

什么是Prometheus

关于Prometheus

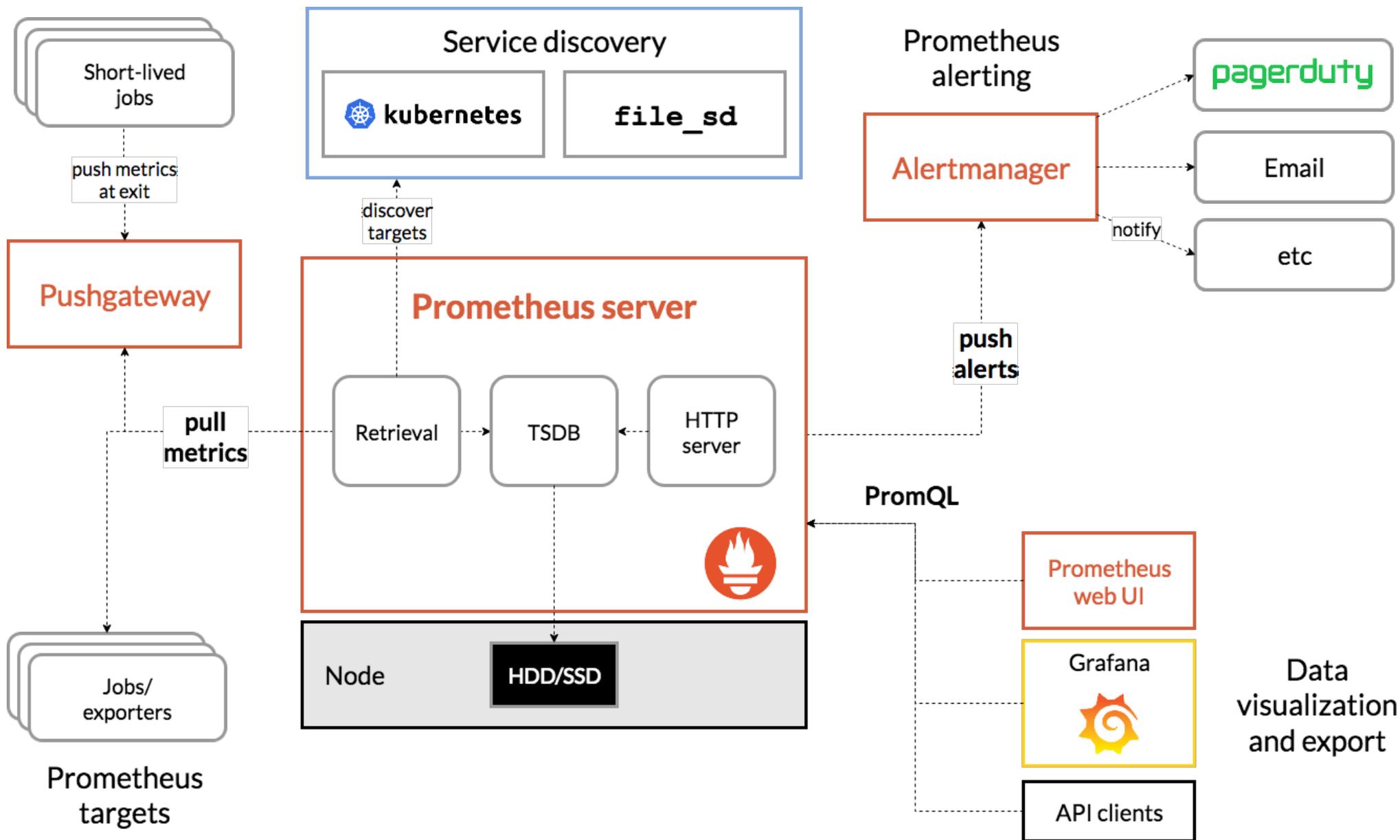
Prometheus 是一套开源的系统监控报警框架。它的设计灵感源于 Google 的 borgmon 监控系统,由SoundCloud 在 2012 年创建,后作为社区开源项目进行开发,并于 2015 年正式发布。2016 年,Prometheus 正式加入 Cloud Native Computing Foundation(CNCF),成为受欢迎度仅次于 Kubernetes 的项目,目前已广泛应用于Kubernetes集群监控系统中,大有成为Kubernetes集群监控标准方案的趋势。

