第一章 zabbix 简述

1.1 概述

zabbix是一个基于WEB界面的提供分布式系统监视以及网络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 与可选组件 zabbix agent。

zabbix 支持很多监控方式：agent，agent(主动模式)，简单监测，snmp，snmp trap 等

zabbix server 与 zabbix agent 支持多种平台 Linux，bsd，windows，mac 等

1.2 基本概念

- 项目(iterm)：定义收集被监控的数据的项，如收集被监控机内存使用情况
- 集合(application)：是一些项目的汇总，目录与文件的关系
- 触发器(trigger)：通过项目获得的数据(或者通过计算)来判断主机状态的定义
- 图形(graph)：通过项目获得数据，以图形方式展示
- 模板(template)：可将项目，集合，触发器，图形汇总为一个模板，直接链接到一类主机，实现批量定义
- 主机(host)：被监控对象，服务器或其他网络设备
- 主机组(host group)：一类主机可定义到一个主机组
- 动作(action)：触发器、自动发现或自动注册状态改变触发的动作
- 自动发现(discovery)：通过定义自动发现条件，配合动作批量添加监控主机
- 自动注册(auto-registraion)：agent 向 server 发送注册请求，server 定义自动注册条件批量添加主机
- 低级自动发现(low_discovery)：通过低级自动发现可以简单定义一种类型有多个项的的情况，如：对磁盘容量监控，通常磁盘会有多个分区，我们通过一次定义可以监控磁盘上的所有分区
- 维护(maintenace)：定义主机合适出于维护状态
- 拓扑图(map)：可以主机直接的拓扑
- 屏幕(Screents)：多种类型显示到一个 screent 里
- IT 服务(IT service)：有时一台主机宕掉可能不会影响服务，IT 服务定义服务容忍的限度
- 仪表盘(dashboard)：监控的整体状态显示
- 总览(overview)：显示所有机器的数据或者触发器状态
- web：通过定义场景监控 web 服务器
- 最新数据(last data)：可查看主机项目获得的最新数据
- 事件(Event)：触发器状态改变的记录

第二章 zabbix server 安装

2.1 环境描述

环境:CentOS 5.8 x64 iptables 关闭 Selinux 关闭
Zabbix 版本:2.0.6
Server ip：202.108.1.52
被监控端 ip：202.108.1.51

2.2 安装 lamp 环境

Zabbix server 通过 web 界面来管理的，并且 zabbix server 收集到数据是保存到 mysql 中的

2.2.1 通过 yum 安装

yum -y install gcc gcc-c++ autoconf httpd php mysql mysql-server php-mysql httpd-manual mod_ssl mod_perl mod_auth_mysql php-gd php-xml php-mbstring php-ldap php-pear php-xmlrpc php-bcmath mysql-connector-odbc mysql-devel libdbi-dbd-mysql net-snmp-devel curl-devel

2.2.2 启动服务，设置 mysql 账号密码

service mysqld start
service httpd start
mysqladmin password redhat

2.3 zabbix server 安装

2.3.1 下载解压

wget http://jaist.dl.sourceforge.net/project/zabbix/ZABBIX%20Latest%20Stable/2.0.6/zabbix-2.0.6.tar.gz
tar zxf zabbix-2.0.6.tar.gz
cd zabbix-2.0.6

2.3.2 创建 zabbix 运行需要的用户

groupadd zabbix
useradd zabbix -g zabbix

2.3.3 创建所需数据库并授权用户

mysql>create database zabbix character set utf8;

mysql>grant all on zabbix.* to zabbix@localhost identified by 'redhat';

2.3.4 导入 zabbix 定义的表结构和数据

cd database/mysql/
mysql -uzabbix -predhat zabbix < schema.sql
mysql -uzabbix -p zabbix < images.sql
mysql -uroot -predhat zabbix < data.sql

2.3.5 编译安装 Zabbix server

cd ../..
./configure \
--prefix=/usr/local/zabbix --enable-server --enable-agent --with-mysql \
--with-net-snmp --with-libcurl
make && make install

配置参数说明：

```
--enable-server 安装 Zabbix Server
--enable-proxy 安装 Zabbix Proxy
--enable-agent 安装 Zabbix Agent
--with-mysql 使用 mysql 做数据库服务器
--with-net-snmp 支持 SNMP
--with-libcurl 支持 curl，用于 web 监控
```

2.3.6 服务端口定义

vim /etc/services ##在后面追加：

```
zabbix-agent    10050/tcp    #Zabbix Agent
zabbix-agent    10050/udp    #Zabbix Agent
zabbix-trapper  10051/tcp    #Zabbix Trapper
zabbix-trapper  10051/udp    #Zabbix Trapper
```

2.3.7 修改 zabbix server 配置文件

vim /usr/local/zabbix/etc/zabbix_server.conf

```
LogFile=/tmp/zabbix_server.log    ##日志位置，根据需求修改；
PidFile=/tmp/zabbix_server.pid    ##PID 所在位置
DBHost=localhost                  ##如果不是在本机，请修改
DBName=zabbix                     ##数据库名称
DBUser=zabbix                     ##数据库用户名
DBPassword=redhat                 ##数据库密码
```

2.3.8 安装启动脚本,添加可执行权限

cp misc/init.d/fedora/core/zabbix_server /etc/init.d
chmod +x /etc/init.d/zabbix_server

