**Zipkin** is a **Java-based distributed tracing system** that helps collect and analyze timing data in service architectures, aiding in troubleshooting latency issues. [It allows you to track requests across microservices and understand where time is spent during their execution1](https://www.sentinelone.com/blog/zipkin-tutorial-distributed-tracing/).

Here are **five free resources** to learn more about Zipkin:

1. [**Zipkin Tutorial: Get Started Easily With Distributed Tracing**](https://www.sentinelone.com/blog/zipkin-tutorial-distributed-tracing/): This tutorial provides an overview of Zipkin, its importance in microservices, and practical examples of sending and viewing traces[1](https://www.sentinelone.com/blog/zipkin-tutorial-distributed-tracing/).
2. [**Spring Cloud Zipkin and Sleuth - Examples Java Code Geeks**](https://examples.javacodegeeks.com/spring-cloud-zipkin-and-sleuth-example/): Learn how to integrate Zipkin with Spring Cloud using Sleuth for distributed tracing in Java applications[2](https://www.baeldung.com/tracing-services-with-zipkin).
3. [**Zipkin on StackShare**](https://www.stackshare.io/zipkin): Explore Zipkin’s features, pros, and cons, along with real-world examples from companies using it in their tech stacks[3](https://www.stackshare.io/zipkin).
4. [**Zipkin Official Documentation**](https://zipkin.io/): The official Zipkin documentation covers installation, configuration, and usage details. [You can find quick-start guides, examples, and more](https://www.sentinelone.com/blog/zipkin-tutorial-distributed-tracing/)[4](https://zipkin.io/).
5. [**GitHub Repository for Zipkin**](https://github.com/openzipkin/zipkin): Dive into the source code, contribute, and explore various aspects of Zipkin on its GitHub repository[5](https://github.com/openzipkin/zipkin).

Remember to explore these resources to gain a deeper understanding of Zipkin and its role in distributed systems! 🚀