

zabbix邮件--微信报警

永久关闭防火墙和selinux

公网邮箱报警：

邮件系统简要介绍：

电子邮件系统包括两个组件：MUA(Mail User Agent,邮件用户代理)和MTA(Mail Transport Agent,邮件传送代理)。MUA是邮件系统为用户提供的可以读写邮件的界面；而MTA是运行在底层，能够处理邮件的收发工作的程序。简单的说，用户可以使用MUA写信、读信、而通过MTA收信、发信。Foxmail、Outlook、Webmail都属于MUA。邮件的接收是MTA和MUA配合完成的。远程的MUA首先向远程MTA连接并验证发信人身份，然后由远程MTA向本地MTA发送邮件。接受者通过本地MUA接收阅读邮件。邮件的发信也是MTA和MUA配合完成的，不过方向正好相反。本地MUA首先向本地的MTA连接并验证发信人身份，然后由本地MTA向远程MTA发送邮件，再由远程的MUA读取邮件。

mail和mailx即为负责查看、编写邮件和向MTA发送邮件的MUA。mailx是mail的功能加强版。

sendmail即为负责邮件在网络上传输的MTA，将邮件从一个MTA传送至另一个MTA。

邮件服务器：

发邮件的人 -----放到投递箱 -----邮递员 -----放到邮局 -----分拣 -----邮递员-----邮局----分拣 -----收件箱-----
(pop3) 收信人

人：用户：root tom

投递箱：MUA -----web界面(可以直接登陆)和客户端应用程序

客户端应用程序：foxmail，outlook，手机邮箱app

邮递员：通信：SMTP 25号端口。

邮局：MTA：邮件传输代理

sendmail(已经过时)----postfix---->部署邮件服务器

分拣：MDA：邮件分发代理---dovecot(已经不用)-----maildrop

邮递员：SMTP：25

收信和发信的协议不一样

pop3：邮局协议第三版，-----端口110

pop3可以换成：imap4 -----端口143

发信端口25，收信端口110或者143 可以反过来

zabbix用到的：

mail或者mailx

公网邮箱：postfix

报警三大要素：

- 1、示警媒体(Media)
 - 2、触发器(Triggers)
 - 3、动作(Action)
-

开干：

- 1.注册163邮箱
 - 2.网页邮箱设置客户端授权密码
-

server服务器端：

安装MUA软件：mailx

yum install mailx

查看版本：
mail -V
12.5 7/5/10

2.配置公网邮箱信息：发邮件：
vim /etc/mail.rc 追加以下内容

```
set from=xuan668jv@163.com      //( 邮箱地址 )
set smtp=smtp.163.com          //(smtp服务器 ) 发邮件服务器 ---163默认
set smtp-auth-user=xuan668jv@163.com  //(用户名)
set smtp-auth-password=123456zc      //( 邮箱密码 ) 授权之后的密码
set smtp-auth=login              //默认不用动
```

可以发给任何一个收件邮箱

在终端使用**mailx**发邮件的方式：

方式1：mailx -s "邮件标题" 收件箱Email < 包含正文的文件

方式2：cat 包含正文的文件 | mailx -s "邮件标题" 收件箱Email

方式3：echo "正文内容" | mailx -s "邮件标题" 收件箱Email

方式4：mailx -s "邮件标题" 收件箱Email，回车按CTRL+D发送

或者，将邮件内容放到一个文件中content.txt

```
# cat content.txt | mailx -v -s 'hello' 'xuan668jv@163.com'
```

手动发送邮件测试：

```
# mailx -v -s 'hello' 'xuan668jv@163.com'
```

nihao (回车，然后ctrl+D

EOT

-v：显示发送过程

查看邮箱是否有消息如果有消息表示成功！

zabbix添加邮件报警功能：

注：使用新的方式--利用公网邮件服务器发送报警，需要关闭postfix服务

配置 zabbix 的邮件报警功能需要以下三个角色的参与。

- 1、示警媒体(Media)
- 2、触发器(Triggers)
- 3、动作(Action)

示警媒体：

指的是 zabbix 采用何种方式进行报警,目前 Zabbix 支持的示警媒体包括邮件、Jabber、短信。jabber是linux下一种即时通讯工具,可以和yahoo,icq等工具通信

