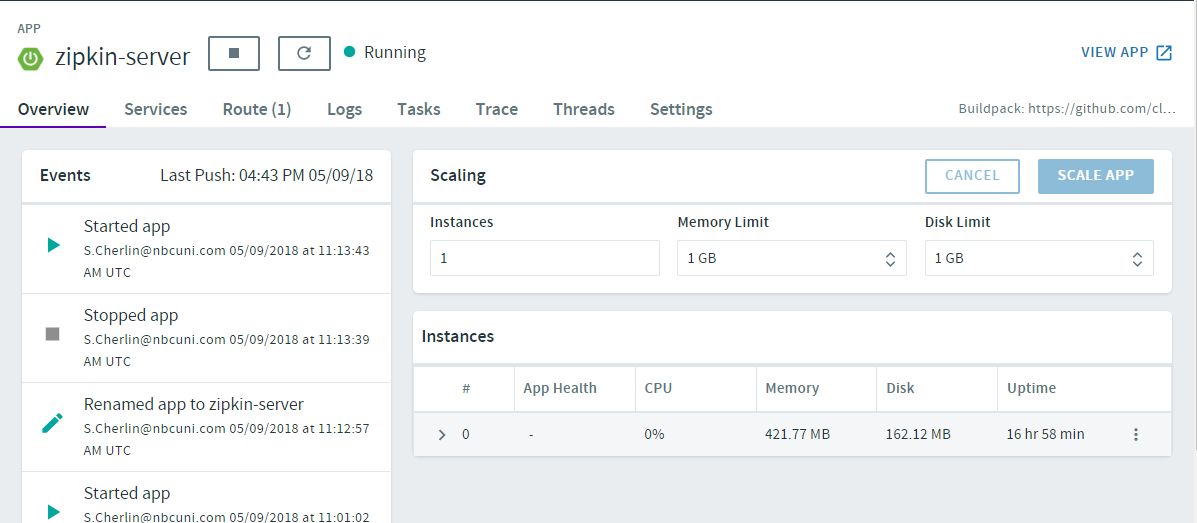
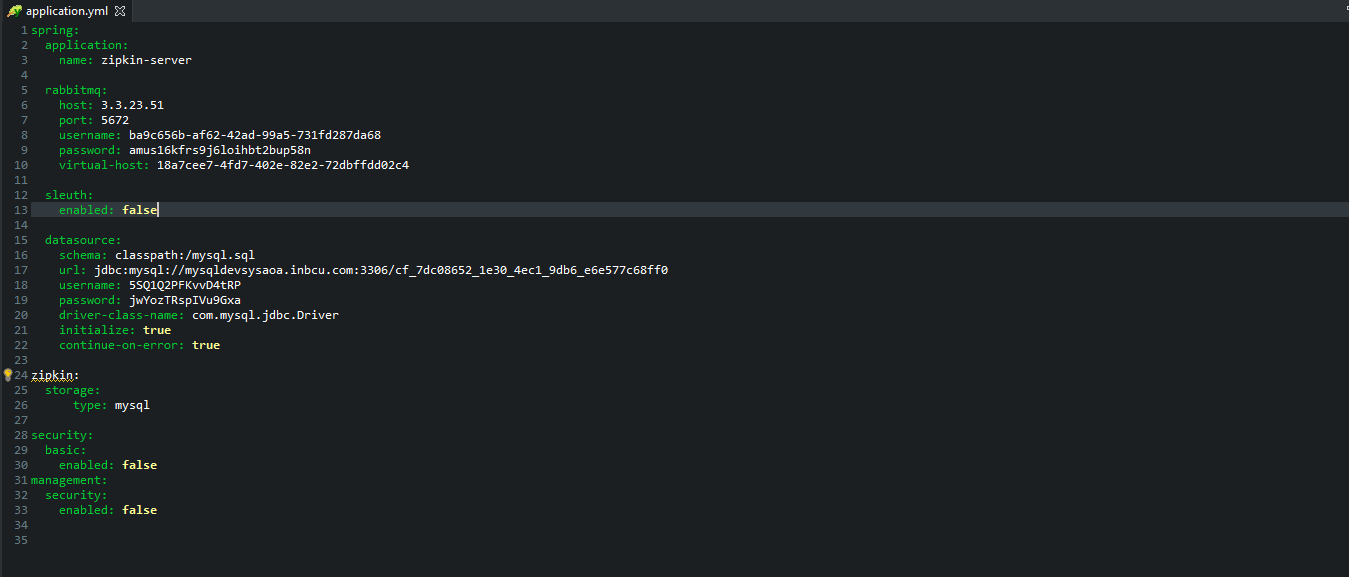
**Splunk, Zipkin with Sleuth**

This is used to track the micro service flow and it will display waterfall modal of the micro service call and its metrics details and the payload details

**Zipkin server:**

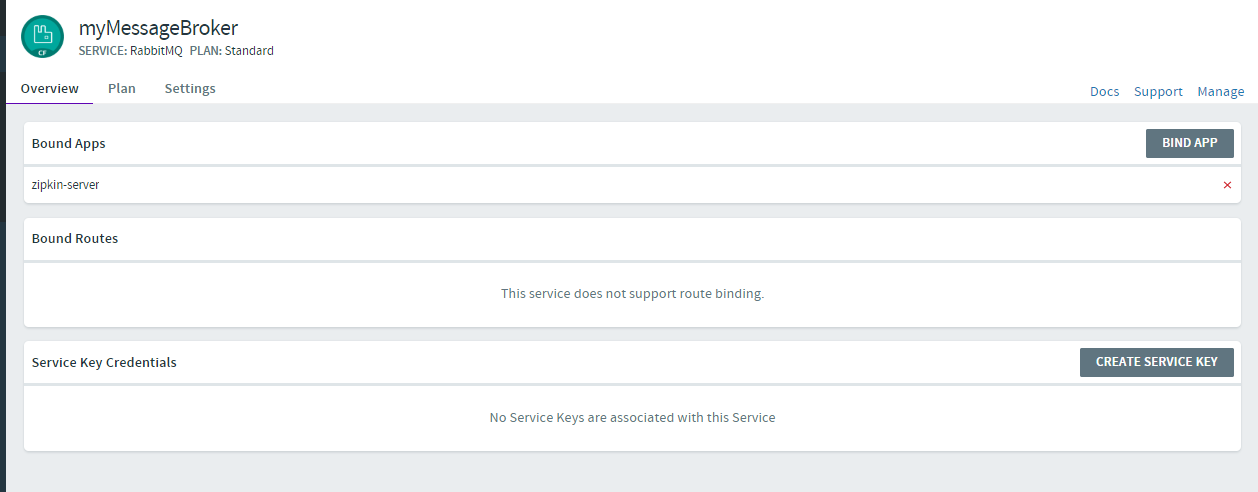


application.yml:

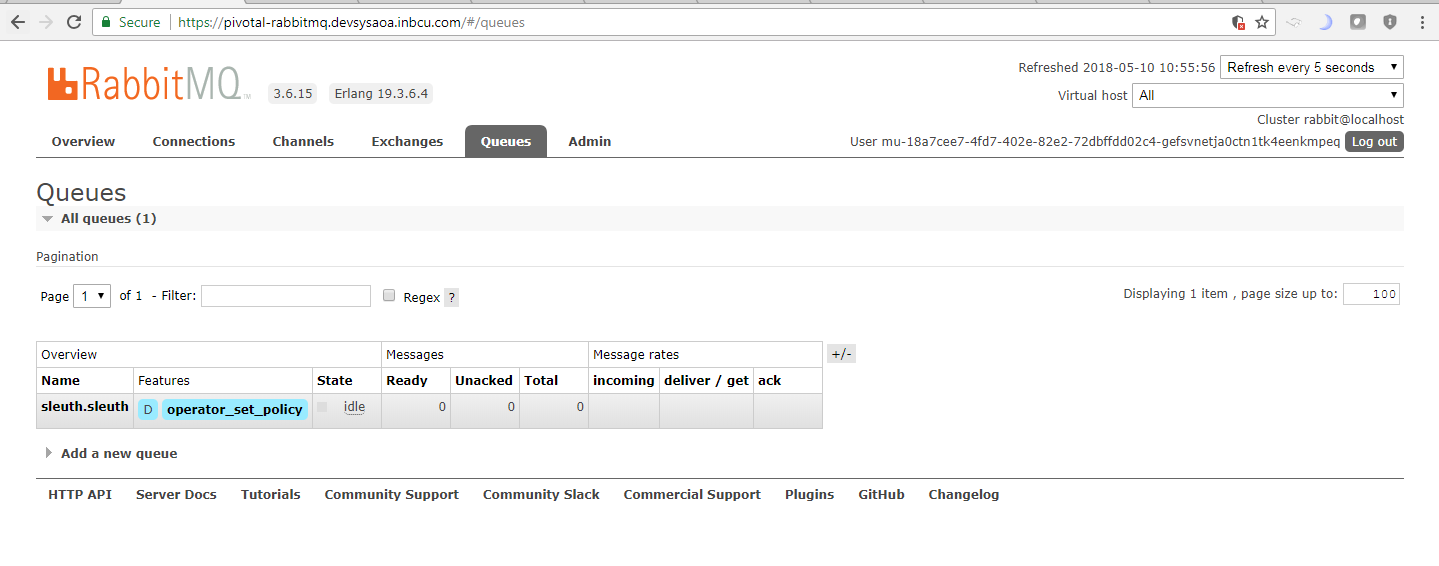


**RabbitMQ Service is created:**

And this service is binded with Zipkin server app, It act as a message broker to transfer logs to the database



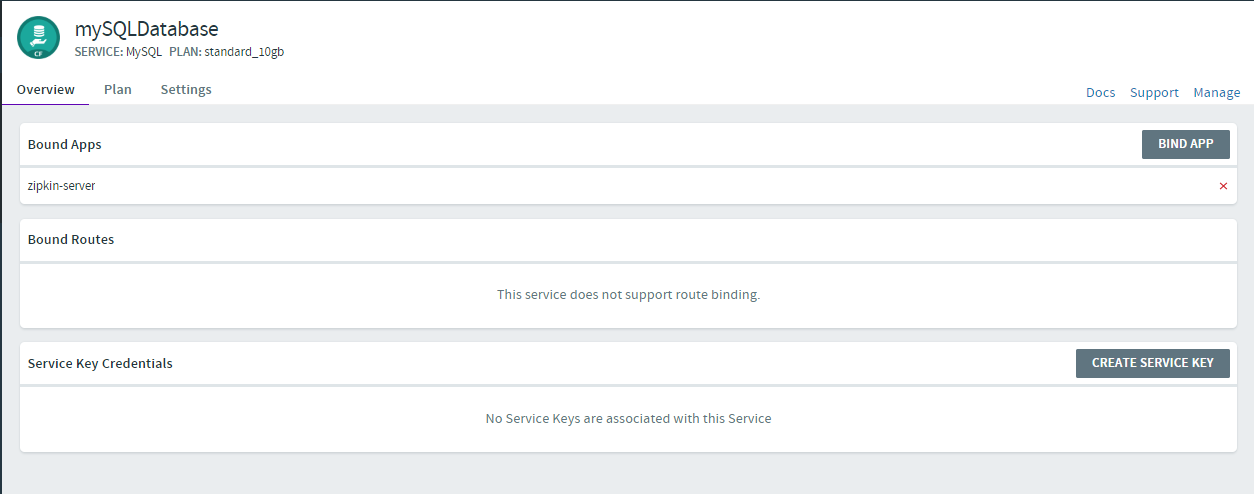
Queue will created automatically with the name **sleuth.sleuth**



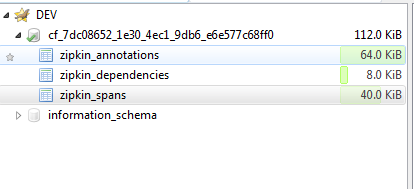
**MySQL Database:**

It is create to store the logs of the micro service. And it is binded with the Zipkin server.

