

## 4 监控

### 一、简述常见的监控软件？

**Cacti:** 是一套基于 PHP、MySQL、SNMP 及 RRDTool 开发的网络流量监测图形分析工具。

**Zabbix:** Zabbix 是一个企业级的高度集成开源监控软件，提供分布式监控解决方案。可以用来监控设备、服务等可用性和性能。

**Open-falcon:** open-falcon 是一款用 golang 和 python 写的监控系统，由小米启动这个项目。

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

### 二、简述 Prometheus 及其主要特性？

#### 1、简述

Prometheus 是一个已加入 CNCF 的开源监控报警系统和时序数据库项目，通过不同的组件完成数据的采集，数据的存储和告警。

#### 2、主要特性

##### 1. 多维数据模型

时间序列数据通过 **metric** 名和键值对来区分。  
所有的 **metrics** 都可以设置任意的多维标签。  
数据模型更随意，不需要刻意设置为以点分隔的字符串。  
可以对数据模型进行聚合，切割和切片操作。  
支持双精度浮点类型，标签可以设为全 **unicode**。

##### 2. 灵活的查询语句 (PromQL)

可以利用多维数据完成复杂的查询

##### 3. 集成度高

Prometheus server 是一个单独的二进制文件，不依赖（任何分布式）存储，支持 local 和 remote 不同模型

##### 4. 数据拉取

采用 http 协议，使用 pull 模式，拉取数据，或者通过中间网关推送方式采集数据

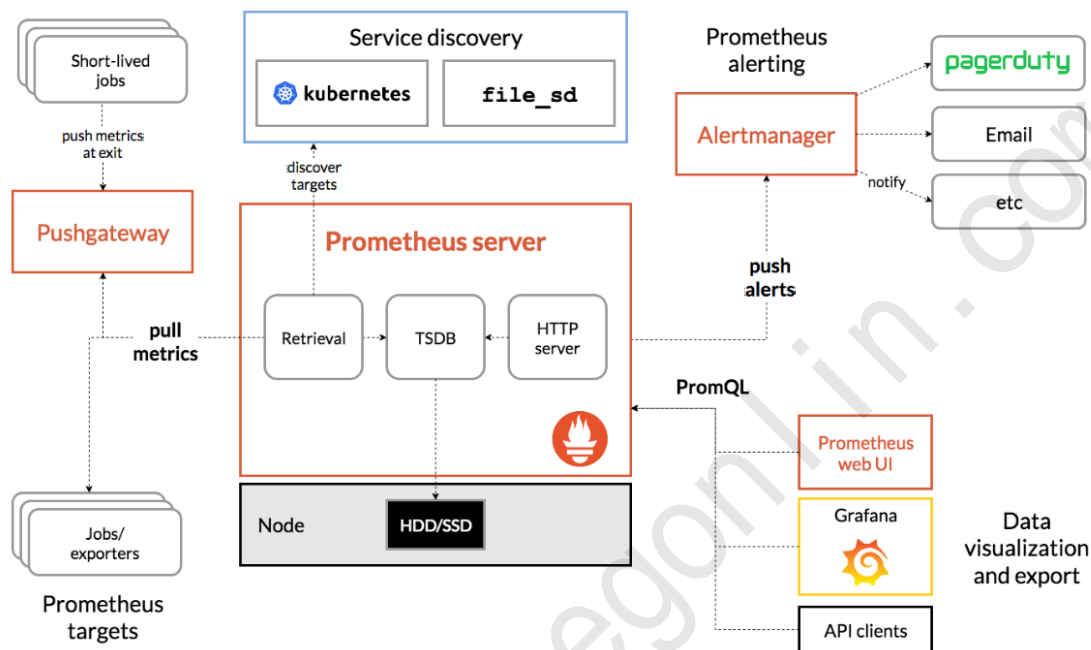
##### 5. 目标发现

监控目标，可以采用服务发现或静态配置的方式

## 6. 高效

一个 Prometheus server 可以处理数百万的 metrics 适用于以机器为中心的监控以及高度动态面向服务架构的监控

## 三、简述 Prometheus 主要组件及其功能？



### 1. Prometheus Server

#### Retrieval

获取监控数据

#### TSDB:

时间序列数据库(Time Series Database)，我们可以简单的理解为一个优化后用来处理时间序列数据的软件，并且数据中的数组是由时间进行索引的。具备以下特点：

大部分时间都是顺序写入操作，很少涉及修改数据

删除操作都是删除一段时间的数据，而不涉及到删除无规律数据

读操作一般都是升序或者降序

#### HTTP Server

为告警和出图提供查询接口

### 2. Pull metrics 指标采集

#### Exporters:

Prometheus 的一类数据采集组件的总称。它负责从目标处搜集数据，并将其转化为 Prometheus 支持的格式。与传统的数据采集组件不同的是，它并不向中央服务器发送数据，而是等待中央服务器主动前来抓取

用于暴露已有的第三方服务的 metrics 给 Prometheus。



## Pushgateway

用于网络不可直达或者生命周期比较短的数据采集 job，居于 exporter 与 server 端的中转站，将多个节点数据汇总到 Push Gateway，再统一推送到 server。

## 3. service discovery 服务发现

### Kubernetes\_sd

支持从 Kubernetes 中自动发现服务和采集信息。而 Zabbix 监控项原型就不适合 Kubernetes，因为随着 Pod 的重启或者升级，Pod 的名称是会随机变化的。

### file\_sd

通过配置文件来实现服务的自动发现

## 4.Alertmanager 单独抽离的告警组件

从 Prometheus server 端接收到 alerts(告警)后，会进行去除重复数据，分组，并路由到对收的接受方式，发出报警。常见的接收方式有：电子邮件，pagerduty，OpsGenie，webhook 等。

