

01 | 背景信息：监控需求以及开源方案的横评对比

2023-01-09 秦晓辉 来自北京



天下无鱼

<https://shikey.com/>

课程介绍 >

《运维监控系统实战笔记》



讲述：秦晓辉

时长 15:50 大小 14.47M



你好，我是秦晓辉。

今天我们就正式开始监控系统的学习之旅了，作为课程的第一讲，我想先让你了解一下监控相关的背景信息，对监控系统有一个整体性的了解。所以今天我们会先聊一聊监控的需求来源，也就是说监控系统都可以用来做什么，然后再跳出监控，从可观测性来看，监控与日志、链路之间的关系以及它们各自的作用。最后我们会介绍开源社区几个有代表性的方案以及它们各自的优缺点，便于你之后做技术选型。

掌握这些背景信息，是我们学习监控系统的基础。下面我们就先来了解一下监控的需求来源。

监控需求来源

最初始的需求，其实就是一句话：**系统出了问题我们能及时感知到**。当然，随着时代的发展，我们对监控系统提出了更多的诉求，比如：

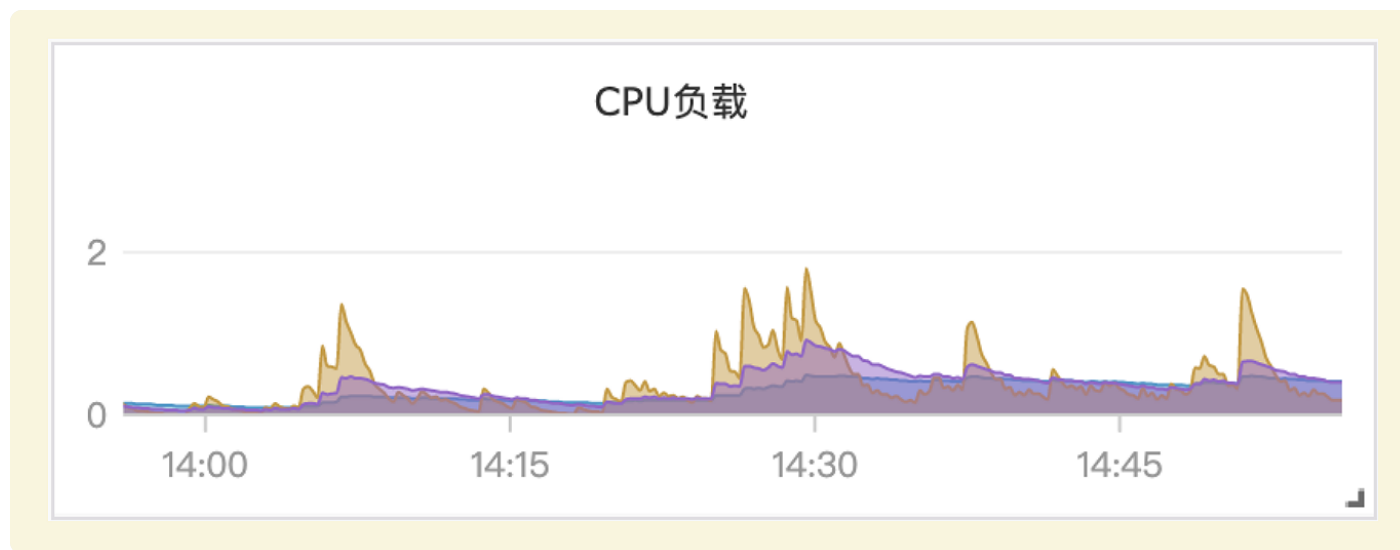


- 通过监控了解数据趋势，知道系统在未来的某个时刻可能出问题，预知问题。
- 通过监控了解系统的水位情况，为服务扩缩容提供数据支撑。
- 通过监控来给系统把脉，感知到哪里需要优化，比如一些中间件参数的调优。
- 通过监控来洞察业务，提供业务决策的数据依据，及时感知业务异常。

目前监控系统越来越重要，同时也越来越完备。不但能够很好地解决上面这几点诉求，还沉淀出了很多监控系统中的稳定性相关的知识。当然，这得益于对监控体系的持续运营，特别是一些资深工程师的持续运营的成果。

可观测性三大支柱

我们所说的监控系统，其实只是指标监控，通常使用折线图形态呈现在图表上，比如某个机器的 CPU 利用率、某个数据库实例的流量或者网站的在线人数，都可以体现为随着时间而变化的趋势图。