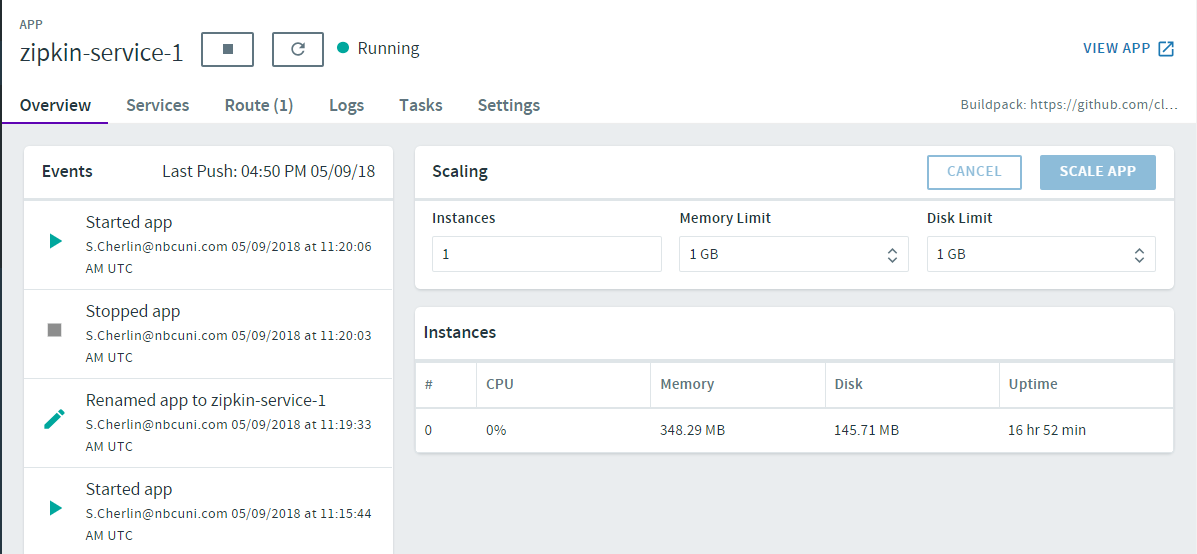
**Note**: Only **sleute** log (Contains trace id and span id of the micro service) will be stored in this database



Three tables will be automatically created with Zipkin server configuration:

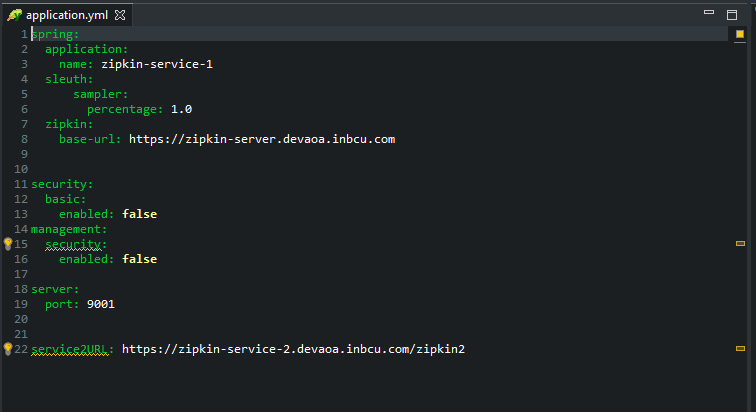


**Zipkin Client 1**: (Microservice 1)

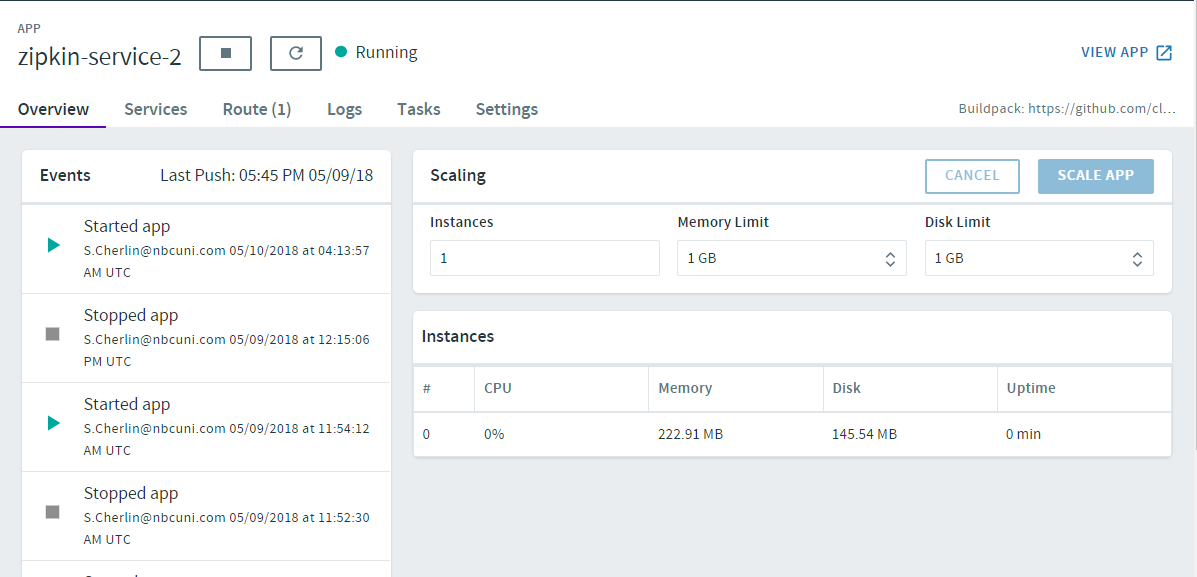


**application.yml:**

This particular Microservice logs will be streamed to the Zipkin server and it will store in the MySQL Database

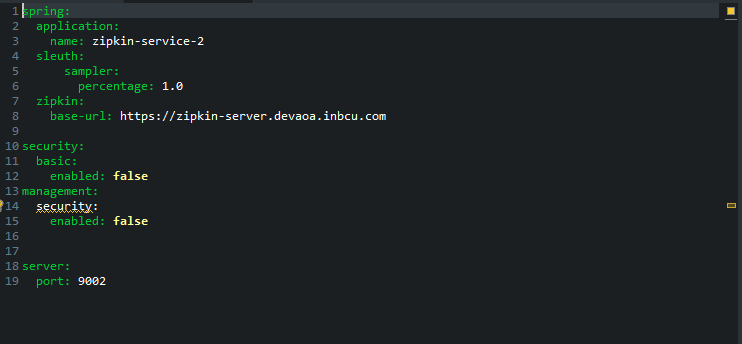


**Zipkin Client 2: (Microservice 2)**



**application.yml**

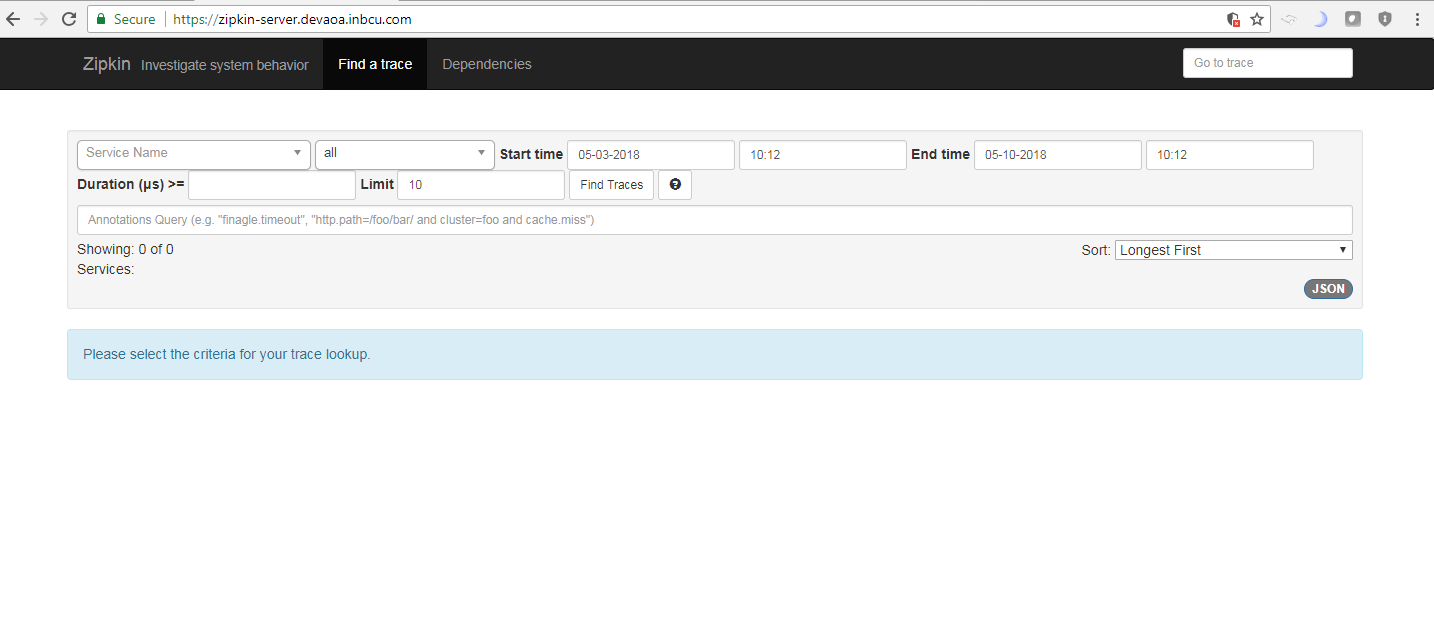
This particular Microservice logs will also be streamed to the Zipkin server and it will store in the MySQL Database



