目录

[第一章：zabbix简介 2](#_Toc534156571)

[1、安装zabbix 2](#_Toc534156572)

[1.1、安装mariadb并启动 2](#_Toc534156573)

[1.2、安装zabbix3.4 2](#_Toc534156574)

[1.3、设置mariadb 2](#_Toc534156575)

[1.4、导入数据库 2](#_Toc534156576)

[1.5、配置数据库用户及密码 2](#_Toc534156577)

[1.6、启动zabbix server并设置开机启动 3](#_Toc534156578)

[1.7、设置zabbix web 4](#_Toc534156579)

[1.8、登陆zabbix监控平台 8](#_Toc534156580)

[2、zabbix-agent安装及配置 10](#_Toc534156581)

[2.1、linux配置zabbix-agent 10](#_Toc534156582)

[2.2、Windows配置zabbix-agent 11](#_Toc534156583)

[3、配置监控主机 12](#_Toc534156584)

[第二章：监控常用程序 14](#_Toc534156585)

[1、基本监控项配置 14](#_Toc534156586)

[1.1、定义宏监控端口和进程 14](#_Toc534156587)

[1.2、ping监控 17](#_Toc534156588)

[1.3、Telnet监控 17](#_Toc534156589)

[1.4、监控网络设备 18](#_Toc534156590)

[2、监控tomcat 18](#_Toc534156591)

[3、监控mysql 18](#_Toc534156592)

[3.1、监控mysql.5.7版本 18](#_Toc534156593)

[3.2、其他监控参数 22](#_Toc534156594)

[3.3、监控主从状态 23](#_Toc534156595)

[4、监控nginx 25](#_Toc534156596)

[5、web页面URL监控 27](#_Toc534156597)

[6、监控API接口性能及可用性 31](#_Toc534156598)

[7、监控日志关键字 31](#_Toc534156599)

[12.10、告警发送QQ和微信消息 35](#_Toc534156600)

[12.11、告警发送邮件 42](#_Toc534156601)

[12.12、语音/短信告警 48](#_Toc534156602)

[12.13、zabbix+grafana实现可视化监控图 48](#_Toc534156603)

[12.14、zabbix监控redis 48](#_Toc534156604)

可参考：<https://mp.weixin.qq.com/s/kvVctQdJnpwrbzfNm4cVMQ>

zabbix是一个基于WEB界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案。

**zabbix**由2部分构成，**zabbix server**与**zabbix agent**。

**zabbix server**可以单独监视远程服务器的服务状态；同时也可以与**zabbix agent**配合，可以轮询**zabbix agent**主动接收监视数据（trapping方式），同时还可被动接收**zabbix agent**发送的数据（trapping方式）。  
另外**zabbix server**还支持SNMP (v1,v2)，可以与SNMP软件(例如：net-snmp)等配合使用。

**zabbix agent**需要安装在被监视的目标服务器上，它主要完成对硬件信息或与操作系统有关的内存，CPU等信息的收集。

相关软件下载地址：http://repo.zabbix.com/zabbix/3.4/rhel/7/x86\_64/

# 第一章：zabbix简介

## 1、安装zabbix

systemctl stop firewalld.service #关闭防火墙

systemctl disable firewalld.service #开机不启动防火墙

**禁用selinux**

vim /etc/selinux/config

将SELINUX=enforcing改为SELINUX=disabled 保存后退出 重启机器后在安装zabbix

### 1.1、安装mariadb并启动

yum install mariadb-server mariadb –y

systemctl start mariadb #启动MariaDB

systemctl stop mariadb #停止MariaDB

systemctl restart mariadb #重启MariaDB

systemctl enable mariadb #设置开机启动

### 1.2、安装zabbix3.4

rpm -ivh http://repo.zabbix.com/zabbix/3.4/rhel/7/x86\_64/zabbix-release-3.4-2.el7.noarch.rpm

yum install zabbix-server-mysql zabbix-web-mysql -y

yum install zabbix-agent zabbix-get openssl openssl-devel -y

### 1.3、设置mariadb

mysql #进入mariadb

create database zabbix character set utf8 collate utf8\_bin; #创建zabbix数据库

grant all privileges on zabbix.\* to zabbix@localhost identified by 'zabbix'; #创建用户并授权登陆

grant all privileges on zabbix.\* to zabbix@'%' identified by 'zabbix';

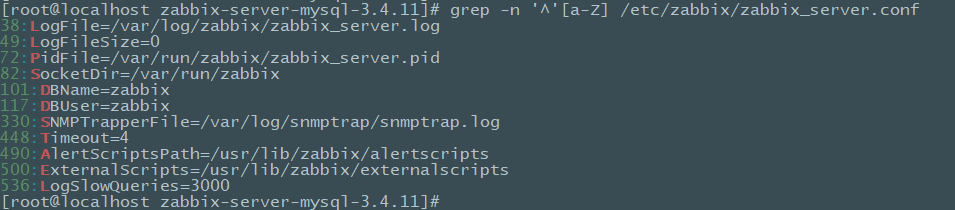
exit #退出

### 1.4、导入数据库

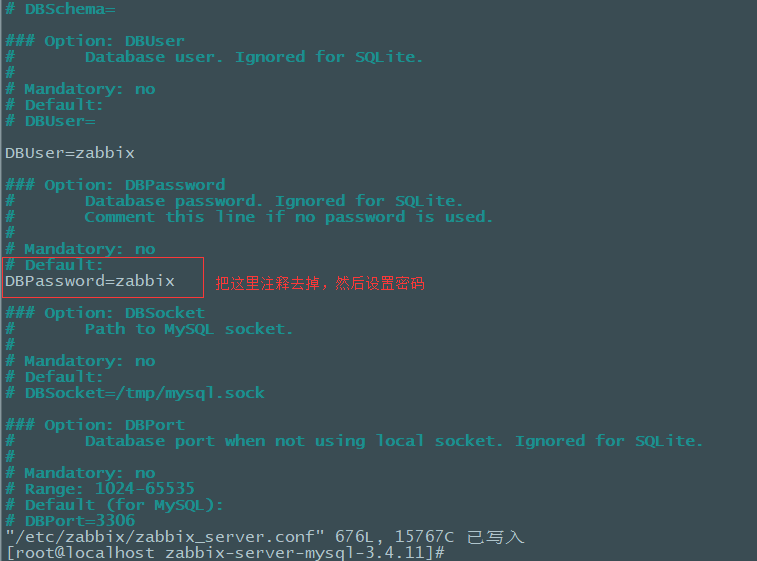
zcat /usr/share/doc/zabbix-server-mysql-3.4.11/create.sql.gz |mysql -uzabbix -pzabbix zabbix

### 1.5、配置数据库用户及密码

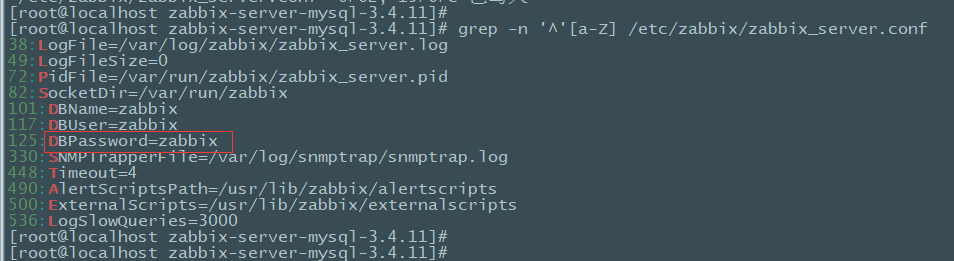
检查配置grep -n '^'[a-Z] /etc/zabbix/zabbix\_server.conf



