

Prometheus中间件开发

联系QQ: 2816010068, 加入会员群

目录

- Prometheus介绍
- Grafana介绍和使用
- Prometheus中间件开发
- 测试Prometheus中间件

Prometheus介绍

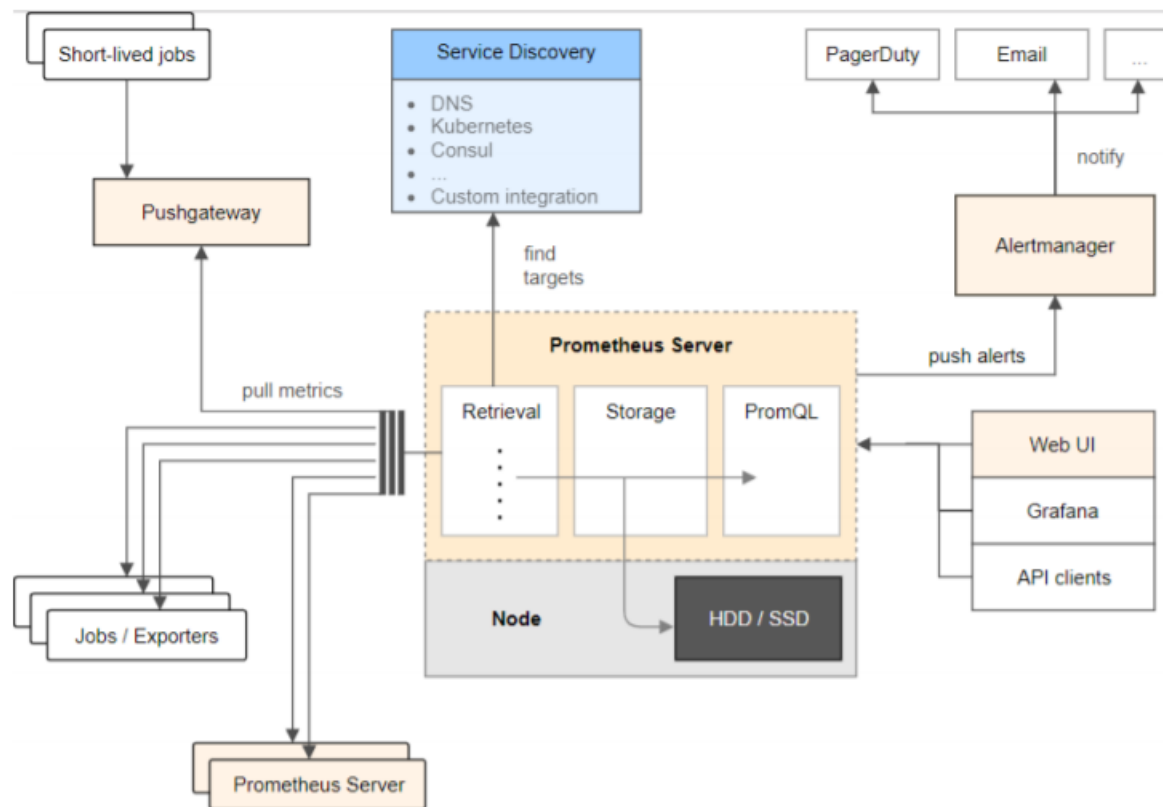
- 普罗米修斯：希腊神话中最具智慧的神明之一，创造了人类（神话），给人类带来了火，还教会了他们许多知识和技能。