**Zipkin Server:**

**URL**: <https://zipkin-server.devaoa.inbcu.com>

**Dashboard:**



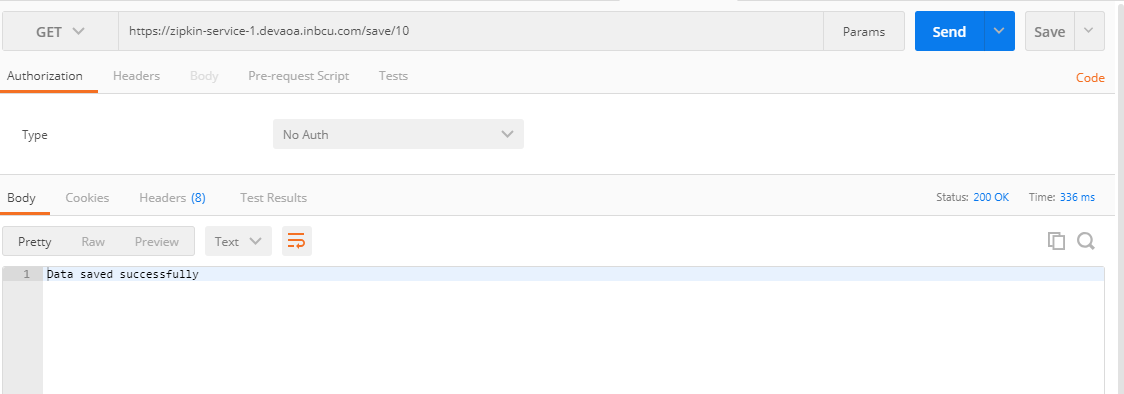
**Service Call**

**Scenario 1: All the service are up and the flow happen properly without any issue**

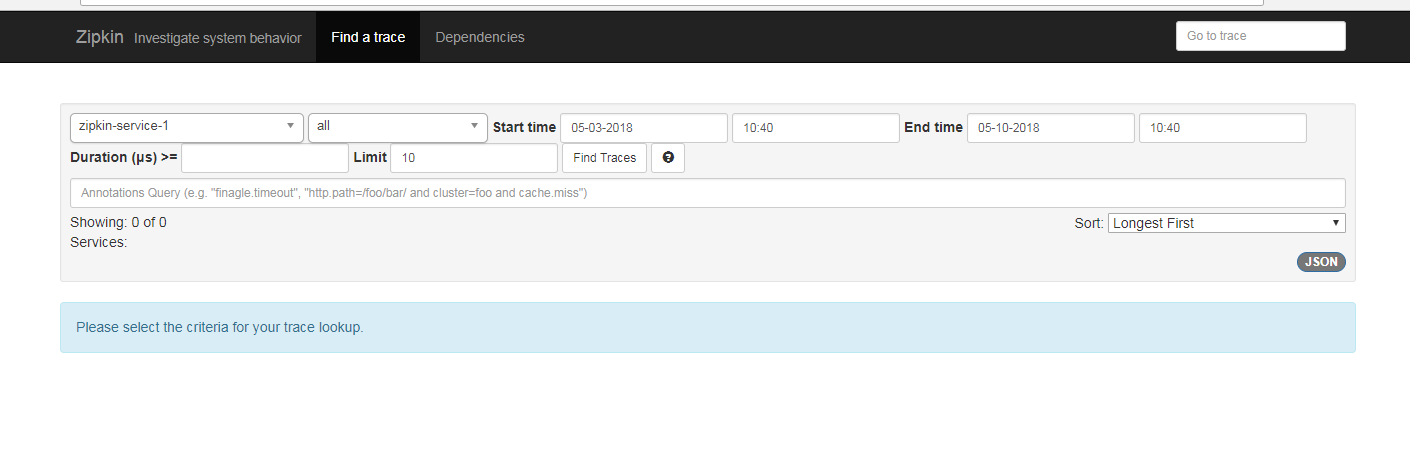
Zipkin Service 1:

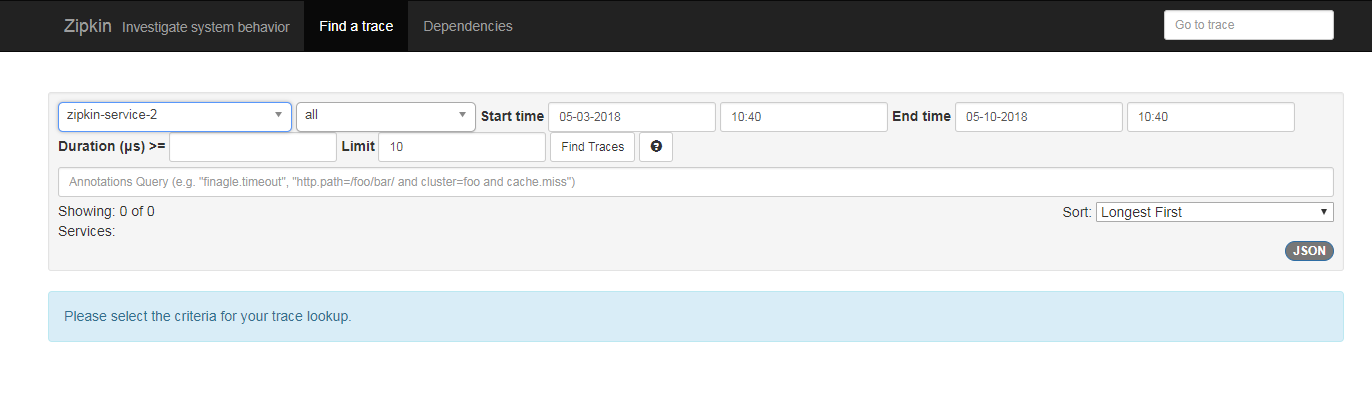
GET: <https://zipkin-service-1.devaoa.inbcu.com/save/10>

**Zipkin Service 1** will internally call the **Zipkin service 2** and post the data in json format. And the response is returned form the **Zipkin Service 2.**



After Hitting the GET method **Zipkin service 1** and **service 2** will be regiserd in the zipkin server as shown below

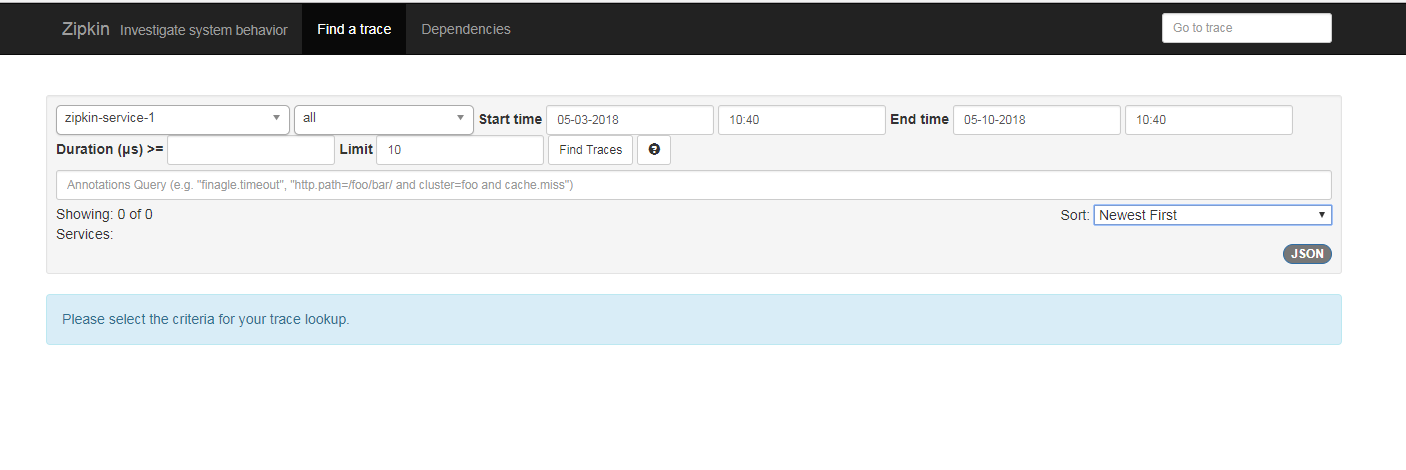




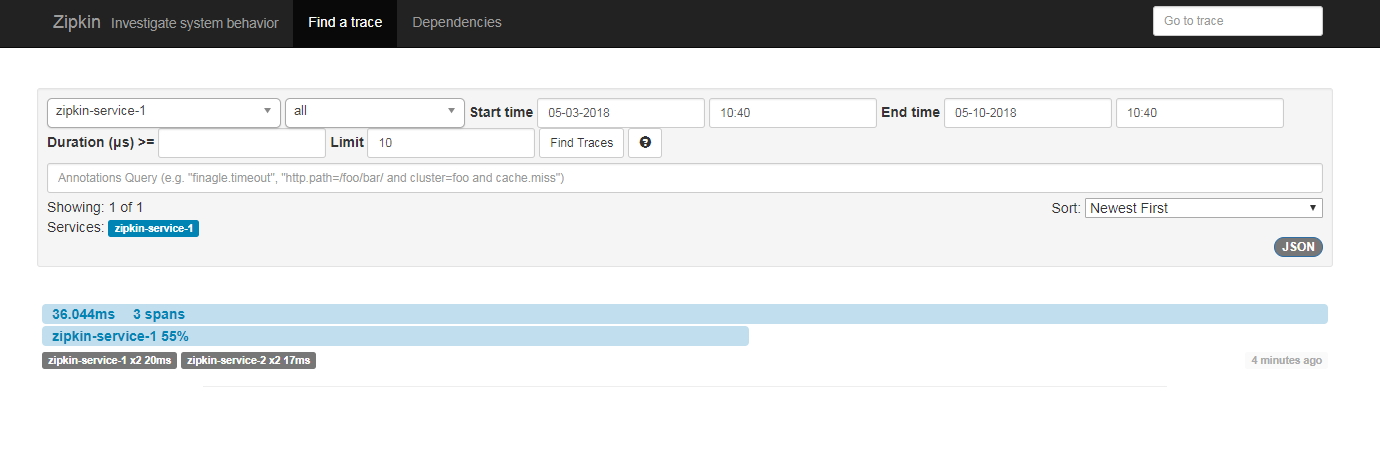
**Select the service which you want to verify:**

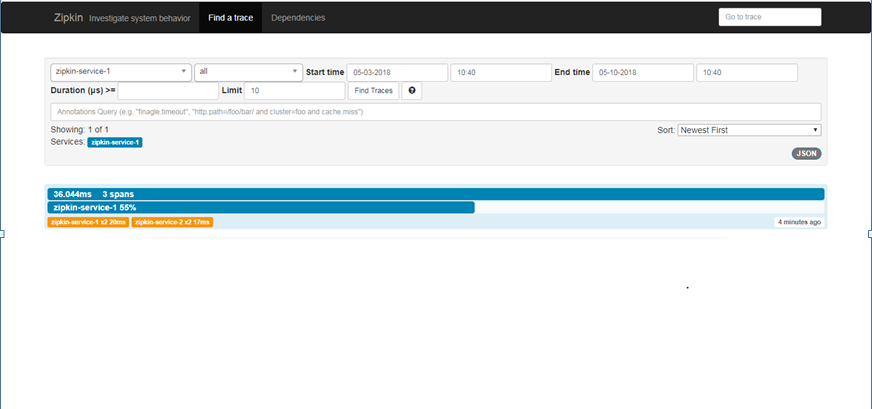
Here in this case I want to verify the **Zipkin Service 1 flow**, so go the Zipkin Server Dashboard and filter with time and service.

