

# Oracle® Banking Microservices Architecture

## ANNEXURE - 2



Release 14.6.1.0.0

F61114-01

August 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

F61114-01

Copyright © 2018, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Purpose	iv
Audience	iv
Acronyms and Abbreviations	iv
List of Topics	iv
Related Documents	v

## 1 Document Tracing Zipkin

---

1.1 Zipkin Installation	1-1
1.1.1 Download the Zipkin	1-1
1.1.2 Configure the Zipkin	1-1
1.2 Troubleshoot Zipkin	1-2

## 2 Monitor ELK

---

2.1 Architecture	2-1
2.2 Install and Configure ELK	2-2
2.2.1 Start Elastic Search	2-3
2.2.2 Setup and Start Logstash	2-3
2.2.3 Setup and Start Kibana	2-4

## Index

---

# Preface

- [Purpose](#)
- [Audience](#)
- [Acronyms and Abbreviations](#)
- [List of Topics](#)
- [Related Documents](#)

## Purpose

This guide is a supporting document for the installation of Zipkin and ELK. The user can find the reference in the respective installation guides.

## Audience

This guide is intended for WebLogic admin or ops-web team who are responsible for installing OFSS Banking Products.

## Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

**Table 1 Acronyms and Abbreviations**

Abbreviation	Description
ELK	Elasticsearch, Logstash, Kibana

## List of Topics

This guide is organized as follows:

**Table 2 List of Topics**

Topics	Description
<a href="#">Document Tracing Zipkin</a>	This topic provides the information about installation and tracing the Zipkin.
<a href="#">Monitor ELK</a>	This topic provides the information about monitoring the ELK.

## Related Documents

For more information, refer to the following documents:

- Product Installation Guide

# 1

## Document Tracing Zipkin

This topic describes the procedure to install and run the document tracing zipkin.

- [Zipkin Installation](#)  
This topic describes the systematic instructions to download and install the zipkin.
- [Troubleshoot Zipkin](#)  
This topic describes the systematic instructions to troubleshoot using Zipkin Traces.

### 1.1 Zipkin Installation

This topic describes the systematic instructions to download and install the zipkin.

- [Download the Zipkin](#)  
This topic provides systematic instructions to download the Zipkin.
- [Configure the Zipkin](#)  
This topic provides systematic instructions to configure the Zipkin.

#### 1.1.1 Download the Zipkin

This topic provides systematic instructions to download the Zipkin.

The Zipkin works as an independent application and can be downloaded as a runnable jar from the official website of Zipkin <https://zipkin.io/>. The latest version of Zipkin needs a Java version above 8.



#### Note:

For the exact version to be installed, refer to **Software Prerequisites** section in **Release Notes**.

1. Download the Zipkin from the direct link of jar is [https://search.maven.org/remote\\_content?g=io.zipkin&a=zipkin-server&v=LATEST&c=exec](https://search.maven.org/remote_content?g=io.zipkin&a=zipkin-server&v=LATEST&c=exec).
2. Execute the downloaded jar using the `java -jar JAR_NAME` command.

#### 1.1.2 Configure the Zipkin

This topic provides systematic instructions to configure the Zipkin.

The configuration of Zipkin can be done with environment variables.

1. Set the port of the Zipkin using the `QUERY_PORT` environment variable.  
The application starts on the port number assigned for `QUERY_PORT` environment variable or its default value of 9411.
2. Access the web UI of Zipkin at `http://localhost:PORT`.

## 1.2 Troubleshoot Zipkin

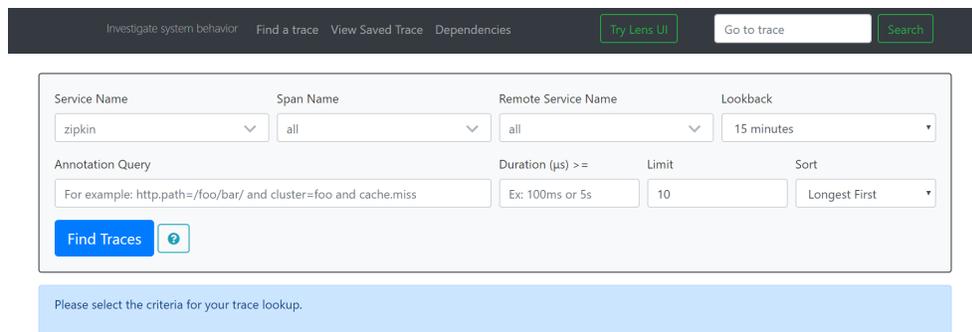
This topic describes the systematic instructions to troubleshoot using Zipkin Traces.

