**zabbix报警机制**

基本概念

概念介绍

自定义的监控默认不会自动报警

首页也不会提示错误

需要配置触发器与报警动作才可以自定报警

触发器

表达式，如内存不足300M，用户超过30个等

当触发条件发生后，会导致一个触发事件

触发事件会执行某个动作

动作

触发器的条件被触发后的行为

可以是发送邮件、也可以是重启某个服务等

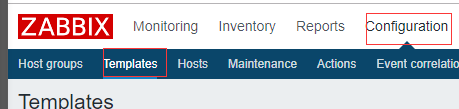
触发器

创建触发器

通过configuration->templates

选择模版后面的triggers->create trigger

强烈建议使用英文创建





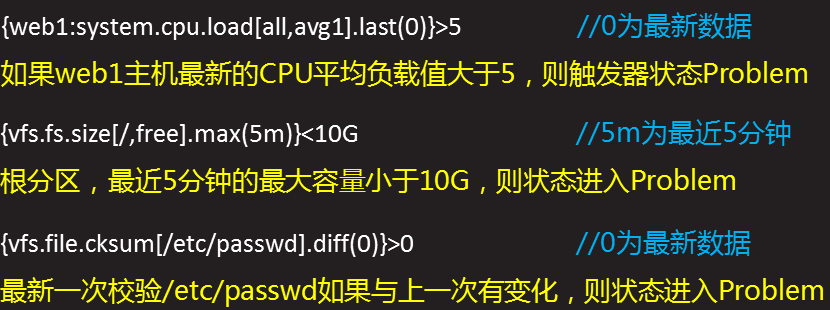
触发器表达式

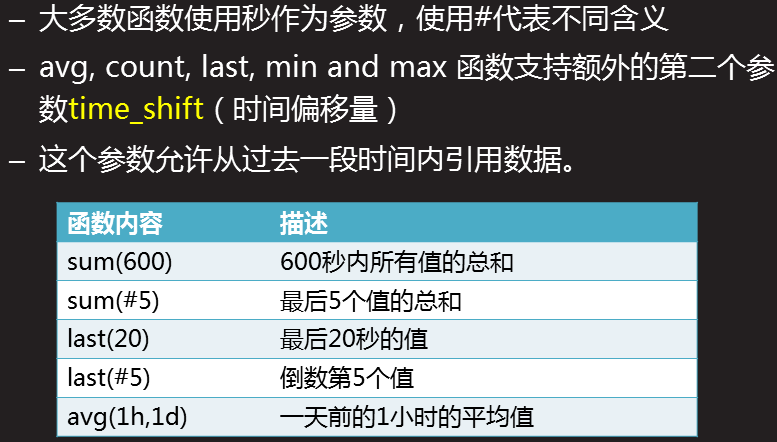
expression表达式：触发异常的条件

{<server>:<key>.<function>(<parameter>)}<operator><constant>

{主机：key.函数(参数)}<表达式>常数

触发器表达式案例



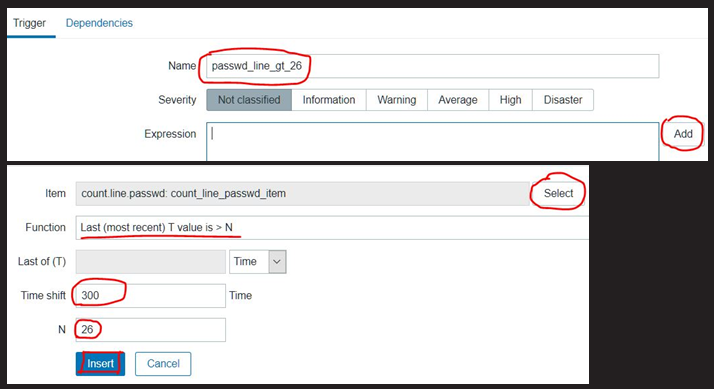


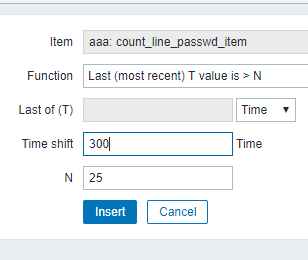
配置触发器

设置触发器名称，点击add添加表达式

填写表达式

监控为账户数量，最近300秒账户数量大于26

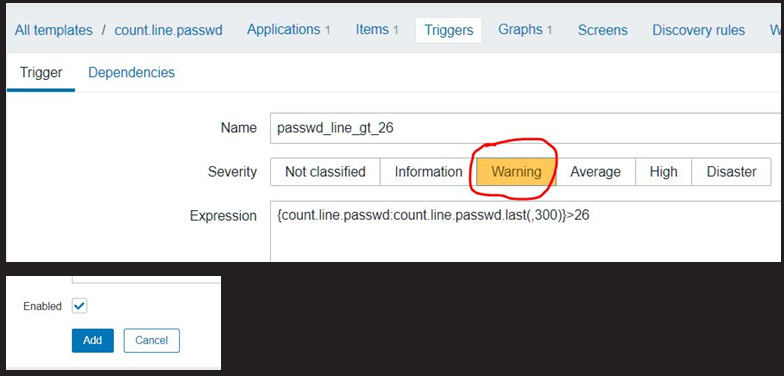




item监控项、function函数、N值、time shift时间偏移量

选择触发器报警级别

add创建该触发器





设置邮件

创建media

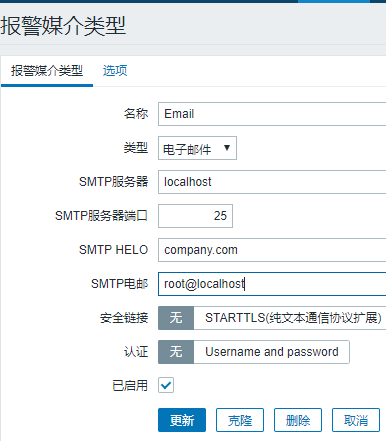
设置邮件服务器

administration->media type->选择email邮件

设置邮件服务器信息







为账户添加media

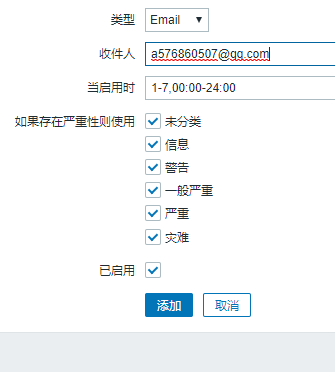
在administration->users中找到admin账户



选择media菜单->点击add添加报警媒介

在meida type中填写报警类型，收件人，时间等信息







创建active

active（行为）

定义当触发器被触发时，执行什么action

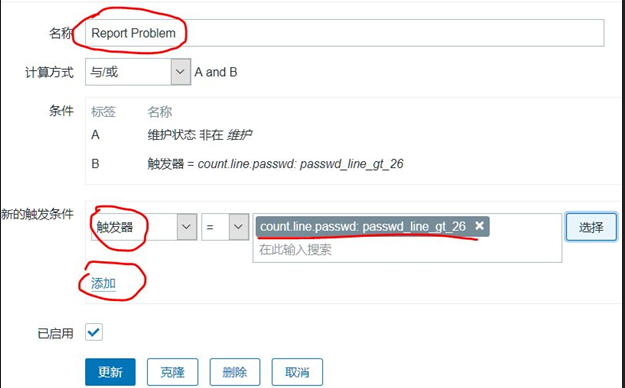
通过configuration->actions->create action创建