Click on Find trace

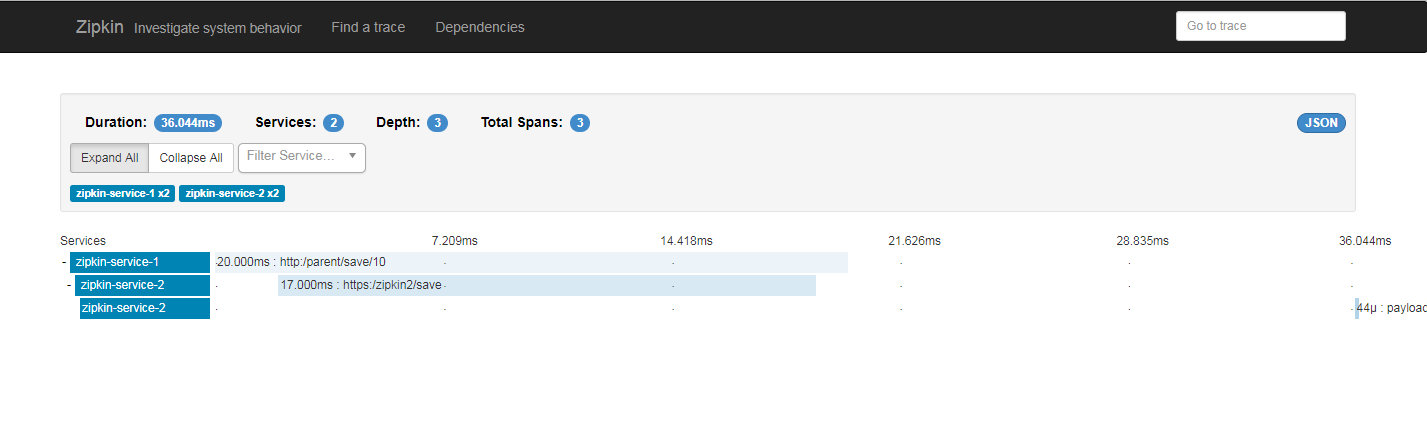


So you will get the data as shown below:

Click on the data:

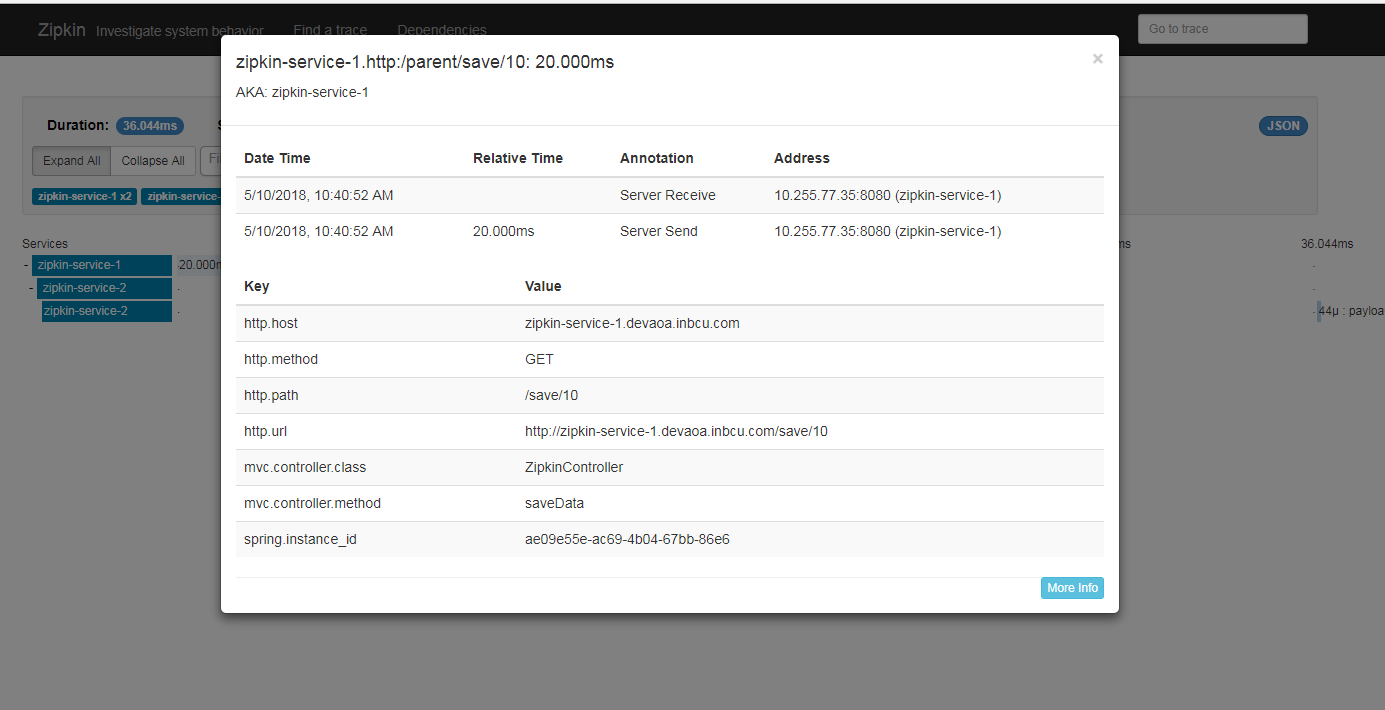


So you will able to see the micro service flow:



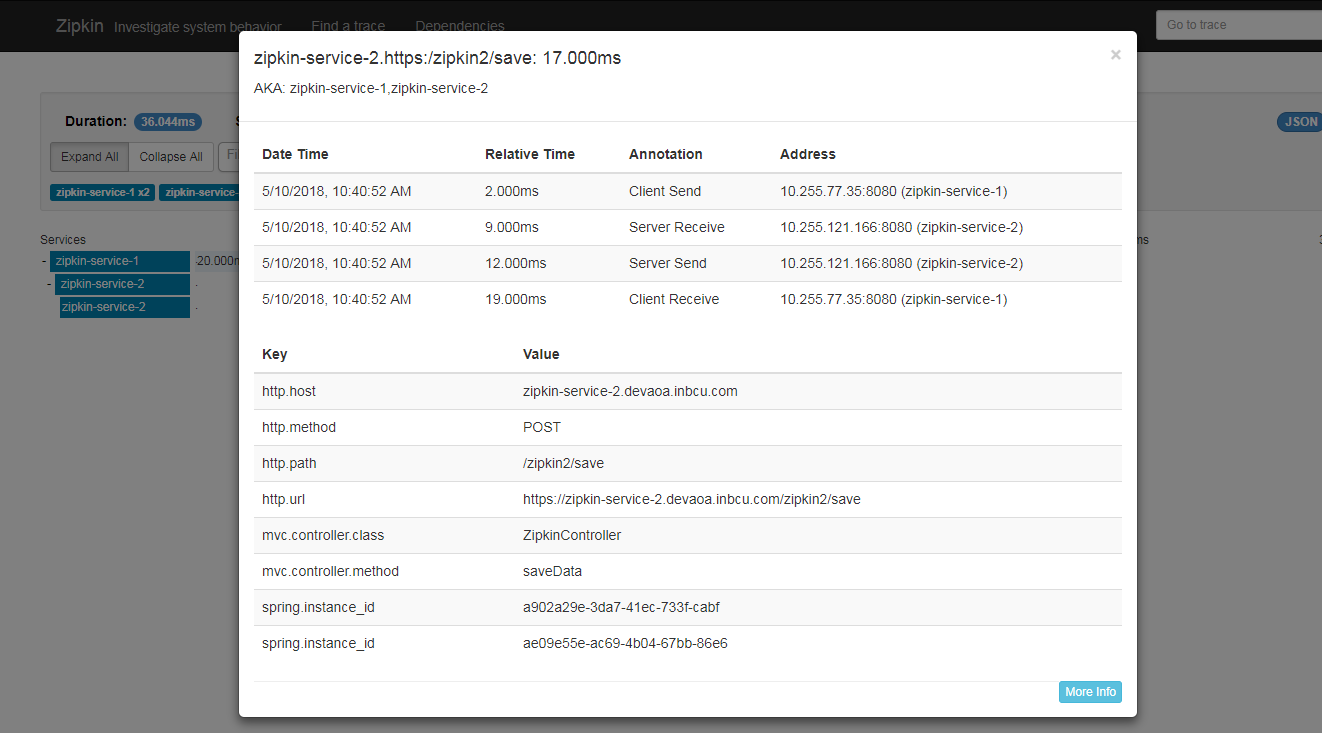
**Click on Zipkin service 1:**

You will all the details link method type, URL etc.



