零代码如何打造自己的实时监控预警系统

原创 2017-09-11 James 一个码农的日常

概要

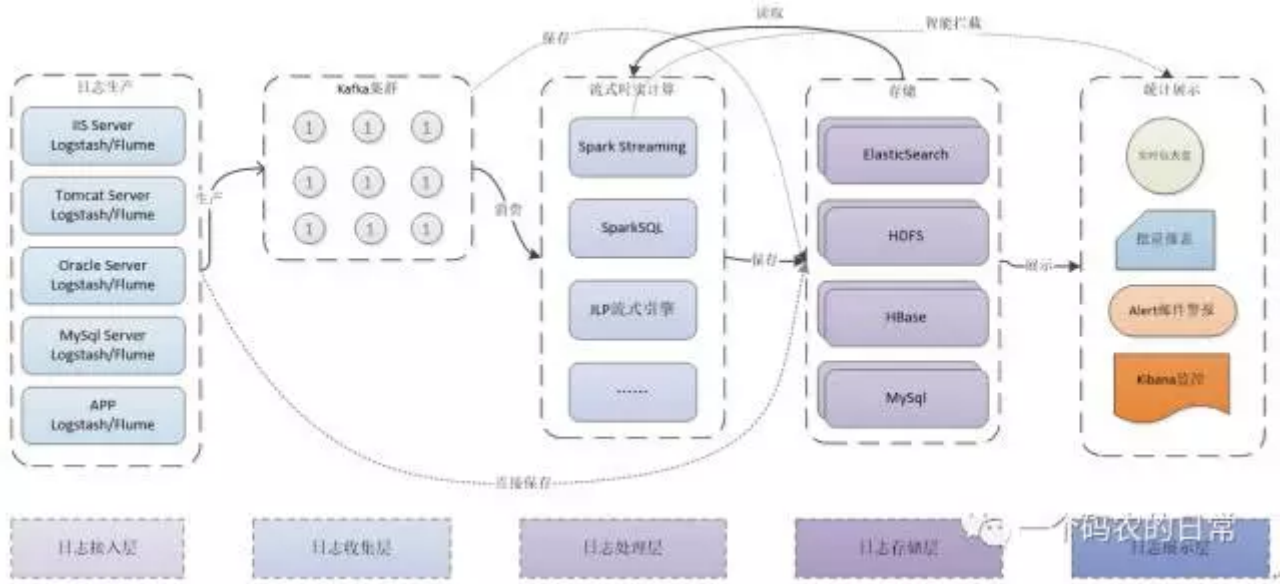
为什么要做监控

线上发布了服务，怎么知道它一切正常，比如发布5台服务器，如何直观了解是否有请求进来，访问一切正常。
当年有一次将线上的库配置到了Beta，这么低级的错误，排错花了一个通宵，十几个人。
某个核心服务挂了，导致大量报错，如何确定到底是哪里出了问题。
SOA带来的问题，调用XX服务出问题，很慢，是否可以衡量？

由于业务系统数量大，每天都会产生大量的系统日志和业务日志，单流式业务的一台服务器产生的日志达400M 想直接查看内容打开可能几分钟，而且内容之多根本无法查看，给开发和运维带来诸多不便，现业务都是分布式的，日志也是分布在每台服务器上，所以查看日志和统计更是效率低下。实时收集分布在不同节点或机器上的日志，供离线或在线查阅及分析来提升工作效率的需求异常迫切，在此背景下，特对公司统一日志平台进行初步架构设计。

在信息化时代，日志的价值是无穷的。为了对系统进行有效的监控、维护、优化、改进，都离不开对日志的收集和分析，接下来我们来看看秉着“短平快”的互联网精神，构建的这套适合现有业务系统的统一日志平台，总体分为业务日志监控平台和软硬件服务监控平台。

