＝

**[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/SECURITY/DAY02/CASE/01/index.html" \l "page_top_case)**

# NSD SECURITY DAY02

1. [案例1：实现Zabbix报警功能](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/SECURITY/DAY02/CASE/01/index.html" \l "case1)

**概念**

自定义的监控项默认不会自动报警

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动报警

**概念**

触发器(trigger)

表达式，如内存不足300m，用户超过自己定义的

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)

触发器的条件触发后的行为

可以是发送邮件，也可以是重启某个服务等

创建触发器配置

1名cfq1 表达式

2设置邮件

指定邮件服务器

添加Media

创建动作　名dz1

3测试配置

1. [案例2：Zabbix自动发现](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/SECURITY/DAY02/CASE/01/index.html" \l "case2)

概述

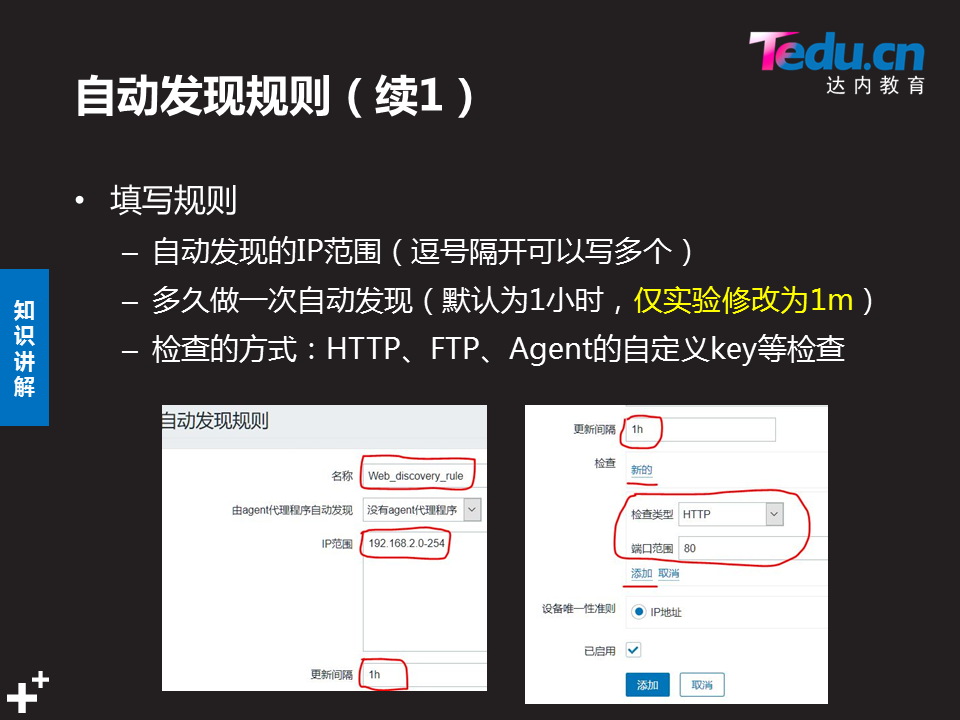
自动发现(Discovery)

当zabbix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，可以考虑使用自动发现功能



　自动发现可以实现

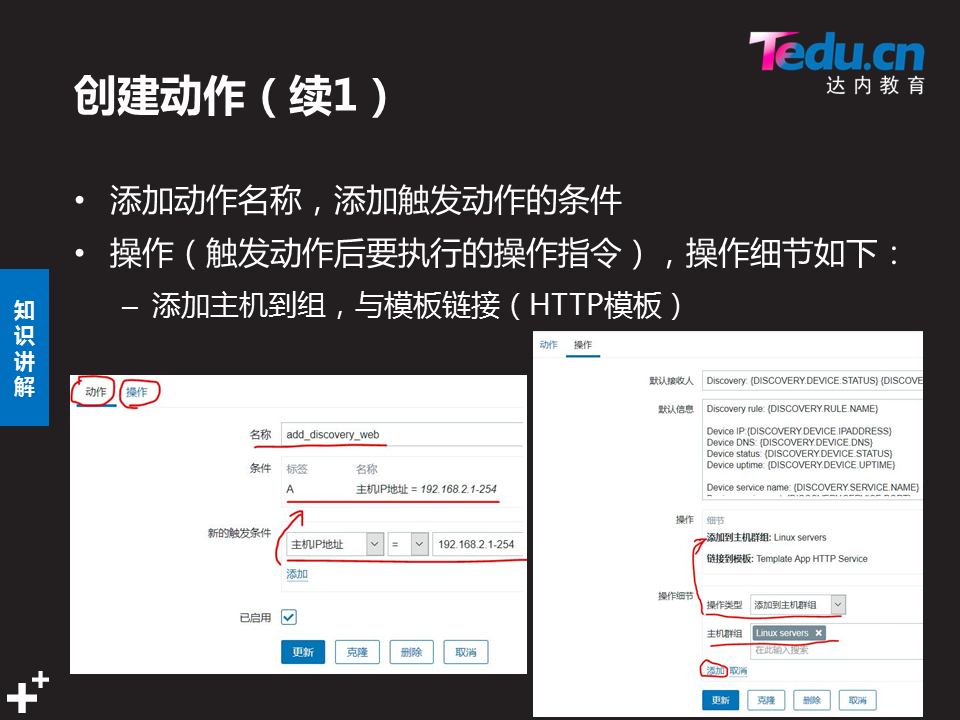
发现主机，添加主机，添加主机到组，连接模板等

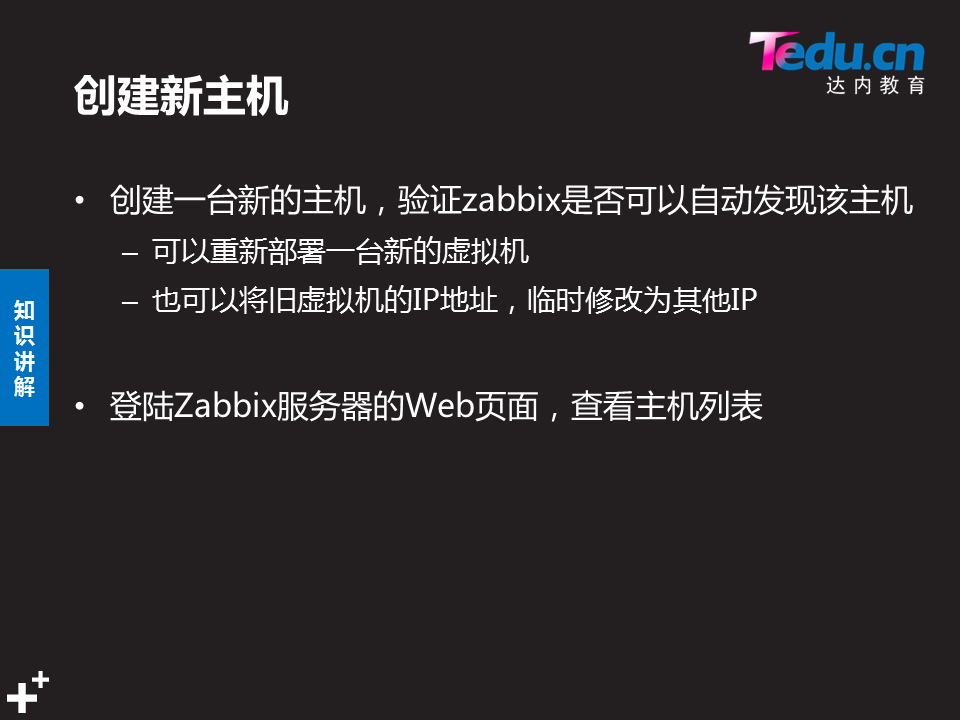


　 　自动发现(Discovery)流程

创建Ａction动作(发现主机后自动执行什么动作)