趋势图示例

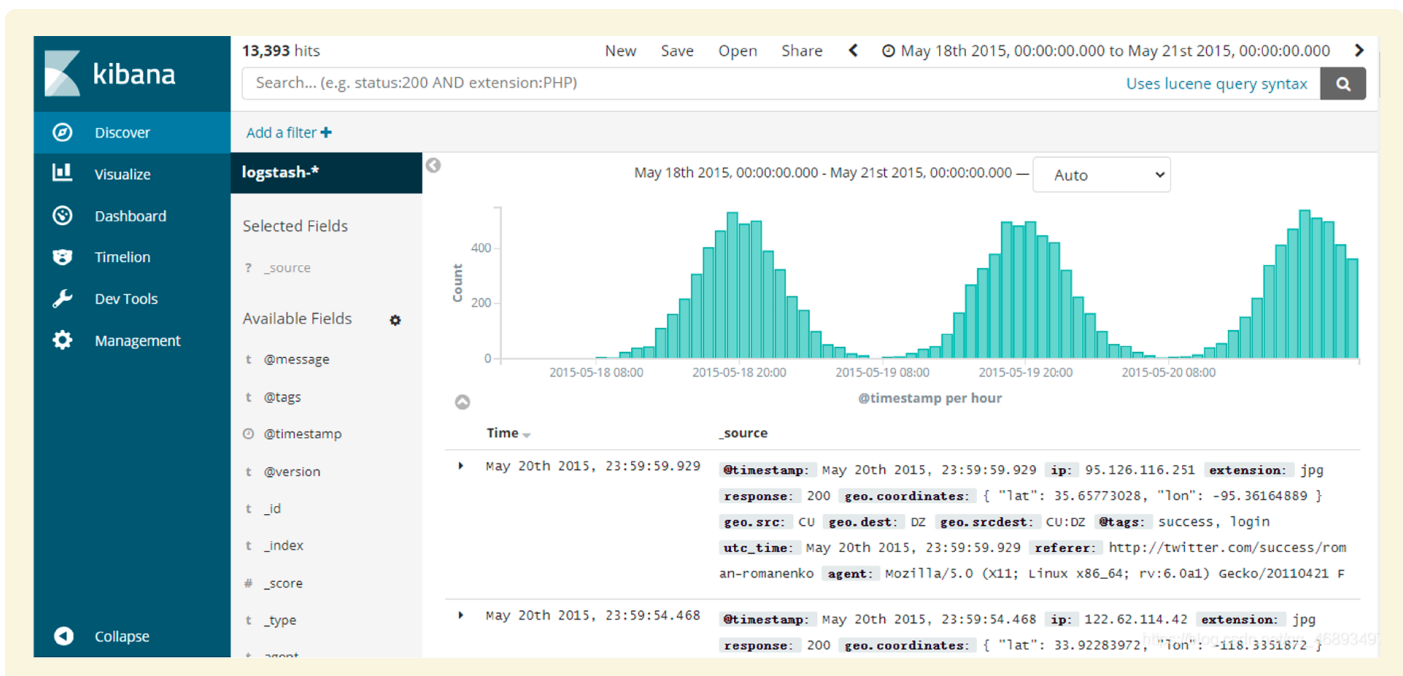
指标监控只能处理数字，但它的历史数据存储成本较低，实时性好，生态庞大，是可观测性领域里最重要的一根支柱。聚焦在指标监控领域的开源产品有 Zabbix、Open-Falcon、Prometheus、Nightingale 等。

除了指标监控，另一个重要的可观测性支柱是**日志**。从日志中可以得到很多信息，对于了解软件的运行情况、业务的运营情况都很关键。比如操作系统的日志、接入层的日志、服务运行日志，都是重要的数据源。



从操作系统的日志中，可以得知很多系统级事件的发生；从接入层的日志中，可以得知有哪些域名、IP、URL 收到了访问，是否成功以及延迟情况等；从服务日志中可以查到 **Exception** 的信息，调用堆栈等，对于排查问题来说非常关键。但是日志数据通常量比较大，不够结构化，存储成本较高。

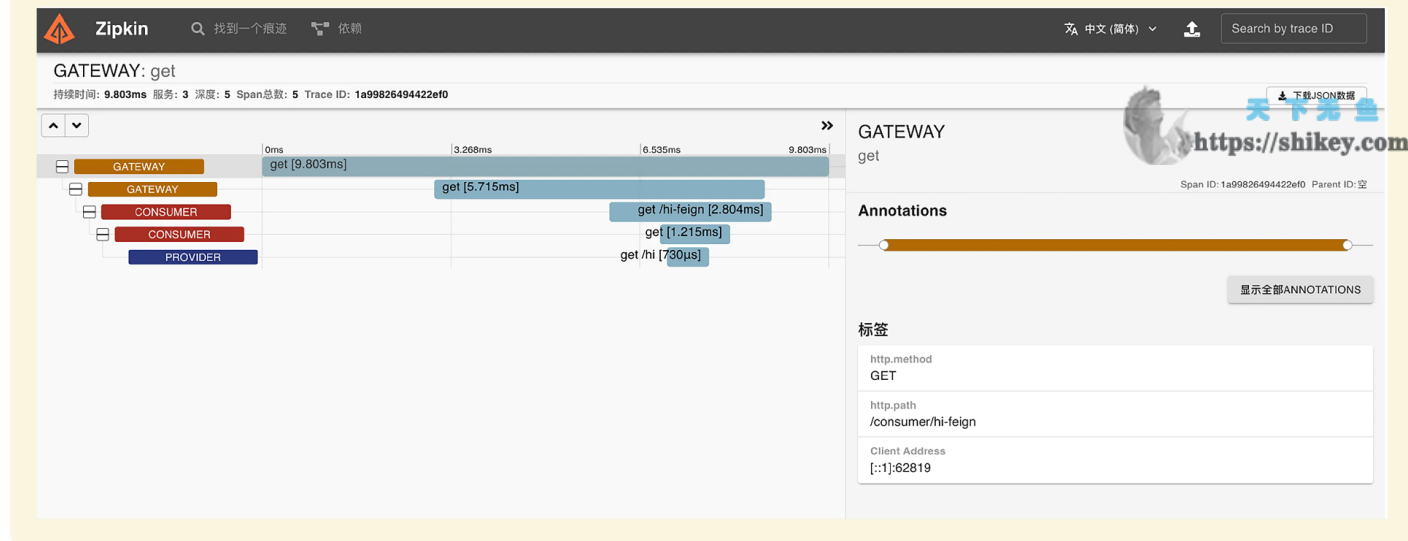
处理日志这个场景，也有很多专门的系统，比如开源产品 **ELK** 和 **Loki**，商业产品 **Splunk** 和 **Datadog**，下面是在 **Kibana** 中查询日志的一个页面。



图片来自官网

可观测性最后一大支柱是**链路追踪**。随着微服务的普及，原本的单体应用被拆分成很多个小的服务，服务之间有错综复杂的调用关系，一个问题具体是哪个模块导致的，排查起来其实非常困难。