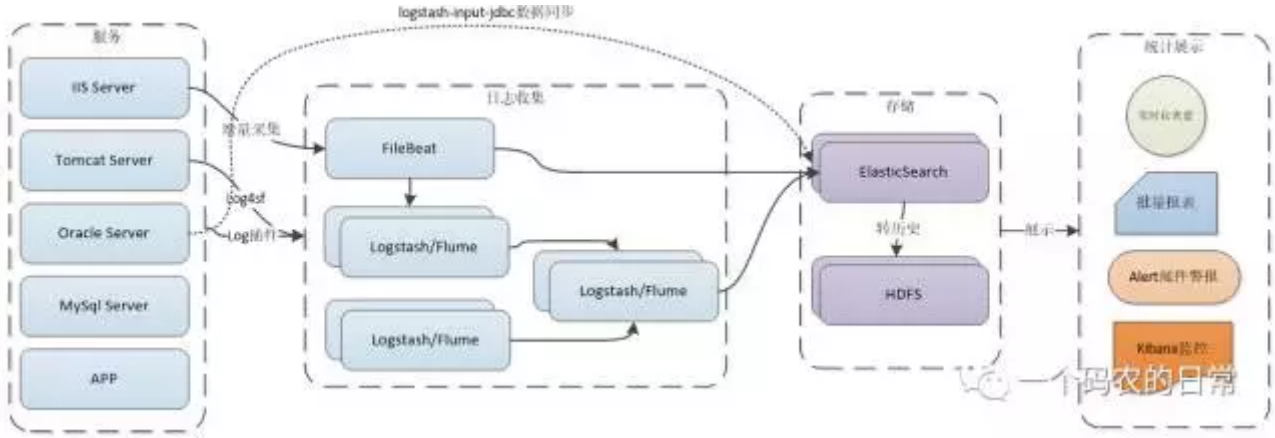
业务日志平台总体设计



以上是最终的一个最终的一个架构规划，统一日志监控系统负责将所有系统日志和业务日志集中，再通过flume或logstash上传到日志中心(kafka集群)，然后供Storm、Spark及其它系统实时

分析处理日志，或直接将日志持久化存储到HDFS供离线数据分析处理，或写入ElasticSearch提供数据查询，或直接发起异常报警或提供指标监控查询。

根据现有业务量来看，以上架构有点“重”，可以作为以后的目标，现阶段来说可以参考以下架构：



以上内容皆以配置为主，对现有业务没有影响，针对于Windows环境可以用FileBeat监控本地日志全量、增量的上传日志，对于一些稳定的日志，比如系统日志或框架日志(如HAproxy访问日志、系统异常日志等)，通过rsyslog写到本地目录local0，然后logstash根据其配置，会将local0中的增量日志上传到日志中心。Java环境下可以采用log4j直接发送到Logstash。

日志处理层

可以在Logstash中对日志作简单的分类加工处理再发送出去。

我们可以将日志聚合，根据业务不同，建立不同的索引，存入ElasticSearch提供查询。 发现异常日志时，发往监控中心，向对应的业务方发起报警，发现和预发问题的实时性提高了。统计一些访问日志或调用日志等指标信息，发往监控中心来掌握相关调用趋势。调用链开始做起来了，系统性能瓶颈一目了然了。