通过动作，执行添加主机，链接模板到主机等操作





1. [案例3：Zabbix主动监控](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/SECURITY/DAY02/CASE/01/index.html" \l "case3)

概述

自动发现(Discovery)

当zabbix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，可 　以考虑使用自动发现功能

　自动发现可以实现

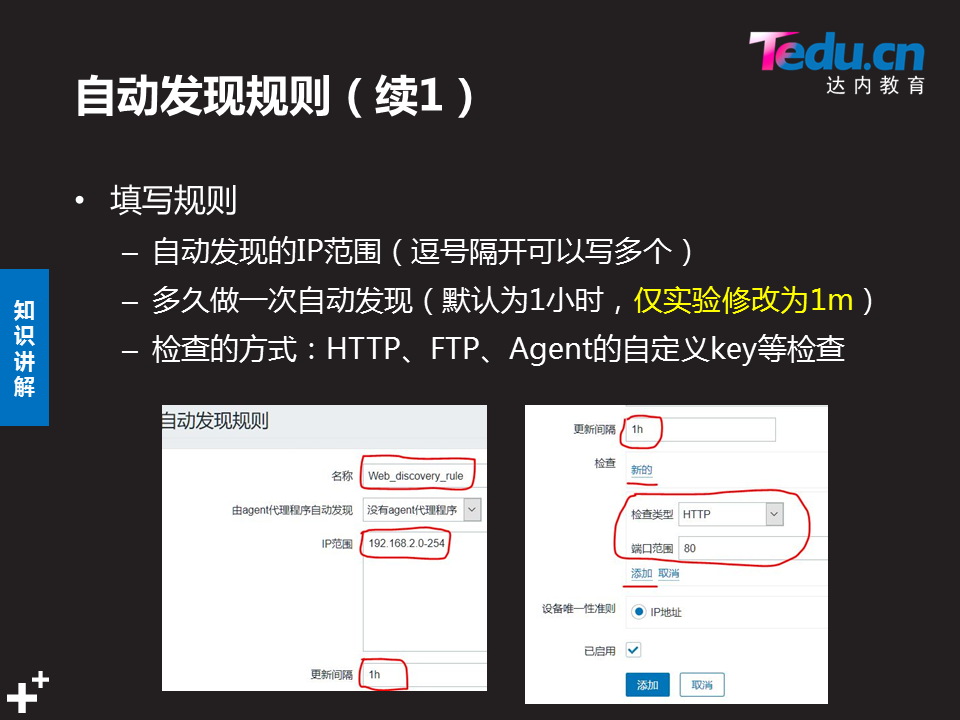
发现主机，添加主机，添加主机到组，连接模板等

自动发现(discovery)流程

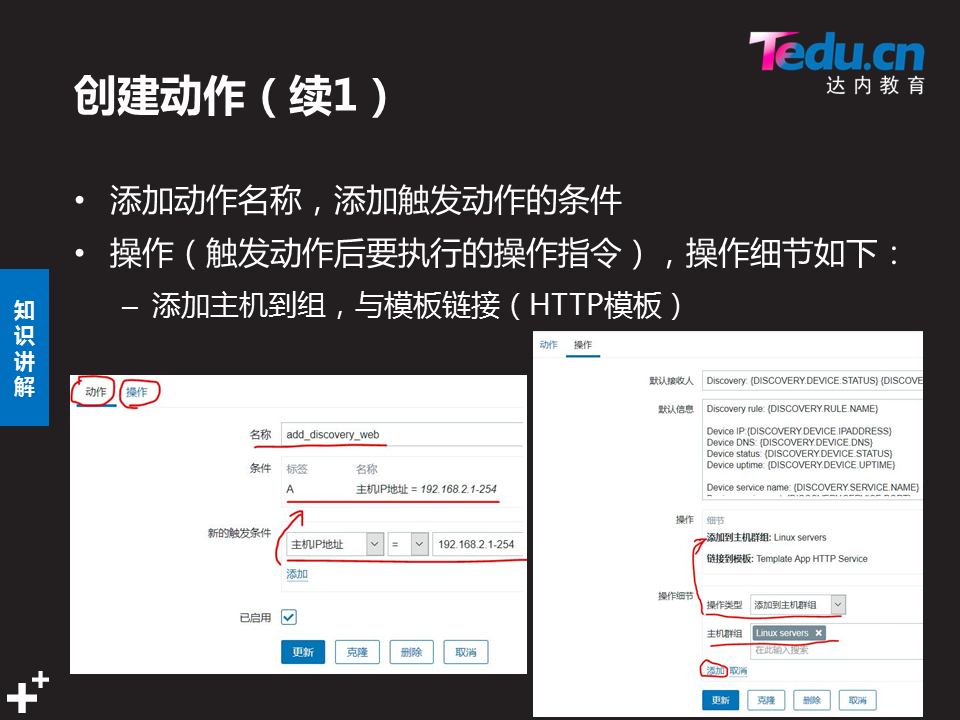
创建自动发现规则

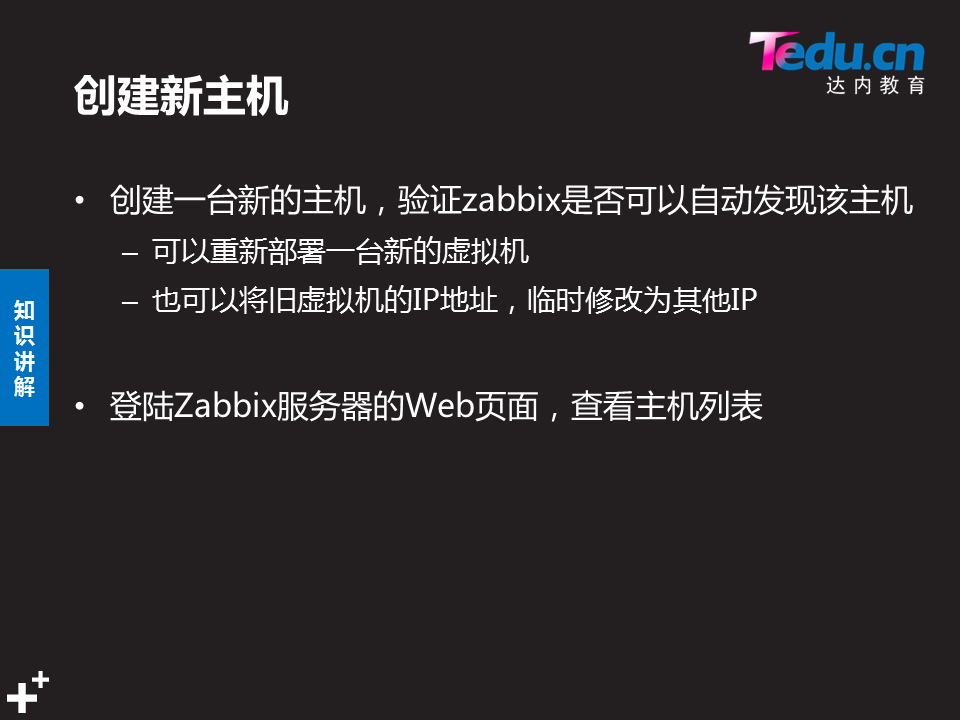
创建Action动作(发现主机后自动执行什么动作)