链路追踪的思路是以请求串联上下游模块，为每个请求生成一个随机字符串作为请求 ID。服务之间互相调用的时候，把这个 ID 逐层往下传递，每层分别耗费了多长时间，是否正常处理，都可以收集起来附到这个请求 ID 上。后面追查问题时，拿着请求 ID 就可以把串联的所有信息提取出来。链路追踪这个领域也有很多产品，比如 **Skywalking**、**Jaeger**、**Zipkin** 等，都是个中翘楚。下面是 **Zipkin** 的一个页面。



图片来自官网

虽然我们把可观测性领域划分成了 3 大支柱，但实际上它们之间是有很强的关联关系的。比如我们经常会从日志中提取指标，转存到指标监控系统，或者从日志中提取链路信息来做分析，这在业界都有很多实践。

我们这个课程会聚焦在指标监控领域，把这个领域的相关知识讲透，希望可以帮助你在工作中快速落地实践。下面我们就来一起梳理一下业界常见的开源解决方案。

业界方案横评

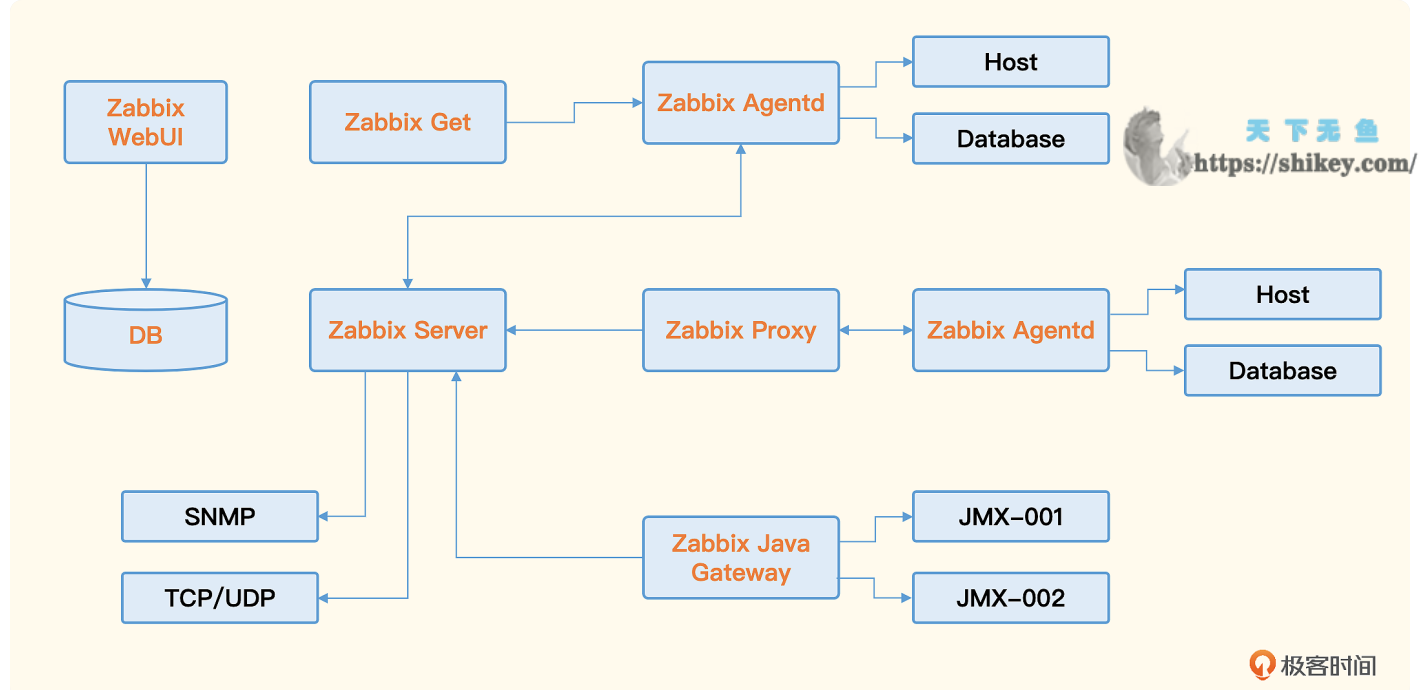
了解业界典型方案的一些优缺点，对选型有很大帮助。这里我们主要是评价开源方案，其实业内还有很多商业方案，特别是像 IBM Tivoli 这种产品，更是在几十年前就出现了，但是因为是商业产品，接触的人相对较少，这里就不点评了。

老一代整体方案的代表 Zabbix

Zabbix 是一个企业级的开源解决方案，擅长设备、网络、中间件的监控。因为前几年使用的监控系统主要就是用来监控设备和中间件的，所以 Zabbix 在国内应用非常广泛。

Zabbix 核心由两部分构成，Zabbix Server 与可选组件 Zabbix Agent。Zabbix Server 可以通过 SNMP、Zabbix Agent、JMX、IPMI 等多种方式采集数据，它可以运行在 Linux、Solaris、HP-UX、AIX、Free BSD、Open BSD、OS X 等平台上。

Zabbix 还有一些配套组件，Zabbix Proxy、Zabbix Java Gateway、Zabbix Get、Zabbix WEB 等，共同组成了 Zabbix 整体架构。



Zabbix 的优点

- 对各种设备的兼容性较好，Agentd 不但可以在 Windows、Linux 上运行，也可以在 Aix 上运行。
- 架构简单，使用数据库做时序数据存储，易于维护，备份和转储都比较容易。
- 社区庞大，资料多。Zabbix 大概是 2012 年开源的，因为发展的时间比较久，在网上可以找到海量的资源。