## 5. 图形化展示

通过 PromQL 语句查询指标信息，并在页面展示。虽然 Prometheus 自带 UI 界面，但是大部分都是使用 Grafana 出图。另外第三方也可以通过 API 接口来获取监控指标。

## 四、简述 prometheus 工作流程

- 1、Prometheus server 定期从配置好的 jobs 或者 exporters（出口）中拉 metrics（指标），或者接收来自 Pushgateway 发过来的 metrics（指标），或者从其他的 Prometheus server 中拉 metrics（指标）。
- 2、默认使用的拉取方式是 pull，也可以使用 pushgateway 提供的 push 方式获取各个监控节点的数据。
- 3、将获取到的数据存入 TSDB，一款时序型数据库。
- 4、此时 prometheus 已经获取到了监控数据，可以使用内置的 PromQL 进行查询。
- 5、它的报警功能使用 Alertmanager 提供，Alertmanager 是 prometheus 的告警管理和发送报警的一个组件。
- 6、prometheus 原生的图标功能过于简单，可将 prometheus 数据接入 grafana，由 grafana 进行统一管理。

## 五、说一下 prometheus 配置？

### 1.全局配置

global 标签下的配置，会作用到所有的抓取目标。有 4 种配置

scrape\_interval: 拉取 targets 的默认时间间隔。

scrape\_timeout: 拉取一个 target 的超时时间。

evaluation\_interval: 评估 rules 的时间间隔。

external\_labels: 和外部系统交流时，添加到数据中的标签。

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  evaluation_interval: 15s
```

## 2. 规则文件配置

rule\_files 标签下的配置，它指定了一系列的 rules 文件。

```
rule_files:
  - "rules/simple.rules"
  - "rules/rules/*.rules"
```

## 3. 数据源配置

scrape\_configs 节点下的配置。主要用来配置拉取的数据。

```
scrape_configs:
- job_name: node
  metrics_path: /metrics
  scheme: http
  static_configs:
    - targets:
        - 10.1.204.109:9100
      labels:
        instance: service1
    - targets:
        - 10.1.205.235:9100
      labels:
        instance: private
- job_name: consul-knowledge
  metrics_path: /actuator/prometheus
  scheme: http
  consul_sd_configs:
    - server: 10.1.204.79:8500
      tag_separator: ','
      scheme: http
      allow_stale: true
      refresh_interval: 30s
      services:
        - knowledge-service
        - gateway-service
```

## 4. 告警配置

alerting 节点下的配置，用来指定 Alertmanager 相关的配置。

# Alerting specifies settings related to the Alertmanager.

```
alerting:
  alert_relabel_configs:
    [ - <relabel_config> ... ]
  alertmanagers:
    [ - <alertmanager_config> ... ]
```



## 5. 服务发现配置

```
- job_name: kubernetes-nodes
kubernetes_sd_configs:
- role: node
```

## 6. 热更新

prometheus 热更新: `curl -X POST "http://10.102.74.90:9000/-/reload"`

# zabbix

## 一、zabbix 的优势

Zabbix 是一个企业级的高度集成开源监控软件，提供分布式监控解决方案。可以用来监控设备、服务等可用性和性能。其主要优势有：

- 1、自由开放源代码产品，可以对其进行任意修改和二次开发，采用 GPL 协议；
- 2、安装和配置简单；
- 3、搭建环境简单，基于开源软件构建平台；
- 4、完全支持 Linux、Unix、Windows、AIX、BSD 等平台，采用 C 语言 5、编码，系统占用小，数据采集性能和速度非常快；
- 6、数据采集持久存储到数据库，便于对监控数据的二次分析；
- 7、非常丰富的扩展能力，轻松实现自定义监控项和实现数据采集。

## 二、zabbix 结构

### 1、zabbix server

**Zabbix Server:** 负责接收 agent 发送的报告信息的核心组件，所有配置、统计数据及操作数据均由其组织进行。

### 2、database

专用于存储所有配置信息，以及有 zabbix 收集的数据。

### 3、web interface

zabbix 的 GUI 接口，通常与 server 运行在同一台机器上。

### 4、proxy

可选组件，常用于分布式监控环境中，代理 Server 收集部分被监控数据并统一发往 Server 端。

### 5、agent

部署在被监控主机上，负责收集本地数据并发往 Server 端或者 Proxy 端。

## 三、zabbix 支持的监控方式

### 1、Zabbix agent checks

这些客户端来进行数据采集，又分为 Zabbix agent（被动模式：客户端等着服务器端来要数据），Zabbix agent (active)（主动模式：客户端主动发送数据到服务器端）

## 2、SNMP agent checks

SNMP 方式，如果要监控打印机网络设备支持 SNMP 设备的话，但是又不能安装 agent 的设备。

## 3、SNMP traps

## 4、IPMI checks

IPMI 即智能平台管理接口，现在是业界通过的标准。用户可以利用 IPMI 监视服务器的物理特性，如温度、电压、电扇工作状态、电源供应以及机箱入侵等。

## 四、简述 zabbix 分布式及其适应场景

zabbix proxy 可以代替 zabbix server 收集性能和可用性数据，然后把数据汇报给 zabbix server，并且在一定程度上分担了 zabbix server 的压力。

此外，当所有 agents 和 proxy 报告给一个 Zabbix server 并且所有数据都集中收集时，使用 proxy 是实现集中式和分布式监控的最简单方法。

zabbix proxy 使用场景：

- 监控远程区域设备
- 监控本地网络不稳定区域
- 当 zabbix 监控上千设备时，使用它来减轻 server 的压力
- 简化分布式监控的维护