vim /etc/zabbix/zabbix\_server.conf #修改配置文件添加用户和密码



在查看多了密码配置



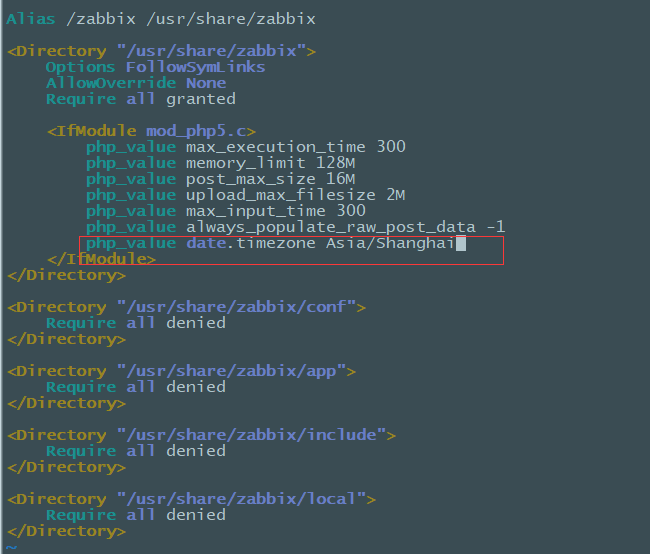
### 1.6、启动zabbix server并设置开机启动

systemctl start zabbix-server

systemctl enable zabbix-server

编辑Zabbix前端PHP配置,更改时区修改为亚洲上海

vim /etc/httpd/conf.d/zabbix.conf



**设置selinux**

setsebool -P httpd\_can\_connect\_zabbix on

setsebool -P httpd\_can\_cetwork\_connect\_db on

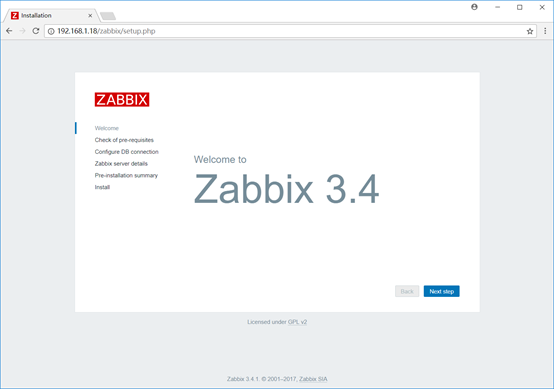
**启动httpd并设置开机启动**

systemctl start httpd #启动httpd服务

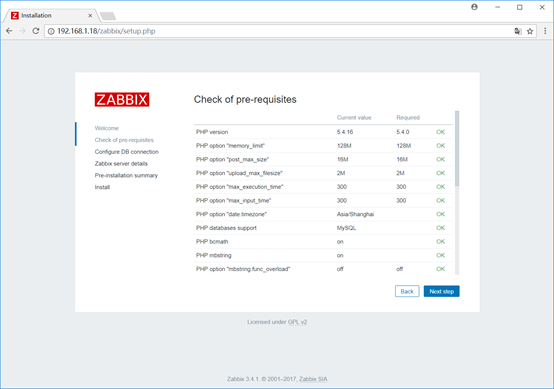
systemctl enable httpd #设置开机启动httpd服务

### 1.7、设置zabbix web

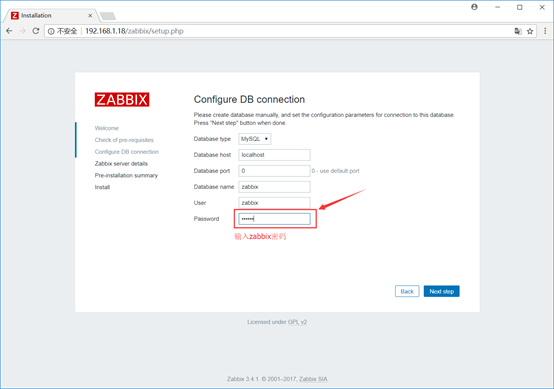
使用浏览器访问zabbix web



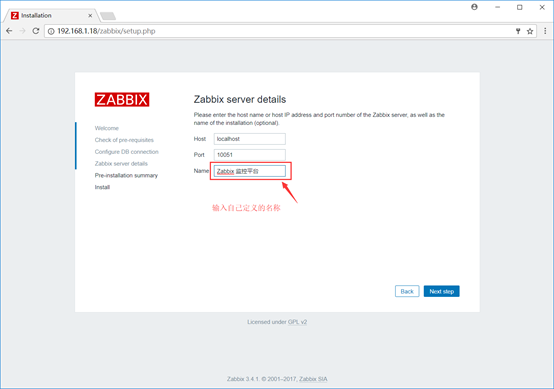
点击next step



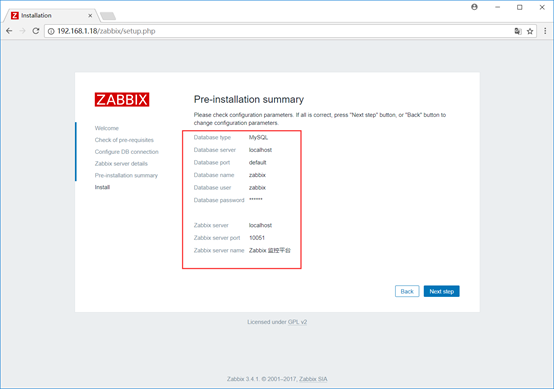
点击 next step



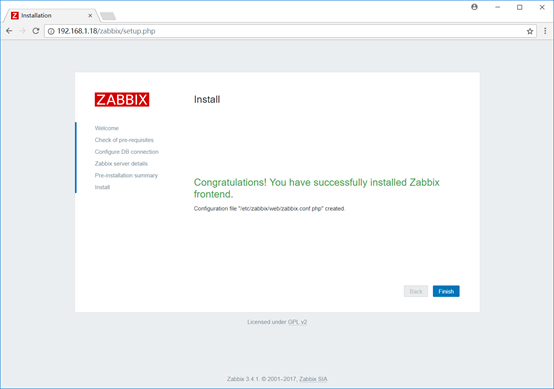
**输入数据库的密码,然后**点击next step



输入名称，然后点击 next step



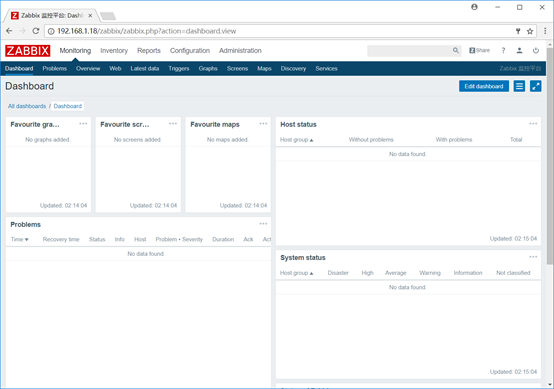
确认信息没错以后,点击 next step



完成安装，会将在/etc/zabbix/web/zabbix.conf.php生成配置文件

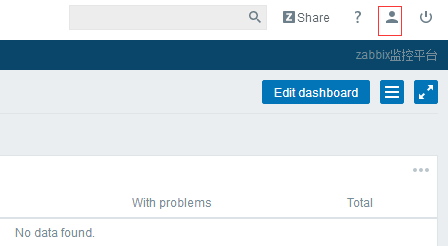
### 1.8、登陆zabbix监控平台

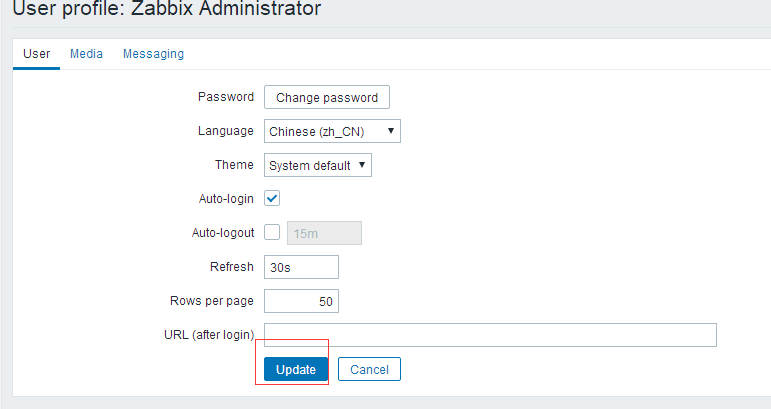
登录最新版Zabbix3.4 默认用户Admin 默认密码zabbix,登陆后界面如下



**设置中文**

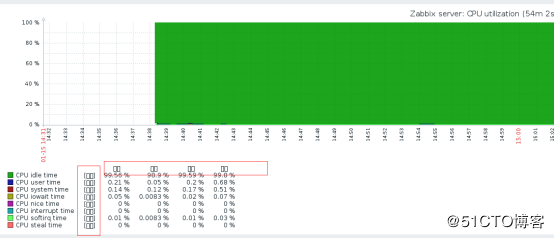
点击用户头像,然后有语言选择中文。



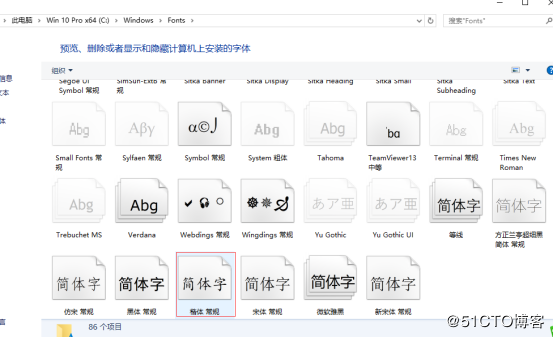


**解决中文在图形界面上的乱码**

一般情况下还是会出现中文乱码的情况



因为zabbix自身对中文简体的支持不完善，需要我们手动的去上传新的字体进行替换：在C:\Windows\Fonts中复制想要的字体，后缀为ttf，把文件复制到桌面。



上传至zabbix服务器的/usr/share/zabbix/fonts 目录中,记得要备份原来的字体,

mv graphfont.ttf  graphfont.ttf.bak #备份原来的字体

mv simkai.ttf graphfont.ttf #把上传的字体改成原来字体的名称

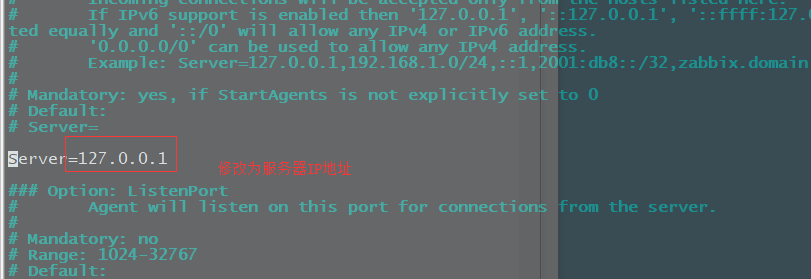
服务端到此安装完成，下面安装都是客户端

## 2、zabbix-agent安装及配置

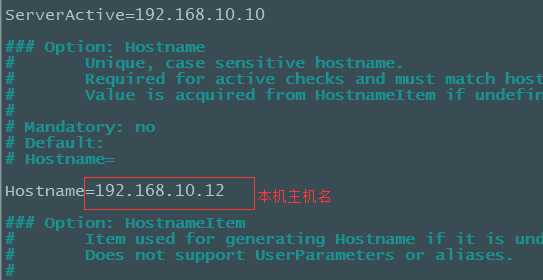
安装命令：yum install zabbix-agent -y

### 2.1、linux配置zabbix-agent

vim /etc/zabbix/zabbix\_agentd.conf







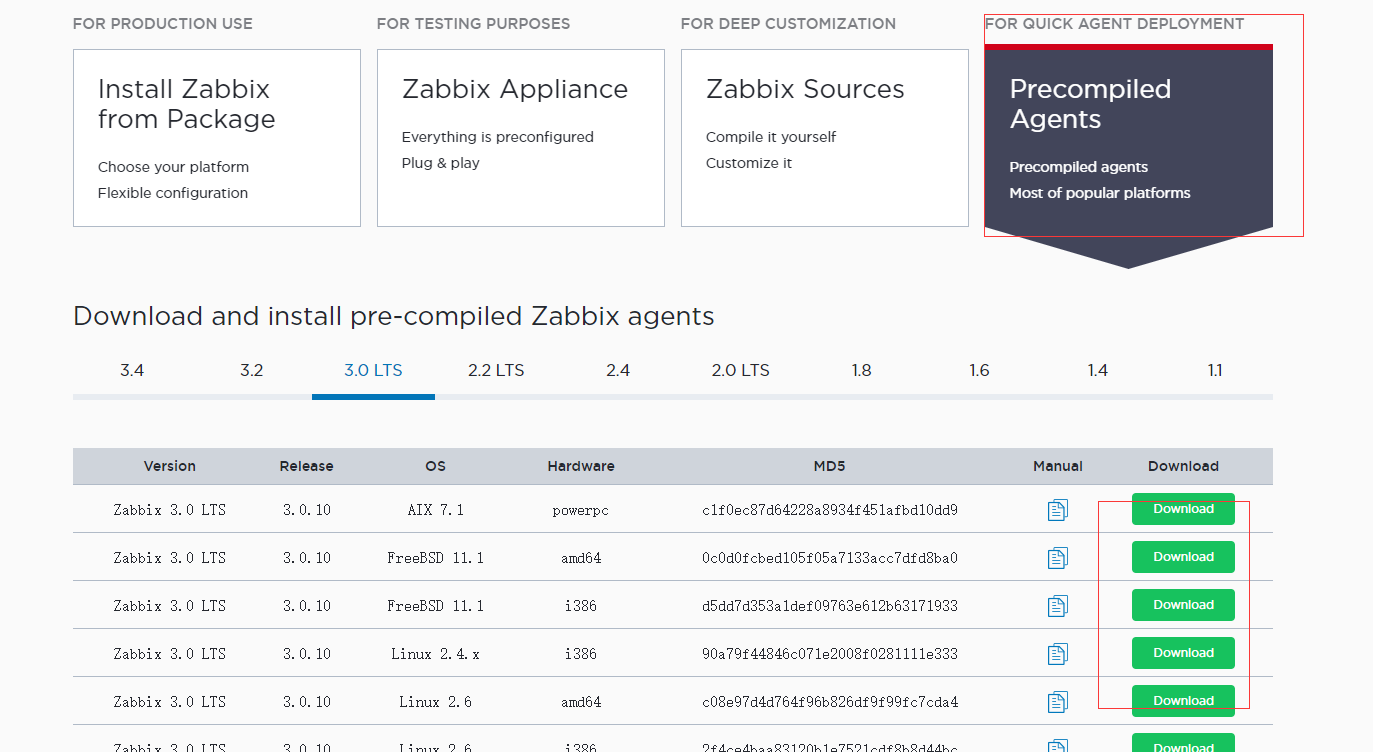
**然后启动zabbix-agent服务**

systemctl restart zabbix-agent.service

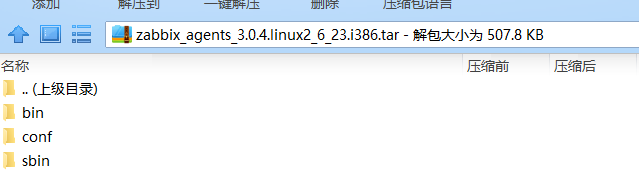
systemctl enable zabbix-agent.service

**第二种方式编译包安装：**

下载程序包：https://www.zabbix.com/download\_agents



下载后又下面这些目录



放到Linux机器上，创建zabbix用户组

groupadd zabbix

useradd -g zabbix zabbix

参考上面配置zabbix-agent方式，然后启动

/opt/zabbix-agent/sbin/zabbix\_agentd -c /opt/zabbix-agent/conf/zabbix\_agentd.conf

如果启动报错缺少 /lib/ld-linux.so.2 是因为64系统安装了32位程序 yum install glibc.i686 可以解决，如果还有报错缺少libstdc++.so.6通过yum安装yum install libstdc++.so.6 可解决

设置开启自启动

vim /etc/rc.local

/opt/zabbix-agent/sbin/zabbix\_agentd -c /opt/zabbix-agent/conf/zabbix\_agentd.conf

### 2.2、Windows配置zabbix-agent

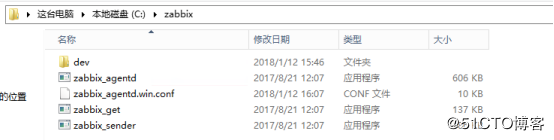
关闭windows防火墙或防火墙入放行zabbix\_agentd客户端口号10050 (TPC/UDP)。

下载后解压zabbix\_agents\_3.4.0.win.zip压缩包。里面有两个文件夹，一个是bin文件夹，另一个是conf文件夹。

**Bin文件夹里面有两个文件夹**，一个是win32文件夹里存放zabbix\_agentd安装程序应用于windows 32位操作系统，一个是win64文件夹里存放zabbix\_agentd安装程序应用于windows 64位操作系统。

**Conf文件夹里存放是配置文件zabbix\_agentd.win.conf**

在windows操作系统下， C盘目录下创建一个zabbix文件夹，把刚下载的zabbix\_agentd压缩包里的win64位文件夹复制到zabbix文件夹里。把conf文件夹zabbix\_agentd.win.conf复制到新创的zabbix目录下。



右键以文本格式编辑zabbix\_agentd.win.conf 配置文件，使用Notrpad++编辑,修改下面几项

EnableRemoteCommands=1           #允许在本地执行远程命令

LogRemoteCommands=1               #执行远程命令是否保存操作日志

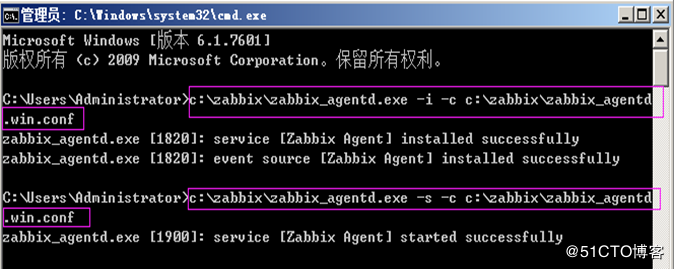
Server=192.168.3.50                       #填写zabbix-server服务器IP地址

ServerActive=192.168.3.50             #填写zabbix-server服务器IP地址

Hostname=server2012                    #zabbix\_agent客户端计算机名 (被监控主机)

打开DOS命令窗口---- 输入以下两条命令进行zabbix客户端安装。(必须要以管理员身份运行打开DOS命令窗口)

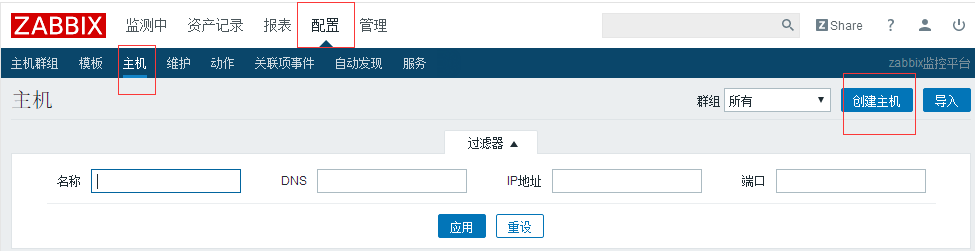
C:\zabbix\zabbix\_agentd.exe -i -c C:\zabbix\zabbix\_agentd.win.conf #安装zabbix客户端C:\zabbix\zabbix\_agentd.exe -s -c C:\zabbix\zabbix\_agentd.win.conf #启动zabbix服务



## 3、配置监控主机

登陆zabbix web端操作

选择配置 --🡪 主机 --🡪 创建主机。



输入客户端计算机名 -🡪 可见名称自定义 --🡪 群组自行选择 --🡪 输入客户端计算IP地址 --🡪 勾选已启用 --🡪选择添加。

添加 zabbix\_agentd 客户端监控模版。





## 4、分布式监控部署

# 第二章：监控常用程序

## 1、基本监控项配置

### 1.1、定义宏监控端口和进程

创建模板然后通过宏变量配置监控项，那么什么是宏?

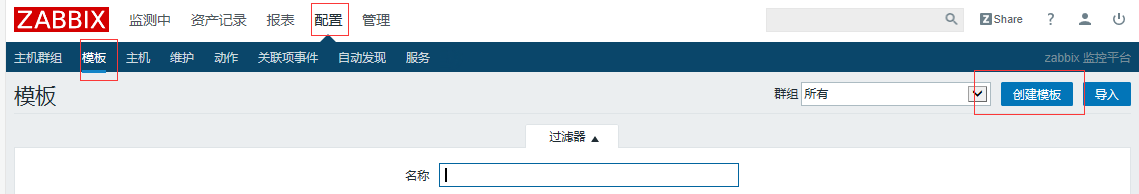
宏是一种抽象，根据一系列预定义的规则替换一定的文本模式，而解释器或编译器在遇到宏时会自动进行这一模式替换，可以理解为变量。zabbix有许多内置的宏，如{HOST.NAME}、{HOST.IP}、{TRIGGER.DESCRIPTION}、{TRIGGER.NAME}、{TRIGGER.EVENTS.ACK}等。

为了更强的灵活性，zabbix支持全局、模版或主机级别自定义宏，用户自定义宏要使用”{$MACRO}”这种特殊的语法格式，宏的名称只能使用大写字母、数字及下划线。

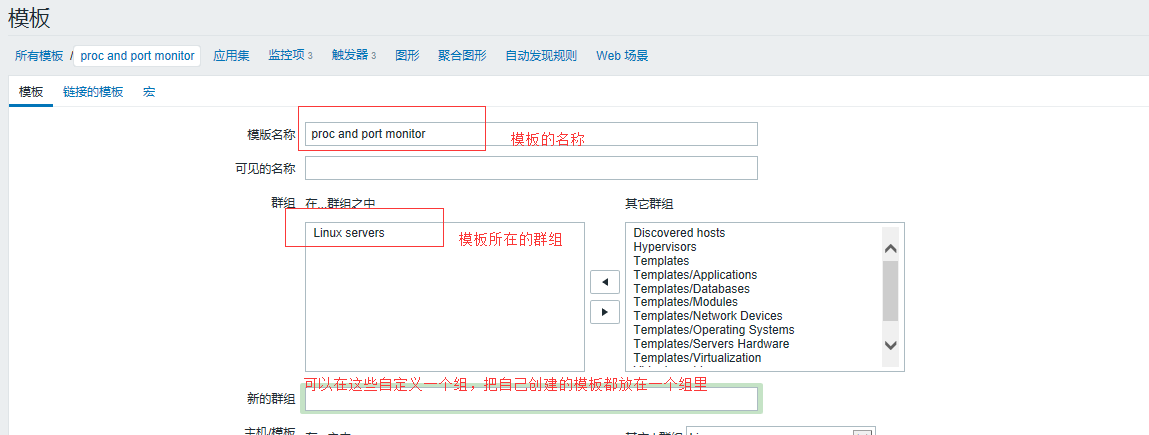
宏可以应用在item keys和descriptions、trigger名称和表达 式、主机接口IP/DNS及端口、discovery机制的SNMP协议 的相关信息中等。

配置参考网址：https://blog.csdn.net/wszll\_alex/article/details/77164892

**创建模板，安装图提示如下：**



点击创建模板，模板名称不能是中文，按照下图创建，创建好后然后点更新

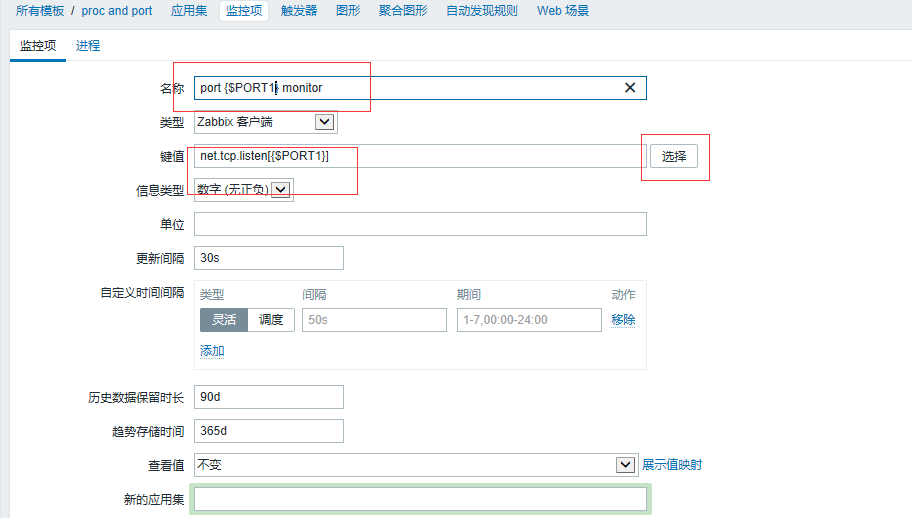


找到刚刚创建的模板，选中点进去



点击监控项，创建监控项,然后填写监控名称，{$PORT1} 是宏，监控的时候在赋值，有多个端口就加多些监控项，然后点击最下面的添加

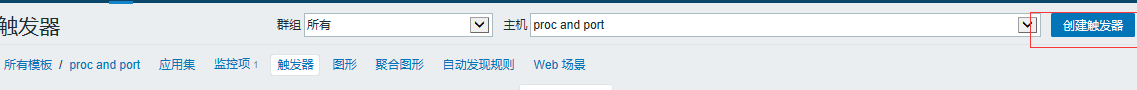




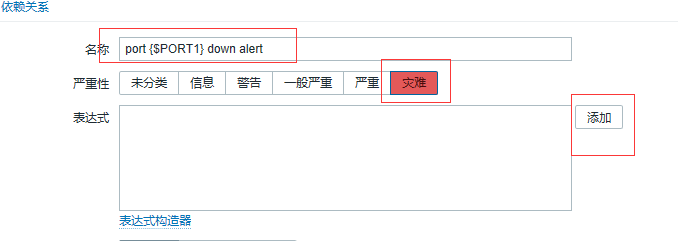
这里配置了2个端口的监控项和一个进程的监控项,监控什么进程可以把名字写成什么进程。



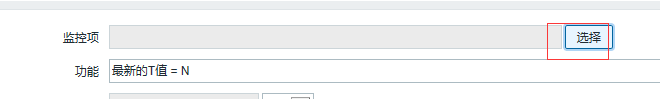
添加完成监控项后，然后配置触发器。



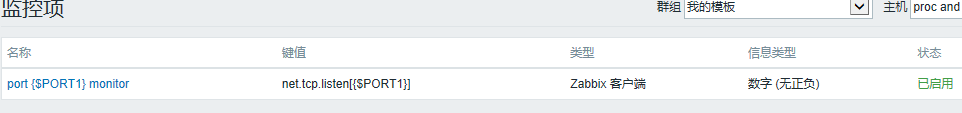
填写告警名称，告警的登记和表达式



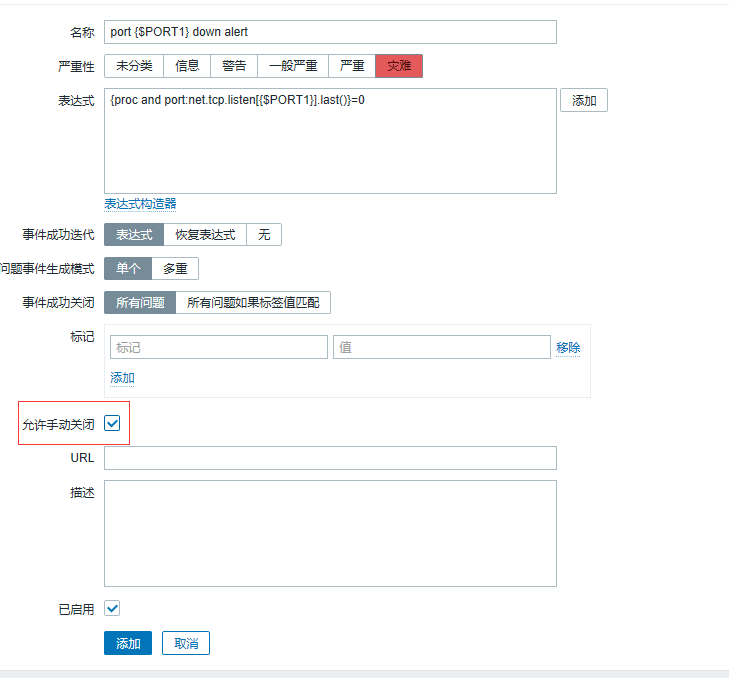
添加添加可以选择表达式



选择对应的监控项，然后点击插入



配置成效果如下，如果想要告警的时候可以手动关闭掉告警信息这里打上对号，然后点击添加



模板配置好后就可以在主机里应用模板，和添加主机里添加告警模板一样，选择我们刚刚创建的告警模板就可以了。



然后选择宏，填写上刚才创建的宏名称，对应好需要监控的进程和端口就可以了



### 1.2、ping监控

参考：https://blog.csdn.net/tianhua79658788/article/details/77930664

<https://www.cnblogs.com/ITflying/p/8004951.html>

**配置如下，直接定义KEY，然后填写检测命令**

vim /etc/zabbix/zabbix\_agentd.d/fping.conf # 返回0表示正常，返回1 表示异常

UserParameter=pingsys,ping -c 1 192.168.10.254 |grep "Destination Host Unreachable"| wc -l

然后在zabbix服务端验证,返回正常

zabbix\_get -s 192.168.10.12 -p 10050 -k pingsys

最后在zabbix-web添加监控项。

### 1.3、Telnet监控

参考：https://www.cnblogs.com/loki717/p/7991081.html

vim /etc/zabbix/zabbix\_agentd.d/telnet.conf #填写以下内容通返回1 通返回0

UserParameter=telnet.host,nmap 119.75.213.61 -p 4444 |grep open | wc -l

也可以写个脚本检测端口是否通，可以通过这个语句编写脚本

vim /etc/zabbix/script/telnet\_host.sh

#!/bin/bash

port\_stats=`nmap $1 -p $2 |grep open | wc -l`

echo "$port\_stats"