Zabbix 的缺点

- 使用数据库做存储，无法水平扩展，容量有限。如果采集频率较高，比如 10 秒采集一次，上限大约可以监控 600 台设备，还需要把数据库部署在一个很高配的机器上，比如 SSD 或者 NVMe 的盘才可以。
- Zabbix 面向资产的管理逻辑，监控指标的数据结构较为固化，没有灵活的标签设计，面对云原生架构下动态多变的环境，显得力不从心。

老一代国产代表 Open-Falcon

Open-Falcon 出现在 Zabbix 之后，开发的初衷就是想要解决 Zabbix 的容量问题。Open-Falcon 最初来自小米，14 年开源，当时小米有 3 套 Zabbix，1 套业务性能监控系统 perfcounter。Open-Falcon 的初衷是想做一套大一统的方案，来解决这个乱局。你可以看一下 Open-Falcon 的架构图。

- 可以处理大规模监控场景，比 Zabbix 的容量要大得多，不仅可以处理设备、中间件层面的监控，也可以处理应用层面的监控，最终替换掉了小米内部的 perfcounter 和三套 Zabbix。
- 组件拆分得比较散，大都是用 Go 语言开发的，Web 部分是用 Python，易于做二次开发。



Open-Falcon 的缺点

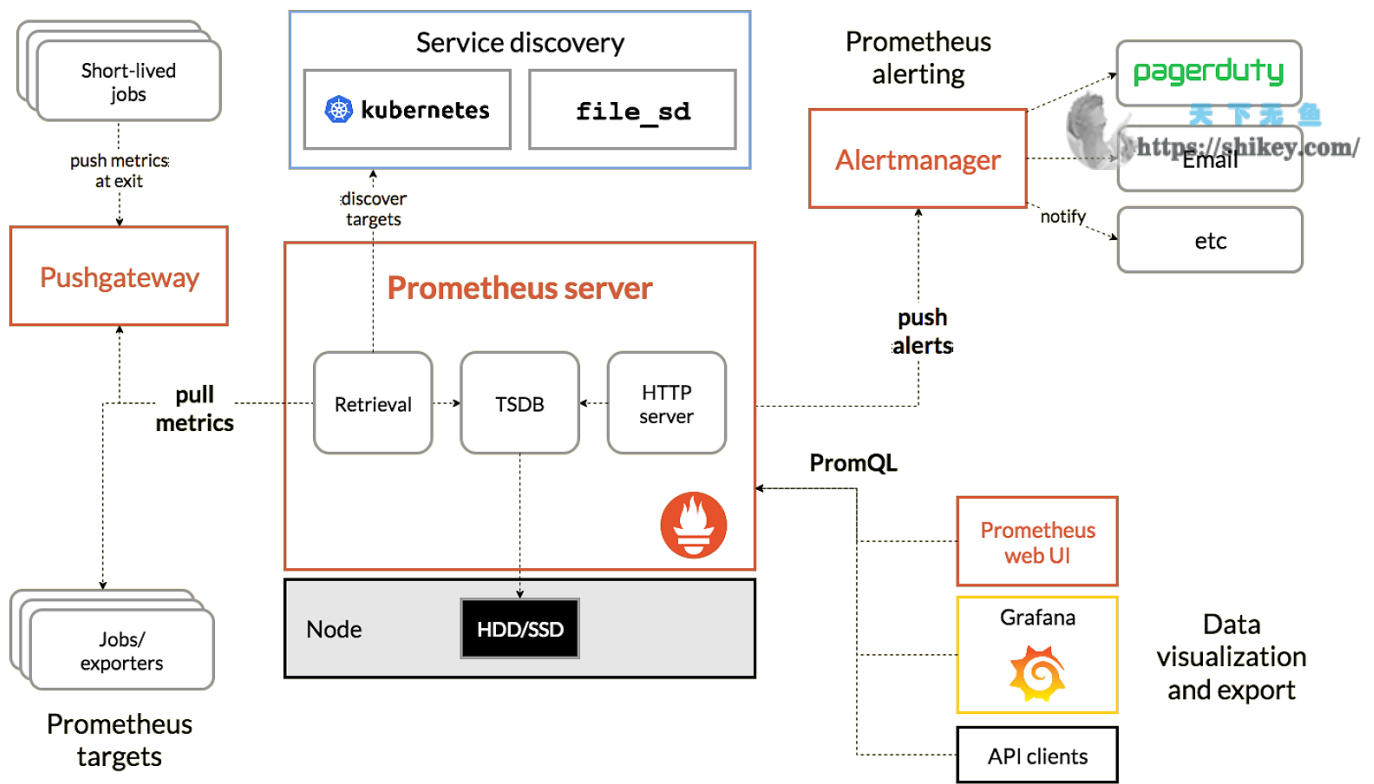
- 生态不够庞大，是小米公司在主导，很多公司做了二次开发，但是都没有回馈社区，有一些贡献者，但数量相对较少。
- 开源软件的治理架构不够优秀，小米公司的核心开发人员离职，项目就停滞不前了，小米公司后续也没有大的治理投入，相比托管在基金会的项目，缺少了生命力。

新一代整体方案代表 Prometheus

Prometheus 的设计思路来自 Google 的 Borgmon，师出名门，就像 Borgmon 是为 Borg 而生的，而 Prometheus 就是为 Kubernetes 而生的。它针对 Kubernetes 做了直接的支持，提供了多种服务发现机制，大幅简化了 Kubernetes 的监控。

在 Kubernetes 环境下，Pod 创建和销毁非常频繁，监控指标生命周期大幅缩短，这导致类似 Zabbix 这种面向资产的监控系统力不从心，而且云原生环境下大都是微服务设计，服务数量变多，指标量也呈爆炸态势，这就对时序数据存储提出了非常高的要求。

