: ["<http://localhost:9200>"]’ in kibana.yml in kibana/config folder
5. Then run elasticsearch and wait till it starts 🡪 elasticsearch-6.6.1/bin 🡪 ./elasticsearch  
   Verify - <http://localhost:9200/> 🡪 Response = 200
6. Now run, logstash, wait till it starts. 🡪 logstash-6.6.1/bin 🡪 ./logstash -f logstash.conf  
   verify - <http://localhost:9600/> 🡪 response = 200
7. Then run kibana 🡪 kibana-6.6.1-darwin-x86\_64/bin/ 🡪 ./kibana
8. Open kibana dashboard 🡪 http://localhost:5601/app/kibana
9. Create index with logstash-\* 🡪 Management 🡪 Index Patterns 🡪 create index pattern 🡪 enter ‘logstash-\*’ in Index pattern 🡪 click next step 🡪 select @timestamp in time filter field name and then click create index pattern
10. Go to discover tab in kibana dashboard
11. Set auto-refresh to 5 sec.



1. Make sure that logstash-\* index is selected
2. Hit the application – ‘http://localhost:6081/bootelkzipkin’
3. Check kibana dashboard



Hence we are able to aggregate all app logs and view/filter in kibana.

Note: kibana is a web interface to create indexes in elasticsearch. elasticSearch also has APIs to create indexes.

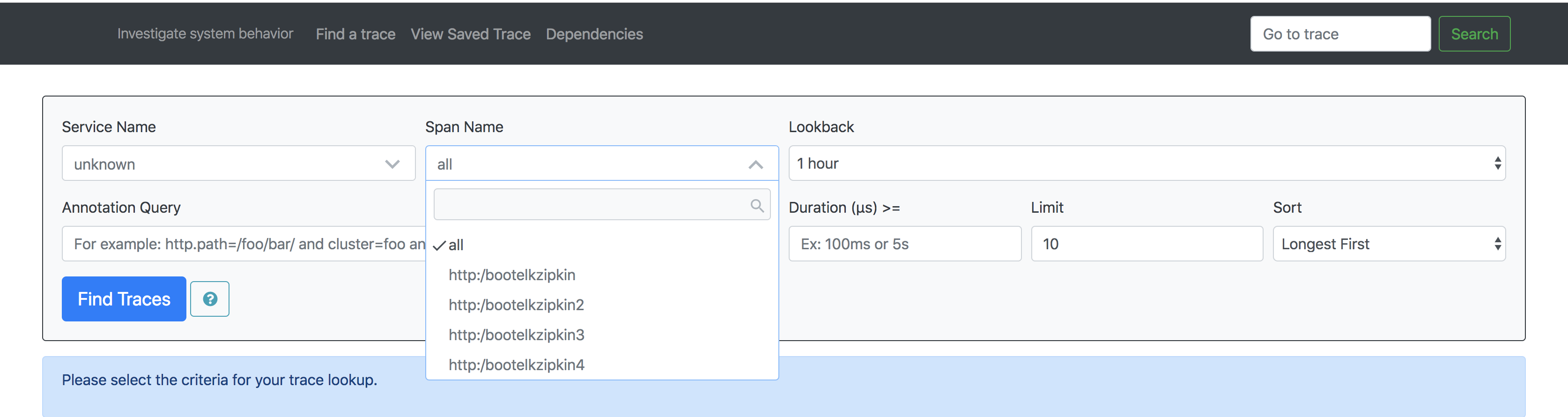
Lets integrate zipkin with ELK and view its log in kibana

1. Download zipkin jar
2. Run zipkin server 🡪 java -DSTORAGE\_TYPE=elasticsearch -DES\_HOSTS=http://<elasticsearch\_server\_ip>:://<elasticsearch\_server\_port>-jar zipkin-server-2.12.2-exec.jar

e.g. – java -DSTORAGE\_TYPE=elasticsearch -DES\_HOSTS=http://127.0.0.1:9200 -jar zipkin-server-2.12.2-exec.jar