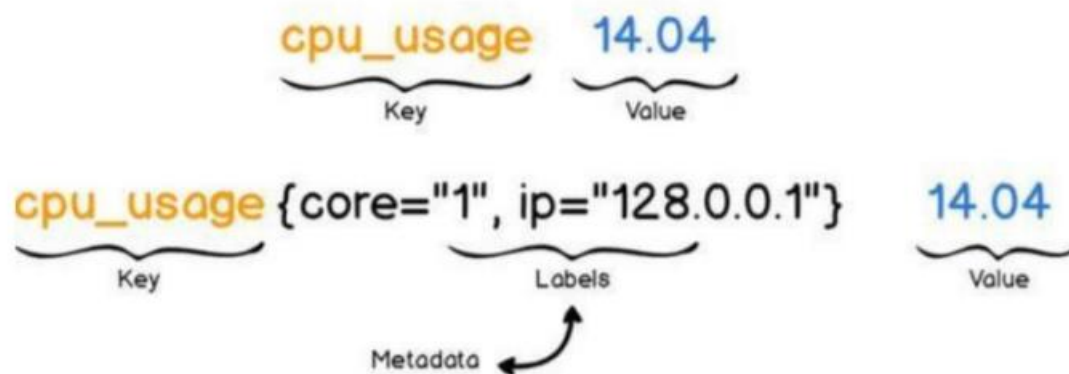
Prometheus介绍

- 分布式监控系统
- 完全开源，使用Go进行开发
- 被广泛用来监控整个云基础设施设施

Prometheus架构



Prometheus 数据模型



2 Data Model Filtering



度量类型

- 计数器Counter采样
 - 计数器可能是我们可以使用的最简单的度量标准形式。就想它字面意思一样，计数器是随着时间的增长的计算元素
 - 计数器的值只能增加或重置为0。计数器特别适合计算某个时段上某个事件的发生次数，即指标随时间演变的速率。
 - 比如，要计算服务器上的HTTP错误数或网站上的访问次数，这时候就使用计数器。

Gauges采样

- Gauges用于处理可能随时间减少的值。Gauge类型的值可以上升和下降，可以是正值或负值。
- 用来记录瞬时的值，比如内存变化、温度变化。连接池的连接数。

柱状图Histogram采样

- 对每个采样点进行统计（并不是一段时间的统计），打到各个桶(bucket)中
- 对每个采样点值累计和(sum)
- 对采样点的次数累计和(count)

buckets	Observe	Write	实际采样点			
10	1	6	8.07462357			
5	3	5	1.90329063	4.90508872	2.29902037	
1	2	2	0.80926036	0.84659086		
count	6					
sum	18.8378745					

Summary采样

- 在客户端对于一段时间内（默认是10分钟）的每个采样点进行统计，并形成分位图。

```
1 # HELP prometheus_tsdb_wal_fsync_duration_seconds Duration of WAL fsync.
2 # TYPE prometheus_tsdb_wal_fsync_duration_seconds summary
3 prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.5"} 0.012352463
4 prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.9"} 0.014458005
5 prometheus_tsdb_wal_fsync_duration_seconds{quantile="0.99"} 0.017316173
6 prometheus_tsdb_wal_fsync_duration_seconds_sum 2.888716127000002
7 prometheus_tsdb_wal_fsync_duration_seconds_count 216
```