触发器：

指的是当监控对象达到某个条件或条件集合的时候,触发 Zabbix 产生事件。

动作：

指的是 Zabbix 产生对应事件后,它通过示警媒体发送报警。

接下来,我们配置一个邮件报警功能的范例。效果是当 WEB 页面获取时间超过 3 秒,则触发报警,管理员将会收到一封 Zabbix 发出的报警邮件。

示警媒体的配置：

首先需要配置 Zabbix 的邮件功能。

点击 管理->报警媒介类型->创建媒体类型



然后在页面中填入你的报警媒介类型信息,例如下图所示:

注: 脚本名称任意, 存放于 `/usr/lib/zabbix/alertscripts` (生产上的测试服放这: `/usr/local/zabbix/share/zabbix/alertscripts`)

名称: sendmail //名称任意
类型: 脚本
脚本名称: sendmail.sh
脚本参数: //一定要写, 否则可能发送不成功
 {ALERT.SENDTO} //照填, 收件人变量
 {ALERT.SUBJECT} //照填, 邮件主题变量, 变量值来源于'动作'中的'默认接收人'
 {ALERT.MESSAGE} //照填, 邮件正文变量, 变量值来源于'动作'中的'默认信息'

配置完成后,不要忘记点击存档,保存你的配置。

名称

sendmail

类型

脚本

脚本名称

sendmail.sh

脚本参数

参数	动作
{ALERT.SENDTO}	移除
admin{ALERT.SUBJECT}	移除
{ALERT.MESSAGE}	移除

添加

已启用

☒

更新

克隆

删除

取消

修改zabbix服务端配置文件 & 编写脚本:

```
查看指定脚本的存储路径：
#vim /etc/zabbix/zabbix_server.conf
AlertScriptsPath=/usr/lib/zabbix/alertscripts
```

```
编写邮件脚本：
#cd /usr/lib/zabbix/alertscripts
#vim sendmail.sh
#!/bin/sh #export.UTF-8  -----字符集可以删除掉
echo "$3" | sed s/"\r"/\n/g | mailx -s "$2" $1
```

\$1:接受者的邮箱地址：sendto
\$2:邮件的主题：subject
\$3:邮件内容：message

```
修改权限：
#chmod u+x sendmail.sh && chown zabbix.zabbix sendmail.sh
```