日志存储层

ElosticSearch中按照不同业务建索引主题（数据库），业务里面再按照需求建类型（表），不需要的历史数据可按需要持久化到HDFS，以减少ES的压力。

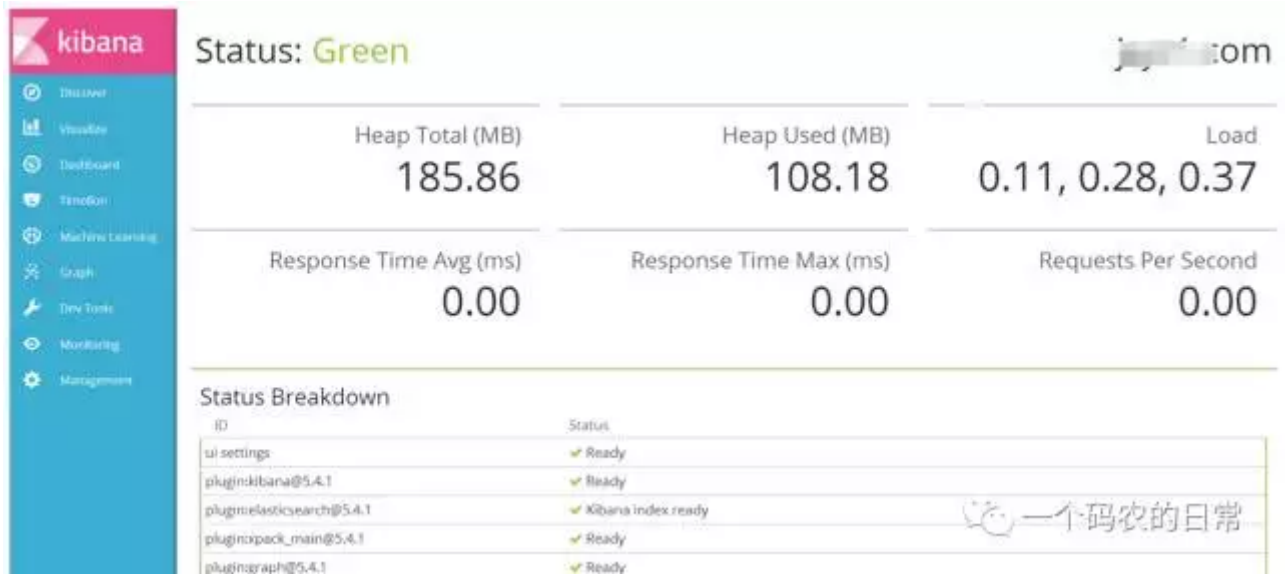
展示层Kibana

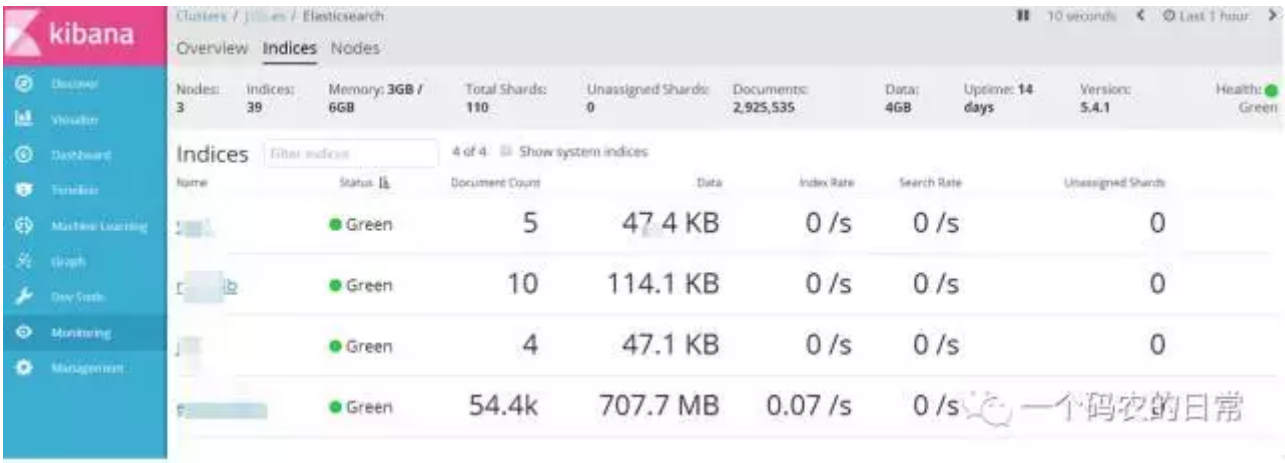
Kibana是ELK中的组件，是一个针对Elasticsearch的开源分析及可视化平台，用来搜索、查看交互存储在Elasticsearch索引中的数据。使用Kibana，可以通过各种图表进行高级数据分析及展示。

