**监控与服务安全**

安全: 系统常规加固 服务安全 数据安全 网路安全

环境:

zaserver: eth1: 192.168.2.5

zaweb1 : eth1: 192.168.2.100

zaweb2: eth1: 192.168.2.200

一、监控概述

1.监控的目的

a)报告系统运行状况

每一部分必须同时监控

内容包括 吞吐量、反应时间、使用率等

b)提前发想问题

进行服务器性能调整前,知道调整什么

找出系统的瓶颈在什么地方

2.监控的资源类别

a)公开数据

Web、FTP、数据库等应用服务

TCP或UDP端口

b)私有数据

CPU、内存、磁盘、网卡流量等使用信息

用户、进程等运行信息

3.监控软件

a)系统监控命令

ps 查看系统的进程

ifconfig 查看网卡的信息

uptime 查看cpu的负载

netstat或ss 查看进程的状态

free 查看内存

ping 查看网络连通性的

swapon -s 交换信息

traceroute 路由追踪,显示到达目标主机的路由所经过的信息

(安装包traceroute)

df -h 查看分区的使用情况的

iostat (sysstat) 查看硬盘读写速度

(安装包sysstat)

b)自动化监控系统

Cacti (仙人掌)

基于SNMP协议的监控软件,强大的绘画能力

Nagios

基于Agent监控,强大的状态检查与报警机制

可以通过短信,有邮件方式发送报警,只能监控状态

Zabbix(开源软件)

基于多种监控机制,支持分布式监控

是Cacti和Nagios的结合,有绘图能力又有状态检查和报警能力

1. 常用系统监控命令

使用系统命令查看系统性能参数

--查看内存信息

--查看交换分区信息

--查看磁盘信息

--查看CPU信息

--查看网卡信息

--查看端口信息

--查看网络连接信息

二、Zabbix基础

1简介:

是一个高度集成的监控解决方案,

可以实现企业级的开源分布式监控

通过C/S模式采集监控数据

通过B/S模式实现Web管理

c:client 客户端 s:server 服务端 b:浏览器首字母

c/s模式:客户端服务器模式 监控数据 B/S模式:浏览器服务器模式管理

2监控拓扑

监控服务器

监控服务器可以通过SNMP或Agent采集数据

数据可以写入MySQL、Oracle等数据库中

服务器使用LNMP实现web前段的管理

被监控主机

被监控主机需要安装Agent

常见的网络设备一般支持SNMP

可以监控什么样的设备,监控的对象:

SNMP:简单网络管理协议的设备(比如路由器,交换机)

linux,Windows,虚拟操作系统,网站,数据库,邮件,各种硬盘cpu,内存

三、部署环境:

[student@room9pc01 03]$ cd

[student@room9pc01 ~]$ cd /linux-soft/03/

[student@room9pc01 03]$ for i in {5,100,200}

do

scp -r Zabbix/ root@192.168.2.$i:/root/

done

cd Zabbix/

rm -rf php-fpm-5.4.16-42.el7.x86\_64.rpm

rm -rf php-mbstring-5.4.16-42.el7.x86\_64.rpm

rm -rf php-bcmath-5.4.16-42.el7.x86\_64.rpm

rm -rf libevent-devel-2.0.21-4.el7.x86\_64.rpm

**1.部署LNMP**

监控服务器

主机名(zabbix server) IP地址(192.168.2.5) 关闭防火墙、SElinux

监控客户端(2.100和2.200)

web1 (192.168.2.100)

web2 (192.168.2.200)

[root@zzserver ~]# yum -y install gcc pcre-devel zlib-devel openssl-devel

[root@zzserver zabbix]# tar -xf nginx-1.12.2.tar.gz

[root@zzserver zabbix]# cd nginx-1.12.2/

[root@zzserver nginx-1.12.2]# ./configure --with-http\_ssl\_module

[root@zzserver nginx-1.12.2]# make && make install

[root@zzserver nginx-1.12.2]# yum -y install php php-mysql php-fpm

mariadb mariadb-devel mariadb-server

[root@zzserver ~]# vim /usr/local/nginx/conf/nginx.conf

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

起服务

[root@zzserver ~]# ln -s /usr/local/nginx/sbin/nginx /sbin/nginx

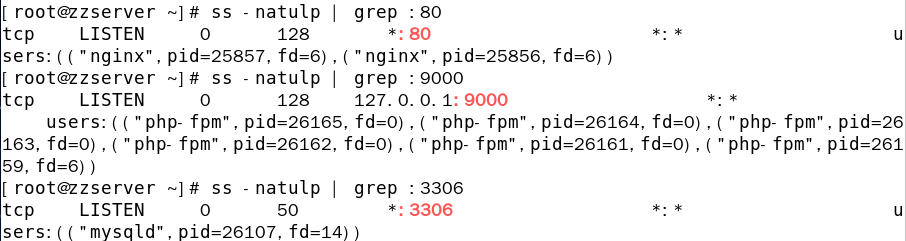
[root@zzserver ~]# nginx

[root@zzserver ~]# systemctl start mariadb (3306端口)

[root@zzserver ~]# systemctl start php-fpm (9000端口)

[root@zzserver ~]# systemctl enable mariadb php-fpm

查看是否有三个端口



写一个测试页面

[root@zzserver ~]# vim /usr/local/nginx/html/test.php

<?php

$i=33;

echo $i;

?>

[root@zzserver ~]# curl http://192.168.4.115/test.php

33

**2.部署源码zabbix软件**

2.1[root@zaserver ~]#

检查yum源是否有libevent-devel 依赖包

[root@zzserver ~]# yum list | grep -i libevent-devel

yum -y install net-snmp-devel curl-devel libevent-devel

cd Zabbix/

tar -xf zabbix-3.4.4.tar.gz

cd zabbix-3.4.4/

./configure

--enable-server //安装监控服务程序

--enable-proxy //安装分布式程序,做分布式监控用

--enable-agent //自己监控自己(agent客户端程序)

--with-mysql=/usr/bin/mysql\_config //mysql配置文件路径

--with-net-snmp //让zabbix支持简单网络管理协议

--with-libcurl //让zabbix支持用curl访问web程序

make install

ls /usr/local/etc/ //配置文件路径

ls /usr/local/bin/ //可执行命令路径

ls /usr/local/sbin/ //服务的启动程序路径

**2.2初始化准备.定义存放数据的库,**

准备存放数据使用的库,表,及连接用户,zabbix软件默认存放的库名就叫zabbix,用户是zabbix,密码也是zabbix, 表不用创建,从备份文件恢复就可以

[root@zzserver ~]# mysql

>create database zabbix character set utf8;

>grant all on zabbix.\* to zabbix@'localhost' identified by "zabbix";

