# Prometheus简介

Prometheus是由SoundCloud开发的开源监控报警系统和时序列数据库(TSDB)。Prometheus使用Go语言开发，是Google BorgMon监控系统的开源版本

## 特点

* 多维度数据模型。
* 灵活的查询语言。
* 不依赖分布式存储，单个服务器节点是自主的。
* 通过基于HTTP的pull方式采集时序数据。
* 可以通过中间网关进行时序列数据推送。
* 通过服务发现或者静态配置来发现目标服务对象。
* 支持多种多样的图表和界面展示，比如Grafana等。

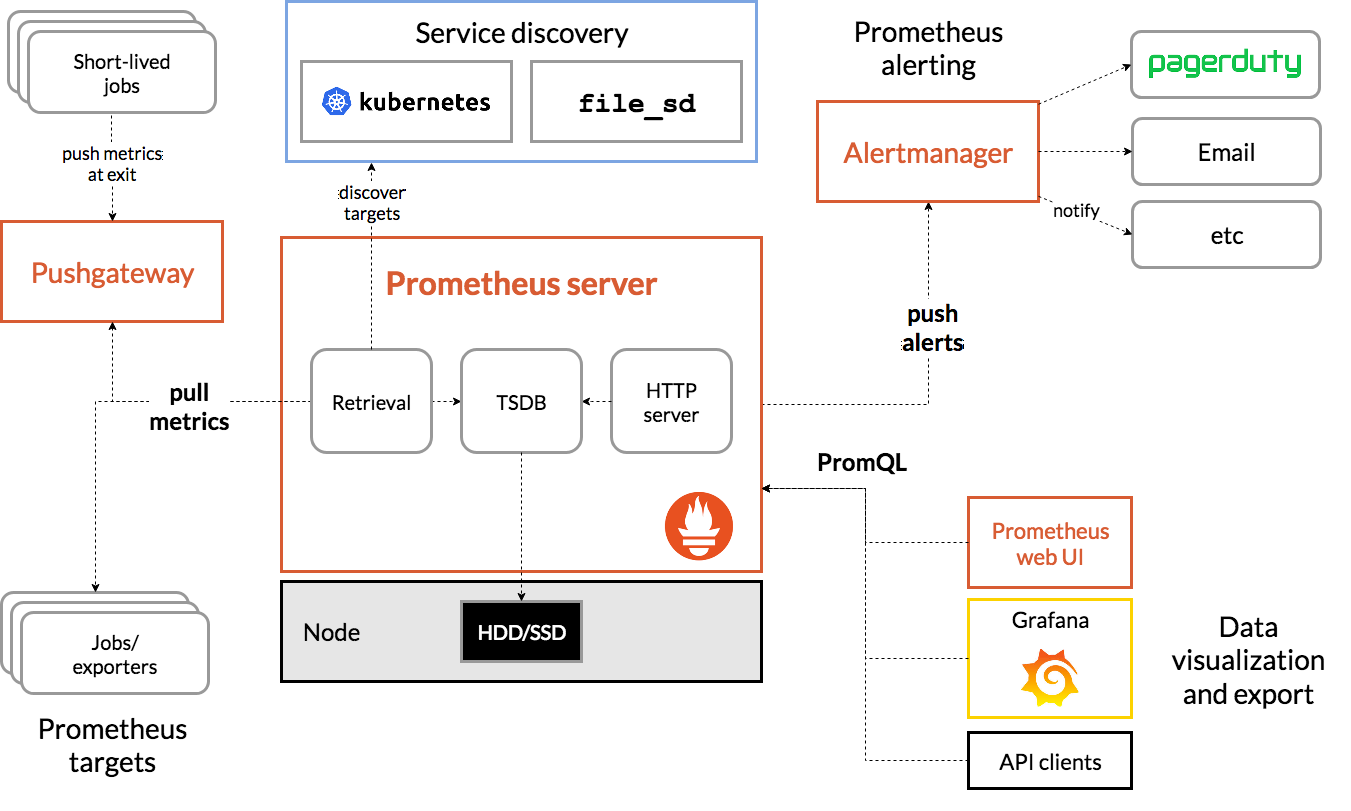
## 基本原理

Prometheus的基本原理是通过HTTP协议周期性抓取被监控组件的状态，任意组件只要提供对应的HTTP接口就可以接入监控。不需要任何SDK或者其他的集成过程。这样做非常适合做虚拟化环境监控系统，比如VM、Docker、Kubernetes等。输出被监控组件信息的HTTP接口被叫做exporter 。目前互联网公司常用的组件大部分都有exporter可以直接使用，比如Varnish、Haproxy、Nginx、MySQL、Linux系统信息(包括磁盘、内存、CPU、网络等等)。

## 服务过程

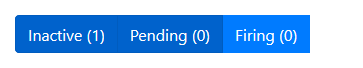
* Prometheus Daemon负责定时去目标上抓取metrics(指标)数据，每个抓取目标需要暴露一个http服务的接口给它定时抓取。Prometheus支持通过配置文件、文本文件、Zookeeper、Consul、DNS SRV Lookup等方式指定抓取目标。Prometheus采用PULL的方式进行监控，即服务器可以直接通过目标PULL数据或者间接地通过中间网关来Push数据。
* Prometheus在本地存储抓取的所有数据，并通过一定规则进行清理和整理数据，并把得到的结果存储到新的时间序列中。
* Prometheus通过PromQL和其他API可视化地展示收集的数据。Prometheus支持很多方式的图表可视化，例如Grafana、自带的Promdash以及自身提供的模版引擎等等。Prometheus还提供HTTP API的查询方式，自定义所需要的输出。
* PushGateway支持Client主动推送metrics到PushGateway，而Prometheus只是定时去Gateway上抓取数据。
* Alertmanager是独立于Prometheus的一个组件，可以支持Prometheus的查询语句，提供十分灵活的报警方式。

## 体系架构



## 报警规则

报警分为三种状态：



Inactive：正常

Pending：即将报警

Firing：触发报警，比如会发送邮件

比如下面cpu大于70，并且时间达到1分钟，将会触发报警

groups:

- name: node\_alerts

rules:

- alert: HighNodeCPU