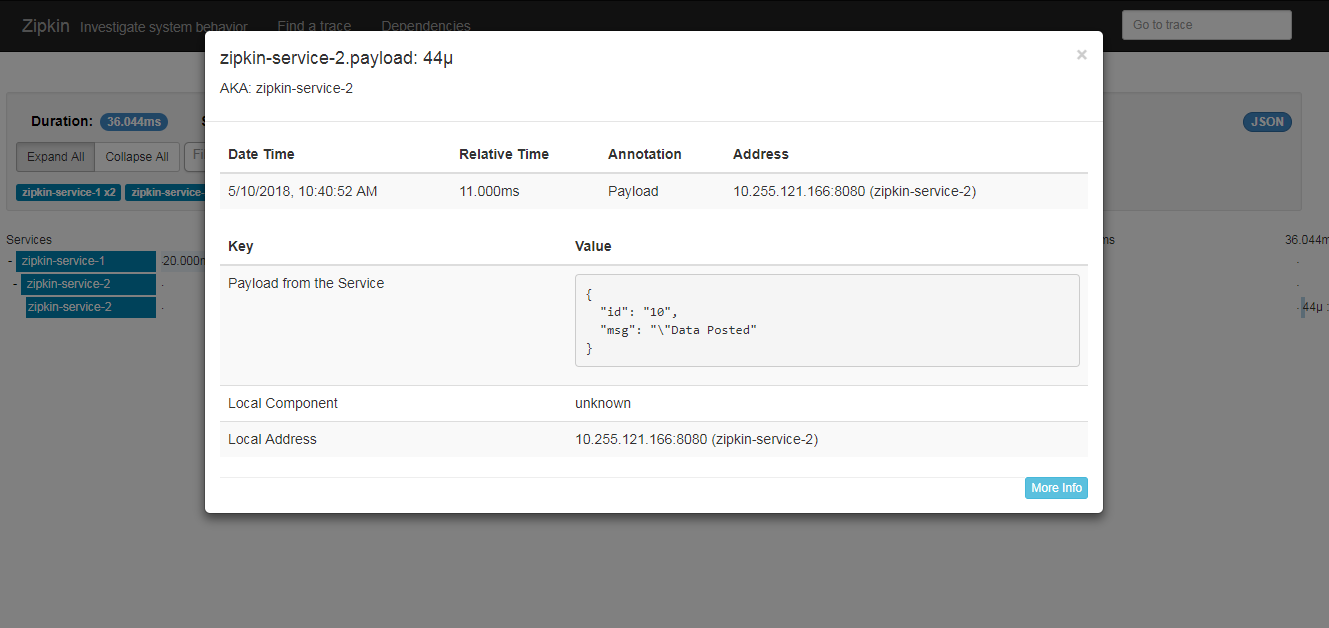
**Click on Zipkin service2:**

You can see that from Zipkin service 1, Zipkin service 2 is called with **POST** method as shown below



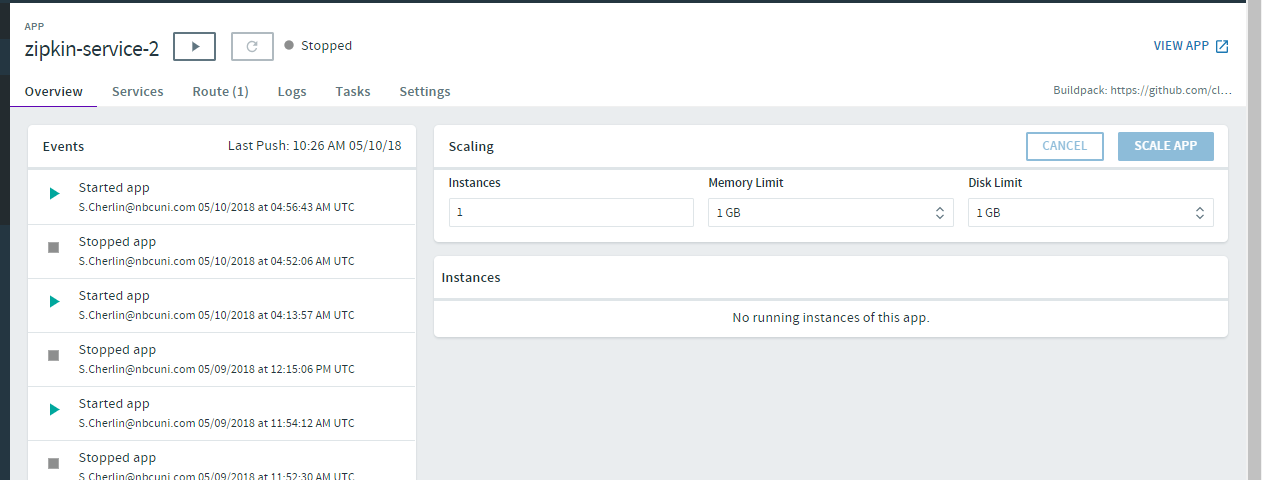
**Click on third block Zipkin service 2:**

Here you can able to see the payload received for the Zipkin service2



**Scenario 2: One of the service are down and the flow will terminate with warning**

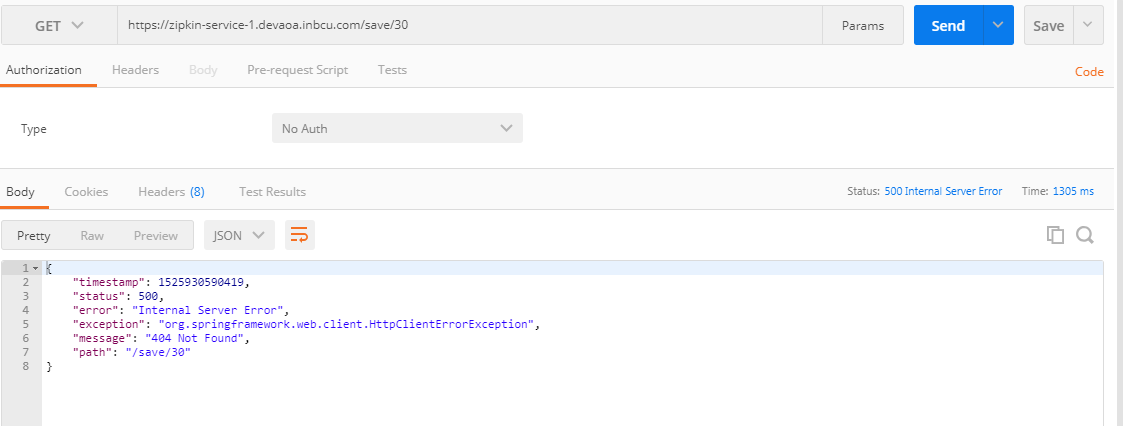
I stopped the Zipkin service 2



Zipkin Service 1:

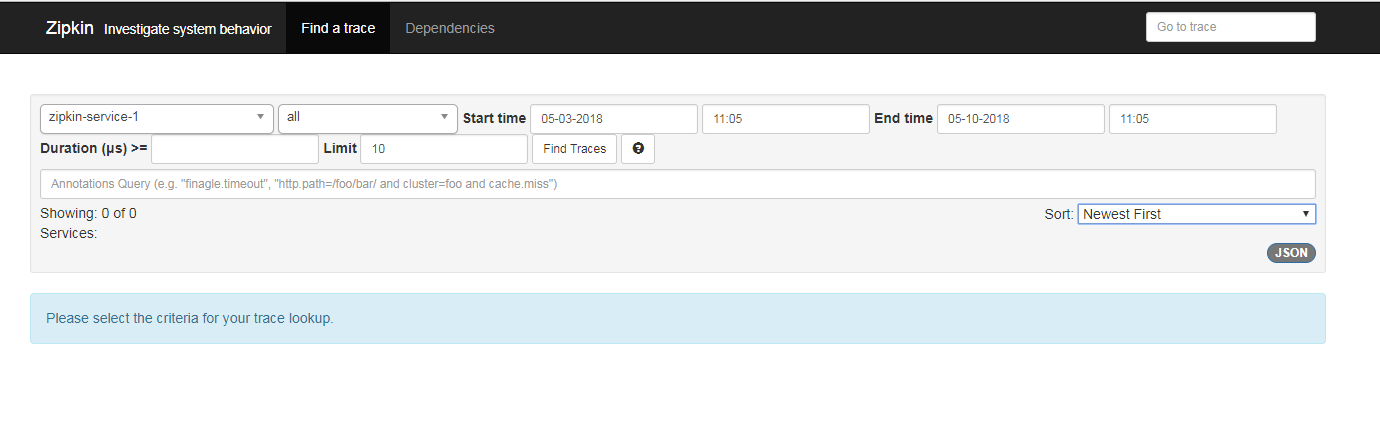
GET: <https://zipkin-service-1.devaoa.inbcu.com/save/10>

**Zipkin Service 1** will internally call the **Zipkin service 2** and post the data in json format. Here since the service 2 is down And the response code here is **500**