>exit

[root@zzserver ~]#cd Zabbix/zabbix-3.4.4/database/mysql/

[root@zzserver ~]#mysql -uzabbix -pzabbix zabbix < schema.sql

[root@zzserver ~]#mysql -uzabbix -pzabbix zabbix < images.sql

[root@zzserver ~]#mysql -uzabbix -pzabbix zabbix < data.sql

**2.3上线zabbix页面(拷贝zabbix页面)**

[root@zzserver ~]#cd Zabbix/zabbix-3.4.4/frontends/php

[root@zzserver ~]#cp -a \* /usr/local/nginx/html/

[root@zzserver ~]#chmod -R 777 /usr/local/nginx/html/\*

**3初始化页面**

**3.1解决依赖**

打开真机浏览器输入网址 http://192.168.2.5/index.php 会报错

修改nginx配置文件

[root@zzserver ~]#vim /usr/local/nginx/conf/nginx.con

http{

fastcgi\_buffers 8 16k; //缓存php生成的页面内容,8个16K

fastcgi\_buffer\_size 32k; //缓存php生产的头部信息

fastcgi\_connect\_timeout 300; //连接PHP的超时时间

fastcgi\_send\_timeout 300; //发送请求的超时时间

fastcgi\_read\_timeout 300; //读取请求的超时时间

**3.2安装依赖软件**

[root@zzserver ~]#yum -y install php-gd php-xml php-ldap

php-bcmath php-mbstring

php-gd------------支持图片的软件包

php-xml

php-ldap

php-bcmath-----------\

php-mbstring---------/支持多字符集

**修改php进程的参数**

[root@zzserver ~]#vim /etc/php.ini

878 date.timezone =Asia/Shanghai //设置时区

384 max\_execution\_time = 300 //最大执行时间,秒

672 post\_max\_size = 16M //POST数据最大容量

394 max\_input\_time = 300 //服务器接收数据的时间限制

**重启服务**

[root@zzserver ~]#nginx -s stop

[root@zzserver ~]#nginx

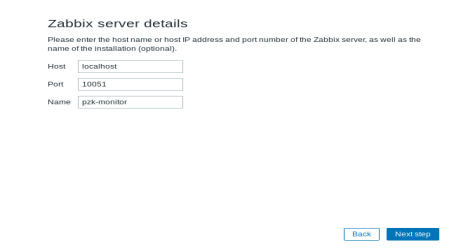
[root@zzserver ~]#systemctl restart php-fpm

打开真机浏览器输入网址 http://192.168.2.5/index.php

**点击---> Next step**

其他默认 密码:zabbix

**点击---> Next step**





用户: admin

密码:zabbix

**4修改配置文件**

(监控服务的主配置文件/usr/local/etc/zabbix\_server.conf)

[root@zaserver ~]# vim /usr/local/etc/zabbix\_server.conf

38 LogFile=/tmp/zabbix\_server.log //设置日志

85 DBHost=localhost //数据库主服务器

95 DBName=zabbix //设置数据库名称

111 DBUser=zabbix //设置数据库账户

119DBPassword=zabbix //设置数据库密码

**5起服务**

[root@zaserver ~]# useradd zabbix //添加用户(不创建用户无法启动)

[root@zaserver ~]# zabbix\_server //启动脚本

[root@zaserver ~]# netstat -utnlp | grep :10051 //查看进程

tcp 0 0 0.0.0.0:10051 0.0.0.0:\* LISTEN 4494/zabbix\_server

zabbix**报错**日志文件 /tmp/zabbix\_server.log

tail -f /tmp/zabbix\_server.log

存放**初始化**配置文件

cat /usr/local/nginx/html/conf/zabbix.conf.php

**三、Zabbix监控服务**

**1.基础监控**

1.1配置**被监控主机**

安装zabbix软件提供zabbix\_agentd服务

[root@zaweb1 ~]# yum -y install gcc pcre-devel

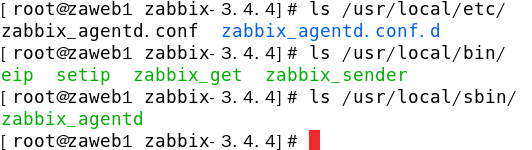
[root@zaweb1 ~]# cd Zabbix/

[root@zaweb1 Zabbix]# tar -xf zabbix-3.4.4.tar.gz

[root@zaweb1 Zabbix]# cd zabbix-3.4.4/

[root@zaweb1 zabbix-3.4.4]# ./configure **--enable-agent**

[root@zaweb1 zabbix-3.4.4]# make install



1.2修改服务的主配置文件

[root@zaweb1 zabbix-3.4.4]# vim /usr/local/etc/zabbix\_agentd.conf

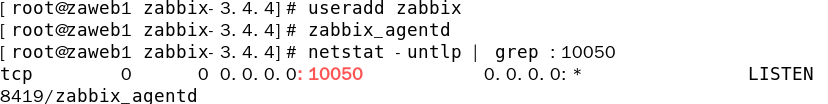
93 Server=127.0.0.1,192.168.2.5 //允许访问服务器地址表