Prometheus 1.0 的版本设计较差，但从 2.0 开始，它重新设计了时序库，性能、可靠性都有大幅提升，另外社区涌现了越来越多的 Exporter 采集器，非常繁荣。你可以看一下 Prometheus 的架构图。



图片来自官网

Prometheus 的优点

- 对 Kubernetes 支持得很好，目前来看，Prometheus 就是 Kubernetes 监控的标配。
- 生态庞大，有各种各样的 Exporter，支持各种各样的时序库作为后端的 Backend 存储，也有很好的支持多种不同语言的 SDK，供业务代码嵌入埋点。

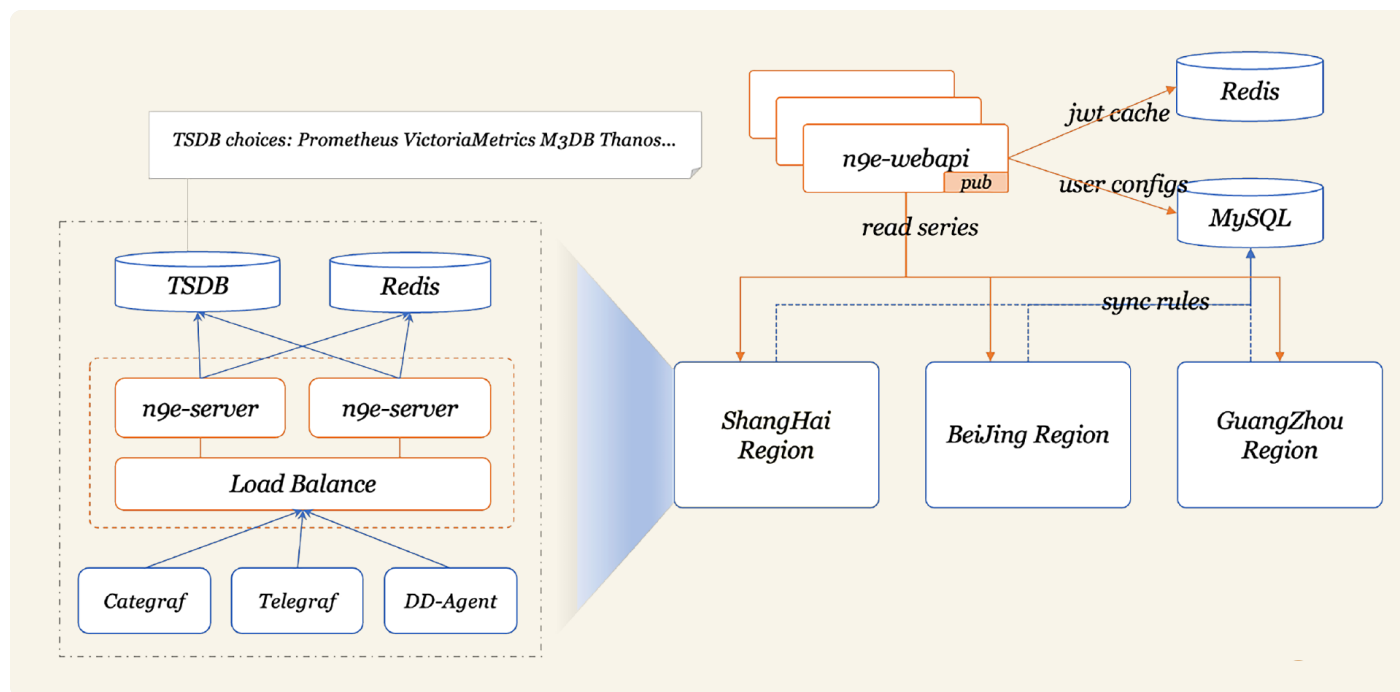
Prometheus 的缺点

- 易用性差一些，比如告警策略需要修改配置文件，协同起来比较麻烦。当然了，对于 IaC 落地较好的公司，反而认为这样更好，不过在国内当下的环境来看，还无法走得这么靠前，大家还是更喜欢用 Web 界面来查看监控数据、管理告警规则。
- Exporter 参差不齐，通常是一个监控目标一个 Exporter，管理起来成本比较高。
- 容量问题，Prometheus 默认只提供单机时序库，集群方案需要依赖其他的时序库。

新一代国产代表 Nightingale

Nightingale 可以看做是 Open-Falcon 的一个延续，因为开发人员是一拨人，不过两个软件的定位截然不同，Open-Falcon 类似 Zabbix，更多的是面向机器设备，而 Nightingale 不止解决设备和中间件的监控，也希望一并解决云原生环境下的监控问题。

但是在 Kubernetes 环境下，Prometheus 已经大行其道，再重复造轮子意义不大，所以 Nightingale 的做法是和 Prometheus 做良好的整合，打造一个更完备的方案。当下的架构，主要是把 Prometheus 当成一个时序库，作为 Nightingale 的一个数据源。如果不使用 Prometheus 也没问题，比如使用 VictoriaMetrics 作为时序库，也是很多公司的选择。



图片来自网络

Nightingale 的优点

- 有比较完备的 UI，有权限控制，产品功能比较完备，可以作为公司级统一的监控产品让所有团队共同使用。Prometheus 一般是每个团队自己用自己的，比较方便。如果一个公司用同一套 Prometheus 系统来解决监控需求会比较麻烦，容易出现我们上面说的协同问题，而 Nightingale 在协同方面做得相对好一些。
- 兼容并包，设计上比较开放，支持对接 Categraf、Telegraf、Grafana-Agent、Datadog-Agent 等采集器，还有 Prometheus 生态的各种 Exporter，时序库支持对接 Prometheus、VictoriaMetrics、M3DB、Thanos 等。