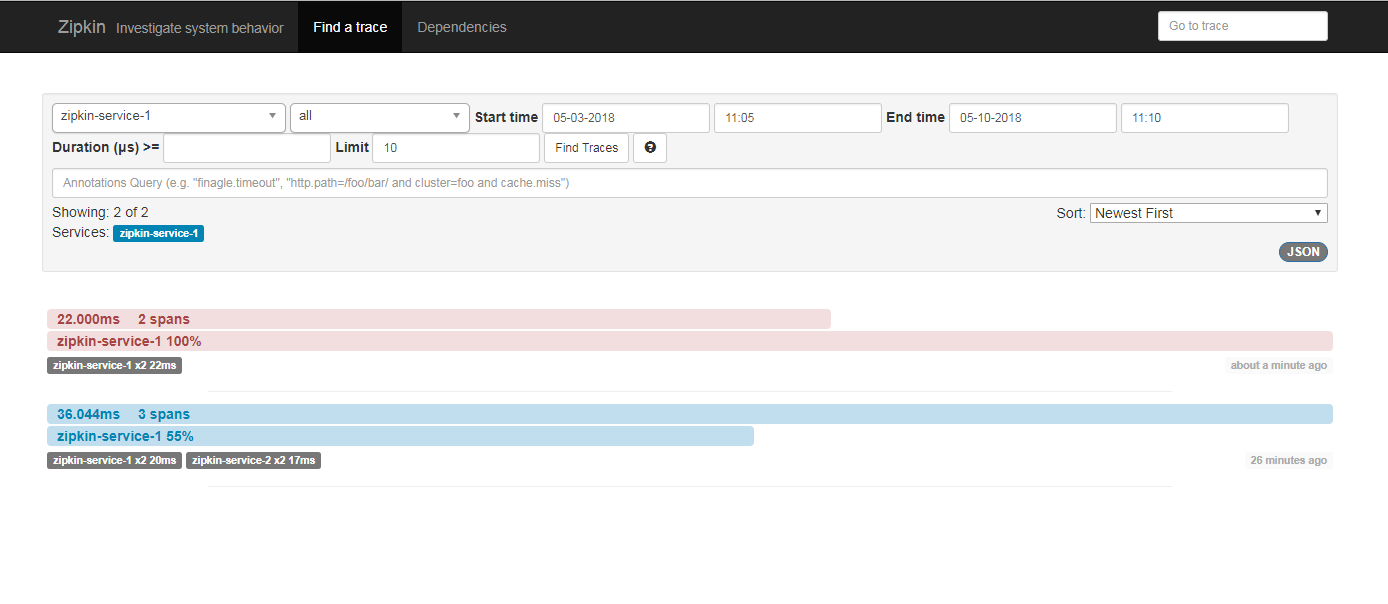


Now go to Zipkin Dashboard to analyze the issue

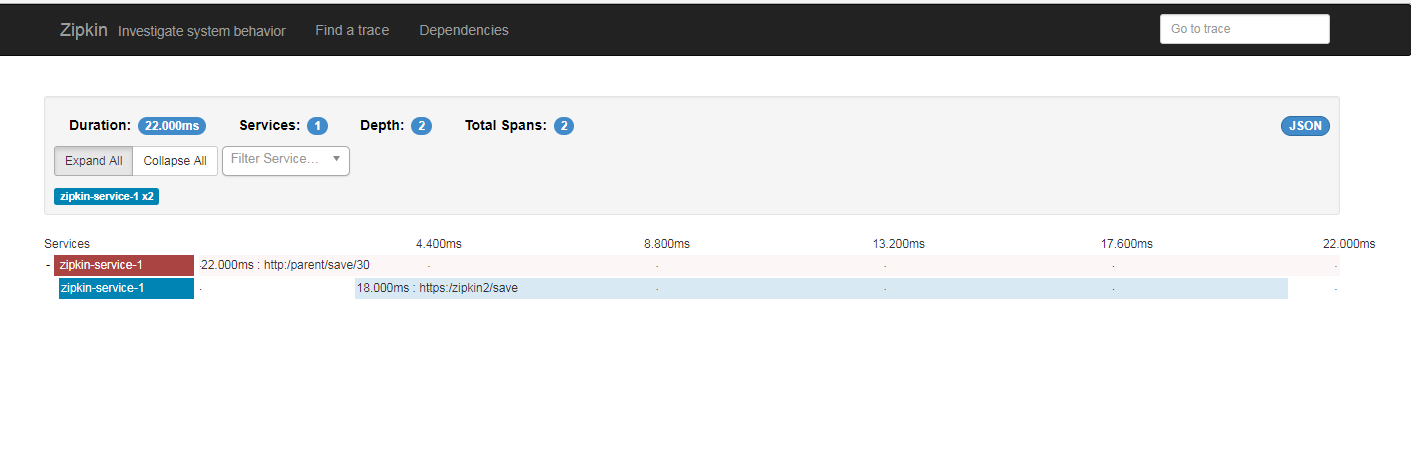
Filter with the time and click on find trace :



Now can able to see that span with red color indicate that there is some error:

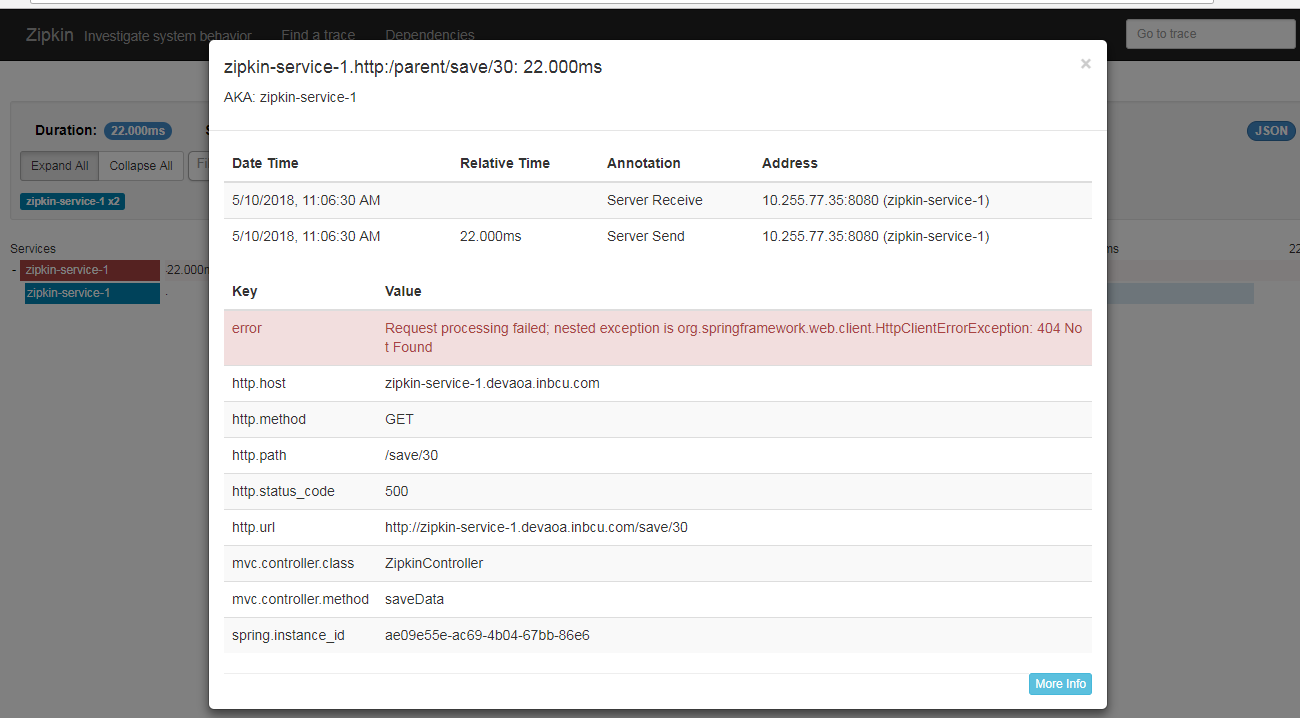
**Click on that span:**

**Note**: Since there is **no Zipkin service 2 span** that’s the first thing we need to note down that Zipkin service 2 is not called.



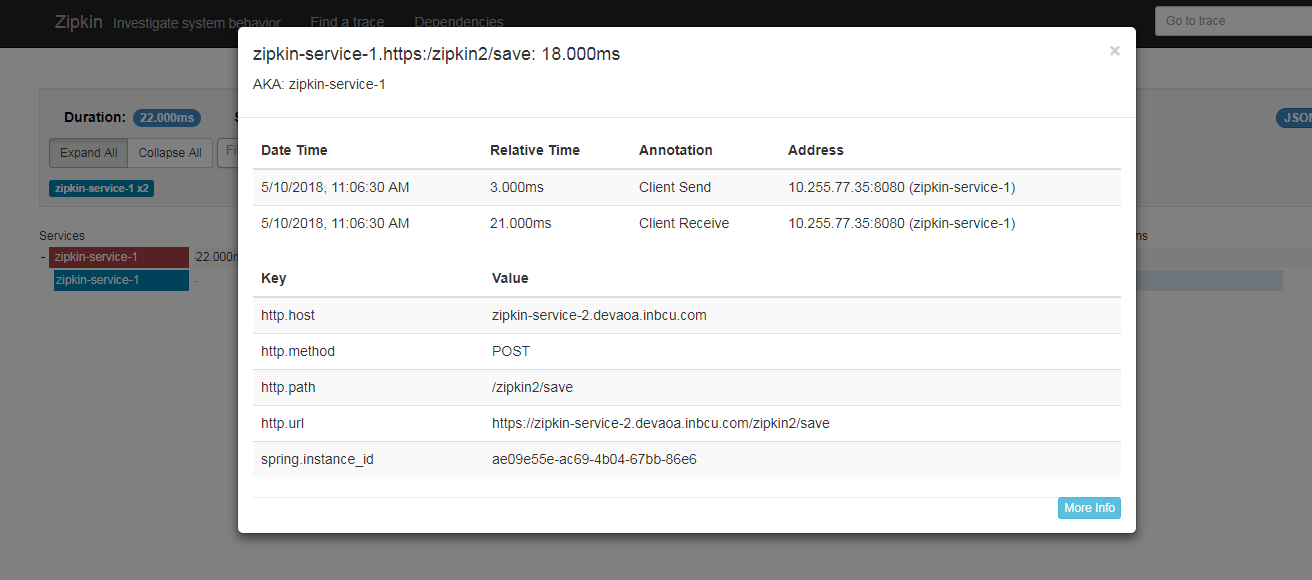
**Click on Zipkin service 1 first span:**

You can see the error message



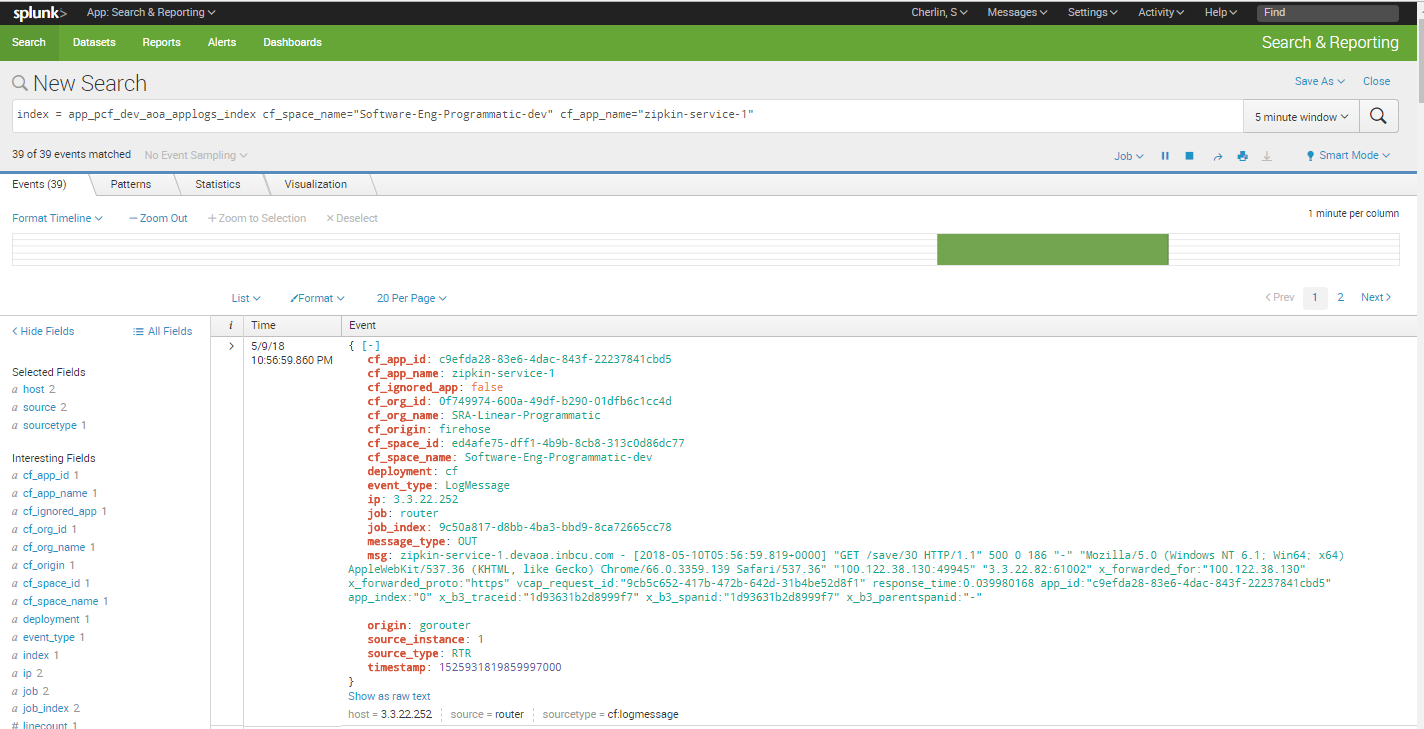
**Click on Zipkin service 1 second span:**