Kibana让海量数据更容易理解。它操作简单，基于浏览器的用户界面可以快速创建仪表板（dashboard）实时显示Elasticsearch查询动态。

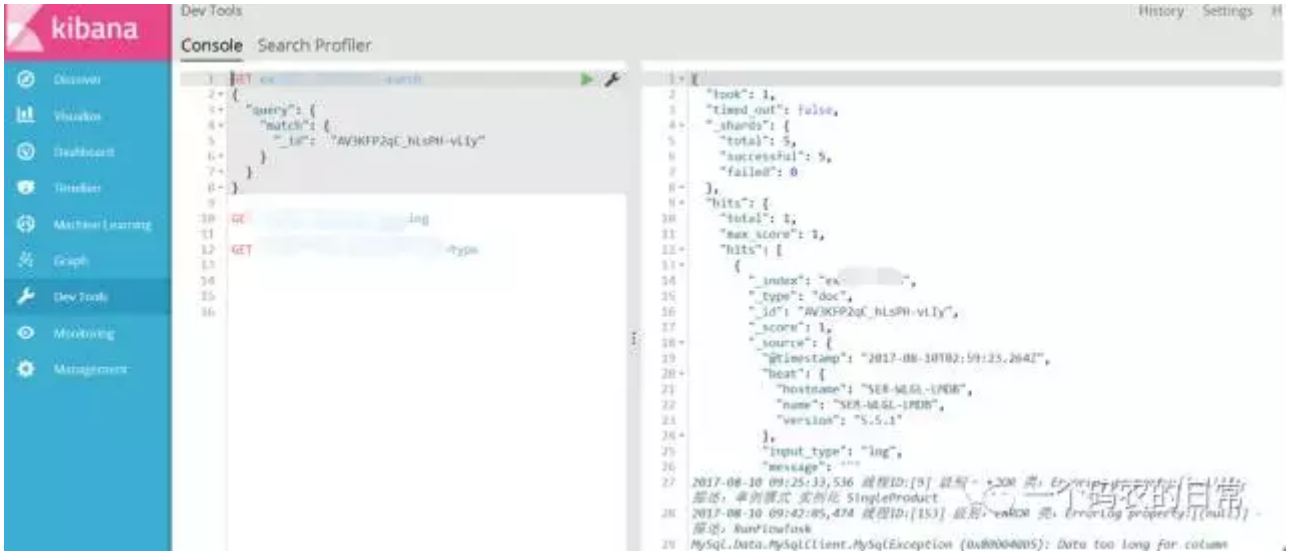
Kibana可以非常方便地把来自Logstash、ES-Hadoop、Beats或第三方技术的数据整合到Elasticsearch，支持的第三方技术包括Apache Flume、Fluentd等。

监控ES的整体健康状态

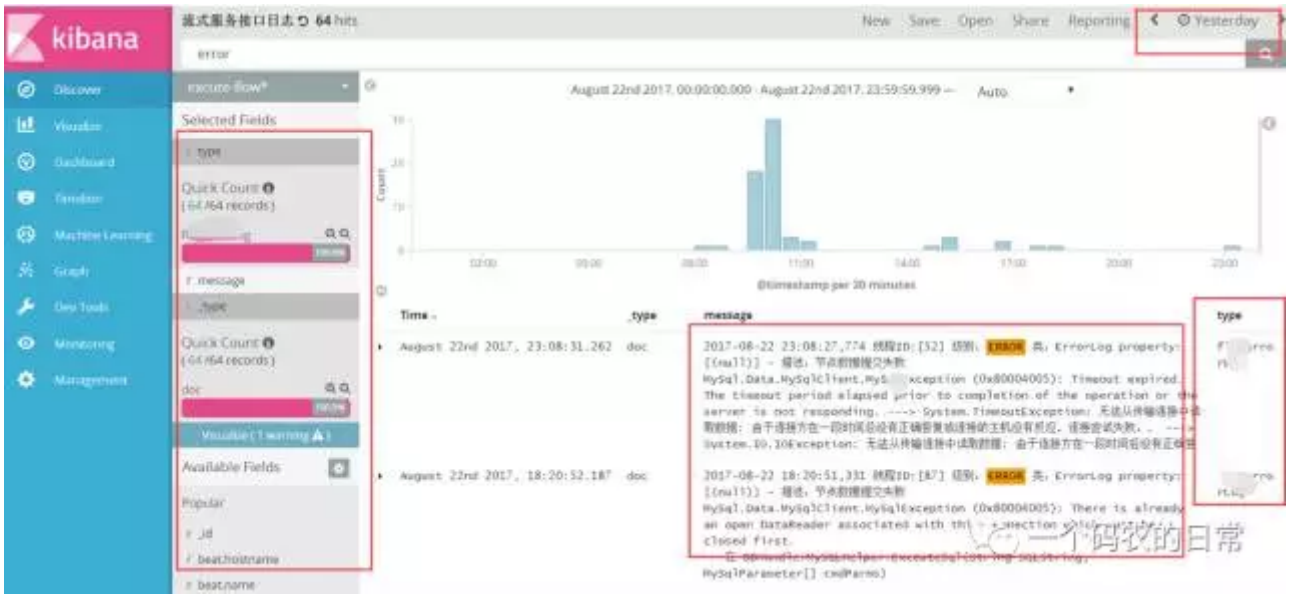




直接查询ES索引内容



简单的查询过滤日志数据窗口



可实时的图形统计展示

采用ElastAlert实现日志监控告警

平台缺失针对mysql连接数的告警，指定业务如流式服务数据异常，当异常触发时能够及时通过短信、邮件等方式通知相关负责人员

如故障信息：

以上说的“日志”不仅限于日志信息，也可以是业务数据。

软硬件服务监控平台设计

当业务层日志发现异常时如保存数据到Mysql时经常性报连接数据库超时，只有当业务人中发现再通知我们时已经过了一段时间才发现问题，但已无法重现当时的生产环境，也就靠经验来猜原因是服务器的网络问题还是数据库的真实连接满了还是程序的写法出现问题，因此就需要监控当时生产环境的软硬件监控数据。