Nightingale 的缺点

- 考虑到机房网络割裂问题，告警引擎单独拆出一个模块下沉部署到各个机房，但是很多中小公司无需这么复杂的架构，部署维护起来比较麻烦。
- 告警事件发送缺少聚合降噪收敛逻辑，官方的解释是未来会单独做一个事件中心的产品，支持 Nightingale、Zabbix、Prometheus 等多种数据源的告警事件，但目前还没有放出。

上面我介绍了 4 种典型方案，每种方案各有优缺点，如果你的主要需求是监控设备，推荐你使用 Zabbix；如果你的主要需求是监控 Kubernetes，可以选择 Prometheus+Grafana；如果你既要兼顾传统设备、中间件监控场景，又要兼顾 Kubernetes，做成公司级方案，推荐你使用 Nightingale。

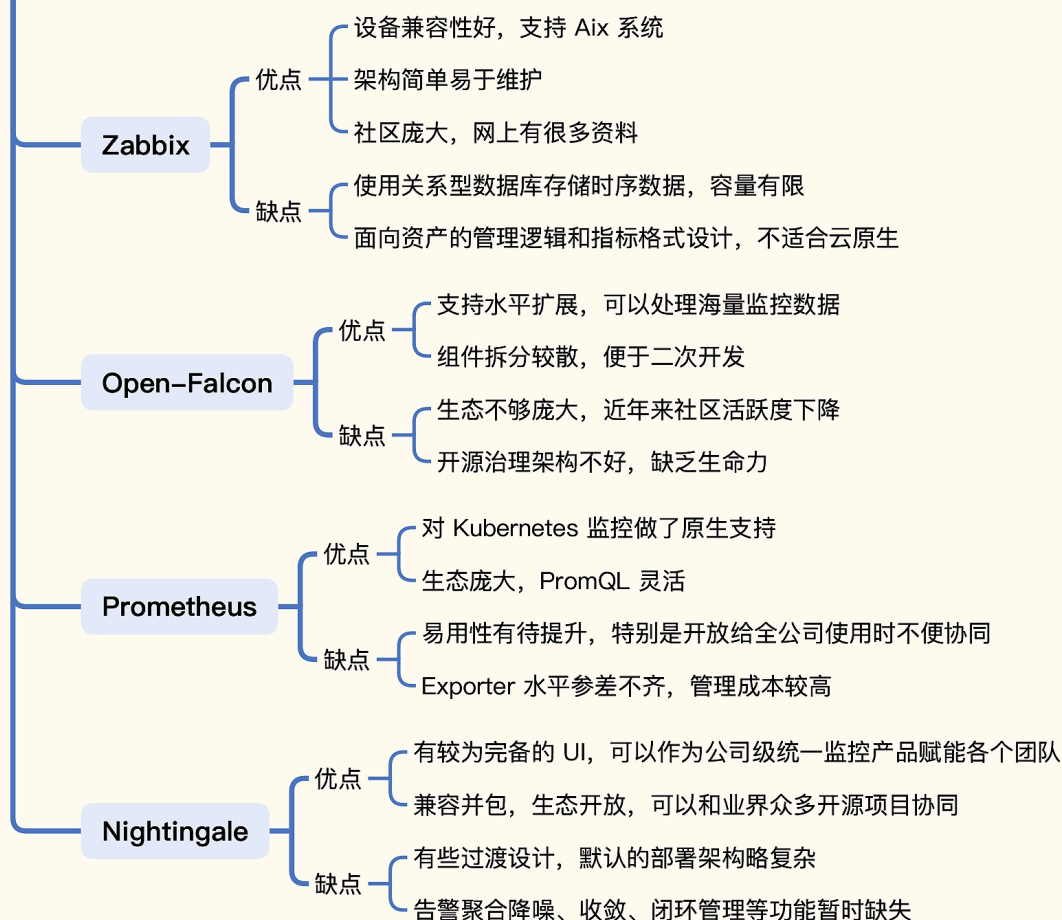
小结

最后，我们来回顾一下这一讲的主要内容。

这一讲我们了解了监控产品的需求来源，即监控问题域，从最开始的一句话需求——及时感知系统出现的问题，到现在希望预知问题，并且可以洞察业务经营数据，越来越多的诉求让我们意识到监控的重要性。

指标监控是可观测性三大支柱产品之一，除了指标监控之外，还有日志监控和链路追踪。这三者并不是独立的，它们之间联系紧密，共同辅助我们衡量系统内外部的健康状况。其中指标监控因历史数据存储成本较低，实时性好，生态庞大，是可观测性领域里最重要的一根支柱，也是我们关注的重点。

最后我们对指标监控领域的多个开源解决方案做了横评对比，帮助你做技术方案的选型。针对指标监控的几个开源方案的优缺点比较，我做了一个脑图，帮助你对比记忆。



互动时刻

指标监控领域还有很多其他的解决方案，你还知道哪些其他产品？欢迎留言分享，你可以简单说一下产品名字、适用场景、优缺点，三个臭皮匠顶个诸葛亮，我们一起讨论，互相帮助。也欢迎你把今天的内容分享给你身边的朋友，邀他一起学习。我们下一讲再见！

点击加入  课程交流群

分享给需要的人，Ta购买本课程，你将得 18 元

 生成海报并分享



上一篇 开篇词 | 每个关注高可用的人，都应该了解监控知识

下一篇 02 | 基本概念：监控圈子有哪些行业黑话？

精选留言 (25)