创建的脚本名称要和定义脚本名称一样

修改admin用户的报警媒介：
用户默认是没有设置报警媒介的，设置后就可以接收报警消息了。

ZABBIX

监测中

资产记录

报表

配置

管理

一般

agent代理程序

认证

用户群组

用户

报警媒介类型

脚本

队列

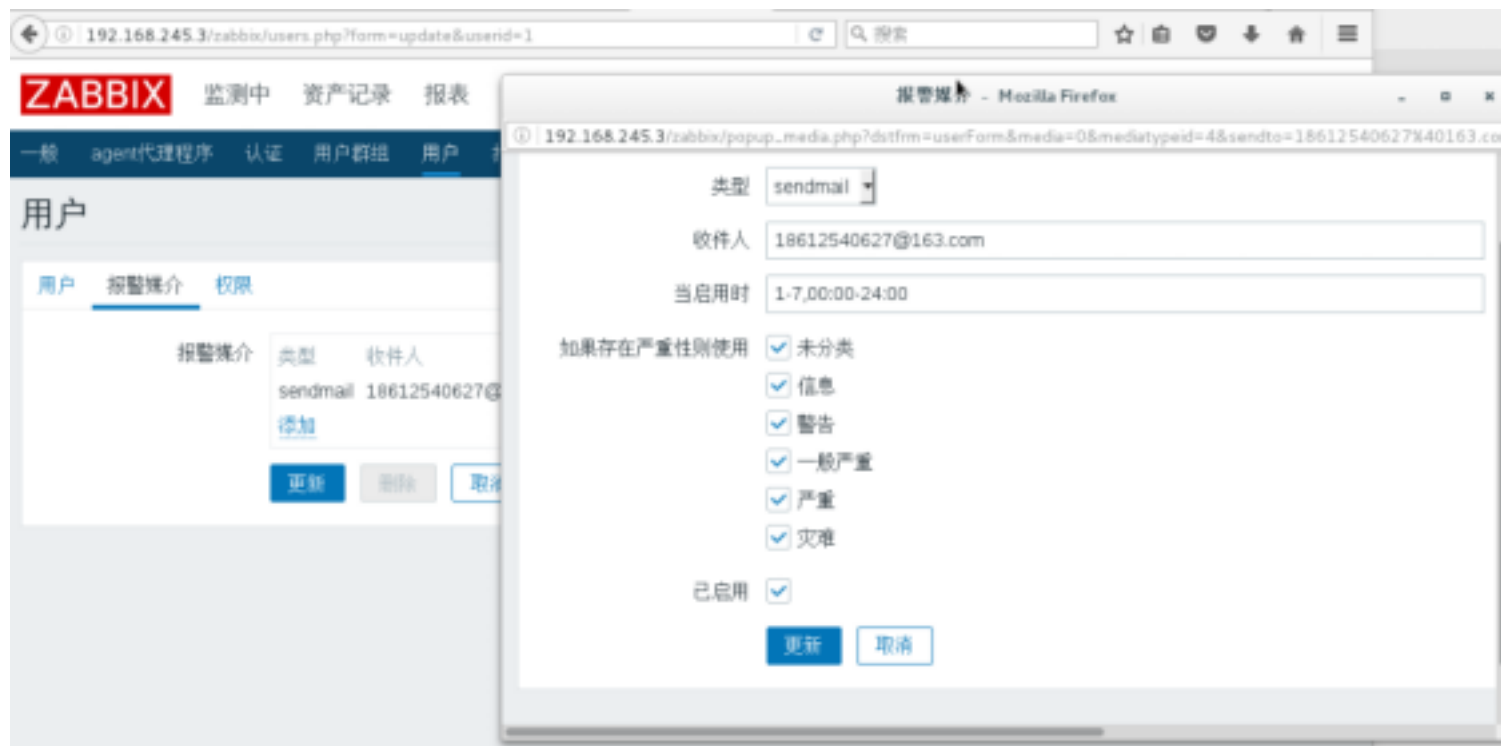
用户

<input type="checkbox"/> 别名 ▲	用户名第一部分	姓氏
<input type="checkbox"/> Admin	Zabbix	Administrator
<input type="checkbox"/> guest		

