Prometheus

1、云原生监控系统，是以一个metric（度量指标）监控系统

2、已成为它已经成为炙手可热的 Kubernetes 生态圈中的核心监控系统。

3、普罗米修斯与雅典娜一起创造了人类

4、开源监控技术

Prometheus架构、特点、核心概念

架构

特点

1、多维度数据模型，一个时间序列由一个度量指标和多个标签键值对确定

2.灵活的查询语言，对收集的时序数据进行重组

3.强大的数据可视化功能，除了内置浏览器，也支持跟grafana集成

4.高效存储，内存加本地磁盘，可以通过功能分片和联盟来扩展性能

5.运维简单，只依赖本地磁盘，二进制安装包，没有任何其他库包依赖

6.提供许多导出器来收集常见的系统指标

核心概念

数据模型：

度量指标（metric）、标签（tag）、采样值（Sameple）、注解（Anotation）

度量指标：

计数器（counter）、计量器（Gauge）、直方图、汇总

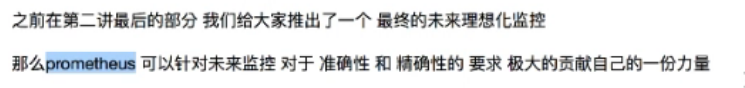
任务和实例：

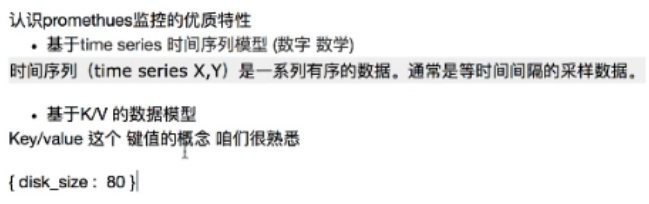
任务（job）、实例(instance)

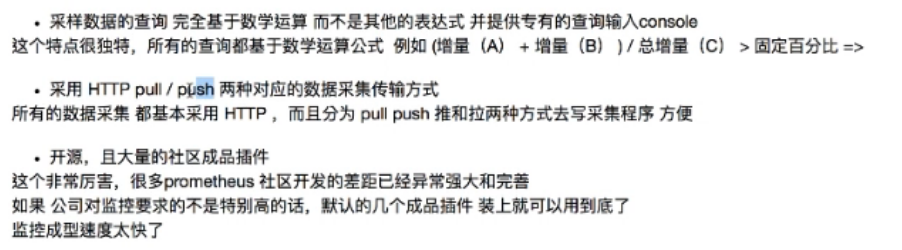
配置：对于静态配置只需要配置节点和端口http://localhost:9090/metrics