💬 写留言



顶级心理学家

2023-01-09 来自上海

秦总，IaC 落地 概念不是很清楚，想深入了解下，感谢

作者回复: IaC其实是 Infrastructure as Code 的缩写，可以Google一下这个关键词，或者看看这篇文章：<https://www.redhat.com/zh/topics/automation/what-is-infrastructure-as-code-iac> 另外 HashiCorp 搞了一个开源工具叫 Terraform 来践行 IaC，非常火爆，可以了解一下 Terraform 的基本工作机理，对 IaC 的了解也有帮助。举个例子，比如我要在公有云部署一个服务，需要一个mysql一个redis，一个LB，之前的做法是手工创建这些资源，应用了 IaC 之后（比如使用Terraform），就可以使用一个配置模板，和云厂商的OpenAPI联动，每次要创建这么一套环境的时候，就应用一下这个配置模板，Terraform就自动帮你创建、配置相关的资源。比如你测试完了之后可以销毁这些云资源，后面再想搭建这个环境的时候再应用一下这个配置模板，过一会这套软件又被拉起，非常方便。更多信息还是需要Google IaC这个关键词了解哈

共 3 条评论 >

👍 15



StackOverflow

2023-01-09 来自上海

监控不同指标要配置一堆exporter维护起来也很麻烦

作者回复: 嗯，exporter做采集器确实有这个问题，可以试试telegraf catagraf grafana-agent datadog-agent这些all-in-one的采集器，一个采集器就可以采集各类机器、中间件的监控指标

共 2 条评论 >

👍 12



怀朔

2023-01-09 来自上海

全球的化节点部署或者多机房的机房部署。运维维护往往其实还是多套数据，同一个展示 或者多个数据 多地方展示 因为要考虑的权限 容量 告警聚合收敛等问题

作者回复: 这是行家里手

共 2 条评论 >

👍 6



天下无鱼

<https://shikey.com/>



无聊的上帝

2023-01-11 来自上海

老师你好,在工作中遇到了日志监控和链路追踪很难落地的问题.

被挑战的点如下,请教老师这种局可有破解方法?

1. ELK成本较高,价值性较低.出现问题研发直接看pod的log.代码质量确实高,线上环境从未遇见严重bug.
2. 链路追踪的价值是什么,能给业务带来哪些提升?

作者回复: 咱们这个专栏主要还是聊监控和稳定性的话题。从稳定性角度出发的话,落地ELK、链路追踪的系统,核心还是想解决故障定位、可观测性的问题,如果在这方面没有痛点,那确实没有落地的必要,去找点其他更能体现价值的事情做一下。

如果还是想在这方面找出一些价值点,可以问这么几个问题:

- 1、Pod销毁比较频繁,如果有个异常日志还没来得及看的时候Pod被销毁了,是否是个问题
- 2、如果把这些可观测性数据都收集到中心,可以在中心做一些串联打通,比如指标掉底了,可以方便的跳转到日志系统里看日志,在terminal里查看日志显然做不到这个效果,这个收益是否足够有吸引力
- 3、链路追踪通常用在微服务场景,服务越多,效果越明显,如果微服务不多,出了问题我们可以快速知道是哪个模块,确实很难讲清楚价值

临时想到这些,欢迎其他同学补充~

共 2 条评论 >

👍 5



LiangDu

2023-01-09 来自上海

希望老师提供完善的告警规则和grafana仪表盘文件,对很多小白来说这两块才是核心。

作者回复: 课程主要还是想讲出所以然,不过实战部分可能会有一些帮助

共 5 条评论 >

👍 5




陈陈陈陈陈

2023-01-10 来自上海

目前的困境是告警泛滥,希望能减少不必要的告警指标,但又会顾虑正式这些指标的缺失导致

问题的发生

作者回复: 需要告警合并, 告警收敛, 告警分级治理的一些手段, 后面会有两讲介绍告警管理, 希望能给你提供一些思路  <https://shikey.com/>



👍 4



Gregory

2023-01-09 来自上海

多套监控系统维护确实是个问题 目前还没太好的方案

作者回复: 的确, 监控数据可视化、告警规则管理、告警事件管理, 这三块要是能有统一的一个产品来搞定就好了, 专栏中也提到了一些方案, 回头可以一起学习探讨



👍 3



第一装甲集群司令克莱...

2023-01-10 来自上海

想当年, 自己也参与过公司核心业务监控系统watchdog的开发。

作者回复: 同道中人



👍 1



奥特曼不会写代码

2023-01-10 来自上海

老师你好, 想请教一下对于网络连通性受限的场景下除了 Pushgateway 还有更好的方案不, 因为 Pushgateway 使用下来的体验确实不尽如人意, 公有云厂商的云主机也是通过类似于Pushgateway 的机制对外推送指标吗?

作者回复: 可以考虑remote write的方式哈, 比如机器上部署catergraf 或者 grafana-agent 或者 telegraf, 采集了数据之后通过remote write 推给远端时序库比较方便

共 2 条评论 >

👍 2



LEON

2023-01-09 来自上海

老师, 我是纯纯小白Exporter 是什么意思?