配置action（填写名称）

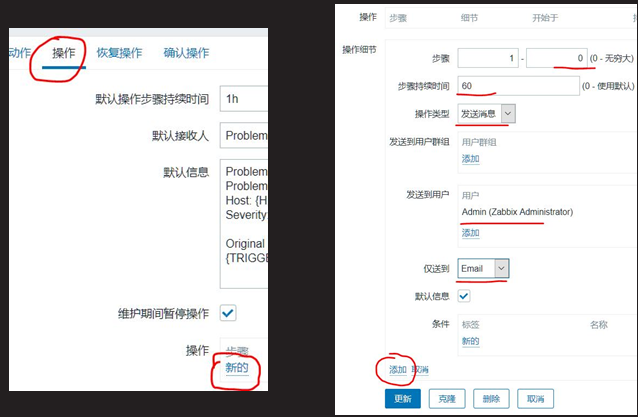
配置导致动作的触发条件（账户大于26）





配置动作的具体操作行为（发送信息或执行远程命令）

无限次数发送邮件，60秒1次，发送给admin用户





**zabbix进阶操作**

概述

自动发现

当abbix需要监控的设备越来越多，手动添加监控设备越来越有挑战，此时，可以考虑自动发现功能

自动发现可以实现：

自动发现，添加主机，自动添加主机到组

自动连接模版到主机，自动创建监控项目与图形等

自动发现流程

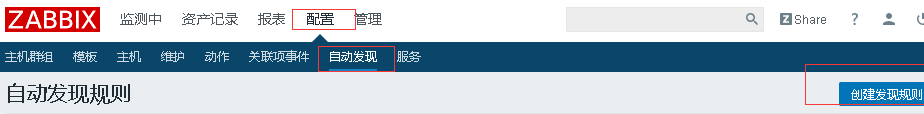
创建自动发现规则

创建action动作，说明发现主机后自动执行什么操作

通过动作，执行添加主机，连接模版到主机等操作

自动发现规则

创建自动发现规则：configuration->discovery->create discovery rule

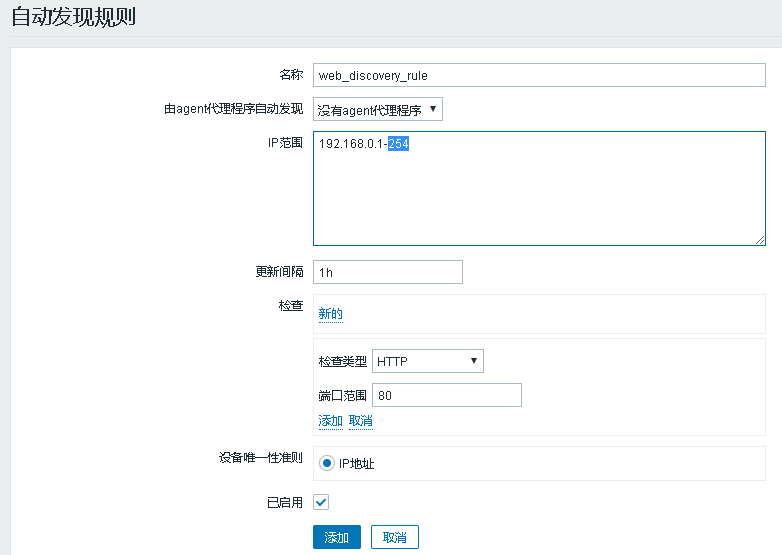


填写规则

自动发现的IP范围（逗号隔开可以写多个）

多久发现一次（默认为一小时，仅实验修改为1m）

检查的方式：http ftp agent的自定义key等检查



创建动作

configuration->actions

event source(discovery)->create action

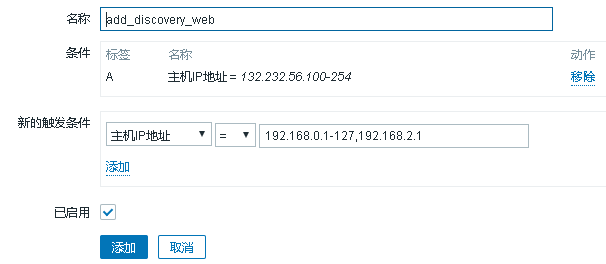
注意：选择事件源为：自动发现



添加动作名称，添加触发动作的条件

操作（触发动作后要执行的操作指令），操作细节如下：

添加主机到组，与模版链接（HTTP模版）





创建新主机

创建一台新的主机，验证zabbix是否可以自动发现该主机

登录zabbix服务器的web页面，查看主机列表



主被动监控

概述

主动和被动都是对被监控端主机而言的

默认zabbix采用的是被动监控

被动监控：sever向agent发起连接，发送监控key，agent接受请求，响应监控数据

主动监控：agent向server发起连接，agent请求需要检测的监控项目列表，server响应agent发送一个items列表，agent确认收到监控列表，TCP连接完成，会话关闭，agent开始周期性地手机数据即可

