

Prometheus监控

10/16/22

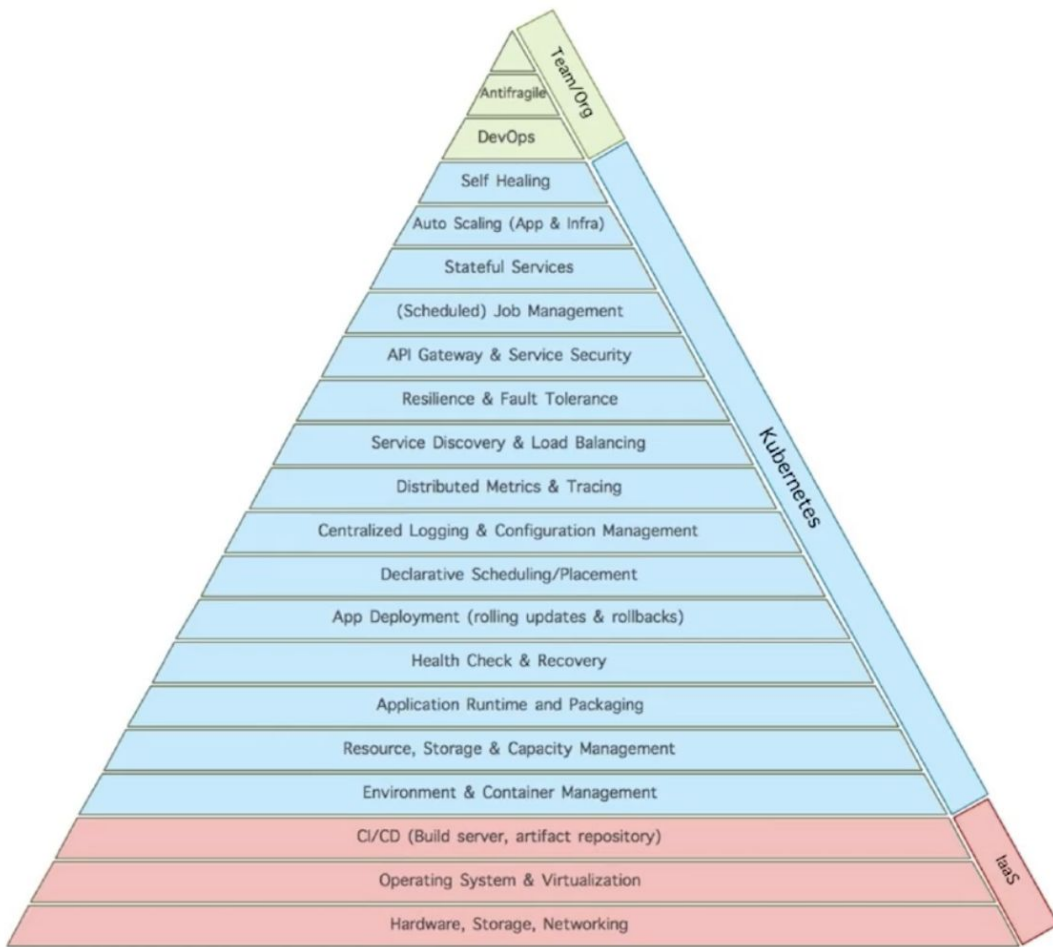
监控新背景

- 容器化, 微服务, 云原生的发展
- IT架构的发展: 物理机 -> IaaS云和容器云

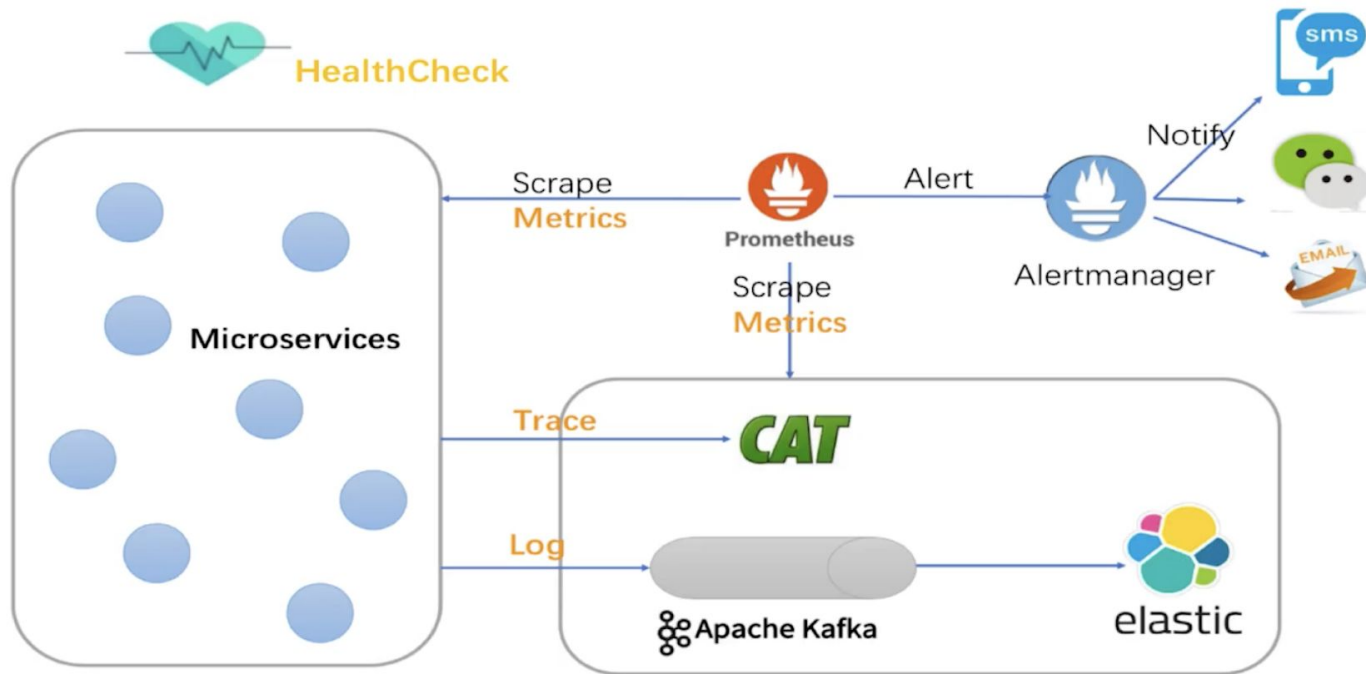
1. 监控的意义

- Long-term trend analysis
- Control analysis
- Alert
- Fault analysis and location
- Visualization Dashboard

- 端监控
- 业务层监控