通过动作，执行添加主机，链接模板到主机等操作









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:Server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

区别

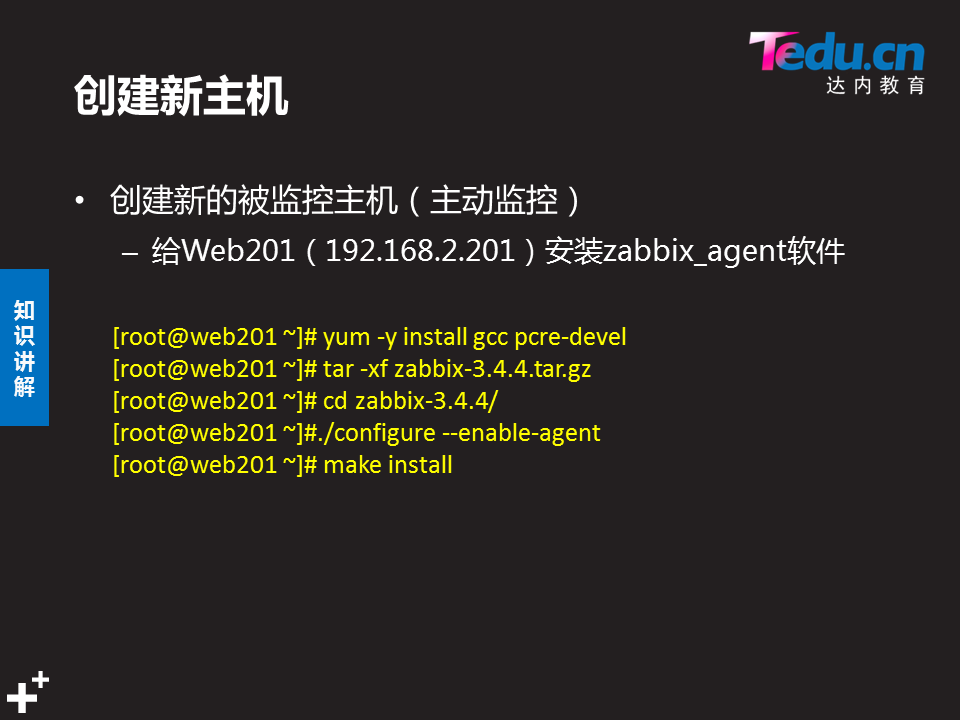
　 Server不用每次需要数据都有连接Agent，Agent会 自己收 集数据并处理数据，Server仅需要保存数据即可