1. Launch the Zipkin URL.

 **Note:**

The basic layout of Zipkin is shown below.

**Figure 1-1** Layout of Zipkin



The screenshot shows the Zipkin search interface. At the top, there are navigation links: "Investigate system behavior", "Find a trace", "View Saved Trace", and "Dependencies". To the right, there are buttons for "Try Lens UI", "Go to trace", and "Search". Below this is a search form with the following fields:

Service Name	Span Name	Remote Service Name	Lookback
zipkin	all	all	15 minutes

Below the dropdowns are fields for "Annotation Query" (with an example: "http.path=/foo/bar/ and cluster=foo and cache.miss"), "Duration (µs) >=" (with an example: "100ms or 5s"), "Limit" (set to "10"), and "Sort" (set to "Longest First"). A "Find Traces" button and a help icon are at the bottom of the form. A blue message box below the form says: "Please select the criteria for your trace lookup."

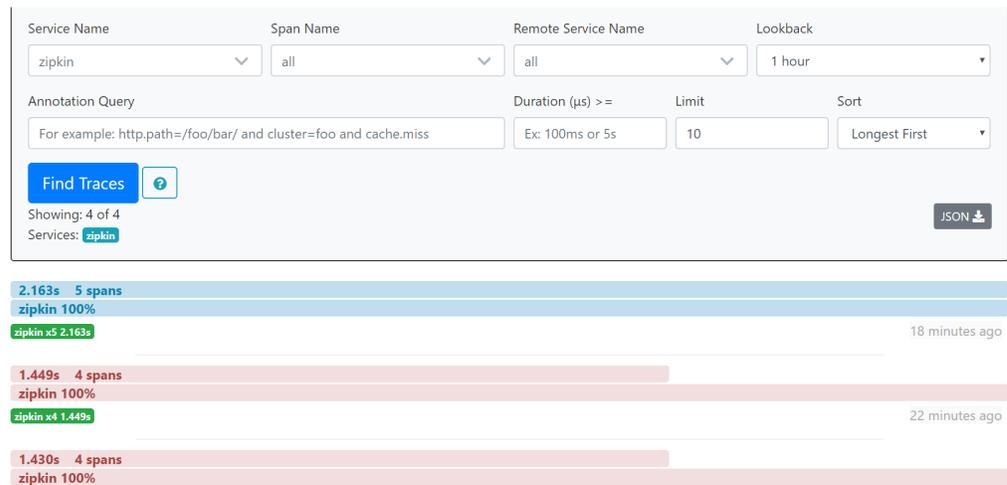
2. Use **Search** to find the traces of required API calls and services.

 **Note:**

The search options given in the user interface are self-explanatory, and there is another UI option (**Try Lens UI**). It is given a different user interface with the same functionality.

Some error API calls are made to showcase how to track errors. The blue listing shows the successful API hits, and the red listing indicates the errors. Each block indicates a single trace in the listing. The below figure shows the list of traces.

**Figure 1-2 List of Traces**

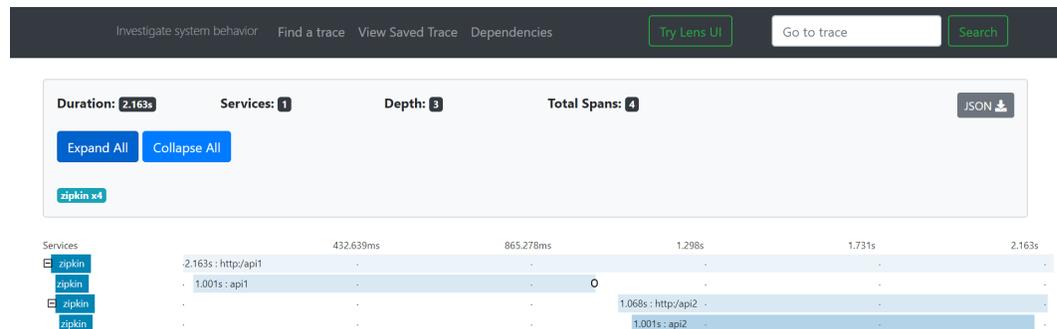


**3. Open an individual trace.**

It describes the time taken for each block. As the two custom spans are created inside two service calls, user can find a total of four blocks.