- 应用层监控
- 中间件监控
- 系统层监控



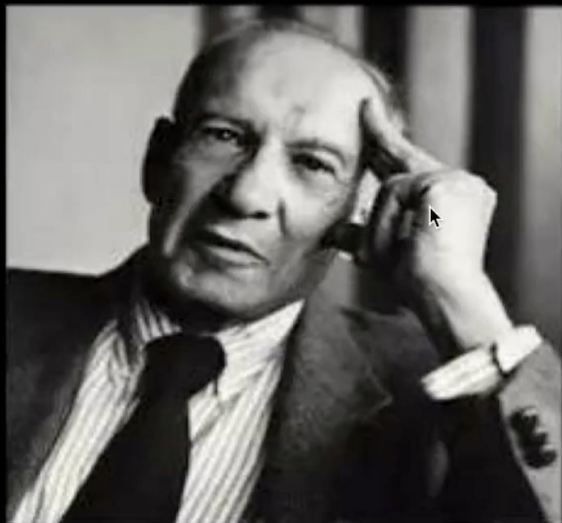
监控的分类



Prometheus的独孤九剑

1. 破剑式 - 监控微服务
2. 破刀式 - 监控OS(`node_exporter`)
3. 破枪式 - 监控DB (例如`mysqld_exporter`)
4. 破鞭式 - 监控Redis
5. 破索式 - 监控流处理例如Flink
6. 破掌式 - 监控Kafka
7. 破箭式 - 监控etcd(Prometheus Operator)
8. 破气式 - 监控K8S
9. 总决式 - 各种衍生的监控方案