区别：server不用每次需要数据都连接agent，agent会自己收集数据并处理数据，sercer仅保存数据即可

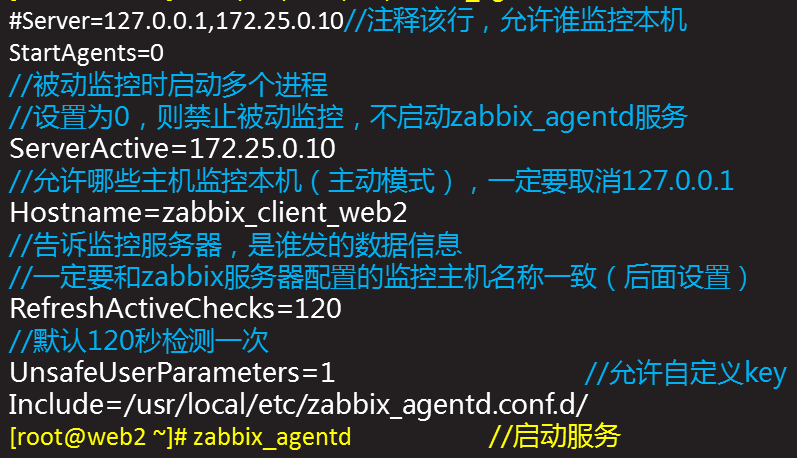
当监控主机达到一定量级后，Zabbix服务器会越来越慢

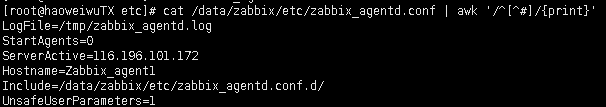
此时，可以考虑使用主动监控，释放服务器的压力

另外，zabbix也支持分布式监控，也是可以考虑的方案

创建新的被监控主机，并安装agent软件

修改配置文件





克隆模版

为了方便，克隆系统自带模版（在此基础上就该更方便）

configuration->templates

选择template os Linux

全克隆该模版，新建一个新的模版

新模版名称为：template os Linux serveractive





修改监控项模式

将模板中的所有监控项目全部修改为主动监控模式

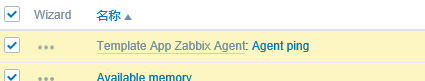
configuration->templates

选择新克隆的模版，点击后面的items（监控项）

点击全选，选择所有监控项，点击批量更新

将类型修改为：zabbix agent（active主动模式）







批量修改监控项的监控模式后，并非所有监控项目都支持主动模式

批量修改后，会发现有几个没有修改主动模式成功

说明，这些监控项目不支持主动模式，关闭即可

可以点击类型排序，方便操作，点击状态关闭即可关闭



添加监控主机

在zabbix监控服务器中，添加被监控的主机（主动模式）

名称与被监控端的配置文件hostname一致，将主机添加到linux servers组

IP地址为0.0.0.0，端口为0，不填写IP无法创建成功

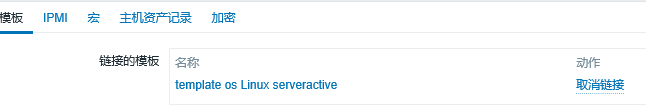
为主机添加监控模版

选择刚刚创建的模版（主动模式）

添加链接模版到主机





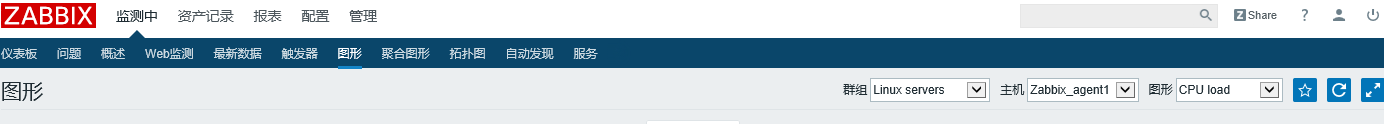


验证效果

查看数据图表

monitoring->graphs

选择需要查看的主机组、主机以及图形



查看图表时无数据

因为分区数据采用的是自动发现监控，与普通监控一样，改为主动模式即可

选择template os linux serveractive模版

修改discovery自动发现为主动模式





拓扑图与聚合图形

拓扑图

绘制拓扑图可以快速了解服务器架构

monitoring->maps（拓扑图）

选择默认的local network拓扑图，编辑即可





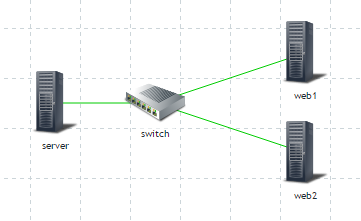
操作说明

icon（图形），添加新的设备后可以点击图标修改属性

shape（形状）

link（连线），先选择两个图标，再选择连线

