



tailnode 发布于 冥灭

2017年03月14日 · 8.7k 次阅读

alertmanager报警规则详解

转载请注明出处，原文链接<http://tailnode.tk/2017/03/al...>

说明

这篇文章介绍prometheus和alertmanager的报警和通知规则，prometheus的配置文件名为

`prometheus.yml`，alertmanager的配置文件名为 `alertmanager.yml`

报警：指prometheus将监测到的异常事件发送给alertmanager，而不是指发送邮件通知

通知：指alertmanager发送异常事件的通知（邮件、webhook等）

报警规则

在 `prometheus.yml` 中指定匹配报警规则的间隔

```
# How frequently to evaluate rules.  
[ evaluation_interval: <duration> | default = 1m ]
```

在 `prometheus.yml` 中指定规则文件（可使用通配符，如`rules/*.rules`）

```
# Load rules once and periodically evaluate them according to the global 'evaluation_interval'  
rule_files:  
- rules/mengyuan.rules
```

在rules目录中添加 `mengyuan.rules`

ALERT goroutines_gt_70



首页



问答



专栏



讲堂



更多

```
LABELS { status = "yellow" }
ANNOTATIONS {
  summary = "goroutines 超过 70, 当前值{{ $value }}",
  description = "当前实例 {{ $labels.instance }}",
}
```

ALERT goroutines_gt_90

```
IF go_goroutines > 90
FOR 5s
LABELS { status = "red" }
ANNOTATIONS {
  summary = "goroutines 超过 90, 当前值{{ $value }}",
  description = "当前实例 {{ $labels.instance }}",
}
```

配置文件设置好后，需要让prometheus重新读取，有两种方法：

1. 通过HTTP API向 `/-/reload` 发送POST请求，例：`curl -X POST http://localhost:9090/-/reload`
2. 向prometheus进程发送SIGHUP信号

将邮件通知与rules对比一下（还需要配置 `alertmanager.yml` 才能收到邮件）



通知规则

设置 `alertmanager.yml` 的route与receivers

route:

```
# The labels by which incoming alerts are grouped together. For example,
# multiple alerts coming in for cluster=A and alertname=LatencyHigh would
# be batched into a single group.
group_by: ['alertname']

# When a new group of alerts is created by an incoming alert, wait at
# least 'group_wait' to send the initial notification.
# This way ensures that you get multiple alerts for the same group that start
# firing shortly after another are batched together on the first
# notification.
group_wait: 5s

# When the first notification was sent, wait 'group_interval' to send a batch
```



首页



问答



专栏



讲堂



更多

```
# If an alert has successfully been sent, wait 'repeat_interval' to
# resend them.
repeat_interval: 3h

# A default receiver
receiver: mengyuan
```

名词解释

Route

`route` 属性用来设置报警的分发策略，它是一个树状结构，按照深度优先从左向右的顺序进行匹配。

```
// Match does a depth-first left-to-right search through the route tree
// and returns the matching routing nodes.
func (r *Route) Match(lset model.LabelSet) []*Route {
```

Alert

Alert是alertmanager接收到的报警，类型如下。

```
// Alert is a generic representation of an alert in the Prometheus eco-system.
type Alert struct {
    // Label value pairs for purpose of aggregation, matching, and disposition
    // dispatching. This must minimally include an "alertname" label.
    Labels LabelSet `json:"labels"`

    // Extra key/value information which does not define alert identity.
    Annotations LabelSet `json:"annotations"`

    // The known time range for this alert. Both ends are optional.
    StartsAt      time.Time `json:"startsAt,omitempty"`
    EndsAt        time.Time `json:"endsAt,omitempty"`
    GeneratorURL  string    `json:"generatorURL"`
}
```

具有相同 `Labels` 的Alert (key和value都相同) 才会被认为是同一种。在prometheus rules文件配置的一条规则可能会产生多种报警

Group



首页



问答



专栏



讲堂



更多

alertmanager会根据 `group_by` 配置将Alert分组。如下规则，当`go_goroutines`等于4时会收到三条报警，alertmanager会将这三条报警分成两组向receivers发出通知。

```
ALERT test1
  IF go_goroutines > 1
    LABELS {label1="l1", label2="l2", status="test"}
ALERT test2
  IF go_goroutines > 2
    LABELS {label1="l2", label2="l2", status="test"}
ALERT test3
  IF go_goroutines > 3
    LABELS {label1="l2", label2="l1", status="test"}
```

主要处理流程

1. 接收到Alert，根据labels判断属于哪些Route（可存在多个Route，一个Route有多个Group，一个Group有多个Alert）
2. 将Alert分配到Group中，没有则新建Group
3. 新的Group等待 `group_wait` 指定的时间（等待时可能收到同一Group的Alert），根据 `resolve_timeout` 判断Alert是否解决，然后发送通知
4. 已有的Group等待 `group_interval` 指定的时间，判断Alert是否解决，当上次发送通知到现在的间隔大于 `repeat_interval` 或者Group有更新时会发送通知

TODO

- 重启对发送报警与通知的影响
- 能否组成集群

参考

- <https://github.com/prometheus...>
- <https://prometheus.io/blog/20...>
- <http://studygolang.com/articl...>
- <http://www.admpub.com/blog/po...>



首页



问答



专栏



讲堂



更多

赞 | 1

收藏 | 7

你可能感兴趣的文章

- [Prometheus AlertManager 微信报警配置](#) 笨兔儿 prometheus
- [alertmanager邮件模版](#) tailnode prometheus
- [使用 Prometheus 监控 Docker 容器](#) yexiaobai container monitor 容器 docker prometheus
- [k8s与监控--从kubernetes监控谈prometheus的federation机制](#) iyacontrol k8s prometheus kubernetes
- [Prometheus的架构及持久化](#) codecraft prometheus
- [k8s与监控--解读prometheus监控kubernetes的配置文件](#) iyacontrol prometheus go语言 kubernetes
- [使用Prometheus和Grafana监控Mysql服务器性能](#) icyfire monitoring mysql grafana prometheus
- [Prometheus实战--存储篇](#) iyacontrol databaes kubernetes golang prometheus

评论

默认排序

时间排序

文明社会，理性评论

发布评论