2.3.9 修改启动脚本，启动 zabbix server

vim /etc/init.d/zabbix_server

BASEDIR=/usr/local/zabbix ##修改这个，zabbix 的安装目录

CONFILE=$BASEDIR/etc/zabbix_server.conf ##添加这一行，定义配置文件位置

#搜索 start,修改启动选项，默认是去/etc 下去找配置文件的

action $"Starting $BINARY_NAME: " $FULLPATH ***-c $CONFILE***

service zabbix_server start

2.4 安装 zabbix web 界面

2.4.1 复制 web 代码到 httpd 配置文件指定的目录下，一般是/var/www/html

cp -r frontends/php /var/www/html/zabbix

chown -R apache:root /var/www/html/zabbix

2.4.2 访问 http://serverip/zabbix，通过界面安装 zabbix web 端

有关于 date()的错误提示解决方法：

vim /etc/php.ini

date.timezone = Asia/Chongqing ##时区修改为重庆，刷新浏览器

2.4.3 步骤简要说明

- 检查安装环境

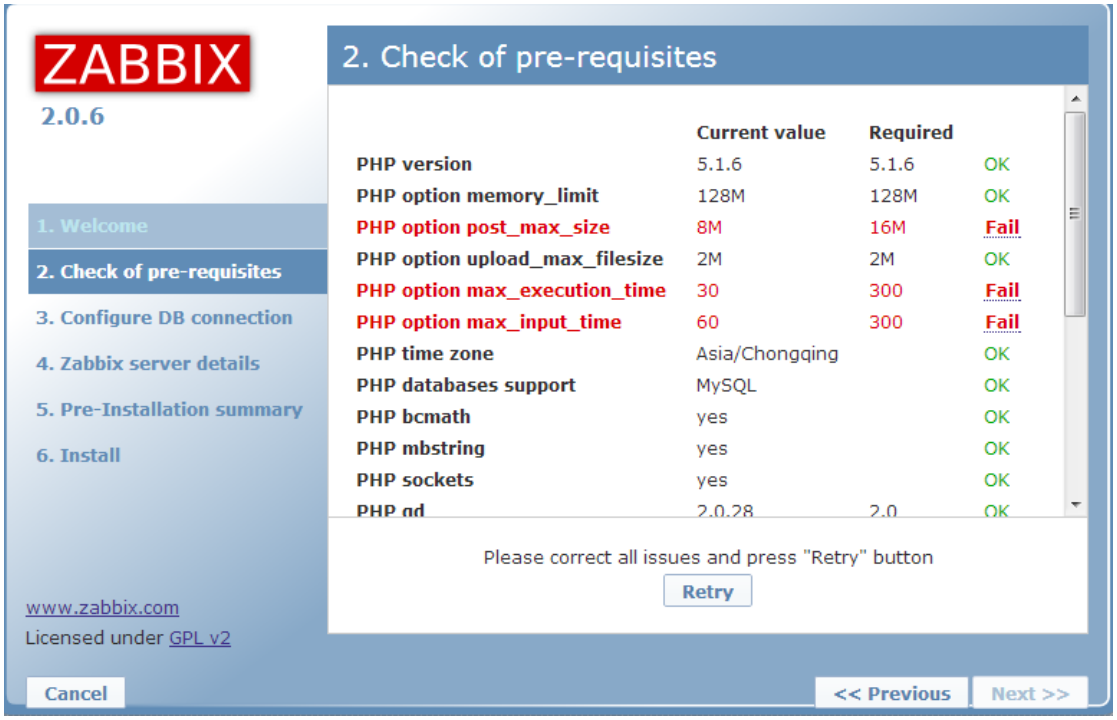


图 2.1

根据提示修改/etc/php.ini 如下项，如果还有别的可能是你的 php 模块没有安装全，Retry

post_max_size = 16M

max_execution_time = 300
max_input_time = 300

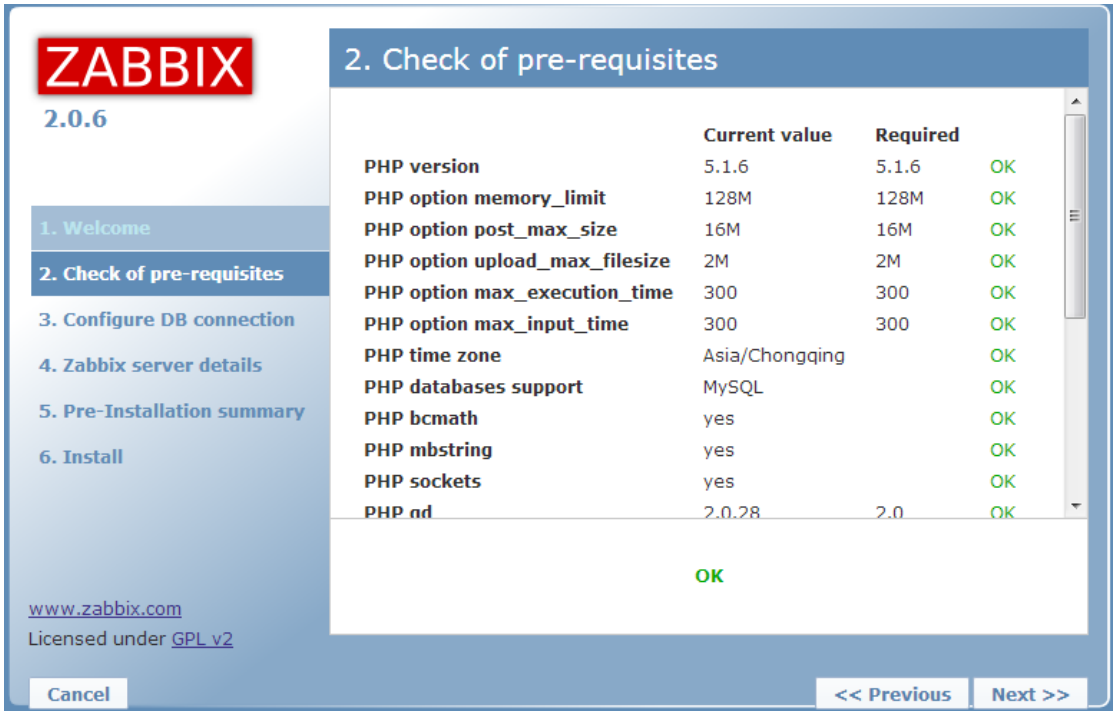


图 2.2

- 连接 MySQL 的参数

图 2.3

- zabbix server 详情

图 2.4

- 最后显示所有配置信息

图 2.5

- 完成，如果出现下面情况是 apche 没有/var/www/html/zabbix 写入权限，修改后 Retry

图 2.6

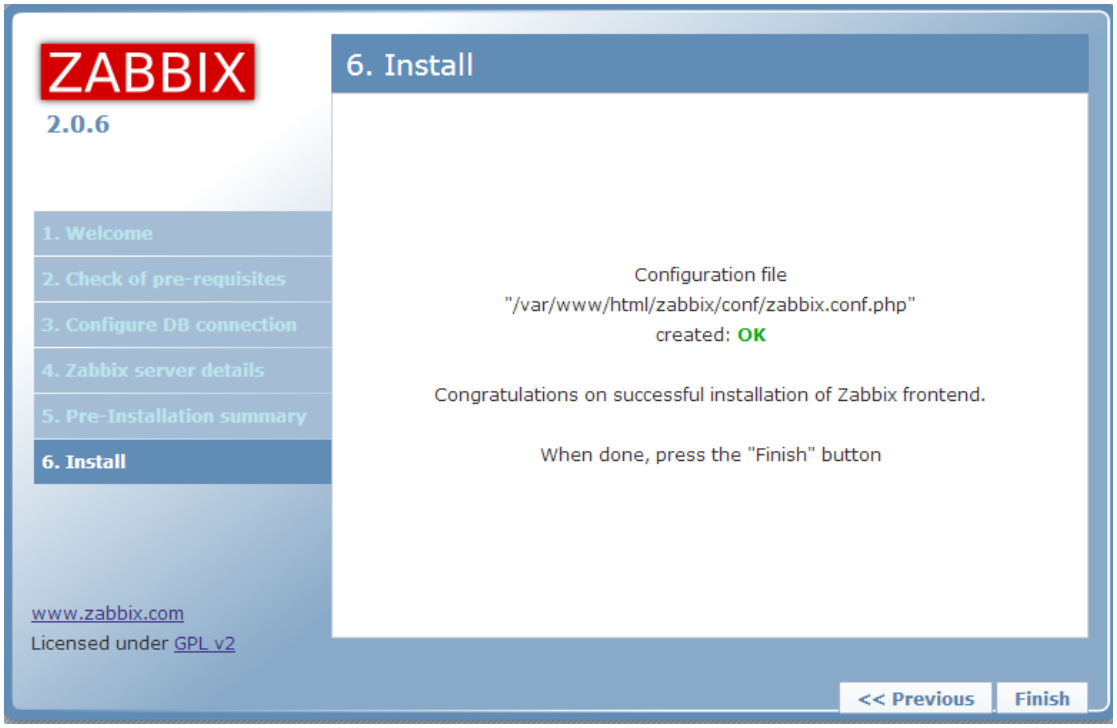


图 2.7

- 访问 http://serverip/zabbix 登陆测试,默认账号 admin 密码 zabbix

图 2.8

图 2.9

Tips:修改中文　　右上角 profile→Language 改成 chinese 即可

图 2.10

第三章 zabbix agent 的安装

3.1 Linux agent 的安装(在另一台机器上安装)

3.1.1 下载解压编译好的客户端

wget http://www.zabbix.com/downloads/2.0.6/zabbix_agents_2.0.6.linux2_6.amd64.tar.gz
mkdir /usr/local/zabbix
tar zabbix_agents_2.0.6.linux2_6.amd64.tar.gz -C /usr/local/zabbix

3.1.2 编辑配置文件

cd /usr/local/zabbix/etc
vim zabbix_agentd.conf

```
LogFile=/tmp/zabbix_agentd.log
Server=202.108.1.52              ##服务器 IP
ServerActive=202.108.1.52     ##主动模式服务器 IP
Hostname=202.108.1.51         ##设定主机名
```