The time taken for an individual block is shown below.

**Figure 1-3 Individual Trace**



**4. Click an individual block to display the details.**

**Figure 1-4 Details of Individual Block**

Date Time	Relative Time	Annotation	Address
9/10/2019, 4:11:23 PM		Server Start	10.184.89.16:8080 (zipkin)
9/10/2019, 4:11:25 PM	2.163s	Server Finish	10.184.89.16:8080 (zipkin)

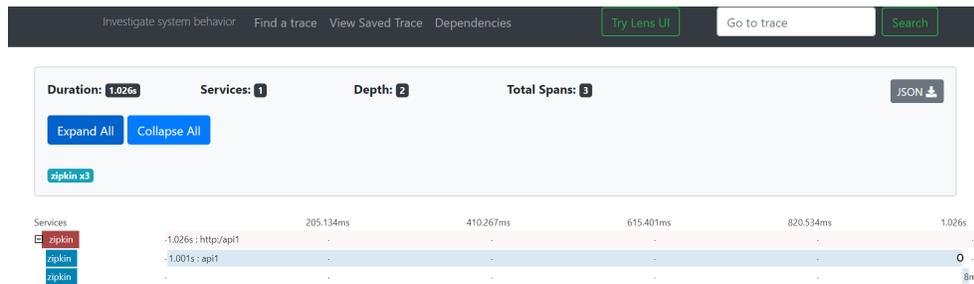
Key	Value
http.host	localhost
http.method	GET
http.path	/api1
http.status_code	200
http.url	http://localhost:8080/api1
mvc.controller.class	Controller
mvc.controller.method	api1
spring.instance_id	eswarperabathini.in.oracle.com:Zipkin

Key	Value
traceld	9d63642d72ab6f9f
spanid	9d63642d72ab6f9f

The user can also view the logging events in the Zipkin UI as small circular blocks. An example of an error log is shown below.

**Figure 1-5 Sample Error Log**



5. Click the error to get clear details and place of the error.

**Figure 1-6 Details of Error**

Date Time	Relative Time	Annotation	Address
9/11/2019, 6:09:01 PM		Server Start	10.184.89.16:8080 (zipkin)
9/11/2019, 6:09:02 PM	1.026s	Server Finish	10.184.89.16:8080 (zipkin)

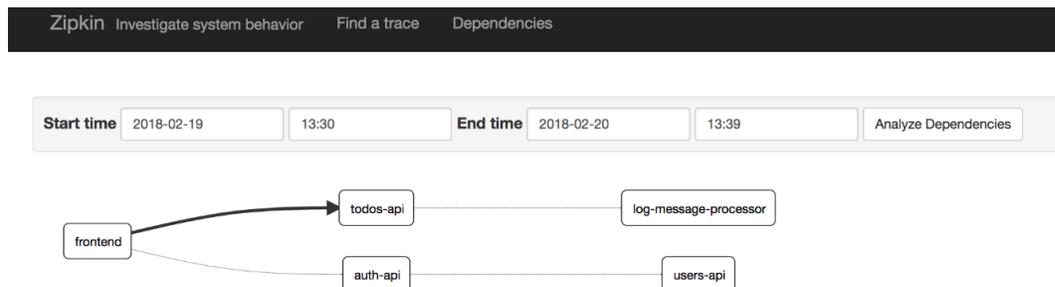
  

Key	Value
error	Request processing failed; nested exception is org.springframework.web.client.HttpServerErrorException: 500 null
http.host	localhost
http.method	GET
http.path	/api1
http.status_code	500
http.url	http://localhost:8080/api1
mvc.controller.class	BasicErrorController
mvc.controller.method	errorHtml
spring.instance_id	eswarperabathini.in.oracle.com:Zipkin

If the Lens UI is used in Zipkin, the above figures are not applicable but are relatable to the Lens UI as well. Traces of the application can be found using Traceld. The Traceld can be found in the debug logs of the deployment when spring-cloud-sleuth is included in the dependencies (included in spring-cloud-starter-Zipkin dependency).