作者回复: Exporter是Prometheus生态的监控数据采集器的统称。比如机器层面的监控有node-exporter, MySQL的监控数据采集有mysqld_exporter

**Geek6570**

2023-01-09 来自上海

希望可以学习完，有方向或能力搭建混合云下的云原生环境监控，加油

作者回复: 加油，有志者事竟成～



1

**三年二班邱小东**

2023-01-13 来自吉林

老师好，我们公司主要就是监控网络设备和服务器，但却用Prometheus，他能胜任吗？感觉Grafana界面扩展性好差，对外行很不友好，我也不能决定技术选型，Prometheus+Grafana监控硬件可以做的zabbix一样方便吗？

作者回复: 从技术角度来看，是可以的。只是从页面交互而言，Zabbix更加面向设备，Prometheus不会对设备做特殊对待，只是针对服务器和网络设备而言，Zabbix可能更符合直观使用习惯。

另外，Zabbix内置了很多模板，尤其是网络设备的模板，也能省事不少。

Prometheus+Grafana的话，采集、告警、看图都没问题，不过采集层面对网络设备而言估计要付出更多成本。

**takumi**

2023-01-11 来自上海

大佬你好，比较好奇exporter和agnet有什么区别的区别嘛？看了一些文章里面涉及到Push和Pull两种模型，两者好像是模型上的区别？在一个可能agent能做的事情更多？期待大佬解答

作者回复: 后面的章节会讲到哈:)

**不经意间**

2023-01-11 来自北京

目前我这边也存在传统的服务器监控和基于k8s的云原生服务的监控。
看看zabbix也支持了k8s的pod自动发现，有点犹豫要不要继续支持zabbix(其实zabbix最核心的还是自动发现对于一些指标的采集)，我目前还用exporter在zabbix上利用promql做指标的自

动发现和处理。其实其他的也行，就看值不值得这么折腾。还有就是zabbix是支持单个指标的ttl的，不过好像prom设计理念中就不需要数据的长时间存储？(2年即以上时间)



不过现在也用了prom，直接上搭配的thanos和minio。

作者回复: 其实有时未必一定是二选一的，我的观察是大部分公司都有多套监控系统，不同的场景使用不同的系统。

比如 Kubernetes，如果监控的非常到位，每个组件的监控都很完备，Pod 里的应用也都埋点了，这个指标数量是很大的，Zabbix如果是使用MySQL存储，K8s稍微有点量Zabbix肯定就扛不住了，另外一些比较高端的场景，比如原本PromQL里group_left group_right解决的问题，Zabbix显然也是搞不定的。所以Kubernetes的监控，真的想监控好，Zabbix我觉得是不合适的。

但是Zabbix就是很擅长设备的监控，比如想监控aix小机，目前开源方案基本只能是用zabbix。

共 2 条评论 >



Howe

2023-01-11 来自广东

一直在想能不能基于Prometheus、Loki、skywalking，集成一个统一的管理平台，能实现监控部署的界面化操作，还能节省exporter部署的麻烦

作者回复: 这是真需求，未来肯定有人做，我们也有类似的计划



jay

2023-01-11 来自上海

请问老师:如果要监控遍布全国31省份的几百个节点，几千台物理服务器，上万的虚拟机。未来还要运行容器。用那个方案比较合适？ 我考虑过zb，但是性能比较差；Prometheus做邦联似乎可行，但是维护的压力也不小

作者回复: 如果要搞容器，肯定是Prometheus生态的技术栈会好一些，核心还是时序库的问题，后面有一讲会介绍如何提升Prometheus的存储能力，快讲到了:)



hshopeful

2023-01-11 来自上海

关于指标监控，在 prometheus 还没有盛行的时候，百度内部的 argus 监控系统做得很牛逼

(数据准确性、时效性、丰富的功能)，后面开源的 **open falcon** 的架构跟百度监控系统的架构比较类似；阿里的鹰眼系统在链路追踪方面做得不错；腾讯内部各个 **BG** 基本上都有做监控系统，甚至不同部门，不同中心都单独造轮子，并没有形成合力，所以不是很好用。现在走在技术最前沿的可能是字节基于 **eBPF** 的可观测实践。



<https://shikey.com/>

作者回复: 是的，大厂因为场景大，走的比较靠前，Google的Borgmon这都是20多年前都有了可能~~



1



文康

2023-01-10 来自上海

能不能讲一下指标的正常值范围和极限值范围？比如mysql慢查询多久是慢查询？硬盘的iops多少是正常？极限值是多少？

作者回复: 生产环境指标量是非常非常大的，后面的章节会挑重点的做介绍的哈

共 3 条评论 >



1



peter

2023-01-10 来自上海

请教老师几个问题：

Q1: 生产环境中日志是开启的吗？

出了问题以后，通过日志来定位问题。但是，生产环境中一般不能开启日志吧。如果不开启的话，怎么利用日志来定位问题呢。好像是个矛盾的事情。

Q2: 大厂开发人员是怎么查看日志的？

对于日志，开发人员是直接**Editplus**一类的软件来打开看吗？还是说会用专门的工具软件来查看日志文件？如果用工具软件，用开源的软件还是公司自研的软件？

Q3: **open-falcon**架构图中怎么没有**server**？

Zabbix有**server**，**Open-falcon**是基于**Zabbix**发展起来的，按理说也应该有一个**server**，但架构图中看不出来哪个部分是**server**。

Q3: **Prometheus**两个问题

1 没有采用**k8s**的网站系统，可以用**Prometheus**吗？

2 **Prometheus**可以完成全面的监控吗？包括机器、网络、应用、各个中间件等。

Q4: 指标监控数据一般怎么存储的？存在**MySQL**中吗？

作者回复: 1，生产环境也会开启日志打印的

2，一般用**less**、**more**、**tail**等命令

- 3, open-falcon尝试解决zabbix的容量问题，但并不是基于zabbix的架构，并且服务端组件拆得比较散，transfer、hbs、judge、graph等都是服务端组件
- 4, 没用k8s也可以用prometheus
- 5, prometheus可以完成全面的指标监控
- 6, 一般存储在时序库中，下一讲就开始介绍常用时序库了，zabbix是存mysql，这种不多见，一般都是使用专门的时序库



天下无鱼

<https://shikey.com/>

共 2 条评论 >



薛梁Lucien

2023-01-10 来自上海

夜莺发展势头一直不错，之前我也一直在关注，希望通过这个课程能看到夜莺的架构设计，然后我也借鉴借鉴，学习学习。

作者回复:

