TTS 10.0 COOKBOOK

(NSD SECURITY DAY02)

版本编号 10.0

2019-06

达内 IT 培训集团

NSD SECURITY DAY02

1. 案例 1：实现 Zabbix 报警功能

- **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

- **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

- **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。

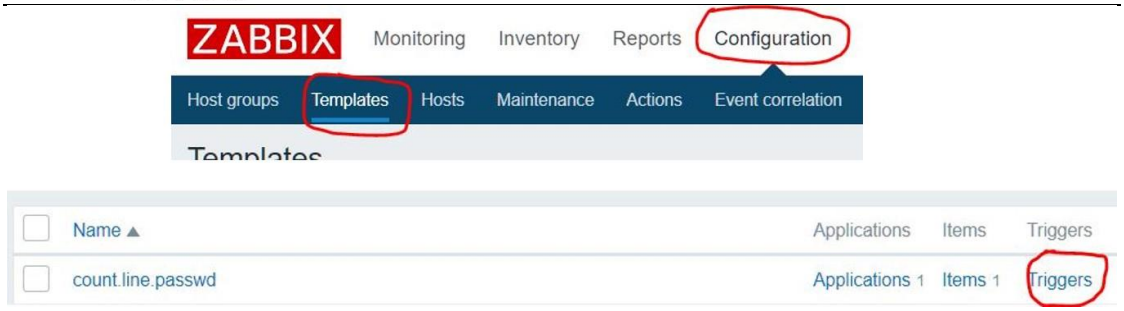

图-1

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。

表-1

函数内容	功能描述
sum(600)	600 秒内所有值的总和
sum(#5)	最后 5 个值的总和
last(20)	最后 20 秒的值
last(#5)	倒数第 5 个值
avg(1h,1d)	一天前的 1 小时的平均值