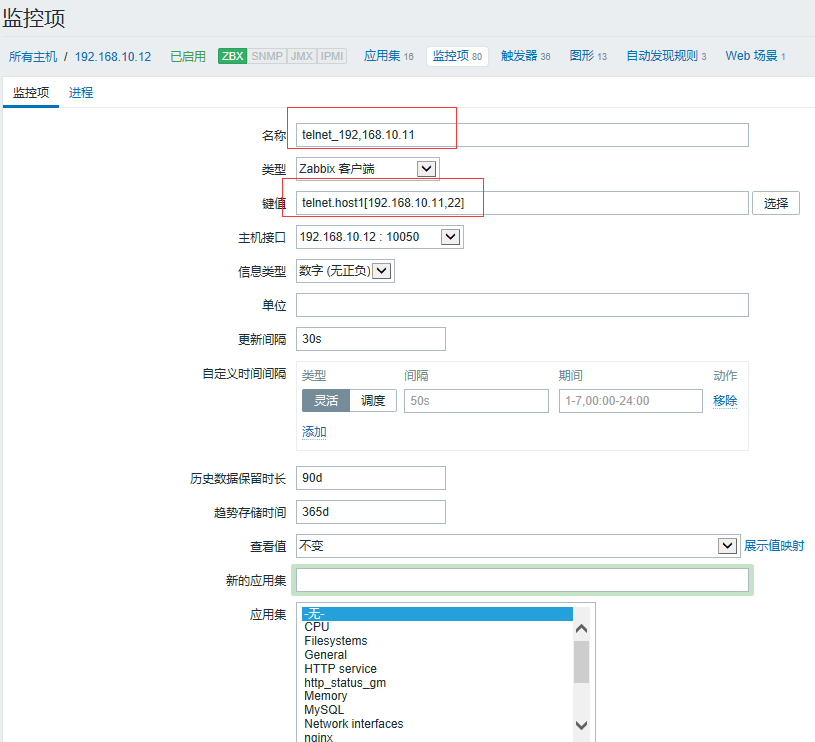
vim /etc/zabbix/zabbix\_agentd.d/telnet.conf 配置如下

UserParameter=telnet.host[\*],/etc/zabbix/script/telnet\_host.sh $1 $2

在zabbix服务器上验证，正常返回1，异常返回0

zabbix\_get -s 192.168.10.12 -p 10050 -k telnet.host1[192.168.10.11,22]

**添加监控项**



### 1.4、监控网络设备

参考：https://jingyan.baidu.com/article/4b52d702ce948ffc5c774bb0.html

网络设备监控模板：https://www.zabbix.org/wiki/Zabbix\_Templates

## 2、监控tomcat

参考链接：https://www.cnblogs.com/bugsbunny/p/7204011.html

https://www.jianshu.com/p/bfeda2bba15a

## 3、监控mysql

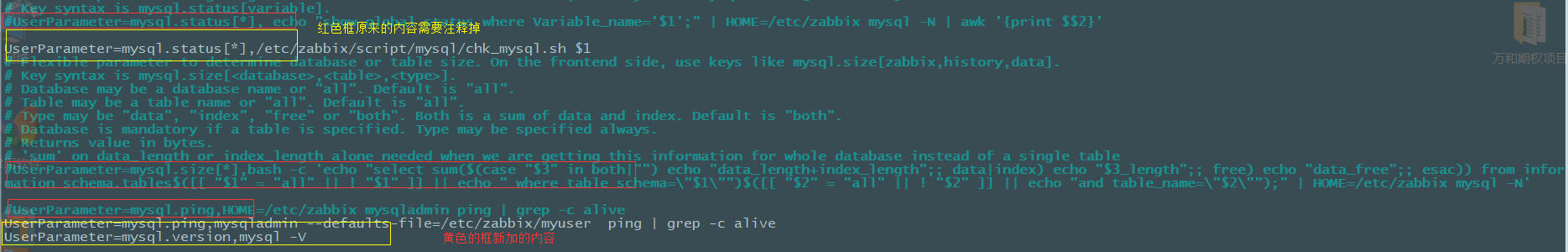
### **3.1、监控mysql.5.7版本**

**(使用自带的监控模板)**

参考链接http://blog.51cto.com/hequan/2055492?utm\_source=oschina-app

把默认的**userparameter\_mysql.conf替换下内容**

vim /etc/zabbix/zabbix\_agentd.d/userparameter\_mysql.conf



UserParameter=mysql.status[\*],/etc/zabbix/script/mysql/chk\_mysql.sh $1

UserParameter=mysql.ping,mysqladmin --defaults-file=/etc/zabbix/myuser ping | grep -c alive

UserParameter=mysql.version,mysql -V

vim /etc/zabbix/script/mysql/chk\_mysql.sh

# 用户名，要对fangx用户授权本地登陆权限

MYSQL\_USER='fangx'

# 密码

MYSQL\_PWD='Fan2018!'

# 主机地址/IP

MYSQL\_HOST='192.168.10.12'

# 端口

MYSQL\_PORT='3306'

export MYSQL\_PWD=Fan2018!

# 数据连接

MYSQL\_CONN="mysqladmin -u${MYSQL\_USER} -h${MYSQL\_HOST} -P${MYSQL\_PORT}"

# 参数是否正确

if [ $# -ne "1" ];then

echo "arg error!"

fi

# 获取数据

case $1 in

Uptime)

result=`${MYSQL\_CONN} status|cut -f2 -d":"|cut -f1 -d"T"`

echo $result

;;

Com\_update)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_update"|cut -d"|" -f3`

echo $result

;;

Slow\_queries)

result=`${MYSQL\_CONN} status |cut -f5 -d":"|cut -f1 -d"O"`

echo $result

;;

Com\_select)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_select"|cut -d"|" -f3`

echo $result

;;

Com\_rollback)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_rollback"|cut -d"|" -f3`

echo $result

;;

Questions)

result=`${MYSQL\_CONN} status|cut -f4 -d":"|cut -f1 -d"S"`

echo $result

;;

Com\_insert)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_insert"|cut -d"|" -f3`

echo $result

;;

Com\_delete)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_delete"|cut -d"|" -f3`

echo $result

;;

Com\_commit)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_commit"|cut -d"|" -f3`

echo $result

;;

Bytes\_sent)

result=`${MYSQL\_CONN} extended-status |grep -w "Bytes\_sent" |cut -d"|" -f3`

echo $result

;;

Bytes\_received)

result=`${MYSQL\_CONN} extended-status |grep -w "Bytes\_received" |cut -d"|" -f3`

echo $result

;;

Com\_begin)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_begin"|cut -d"|" -f3`

echo $result

;;

\*)

echo "Usage:$0(Uptime|Com\_update|Slow\_queries|Com\_select|Com\_rollback|Questions|Com\_insert|Com\_delete|Com\_commit|Bytes\_sent|Bytes\_received|Com\_begin)"

;;

esac



vim /etc/zabbix/myuser

[client]

host='localhost'

user='zabbix'

password='Fan2018!'

socket=/var/lib/mysql/mysql.sock

在zabbix-web端主机添加模板链接



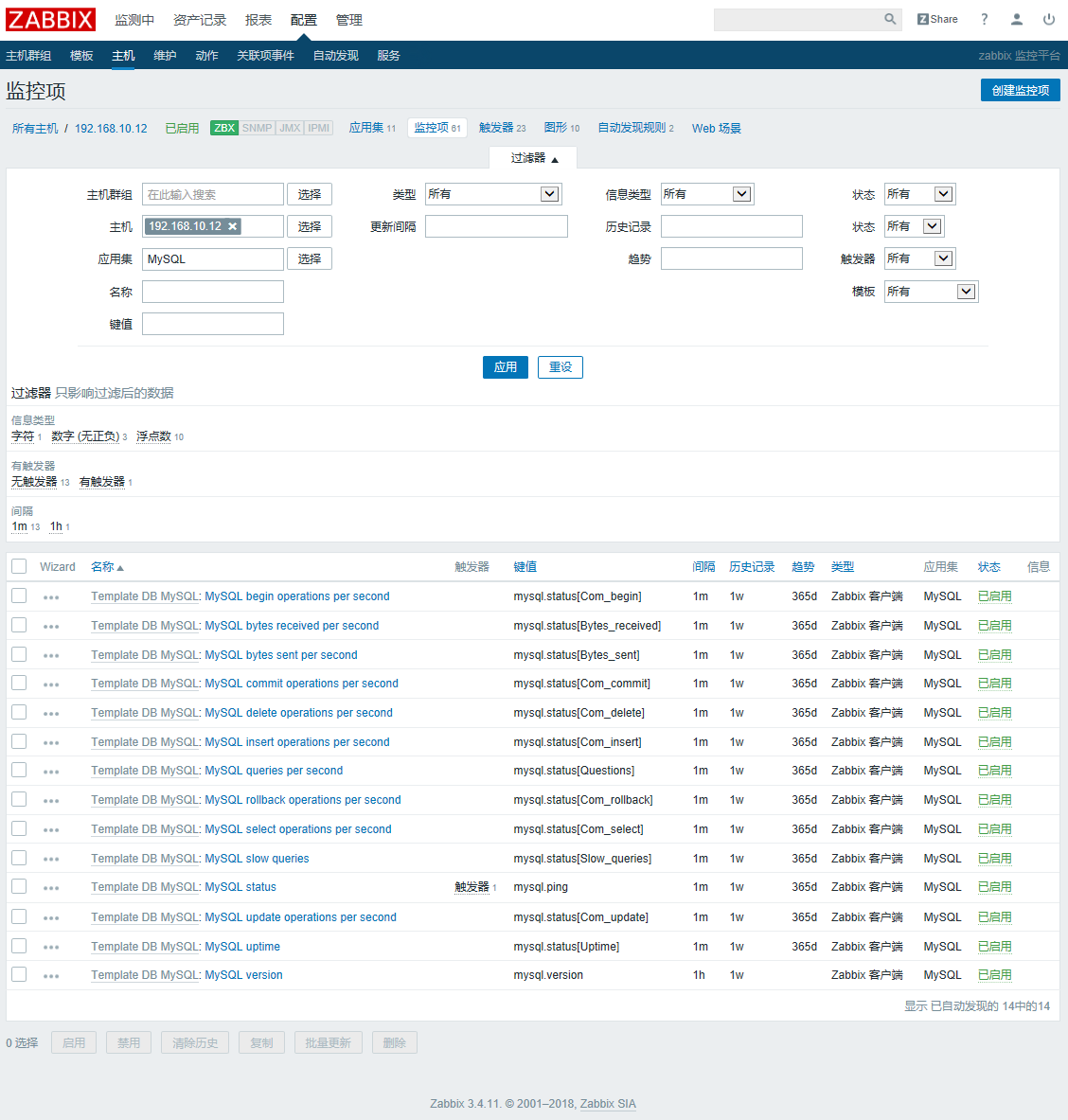
service zabbix-agent restart #重启agent

在zabbix服务器上测试：zabbix\_get -s 192.168.10.12 -p10050 -k mysql.ping#返回0表示失败,1表示正常

zabbix\_get -s 192.168.10.12 -p10050 -k "mysql.status[Uptime]" #返回正常数值

如有报错可能是selinux没有关闭的问题，需要把selinux关闭

zabbix正常显示如下，也可以查看监测的图形



### 3.2、其他监控方式

**(使用自带模板)**

参考：https://renwole.com/archives/549

vim /usr/local/zabbix/etc/zabbix\_agentd.conf

Include=/usr/local/zabbix/etc/zabbix\_agentd.conf.d/\*.conf

**mysql授权**

> GRANT SELECT ON \*.\* TO 'zabbix'@'localhost' IDENTIFIED BY 'Renwolecom';

> FLUSH PRIVILEGES;

> exit

$ vim /usr/local/zabbix/etc/.my.cnf

# Zabbix Agent

[mysql]

host=localhost

user=zabbix

password="Renwolecom"

socket=/tmp/mysql.sock

[mysqladmin]

host=localhost

user=zabbix

password="Renwolecom"

socket=/tmp/mysql.sock

**修改后的内容如下：**

$ cd /usr/local/zabbix/etc/zabbix\_agentd.conf.d/

$ egrep -v "(^#|^$)" userparameter\_mysql.conf

UserParameter=mysql.status[\*],echo "show global status where Variable\_name='$1';" | HOME=/usr/local/zabbix/etc/ mysql -N | awk '{print $$2}'

UserParameter=mysql.size[\*],bash -c 'echo "select sum($(case "$3" in both|"") echo "data\_length+index\_length";; data|index) echo "$3\_length";; free) echo "data\_free";; esac)) from information\_schema.tables$([[ "$1" = "all" || ! "$1" ]] || echo " where table\_schema=\"$1\"")$([[ "$2" = "all" || ! "$2" ]] || echo "and table\_name=\"$2\"");" | HOME=/usr/local/zabbix/etc mysql -N'

