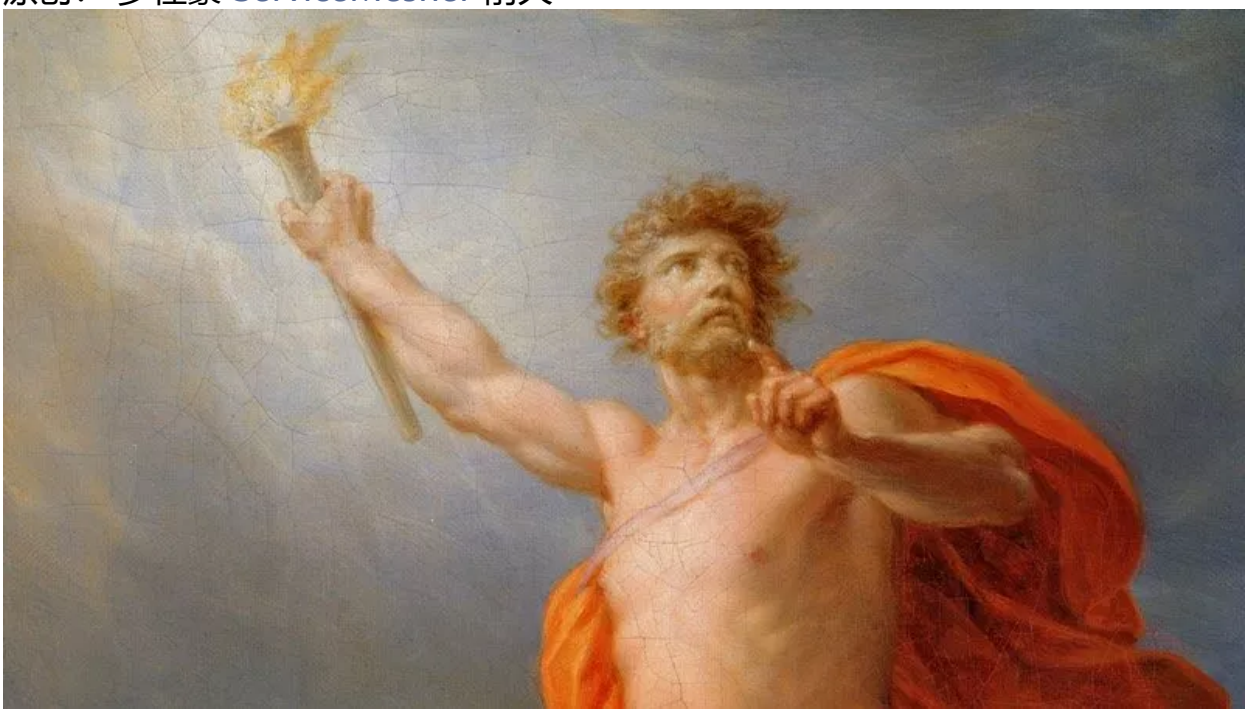
Prometheus监控Kubernetes系列1——监控框架

原创： 罗佳豪 ServiceMesher 前天



作者：罗佳豪 审校：孙海洲、李征、吴钧泽

由于容器化和微服务的大力发展，Kubernetes基本已经统一了容器管理方案，当我们使用Kubernetes来进行容器化管理的时候，全面监控Kubernetes也就成了我们第一个需要探索的问题。我们需要监控kubernetes的ingress、service、deployment、pod.....等等服务，以达到随时掌握Kubernetes集群的内部状况。此文章是Prometheus监控系列的第一篇，目的也很明确，旨在于寻找一套能够胜任kubernetes集群监控的架构。