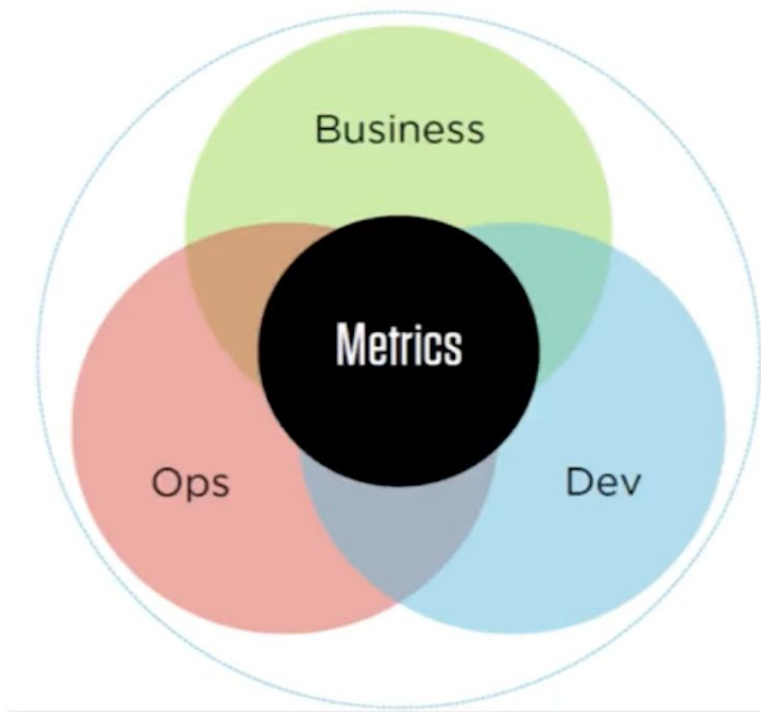
监控的哲学



**“If you can’t
measure it,
you can’t
manage it”**

Peter Drucker

2. Metric Design Development (MDD)



MDD三个步骤:

- 分配metrics给所有者, 例如业务, 应用, 基础架构等
- 创建分层metrics并关联趋势
- 制定决策时使用metrics

对于研发 - 实时感知**应用**的各项metrics, focus应用的优化;
对于运维 - 实时感知**系统**的各项metrics, focus到快速定位问题
对于PM - 实时感知**业务**的各项metrics, focus数据驱动决策