Prometheus 安装

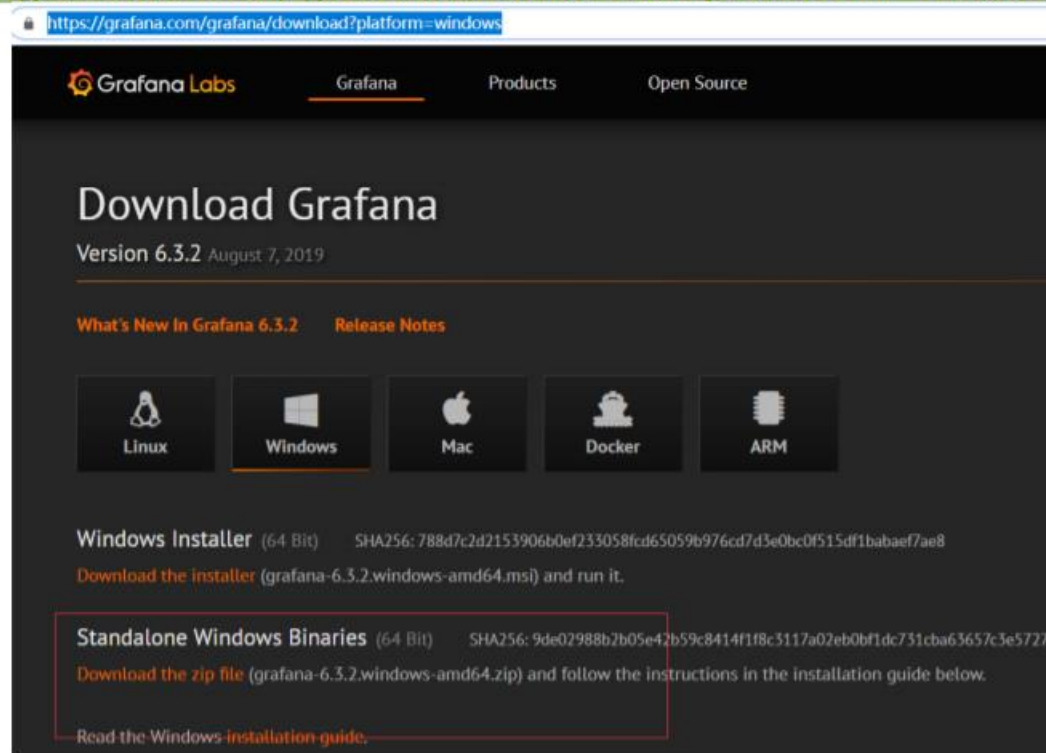
- <https://prometheus.io/download/> 下载对应的安装
- 解压后，直接执行 `./prometheus.exe` 就可以运行

Grafana介绍

- Grafana是一个跨平台的开源的度量分析和可视化工具，可以通过将采集的数据查询然后可视化的展示，并及时通知。
- 代码完全开源，并且是采用golang开发的
 - 展示方式：快速灵活的客户端图表，面板插件有许多不同方式的可视化指标和日志，官方库中具有丰富的仪表盘插件，比如热图、折线图、图表等多种展示方式；
 - 数据源：Graphite, InfluxDB, OpenTSDB, Prometheus, Elasticsearch, CloudWatch和KairosDB等；
 - 通知提醒：以可视方式定义最重要指标的警报规则，Grafana将不断计算并发送通知，在数据达到阈值时通过Slack、PagerDuty等获得通知；
 - 混合展示：在同一图表中混合使用不同的数据源，可以基于每个查询指定数据源，甚至自定义数据源；

Grafana安装

- 下载地址: <https://grafana.com/grafana/download?platform=windows>



Grafana运行

- 使用 `./grafana-server.exe`进行运行

Grafana+prometues整合

- Prometues中配置要监控的对象
- Granfana配置需要展示的图表

Prometues基本使用

```
package main

import (
    "flag"
    "log"
    "net/http"

    "github.com/prometheus/client_golang/prometheus/promhttp"
)

var addr = flag.String("listen-address", ":8080", "The address to listen on for HTTP requests.")

func main() {
    flag.Parse()
    http.Handle("/metrics", promhttp.Handler())
    log.Fatal(http.ListenAndServe(*addr, nil))
}
```


Prometheus基本使用

- Counter计数器

Prometheus Gauge使用

Prometheus Histogram使用

Promethues中间开发

- 当前服务的元信息封装
 - 当前的服务名
 - 当前的方法名
 - 当前的环境，比如生成环境、测试环境
 - 当前服务的集群
 - 当前服务的机房
 - 当前请求的trace_id
 - 当前服务器的ip
 - 客户端请求的ip
- 如何存储和传递
 - 通过context进行传递
- 何时进行初始化
 - 路由的时候，在方法入口处进行初始化

Promethues中间开发

- 监控请求数量
- 监控请求错误数
- 监控请求耗时

Promethues中间开发

- 和koala框架集成
 - 路由生成器模块改造，初始化服务元信息
 - Main生成器模块改造，集成prometheus

Promethues中间件测试

- 配置prometheus抓取hello service
- 配置grafana指标