0 选择

解锁

删除



触发器的配置：

接下来,点击配置->主机

我们给 agent-19 这台主机增加一个触发器。点击 agent-19 这一行中的“触发器”,然后再点击创建触发器。该页各配置项含义如下：

名称:填入触发器的名字

表达式:用于配置触发器的触发条件,点击添加按钮有条件选项。 ----键值

多重事件产生:如果选中,则问题如果持续多重的发生则每次都触发,否则只触发一次

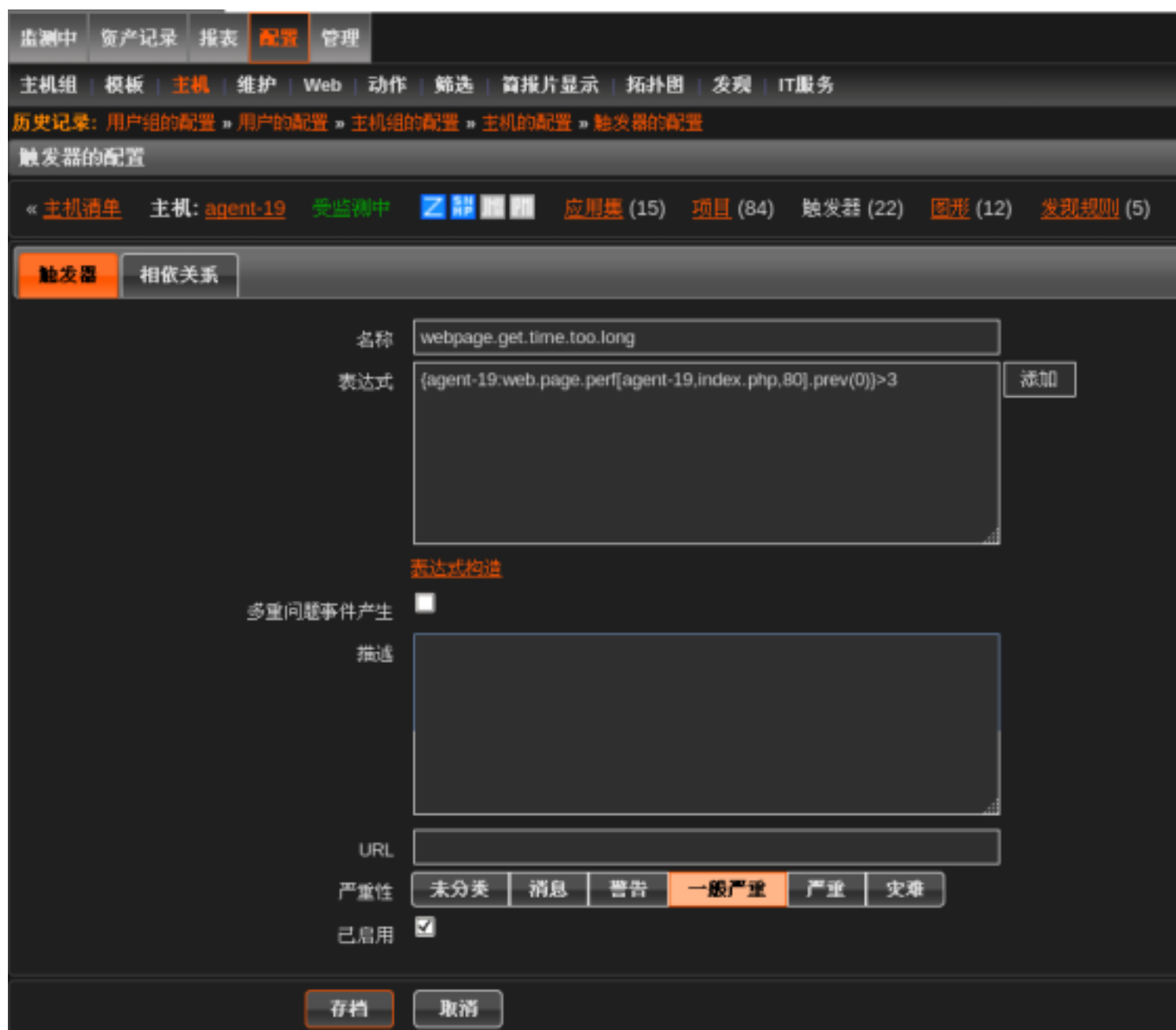
点击表达式右侧的添加按钮：



再点击项目右侧的**选择**,选择我们之前配置过的“web.server.online.monitor”,并设置触发的阈值,如下图所示



Zabbix 会自动生成表达式。接下来根据情况选择事件的严重性。配置完毕后,点击存档保存。



动作的配置:

点击:配置->动作->事件源下拉菜单中选择触发器->创建动作

可以在内容中使用 Zabbix 内置宏,邮件发出时会自动将宏替换成对应的值。

名称：
任意写

默认接收人：

故障级别：{TRIGGER.STATUS}。服务器：【{HOSTNAME1}】 发生：{TRIGGER.NAME} 故障！

注：默认接收人：相当于邮件的主题

默认信息：邮件的主题

告警主机：{HOSTNAME1}

告警时间：{EVENT.DATE} {EVENT.TIME}

告警等级：{TRIGGER.SEVERITY}

告警信息：{TRIGGER.NAME}

告警项目：{TRIGGER.KEY1}

问题详情：{ITEM.NAME}：{ITEM.VALUE}

当前状态：{TRIGGER.STATUS}：{ITEM.VALUE1}

事件ID：{EVENT.ID}

恢复邮件：

恢复主题：

服务器：【{HOSTNAME1}】故障已恢复。故障原因：{TRIGGER.NAME}

恢复信息：恢复邮件的正文。当故障恢复正常后也发邮件通知一下。

The screenshot shows the Zabbix web interface at the URL 192.168.245.3/zabbix/actionconf.php?form=update&actionid=7. The page title is 'ZABBIX' and the navigation bar includes '监测中', '资产记录', '报表', '配置', and '管理'. The main menu has '主机群组', '模板', '主机', '维护', '动作', '关联项事件', '自动发现', and '服务'. The '动作' (Action) tab is selected, and the sub-tab '动作' (Action) is active. The action name is 'web1-httpd-index-monitor'. The calculation method is '与/或' (A and B). The conditions are listed in a table:

标签	名称	动作
A	维护状态 非在 维护	移除
B	触发器 = web1: web1-index-baojing	移除

Below the table, there is a section for '新的触发条件' (New trigger condition) with a dropdown for '触发器名称' (Trigger name) and a dropdown for '似' (Like). There is a '添加' (Add) button. At the bottom, there is a checkbox for '已启用' (Enabled) which is checked. There are buttons for '更新' (Update), '克隆' (Clone), '删除' (Delete), and '取消' (Cancel). A '调试' (Debug) button is in the bottom right corner.