#expr: go\_threads >14

expr: ((node\_memory\_MemTotal\_bytes - node\_memory\_MemFree\_bytes - node\_memory\_Buffers\_bytes - node\_memory\_Cached\_bytes) / (node\_memory\_MemTotal\_bytes )) \* 100 > 70

for: 1m

labels:

serverity: warning

annotations:

summary: High Node CPU for 1 m

console: You might want to check the Node Dashboard

# Prometheus 监控 Java 应用

## 监视方式

### 新项目嵌入jar包，编写代码

#### 接入依赖

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-actuator</**artifactId**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>io.micrometer</**groupId**>  
 <**artifactId**>micrometer-registry-prometheus</**artifactId**>  
 <**version**>1.1.3</**version**>  
</**dependency**>

#### 自定义指标



主要用途：

* Api访问次数
* 定时任务执行次数，成功与否
* 测试到数据库，缓存或者和其他服务之间的连通性

### prometheus的jmx\_exporter

使用jmx\_exporter的方式来监控我们的java应用程序。我们的java应用基本上是使用tomcat作为服务器的。这种情况下有两种方式，一种是基于springboot的jar包启动方式，一种是直接下载tomcat软件之后，将应用打成war包部署的方式。

#### Jar

java -javaagent:./jmx\_prometheus\_javaagent-0.3.0.jar=9151:config.yaml -jar yourJar.jar，这种方式启动。这种属于在应用启动的时候就给它加上代理

#### war

修改tomcat中catalina.sh文件，加上JAVA\_OPTS代理。这样启动后多一个端口用来提供给prometheus进行监控

# Alertmanager报警

一个独立的告警模块，接收Prometheus等客户端发来的警报，之后通过分组、删除重复等处理，并将它们通过路由发送给正确的接收器

# Grafana监控

是一款采用 go 语言编写的开源应用，主要用于大规模指标数据的可视化展现，是网络架构和应用分析中最流行的时序数据展示工具，目前已经支持绝大部分常用的时序数据库