UserParameter=mysql.ping,HOME=/usr/local/zabbix/etc/ mysqladmin ping | grep -c alive

UserParameter=mysql.version,mysql -V

UserParameter=mysql.vars[\*],echo "show variables where Variable\_name='$1';" | HOME=/usr/local/zabbix/etc/ mysql -N | awk '{print $$2}'

然后重启，在zabbix-web上添加mysql模板

**如果不能修改my.cnf把mysql账户密码写到**userparameter\_mysql.conf里写到每个userparameter的mysql 后面mysql -uzabbix -pzabbix 2>/dev/null

### 3.3、监控主从状态

模板里没有mysql主从状态，需要要自己定义key

编写脚本 vim /etc/zabbix/script/mysql/mysql\_slave\_status.sh

#!/bin/bash

mysql --defaults-file=/etc/zabbix/myuser -e 'show slave status\G' | grep -E "Slave\_IO\_R

unning:|Slave\_SQL\_Running:" | awk '{print $2}' | grep -c Yes

通过这个脚本可以查看关键字行数，是不是2

然后在 /etc/zabbix/zabbix\_agentd.d目录创建key文件

cd /etc/zabbix/zabbix\_agentd.d

vim mysql\_slave\_status.conf #填写一行内容如下

UserParameter=mysql.slave,/etc/zabbix/script/mysql/mysql\_slave\_status.sh

mysql.slave 是key值，就是监控的键值,后面是脚本路径

重启agent之后,在zabbix服务器上验证下是不是可以获取到这个key

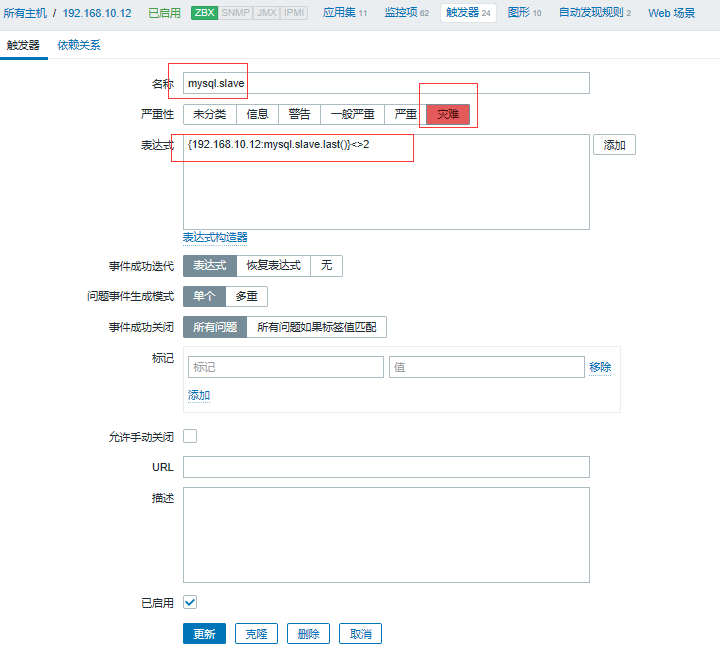
zabbix\_get -s 192.168.10.12 -p10050 -k mysql.slave

然后创建监控项，输入好名称和键值，键值就是刚才文件写的mysql.slave,然后在创建触发器





表达式就在添加里选择→监控项→选中改成添加的监控项mysq.slave,然后设置大于或者小于2就告警。



## 4、监控nginx

参考：http://www.ttlsa.com/zabbix/zabbix-monitor-nginx-performance/

需要在nginx上启用nginx status，添加方法如下

location /nginx\_status

{

stub\_status on;

access\_log off;

allow 127.0.0.1;

deny all;

}

访问后如下curl http://127.0.0.1/nginx\_status

Active connections: 1

server accepts handled requests

10942 10942 10942

Reading: 0 Writing: 1 Waiting: 0

编写nginx检测脚本cd /etc/zabbix/script

vim nginx.sh

#!/bin/bash

HOST="127.0.0.1"

PORT="80"

function ping {pidof nginx | wc -l}

function active {curl "http://$HOST:$PORT/nginx\_status/" 2>/dev/null| grep 'Active' | awk '{print $NF}'}

function reading {curl "http://$HOST:$PORT/nginx\_status/" 2>/dev/null| grep 'Reading' | awk '{print $2}'}

function writing {curl "http://$HOST:$PORT/nginx\_status/" 2>/dev/null| grep 'Writing' | awk '{print $4}'}

function waiting {curl "http://$HOST:$PORT/nginx\_status/" 2>/dev/null| grep 'Waiting' | awk '{print $6}'}

function accepts {curl "http://$HOST:$PORT/nginx\_status/" 2>/dev/null| awk NR==3 | awk '{print $1}'}

function handled {curl "http://$HOST:$PORT/nginx\_status/" 2>/dev/null| awk NR==3 | awk '{print $2}'}

function requests {curl "http://$HOST:$PORT/nginx\_status/" 2>/dev/null| awk NR==3 | awk '{print $3}'}

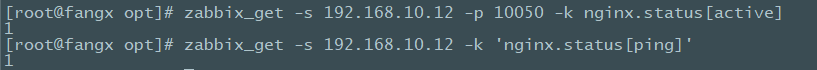
$1



添加参数到zabbix-agent配置文件中vim nginx\_status.conf,然后重启zabbix-agent

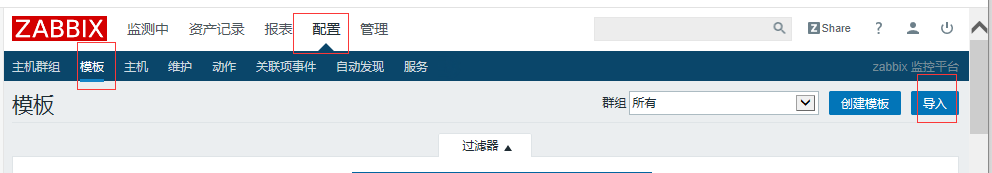
UserParameter=nginx.status[\*],/etc/zabbix/script/nginx.sh $1

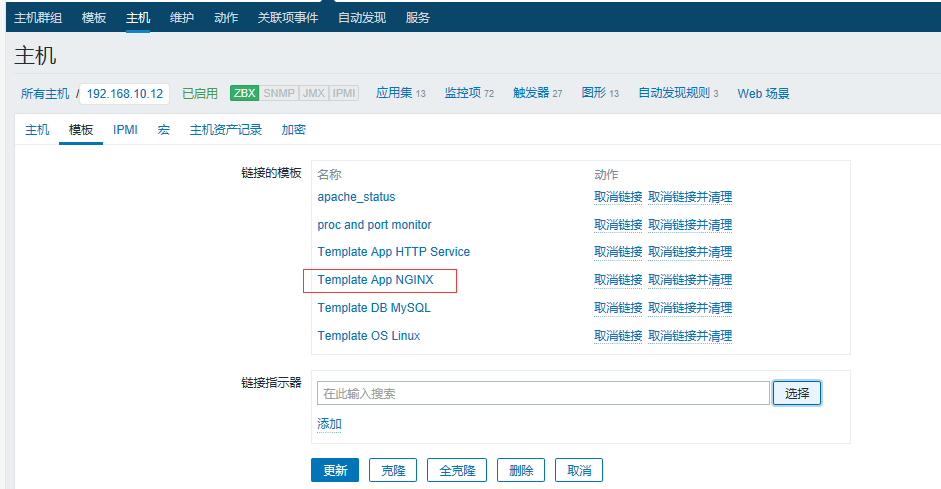
在zabbix服务端测试如下 zabbix\_get -s 192.168.10.12 -p 10050 -k nginx.status[active]