6. Click **Dependencies** to get the dependency graph information between micro-services.

**Figure 1-7 Sample Dependency Graph**



# 2

## Monitor ELK

This topic describes the procedure for installing and configuring the ELK.

The ELK Stack is a collection of the following open-source products:

- **Elasticsearch:** It is an open-source, full-text search, and analysis engine based on the Apache Lucene search engine.
- **Logstash:** Logstash is a log aggregator that collects data from various input sources, executes different transitions and enhancements, and then transports the data to various supported output destinations.
- **Kibana:** Kibana is a visualization layer that works on top of Elasticsearch, providing users with the ability to analyze and visualize the data.

These components together are most commonly used for monitoring, troubleshooting, and securing IT environments. Logstash takes care of data collection and processing, Elasticsearch indexes and stores the data, and Kibana provides a user interface for querying the data and visualizing it.

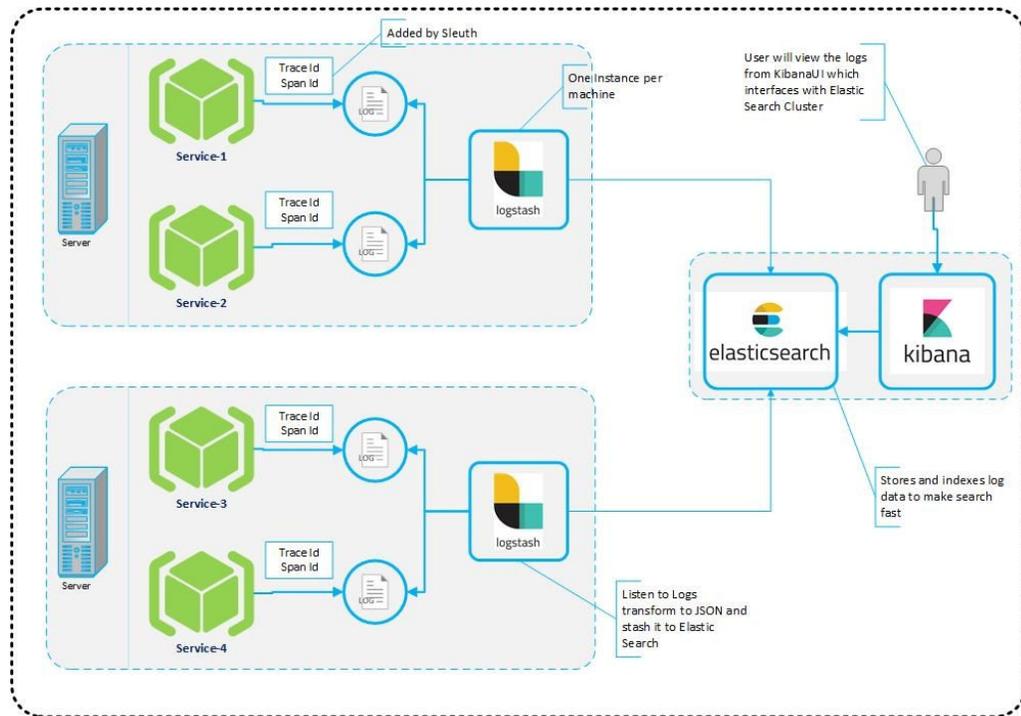
- [Architecture](#)  
This topic describes about architecture.
- [Install and Configure ELK](#)  
This topic describes about the installation and configuration of ELK.

### 2.1 Architecture

This topic describes about architecture.

It provides a comprehensive solution for handling all the required facets.

Figure 2-1 Architecture



Spring Cloud Sleuth also provides additional functionality to trace the application calls by providing us with a way to create intermediate logging events. Therefore, Spring Cloud Sleuth dependency must be added to the applications.

## 2.2 Install and Configure ELK

This topic describes about the installation and configuration of ELK.

### Note:

To install and configure ELK Stack, make sure the versions of the three software's are the same. For the exact version to be installed, refer to Software Prerequisites section in Release Notes.

The user must download the latest version for all three software's and for installation guides, refer to links below:

- Logstash: <https://www.elastic.co/guide/en/logstash/current/installing-logstash.html>
- Elastic Search: <https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html>
- Kibana: <https://www.elastic.co/guide/en/kibana/current/install.html>
- [Start Elastic Search](#)  
This topic provides systematic instructions to start Elastic Search.