3.1.3 安装修改启动脚本

下载的这个里面没有脚本，但在 202.108.1.52 zabbix 源码包里有
scp misc/init.d/fedora/core/zabbix_agentd 202.108.1.51:/etc/init.d
vim /etc/init.d/zabbix_agentd

```
BASEDIR=/usr/local/zabbix                    ##修改这个
CONFILE=$BASEDIR/etc/zabbix_agentd.conf ##添加这行，搜索 start 添加蓝色代码
action $"Starting $BINARY_NAME: " $FULLPATH -c $CONFILE
```

service zabbix_agentd start

3.2 Windows 客户端安装

3.2.1 下载客户端，解压缩

http://www.zabbix.com/downloads/2.0.6/zabbix_agents_2.0.6.win.zip

3.2.2 在合适位置建立 zabbix 文件夹

我是建立在 d:/下了，根据你的 windows 版本(32bit or 64bit),把 bin 下的版本目录下的文件拷贝过去，再把 conf 下的配置文件拷贝到 zabbix 文件夹下，更名叫 zabbix_agentd.conf

3.2.3 修改配置文件 zabbix_agentd.conf

LogFile=c:\zabbix_agentd.log

Server=202.108.1.52

ServerActive=202.108.1.52

Hostname=Windows_2003_1.52

3.2.4 安装 zabbix_agentd 为服务

开始→运行→cmd

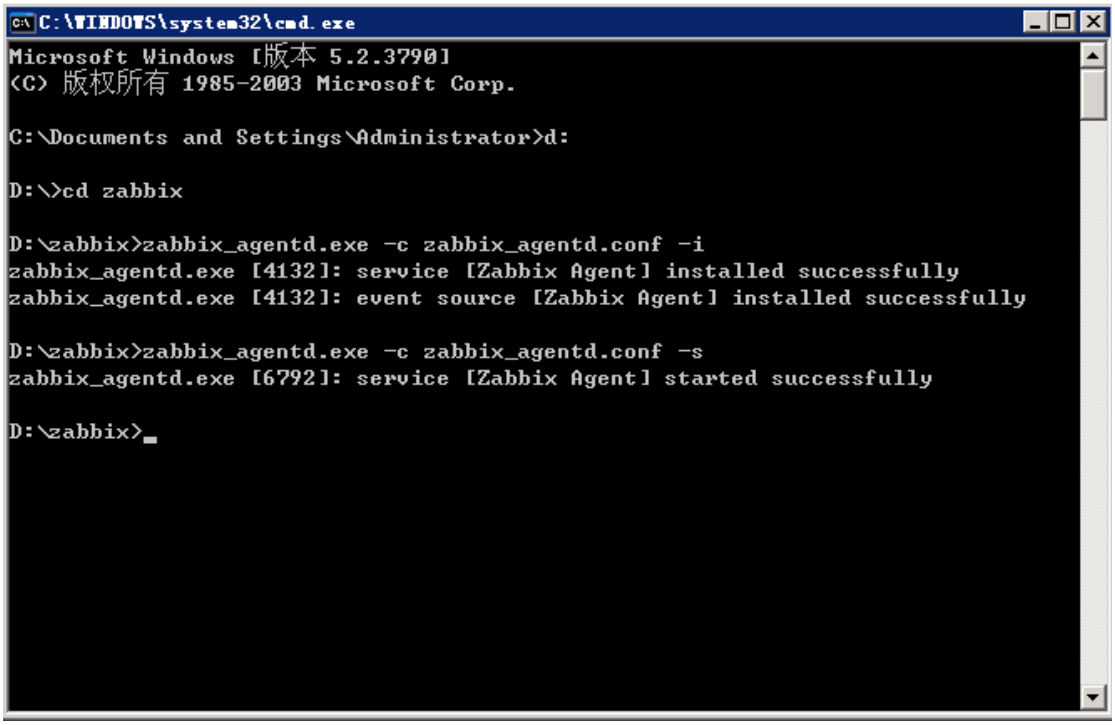


图 3.1

-c 指定配置文件所在位置

-i 安装客户端
-s 启动客户端
-x 停止客户端
-d 卸载客户端

到此客户端安装完毕！

第四章 配置监控

4.1 添加监控主机

4.1.1 配置→主机→创建主机(右上角)

图 4.1

主机名称：这个应该是你 agent 配置文件定义的 Hostname，否则 agent 的主动模式收不到数据，日志里会报*找不到主机名****，不能发送 active list* 的错误！

可见名称：这个就是显示名称，自定义即可
组：将主机归类到哪个组中
下面的就是监控方式，根据你选的监控方式来定义下面参数，
代理接口：就是主机上安装 agent 的
SNMP 接口：通过 snmp 来监控的
JMX 接口：监控 JAVA 程序的接口
IPMI 接口：通过 IPMI 接口来监控硬件
有代理节点监测：通过 proxy 来监控
状态：指明是否立刻监控该主机

4.2 定义模板

我们通常是将项目，触发器，图形等等定义在模板上，然后由模板链接到主机上，来实现批量监控主机的！

4.2.1 添加模板

配置→模板→创建模板

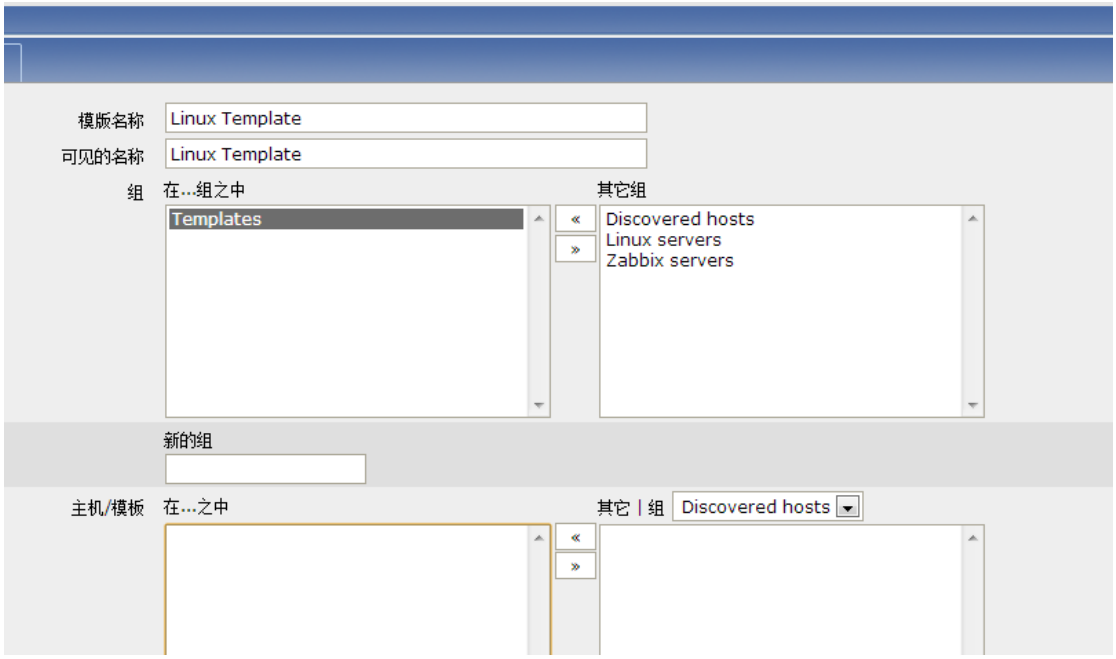


图 4.2

模板名称：自定义一个模板名称，通过该名字引用模板
可见名称：在模板中看到的模板名字
组：定义改模板属于的分组
新的组：如果没有中意的组可以添加到新的组中
主机/模板：链接该模板的模板或主机
其它|组：没有链接该模板的主机或者模板，可通过<<添加进来