134 ServerActive=192.168.2.5:10051 //监控服务器ip地址

30 LogFile=/tmp/zabbix\_agentd.log //日志文件

1.3启动服务

查看监控服务



2配置监控服务器,管理员登录管理页面作如下操作 <http://192.168.2.5/index.php>

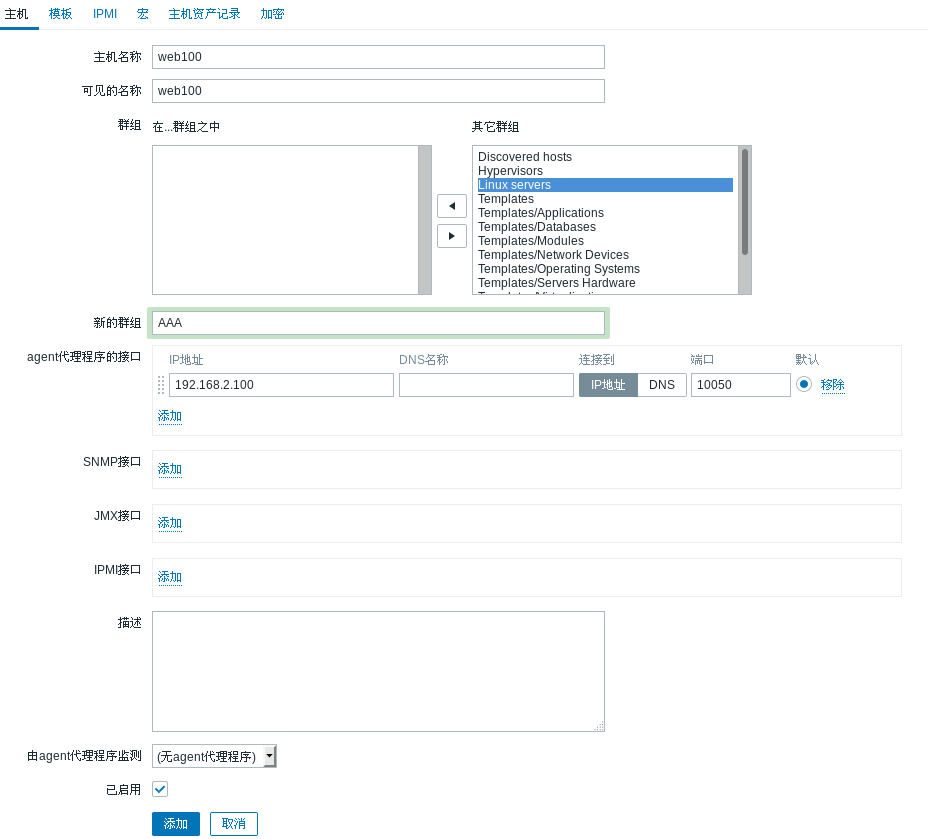
1 添加监控主机

2 选择监控模板

3 查看监控数据

**1)添加监控主机**

**点击---->配置--->主机--->创建主机**



**2) 选择监控模板**

**创建模板 添加 HTTP Linux 这两个模板**

**点web100---->模板--(添加 HTTP Linux 这两个模板)--->更新**

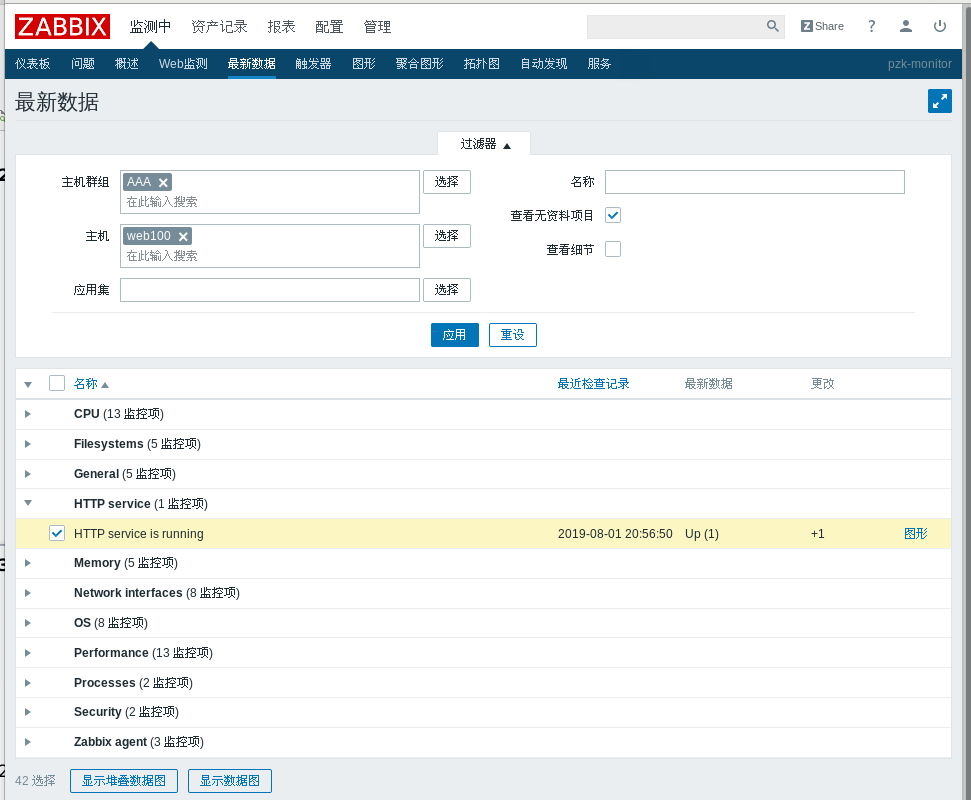


1. **查看监控数据**

[root@zaweb1 ~]# yum -y install httpd

[root@zaweb1 ~]# systemctl restart httpd

点击----->检测中-->最新数据-->(主机群 主机 应用)

****

**2.1自定义监控**:(**客户端定义监控命令给监控服务器使用**)

1.2.1配置客户端 192.168.2.100

**1 启用自定义配置文件**

[root@zaweb1 ~]# vim /usr/local/etc/zabbix\_agentd.conf

280 UnsafeUserParameters=1 /允许自定义key

265 Include=/usr/local/etc/zabbix\_agentd.conf.d/ //加载配置文件目录

**2.定义监控命令**

[root@zaweb1 ~]# cd /usr/local/etc/zabbix\_agentd.conf.d/

[root@zaweb1 zabbix\_agentd.conf.d]# vim count.line.passwd.conf

UserParameter=count.line.passwd,wc -l /etc/passwd | awk '{print $1}'

**3重启服务**

[root@zaweb1 zabbix\_agentd.conf.d]# killall -9 zabbix\_agentd

[root@zaweb1 zabbix\_agentd.conf.d]# zabbix\_agentd

[root@zaweb1 zabbix\_agentd.conf.d]# zabbix\_get -s 127.0.0.1 -k count.line.passwd

**22**

22:目前用户个数22个

2.2配置监控服务器

1 创建监控模板

2 创建应用集

3 创建监控项

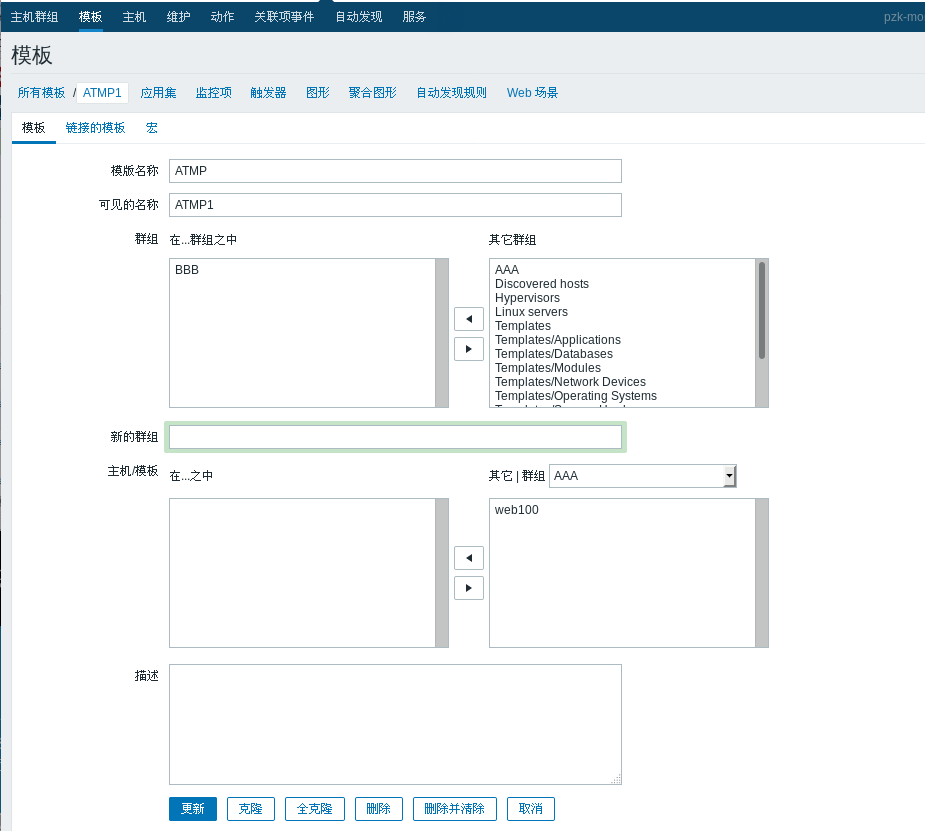
4 使用新模板监控主机100

5 查看监控数据

登录监控服务器web管理页面,做如下配置

1. 创建监控模板

点击 配置---->模板--->创建模板----(模板名称<ATMP1> 可见的名称<ATMP1> 新的群组<BBB> )更新



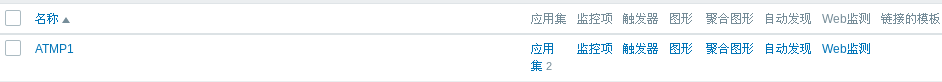
1. 创建应用集

点击 ATMP1里的--->应用集--->创建应用集

(名称<yyj1>"应用集1" )



1. 创建监控项



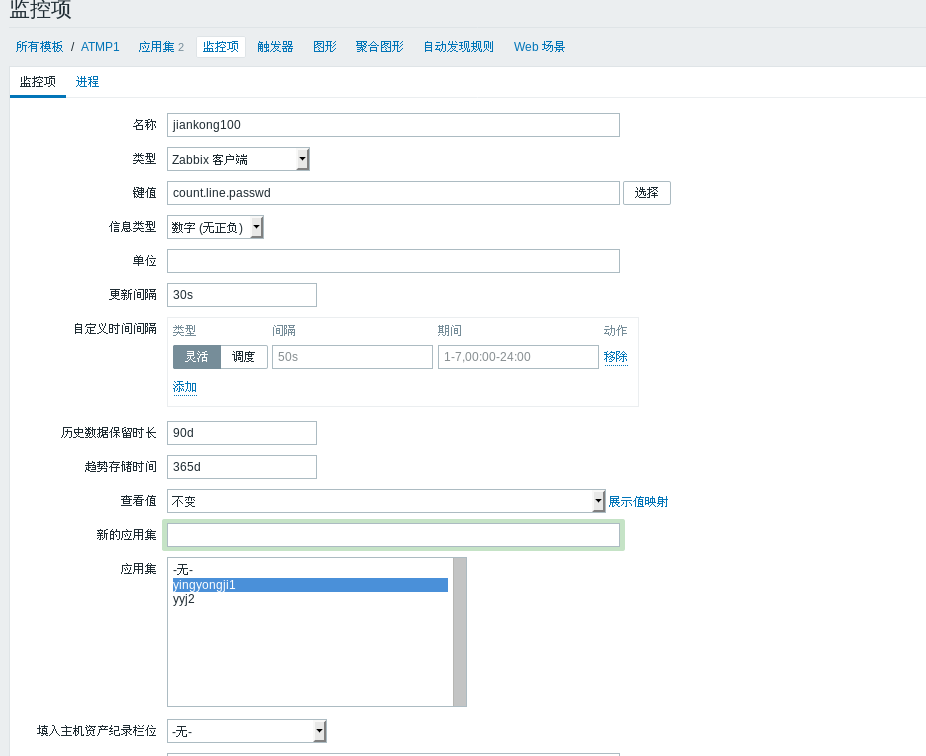
点击 ATMP1里的--->监控项--->创建监控项

名称(jiankong100)