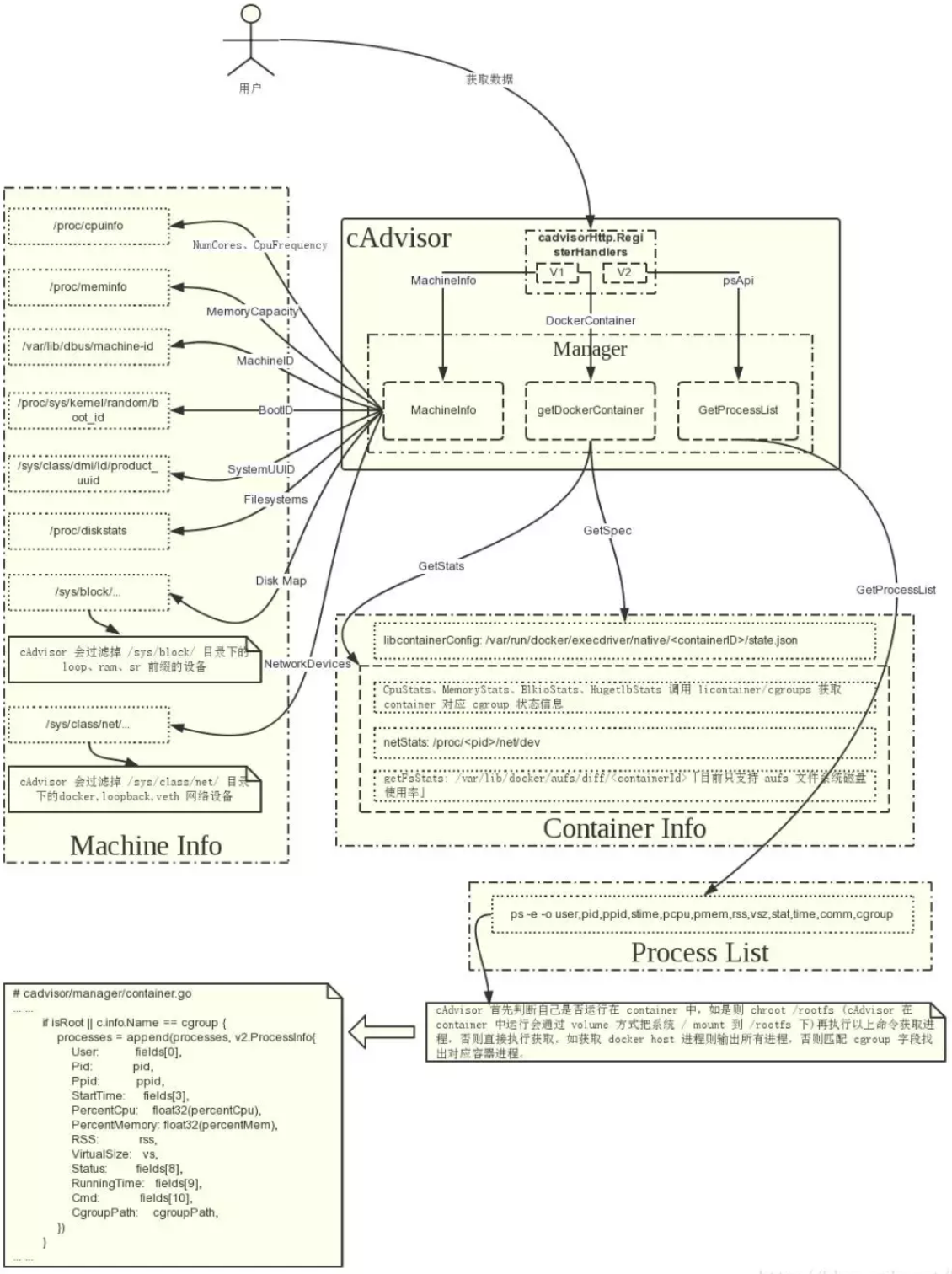
k8s监控方案调研

1、cAdvisor + InfluxDB + Grafana

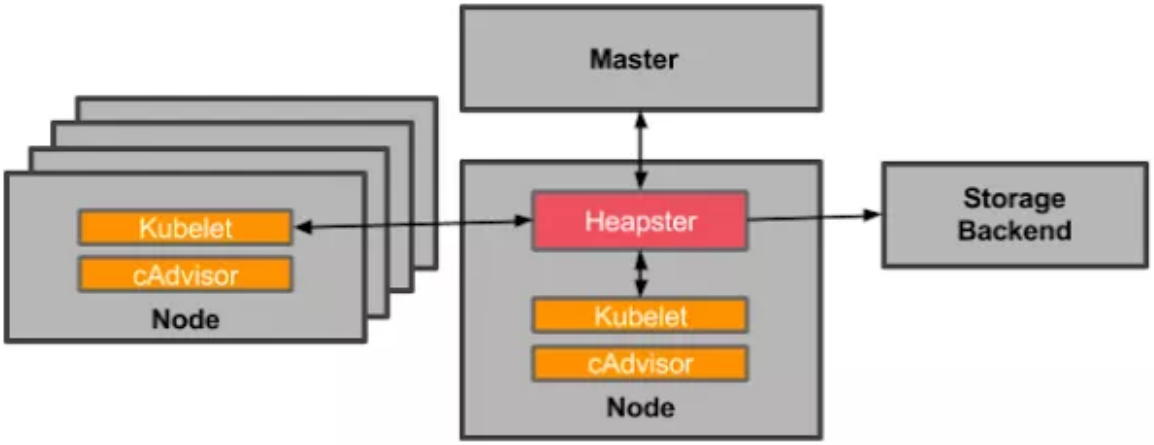
2、Heapster + InfluxDB + Grafana

3、Promethus + kube-state-metrics + Grafana

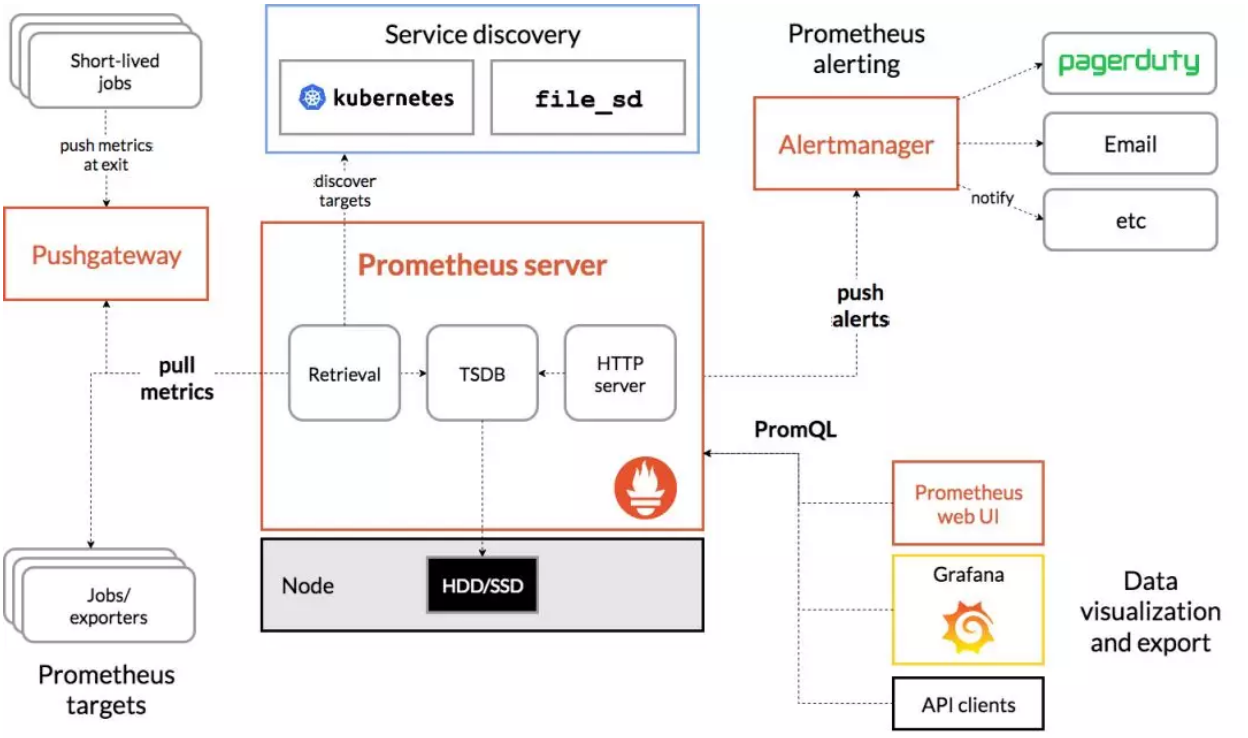
- **Grafana**: 开源DashBoard，后端支持多种数据库，如：Influxdb、Prometheus...，插件也比较多，功能强大。非常适合用于做展示。
- **InfluxDB**: 开源时间序列数据库，性能高效
- **cAdvisor**: 来自 Google 的容器监控工具，也是 Kubelet 内置的容器资源收集工具。它会自动收集本机容器 CPU、内存、网络和文件系统的资源占用情况，并对外提供 cAdvisor 原生的 API。随 kubelet 启动 --cadvisor-port = 1