可以直接使用已经创建好的模板，然后应用到需要监控的机器上







以上监控nginx就配置完成了

## 5、web页面URL监控

参考：http://www.ttlsa.com/?s=zabbix%E7%9B%91%E6%8E%A7web

https://blog.csdn.net/slovyz/article/details/53101566

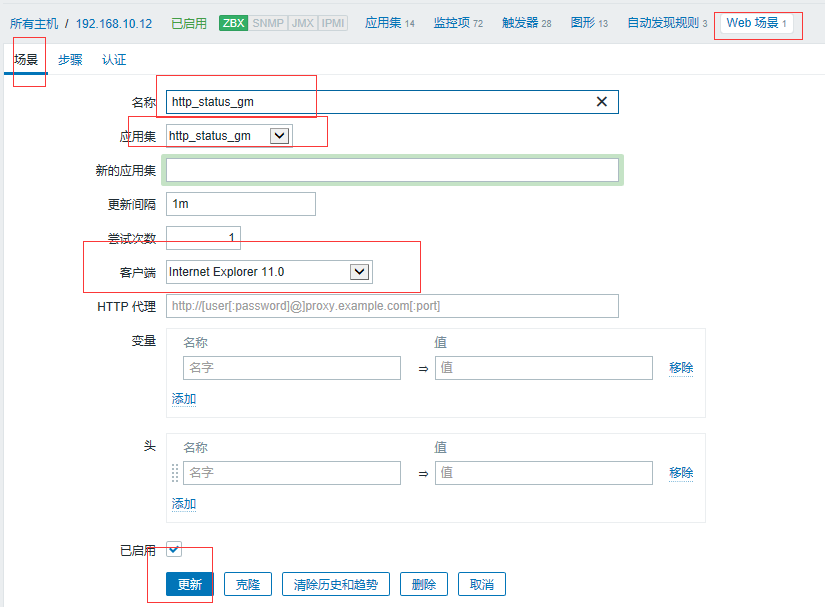
首先要添加web检测场景



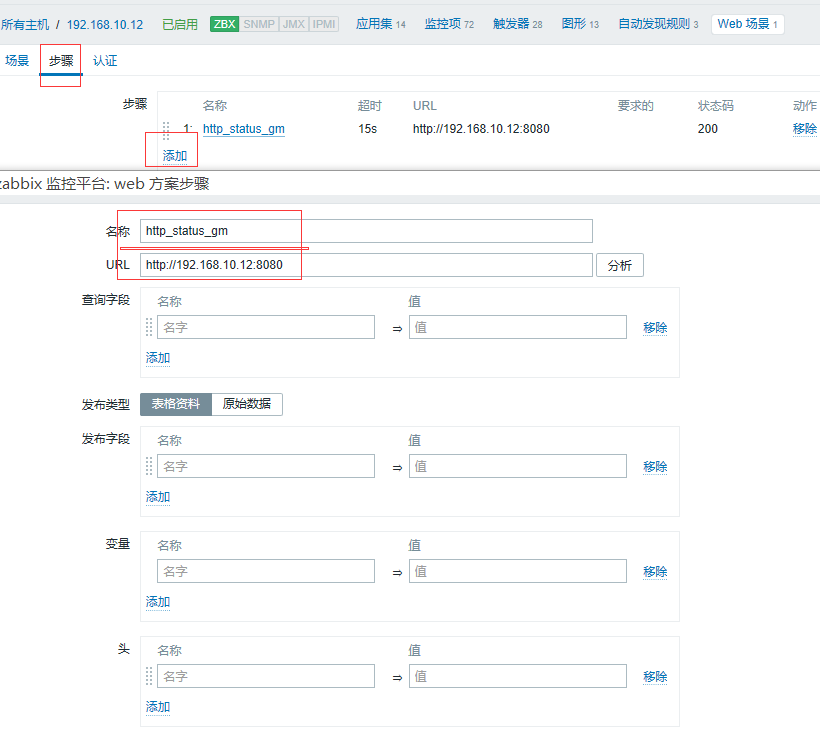




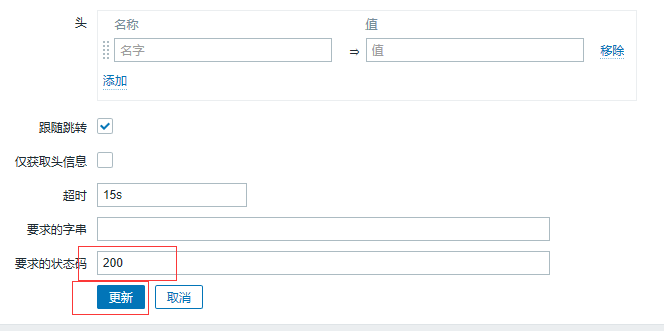
填写检测场景名称



填写需要检测的网站

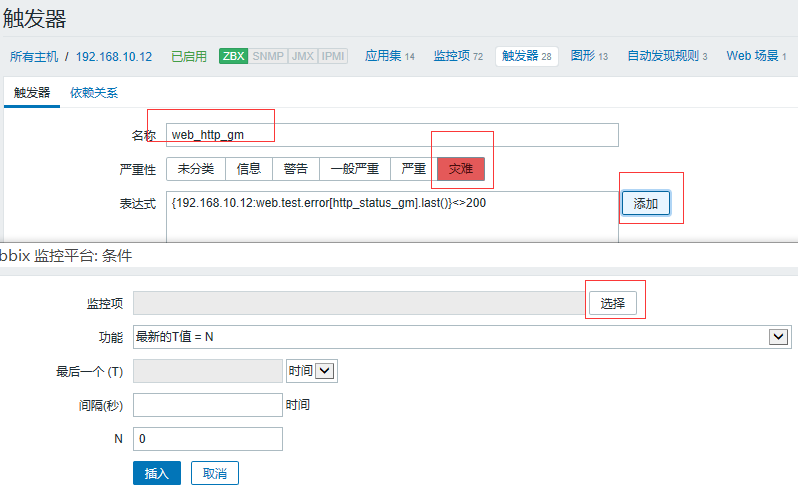


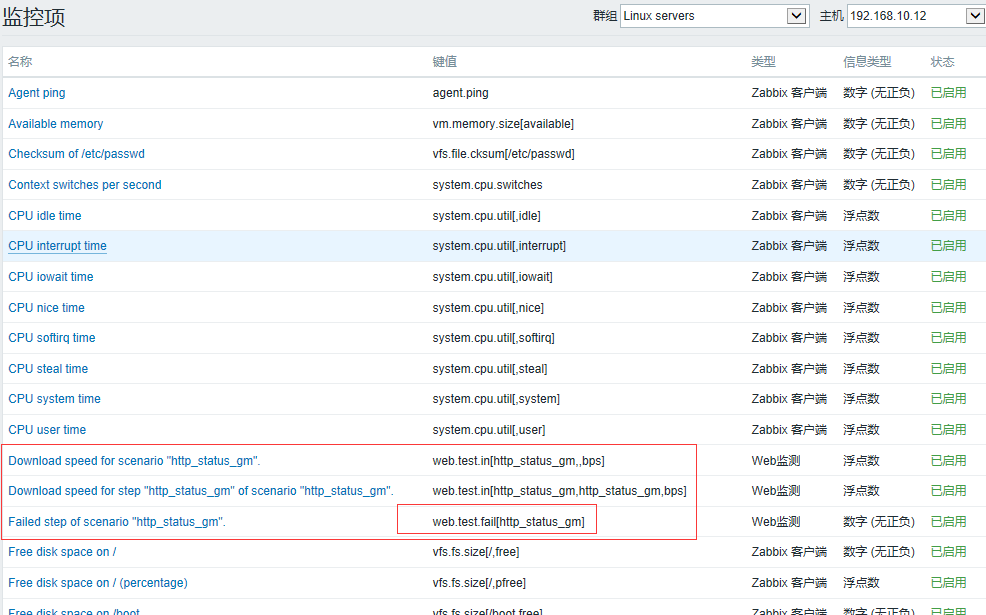
填写要求的状态吗，我们这样要求200

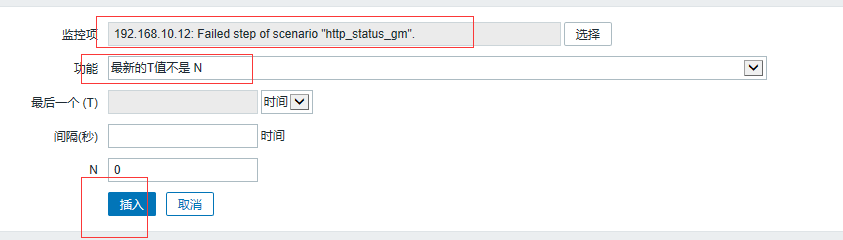


添加触发器

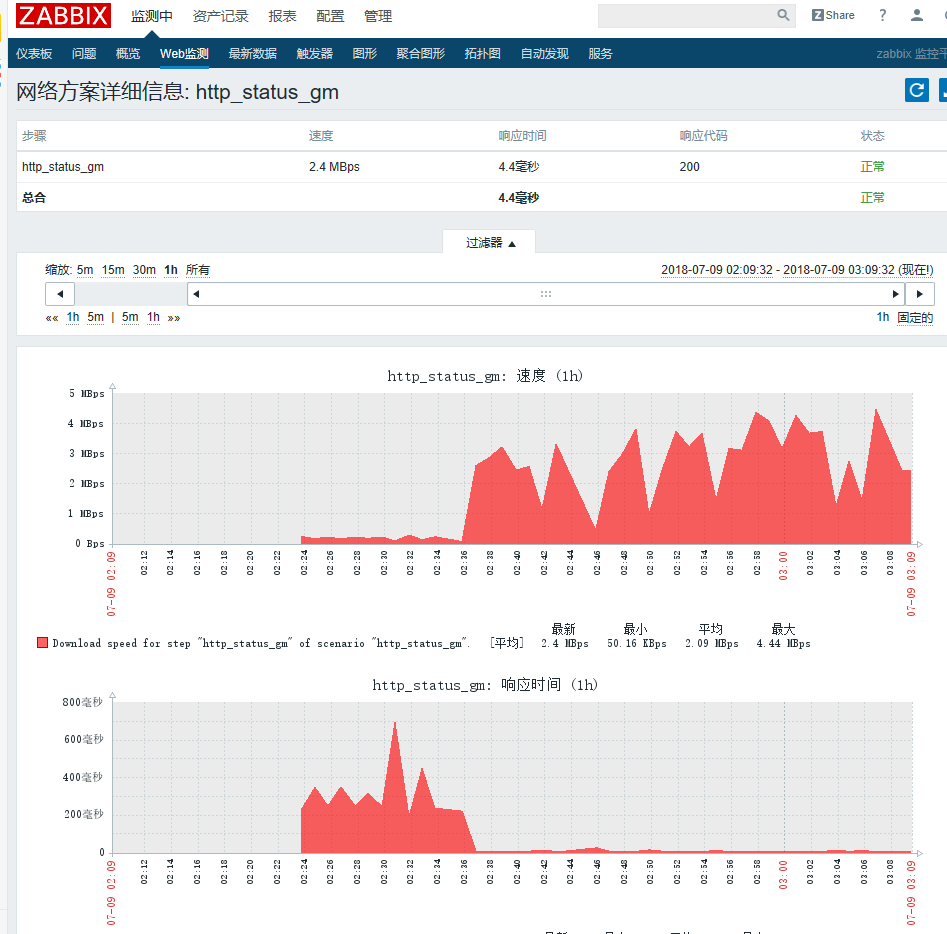








**测试**



## 6、监控API接口性能及可用性

参考：http://www.ttlsa.com/zabbix/zabbix-monitor-web-api/

## 7、监控日志关键字

参考：https://www.cnblogs.com/coolking/p/5918878.html

https://blog.csdn.net/reblue520/article/details/62234834

编写获取日志的shell脚本

vim /etc/zabbix/script/kow\_errlog.sh

#!/bin/bash

msg=`tail -n 100 /opt/apache-tomcat-\*/logs/catalina.out |grep -iE "error|timeout"|wc -l `

count=`echo "$msg"`

if [ ! -n "$msg" ];then

echo "OK"

exit 0

fi

if [ $count -gt 0 ];then

echo "$msg"

exit 1

else

echo "OK"

exit 0

fi

或者脚本如下

#!/bin/bash

# 日志文件目录

path=/opt/apache-tomcat-kow/logs/

# 找到最新的日志文件名 ls -t 按照时间排序，最新的在上面

esb\_file=`ls -t "${path}" | head -1`

fail\_count=`tail -n 200 $path/${esb\_file} |grep 'failure'|wc -l`

echo $fail\_count

配置zabbix-agent客户端key

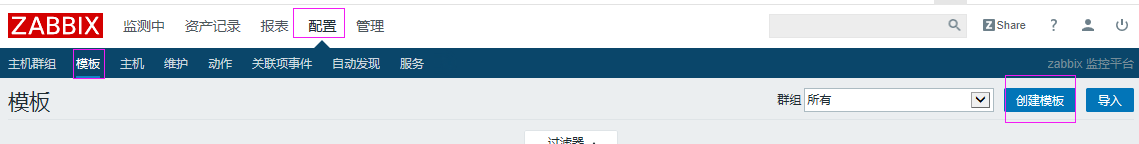
vim /etc/zabbix/zabbix\_agentd.d/logs\_error.conf #内容如下

UserParameter=kow\_errlog,/etc/zabbix/script/kow\_errlog.sh

重启zabbix-agent后，key在zabbix服务端通过zabbix\_get命令验证,

zabbix\_get -s 192.168.10.12 -p 10050 -k kow\_errlog #返回的结果和脚本执行结果一致表示正常

**在zabbix-web上配置监控模板**



配置触发器





触发器的表达式：

{errlog:kow\_errlog.last(0)}>1

也可以这样写

{errlog:kow\_errlog.last(ok)}=0



**监控配置完成**

## 8、zabbix监控redis

参考：https://jasonhzy.github.io/2018/05/12/zabbix-redis/

# 第三章：告警通知

## 1、告警发送QQ和微信消息

参考：<https://www.huangfh.cn/954.html> <http://blog.51cto.com/1130739/1903722> <https://www.cnblogs.com/zhaijunming5/p/6899795.html>

需要了解下Mojo-webqq：参考https://metacpan.org/pod/Mojo::Webqq

### 1.1、QQ通知告警

系统版本：contos7.5，zabbix版本：3.4.11，Mojo-webqq2.2.2

安装必须软件包

yum install lrzsz chrony gcc gcc-c++ git openssl-devel perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker -y

启用系统时间自动同步

systemctl enable chronyd

systemctl start chronyd

安装CPANM工具

curl -L https://cpanmin.us/ -o /usr/bin/cpanm && chmod +x /usr/bin/cpanm

安装相关依赖的PERL插件,等待时间比较久

cpanm --mirror http://mirrors.sohu.com/CPAN/ Digest::MD5 Encode::Locale IO::Socket::SSL Mojolicious Time::Piece Time::Seconds Compress::Raw::Zlib Digest::SHA –force

安装Mojo-Webqq

cd /usr/src/

git clone https://github.com/sjdy521/Mojo-Webqq.git

cd Mojo-Webqq/

perl Makefile.PL

make && make install

安装Webqq::Encryption 模块

yum install -y perl-Crypt-OpenSSL-RSA perl-Crypt-OpenSSL-Bignum

创建Mojo-Webqq启动脚本

vim /usr/local/bin/qqlogin.pl



#!/usr/bin/env perl

use Mojo::Webqq;

use Digest::MD5 qw(md5\_hex);

my ($host,$port,$post\_api);

$host = "0.0.0.0";

$port = 5000;

my $client=Mojo::Webqq->new(

account => 772823217, #QQ账号

pwd => md5\_hex('fan2018/\*-+'), #登录密码

http\_debug => 0, #是否打印详细的debug信息

log\_level => "info", #日志打印级别，debug|info|msg|warn|error|fatal

login\_type => "login", #登录方式，login 表示账号密码登录

);

#my $client = Mojo::Webqq->new(

# http\_debug => 0,

# log\_level => "info",

# login\_type => "qrlogin", );

$client->load("ShowMsg");

$client->load("Openqq",data=>{listen=>[{host=>$host,port=>$port}], post\_api=>$post\_api});

$client->on(ready=>sub{

my $client = shift;

my @groups = $client->groups();});

$client->run();

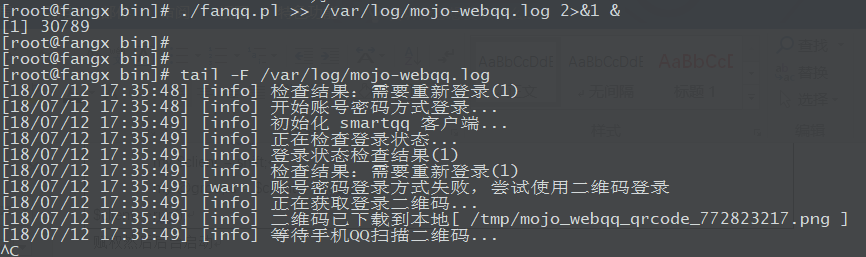
赋权然后后台启动

cd /usr/local/bin/

chmod 755 qqlogin.pl

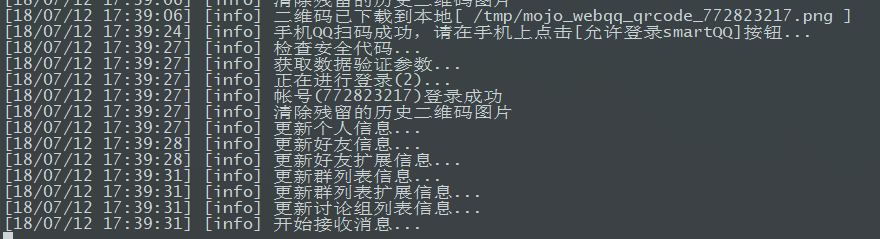
./fanqq.pl >> /var/log/mojo-webqq.log 2>&1 & #后台启动

