31 Zipkin 入门介绍

更新时间：2019-07-29 10:32:46

> 每个人的生命都是一只小船，理想是小船的风帆。
>
> ——张海迪

12-1 Zipkin 入门介绍

Spring Cloud 中的各大组件我们已经介绍了很多，相信经过前面的学习大家对于 Spring Cloud 的组件生态也已经有了一个基本的认知，可以在心中绘制出一副 Spring Cloud 生态图了。不过我们的介绍仍未结束，今天要和大家再来聊一个组件，那就是 Zipkin。

Zipkin 简介

Zipkin 架构从整体上来说，可以分为 Zpikin Server 和 Zipkin Client 两部分，其中 Zipkin Server 提供了数据采集、分析以及展示等功能，Zipkin Client 则是一些机遇不同开发语言封装的客户端工具，这些工具用来实现追踪数据的生成与上报功能，反映到我们的微服务架构中，一个一个的微服务就相当于 Zipkin Client ，下面对这两个，我们分别来介绍。

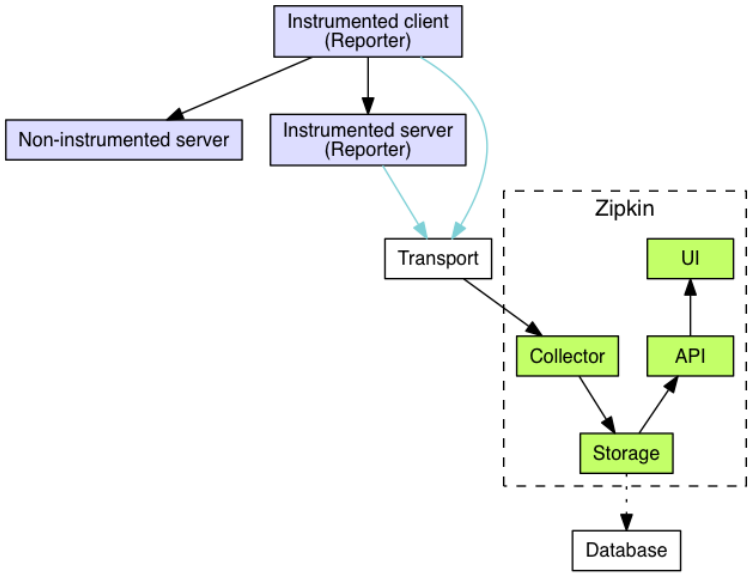
Zipkin Server

Zipkin 是一个开放源代码的分布式跟踪系统，由Twitter公司开源，它致力于收集服务的定时数据，以解决微服务架构中的延迟问题，包括数据的收集、存储、查找和展现。

每个服务向 Zipkin 报告计时数据，Zipkin 会根据调用关系通过 Zipkin UI 生成依赖关系图，显示多少跟踪请求通过每个服务。该系统让开发者可通过一个 Web 前端轻松的收集和分析数据，例如用户每次请求服务的处理时间等，来方便的监测系统中存在的瓶颈。

Zipkin 提供了可插拔数据存储方式：In-Memory、MySql、Cassandra 以及 Elasticsearch ，一般在生产环境中我们推荐使用 Elasticsearch。

我们先来看一张 Zipkin 的工作流程图：



这是一张来自 Zipkin 官网的工作流程图，从这个图中，我们可以看到 Zipkin 一共有如下四个工作模块：

- Collector

Collector 主要用来收集数据，一旦跟踪数据到达 Zipkin 的 Collector，它就会被 Collector 验证、存储和索引以支持后面的操作。

- Storage

Zipkin 最初是为了在 Cassandra 上存储数据而构建的，因为 Cassandra 是可扩展的，具有灵活的模式，并且在 Twitter 中被大量使用。但Zipkin 中的存储组件也是可插拔的，除了 Cassandra，也支持 ElasticSearch 和 MySQL 等，当然这些数据也可以直接存储在内存中，而不进行持久化操作。 Storage 主要用来处理 Collector 接收到的跟踪信息，将这些跟踪信息进行持久化（如果需要的话）。

- RESTful API

一旦数据被存储和索引，我们需要一种方法来提取它，查询守护程序提供了一个简单的 JSON API，通过这些 API 可以将跟踪信息展示给客户端，此API的主要使用者是Web UI。