- **Heapster**: 由于 cAdvisor 只提供了单机的容器资源占用情况，而 Heapster 则提供了整个集群的资源监控（kubernetes 1.11 之前，hpa都是从heapster获取数据），并支持持久化数据存储到 InfluxDB
-



- **Promethues**: 提供强大的数据采集、数据存储、数据展示、告警等，天生完美支持kubernetes，CNCF基金会的第二个成员，第一个是Kubernetes。而且Prometheus里面很多思想都来源于Google内部的监控系统Borgmon，可以说是Google的干儿子。



- **kube-state-metrics**在这里作为prometheus的一个exporter来使用，提供deployment、daemonset、cronjob等服务的监控数据，由kubernestes官方提供，与prometheus紧密结合。 更多关于kube-state-metrics的信息：https://github.com/kubernetes/kube-state-metrics

Prometheus优势

Prometheus和kubernetes相亲相爱

Google干儿子，大厂维护，而且最重要的一点是完美支持Kubernetes

规范定义

Prometheus对于应用层的监控，定义了一个良好的规范，只需要应用提供接口获取日志就可以了

Prometheus可以在各个层面实现监控，如下

- 基础设施层：监控各个主机服务器资源(包括Kubernetes的Node和非Kubernetes的Node)，如CPU,内存,网络吞吐和带宽占用,磁盘I/O和磁盘使用等指标。
- 中间件层：监控独立部署于Kubernetes集群之外的中间件，例如：MySQL、Redis、RabbitMQ、ElasticSearch、Nginx等。
- Kubernetes集群：监控Kubernetes集群本身的关键指标
- Kubernetes集群上部署的应用：监控部署在Kubernetes集群上的应用

