<https://onlyfullstack.blogspot.com/2018/07/setting-up-zipkin-server.html>

<https://cwiki.apache.org/confluence/display/ZIPKIN/Kafka>

<https://github.com/openzipkin/zipkin/tree/master/zipkin-server#kafka-collector>

Download zipkin jar

<https://zipkin.io/pages/quickstart.html>

**Starting Casandra server**

C:\project\softwares\apache-cassandra-3.11.4\bin>cassandra –f

**For downloading Cassandra gui**

<https://razorsql.com/articles/razorsql_cassandra.html>

**Start zipkin server**

C:\project\softwares>java -jar zipkin-server-2.15.0-exec.jar

**Connect Kafka to zipkin server**

C:\project\softwares>java -jar zipkin-server-2.15.0-exec.jar

**For Linux**

$ KAFKA\_BOOTSTRAP\_SERVERS=127.0.0.1:9092 java -jar zipkin.jar

**For Windows**

C:\project\softwares> set KAFKA\_BOOTSTRAP\_SERVERS=127.0.0.1:9092

C:\project\softwares> java -jar zipkin-server-2.15.0-exec.jar

**Connecting Zipkin to Cassandra**

**For Linux**

STORAGE\_TYPE=cassandra3 java -jar zipkin.jar --logging.level.zipkin=trace --logging.level.zipkin2=trace --logging.level.com.datastax.driver.core=debug

**For Windows**

C:\project\softwares> set STORAGE\_TYPE=cassandra3

C:\project\softwares> set KAFKA\_BOOTSTRAP\_SERVERS=127.0.0.1:9092

C:\project\softwares> java -jar zipkin-server-2.15.0-exec.jar --logging.level.zipkin=trace --logging.level.zipkin2=trace --logging.level.com.datastax.driver.core=debug

[Skip to end of metadata](https://cwiki.apache.org/confluence/display/ZIPKIN/Kafka#page-metadata-end)

* Created by [Adrian Cole](https://cwiki.apache.org/confluence/display/~adriancole), last modified on [Oct 29, 2018](https://cwiki.apache.org/confluence/pages/diffpagesbyversion.action?pageId=95653197&selectedPageVersions=2&selectedPageVersions=3)

[Go to start of metadata](https://cwiki.apache.org/confluence/display/ZIPKIN/Kafka#page-metadata-start)

This document was originally taken from a [pull request](https://github.com/openzipkin/sleuth-webmvc-example/pull/8) showing how to setup spring boot to send to Zipkin over Kafka. It needs help to generalize support notes that too specific to Sleuth (the library that traces spring boot applications). Until then, it still may be helpful to others as there are some troubleshooting notes here. Beyond troubleshooting, we should also answer frequently requested information about Kafka setup that are not in the README files, for example, what does a site look like? (in this case we can link to the wiki on a few well-known kafka sites).

Per sleuth docs, we add the dependency "spring-kafka" and set spring.zipkin.sender.type to kafka to enable kafka. We don't explain things already covered in zipkin, such that kafka is running and zipkin is connected to it, or how to test that anything mentioned is true. I'm going to repeat some things here in docs just to make sure we are on the same page in case someone in the future asks, perhaps we can somehow improve docs.

TL;DR;

To start, you don't need to do anything very special. You put the jar in sleuth's classpath as noted in its README (and the only change in this example PR). Add the KAFKA\_BOOTSTRAP\_SERVERS property to a stock zipkin server (as noted in its README). Assuming your versions are recent, kafka setup is ok, etc, you are good to go.

Producer-side: in this case spring-cloud sleuth

**How does someone learn how to connect sleuth to Kafka?**

<https://cloud.spring.io/spring-cloud-sleuth/single/spring-cloud-sleuth.html#_sleuth_with_zipkin_over_rabbitmq_or_kafka> says to add the spring-kafka dep and set and set spring.zipkin.sender.type to kafka.

It doesn't mention that sleuth needs to be at least 1.3, or that the spring-kafka dep has to transitively include at least a 0.10 client. However, if you use the sleuth 1.3+ or 2.0 it certainly connects to kafka and sends messages.

The most important setup (which is noted in sleuth docs!) are this in the pom file:

|  |
| --- |
| <dependency>    <groupId>org.springframework.kafka</groupId>    <artifactId>spring-kafka</artifactId>  </dependency> |

and this in the properties

|  |
| --- |
| spring.zipkin.sender.type=kafka  # If using the test image (openzipkin/zipkin-kafka) in normal docker (not docker machine)  spring.kafka.bootstrap-servers=localhost:19092 |

This is tested working with  [sleuth-webmvc-example](https://github.com/openzipkin/sleuth-webmvc-example/pull/8/files) against Zipkin's [test docker setup](https://github.com/openzipkin/docker-zipkin#kafka). When people ask about this, please point them to the demo and ask them to try it before asking more questions.

How does someone learn if the connection between Sleuth and kafka are working?

**Basic health**

Basic health isn't documented in sleuth, as what choices are available are not known ahead of time (ex if health endpoints will be configured etc). One undocumented side-effect is that when the first span is supposed to go to zipkin, a log statement should emit showing lazy initialization of the kafka component.