Prometheus的查询语句

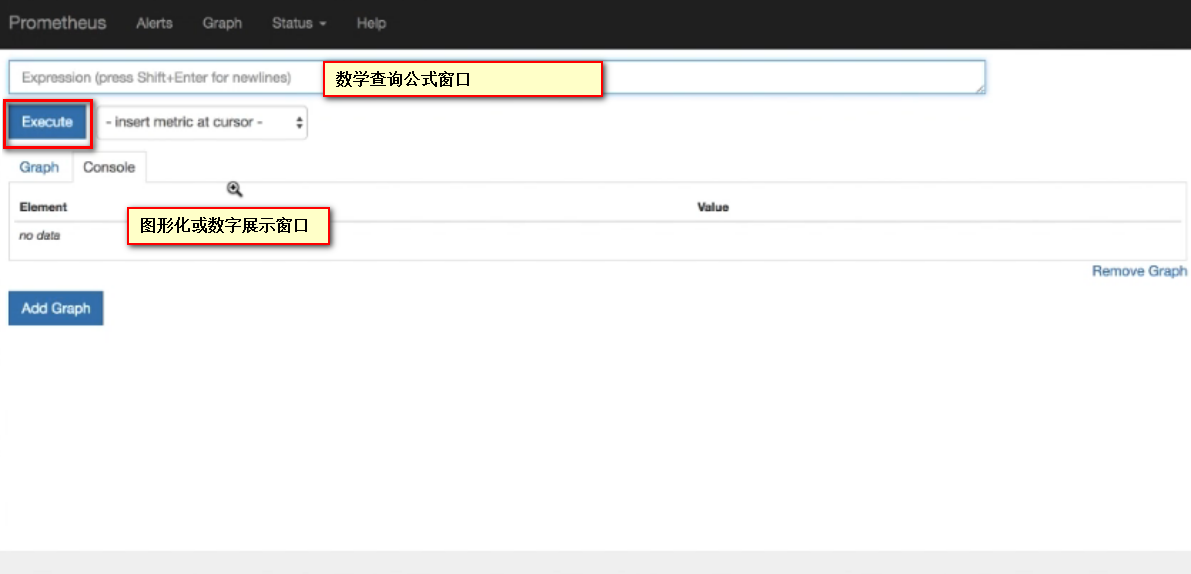


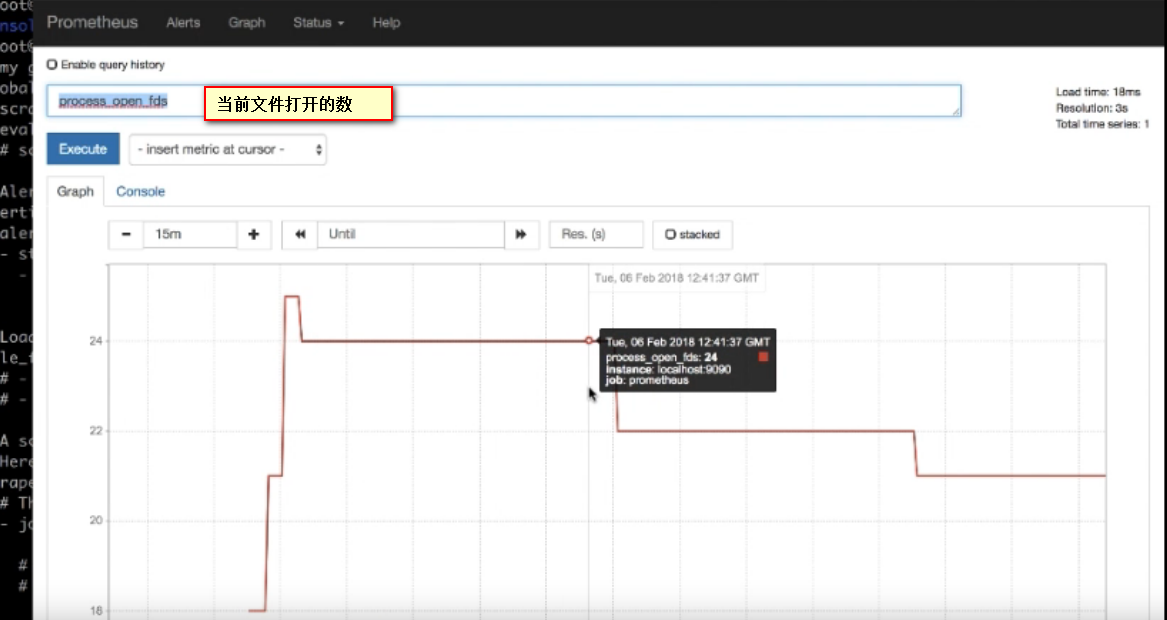




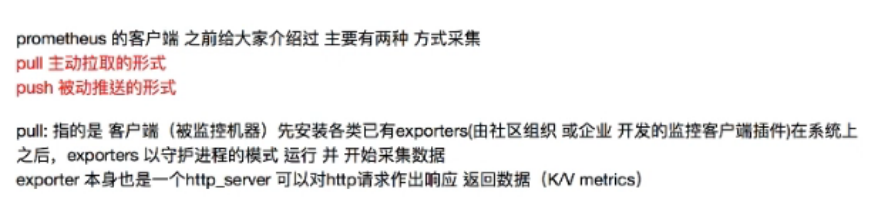


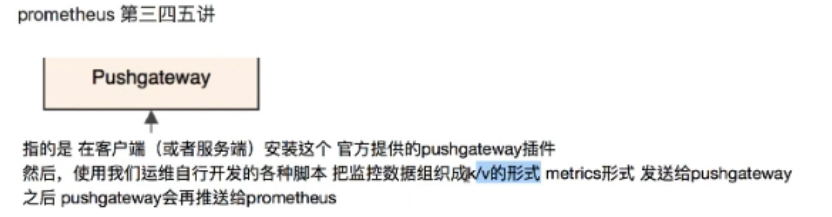
图形操作界面

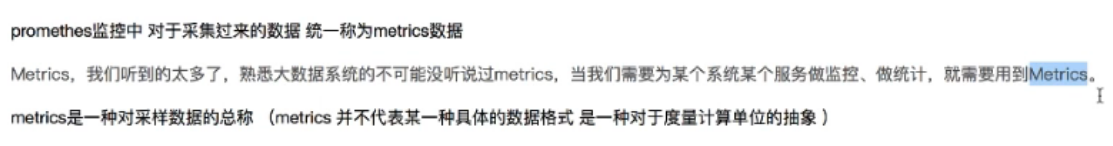


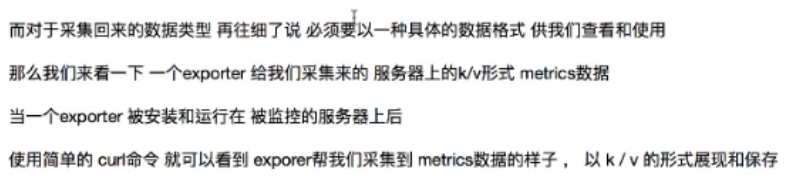


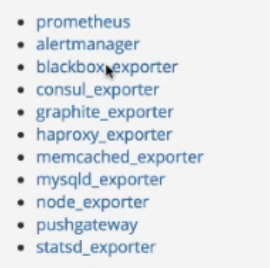




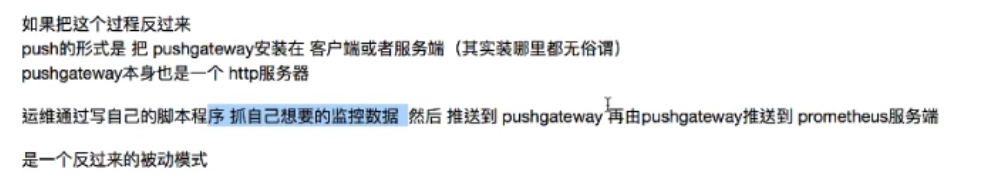


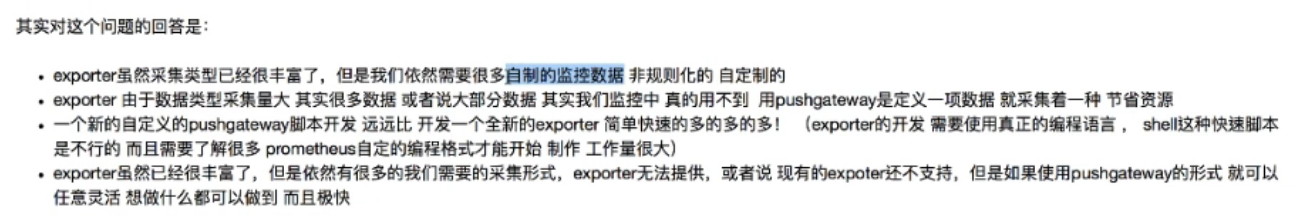


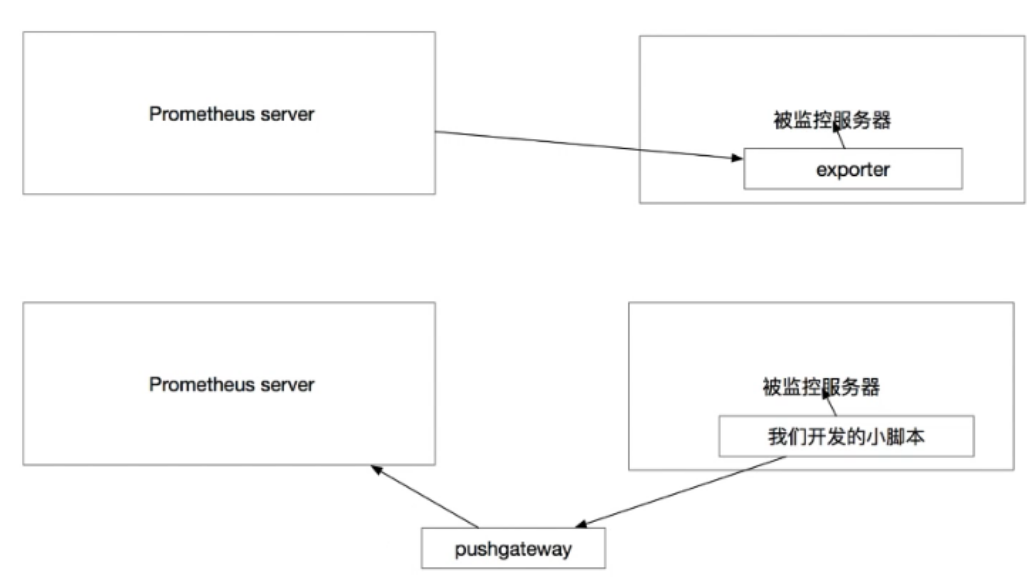




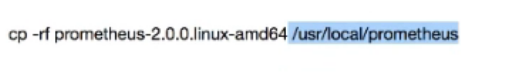


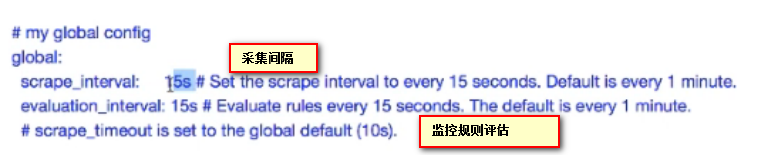


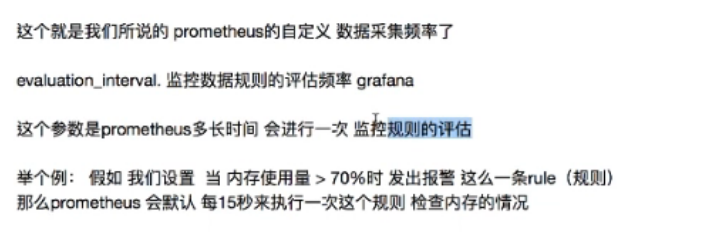


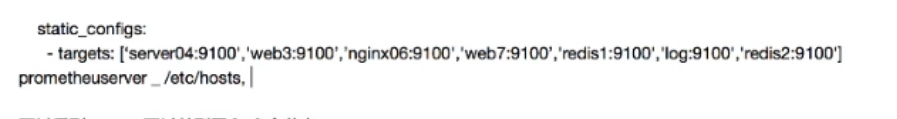
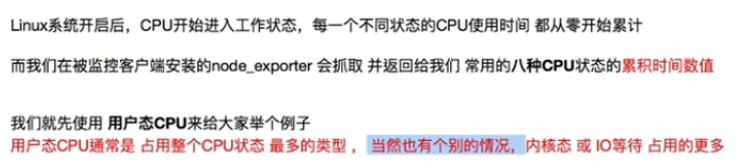


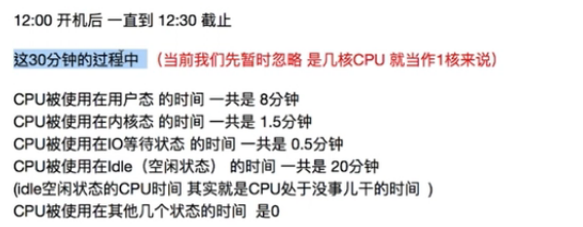
安装