3. wait till zipkin starts 🡪 verify - <http://localhost:9411/zipkin/>

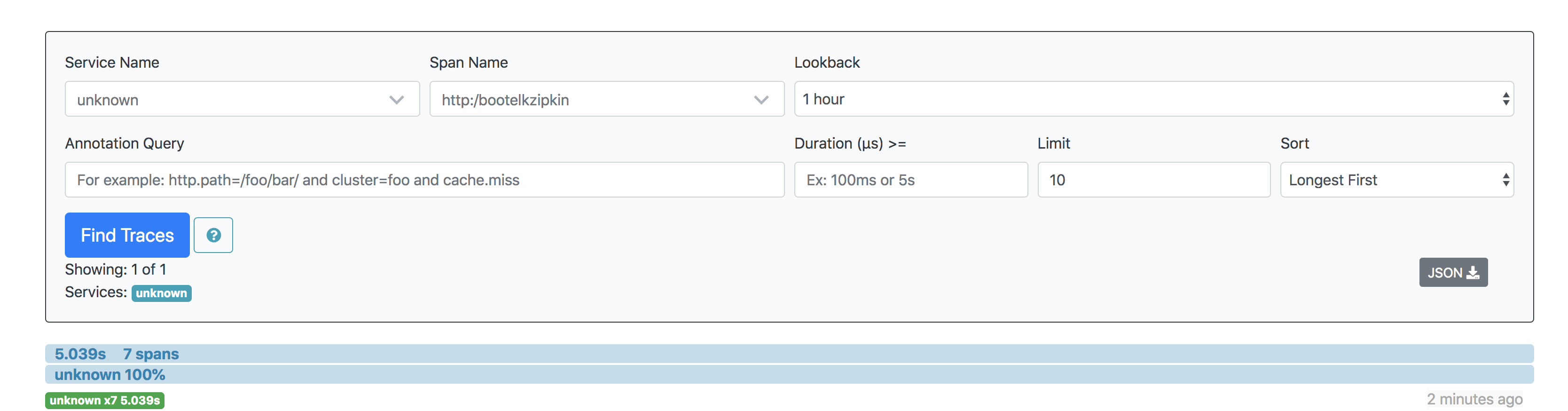
4. go to zipkin UI - <http://localhost:9411/zipkin/>

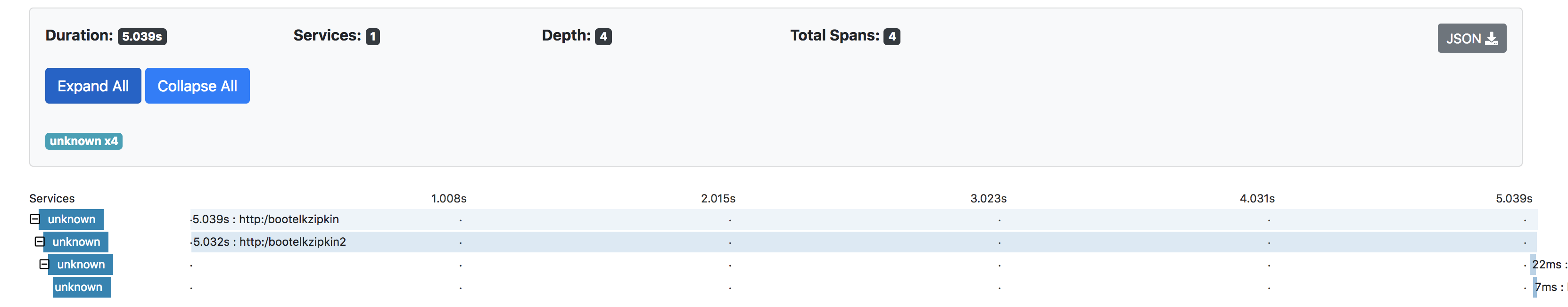


NOTE: the Service Name in above screenshot says unknown. To get the name of the parent application, its name should be given in application.properties in key: spring.application.name

e.g. - spring.application.name = server1

1. Hit application endpoint again - http://localhost:6081/bootelkzipkin
2. Select a span and click Find Traces



1. Click on the appeared rows to view the time taken by internal services.  
   
2. Create ‘zipkin\*’ index in kibana as we created ‘lolgstash-\*’ previously
3. Select ‘zipkin\*’filter in kibana dashboard.