基于以上三点，所以最终选择使用Prometheus来监控Kubernetes集群。

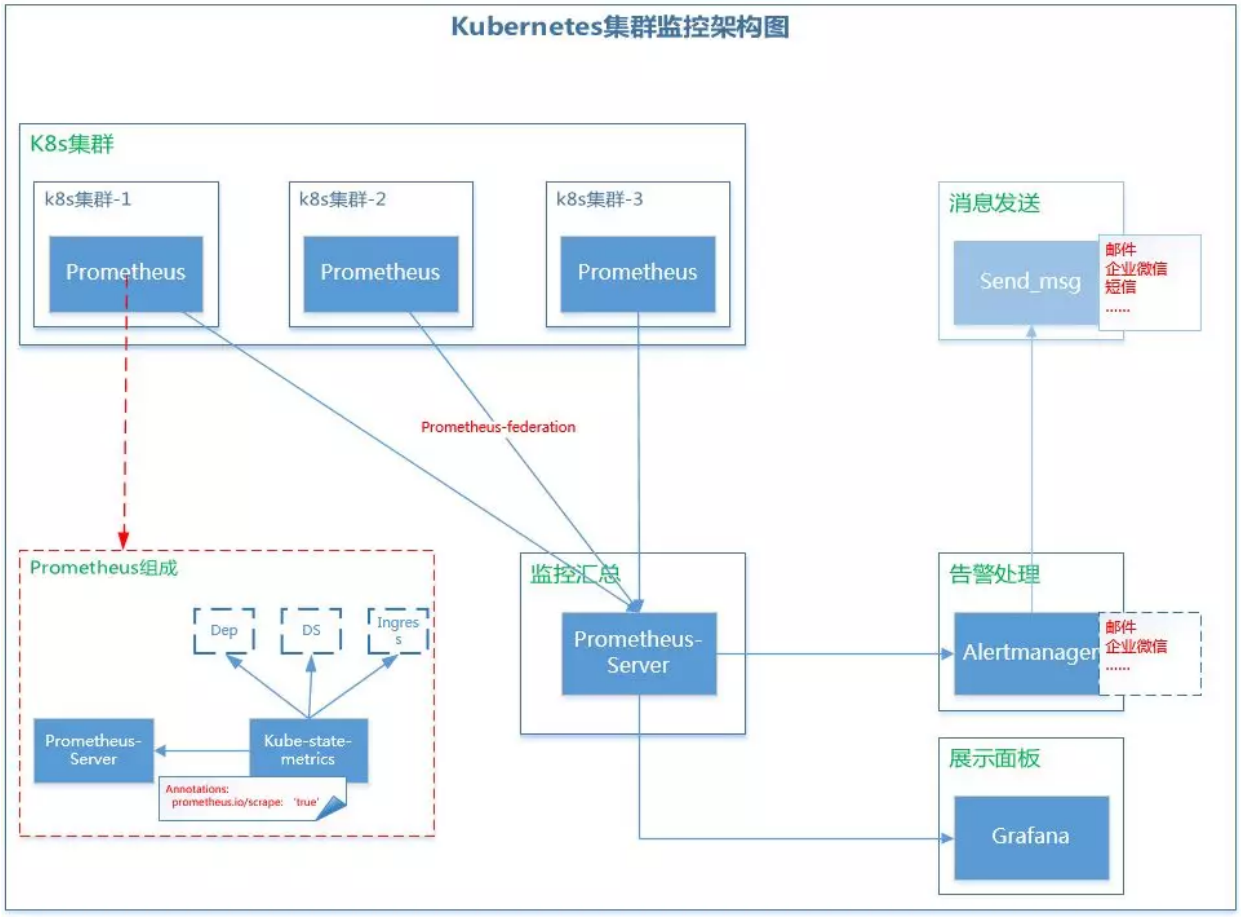
Kubernetes集群监控架构

在具体讨论Prometheus监控架构之前，再来看几个实际的问题

1. 如果有多个Kubernetes集群，怎么做?
2. 多个Kubernetes集群的监控数据怎么处理?
3. 告警应该怎么集中并去重?

好在这些问题对Prometheus来说都不是难事，最终，我们采取 Prometheus + kube-state-metrics + Alertmanager + Grafana 架构来做Kubernetes集群监控。

监控系统具体架构如下



使用这个架构，那上面所提到的三个问题将不再是问题。

详解

K8s集群：

k8s集群-1/-2/-3为需要被监控的集群，就是业务集群。每个集群内部都部署了一个Prometheus，主要由两部分组成 prometheus-server + kube-state-metrics。

prometheus-server：使用一个带RBAC权限的账号采集集群中现有监控信息（其实是从cadvisor获取）和节点信息。

kube-state-metrics：这里作为prometheus的exporter使用。因为prometheus不能获取集群中Deployment, Job, CronJob的监控信息。

部署kube-state-metrics的时候，svc一定要带一个annotations：

prometheus.io/scrape: 'true'（**这非常重要**）

监控汇总

监控汇总其实就是一个Prometheus-server，用于将各个散落在各地的监控数据汇总起来，统一管理。

核心思想是利用Prometheus的federation机制，从其他集群pull数据。这样其他集群的prometheus只需要短暂存储数据，汇总之后再做长期存储；同时还可以统一做告警判断和数据展示。

Prometheus官方Federation示例

```
- job_name: 'federate'
  scrape_interval: 15s
  honor_labels: true
  metrics_path: '/federate'
  params:
    'match[]':
      - '{job="prometheus"}'
      - '{__name__=~"prometheus_job:.*"}'
  static_configs:
    - targets:
      - 'source-prometheus-1:9090'
      - 'source-prometheus-2:9090'
      - 'source-prometheus-3:9090'
```

这段配置所属的Prometheus将从source-prometheus-1 ~ 3这3个Prometheus的/federate端点拉取监控数据。 match[]参数指定了只拉取带有job="prometheus"标签的指标，或者名称以prometheus_job开头的指标。

展示面板

展示面板就是一个Grafana，支持使用Prometheus做为数据源进行绘图展示。

告警处理

告警是利用Prometheus官方提供的Altermanager模块。Alermanager模块从Prometheus-Server接收告警信息，然后进行汇总、屏蔽、告警...等等操作。Alertmanager告警途径支持有email、wechat、webhook、slack等等，非常丰富。但是这里使用的是自身开发的Send_msg模块。

消息发送

自主开发的消息发送模块，集成email、微信、钉钉、短信等方式。其实不止告警时会发送消息，还有其他地方也会用到消息发送。

监控架构清楚之后，接下来就是实施监控的一个过程了，具体实施步骤请看“Prometheus系列”第二篇文章。

结束

此文章也是“使用prometheus完美监控kubernetes集群”系列的第一篇，对文章有不理解的地方，欢迎随时后台留言。

推荐阅读

Istio中使用Prometheus进行应用程序指标度量

全手动部署prometheus operator监控kubernetes集群以及一些坑

Envoy service mesh、Prometheus和Grafana下的微服务监控

对使用Kubernetes和Istio管理的基于容器的基础设施进行全面服务监控

Twistlock使Istio的安全层更强大，更易于监控