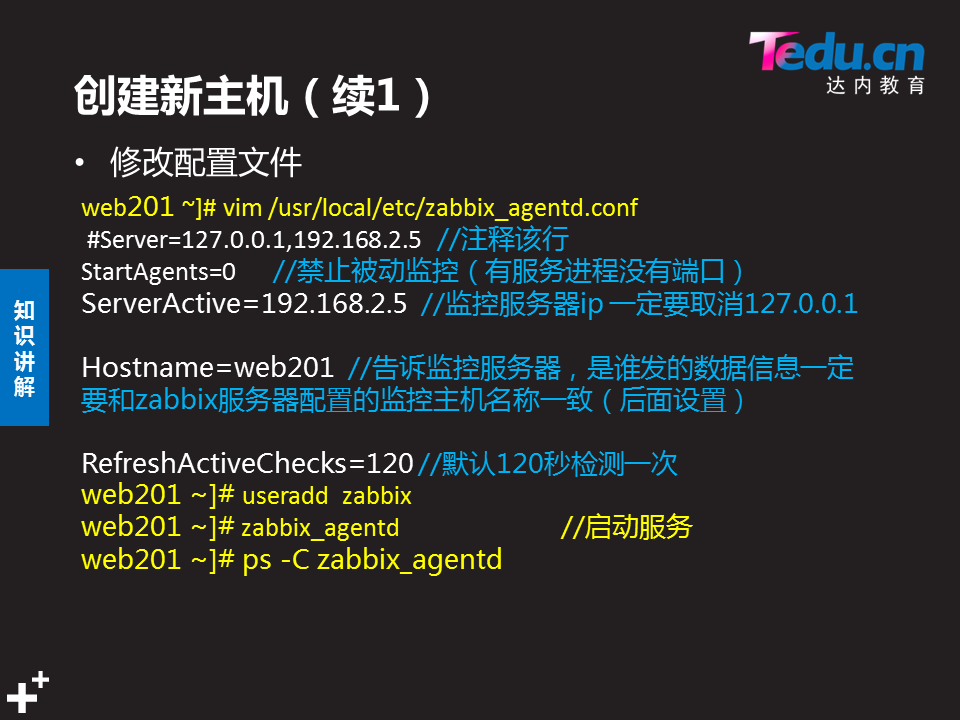
概述

当监控主机达到一定量级后，Zabbix服务器越来越慢

此时，可以考虑使用主动监控，释放服务器的压力

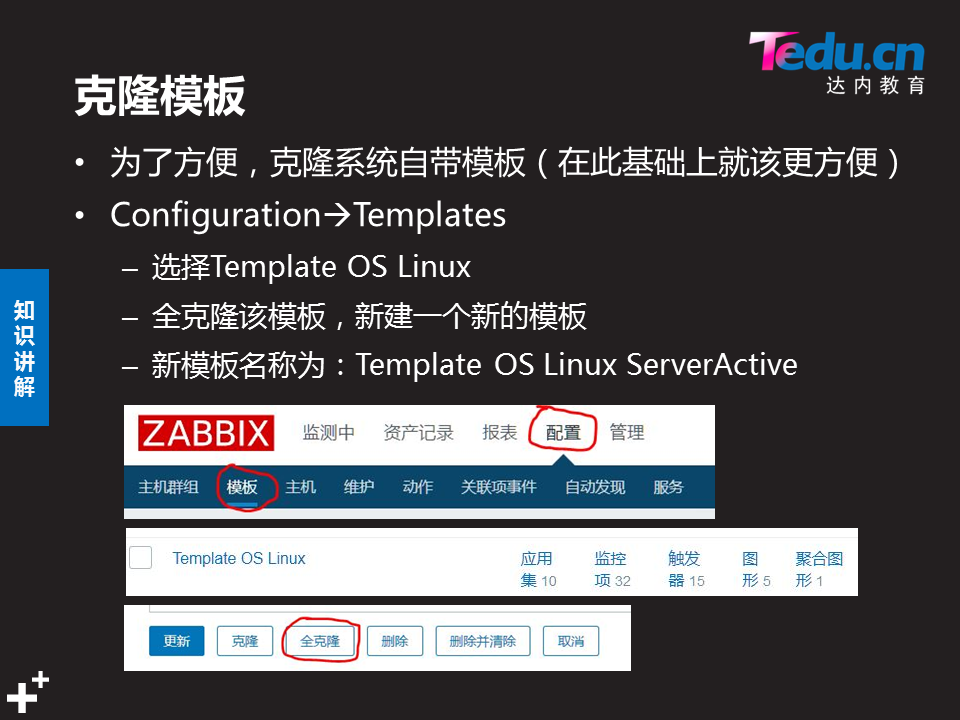
另外，Zabbix也支持分布式监控，也是可以考虑的方案





配置监控服务器

１克隆已有模板　名BTMP2



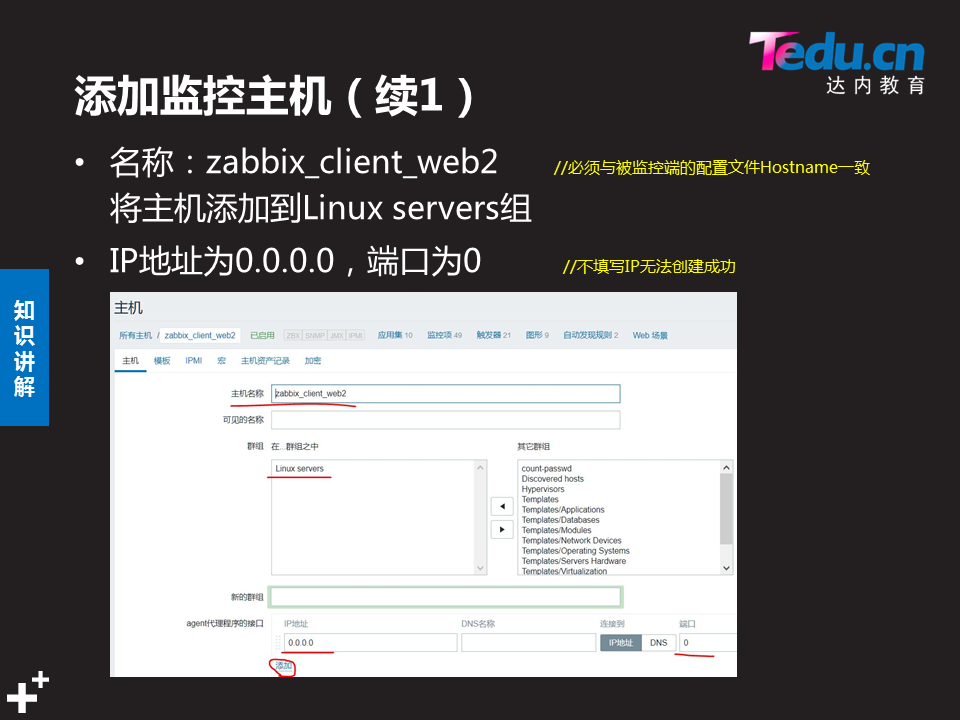
２修改模板监控项的模式为主动模式





３添加监控主机201





４调用克隆的模板监控主机201



５查看数据



1. [案例4：拓扑图与聚合图形](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/SECURITY/DAY02/CASE/01/index.html" \l "case4)

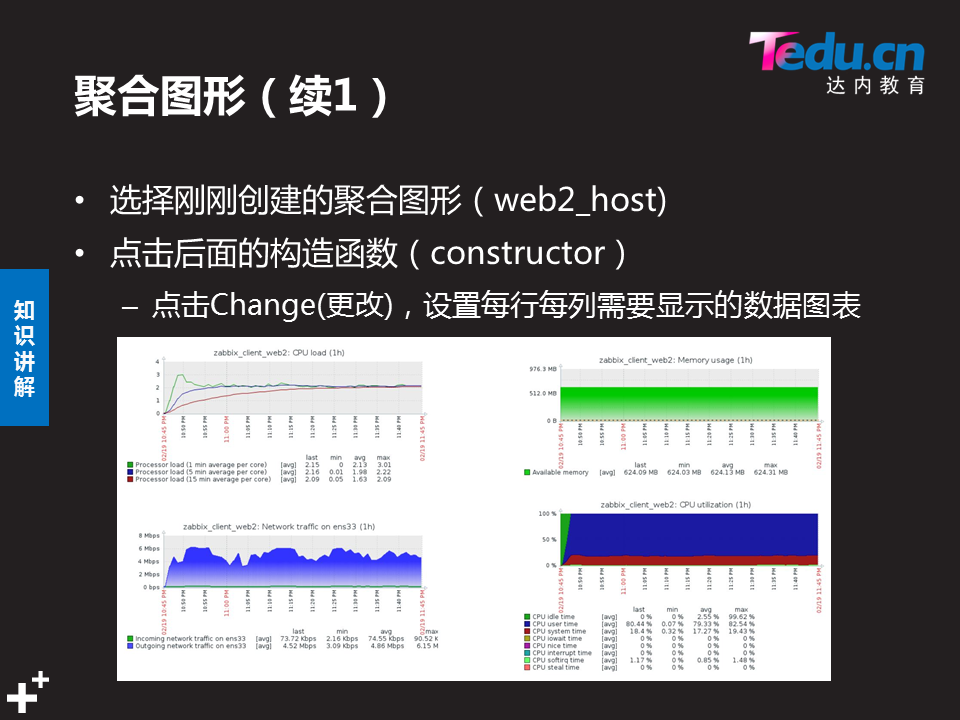
聚合图型:在一张网页页面上显示多个监控图片





拓扑图:在网页页面上绘制网络拓扑图





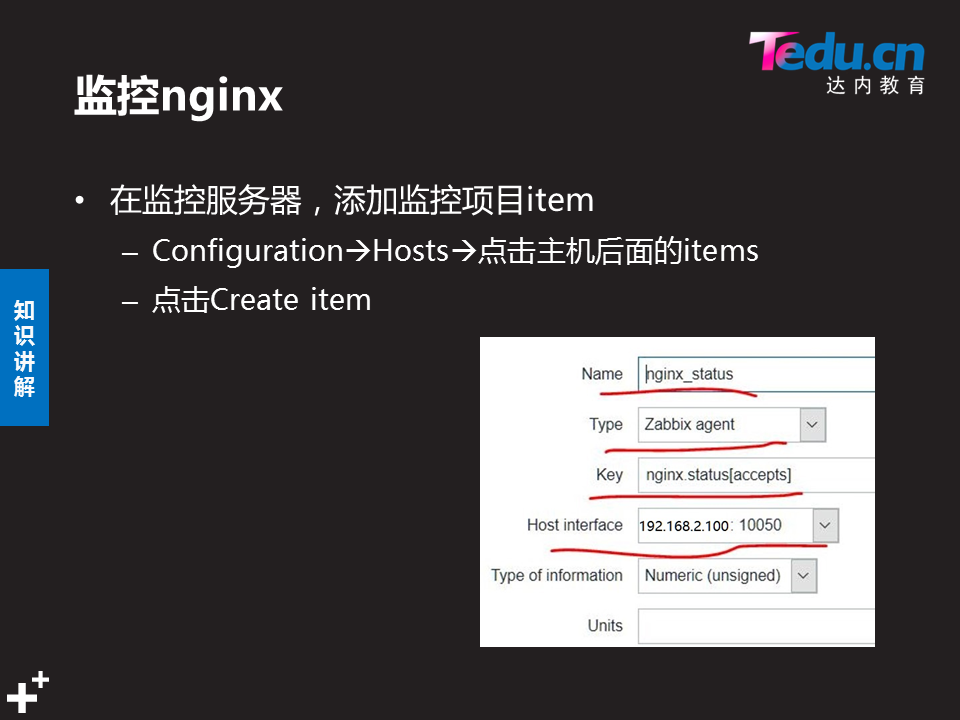
1. [案例5：自定义监控案例](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/SECURITY/DAY02/CASE/01/index.html" \l "case5)