Prometheus历史

Google Borg -> Kubernetes

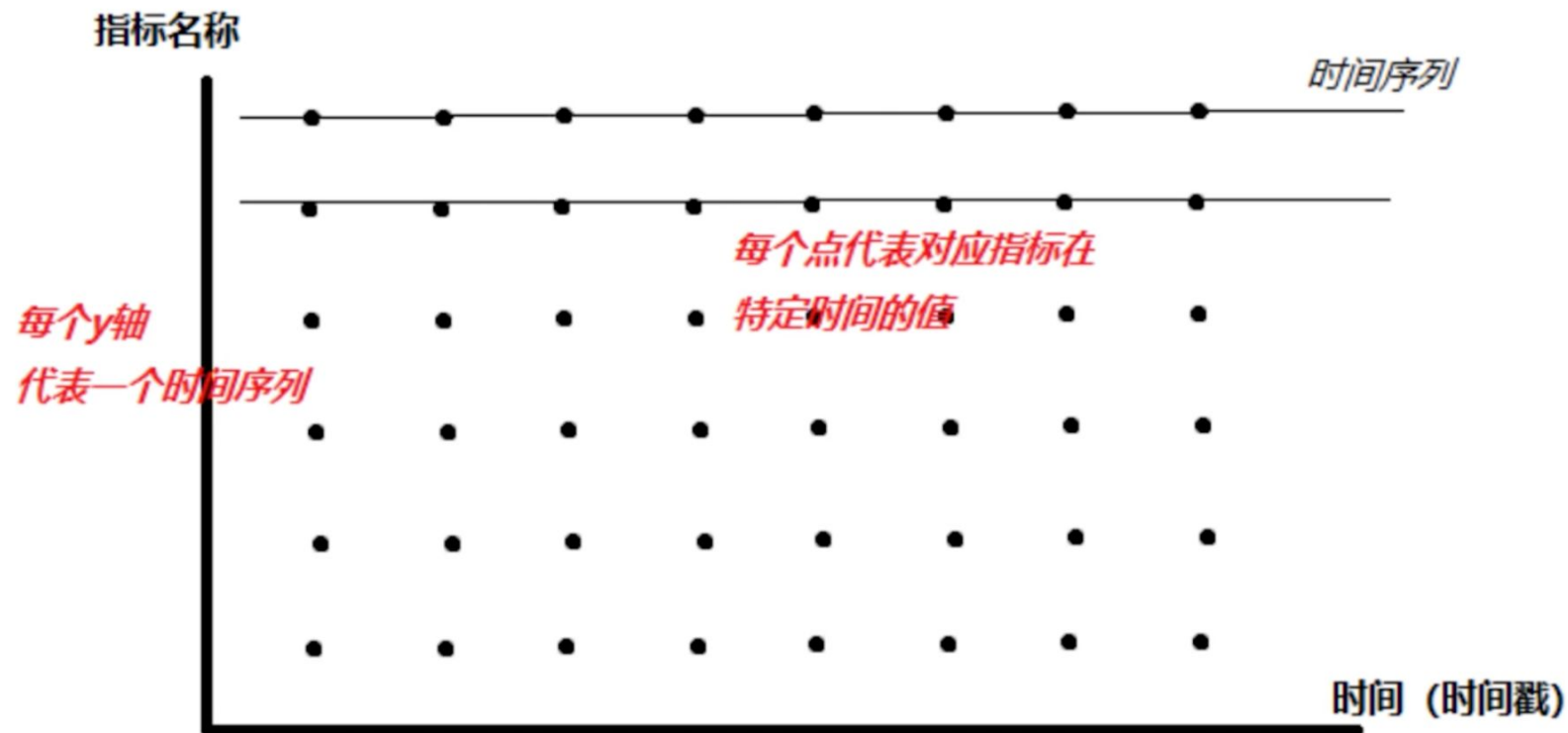
Google Borgmon -> Prometheus, 第二个[CNCF](#)成员项目

3. Prometheus Overview

官网总结:

1. Go语言实现 - cloud native
2. Pulling model为主
3. 多维度的数据tags - 多维度的Data Models和灵活的查询方式 (PromQL, http接口查询)
4. 支持Black Box和White Box测试
5. 单节点性能强劲: 支持上千的targets, 和每秒百万级别的time series