键值 **count.line.passwd** :这个是自己前面定义的值

应用集(yyj1)

添加



1. 使用新模板监控主机100

点击-->配置-->主机-->web100--->模板

选择自己创建的模板 ATMP1----->添加----->更新

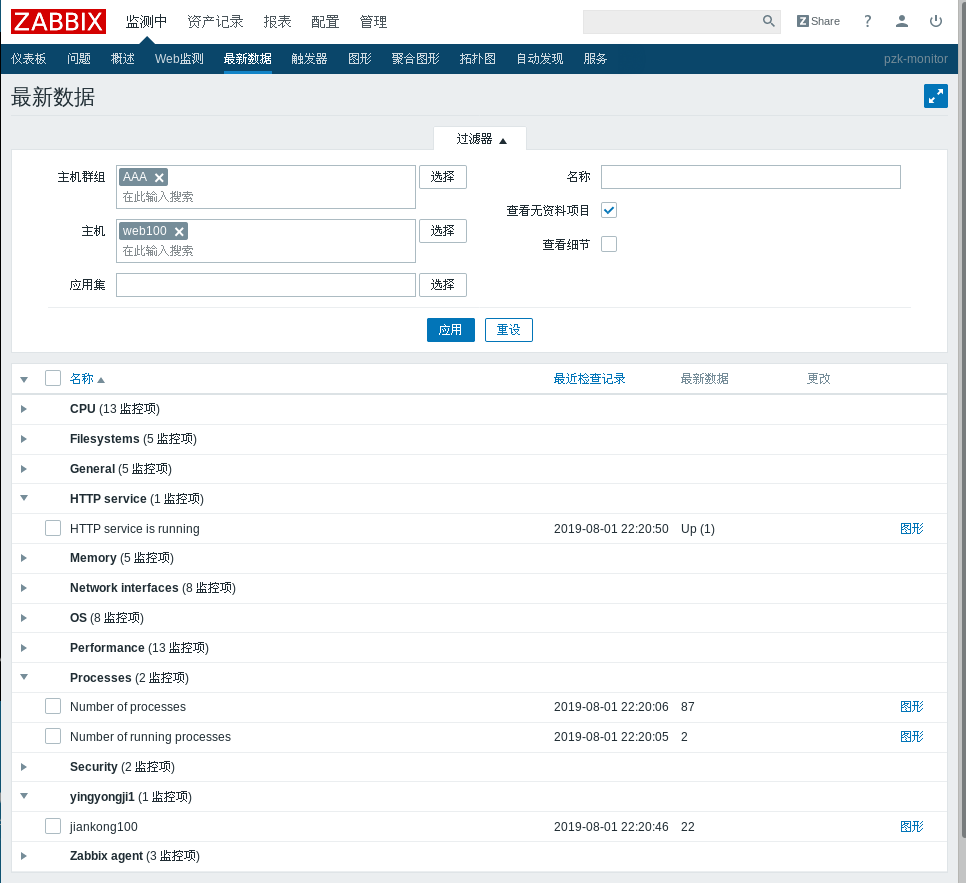


1. 查看监控数据

点击---检测中--最新数据 是否有创建的名称为--yyj1--**jiangkong100**

测试:

如果最初用户数为22,在2.100主机在添加用户,会变成23 (30秒刷一次)



8101

**day02**

**1 zabbix报警机制**

**使工程师不在监控服务器面前,如果web100用户量变了发送报警机制**

**(2.5模板会自带报警机制)**

**需要配置触发器与报警动作才可以自动报警**

**触发器**(trigger)

--表达式,如果内存不足300M,用户超过30个等表达式

--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**(action)

--触发器的条件被处罚后的行为

--可以是发送邮件、也可以是重启某个服务等

**1.2配置监控报警**

点击---->配置-->模板-->ATMP1-->触发器---->创建触发器

名称(cfq1)

严重性(灾难)

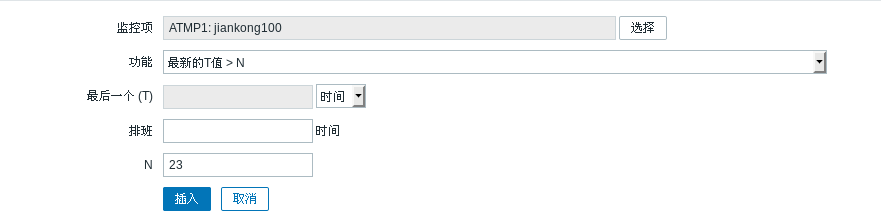
表达式(添加

监控项--添加

)

添加





**1.2创建动作**

**发送报警邮件**

**设置邮件服务器 (192.168.2.5)**

[root@zaserver php]# yum -y install postfix

[root@zaserver php]# yum -y install mailx

[root@zaserver php]# systemctl start postfix

[root@zaserver php]# netstat -untlp | grep 25

tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN 21551/master

验证邮件收发

[root@zaserver php]# echo "127.0.0.1 zaserver" >> /etc/hosts

[root@zaserver php]# mail -s "aaa" zabbix < /etc/hosts

[root@zaserver php]# su - zabbix

[zabbix@zaserver php]$ mail

**1.2.1登录管理页面制定邮件服务器**

**点击-->管理-->报警媒介类型-->Email**

SMTP服务器 localhost

SMTP HELO zaserver

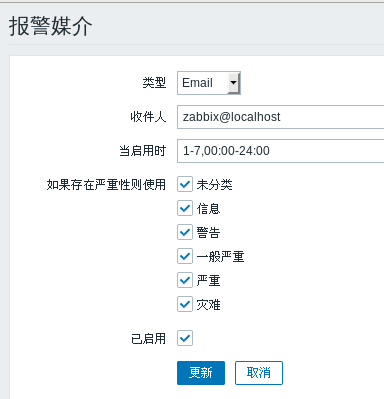
SMTP电邮 root@localhost

更新



1.2.2为账户添加Media(制定收件人)