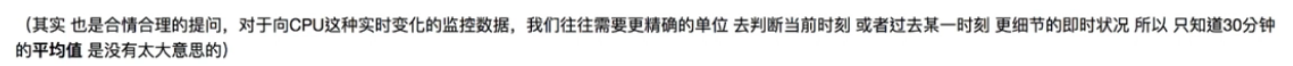


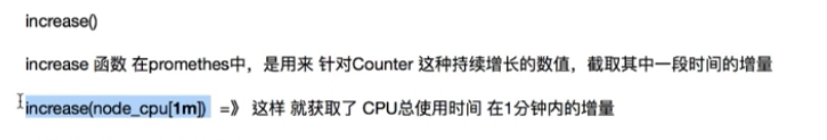










参考博客

<https://www.cnblogs.com/yangxiaoyi/p/7398156.html>

<https://www.liangzl.com/get-article-detail-8844.html>

<https://www.jianshu.com/p/93c840025f01>

<https://songjiayang.gitbooks.io/prometheus/content/pushgateway/how.html>

Prometheus 采用 pull 模式，可能由于不在一个子网或者防火墙原因，导致 Prometheus 无法直接拉取各个 target 数据。在监控业务数据的时候，需要将不同数据汇总, 由 Prometheus 统一收集。Prometheus采集数据是用的pull也就是拉模型，这从我们刚才设置的5秒参数就能看出来。但是有些数据并不适合采用这样的方式，对这样的数据可以使用Push Gateway服务。它就相当于一个缓存，当数据采集完成之后，就上传到这里，由Prometheus稍后再pull过来

Prometheus相比其他监控系统，更加耗费资源，监控的粒度更细。

Prometheus的使用

* 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。
* 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。