１监控nginx服务状态







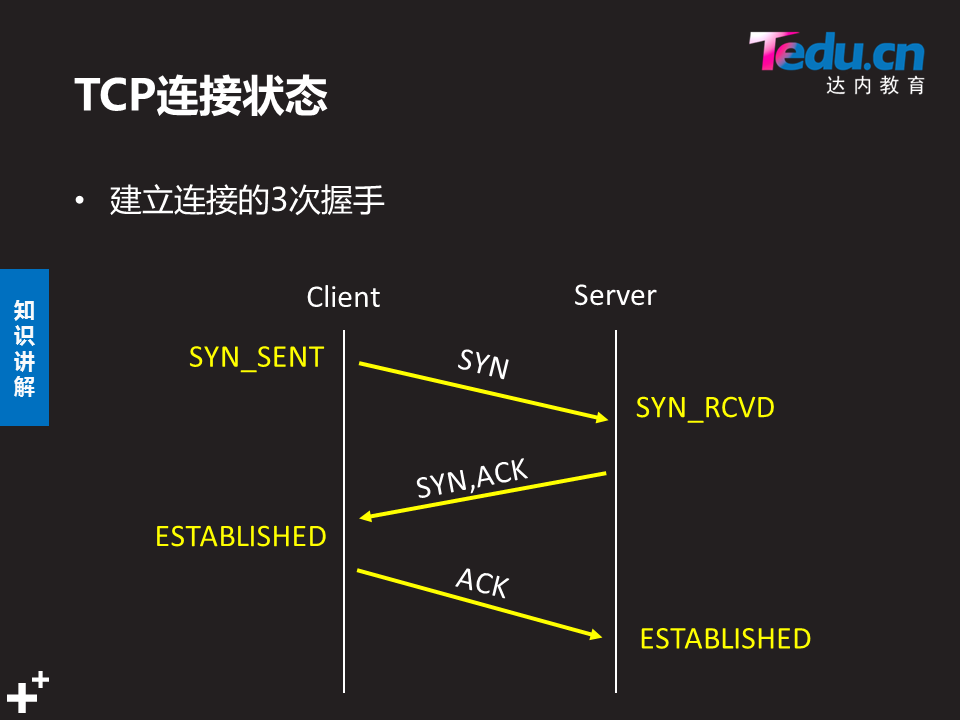


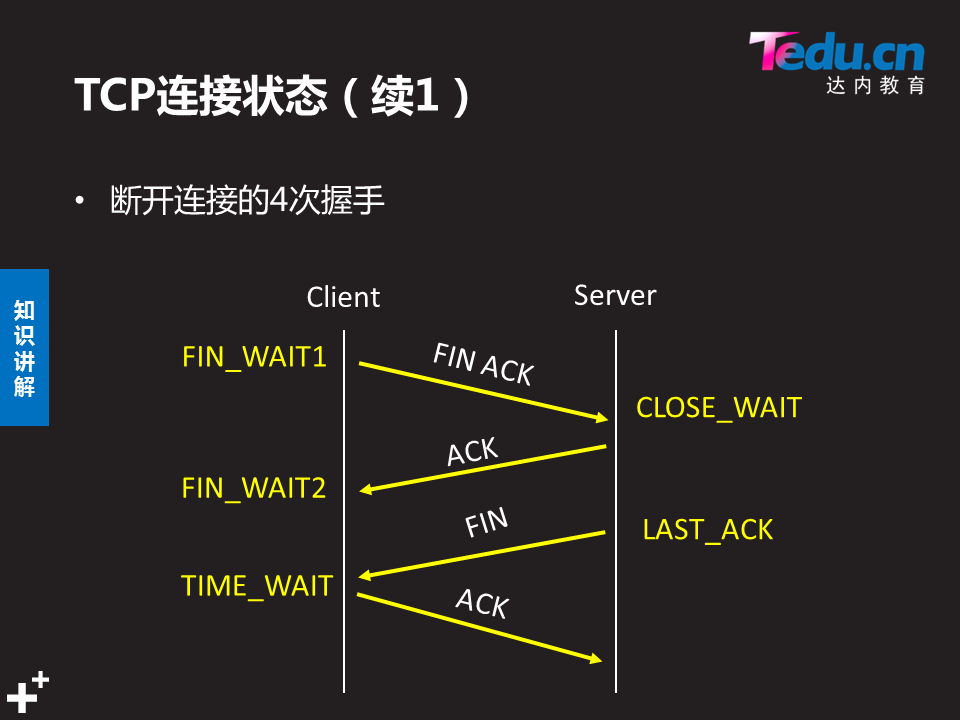
２监控网络连接状态

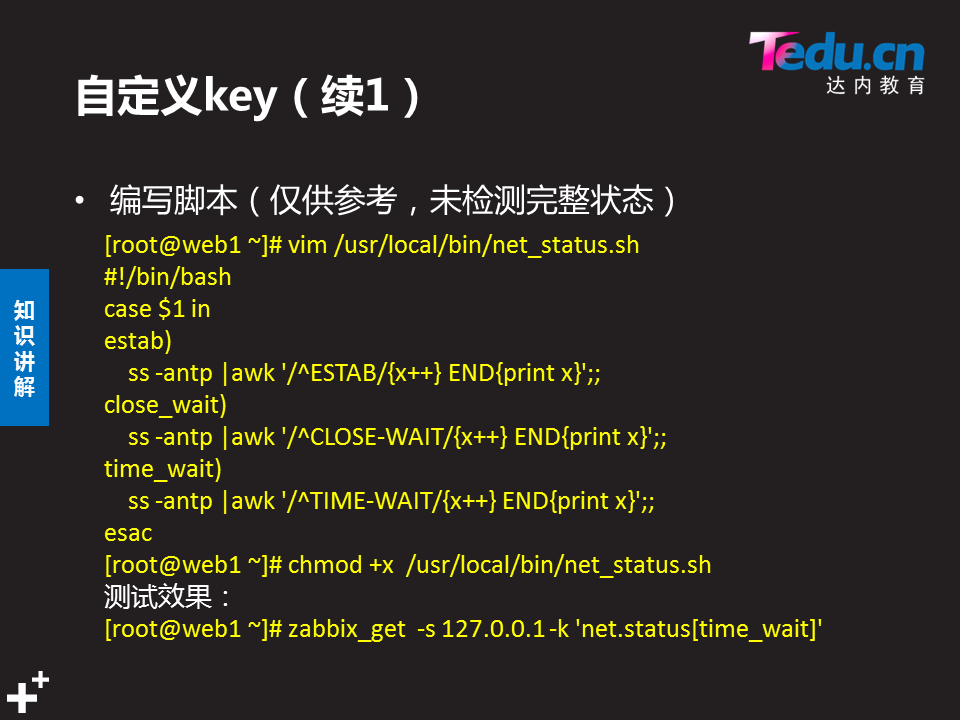
网络连接状态:SYN\_SENT SYN\_RCVD ESTABISHED

FIN\_WAIT1 CLOSE\_WAIT FIN\_WAIT2 LAST\_ACK

TIME\_WAIT(一共８种)