点击-->管理-->用户-->admin-->报警媒介-->收件人( zabbix@localhost)-->更新



1.2.3创建动作

点击--->配置-->动作-->创建动作-->

名称:dz1

触发器名称 似 **cgq1**

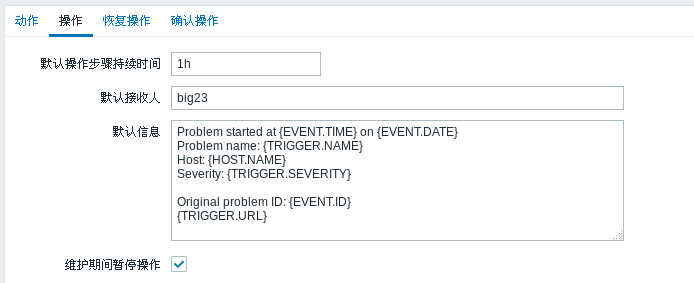
添加



点击-->操作

默认接收人 **big23**

点击--->新的



点击--->操作细节里的 发送到用户--->添加--->Admin-->选择

点击-->仅送到-->Email-->添加-->添加



1.2.4效果测试

在192.168.2.100添加用户,

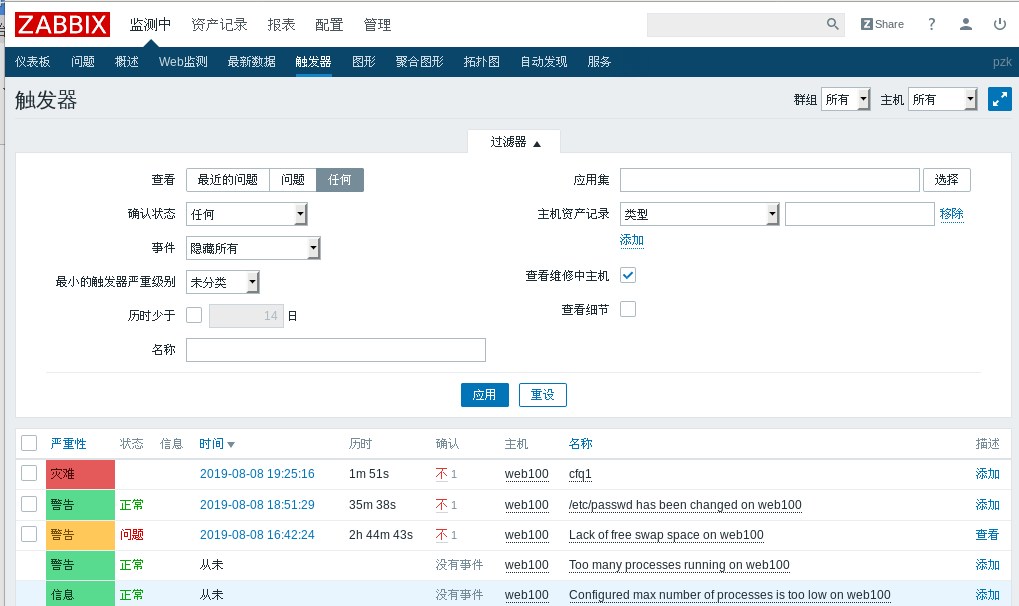
查看监控数据

查看是否收到邮件

[root@zaweb1 ~]# useradd yaya

[root@zaweb1 ~]# useradd yaya1

点击-->检查中-->触发器-->查看-->任何-->应用



在192.168.2.5主机操作

[root@zaserver ~]# su - zabbix

[zabbix@zaserver ~]$ mail



**2 zabbix进阶操作**

2.1**自动发现**

自动发现: 当zabbix需要控制的设备越来越多,手动添加监控设备越来越繁琐,可以考虑自动发现

自动发现可以实现:

发现主机 添加主机 添加到主机组 连接模板等

流程:

1 创建自动发现规则

2 创建Action动作

2.2配置自动发现

点击-->配置-->自动发现-->创建发现规则-->

名称 rule1

ip范围 192.168.2.200-254

更新时间 1m

点击-->新的->http

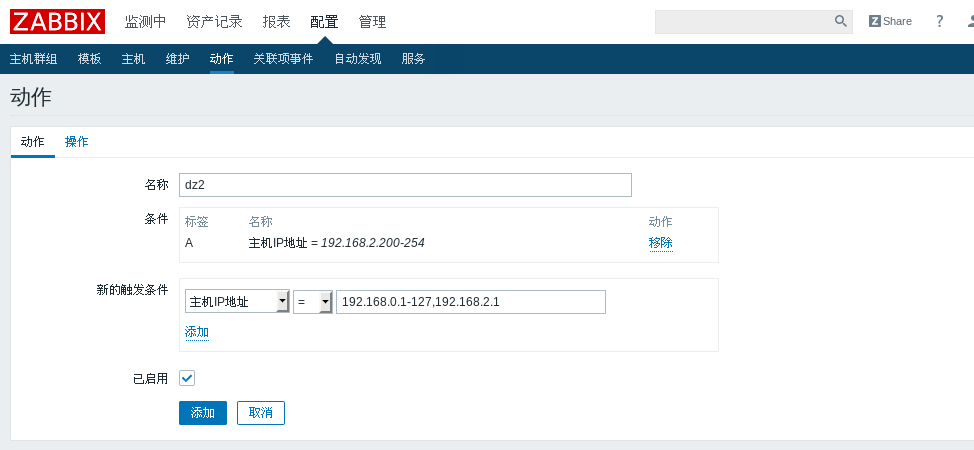


点击-->配置-->动作-->事件源-->自动发现-->

名称 dz2

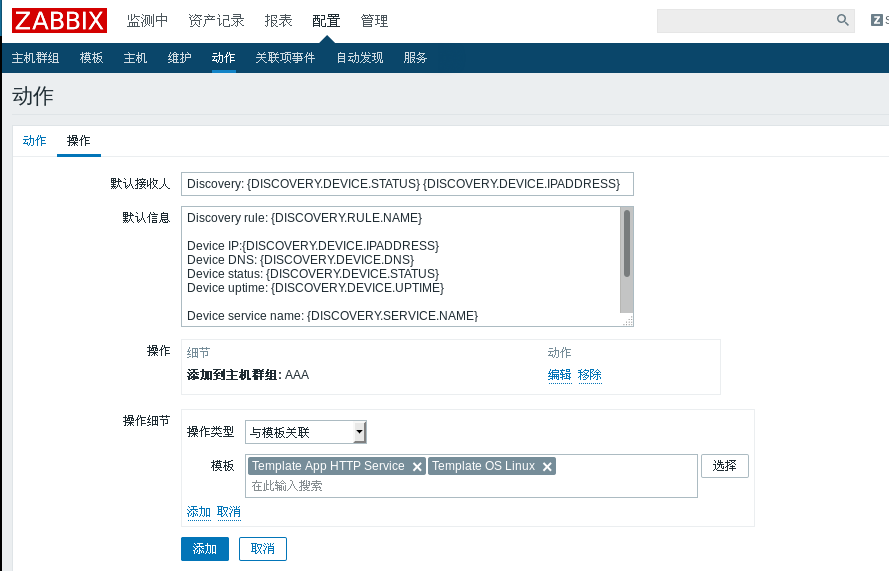
主机IP范围 192.168.2.200-254

添加



点击-->操作-->新的-->操作类型-->添加主机到群组-->主机群组-->选择-->AAA-->选择-->添加

点击-->新的-->与模板关联-->模板-->选择-->703-->HTTP Liunx-->添加-->添加



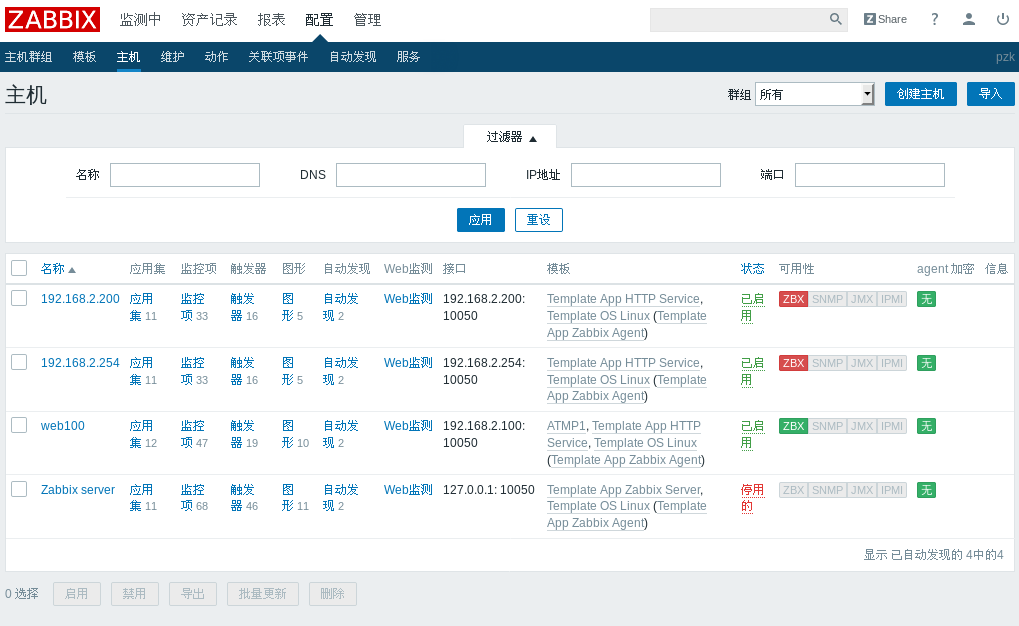
2.3创建新的虚拟机配置IP地址并运行网站HTTP服务

[**root@web2 ~**]# yum -y install httpd

[root@web2 ~]# systemctl start httpd

点击-->配置-->主机

(有192.168.2.200主机)