完成后，点击update（更新）



聚合图形

在一个页面显示多个数据图标，方便了解多组数据

monitoring->screens（聚合图形）->create screen

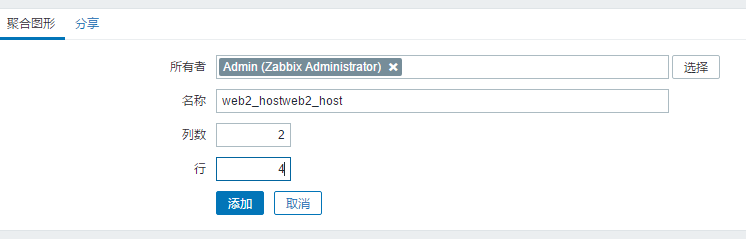
owner：使用默认的admin用户

name：名称设置为web2\_host

columns：列数设置为2列

rows：行数设置为4行

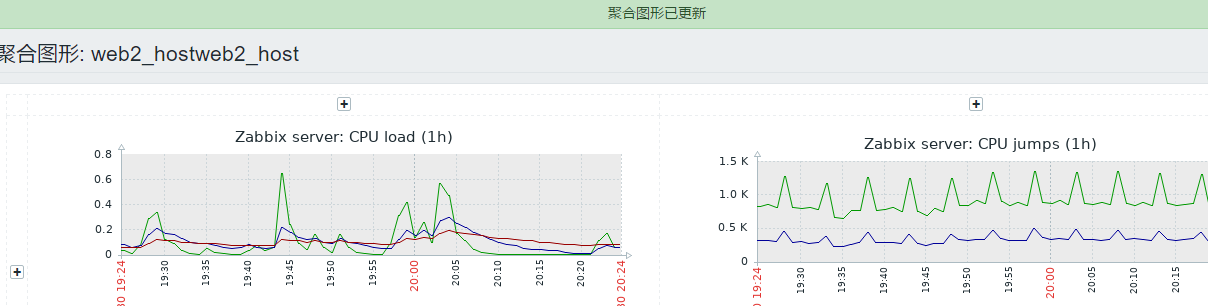




选择刚刚创建的聚合图形

点击后面的构造函数

点击change（更改），设置每行每列需要显示的数据图表



监控案例

监控nginx

环境要求

一台nginx服务器，部署nginx时要加载status模块

cd nginx

./configure --with-http\_stub\_status\_module

make && make install

vim nginx.conf

location /status {

stub\_status on;

}

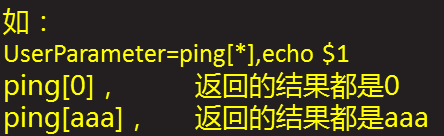
自定义监控key

语法格式

userparameter=key,command

userparameter=key[\*],<command>

key里的所有参数，都会传递给后面命令的位置变量



被监控端修改监控配置文件

注意要允许自定义Key并设置include

vim etc/zabbix\_agent.conf.d/nginx.status

UserParameter=nginx.status[\*],/usr/local/bin/nginx\_status.sh $1

killall zabbix\_agentd

zabbix\_agentd

编写脚本（仅供参考，为检测完整状态）

vim /usr/local/bin/nginx\_status.sh

#!/bin/bash

case $1 in

active)

curl -s <http://127.0.0.1/status> | awk ‘/Active/{print $NF}’;;

waiting)

curl -s <http://127.0.0.1/status> | awk ‘/Waiting/{print $NF}’;;

accepts)

curl -s <http://127.0.0.1/status> | awk ‘NR==3{print $2}’;;

esac

chmod +x /usr/local/bin/nginx\_status.sh

测试效果

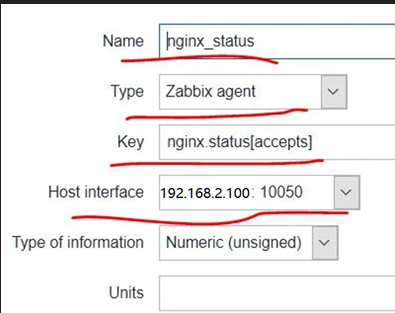
zabbix\_get -s 127.0.0.1 -l ‘nginx.status[accepts]’