３扩展

## **1 案例1：实现Zabbix报警功能**

### **1.1 问题**

沿用第5天Zabbix练习，使用Zabbix实现报警功能，实现以下目标：

1. 监控Linux服务器系统账户
2. 创建Media，设置邮件服务器及收件人邮箱
3. 当系统账户数量超过26人时发送报警邮件

### **1.2 方案**

自定义的监控项默认不会自动报警，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。

什么是触发器（trigger）？

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

什么是动作（action）？

动作是触发器的条件被触发后所执行的行为

可以是发送邮件、也可以是重启某个服务等

参考如下操作步骤：

1. 创建触发器并设置标记
2. 设置邮箱
3. 创建Action动作

### **1.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：创建触发器规则**

1）创建触发器

创建触发器时强烈建议使用英文的语言环境，通过Configuration--> Templates，找到我们之前创建的count.line.passwd模板，点击模板后面的triggers，如图-1所示。

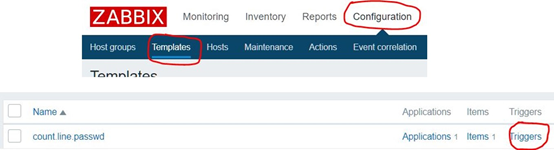


图-1

2）触发器表达式

创建触发器时需要定义表达式，触发器表达式（Expression）是触发异常的条件，触发器表达式格式如下：

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

{主机：key.函数(参数)}<表达式>常数

在如图-2所示的蓝色方框中编写触发器表达式，可以直接手写，也可以通过add选择表达式模板。



图-2

下面，我们看几个表达式的案例：

{web1:system.cpu.load[all,avg1].last(0)}>5 //0为最新数据

如果web1主机最新的CPU平均负载值大于5，则触发器状态Problem

{vfs.fs.size[/,free].max(5m)}<10G //5m为最近5分钟

根分区，最近5分钟的最大容量小于10G，则状态进入Problem

{vfs.file.cksum[/etc/passwd].diff(0)}>0 //0为最新数据

最新一次校验/etc/passwd如果与上一次有变化，则状态进入Problem

大多数函数使用秒作为参数，可以使用#来表示其他含义（具体参考表-1）。

avg, count, last, min and max 等函数支持额外的第二个参数time\_shift（时间偏移量），这个参数允许从过去一段时间内引用数据。

3）配置触发器

设置触发器名称，如图-3所示，点击add添加表达式，填写表达式：监控项为账户数量，最近300秒账户数量大于26（根据系统账户数量实际填写），效果如图-4所示。

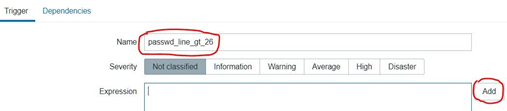


图-3

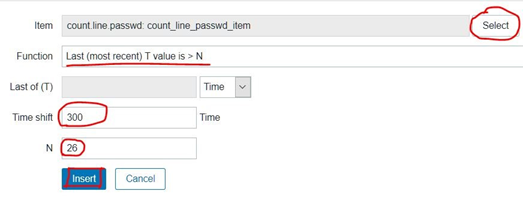


图-4

选择触发器报警级别，如图-5所示，Add创建该触发器，如图-6所示。

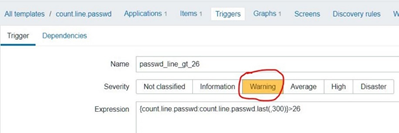


图-5

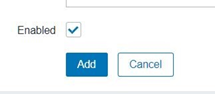


图-6

**步骤二：设置邮件**

1）创建Media

通过Administration（管理）-->Media Type（报警媒体类型）-->选择Email（邮件），如图-7所示。



图-7

设置邮件服务器信息，设置邮件服务器及邮件账户信息，如图-8所示。

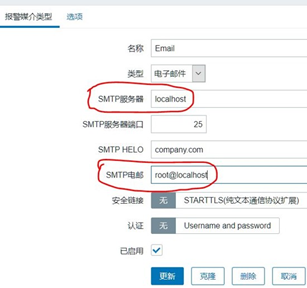


图-8

2)为用户添加Media

在Administration（管理）-->Users（用户）中找到选择admin账户，如图-9所示。



图-9

点击Admin账户后，在弹出的界面中选择Media（报警媒介）菜单-->点击Add(添加)报警媒介，如图-10所示。



图-10

点击Add（添加）后，在Meida Type中填写报警类型，收件人，时间等信息，如图-11所示。



图-11

**步骤三：创建Action动作**

1）Action动作

Action（动作）是定义当触发器被触发时的时候，执行什么行为。

通过Configuration（配置）-->Actions（动作）-->Create action（创建动作），如图-12所示。



图-12

2）配置Action动作的触发条件

填写Action动作的名称，配置什么触发器被触发时会执行本Action动作（账户数量大于26），如图-13所示。

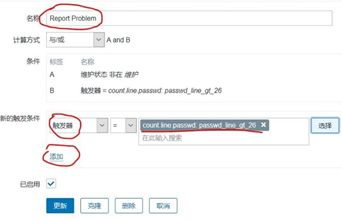


图-13

3）配置Action动作的具体行为

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，如图-14和图-15所示。



图-14

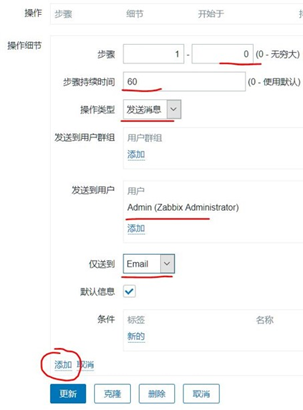


图-15

4）测试效果

在被监控主机创建账户（让账户数量大于26），然后登录监控端Web页面，在仪表盘中查看问题报警（需要等待一段时间），如图-16所示。



图-16

查看报警邮件，在监控服务器上使用mail命令查收报警邮件，如图-17所示。

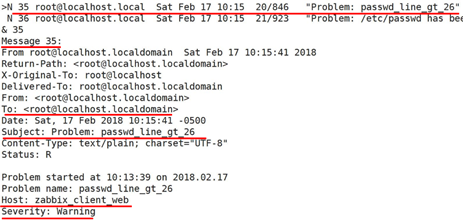


图-17

## **2 案例2：Zabbix自动发现**

### **2.1 问题**

沿用前面的练习，配置Zabbix的自动发现机制，实现以下目标：

1. 创建自动发现规则
2. 创建自动发现后的动作，添加主机、为主机链接模板

### **2.2 方案**

什么是自动发现（Discovery）？

当Zabbix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，可以考虑使用自动发现功能，自动添加被监控主机，实现自动批量添加一组监控主机功能。

自动发现可以实现：

* 自动发现、添加主机，自动添加主机到组；
* 自动连接模板到主机，自动创建监控项目与图形等。

自动发现（Discovery）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动作
* 通过动作，执行添加主机，链接模板到主机等操作

### **2.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：自动发现规则**

1）创建自动发现规则

通过Configuration（配置）-->Discovery（自动发现）-->Create discovery rule（创建发现规则），如图-18所示。



图-18

2）填写规则

填写自动发现的IP范围（逗号隔开可以写多个），多久做一次自动发现（默认为1小时，仅实验修改为1m），如图-19所示。配置检查的方式：Ping、HTTP、FTP、Agent的自定义key等检查，如图-20所示。



