基于 elasticsearch 的自定义业务告警的设计思路

作者：jianzh5
原文链接：https://ld246.com/article/1574136843350
来源网站：链滴

A系统与B系统之间有很多接口交互，但是有一段时间接口经常报错，作为开发如果不能第一时间知道题且及时解决的话就会收到业务投诉，当月绩效凉凉。

如果你也有这种场景，那么你就需要一个及时告警的功能。

实现方案

实现及时告警分以下两种场景：

- 有ELK日志收集
- 没有ELK日志收集

没有ELK日志收集的方案

~~很简单，搭建一个日志收集环境（O(∩_∩)O哈哈~）~~

需要在业务代码中嵌入硬编码，每次catch到异常直接发送告警信息告警平台进行告警

有ELK日志收集的方案

最核心的是 elasticsearch组件，所有的告警方案前提条件都是告警日志需要进ES，然后定时从ES中索出符合业务规定的告警日志（比如ERROR日志），如果检索出来的告警日志满足一定条件就触发告通知。

实现方式主要有以下几种：

- ES WATCHER

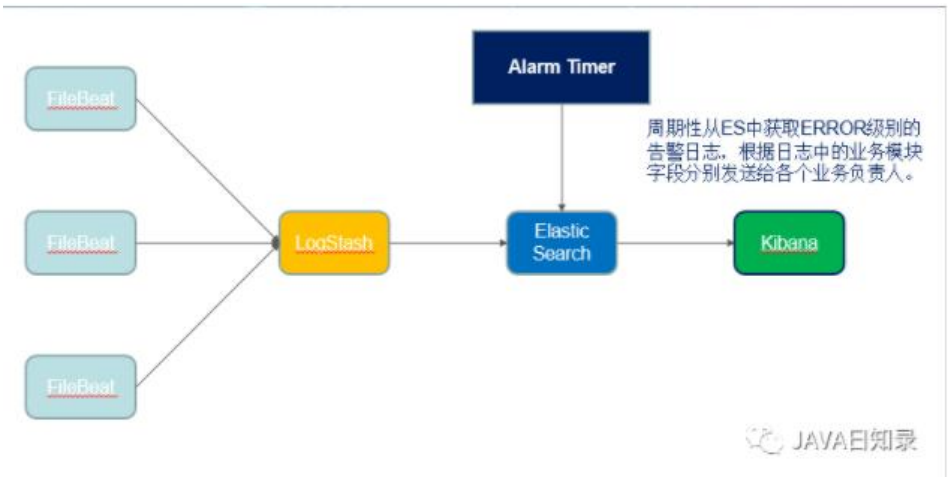
这个是elasticsearch的官方插件，它可以根据数据的变化提供警报和通知，目前是收费的，具体操作置可以参看官方地址

- elastalert

是Yelp公司基于python写的告警框架，大家可以去GitHub上查看具体使用方法。elastalert

- 自定义开发

自定义开发实现



原文链接：基于 elasticsearch 的自定义业务告警的设计思路

主要由以下几个步骤实现：

1. 分离出单独的告警日志，与业务日志分离

2. 在logstash中解析日志，构建格式化的告警日志，需要有以下几个关键参数：

日志级别、日志时间、日志描述、开发模块、关联主键、请求参数、响应参数

3. 定时任务每隔一段时间去ES中检索符合要求的日志，如果检索到就发送告警通知。

核心代码

1. 日志格式化

我们直接在客户端构建好格式化的日志，以json的形式输出到日志文件中，这样在logstash解析的时直接使用json解析即可。

这一步不是必须的，可以自由构建日志格式，然后在logstash解析的时候使用grok语法进行解析。

```
public class AlarmLog {
    /**日志级别*/
    private String logLevel;
    /**日志描述*/
    private String message;
    /**关联主键 一般使用requestId*/
    private String refCode;
    /**请求参数*/
    private String parm;
    /**响应数据*/
    private String response;
    /**开发模块,根据此参数配置模块负责人*/
    private String module;
    /**日志时间*/
    private long logTime;
    ...
}
```

2. 关键查询

在单独的定时器项目中使用如下查询语法就可以检索出具体的告警日志。检索出来就可以根据日志中模块字段找出具体的模块负责人，然后发送告警通知给负责人。

```
public List<LogDoc> findRangeLogByLevel(DateTime minRange, DateTime maxRange, String
ogLevel) {
    //需要强制转换成小写
    logLevel = logLevel.toLowerCase();
    SearchQuery searchQuery = new NativeSearchQueryBuilder()
            .withQuery(boolQuery()
                    //module 必须有值才能告警
                    .must(existsQuery("module"))
                    .must(termQuery("logLevel", logLevel))
                    .must(rangeQuery("logTime")
                            .from(minRange.getMillis())
                            .to(maxRange.getMillis())))
            .build();
```

```
    return elasticsearchTemplate.queryForList(searchQuery, LogDoc.class);
}
```