Prometheus的优势

- 强大的多维度数据模型:
 - 时间序列数据通过 metric 名和键值对来区分。
 - 所有的 metrics 都可以设置任意的多维标签。
 - 数据模型更随意,不需要刻意设置为以点分隔的字符串。
 - 可以对数据模型进行聚合,切割和切片操作。
 - 支持双精度浮点类型,标签可以设为全 unicode。
- **灵活而强大的查询语句(PromQL):**在同一个查询语句,可以对多个 metrics 进行乘法、加法、连接、取分数位等操作。
- **易于管理:** Prometheus server 是一个单独的二进制文件,可直接在本地工作,不依赖于分布式存储。
- **高效:**平均每个采样点仅占 3.5 bytes,且一个 Prometheus server 可以处理数百万的 metrics。
- **动态获取:**可以通过服务发现或者静态配置去获取监控的 targets。
- 使用 pull 模式采集时间序列数据,可以避免有问题的服务器推送坏的 metrics。
- 支持 push gateway 的方式把时间序列数据推送至 Prometheus server 端。
- 多种可视化图形界面。

Prometheus架构及组件

图片源于Prometheus官方文档



上图为Prometheus的架构图,包含了Prometheus的核心模块及生态圈中的组件,简要介绍如下:

- **Prometheus Server:** 用于收集和存储时间序列数据。
- **Client Library:** 客户端库,为需要监控的服务生成相应的 metrics 并暴露给 Prometheus server。当 Prometheus server 来 pull 时,直接返回实时状态的 metrics。
- **Push Gateway:** 主要用于短期的 jobs。由于这类 jobs 存在时间较短,可能在 Prometheus 来 pull 之前就消失了。为此 jobs 可以直接向 Prometheus server 端推送它们的 metrics。这种方式主要用于服务层面的 metrics,对于机器层面的 metrics,建议使用 node exporter。
- **Exporters:** 用于暴露已有的第三方服务的 metrics 给 Prometheus。
- **Alertmanager:** 从 Prometheus server 端接收到 alerts 后,会去除重复数据,分组,并路由到对应的接受方式,发出报警。

工作原理

如上图可见,Prometheus 的主要模块包括:Prometheus server, exporters, Pushgateway, PromQL, Alertmanager 以及图形界面,其大概的工作流程是:

1. Prometheus server 定期从配置好的 jobs 或者 exporters 中拉 metrics,或者接收来自 Pushgateway 发过来的 metrics,或者从其他的 Prometheus server 中拉 metrics。
2. Prometheus server 在本地存储收集到的 metrics,并运行已定义好的 alert.rules,记录新的时间序列或者向 Alertmanager 推送警报。
3. Alertmanager 根据配置文件,对接收到的警报进行处理,发出告警。
4. 在图形界面中,可视化采集数据。

Prometheus 工作的核心,是使用 Pull (抓取)的方式去搜集被监控对象的 Metrics 数据(监控指标数据),然后,再把这些数据保存在一个 TSDB (时间序列数据库,比如 OpenTSDB、InfluxDB 等)当中,以便后续可以按照时间进行检索。

适用场景

Prometheus非常适合记录纯时间序列的数据。它既适用于面向服务器等硬件指标的监控,也适用于高动态的面向服务架构的监控。对于现在流行的微服务,Prometheus的多维度数据收集和数据筛选查询语言也是非常的强大。Prometheus是为服务的可靠性而设计的,当服务出现故障时,它可以使你快速定位和诊断问题。它的搭建过程对硬件和服务没有很强的依赖关系。

Prometheus重视可靠性,即使在故障情况下,您也可以随时查看有系统可用统计信息。如果您需要100%的准确度,例如按请求计费,Prometheus不是一个好的选择,因为收集的数据可能不够详细和完整。

总之,在需要高可用性的业务场景,Prometheus是一个非常好的选择,但对于高精度、高准确率的业务场景,Prometheus并非最佳选择。