U can able to see that Zipkin service 2 is not called. So easily u can able to trace the issue



**Splunk:**

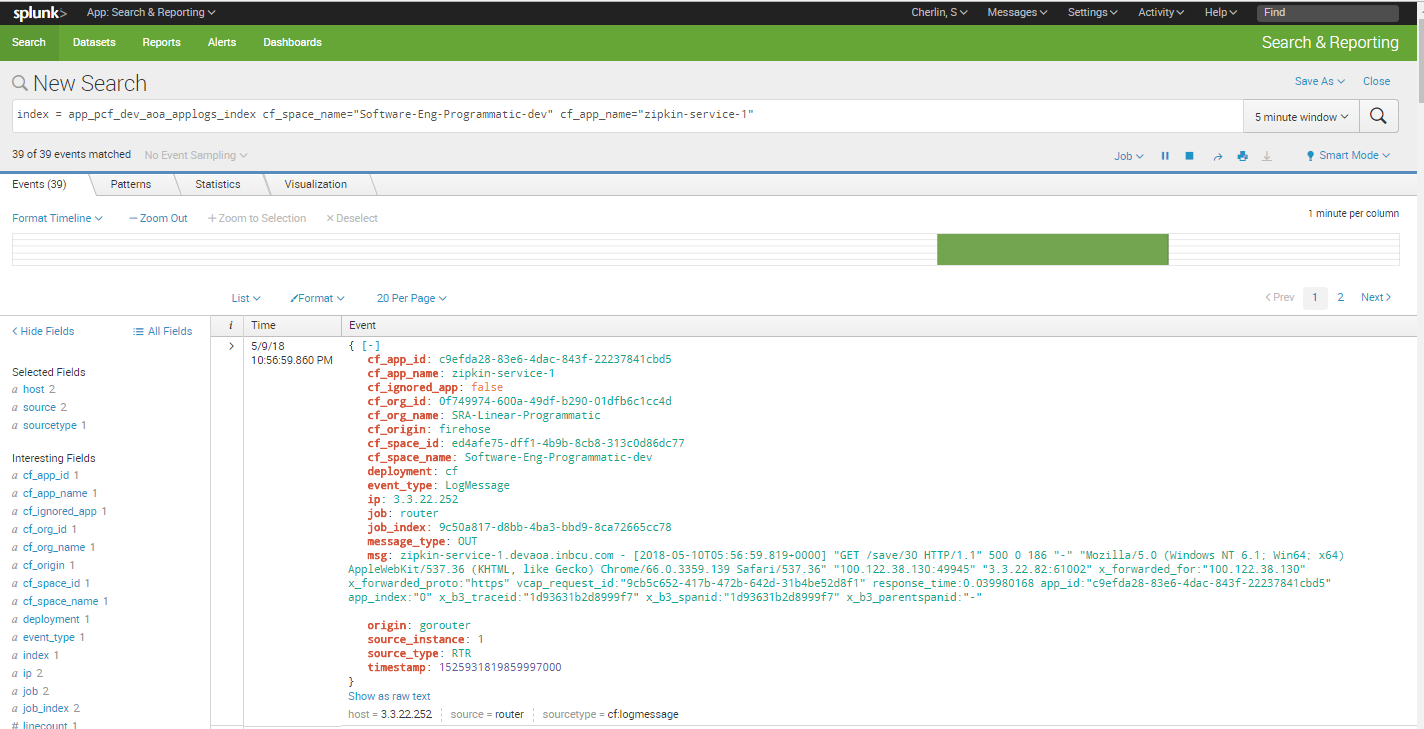
Since in the Zipkin Service 1 , **sleuth dependency** is added it will create a trace id as shown below



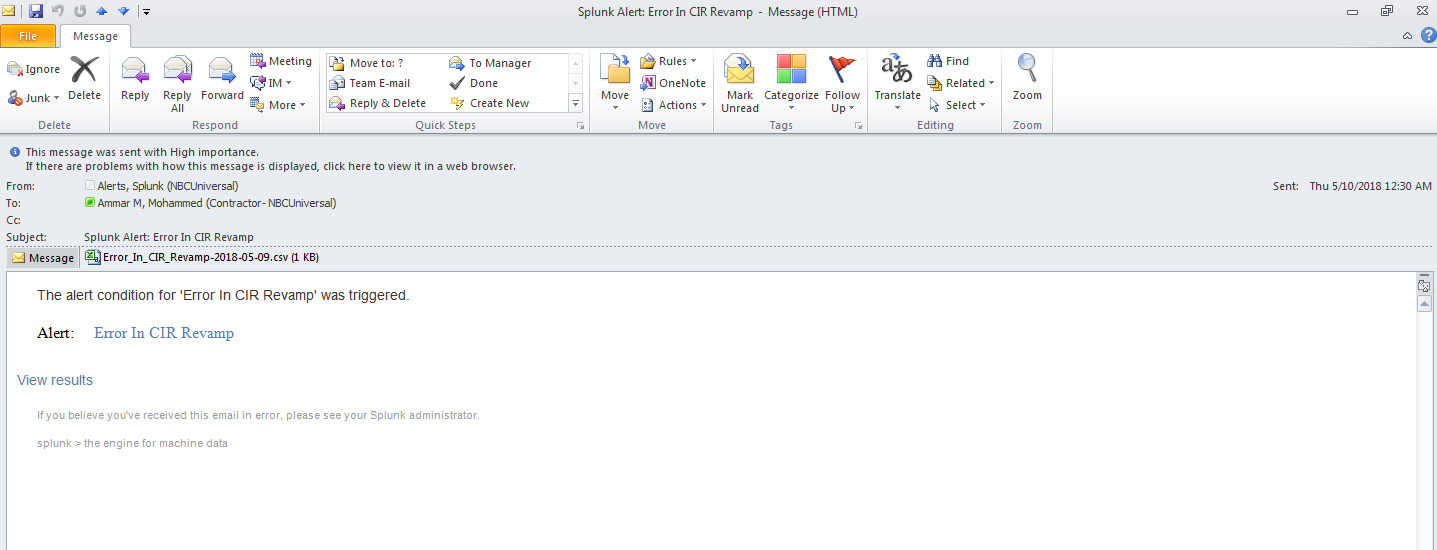
In Splunk we can create an alert for service failed with the status code (**404** OR **500** OR **503** ) )

**Query Example:**

|  |
| --- |
| index = app\_pcf\_dev\_aoa\_applogs\_index cf\_space\_name="Software-Eng-Programmatic-qa" cf\_app\_name="cir-revamp-qa" error OR failed OR severe OR ( sourcetype=access\_\* ( 404 OR 500 OR 503 ) ) |