- Web UI

Web UI 是一个很好的数据可视化界面，它提供了一种基于服务、时间和注释查看跟踪的方法，可以非常直观的查看和分析跟踪数据。另外需要注意：UI中没有内置身份验证。

Zipkin Client

Zipkin Client 包含的客户端工具还是非常丰富的，除了官方支持的语言，还有大量社区支持的语言版本，例如 Java、C++、Python、C#、Go、JavaScript、PHP 以及 Elixir等，其中光是 Java 支持的库就不止 Spring Cloud Sleuth （虽然我们后文的案例是基于此），也还不包括 htrace、Dropwizard Zipkin、cassandra-zipkin-tracing、Wingtips 等，其中，在 Spring Cloud Sleuth 中，可以基于 HTTP 来传输数据，也可以基于消息中间件来传输数据，对 Java 版本的要求是至少 Java7 以上。

> 编著@

> 其中光是 Java 支持的库就不止 Spring Cloud Sleuth （虽然我们后文的案例是基于此），也还不包括 htrace、Dropwizard Zipkin、cassandra-zipkin-tracing、Wingtips 等
>
> 麻烦确定此句表述是否有问题。

大家在网上看到的关于 Zipkin 的教程，大部分都会教你先搭建一个 Zipkin Server ，这种方式也不能算错误，只是它是一种过时的写法。从 Spring Cloud Finchley 版本开始，这种方式就已经不被推荐了，而是需要我们单独的安装 Zipkin 软件来实现数据的收集。大家在创建一个 Spring Boot 项目时，如果采用了最新版本，你会发现在官方可选的依赖中就没有 Zipkin 的依赖，因此，我们在学习 Zipkin 的过程中，需要先来安装相关的环境。

环境配置

环境的安装配置我们依然选择和前面一样的配置，使用 Docker 来进行安装。

在后文的案例中，我们使用 Elasticsearch 来做数据存储，使用 RabbitMQ 来做数据传输，因此在安装 Zipkin 之前我们需要先确保 Elasticsearch 和 RabbitMQ 已经安装成功了。

Elasticsearch 安装

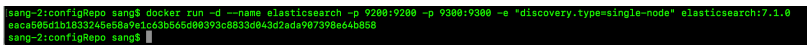
首先我们来看在 Docker 中安装 Elasticsearch ，主要装两个东西，一个是 Elasticsearch 本身，另一个是可视化工具 Elasticsearch-head ，分别来看：

Elasticsearch 安装

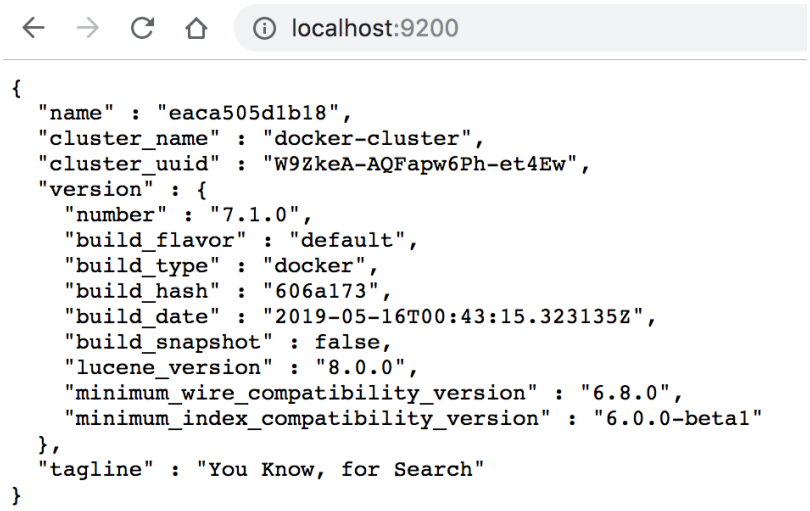
Elasticsearch 安装比较容易， Docker 中的安装命令如下(这里我们安装的是本文写作时最新的 Elasticsearch 7.1.0 版)：

```
docker run -d --name elasticsearch  -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" elasticsearch:7.1.0
```