图-19



图-20

**步骤二：创建动作**

1）创建Action动作

通过Configuration（配置）--> Actions Event source(事件源)：自动发现(Discovery)-->Create action（创建动作），如图-21所示。



图-21

2）配置Action动作具体行为

配置动作，添加动作名称，添加触发动作的条件，如图-22所示。



图-22

点击操作（触发动作后要执行的操作指令），操作细节：添加主机到组，与模板链接（HTTP模板），如图-23所示。

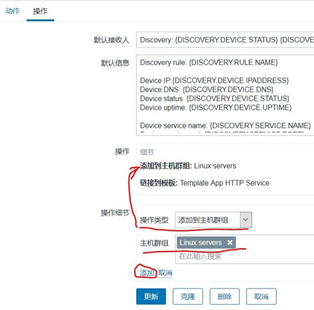


图-23

**步骤二：添加新的虚拟机**

1）创建新的虚拟机（启动HTTP服务器）

创建一台新的主机，验证zabbix是否可以自动发现该主机，可以重新部署一台新的虚拟机（注意前面的课程，我们已经创建了虚拟机zabbixclient\_web2，并且已经安装部署了Zabbix agent，如果没有该虚拟机或没有安装Agent，则需要前在zabbixclient\_web2部署Agent），也可以将旧虚拟机的IP地址，临时修改为其他IP。

2）验证结果

登陆Zabbix服务器的Web页面，查看主机列表，确认新添加的主机是否被自动加入监控主机列表，是否自动绑定了监控模板。

## **3 案例3：Zabbix主动监控**

### **3.1 问题**

沿用前面的练习，配置Zabbix主动监控，实现以下目标：

1. 修改被监控主机agent为主动监控模式
2. 克隆模板，修改模板为主动监控模板
3. 添加监控主机，并链接主动监控模板

### **3.2 方案**

默认zabbix采用的是被动监控，主动和被动都是对被监控端主机而言的！

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。

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。

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，Server仅需要保存数据即可。

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。

### **3.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：添加被监控主机**

1）为被监控主机安装部署zabbix agent

注意：前面的实验，我们已经在zabbixclient\_web2主机安装部署了zabbix agent，如果已经完成，则如下操作可以忽略。

1. **[**root@zabbixclient\_web2 **~]**# yum **-**y install gcc pcre**-**devel
2. **[**root@zabbixclient\_web2 **~]**# tar **-**xf zabbix**-3.4.4.**tar**.**gz
3. **[**root@zabbixclient\_web2 **~]**# cd zabbix**-3.4.4/**
4. **[**root@zabbixclient\_web2 **~]**#**./**configure **--**enable**-**agent
5. **[**root@zabbixclient\_web2 **~]**# make **&&** make install

2）修改agent配置文件

将agent监控模式修改为主动模式。

1. **[**root@zabbixclient\_web2 **~]**# vim **/**usr**/**local**/**etc**/**zabbix\_agentd**.**conf
2. #Server**=127.0.0.1,192.168.2.5**
3. //注释该行，允许谁监控本机
4. StartAgents**=0**
5. //被动监控时启动多个进程
6. //设置为0，则禁止被动监控，不启动zabbix\_agentd服务
7. ServerActive**=192.168.2.5**
8. //允许哪些主机监控本机（主动模式），一定要取消127.0.0.1
9. Hostname**=**zabbixclient\_web2
10. //告诉监控服务器，是谁发的数据信息
11. //一定要和zabbix服务器配置的监控主机名称一致（后面设置）
12. RefreshActiveChecks**=120**
13. //默认120秒检测一次
14. UnsafeUserParameters**=1**
15. //允许自定义key
16. Include**=**/usr/local**/**etc**/**zabbix\_agentd**.**conf**.**d**/**
17. **[**root@zabbixclient\_web2 **~]**# killall zabbix\_agentd                    //关闭服务
18. **[**root@zabbixclient\_web2 **~]**# zabbix\_agentd                            //启动服务

**步骤二：创建主动监控的监控模板**

1）克隆Zabbix自动的监控模板

为了方便，克隆系统自带模板（在此基础上就该更方便）。

通过Configuration（配置）-->Templates（模板）-->选择Template OS Linux

-->全克隆，克隆该模板，新建一个新的模板。如图-24所示。

新模板名称为：Template OS Linux ServerActive。



图-24

2）修改模板中的监控项目的监控模式

将模板中的所有监控项目全部修改为主动监控模式，通过Configuration（配置）-->Templates（模板）-->选择新克隆的模板，点击后面的Items（监控项）-->点击全选，选择所有监控项目，点击批量更新，将类型修改为：Zabbix Agent（Active主动模式），如图-25所示。



图-25

3）禁用部分监控项目

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，点击状态即可关闭。如图-26所示。



图-26

**步骤三：添加监控主机**

1）手动添加监控主机（主动模式监控）

在Zabbix监控服务器，添加被监控的主机（主动模式），设置主机名称：zabbixclient\_web2 （必须与被监控端的配置文件Hostname一致），将主机添加到Linux servers组，IP地址修改为0.0.0.0，端口设置为0，如图-27和图-28所示。



图-27

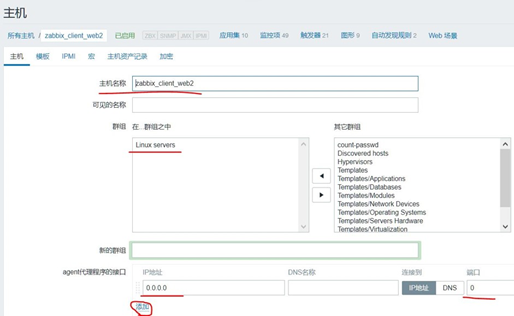


图-28

为主机添加监控模板，选择刚刚创建的模板（主动模式），添加链接模板到主机，如图-29所示。

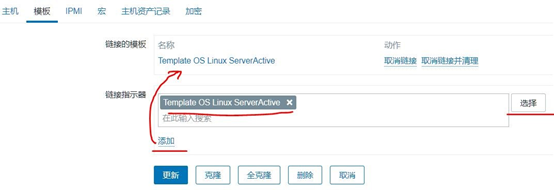


图-29

2）验证监控效果

查看数据图表，通过Monitoring-->Graphs菜单，选择需要查看的主机组、主机以及图形，查看效果，如图-30所示。

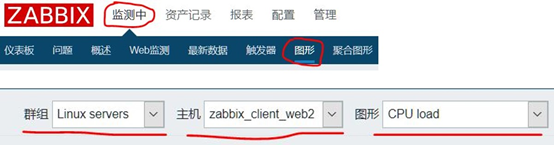


图-30

CPU、内存等其他数据可用正常获取，但是，查看分区图表时并无数据，因为分区数据采用的是自动发现监控，与普通监控项一样，修改为主动模式即可，选择Template OS Linux ServerActive模板，修改Discovery自动发现为主动模式。如图-31所示。



图-31

## **4 案例4：拓扑图与聚合图形**

### **4.1 问题**

沿用前面的练习，熟悉zabbix拓扑图与聚合图形，实现以下目标：

1. 创建修改拓扑图
2. 创建聚合图形

### **4.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：创建拓扑图**

1）创建拓扑

绘制拓扑图可以快速了解服务器架构，通过Monitoring（监控中）-->Maps（拓扑图），选择默认的Local network拓扑图，编辑即可（也可以新建一个拓扑图），如图-32所示。