4.3 添加监控项

在主机中添加监控项与在模板中一样，我们以在模板中添加监控项为例

4.3.1 点击模板上面的项目

图 4.3

弹出窗口填写,本演示添加的是*内存使用率*的监控项

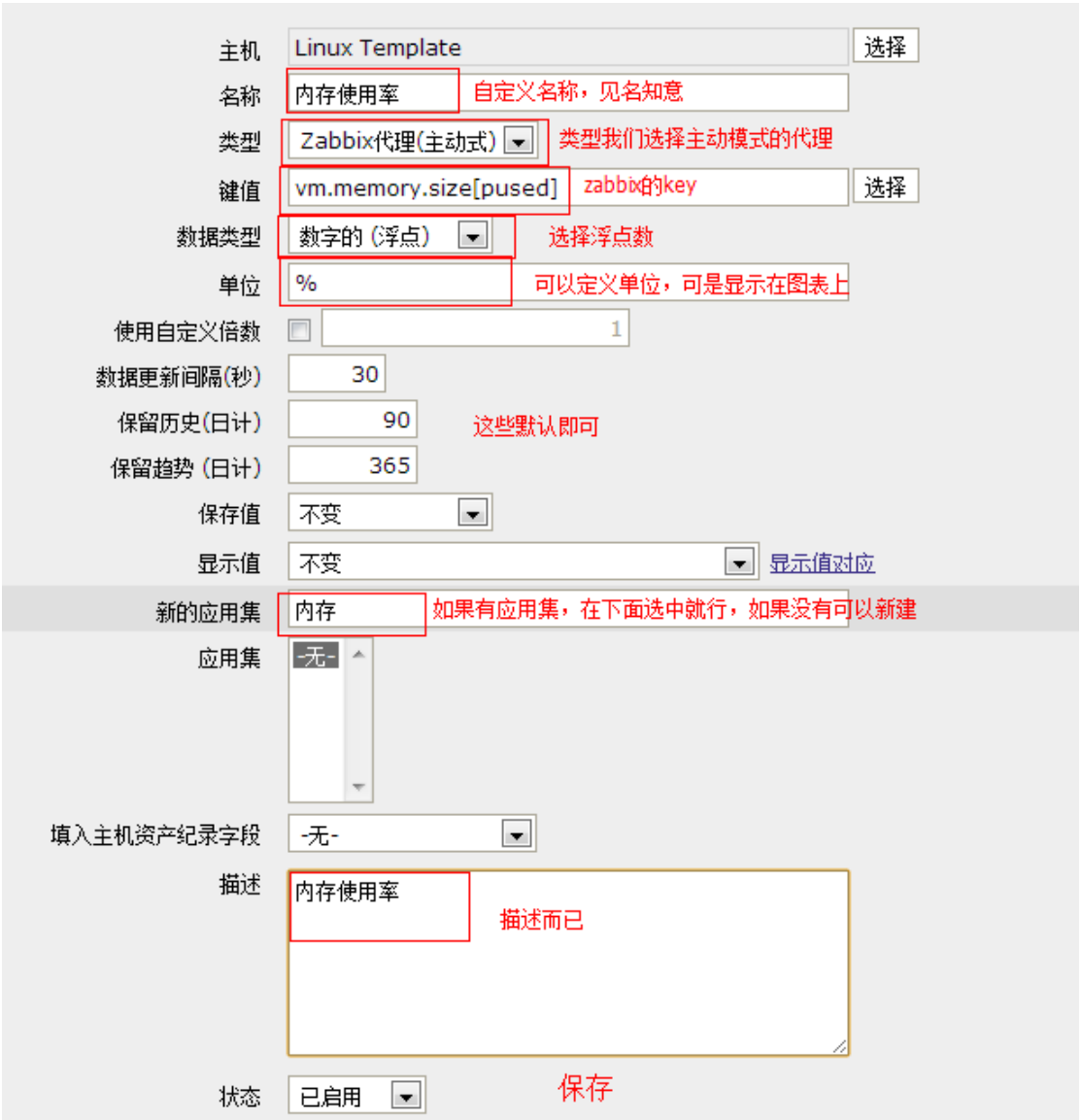


图 4.4

主机：如果是在模板中定义，就是模板的名字，如果是在主机中，就是主机的名称

名称：定义该项目的名称

类型：选择监控方式，本次用的是 agent 的代理模式，稍后会讲其他方式

键值：也是就是 key，通过 key 来获得 agent 上的数据，snmp 的 key 只是为了以后引用可以自定义，agent 内置了好多定义好的 key，vm.memroy.size 是内存使用相关的 key，稍后我们有一章节来将常用 key，所有 key 见 https://www.zabbix.com/documentation/1.8/manual/config/items

数据类型：定义的 key 返回的数据类型

单位：定义单位有两个用途，1 是为了单位换算，2 是为了查看方便

自定义倍数：将获得的数据乘以自定义的倍数作为项的值

数据更新间隔：多长时间更新一下数据

保留历史：保留过去多长时间的数据

保留趋势：保留多长时间的趋势数据

保存值：不变是获得的值直接保存(如果有自定义倍数再乘以自定义倍数)，差量(每秒速率)是把每秒的变化速率保存，如用来计算流量，差量(单纯变化)是保存这一次与上一次的差值。

显示值：提供值与项的映射，只适合与数字，不常用

应用集：项目属于哪个应用集，如果没有合适，可填写新应用集

资产记录：归类的那个资产记录中，不常用

描述：对项目提供直白的描述

状态：立刻监控，还是暂时不监控

4.4 添加触发器

我们需求是当内存使用率超过 95%就报警。

4.4.1 为内存使用率添加触发器

选择模板(或主机)上的触发器→创建触发器

图 4.5

名称：触发器的名称

表达式：触发器激活的条件，当表达式为真时触发器触发。表达式可以手写，也可以通过添加来选择。表达式：{Linux Template:vm.memory.size[pused].last(0)}>95，”：”前

面是主机或者模板的名字，紧跟的后面是监控项的 key，再后面是那个 last(0)是函数(还有哪些函数呢，我们稍后说)，最后{}里面是经过运算后的值，所以该表达式的意思就是模板(Linux Templat)中的监控内存使用率的项(vm.memory.size[pused])最后获得的值大于 95 的话，触发器成立。理解了这个手动选择就简单了，先选择项目(主机的还是模板的)，功能就是运算的函数，如：最后获得值，还是 5 分钟内的最大值，后面就是根据选择的函数确定参数。

描述：对触发器简单描述

严重性：定义触发器的严重性

该触发器定义好了，当内存使用率超过 95%后触发器激活，在仪表盘上会显示触发器状态为 Problem，如果还定义了动作，便执行相应动作，如发邮件告警。

4.5 添加图形

4.5.1 为内存使用率绘图

在模板(或)上点击图形→创建图形(右上角)