2018-06-19 14:00:21.048 2018-06-19 14:00:21,048 [/] INFO [AsyncReporter{KafkaSender{bootstrapServers=192.168.99.100:9092, topic=zipkin}}] o.a.k.c.u.AppInfoParser$AppInfo - Kafka version : 1.0.1

42038 --- [ topic=zipkin}}] o.a.kafka.common.utils.AppInfoParser : Kafka version : 1.0.1

**Health of message production**

Sleuth includes an adapter from zipkin to its SpanMetricReporter. If there were problems sending spans, drop metrics would increase (if metrics are setup).

Consumer-side: Zipkin

**How does someone learn how to start zipkin**

The quick start says to download the jar or run docker <https://github.com/openzipkin/zipkin#quick-start> It doesn't continue on to elaborate every configuration option, rather point to the zipkin-server README which begins that discussion.

**How does someone learn how to connect zipkin to Kafka?**

The zipkin community are well trained to point anything server related to the zipkin-server README here <https://github.com/openzipkin/zipkin/tree/master/zipkin-server#kafka-collector> In this case, it says the minimum steps are to set KAFKA\_BOOTSTRAP\_SERVERS and the example command literally does this for you KAFKA\_BOOTSTRAP\_SERVERS=127.0.0.1:9092 java -jar zipkin.jar

**How does someone learn if the connection between zipkin and kafka are working?**

**Basic health**

The zipkin-server README mentions two sorts of things to know if the connection is working or not. Firstly, to hit the health endpoint ex curl -s localhost:9411/health. [Right now](https://github.com/openzipkin/zipkin/issues/2098), the health check might be misleading. Look at the log entries instead. If you see something like below, the server isn't connecting to the broker:

2018-06-19 13:49:07.749 WARN 37556 --- [pool-2-thread-1] o.a.k.c.NetworkClient : [Consumer clientId=consumer-1, groupId=zipkin] Connection to node -1 could not be established. Broker may not be available.

**Health of message consumption**

Our zipkin-server README discusses collector metrics <https://github.com/openzipkin/zipkin/tree/master/zipkin-server#collector>  
When the collector is up, and you have sleuth or something else setup properly, metrics increase per-transport, including failures. If you see no metrics at all, there could be a subtle problem. For example, if this returns nothing.. there's a problem curl -s localhost:9411/metrics|jq .|grep kafka

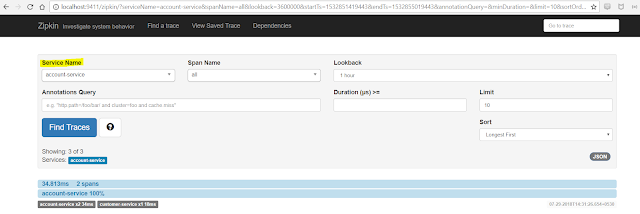
Advanced and Troubleshooting

Beyond above, there's the main versions dance. Especially as sleuth support is new, make sure you are using the very latest sleuth. Zipkin server can be older.

**Further information on the server**

<https://github.com/openzipkin/zipkin/tree/master/zipkin-autoconfigure/collector-kafka> includes more advanced notes. For example, some hints about normal kafka things you might need, such as reseting offsets.

**How to configure the Zipkin Server?**  
Zipkin is a distributed tracing system. It helps gather timing data needed to troubleshoot latency problems in microservice architectures. It manages both the collection and lookup of this data. Zipkin’s design is based on the Google Dapper.  
  
Spring has deprecated the Zipkin UI and Zipkin server from Spring Boot.  
  
Lets configure the external Zipkin server.  
  
1. **Download the zipkin** server from Java section :   
https://zipkin.io/pages/quickstart.html   
  
2. After downloading the jar file fire below command to start the zipkin server  
**java -jar zipkin.jar**  
This will start the zipkin server on http://localhost:9411

[](https://2.bp.blogspot.com/-0C4BHJJfi5w/W2BEeboJhII/AAAAAAAAACE/KSvgFKx27WU_JuJvN3pLxBcfpB3s47gpwCLcBGAs/s1600/Part%2B71%2B-%2BCapture.PNG)

**Blogs related to Zipkin Server**

**Part 7 - Distributed Tracing With Spring Cloud Sleuth**

In this tutorial we will understand below topics  
    - What is Spring Cloud Sleuth?  
    - What is distributed tracing solution?  
    - How to implement the Spring Cloud Sleuth in microservices?  
[PART 7 - MICROSERVICES WITH SPRING CLOUD : DISTRIBUTED TRACING WITH SPRING CLOUD SLEUTH](http://onlyfullstack.blogspot.com/2018/09/microservices-with-spring-sleuth.html)  
  
**Part 8 - Distributed Tracing With Spring Cloud Sleuth & Zipkin**  
    - In this tutorial we will understand below topics  
    - What is Zipkin?  
    - How to log the distributed logs at one place in microservices?  
    - [Setup Zipkin Server](http://onlyfullstack.blogspot.com/2018/07/setting-up-zipkin-server.html)  
    - [Setting up RabbitMQ](http://onlyfullstack.blogspot.com/2018/07/setting-up-rabbitmq.html)   
    - How to implement the Zipkin in microservices?  
[PART 8 - MICROSERVICES WITH SPRING CLOUD : DISTRIBUTED TRACING WITH SPRING CLOUD SLEUTH & ZIPKIN](http://onlyfullstack.blogspot.com/2018/09/microservices-with-spring-sleuth-and-zipkin.html)  
  
**Microservices Tutorial**  
[**https://onlyfullstack.blogspot.com/p/microservices.html**](https://onlyfullstack.blogspot.com/p/microservices.html)