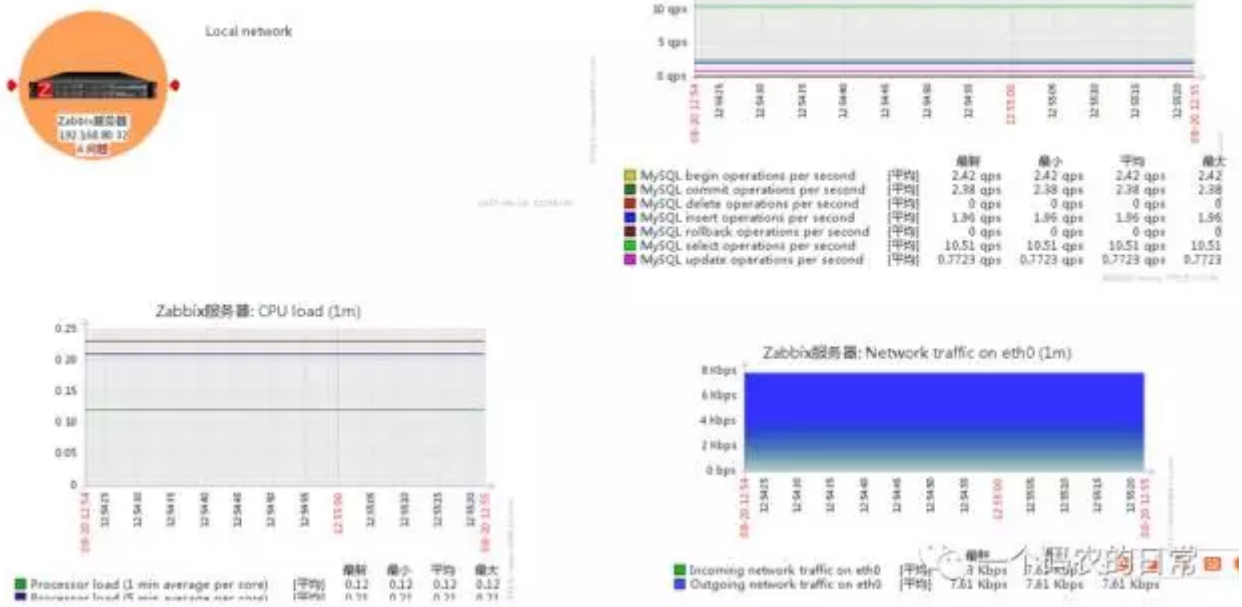
经过多方咨询参考各大厂的监控方案和对比在此采用Zabbix作监控。

最近各服务整体问题一览

针对Web服务器和API的访问性能、HAproxy、IIS、Tomcat

实时绘图监控服务器所有TCP端口的数量和 MySql数据库连接数、Redis性能

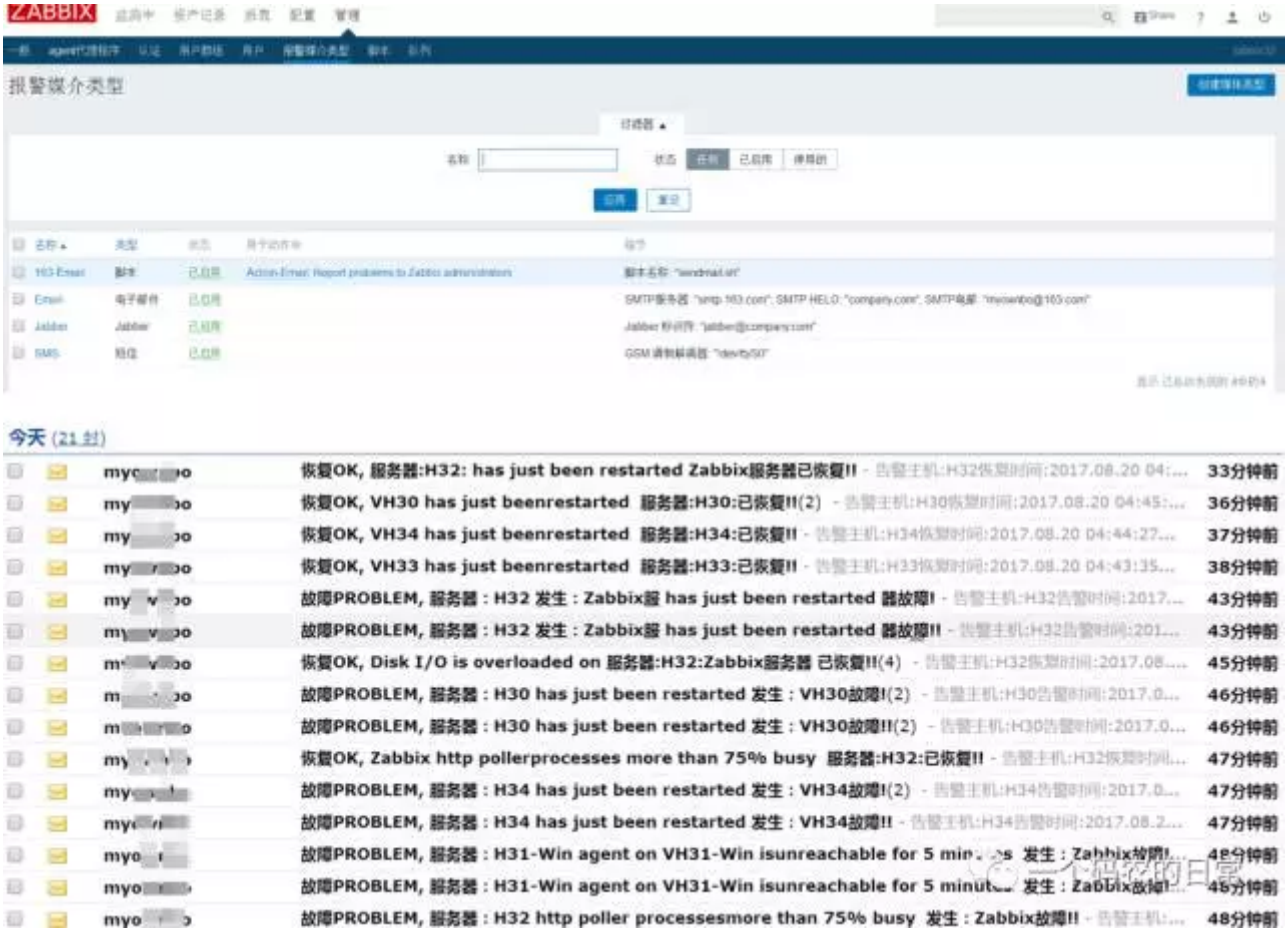
自定义聚合展示服务器各指表最近的状态,CPU、内存、流量。



显示所有服务器的一个健康状况，一目了然

自动注册监控新的服务器

报警机制，Email、微信、短信等



其它特性

可监控Linux、Windows、打印机、文件系统、网卡设备、 SNMP OID、数据库等平台服务状态。

允许灵活地自定义问题阀值， Zabbix 中称为触发器(trigger), 存储在后端数据库中。

高级告警配置，可以自定义告警升级(escalation)、接收者及告警方式。

数据存储在数据库中 历史数据可配置 内置数据清理机制。

web 前端采用 php 访问无障碍。
Zabbix API 提供程序级别的访问接口，第三方程序可以很快接入。

灵活的权限系统。

结合以上业务和软硬件上的日志方便开发和运维实时查找问题提高解决问题的效率，而且前期均可只通过配置0代码就可实现监控和报表展示。

扩展性

可用Spark对数据实时分析，智能拦截异常数据和直接发送异常警报。

在Zabbix上结合自己的业务需求二次开发应用系统层面上的预警监控系统。

以后可加入Kafka将日志集中，至于为什么选用kafka集群来构建日志中心，理由主要如下:

1、分布式架构，可支持水平扩展。

2、高吞吐量，在普通的服务器上每秒钟也能处理几十万条消息(远高于我们的峰值1.5万条/秒)。

3、消息持久化，按topic分区存储，支持可重复消费。

4、可根据broker配置定期删除过期数据。

长按识别二维码，关注一个码农的日常

阅读原文