AlertManager 使用总结

AlertManager

1.一条通知配置多个接收者

每个receiver下可以配置多个接收者，如下配置两个 webhook

```
global:
  resolve_timeout: 5m

route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 1m
  repeat_interval: 5m
  receiver: 'web.hook1'
receivers:
- name: 'web.hook1'
  webhook_configs:
  - url: 'http://127.0.0.1:5000/hook1'
  - url: 'http://127.0.0.1:5000/hook2'
inhibit_rules:
  - source_match:
      severity: 'critical'
    target_match:
      severity: 'warning'
    equal: ['alertname', 'dev', 'instance']
```

也可以配置不同类型的接受者

```
receivers:
- name: 'web.hook1'
```

```
  webhook_configs:
  - url: 'http://127.0.0.1:5000/hook1'
  - url: 'http://127.0.0.1:5000/hook2'
  email_config:
  - to: <mail to address>
  - to: <mail to address>
```

2.持久化通知

目前alertmanager并不支持持久化通知，也就是说告警并不会一直存储在数据库中，而prometheus只是存储告警规则和其状态，并没有像传统告警系统一般，会把什么时候发生的告警、告警接收人，理状态存储为一条记录。简单的说就是不能满足程序查找历史告警记录的需求，目前我是没找到可以接查询到的方法。

不过也是有一个临时解决方案，在不改动prometheus和alertmanger的源码前提下，开发一个webhook，用来接收alertmanager发送的所有告警通知，然后在webhook中处理存储数据库。

3.邮箱的配置

```
global:
  smtp_smarthost: 'smtp.xxx.aliyun.com:465'
  smtp_hello: 'company.com'
  smtp_from: 'username@company.com'
  smtp_auth_username: 'username@company.com'
  smtp_auth_password: password
  smtp_require_tls: false
route:
  group_by: ['alertname']
  receiver: 'default-receiver'

receivers:
  - name: default-receiver
    email_configs:
      - to: <mail to address>
        send_resolved: true
```

在email_configs可以配置的具体选项如下，copy来自官网

```
# Whether or not to notify about resolved alerts.
[ send_resolved: <boolean> | default = false ]

# The email address to send notifications to.
to: <tmpl_string>

# The sender address.
[ from: <tmpl_string> | default = global.smtp_from ]

# The SMTP host through which emails are sent.
[ smarthost: <string> | default = global.smtp_smarthost ]

# The hostname to identify to the SMTP server.
[ hello: <string> | default = global.smtp_hello ]
```

```
# SMTP authentication information.
[ auth_username: <string> | default = global.smtp_auth_username ]
[ auth_password: <secret> | default = global.smtp_auth_password ]
[ auth_secret: <secret> | default = global.smtp_auth_secret ]
[ auth_identity: <string> | default = global.smtp_auth_identity ]

# The SMTP TLS requirement.
# Note that Go does not support unencrypted connections to remote SMTP endpoints.
[ require_tls: <bool> | default = global.smtp_require_tls ]

# TLS configuration.
tls_config:
  [ <tls_config> ]

# The HTML body of the email notification.
[ html: <tmpl_string> | default = '{{ template "email.default.html" . }}' ]
# The text body of the email notification.
[ text: <tmpl_string> ]

# Further headers email header key/value pairs. Overrides any headers
# previously set by the notification implementation.
[ headers: { <string>: <tmpl_string>, ... } ]
```

可惜邮件发送不支持附件、抄送、密送等功能。

4.告警抑制

```
# Matchers that have to be fulfilled in the alerts to be muted.
target_match:
  [ <labelname>: <labelvalue>, ... ]
target_match_re:
  [ <labelname>: <regex>, ... ]

# Matchers for which one or more alerts have to exist for the
# inhibition to take effect.
source_match:
  [ <labelname>: <labelvalue>, ... ]
source_match_re:
  [ <labelname>: <regex>, ... ]

# Labels that must have an equal value in the source and target
# alert for the inhibition to take effect.
[ equal: '[' <labelname>, ... ']' ]
```

抑制规则示例

```
inhibit_rules:
  - source_match:
      severity: 'critical'
    target_match:
      severity: 'warning'
    equal: ['alertname', 'dev', 'instance']
```

告警的标签的key都拥有 alertname 或 dev或 instance, 匹配源告警标签拥有 severity=critical，目

告警符合 severity=warning都将被抑制。

5.对接AlertManager

有时候我们需要不经过prometheus，直接把告警发送给AlertManager，网上找了一堆都是配到使用，但实际场景往往较为复杂，可以从两个方面进行着手：

一、api

二、利用 amtool官方工具

原文链接：AlertManager 使用总结