.提示如下，然后把二维码下载到本地，用手机扫描二维码





扫描后正常提示如下



**配置发送QQ脚本**

cd /usr/lib/zabbix/alertscripts

vim qqclint.sh



#!/bin/bash

#好友的QQ号码

qq=$1

#Mojoqq服务器地址与端口号

server="127.0.0.1"

port="5000"

#消息内容处理下编码，用于合并告警内容的标题和内容，即$2和$3

message=`echo -e "$2\n$3"|od -t x1 -A n -v -w100000000 | tr " " %`

#调用的api接口

#api\_url=<http://$server:$port/openqq/send_friend_message> #发送QQ好友接口

api\_url=<http://$server:$port/openqq/send_group_message> #发送QQ群接口只能启用一个接口,我们告警发送到QQ群

parameter="uid=$qq&content=$message"

#post方式提交请求发送QQ好友消息

curl -d $parameter $api\_url

验证脚本是否可以运行，

./qqclient.sh 820220095 nihao fangx

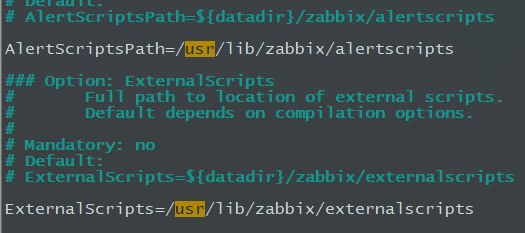
正常提示如下,并接受到QQ群消息





配置zabbix-server启用 AlertScriptsPath=/usr/lib/zabbix/alertscripts 路径后重启zabbix-server

vim /etc/zabbix/zabbix\_server.conf

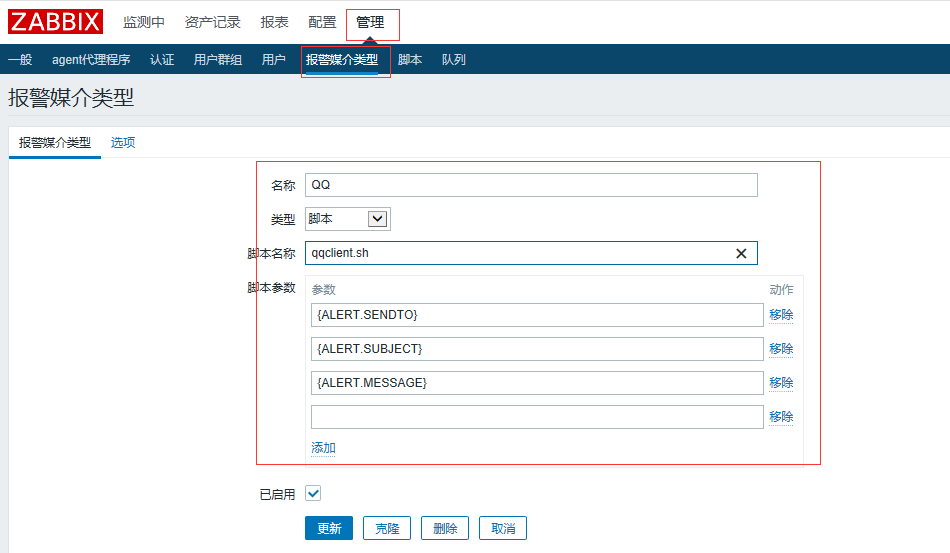


**然后配置zabbix web 发送QQ消息告警**

首先创建报价媒介类型, 管理--->报警媒介类型--->创建媒体类型，配置如下

脚本：qqclient.sh

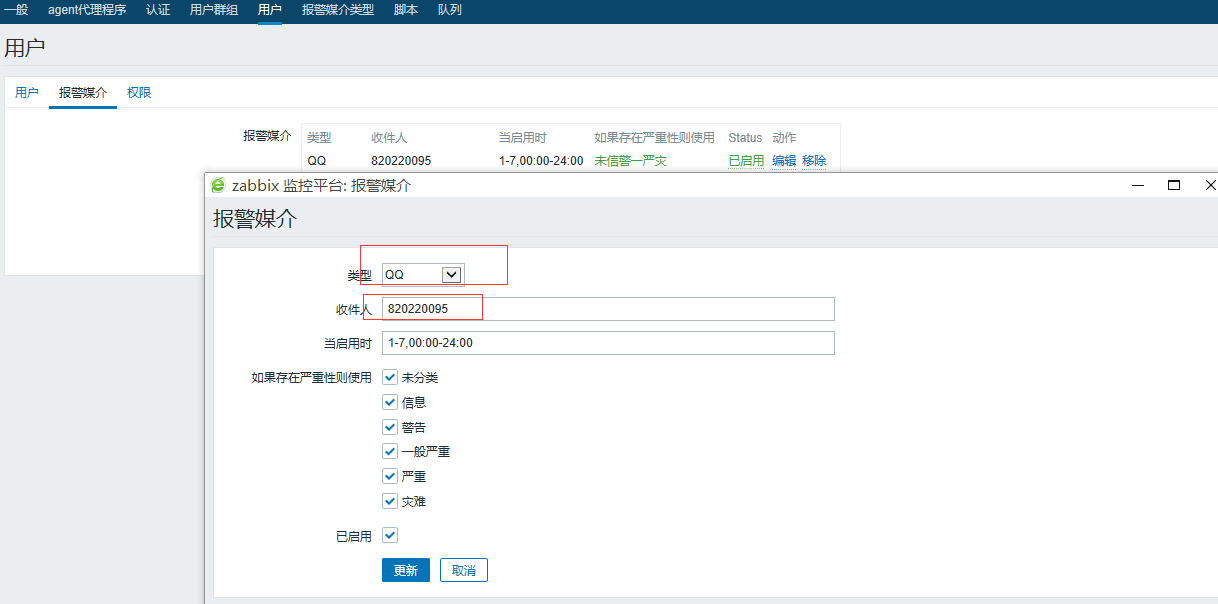
脚本参数：{ALERT.SENDTO} {ALERT.SUBJECT} {ALERT.MESSAGE}



然后配置接受的消息的QQ群

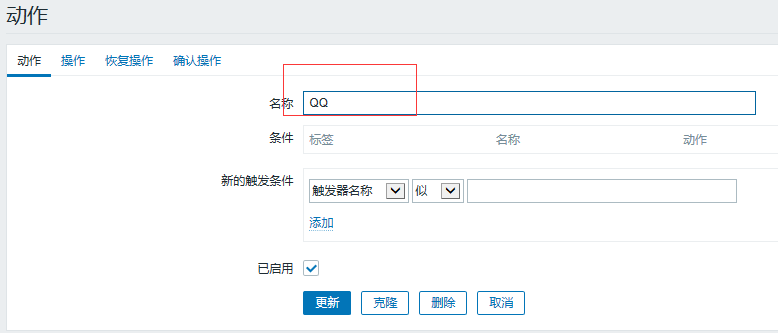
在管理--->用户--->选择Admin---> 报警媒介,配置如下：

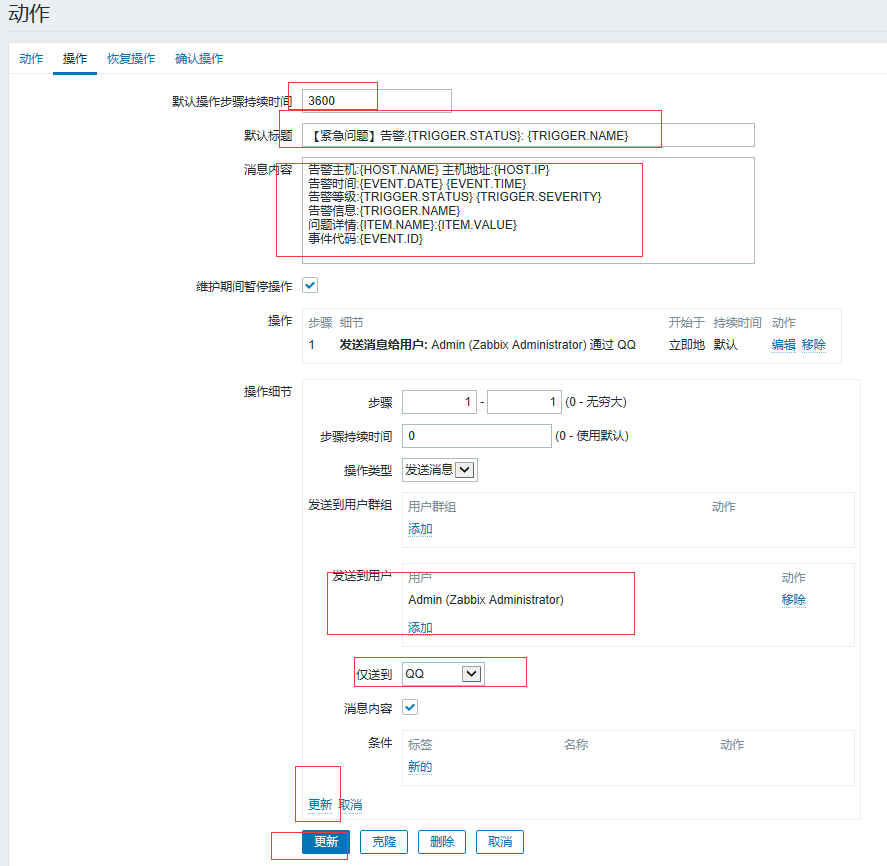
类型选择刚刚创建的QQ，收件人填写QQ群号



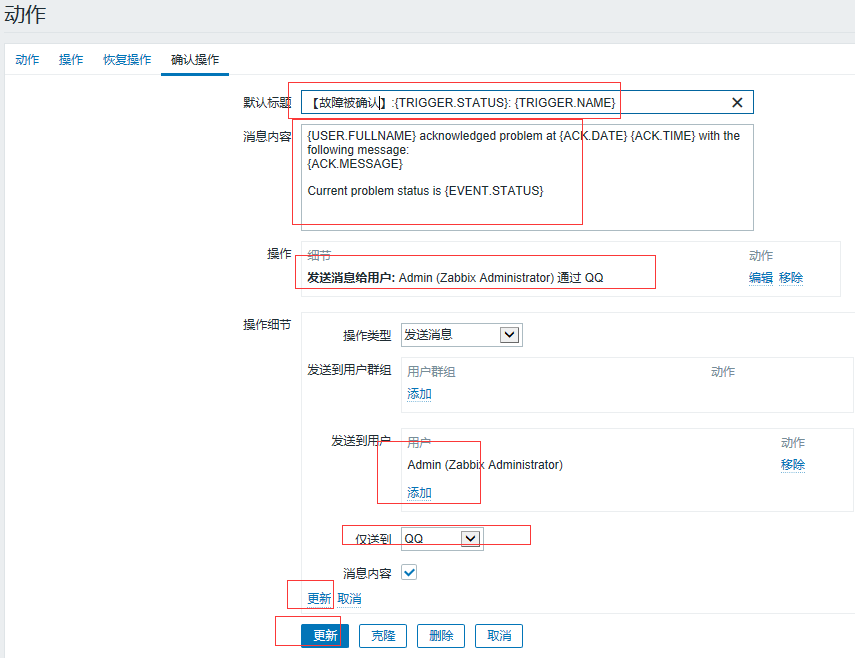
然后配置动作 配置--->动作--->创建动作

告警的消息内容可以参考发送邮件里的消息内容









然后测试收到告警消息如下：

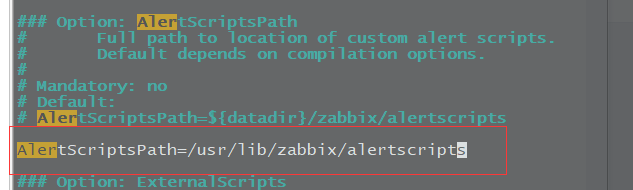


### 1.2、发送告警邮件

配置参考：https://blog.csdn.net/wszll\_alex/article/details/77247696

https://www.cnblogs.com/pythonal/p/7813948.html

修改配置文件：vim /etc/zabbix/zabbix\_server.conf，脚本存放路径，设置如下



上传python脚本到/usr/lib/zabbix/alertscripts目录并加上可执行权限 chmod 755 sned\_mail.py



#!/bin/env python

#coding:utf-8

'''