图 4.6

名称：图形的名称

宽：图形的宽度

高：图形的高度

图形类别：正常，层级，圆饼具体区别自己可以试试

显示图标：就是是否显示每个颜色代表的是什么项，是否显示平均值等

显示工作时间：如果选择，非工作时间背景是灰的

显示触发器：在图形上显示触发器

百分比线：不常用

Y 轴最小值：计算的是 zabbix 运算决定最小值，固定是指手工定义最小值，项是指指定某个项为最小值

Y 轴最大值：同上

项目：选定以哪个项目来绘图

名称：项的名称
功能：以哪个值绘图
绘图样式：什么形状的图，线的，虚线的，粗线的等等
Y 轴：在在左侧还是右侧
颜色：指定颜色

4.6 将模板模板链接到主机

4.6.1 模板连接到主机

点击主机名称→选择模板→添加　选择想要模板链接到主机 没错，就是这么简单。

图 4.7

4.6.2 查看主机项目最新数据

检测中→最新数据→选择相应的主机与项目 查看最新数据

图 4.8

4.6.3 查看主机项目图形

检测中→图形→选择主机→选择查看图形

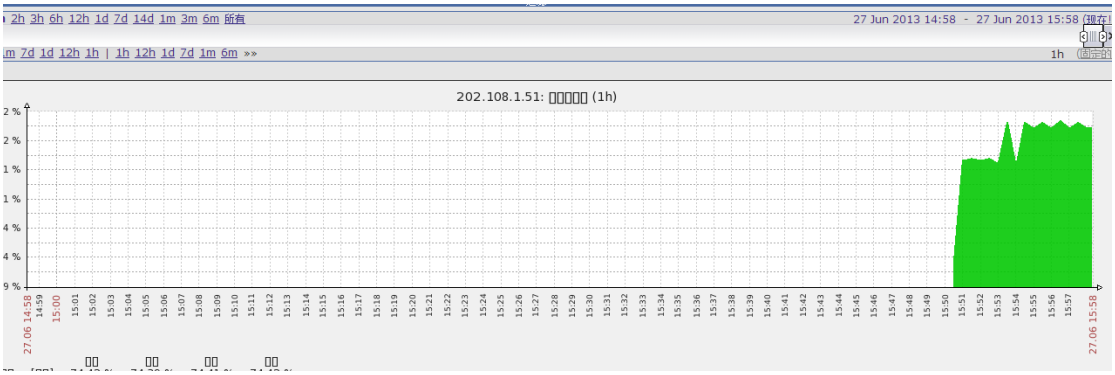


图 4.9

Tips：图形上中文乱码解决方法：

a) 从你 win 机器上选择一个中文字体(如微软雅黑)，导出传到

/var/www/html/zabbix/fonts 中

b) 备份 DejaVuSans.ttf

c) 把微软雅黑字体文件改名为 DejaVuSans.ttf

d) 刷新浏览器 就 ok 了

4.6.4 修改触发器，查看异常状态

把内存使用率的触发器值改为 20(通过图 4.8,4.9 可知我的内存使用差不多在 70%左右)，查看到首页仪表盘报警

图 4.10

图 4.11

4.7 添加 mail 报警方式

4.7.1 通过 sendmail 发送邮件

a) 确保你的主机名字不是 localhost，大部分邮件系统都拒收该邮件，mail 测试
   echo “all servers is ok”| mail -s “server ok” ibuler@qq.com ##如果 ok 请继续
b) 在/usr/local/zabbix 下建立目录 scripts(名字可以自定义)
   mkdir /usr/local/zabbix/scripts
c) 在目录下建立文件 sendmail.sh，给执行权限
   vim sendmail.sh
     #!/bin/bash
     echo $3 | mail -s $2 $1
   chmod +x sendmail.sh
d) 修改 zabbix 配置文件 zabbix_server.conf，告诉 zabbix 自定义报警脚本目录在哪，然后重启 zabbix server
   vim zabbix_server.conf
     AlertScriptsPath=/usr/local/zabbix/scripts ##取消注释，并修改
e) web 管理界面 管理→示警媒体类型→创建示警媒体类型

图 4.12

4.7.2 添加通过 mailx 用其他电子邮件运营商 SMTP 报警

mailx 工具可以使用别的 SMTP 发邮件，在 CentOS5 系列上没有安装，CentOS 6 系列上默认安装的，下面来安装配置它。

a) wget http://nchc.dl.sourceforge.net/project/heirloom/heirloom-mailx/12.4/mailx-12.4.tar.bz2
b) 编译安装
   tar xvf mailx-12.4.tar.bz2
   cd mailx-12.4
   make
   make install UCBINSTALL=/usr/bin/install
c) 修改其配置文件/etc/nail.rc
   vi /etc/nail.rc
   ##最后面添加
   set from=ibuler@qq.com ##你的邮箱地址
   set smtp=smtp.qq.com ##你邮箱提供商的 smtp 服务器地址
   set smtp-auth-user=ibuler ##你的登录账号
   set smtp-auth-password=linux12345 ##你的密码

set smtp-auth=login　　##学过 postfix 都知道这个

d) 发送测试，自己给自己发一封

echo "服务器非常稳定" | mailx -s "服务器信息" ibuler@qq.com

图 4.13

Tips:如果出现乱码，请记得修改服务器编码，客户端编码

服务器:export LANG=zh_CN.utf8

客户端:就是你的 ssh 工具的编码

e) 编写脚本，给执行权限

cd /usr/local/zabbix/scripts

vim mailx.sh

#!/bin/bash

echo "$3" | mailx -s "$2" $1

chmod +x mailx.sh

f) web 管理界面添加示警媒体

图 4.14

4.8 添加飞信报警

4.8.1 下载安装飞信机器人

官方地址：http://bbs.it-adv.net/viewthread.php?tid=1081&extra=page%3D1

下载解压到/usr/local/fetion 中并给执行权限，把库解压到/usr/local/fetion/lib 中

运行 fetion 查看缺少库文件，把刚才解压的库文件软连接到/lib 中，不要全部链接，所有都链接后可能会报 segment fault 。

[root@xk-1-52-a8 fetion]# chmod +x fetion

[root@xk-1-52-a8 fetion]# ./fetion

./fetion: error while loading shared libraries: libACE-5.7.2.so: cannot open shared object ...

ln -s /usr/local/fetion/lib/ libACE-5.7.2.so /lib
重复执行意识动作，直到不再报丢失库文件为止

4.8.2 测试使用飞信机器人

*# ./fetion --mobile=1520105**** --pwd=fetionpsw --to=1520105**** --msg-utf8="message"*

--mobile 发送飞信号
--pwd 该飞信号密码
--to 接受短信手机号
--msg-utf8 短信内容

这时会提示输入生成验证码图片，把图片下载下来，查看图片验证码，输入验证码，短信发送成功

4.8.3 编写发送短信脚本

vim /usr/local/zabbix/scripts/fetion.sh
#!/bin/bash
/usr/local/fetion/fetion --mobile=15201053231 --pwd=lanfeng117 --to=$1 --msg-utf8="$2 $3"
chmod +x fetion.sh

4.8.4 测试脚本

/usr/local/zabbix/scripts/fetion.sh 1520105**** “Server OK” “The Server have been OK”