IMG_287

图-32

2）拓扑图图表说明

* Icon（图标），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）

创建完拓扑图，效果如图-33所示。

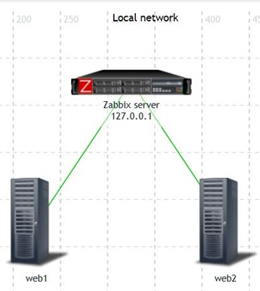


图-33

**步骤二：创建聚合图形**

1）创建聚合图形

聚合图形可以在一个页面显示多个数据图表，方便了解多组数据。

通过Monitoring（监控中）-->Screens（聚合图形）-->Create screen(创建聚合图形)即可创建聚合图形，如图-34所示。



图-34

修改聚合图形参数如下：

* Owner：使用默认的Admin用户
* Name：名称设置为zabbixclient\_web2\_host
* Columns：列数设置为2列
* Rows：行数设置为4行

2）为聚合图形中添加监控图形

选择刚刚创建的聚合图形（zabbixclient\_web2\_host)，点击后面的构造函数（constructor），点击Change(更改)，设置每行每列需要显示的数据图表，如图-35所示。

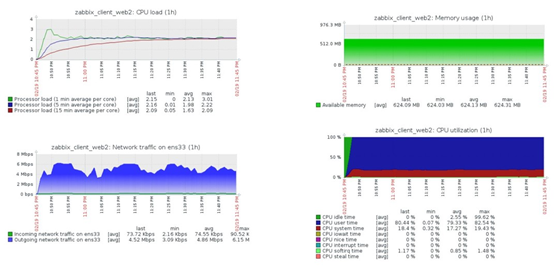


图-35

## **5 案例5：自定义监控案例**

### **5.1 问题**

沿用前面的练习，使用自定义key监控常用监控项目，实现以下目标：

1. 监控Nginx状态
2. 监控网络连接状态

### **5.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：监控Nginx服务状态**

1）准备环境，部署nginx软件

安装nginx软件，开启status模块

1. **[**root@zabbixclient\_web1 nginx**-1.12.2]**# **./**configure **\**
2. **>** **--with-**http\_stub\_status\_module
3. **[**root@zabbixclient\_web1 nginx**-1.12.2]**# make **&&** make install
4. **[**root@zabbixclient\_web1 **~]**# cat **/**usr**/**local**/**nginx**/**conf**/**nginx**.**conf
5. … …
6. location **/**status **{**
7. stub\_status on**;**
8. **}**
9. … …
10. **[**root@zabbixclient\_web1 **~]**# curl http**:**//192.168.2.100/status
11. Active connections**:** **1**
12. server accepts handled requests
13. **10** **10** **3**
14. Reading**:** **0** Writing**:** **1** Waiting**:** **0**

2）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

如：

UserParameter=ping[\*],echo $1

ping[0]， 返回的结果都是0

ping[aaa]， 返回的结果都是aaa

注意：被监控端修改配置文件，注意要允许自定义key并设置Include！

创建自定义key

1. **[**root@zabbixclient\_web1 **~]**# vim **/**usr**/**local**/**etc**/**zabbix\_agentd**.**conf**.**d**/**nginx**.**status
2. UserParameter**=**nginx**.**status**[\*],**/usr/local**/**bin**/**nginx\_status**.**sh $**1**
3. **[**root@zabbixclient\_web1 **~]**# killall zabbix\_agentd
4. **[**root@zabbixclient\_web1 **~]**# zabbix\_agentd

自定义监控脚本（仅供参考，未检测完整状态）

1. **[**root@zabbixclient\_web1 **~]**# vim **/**usr**/**local**/**bin**/**nginx\_status**.**sh
2. #**!**/bin/bash
3. **case** $**1** **in**
4. active**)**
5. curl **-**s http**:**//192.168.2.100/status |awk '/Active/{print $NF}';;
6. waiting**)**
7. curl **-**s http**:**//192.168.2.100/status |awk '/Waiting/{print $NF}';;
8. accepts**)**
9. curl **-**s http**:**//192.168.2.100/status |awk 'NR==3{print $2}';;
10. esac
11. **[**root@zabbixclient\_web1 **~]**# chmod **+**x **/**usr**/**local**/**bin**/**nginx\_status**.**sh

测试效果：

1. **[**root@zabbixclient\_web1 **~]**# zabbix\_get **-**s **127.0.0.1** **\**
2. **-**k 'nginx.status[accepts]'

登陆Zabbix监控Web，创建监控项目item，点击Configuration（配置）-->Hosts(主机)，点击主机后面的items（项目），点击Create item（创建项目）。修改项目参数如图-36所示。



图-36

**步骤二：监控网络连接状态**

1）了解TCP协议

熟悉TCP三次握手，参考图-37。

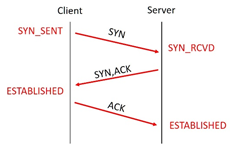


图-37

熟悉TCP连接的四次断开，参考图-38。

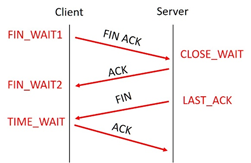


图-38

2）查看网络连接状态

模拟多人并发连接

1. **[**root@zabbixclient\_web1 **~]**# ab **-**c **1000** **-**n **100000** http**:**//192.168.2.100/

查看网络连接状态，仔细观察、分析第二列的数据

1. **[**root@zabbixclient\_web1 **~]**# ss **-**antup
2. //-a显示所有
3. //-t显示TCP连接状态
4. //-u显示UDP连接状态
5. //-n以数字形式显示端口号和IP地址
6. //-p显示连接对应的进程名称

3）创建自定义key

注意：被监控端修改配置文件，注意要允许自定义key并设置Include。

1. **[**root@zabbixclient\_web1 **~]**# vim **/**usr**/**local**/**etc**/**zabbix\_agentd**.**conf**.**d**/**net**.**status
2. UserParameter**=**net**.**status**[\*],**/usr/local**/**bin**/**net\_status**.**sh $**1**
3. **[**root@zabbixclient\_web1 **~]**# killall zabbix\_agentd
4. **[**root@zabbixclient\_web1 **~]**# zabbix\_agentd

自定义监控脚本（仅供参考，未检测完整状态）

1. **[**root@zabbixclient\_web1 **~]**# vim **/**usr**/**local**/**bin**/**net\_status**.**sh
2. #**!**/bin/bash
3. **case** $**1** **in**
4. estab**)**
5. ss **-**antp **|**awk '/^ESTAB/{x++} END{print x}'**;;**
6. close\_wait**)**
7. ss **-**antp **|**awk '/^CLOSE-WAIT/{x++} END{print x}'**;;**
8. time\_wait**)**
9. ss **-**antp **|**awk '/^TIME-WAIT/{x++} END{print x}'**;;**
10. esac
11. **[**root@zabbixclient\_web1 **~]**# chmod **+**x **/**usr**/**local**/**bin**/**net\_status**.**sh

测试效果：

1. **[**root@zabbixclient\_web1 **~]**# zabbix\_get **-**s **127.0.0.1** **\**
2. **-**k 'net.status[time\_wait]'

4) 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，如图-39所示。



图-39