发送txt文本邮件

'''

import smtplib

from email.mime.text import MIMEText

from sys import argv

mailto\_list=[]

mail\_host="smtp.163.com:25" #设置服务器

mail\_user="gxxxxx@163.com" #发件用户名(换成自己的)

mail\_pass="ZHAxxxxxx" #口令（换成自己的）

#mail\_postfix="163.com" #发件箱的后缀

debug\_level=0 #是否开启debug

def send\_mail(to\_list,sub,content):

me=mail\_user

msg = MIMEText(content,\_subtype='plain',\_charset='utf-8')

msg['Subject'] = sub

msg['From'] = me

msg['To'] = ";".join(to\_list)

try:

server = smtplib.SMTP()

server.set\_debuglevel(debug\_level)

server.connect(mail\_host)

server.login(mail\_user,mail\_pass)

server.sendmail(me, to\_list, msg.as\_string())

server.close()

return True

except Exception, e:

print str(e)

return False

if \_\_name\_\_ == '\_\_main\_\_':

try:

mailto\_list=argv[1].split(';')

sub=argv[2]

content=argv[3]

except:

print "python send\_mail.py 'user1@xx.com;user2@xx.com' sub content"

exit()

if send\_mail(mailto\_list,sub,content):

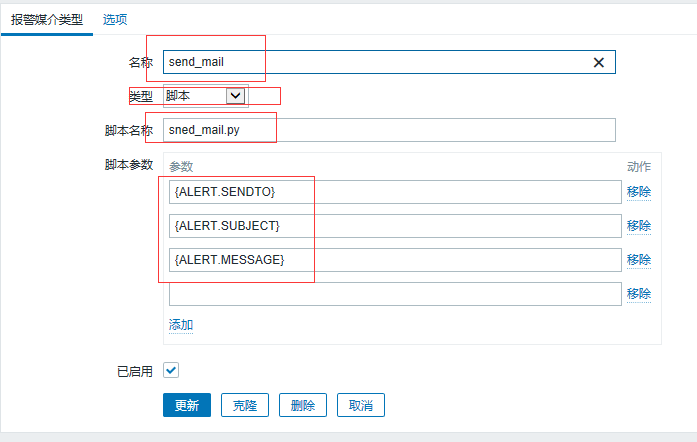
print "发送成功"

else:

print "发送失败"

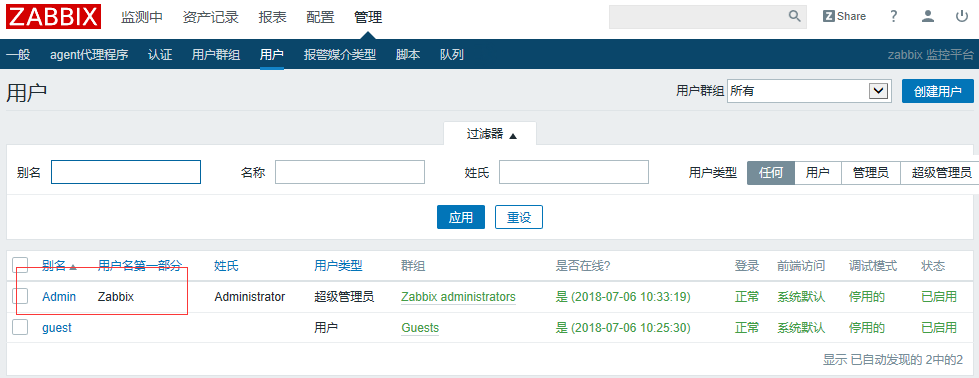
**zabbix web端配置邮件告警**

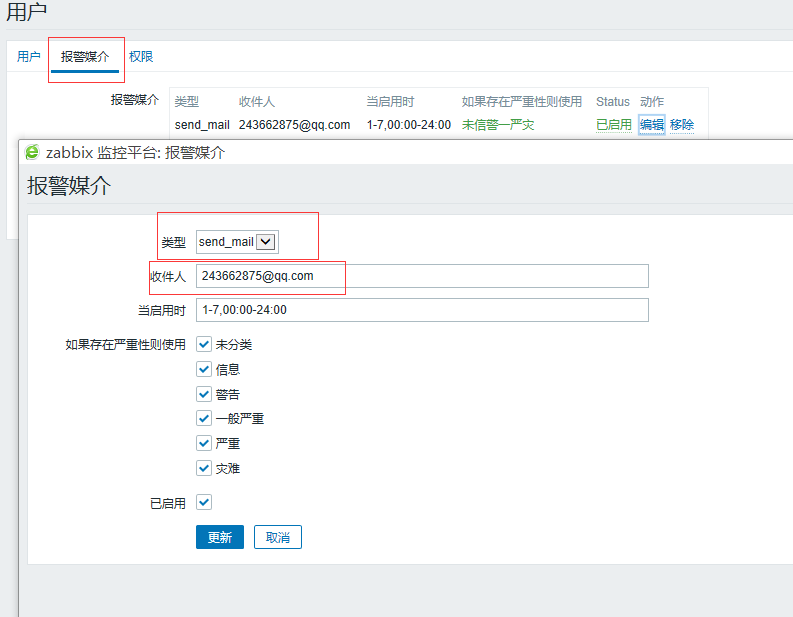
管理——>报警媒介类型——>创建媒体类型 填写内容如下：参数分别对应收件人地址，主题，详细内容

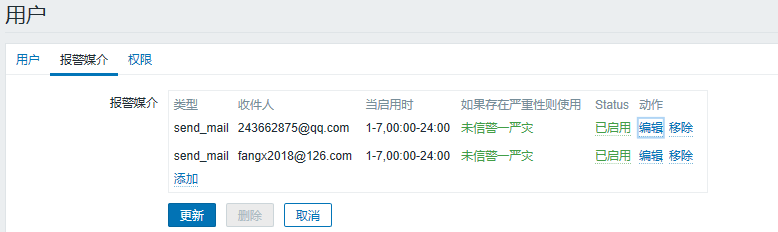




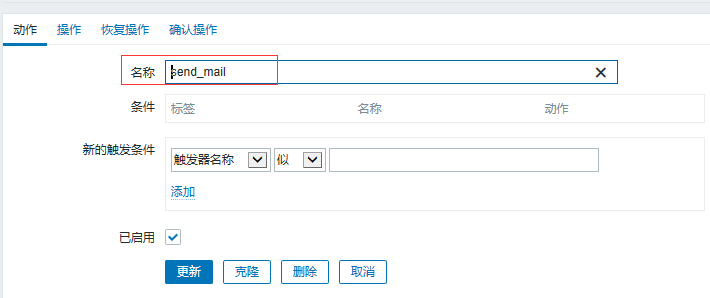
配置接收邮件用户，管理--🡪用户-🡪Admin-🡪在报警媒介处设置接收告警邮件，有多个接收邮件就填写多个

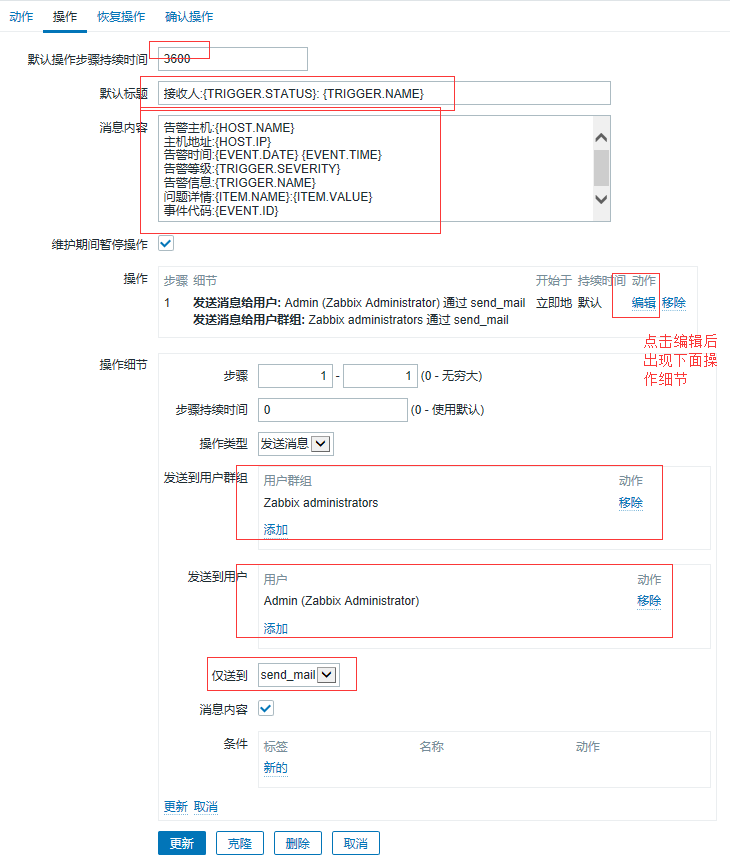


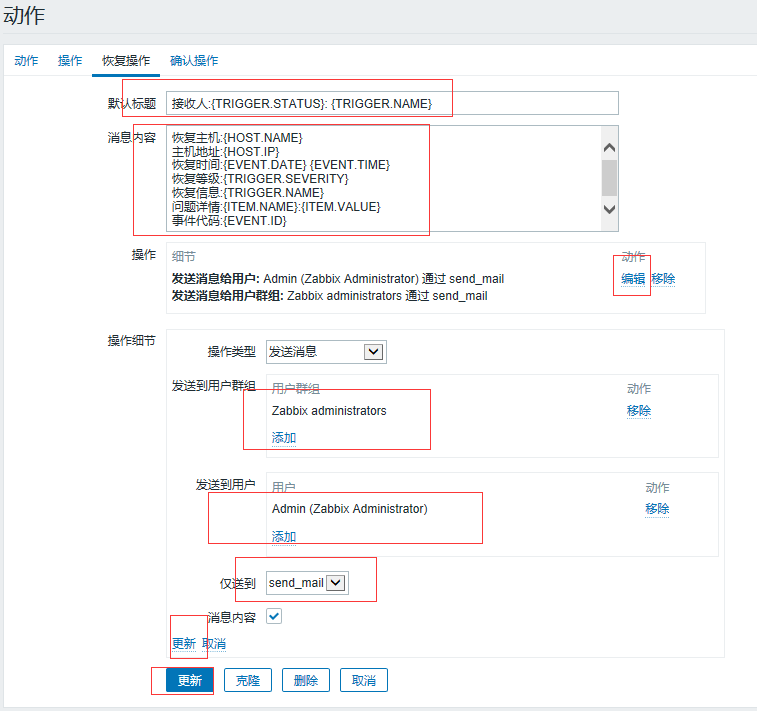




创建动作 配置-🡪动作-🡪创建动作







**问题的告警模板如下**

【**紧急问题**】接收人:{TRIGGER.STATUS}: {TRIGGER.NAME}   
告警主机:{HOST.NAME}   
主机地址:{HOST.IP}   
告警时间:{EVENT.DATE} {EVENT.TIME}   
告警等级:{TRIGGER.SEVERITY}   
告警信息:{TRIGGER.NAME}   
问题详情:{ITEM.NAME}:{ITEM.VALUE}   
事件代码:{EVENT.ID}

-------------------------------------------------------------------------------------------------------

**恢复的告警模板如下**

【**已解决**】接收人:{TRIGGER.STATUS}: {TRIGGER.NAME}

恢复主机:{HOST.NAME}

主机地址:{HOST.IP}

恢复时间:{EVENT.DATE} {EVENT.TIME}

恢复等级:{TRIGGER.SEVERITY}

恢复信息:{TRIGGER.NAME}

问题详情:{ITEM.NAME}:{ITEM.VALUE}

事件代码:{EVENT.ID}

**另一个，可参考**

操作

触发器: {TRIGGER.NAME}

故障状态: {TRIGGER.STATUS}

故障等级: {TRIGGER.SEVERITY}

故障地址: {TRIGGER.URL}

详细状态::

1. {ITEM.NAME1} ({HOST.NAME1}:{ITEM.KEY1}): {ITEM.VALUE1}

时间代码ID: {EVENT.ID}

恢复操作

故障: {TRIGGER.NAME}

故障状态: {TRIGGER.STATUS}

故障等级: {TRIGGER.SEVERITY}

故障地址: {TRIGGER.URL}

详情:

1. {ITEM.NAME1} ({HOST.NAME1}:{ITEM.KEY1}): {ITEM.VALUE1}

事件ID: {EVENT.ID}

然后就可以验证发邮件是否可以了

### 1.3、短信通知告警

### 1.4、钉钉通知告警

### 1.5、微信通知告警

### 1.6、语音通知告警

# 第四章：grafana实现可视化监控图

## 1、zabbix+grafana实现可视化监控图

参考：https://blog.csdn.net/xiaoying5191/article/details/79530280

http://blog.51cto.com/seekerwolf/2122912

wget <https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana-5.2.1-1.x86_64.rpm>