点击:操作->编辑：

192.168.245.3/zabbix/actionconf.php?form=update&actionid=7

ZABBIX 监测中 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

动作

动作 操作 恢复操作 确认操作

默认操作步骤持续时间: 1m

默认接收人: Problem: {TRIGGER.NAME}

默认信息: Problem started at {EVENT.TIME} on {EVENT.DATE}
 Problem name: {TRIGGER.NAME}
 Host: {HOSTNAME}
 Severity: {TRIGGER.SEVERITY}
 Original problem ID: {EVENT.ID}
 {TRIGGER.URL}

维护期间暂停操作: ☒

操作	步骤	细节	开始于	持续时间	动作
1 - 10	发送消息给用户: Admin (Zabbix Administrator)	通过 sendmail	立即地	60	编辑 移除
	发送消息给用户群组: Zabbix administrators	通过 sendmail			

[新的](#)

[更新](#) [克隆](#) [删除](#) [取消](#)

[调试](#)

发送间隔：60秒

步骤：发送10次发送到：admin用户

仅使用：sendmail方式发送 ----脚本。 方式可以自行设置，根据实际工作要求

维护期间暂停操作 ☒

操作

步骤 细节

1 - 10 发送消息给用户: Admin (Zabbix Administrator) 通过 sendmail 立即地 60 [编辑](#) [移除](#)

发送消息给用户群组: Zabbix administrators 通过 sendmail

操作细节

步骤

1 10 (0 - 无约束)

步骤持续时间

60 (0 - 使用默认)

操作类型

发送消息

发送到用户群组

用户群组

Zabbix administrators [移除](#)

[添加](#)

发送到用户

用户

Admin (Zabbix Administrator) [移除](#)

[添加](#)

仅送到

sendmail

默认信息

☒

条件

标签 名称 动作

A 事件已确认 = 非确认 [移除](#)

[新的](#)

[更新](#) [取消](#)

[更新](#) [克隆](#) [删除](#) [取消](#)

需要特别解释一下的是“步骤”部分的配置。所谓步骤是指报警可以有多个步骤,做不同的报警。例如,自从 1 到 3,就是指报警的步骤有三个。步骤持续时间就是一定时间后如果监控人员仍未响应报警就进入下一个报警步骤。

例如,发邮件给你报警,如果60 秒后你没响应,那就发 jabber 信息提醒你。如果 60 秒后还没响应,那就发短信给你。要是还没响应,就没有然后了。你可以形象的把它理解为 Zabbix 的一哭二闹三上吊。

到此,一个邮件报警功能就配置**完毕**了。如果你想立即看到结果,可以修改触发器的条件,将条件的阈值设置为 $N > 0.0003$ 。你马上就会收到 Zabbix 发来的报警邮件了。

微信报警

例子：

准备工作：（实际工作中是你实际公司）

企业ID（corpID）：

ww97aad3e2f93ef227

应用的corpsecret：jQOAtXJbV5_iVnEwA22rSqAtss9vn74wabMwDvJzrWY

AgentID：

1000003

（你所在的部门）云计算部门：

部门id：2

1.大部分操作在PC端后台完成（手机端不行）

1.下载企业微信

2.注册企业

3.创建自己的应用

- 4.添加员工
- 5.添加管理组
- 6.添加部门
- 7.测试企业微信接口调试工具



企业微信接口调试工具

此工具旨在帮助开发者检测调用【企业微信API】时发送的请求参数是否正确，提交相关信息后可获得服务器的验证结果
使用说明：

> 你可以直接在文本框内填入对应的参数值（红色星号表示该字段必填），点击检查问题按钮，即可得到相应的调试信息。
其他工具：

> jsapi_ticket验证工具：<http://work.weixin.qq.com/api/jsapisign>

> jsapi demo演示地址(企业微信里面访问)：<http://work.weixin.qq.com/api/jsapidemo>

一、接口类型 建立连接

二、接口列表 获取AccessToken 方法：GET

三、参数列表

*corpid

公司的id

*corpsecret

企业应用secret

测试：

建立连接：获取AccessToken

请求地址：[https://qyapi.weixin.qq.com/cgi-bin/gettoken?](https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=wwa1da242f211cdf47&corpsecret=rIrN51FEjgW6T2bdmSaEoN8s_mMW-Nu7gHysmagfdXc)