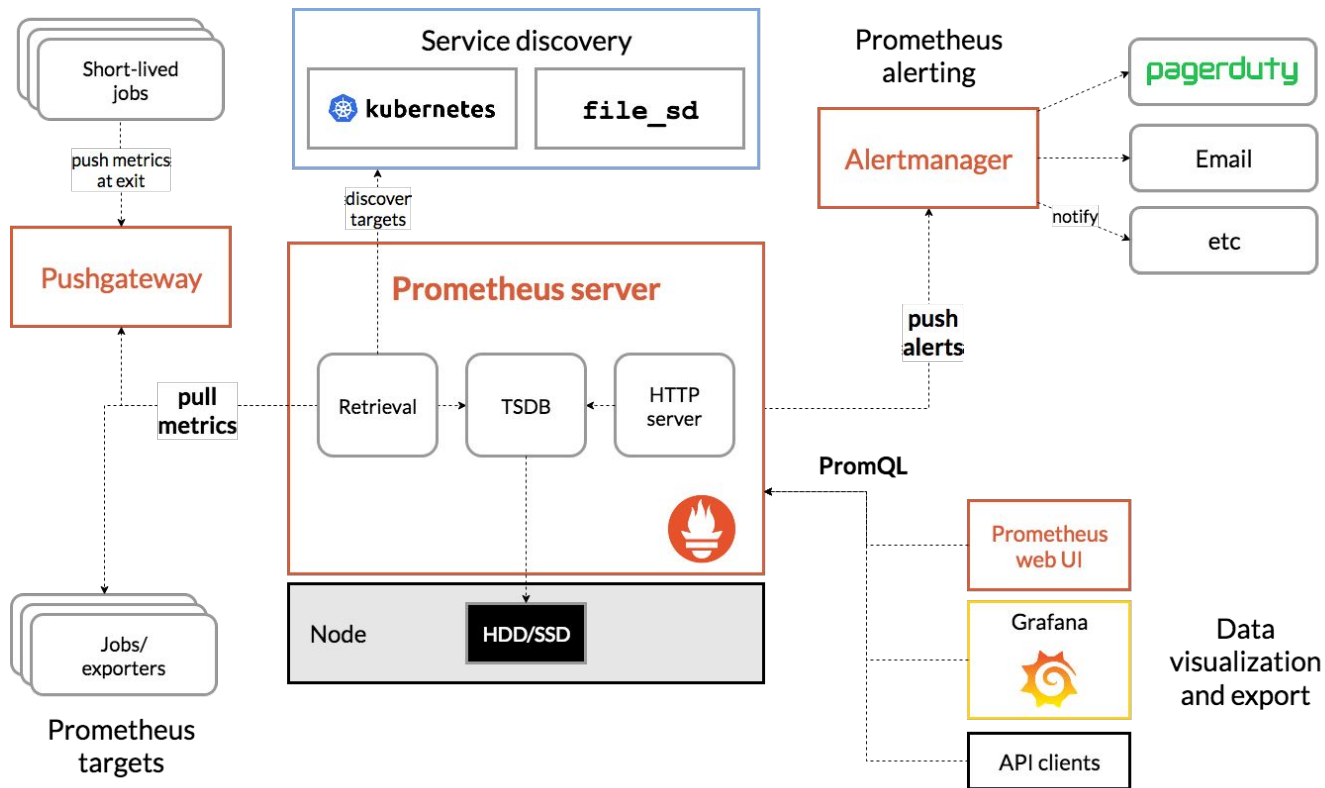
时间序列 (Time Series)