- [Setup and Start Logstash](#)  
This topic provides the systematic instructions to setup and start Logstash.
- [Setup and Start Kibana](#)  
This topic provides the systematic instructions to setup and start Kibana.

## 2.2.1 Start Elastic Search

This topic provides systematic instructions to start Elastic Search.

1. Navigate to Elasticsearch root folder.
2. Use `nohup` to start the Elasticsearch process.

```
> nohup ./bin/elasticsearch
```

## 2.2.2 Setup and Start Logstash

This topic provides the systematic instructions to setup and start Logstash.

1. Create a new **logstash.conf** file that provides the required file parsing and integration for Elasticsearch.

### logstash.conf:

```
#Point to the application logs
input {
  file {
    type => "java"
    path => "/scratch/app/work_area/app_logs/*.log"
    codec => multiline {
      pattern => "^%{YEAR}-%{MONTHNUM}-%{MONTHDAY} %{TIME}.*"
      negate => "true"
      what => "previous"
    }
  }
}

#Provide the parsing logic to transform logs into JSON
filter {
  #If log line contains tab character followed by 'at' then we will tag
  that entry as stacktrace
  if [message] =~ "\tat" {
    grok {
      match => ["message", "^(\\tat)"]
      add_tag => ["stacktrace"]
    }
  }
}

#Grokking Spring Boot's default log format
grok {
  match => [ "message",
    "(?<timestamp>%{YEAR}-%{MONTHNUM}-%{MONTHDAY}
%{TIME}) %{LOGLEVEL:level} %{NUMBER:pid} --- \[(?<thread>[A-Za-z0-9-]+)\]
[A-Za-z0-9.]*\.(?<class>[A-Za-z0-9#_]+)\s*:\s+(?<logmessage>.*)",
    "message",
```

```

                "(?<timestamp>{%YEAR}-{%MONTHNUM}-{%MONTHDAY} {%TIME})
%{LOGLEVEL:level} %{NUMBER:pid} --- .+? :\s+(?<logmessage>.*)"
            ]
        }
        # pattern matching logback pattern
        grok {
            match =>
            { "message" => "%{TIMESTAMP_ISO8601:timestamp}\s+%
{LOGLEVEL:severity}\s+\[%{DATA:service},%{DATA:trace},%{DATA:span},%
{DATA:exportable}\]\s+\[%{DATA:environment}\]\s+\[%{DATA:tenant}\]\
\s+\[%{DATA:user}\]\s+\[%{DATA:branch}\]\s+{%DATA:pid}\s+---\s+\[%
{DATA:thread}\]\s+{%DATA:class}\s+:\s+{%GREEDYDATA:rest}"
            }
        }
        #Parsing out timestamps which are in timestamp field thanks to
previous grok section
        date {
            match => [ "timestamp" , "yyyy-MM-dd HH:mm:ss.SSS" ]
        }
    }
    #Ingest logs to Elasticsearch
    output {
        elasticsearch { hosts => ["localhost:9200"] }
        stdout { codec => rubydebug }
    }
}

```

2. Start the Logstash process using below command.

```
>nohup ./bin/logstash -f logstash.conf
```

## 2.2.3 Setup and Start Kibana

This topic provides the systematic instructions to setup and start Kibana.

1. Navigate to the **kibana.yml** available under <kibana\_setup\_folder>/config.
2. Modify the file to include the below:

```

#Uncomment the below line and update the IP address to your host
machine IP.
server.host: "xx.xxx.xxx.xx"
#Provide the elasticsearch url. If this is running on the same
machine then you can use the below config as is
elasticsearch.url: "http://localhost:9200"

```

3. Start the Kibana process using the below command.

```
>nohup ./bin/kibana
```



# Index

## A

---

Architecture, [2-1](#)

## C

---

Configure the Zipkin, [1-1](#)

## D

---

Document Tracing Zipkin, [1-1](#)

Download the Zipkin, [1-1](#)

## I

---

Install and Configure ELK, [2-2](#)

## M

---

Monitor ELK, [2-1](#)

## S

---

Setup and Start Kibana, [2-4](#)

Setup and Start Logstash, [2-3](#)

Start Elastic Search, [2-3](#)

## T

---

Troubleshoot Zipkin, [1-2](#)

## Z

---

Zipkin Installation, [1-1](#)