[corpid=wwa1da242f211cdf47&corpsecret=rIrN51FEjgW6T2bdmSaEoN8s_mMW-Nu7gHysmagfdXc](https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpid=wwa1da242f211cdf47&corpsecret=rIrN51FEjgW6T2bdmSaEoN8s_mMW-Nu7gHysmagfdXc)

返回结果：HTTP/1.1 200 OK

Connection: keep-alive

Error-Code: 0

Error-Msg: ok

Content-Type: application/json; charset=UTF-8

Content-Length: 277

```
{"errcode":0,"errmsg":"ok","access_token":"9O7DSEqnUsQ8MD2bRGfFs26cabY1XcGEPnnJF_mfBBnI5zsM2I3u9-9Sj0tA  
ggY6BahOByGZvzDNaFoJj3YTp5blSPkm9exizerPUpDZNiGXtw","expires_in":7200}
```

测试好之后可以配置监控脚本了。

python或者shell的都可以

除了执行的方式不一样，其他都一样。

python监控脚本：

有4个地方需要修改，找到他们修改成之前获取的信息

`self.__corpid = 'wwa1da242f211cdf47' //公司的corpid`

`self.__secret = 'rIrN51FEjgW6T2bdmSaEoOIB5UbaBTgddQkuC8cca1w' //应用的secret`

`'toparty':2, //部门id`

```
'agentid':"1000002", //应用id
```

以python的脚本为例：

```
[root@ansible alertscripts]# cat [root@ansible alertscripts]# cat wechat.py
```

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```
import urllib,urllib2,json
import sys
reload(sys)
sys.setdefaultencoding( "utf-8" )
```

```
class WeChat(object):
    __token_id = ""
    # init attribute
    def __init__(self,url):
        self.__url = url.rstrip('/')
        self.__corpid = 'wwa1da242f211cdf47' -----需要修改成你自己的
        self.__secret = 'rIrN51FEjgW6T2bdmSaEoOIB5UbaBTgddQkuC8cca1w' -----需要修改成你自己的

    # Get TokenID
    def authID(self):
        params = {'corpid':self.__corpid, 'corpsecret':self.__secret}
        data = urllib.urlencode(params)

        content = self.getToken(data)

        try:
            self.__token_id = content['access_token']
            # print content['access_token']
        except KeyError:
            raise KeyError

    # Establish a connection
    def getToken(self,data,url_prefix='/'):
        url = self.__url + url_prefix + 'gettoken?'
        try:
            response = urllib2.Request(url + data)
        except KeyError:
            raise KeyError
        result = urllib2.urlopen(response)
        content = json.loads(result.read())
        return content

    # Get sendmessage url
    def postData(self,data,url_prefix='/'):
        url = self.__url + url_prefix + 'message/send?access_token=%s' % self.__token_id
        request = urllib2.Request(url,data)
        try:
            result = urllib2.urlopen(request)
        except urllib2.HTTPError as e:
            if hasattr(e,'reason'):
                print 'reason',e.reason
```

```

        elif hasattr(e,'code'):
            print 'code',e.code
        return 0
    else:
        content = json.loads(result.read())
        result.close()
    return content

# send message
def sendMessage(self,touser,message):
    self.authID()
    data = json.dumps({
        'touser':touser,
        'toparty':2, -----需要修改成你自己的
        'msgtype':"text",
        'agentid':"1000002", -----需要修改成你自己的
        'text':{
            'content':message
        },
        'safe':"0"
    },ensure_ascii=False)

    response = self.postData(data)
    print response

if __name__ == '__main__':
    a = WeChat('https://qyapi.weixin.qq.com/cgi-bin')
    a.sendMessage(sys.argv[1],sys.argv[3])

```

修改权限：