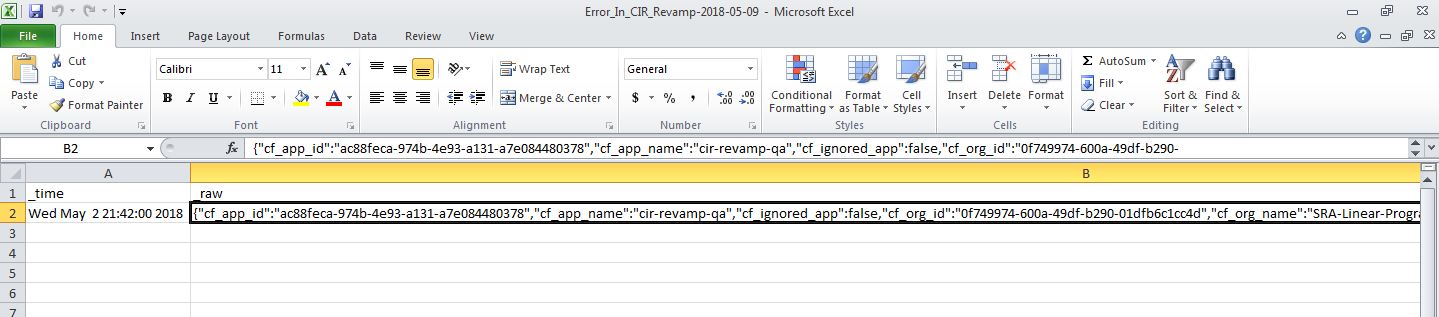


**So we will get an alert mail ,with the error message attached in the csv or excel file.**



****

**So open excel:**



**Copy the raw data and find the trace id:**

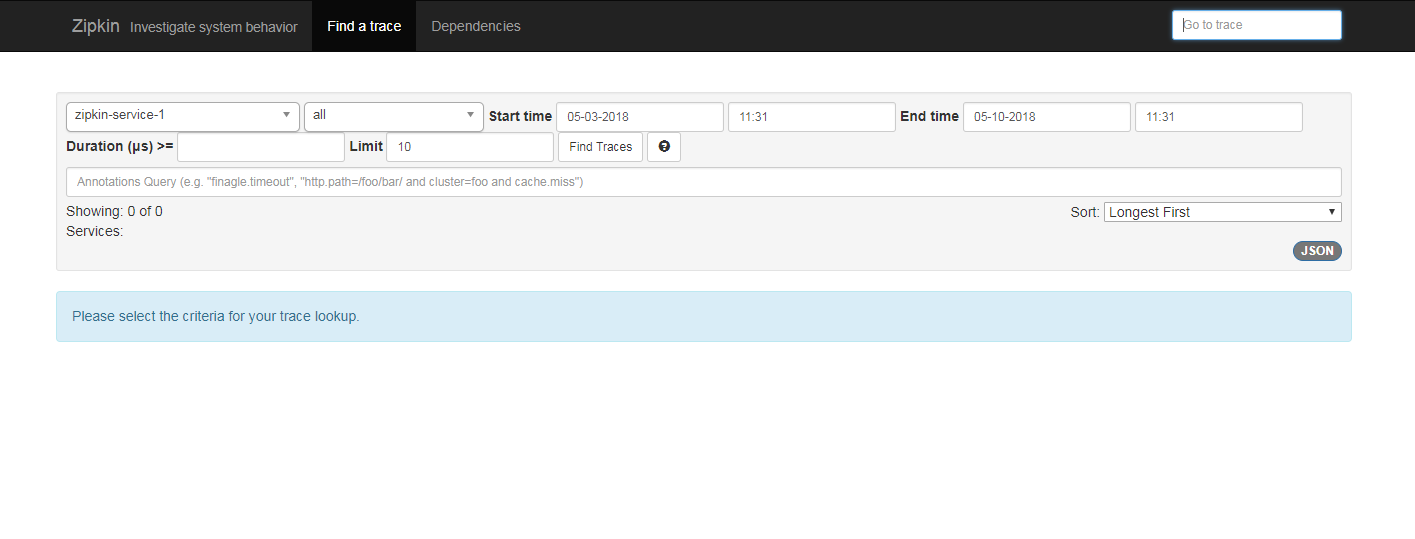
|  |
| --- |
| {"cf\_app\_id":"c9efda28-83e6-4dac-843f-22237841cbd5","cf\_app\_name":"zipkin-service-1","cf\_ignored\_app":false,"cf\_org\_id":"0f749974-600a-49df-b290-01dfb6c1cc4d","cf\_org\_name":"SRA-Linear-Programmatic","cf\_origin":"firehose","cf\_space\_id":"ed4afe75-dff1-4b9b-8cb8-313c0d86dc77","cf\_space\_name":"Software-Eng-Programmatic-dev","deployment":"cf","event\_type":"LogMessage","ip":"3.3.22.252","job":"router","job\_index":"9c50a817-d8bb-4ba3-bbd9-8ca72665cc78","message\_type":"OUT","msg":"zipkin-service-1.devaoa.inbcu.com - [2018-05-10T05:56:59.819+0000] \"GET /save/30 HTTP/1.1\" 500 0 186 \"-\" \"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36\" \"100.122.38.130:49945\" \"3.3.22.82:61002\" x\_forwarded\_for:\"100.122.38.130\" x\_forwarded\_proto:\"https\" vcap\_request\_id:\"9cb5c652-417b-472b-642d-31b4be52d8f1\" response\_time:0.039980168 app\_id:\"c9efda28-83e6-4dac-843f-22237841cbd5\" app\_index:\"0\" x\_b3\_traceid:\"1d93631b2d8999f7\" x\_b3\_spanid:\"1d93631b2d8999f7\" x\_b3\_parentspanid:\"-\"\n","origin":"gorouter","source\_instance":"1","source\_type":"RTR","timestamp":1525931819859997041} |

**So there you will get the trace id of the failed Microservice**

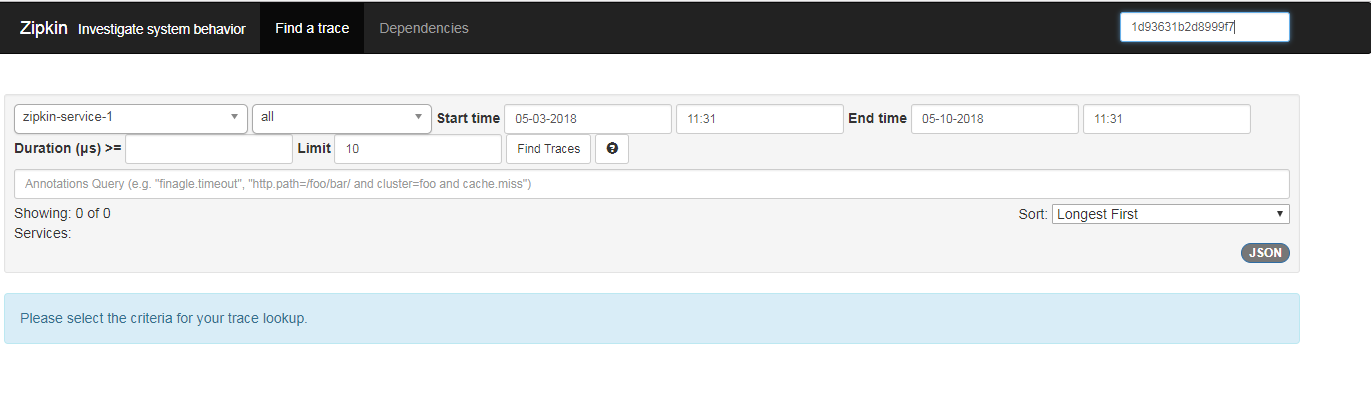
So what we can do is that copy that trace id, go to the Zipkin Dashboard and check the flow where it is failed:

**Trace Id:** 1d93631b2d8999f7

**Go to the Zipkin Dashboard:**

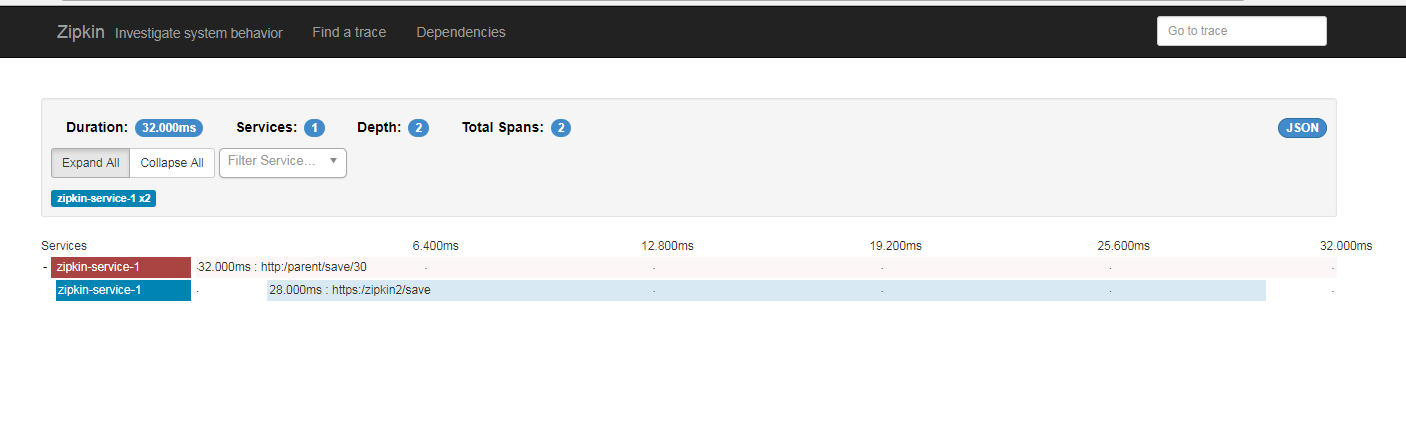


**Paste the trace id:**

**Hit enter:**

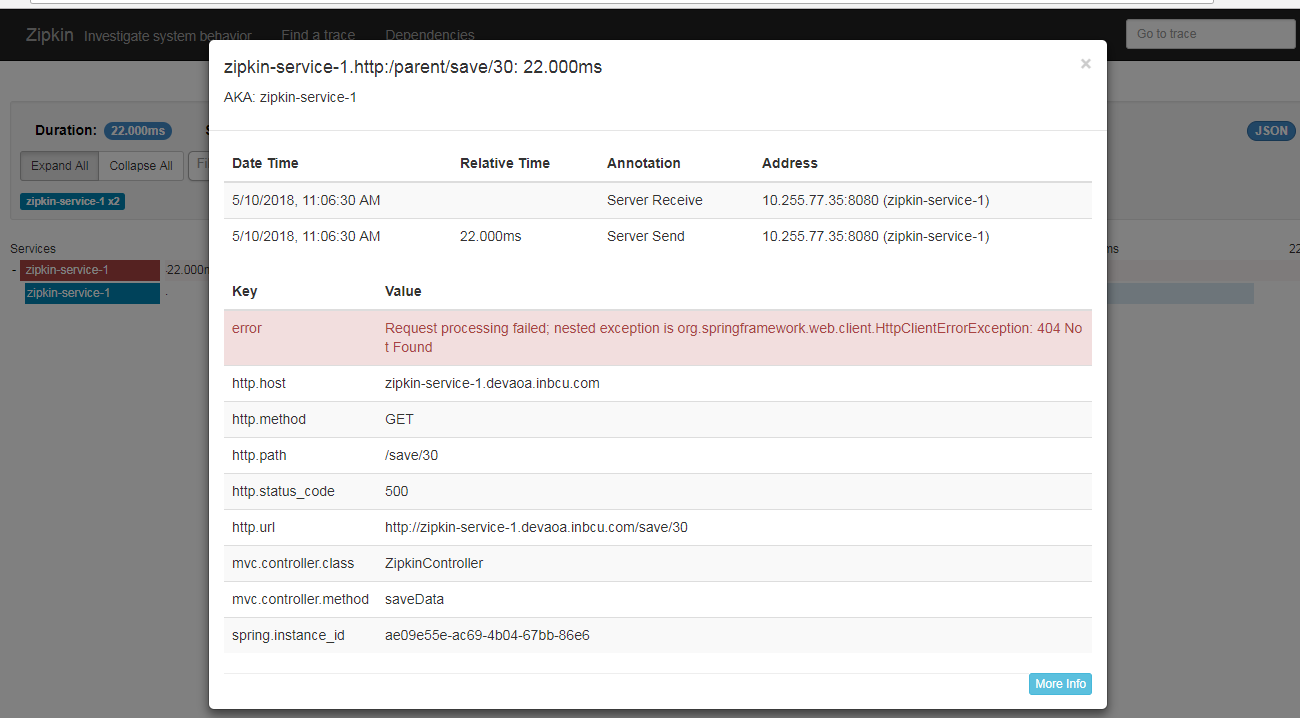
So you will get the service failed details as shown below

**Note**: Since there is **no Zipkin service 2 span** that’s the first thing we need to note down that Zipkin service 2 is not called.



**Click on Zipkin service 1 first span:**

You can see the error message



**Click on Zipkin service 1 second span:**

U can able to see that Zipkin service 2 is not called. So easily u can able to trace the issue