**2.2主被动监控**

2.2.1解控方式介绍

默认zabbix采用的是被动监控

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

服务器主动连接客户端

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开始周期性地收集数据给服务器

客户端主动连接服务器

区别:

Server不用每次需要数据都在连接Agent,Agent会自己收集数据并处理数据,Server仅需要保存数据即可

2.2.2配置主动监控

1 配置客户端  **192.168.2.201**

[student@room9pc01 03]$ scp -r Zabbix/ root@192.168.2.201:/root/

1 安装源码zabbix软件

2 修改配置文件(修改为主动模式)

3 启动服务

4 查看服务信息

[root@web3 ~]# cd Zabbix/

[root@web3 Zabbix]# tar -xf zabbix-3.4.4.tar.gz

[root@web3 zabbix-3.4.4]# yum -y install pcre-devel gcc

[root@web3 zabbix-3.4.4]# ./configure --enable-agent && make install

[root@web3 zabbix-3.4.4]# vim /usr/local/etc/zabbix\_agentd.conf

93 #Server=127.0.0.1

118 StartAgents=0

134 ServerActive=192.168.2.5

145 Hostname=web3

183 RefreshActiveChecks=120

[root@web3 zabbix-3.4.4]# useradd -s /sbin/nologin zabbix

[root@web3 zabbix-3.4.4]# zabbix\_agentd

[root@web3 zabbix-3.4.4]# ps -C zabbix\_agent (没有进程和端口)

[root@web3 zabbix-3.4.4]# netstat -untlp | grep :10050

2 配置服务器 192.168.2.5

1 克隆模板

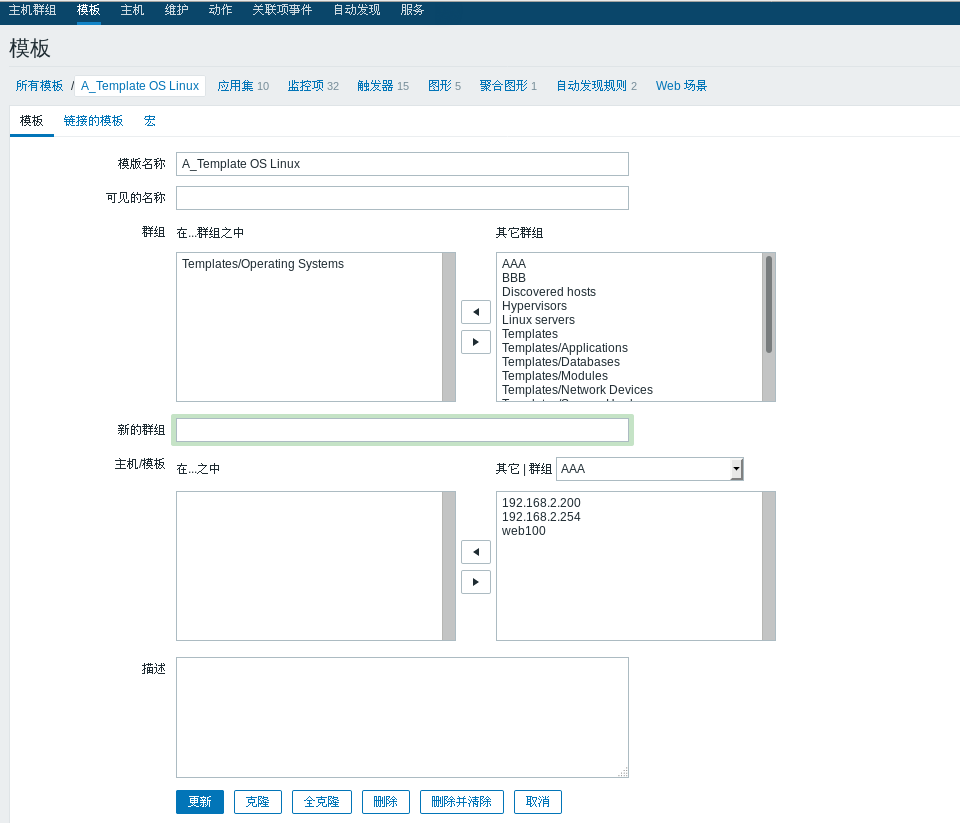
2 修改监控项模式

3 添加监控主机

4 为主机添加监控模板

5 查看数据图表

点击-->配置-->模板-->搜索linux-->[Template OS Linux](http://192.168.2.5/templates.php?form=update&templateid=10001&groupid=0)-->模板名称-->A\_[Template OS Linux](http://192.168.2.5/templates.php?form=update&templateid=10001&groupid=0)-->全克隆



点击--->A\_[Template OS Linux](http://192.168.2.5/templates.php?form=update&templateid=10001&groupid=0)-->监控项-->全选--->批量更新



点击-->类型-->主动模式-->更新



点击-->类型 向上三角 把前三个没有主动式的停用掉



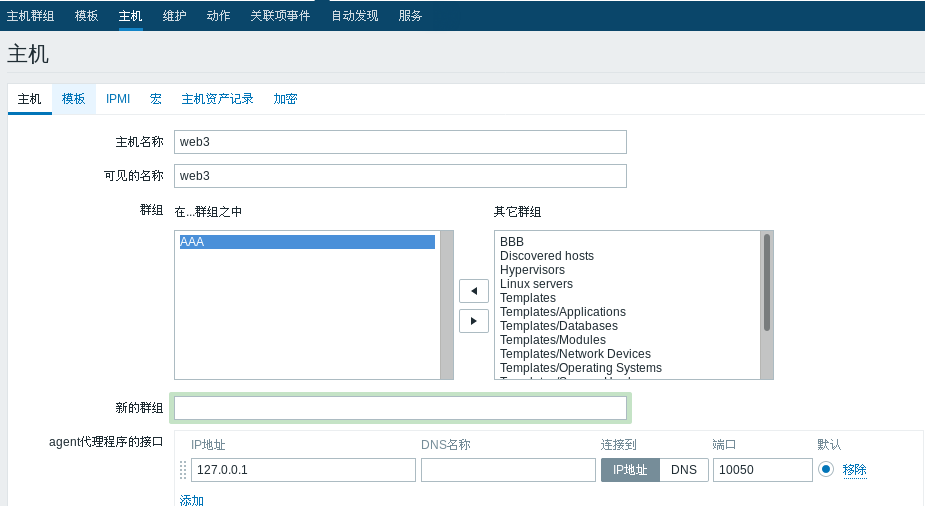
点击-->配置-->主机-->创建主机

主机名称必须与192.168.2.201的主机名一致

主机名称 web3

可见名称 web3

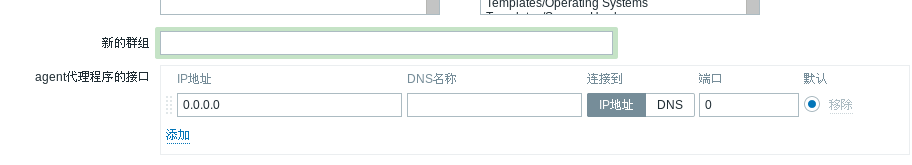
群组 AAAA



点击-->web3-->模板-->选择[A\_Template OS Linux](http://192.168.2.5/templates.php?form=update&templateid=10259" \t "/home/student/文档\\x/_blank)--->添加-->更新