```

[root@ansible alertscripts]# chown zabbix.zabbix /usr/lib/zabbix/alertscripts/wechat.py
[root@ansible alertscripts]# chmod 777 /usr/lib/zabbix/alertscripts/wechat.py

```

测试脚本：下面的invalidparty可以忽略，如果正常会收到消息如下图

```

                                添加的用户 标题 内容
[root@ansible alertscripts]# ./wechat.py wusong test test
{u'invalidparty': u'2', u'invaliduser': u'wusong', u'errcode': 0, u'errmsg': u'ok'}

```



zabbix配置微信报警：

与邮箱配置的过程和应用的内容一样。

创建报警媒介：

管理->报警媒介类型->创建媒体类型



zabbix 3.0 需要在为alert脚本定义参数，以前的版本参数都固定的，现在用户可以自己定义命令行的参数了。添加的参数也就是脚本中的\$1、\$2、\$3等。

2.

报警媒介类型

报警媒介类型

选项

名称

微信报警

类型

脚本

脚本名称

wechat.py

脚本参数

参数

{ALERT.SENDTO}

{ALERT.SUBJECT}

{ALERT.MESSAGE}

添加

动作

移除

移除

移除

已启用

☒

更新

克隆

删除

取消

添加完成后，需要关联到报警用户Administration-->Users-->Media-->add
配置收件人：

注意填写的收件人是：企业微信内的员工帐号，
脚本参数和用邮件报警添加的一样

用户

用户

报警媒介

权限

报警媒介	类型	收件人	当启用时	如果存在严重性则使用	Status	动作
	微信报警	Tigerfive	1-7,00:00-24:00	未信警一严灾	已启用	编辑 移除
	微信报警	YanQiang	1-7,00:00-24:00	未信警一严灾	已启用	编辑 移除
	添加					

更新

删除

取消

报警媒介

类型	微信报警
收件人	YanQiang
当启用时	1-7,00:00-24:00
如果存在严重性则使用	<input checked="" type="checkbox"/> 未分类 <input checked="" type="checkbox"/> 信息 <input checked="" type="checkbox"/> 警告 <input checked="" type="checkbox"/> 一般严重 <input checked="" type="checkbox"/> 严重 <input checked="" type="checkbox"/> 灾难
已启用	<input checked="" type="checkbox"/>
<input type="button" value="更新"/> <input type="button" value="取消"/>	

添加动作，触发条件后报警后发送微信消息。

Configuration-->Actions-Event source(Triggers)-Create action

ZABBIX

监测中 资产记录 报表 配置 管理

主机群组 模板 主机 维护 动作 关联项事件 自动发现 服务

动作

动作 操作 恢复操作 确认操作

名称 web1-httpd-index-monitor

计算方式 与/或 A and B

条件	标签	名称	动作
A		维护状态 非在 维护	移除
B		触发器 = web1: web1-index-baojing	移除

新的触发条件

触发器名称

似

[添加](#)已启用 ☒

动作

动作 操作 恢复操作 确认操作

默认操作步骤持续时间

1m

默认接收人

服务器:{HOST.NAME}发生: {TRIGGER.NAME}故障!

默认信息

告警主机:{HOST.NAME}
告警地址:{HOST.IP}
监控项目:{ITEM.NAME}
监控取值:{ITEM.LASTVALUE}
告警等级:{TRIGGER.SEVERITY}
当前状态:{TRIGGER.STATUS}
告警信息:{TRIGGER.NAME}

维护期间暂停操作 ☒

操作

步骤 细节

开始于 持续时间 动作

1 - 10 发送消息给用户: Admin (Zabbix Administrator) 通过 微信报警 立即地 60

[编辑](#) [移除](#)

发送消息给用户群组: Zabbix administrators 通过 微信报警

操作细节

步骤 1 - 10 (0 - 无穷大)

步骤持续时间 60 (0 - 使用默认)

操作类型 发送消息

发送到用户群组

用户群组

动作

Zabbix administrators

[移除](#)[添加](#)

发送到用户

用户

动作

Admin (Zabbix Administrator)

[移除](#)[添加](#)

仅送到 微信报警

默认信息 ☒

条件

标签

名称

动作

A

事件已确认 = 非确认

[移除](#)[新的](#)[更新](#) [取消](#)[更新](#)[克隆](#)[删除](#)[取消](#)

恢复操作和确认操作都可以配置，同上。