4.8.5 在 web 中添加示警媒体

图 4.15

4.9 添加用户/用户组

4.9.1 管理→用户→创建用户组

图 4.15

4.9.2 *添加用户组许可权*

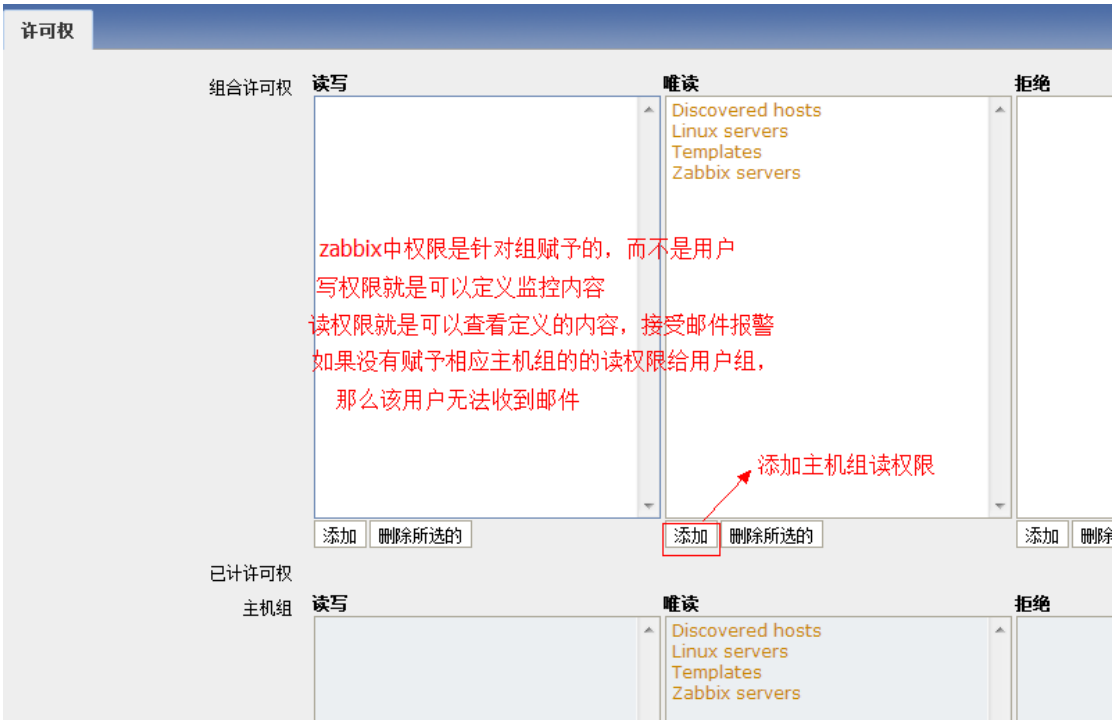


图 4.15

4.9.3 *创建用户到系统运维组*

管理→用户→右上角选择用户→创建用户

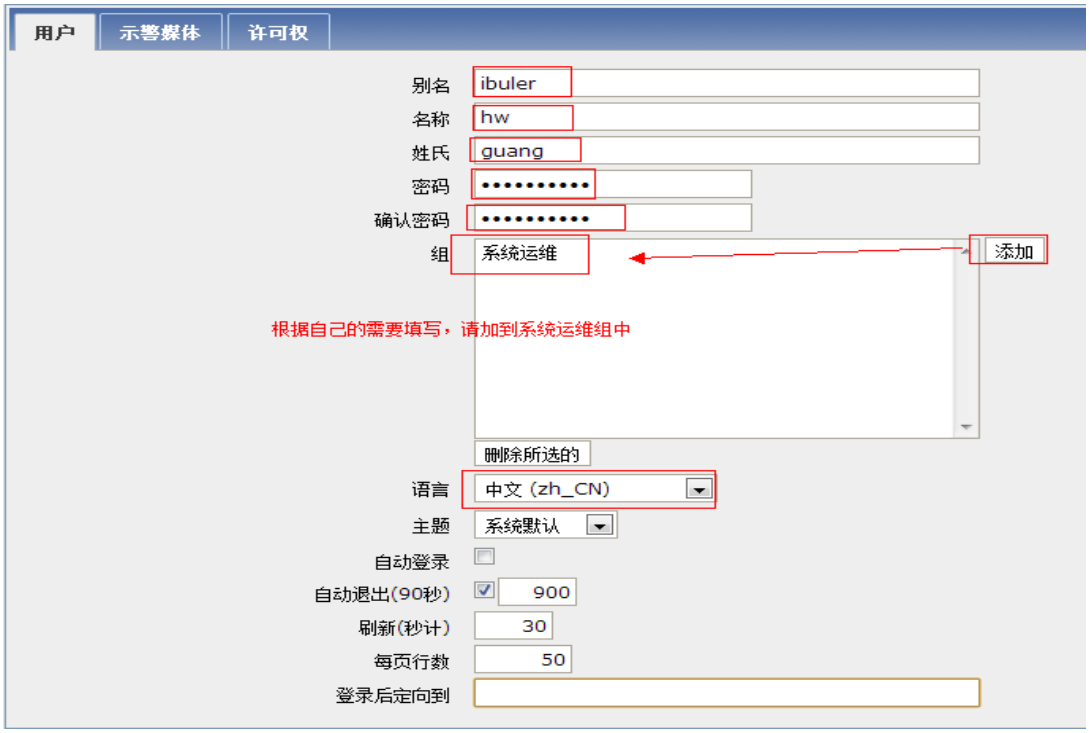


图 4.16

4.9.4 添加示警媒体

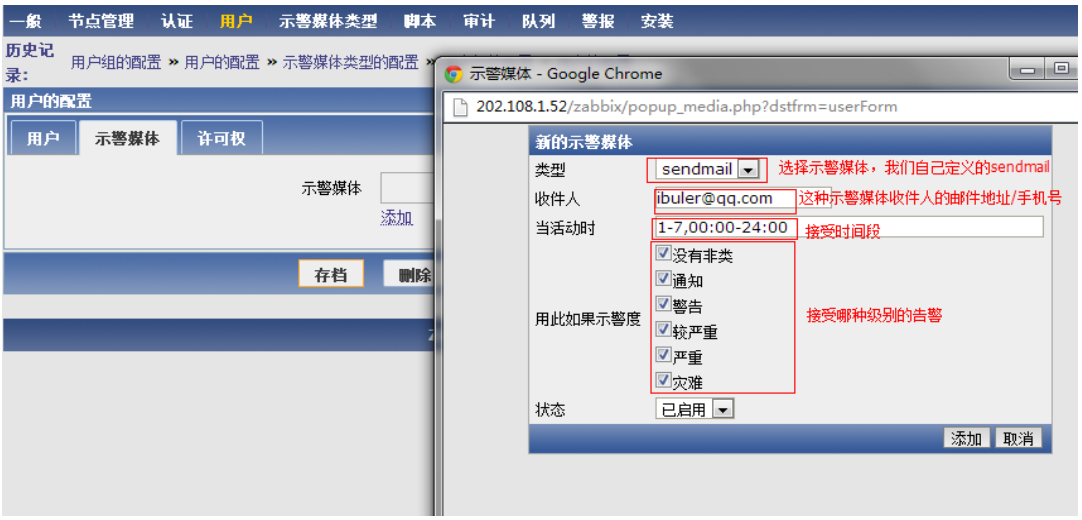


图 4.17

方法一样，添加 mailx，与飞信方式示警媒体，飞信收件人是手机号

图 4.18

4.10 添加报警动作

需求：当状态为 problem,并且状态没在维护中时邮件报警，每隔半小时发一次，最多发五次，问题解决后停止发送。

4.10.1 添加动作

配置→动作→触发器→创建动作

图 4.19

4.10.2 添加触发动作的条件

图 4.20

4.10.3 添加触发的操作

图 4.21

4.11 测试报警

4.11.1 修改内存使用率触发器到正常，再到问题状态，测试报警与否

图 4.22

图 4.23

图 4.24

同理可以测试一下其他方式报警！
到此整个流程相信你已经熟悉了！

第五章 进阶设置

5.1 key 的说明

5.1.1 key 的格式

net.tcp.service[service,<ip>,<port>]

net.tcp.service 是 key 的名字
[]里面跟的是 key 的参数
service 是第一个参数，没有<>代表不可省略
ip 是第二个参数，有<>代表可以省略，省略使用默认值
port 是第三个参数，也可以省略，当前面参数省略，而后面参数不省略，”,”用逗*号分隔*。
如： net.tcp.service[ssh] 监测 ssh 服务，Ip 省略默认是被监控主机的 ip
net.tcp.service[ssh,202.108.1.51] 监测 202.108.1.51 上端口
net.tcp.service[ssh,,2201] 主机 ssh 服务非 22 端口，中间 ”,”**不能省略**

5.1.2 key 的测试

zabbix server 安装完成后会在 bin 目录下生成一个 zabbix_agent 的执行文件，通过它来测试 key
./zabbix_get -s 202.108.1.51 -k net.tcp.service[ssh]
-s 后面指定服务器地址
-k 后面制定使用的 key

5.1.3 常用 key 列举

agent.hostname 返回 agent 定义的 hostname
agent.ping 监测 agent 能否 ping 通，能返回 1，不能返回 0
agent.version 返回 agent 的版本
net.tcp.listen[port] 监测是否监听端口
net.tcp.port[<ip>,port] 监测是否监听端口
net.tcp.service[service,<ip>,<port>] 监测是否启动该服务
net.tcp.service.perf[service,<ip>,<port>] 监测该服务性能
net.udp.listen[port] 监测是否监听该端口 UDP
proc.num[<name>,<user>,<state>,<cmdline>] 返回进程数量