* Alertmanager 根据配置文件，对接收到的警报进行处理，发出告警。
* 在图形界面中，可视化采集数据。

prometheus自带的绘图功能比较弱，如何才能使展示效果高大上呢？这里就要使用Grafana.借助与Grafana可以完全自定义属于自己的Dashboard监控视图

Grafana的默认端口

<http://localhost:3000>

Granafa的使用

<https://blog.csdn.net/cs4290790/article/details/90670014>

为了Prometheus能够更好的工作，提供的一些插件，如服务发现、Exporter、PushGateWay、UI，Alermanger

运维是什么：就是管理服务器，为企业的产品和服务提供稳定运行的环境

Prometheus数据采集方式非常灵活

Evaluation \_interval:监控数据规则评估的频率，也就是Prometheus多长时间会进行一次规则评估，例如：当CPU使用率大于80%，发出报警（rule），那么Prometheus会在这个时间执行规则，对规则进行评估。

配置文件：

抓取数据的配置

任务名称

定义监控项：应用、服务器、进程等等

一旦配置成功，就会从相应的节点抓取数据

Node\_memory\_free:空闲内存

CPU使用率：基本上使用的top进行CPU使用率的查看，除空闲时间之外的，使用时间。

CPU的状态时间，包含很多种情况，那么我们如何获取的CPU使用率呢？

这是Prometheus提供了内置函数进行cpu时间的查询以及使用率的计算

Node\_cpu:返回的是

实际CPU是多核的，