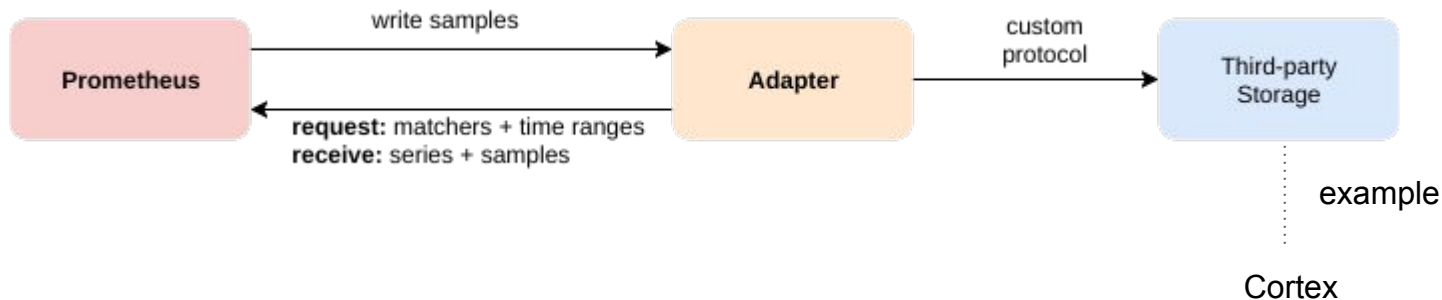
Prometheus Metrics

Metric name	Labels	Timestamp	Sample Value
...			
http_requests_total{status="200",method="GET"}		@1434317560938	94355
http_requests_total{status="200",method="GET"}		@1434317561287	94934
http_requests_total{status="200",method="GET"}		@1434317562344	96483
http_requests_total{status="404",method="GET"}		@1434317560938	38473
http_requests_total{status="404",method="GET"}		@1434317561249	38544
http_requests_total{status="404",method="GET"}		@1434317562588	38663
http_requests_total{status="200",method="POST"}		@1434317560885	4748
http_requests_total{status="200",method="POST"}		@1434317561483	4795
http_requests_total{status="200",method="POST"}		@1434317562589	4833
http_requests_total{status="404",method="POST"}		@1434317560939	122
...			

Prometheus Architect



Prometheus 存储设计: V0~V3



wal: https://en.wikipedia.org/wiki/Write-ahead_logging

TSDB/LevelDB: similar to Gorilla <https://www.vldb.org/pvldb/vol8/p1816-teller.pdf>

Delta Compression: <https://www.timescale.com/blog/time-series-compression-algorithms-explained/>

Prometheus流程总结

1. 通过Pull方式抓取数据
2. 写入head chunk
3. 写满1kb, 就生成新的chunk (完成的chunks是不可以变更的)
4. 根据配置文件, 一部分chunks保存在内存里, 根据LRU将chunks写入到磁盘

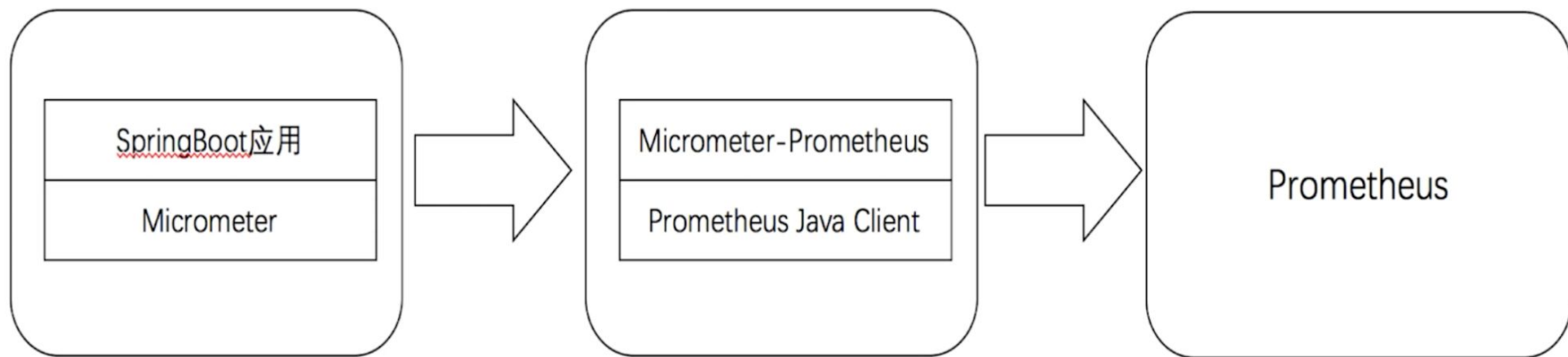
在目前Prometheus的世界, 无论是内存还是磁盘, by default都是以1kb的size进行操作的

3. Practice 1

Prometheus

Grafana

Prometheus监控告警Spring Boot应用 - [Micrometer](#)



Micrometer的分层访问

<http://localhost:8080/actuator>

<http://localhost:8080/actuator/metrics>

<http://localhost:8080/actuator/metrics/tomcat.sessions.created>

<http://localhost:8080/actuator/metrics/PrometheusApplication.demo.counter?tag=name:counter2>

Practice 2

Spring Boot 接入Grafana, 观察Spring Boot的相关metrics

<https://grafana.com/grafana/dashboards/4701-jvm-micrometer/>

<https://ordina-jworks.github.io/monitoring/2020/11/16/monitoring-spring-prometheus-grafana.html>

To be Continued

Prometheus 邮件/text/Slack等等告警

Prometheus 高级功能

References