CPU 监控：

system.cpu.load[<cpu>,<mode>]	
mode ▲	avg1 *(default)*
	avg5
	avg15
cpu ▲	percpu
	all

avg1,avg5,avg15 分别是 1,5,15 分钟平均负载
percpu 每个 cpu 的负载
all 所有 cpu 的负载

system.cpu.util[<cpu>,<type>,<mode>]	
type ▲	user *(default)*
	nice
	idle
	system
	kernel
	iowait
	wait
	interrupt
	softirq
	steal
mode ▲	avg1 *(default)*
	avg5
	avg15

type 的内容与 top 中的值一致，不再细说

网卡监控：

net.if.in[if,<mode>]	
mode ▲	bytes *(default)*
	packets
	errors
	dropped

网卡进流量数值
if 指定网卡
mode 分别代表的字节，包，错误，丢弃的数量

net.if.out[if,<mode>]	
mode ▲	bytes *(default)*
	packets
	errors
	dropped

网卡出的流量数值，与上相同

net.if.total[if,<mode>]	
mode ▲	bytes *(default)*
	packets
	errors
	dropped

网卡总的流量，与上相同

磁盘监控：

vfs.dev.read[<device>,<type>,<mode>]	
type ▲ *(defaults are different under various OSes)*	sectors
	operations
	bytes
	sps
	ops
	bps
mode ▲ *(compatible only with*	avg1 *(default)*
	avg5

type in: sps, ops, bps)	avg15

硬盘读取性能监控

vfs.dev.write[<device>,<type>,<mode>]	
type ▲ *(defaults are different under various OSes)*	sectors
	operations
	bytes
	sps
	ops
	bps
mode ▲ *(compatible only with type in: sps, ops, bps)*	avg1 *(default)*
	avg5
	avg15

硬盘写性能监控

vfs.fs.inode[fs,<mode>]	
mode ▲	total *(default)*
	free
	used
	pfree
	pused

磁盘盘 inode 使用情况

fs 磁盘分区

total 总的 inode 数量

free 空闲的 inode 数量

used 使用的 inode 数量

pfree inode 空闲率

pused inode 使用率

vfs.fs.size[fs,<mode>]	
mode ▲	total *(default)*
	free
	used
	pfree
	pused

磁盘空间占用情况

fs 磁盘分区

内存：

vm.memory.size[<mode>]	
mode ▲	total *(default)*
	free
	used
	shared
	buffers
	cached
	pfree
	pused
	available

内存使用情况
total 总的内存容量
free 空闲内存量
used 使用的内存量
pfree 空闲率
pused 使用率
available 可供使用的，是 free 与 buffer,cache 的和

system.swap.size[<device>,<type>]	
type ▲	free *(default)*
	total
	used
	pfree
	pused

swap 使用情况，与上相同
所有 key 请见 https://www.zabbix.com/documentation/1.8/manual/config/items

5.1.4 自定义 key

有时候我们需要自定义需要的监测服务或其它，比如监控 mysql，这是我们就需要自定义 key 了。

a) 编辑 agent 的配置文件 zabbix_agentd.conf

UserParameter=TestMysql,mysql -uzabbix -predhat -e 'show status' &>/dev/null && echo 1 ##最后添加一行，其中",”前面的 TestMysql 是 key，后面是命令(当然也可以
##是执行一个脚本)，也就是当我们调用 key 值时会将后面命令的值返回
##当能连接 mysql 时返回值 1，否则不返回值

b) 重启 zabbix_agentd
service zabbix_agentd restart
c) 测试该 key
[root@xk-1-52-a8 bin]# ./zabbix_get -s 202.108.1.51 -k TestMysql
1
d) 自定义 key 成功，可以在监控时调用该 key

5.2 触发器

5.2.1 表达式操作符

+ - * / 加，减，乘，除
< > # = 小于，大于，不等于，等于

5.2.2 函数

avg 平均值
{202.108.1.51:vm.memory.size[pused].avg(60)}>80 60s 内平均值大于 80
last 最后收到的值
{202.108.1.51:vm.memory.size[pused].last(0)}>90 最后收到的值大于 90
{202.108.1.51:vm.memory.size[pused].last(#6)}>90 最后收到的第 6 个值大于 90
{202.108.1.51:vm.memory.size[pused].last(0,60)}>90 60s 前收到的值大于 90
nodata 没有收到数据
{202.108.1.51:vm.memory.size[pused].nodata(60)}=1 60s 内没收到值该表达式为真
sum 求和
{202.108.1.51:vm.memory.size[pused].sum(60)}>100 60s 内收到值的和大于 100 为真
{202.108.1.51:vm.memory.size[pused].sum(#4)}>100 最后 4 个值和大于 100 为真
min 最小值
{202.108.1.51:net.if.in[eth0,bytes].min(300)}>100K 5 分钟内流入流量最小大于 100K 为真
count 计数
{202.108.1.51:icmpping.count(1800,0)}>5 三分钟内收到 0 的数量大于 5 为真

详见 https://www.zabbix.com/documentation/1.8/manual/config/triggers

5.2.3 触发器依赖关系

内存使用率→agent.ping→前端交换机
如上所示：三个触发器存在依赖关系，如果不定义依赖关系，前端交换机宕后，三个触发器都是 problem 状态，我们从邮件中无从得知是哪的问题，所以定义依赖关系很有必要。

5.3 通过 SNMP 监控交换机

需求：我们需要监控公司 H3C 5500 交换机

5.3.1 建立交换机模板

5.3.2 添加监控项

5.3.3 添加其它

对于触发器，图形与服务器监控一致

5.4 自动发现

自动发现提供了批量添加主机方法，整个过程是 zabbix_server 根据自动发现的规则，去扫描被监控端，如果符合要求，便可以被发现，配合自动发现的动作批量添加主机。案例：批量添加监控交换机

5.4.1 配置自动发现

配置→发现→创建发现规则

图5.2

名称：发现规则的名称

ip 范围：需要扫描的 ip 范围段，支持掩码格式 202.108.2.0/24

延迟：多长时间间隔再次扫描

检查类型：以什么方式扫描能确定添加到需要的主机，交换机用 snmp 比较好了

端口：默认 snmp 是 161

SNMP community：交换机的 community

OID：检查的 OID，如果存在则被发现

设备唯一性：通过什么来确定设备的唯一性，一般 IP 比较好

5.4.2 添加自动发现动作

配置→动作→发现→创建动作

a) 填写名称

图 5.3

b) 添加触发动作的条件

图 5.4

c) 添加操作

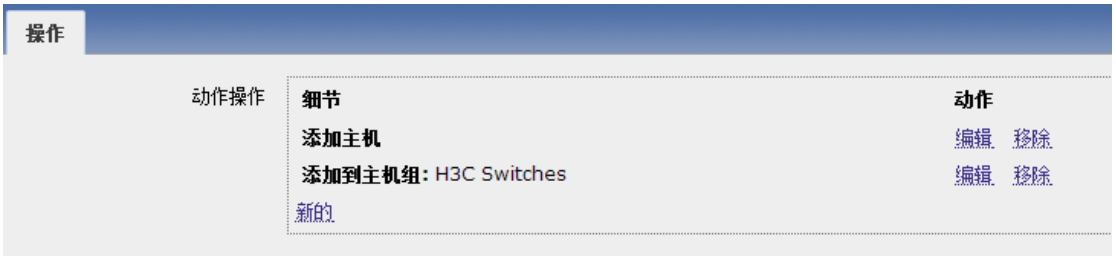


图 5.5

d) 查看是否自动发现并添加了监控交换

图 5.6

5.5 自动注册

自动注册的流程是 agent 向 server 发送注册信息，server 端根据自动注册条件添加主机，所以自动注册需要客户端安装 agent

5.5.1 添加自动注册动作

配置→动作→自动注册→创建动作

5.5.2 确定动作名称

图 5.7

5.5.3 添加自动注册条件

图 5.8

5.5.4 添加操作

图 5.9

5.5.5 验证查看自动注册的主机

5.6 低等级自动发现

低等级自动发现主要用来监控磁盘占用，网卡流量，SNMP 等监控的相同的内容，但项目对象不确定的情况。如：磁盘占用，主机和主机上磁盘分区是不一样的，我们不能定义一个模板来监控所有主机的磁盘占用，低等级自动发现就应用与该场景。