执行结果如下：



安装成功之后，我们在浏览器中输入 http://localhost:9200 ，看到如下页面表示安装成功：



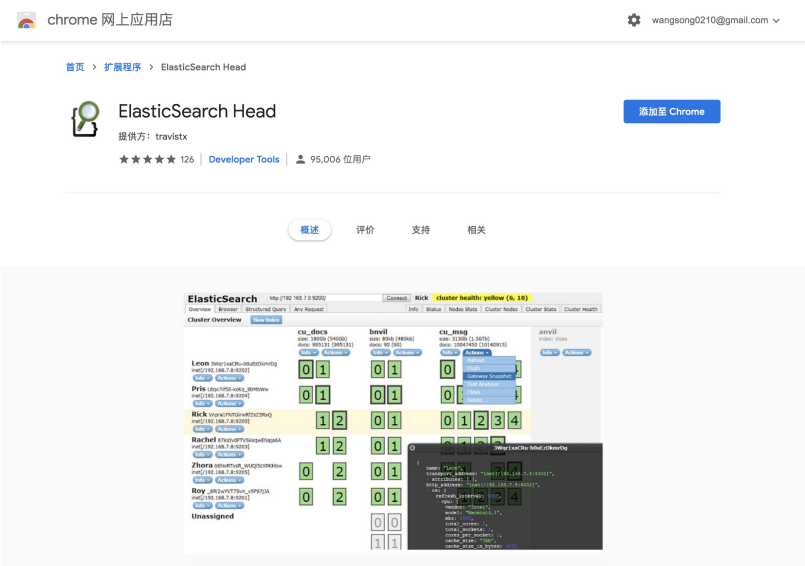
elasticsearch 本身安装成功之后，再来安装 elasticsearch-head：

Elasticsearch-head 可视化工具安装

elasticsearch-head 是一个 elasticsearch 的可视化工具，有三种方式来使用这个工具：

- 直接下载软件本身安装
- 通过 Docker 安装
- 通过安装 Chrome 插件来使用

本文采用第三种方案，直接在 Chrome 商店中搜索 elasticsearch-head ，结果如下图：



安装成功之后，打开浏览器插件，连接上我们的 elasticsearch 即可，如下：

考虑到很多人可能没办法直接从 Chrome 商店中安装 App，这里给大家推荐一个离线安装 Chrome 插件的教程：

- 如何不翻墙下载chrome插件离线安装包

读者有兴趣的话也可以安装一个 Kibana ，一样可以通过可视化页面来查看 Elasticsearch 中的数据，因为这个安装方式比较简单，我这里就不再介绍了。

RabbitMQ 安装

RabbitMQ 我们在前文已经安装过并且使用过多次了，这里我就不再重复安装的步骤了，大家只需要执行如下命令启动已有的 RabbitMQ 容器即可（如果 RabbitMQ 容器已经启动，则本步骤可以忽略）：

```
docker start some-rabbit
```

Zipkin 安装

最后，我们再来使用 Docker 安装 Zipkin ，非常容易，一行命令即可：

```
docker run -d -p 9411:9411 --name zipkin -e ES_HOSTS=192.168.0.109 -e STORAGE_TYPE=elasticsearch -e ES_HTTP_LOGGING=BASIC -e RABBIT_URI=amqp://guest:guest@192.168.0.109:5672 openzipkin/zipkin
```

虽然只有一行命令，不过这一行命令有点长，我们来给大家解释一下：

- -d 表示让容器在后台运行
- -p 表示让宿主机的 9411 端口映射到容器的 9411 端口
- –name 表示给容器取一个名字
- ES_HOSTS 表示 elasticsearch 的地址
- STORAGE_TYPE 表示容器数据的存储容器
- RABBIT_URI 表示 RabbitMQ 的地址

这行命令执行成功之后，我们就成功安装 Zipkin 了，同时还将前面安装的 Elasticsearch 和 RabbitMQ 都整合进来了，此时我们在浏览器中输入 http://localhost:9411 ,就可以看到如下页面：

看到这个页面则表示 Zipkin 已经安装成功了。

注意

在连接容器地址时，不要直接写 127.0.0.1 或者 localhost ，这样会去容器本身查找相关的应用，这里应该写宿主机的地址，才能正确找到应用。

小结

经过上面的准备工作之后，相信大家对 Zipkin 已经有一个基本认知了，同时也已经准备好 Zipkin 所需的各种环境，下篇文章我们就来看看 Zipkin 在微服务中的具体使用。

30 Spring Cloud Sleuth 实践

32 Zipkin 实践

精选留言 0

欢迎在这里发表留言，作者筛选后可公开显示

目前暂无任何讨论