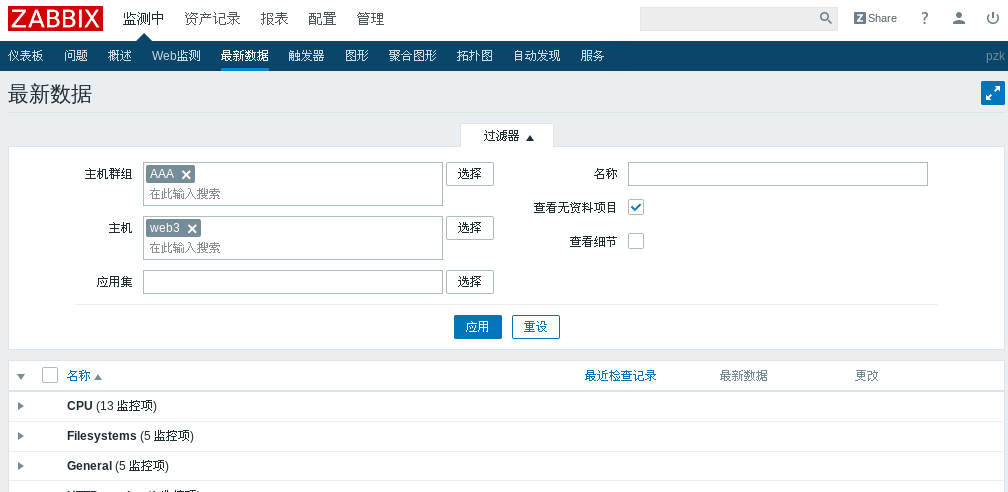


点击-->web3-->IP 0.0.0.0 端口 0



703

点击-->检测中-->最新数据--过滤web3-->应用



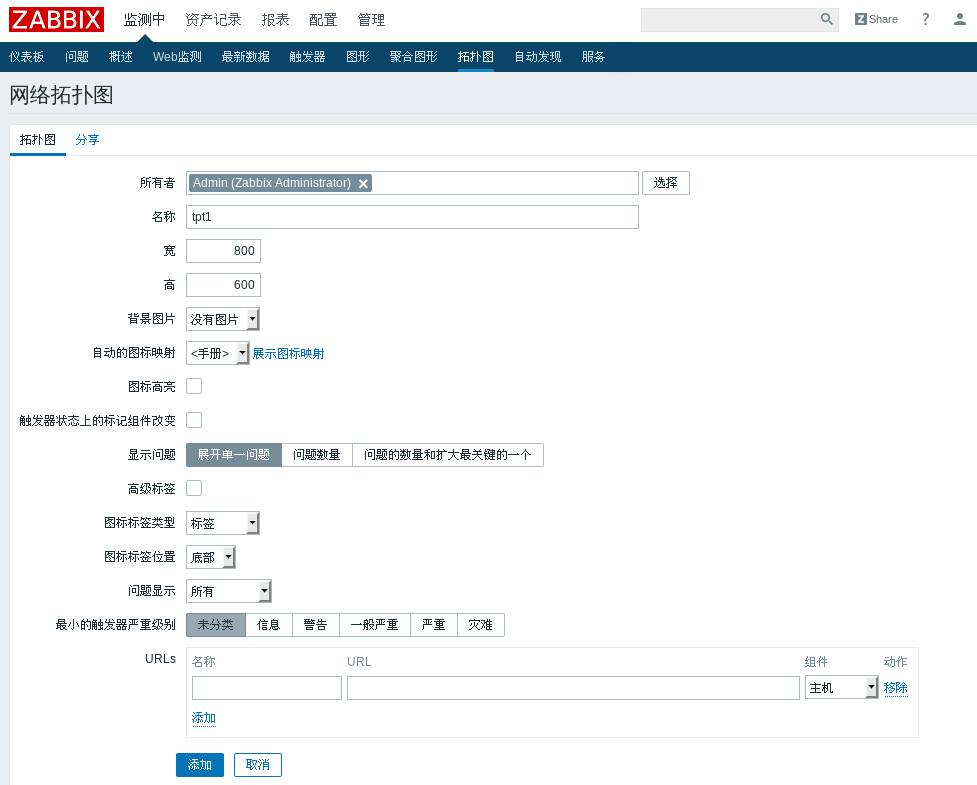
2.3拓扑图与聚合图形

2.3.1 拓扑图 绘制拓扑图使新人快速了解服务器架构

点击-->检测中-->拓扑图-->创建拓扑图-->

名称 tpt1

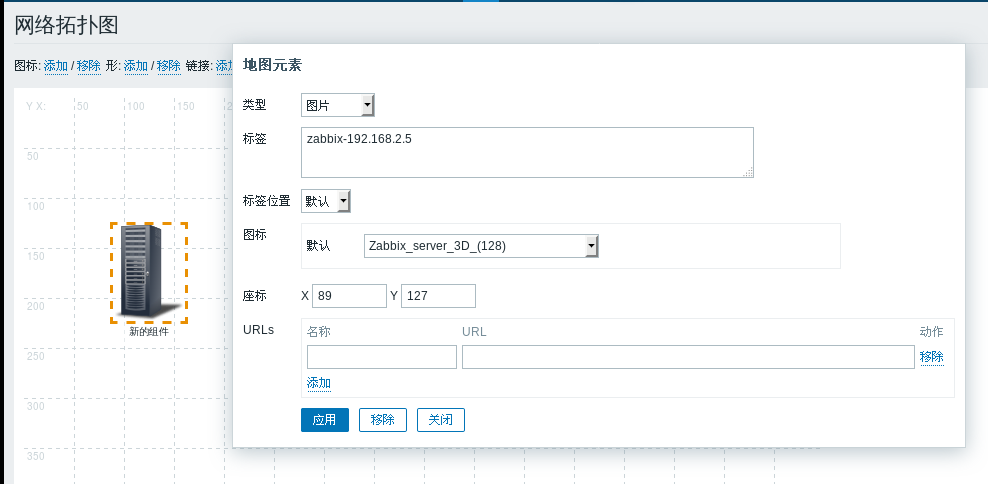
添加



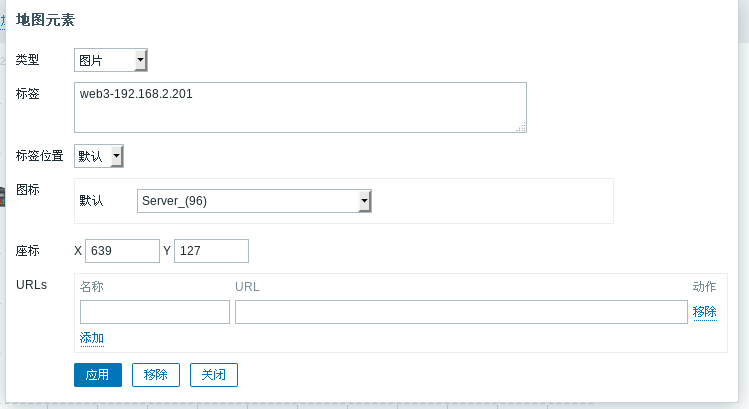
点击tpt1--->构造函数-->



点击-->添加(三个)-->点击图1设置-->应用-->关闭



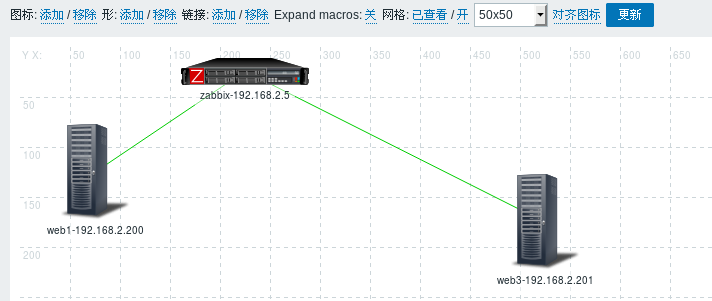
修改其他两个服务器名称web2-192.168.2.200 web3-192.168.2.201