5.6.1 创建低等级自动发现

模板列表→(点击需要建立规则的模板上)发现→创建发现规则

图 5.10

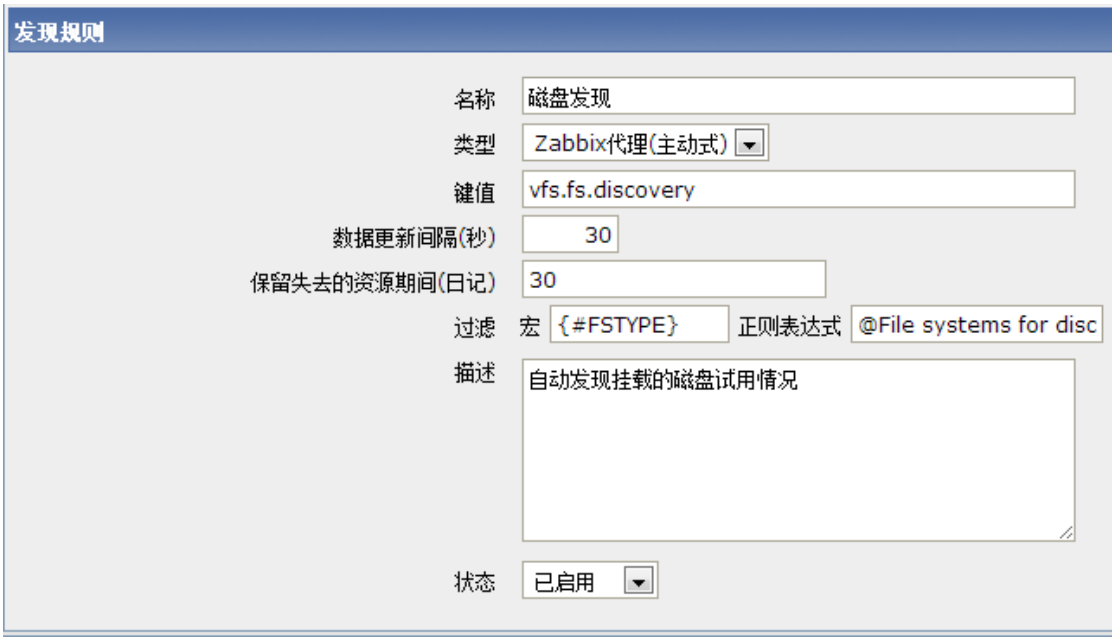


图 5.11

名称：低等级自动发现的名称

类型：监控的类型，与以前的一样

键值：低等级自动发现的 key，可以通过 zabbix_get 测试一下，返回 Json 数据

过滤：通过正则表达式来过滤

宏：key 中定义的宏，通过测试 key 可以查看到

正则表达式：定义的正则表达式，用来过滤不需要的宏内容，管理→一般→正则表达式中定义正则表达式

5.6.2 创建项目原型

在刚才建立的低等级自动发现→选择项目原型→创建项目原型

图 5.12

名称：项目的名称
键值：与正常的 key 无异，只不过 fs 用一个宏来代替而已
其它：与正常定义无异

5.6.3 查看自动发现的项目

在关联主机的项目里面查看是否发现了我们定义的磁盘占用项目

磁盘发现: /空间使用率	触发器 (1)	vfs.fs.size[/,pused]	30	90
磁盘发现: /boot空间使用率	触发器 (1)	vfs.fs.size[/boot,pused]	30	90

图 5.13

5.6.4 创建触发器原型

选择触发器原型→创建触发器原型

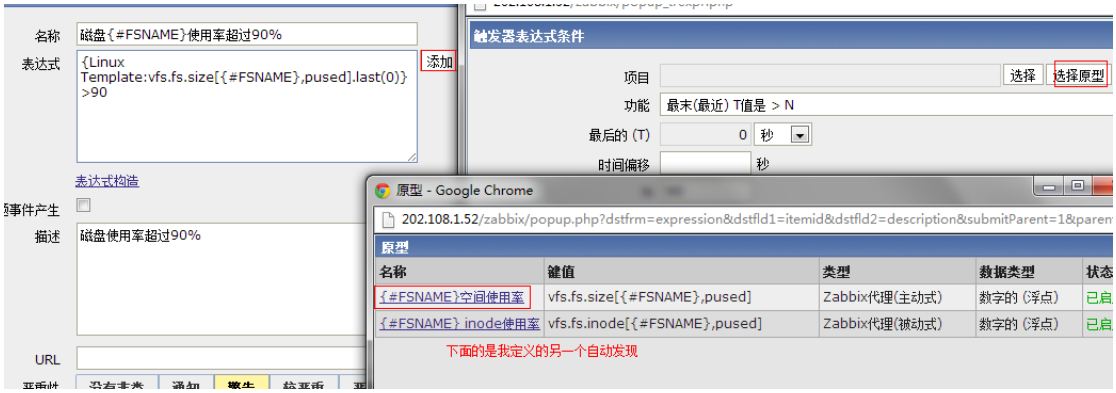


图 5.14

5.6.5 *定义图形原型*

选择该自动发现上的图形原型→创建图形原型 与建立普通图形一样，只不过下面是添加原型而已，名称一定要带宏，否则生不成图像，因为多个图形名称一致

图 5.15

5.6.6 *在模板关联的主机查看相关触发器和图形*

5.7 简单方式监控

通过简单监控我们可以监控主机的存活，某些服务是否开启，被监控端不需要安装 agent

5.7.1 *编译安装 fping*

简单监控通过 fping 来实现的

wget http://www.fping.org/dist/fping-3.5.tar.gz

tar xvf fping-3.5.tar.gz

cd fping-3.5

./configure && make && make install

5.7.2 修改 zabbix server 配置文件 zabbix_server.conf，制定 fping 的位置

vim /usr/local/zabbix/etc/zabbix_server.conf

FpingLocation=/usr/local/sbin

5.7.3 由于普通用户是没有全新构建 icmp 包的所以给与 fping setuid 权限

chmod u+s /usr/local/sbin/fping

5.7.4 重启 zabbix

service zabbix_server restart

5.7.5 web 端配置简单监控

在 Linux Template 模板中添加简单监控项

键值：icmpping[<target>,<packets>,<interval>,<size>,<timeout>] 成功返回 1，失败返回 0

target ： ip 地址

packets ： 数据包数量

interval ：间隔时间
size：包的大小
timeout：超时时间

5.7.6 在最新数据查看是否获得数据
5.7.7 定义触发器（略）

5.8 监控 Windows 主机

监控 windows 主机方式与 Linux 相同，只不过其中的 key 可能有所不同，请参加 windows 的模板，与 key 的详细列表
https://www.zabbix.com/documentation/1.8/manual/config/items

5.9 screen 查看多种图像

通过自定义 screen 我们可以将多种图形以及其它元素整合到一个界面上

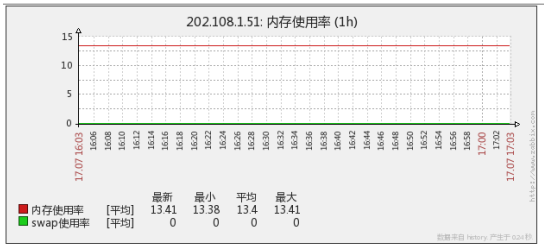
5.9.1 定义 screen

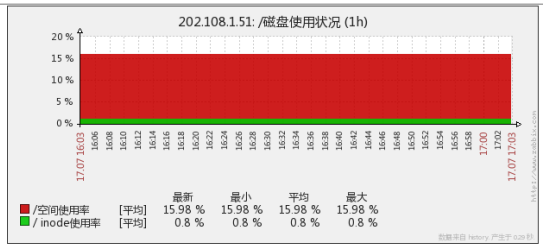
配置→筛选→创建筛选

5.9.2 添加元素

点击定义的 screen 的名称进入

5.9.3 显示结果





更改

更改