监控nginx

在监控服务器，添加监控项目item

configuration->hosts->点击主机后面的items

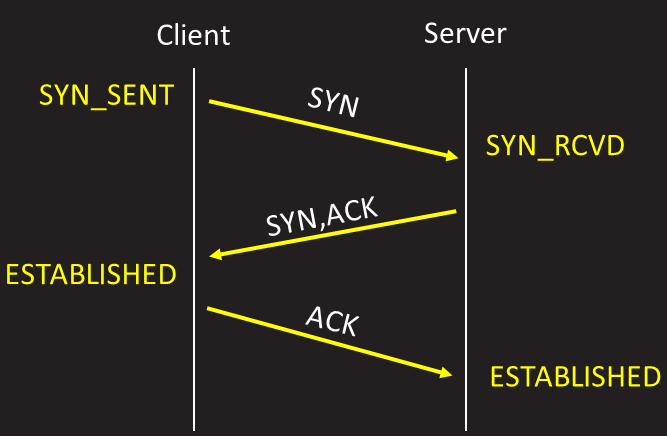
点击create item



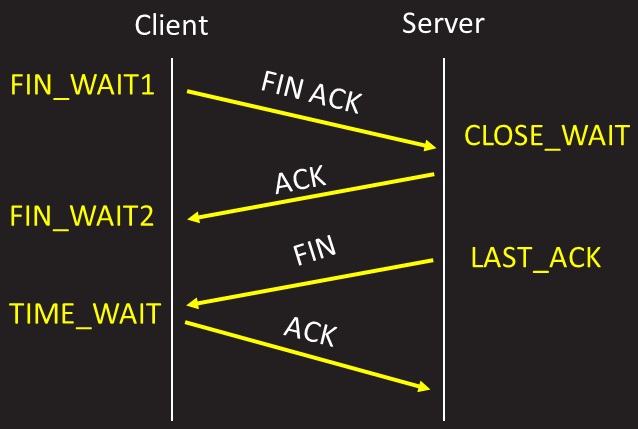
监控网络连接状态

TCP连接状态

建立连接的3次握手



断开连接的4次握手



查看网络连接状态

模拟多人并发连接

ab -c 1000 -n 100000 <http://192.168.2.100>

查看网络连接状态

ss -antup

-a显示所有

-t显示TCP连接状态

-u显示udp连接状态

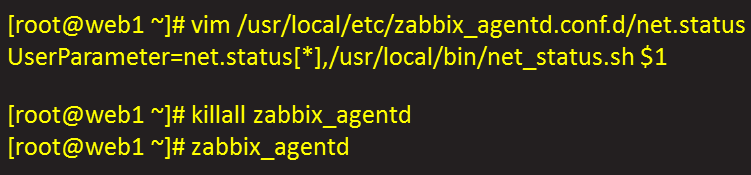
-n以数字形式显示端口号和IP地址

-p显示连接对应的进程名称

自定义监控key

被监控端修改配置文件

注意要允许自定义key并设置inlude



编写脚本

vim /usr/local/bin/net\_status.sh

#!/bin/bash

case $1 in

estab)

ss -antp | awk ‘/^TIME-WAIT/{x++}END{print x}’;;

closer\_wait)

ss -antp | awk ‘/^CLOSE-WAIT/{x++}END{print x}’;;

time\_wait)

ss -antp | awk ‘/^TIME-WAIT/{x++}END{print x}’;;

esac

chmod +x /usr/local/bin/net\_status.sh

zabbix\_get -s 127.0.0.1 -k ‘net.status[time\_wait]’

监控netstatus

在监控服务器，添加监控项目item

configuration->hosts->点击主机后面的items

点击create item



其他课外扩展

常见监控需求

mysql（mysqladmin命令）

并发连接数

慢查询数量

增删改查数量等

nosql数据库（数据库状态）

php-fpm（生成status页面）

并发、队列、进程数量等

tomcat（服务器状态）

硬件设备（交换机、路由器等）