把需要连接的两两选中 点击---链接后面的添加

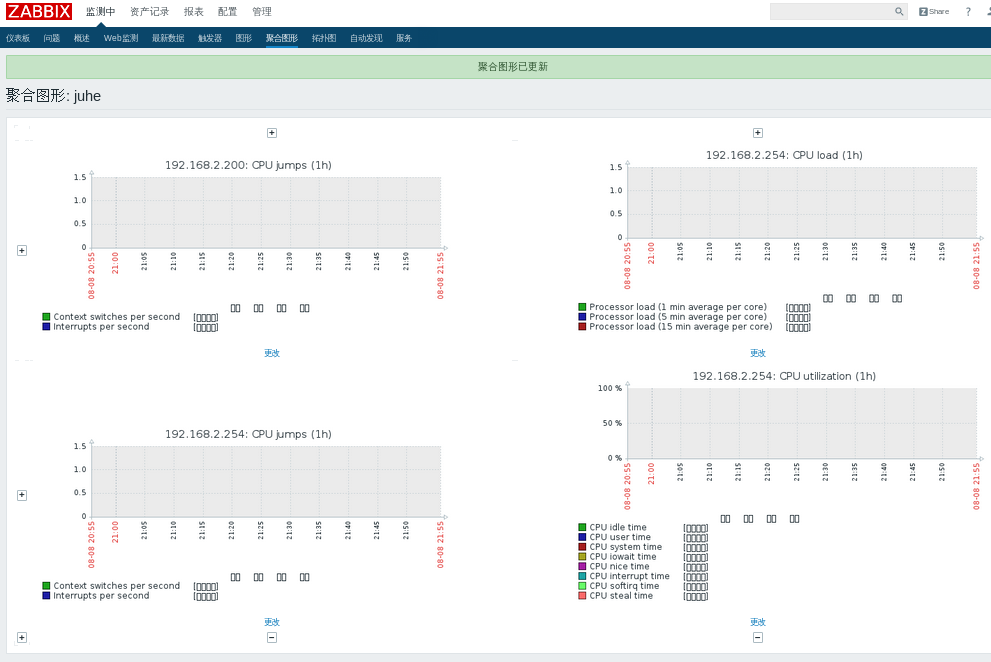
703

连接完后---->更新



**2.3.2 聚合图形**

**在一个页面显示多个数据图表,方便了解多组数据**

****

**59秒**

3Linux基本防护

1 Linux基本防护

chage

[root@50 ~]# useradd bob

[root@50 ~]# chage -l bob

最近一次密码修改时间 ：7月 27, 2019

密码过期时间 ：从不

密码失效时间 ：从不

帐户过期时间 ：从不

两次改变密码之间相距的最小天数 ：0

两次改变密码之间相距的最大天数 ：99999

在密码过期之前警告的天数 ：7

[root@50 ~]# chage -d 0 bob

[root@50 ~]# chage -l bob

最近一次密码修改时间 ：密码必须更改

密码过期时间 ：密码必须更改

密码失效时间 ：密码必须更改

帐户过期时间 ：从不

两次改变密码之间相距的最小天数 ：0

两次改变密码之间相距的最大天数 ：99999

在密码过期之前警告的天数 ：7

[root@50 ~]# chage -E 2019/7/31 bob

[root@50 ~]# chage -l bob

最近一次密码修改时间 ：密码必须更改

密码过期时间 ：密码必须更改

密码失效时间 ：密码必须更改

帐户过期时间 ：7月 31, 2019

两次改变密码之间相距的最小天数 ：0

两次改变密码之间相距的最大天数 ：99999

在密码过期之前警告的天数 ：7

passwd **-S:**看状态 **-l:**锁定 **-u:**解锁

passwd

强制定期修改密码

配置文件: /etc/login.defs

-对新建的用户有效

主要控制属性  
 PASS\_MAX\_DAYS    99999                        //密码最长有效期

PASS\_MIN\_DAYS    0                            //密码最短有效期

PASS\_MIN\_LEN    5                            //密码最短长度

PASS\_WARN\_AGE    7                            //密码过期前几天提示警告信息

伪装登录提示

配置文件 /etc/issue/ (适用于本地的)

/etc/issue.net (远程登录)

分别使用本地、远程登录

默认提示内核、系统等版本信息

文件系统安全

禁用非必要的系统服务RHEL7

]# systemctl enable httpd

]# systemctl disable httpd

禁用非必要的系统服务RHEL6

]# chkconfig httpd off

]# chkconfig httpd on

锁定/解锁保护文件

EXT3/EXT4的文件属性控制

chattr lsattr

属性i:不可变(immutable)

属性a:仅可追加(append only)

添加 chattr +a /etc/passwd

添加 chattr +i /etc/passwd

查看 lsattr -l /etc/passwd

查看 lsattr -l /etc/resolv.conf

删除 chattr -a /etc/passwd

删除 chattr -i /etc/resolv.conf

2 用户切换与提权

su

3 SSH访问控制

3.1 SSH防护概述

存在的安全隐患

-密码嗅探、键盘记录

-暴力枚举账号、猜解密码

常见的防护措施

-用户限制、黑白名单

-更改验证方式(密码---->密钥对)

-防护墙

3.2配置文件 /etc/ssh/sshd\_config

Protocol 2                                         //SSH协议

PermitRootLogin no                          //禁止root用户登录

PermitEmptyPasswords no                    //禁止密码为空的用户登录

UseDNS no                                       //不解析客户机地址

(三次登录失败,1分钟后再登录,这两是连用的,防止暴力破解)

LoginGraceTime 1m                          //登录限时

MaxAuthTries 3                                  //每连接最多认证次数

3.3sshd黑/白名单配置

AllowUsers //定义账户白名单

AllowGroups //定义组白名单

DenyUsers //定义黑名单

DenyGroups //定义组黑名单

AllowUsers root@192.168.4.254 da

3.4 SSH密钥对认证登录

43 #PubkeyAuthentication yes //公钥身份验证

47 AuthorizedKeysFile .ssh/authorized\_keys //授权密钥文件

65 PasswordAuthentication yes //设置密码认证

创建密钥对

[root@51 ~]# rm -rf ~/.ssh

[root@51 ~]# ssh-keygen

[root@51 ~]# ls ~/.ssh

[root@51 ~]# ssh-copy-id root@192.168.4.50

[root@50 ~]# vim /etc/ssh/sshd\_config

65 PasswordAuthentication no //关闭密码认证

[root@51 ~]# ssh -X root@192.168.4.50

//50的root有51公钥,可以连接

[root@51 ~]# ssh -X da@192.168.4.50

//因为50关闭了密码认证,50的da没有51的公钥,只有root有

4 SElinux安全防护

[root@50 ~]# vim /etc/sysconfig/selinux

SELINUX=enforcing

[root@50 ~]# touch /.autorelabel

SELinux status: enabled //SElinux状态

Loaded policy name: targeted //保护策略

Current mode: enforcing //工作模式