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

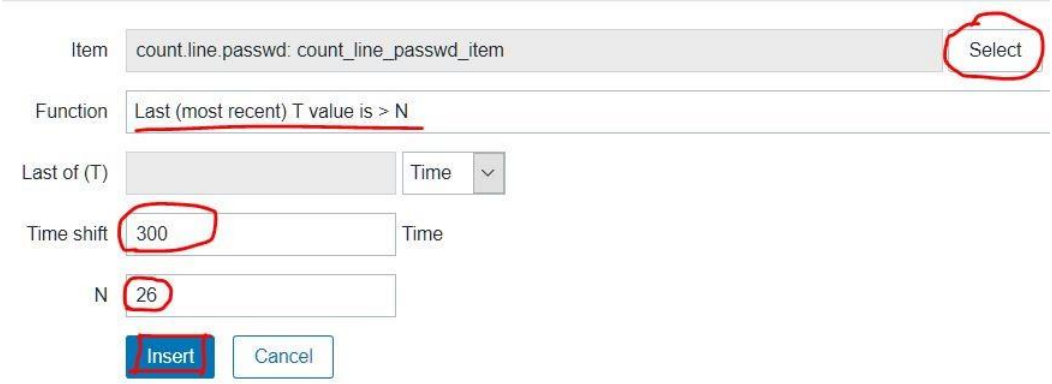

图-4

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。

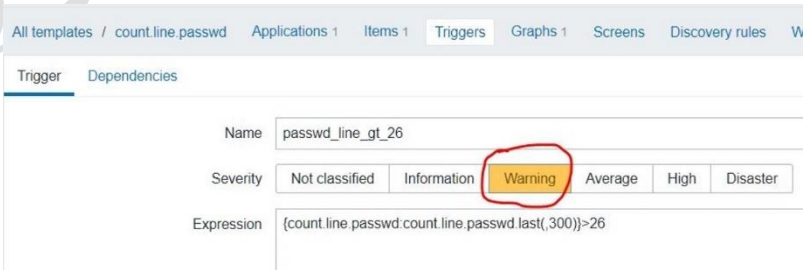

图-5

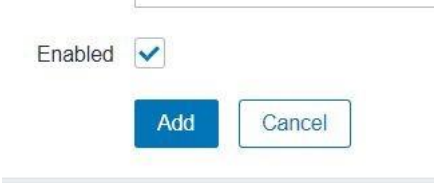

图-6

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。

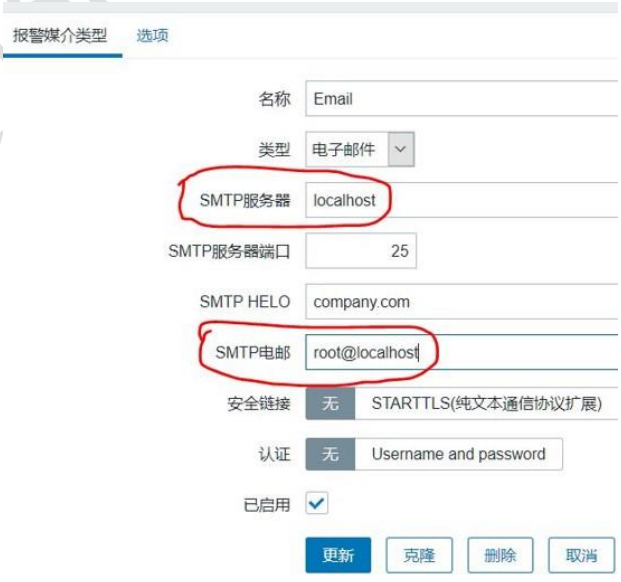

图-8

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。

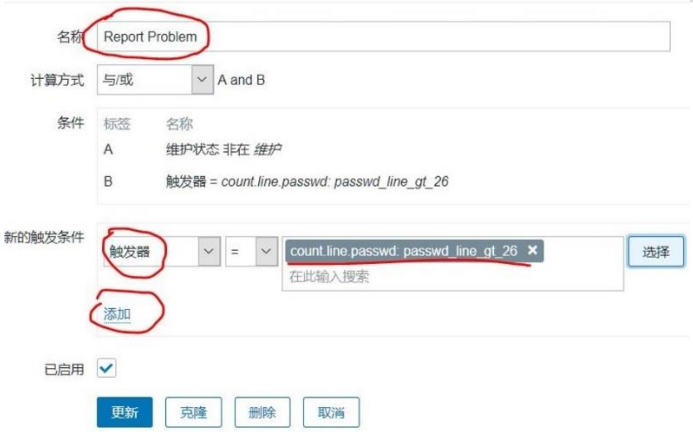

图-13

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。

```
>N 35 root@localhost.local  Sat Feb 17 10:15  20/846   "Problem: passwd_line_gt_26"
 N 36 root@localhost.local  Sat Feb 17 10:15  21/923   "Problem: /etc/passwd has bee
& 35
Message 35:
From root@localhost.localdomain  Sat Feb 17 10:15:41 2018
Return-Path: <root@localhost.localdomain>
X-Original-To: root@localhost
Delivered-To: root@localhost.localdomain
From: <root@localhost.localdomain>
To: <root@localhost.localdomain>
Date: Sat, 17 Feb 2018 10:15:41 -0500
Subject: Problem: passwd_line_gt_26
Content-Type: text/plain; charset="UTF-8"
Status: R

Problem started at 10:13:39 on 2018.02.17
Problem name: passwd_line_gt_26
Host: zabbix_client_web
Severity: Warning
```

图-17

2. 案例 2：Zabbix 自动发现

- **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

- **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

- **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。

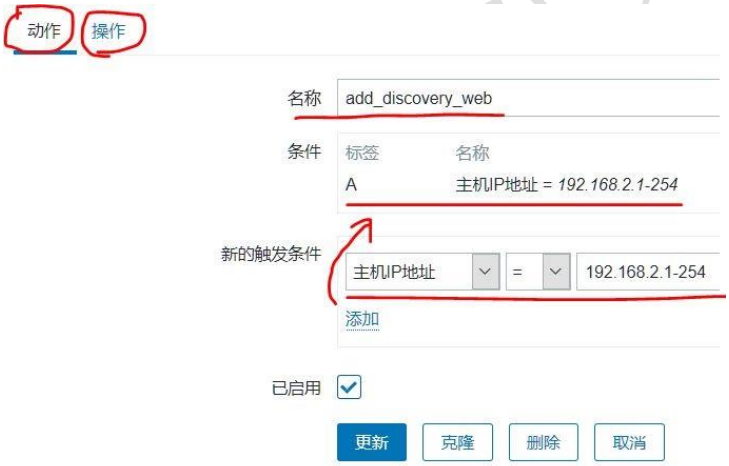

图-22

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。

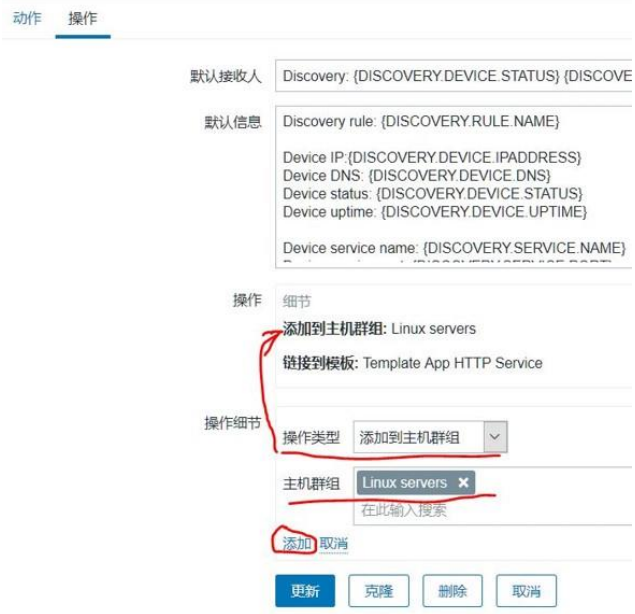

图-23

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

- **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

- **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。

- **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。

```
[root@zabbixclient_web2 ~]# yum -y install gcc pcre-devel
[root@zabbixclient_web2 ~]# tar -xf zabbix-3.4.4.tar.gz
[root@zabbixclient_web2 ~]# cd zabbix-3.4.4/
[root@zabbixclient_web2 ~]#./configure --enable-agent
[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。

```
[root@zabbixclient_web2 ~]# vim /usr/local/etc/zabbix_agentd.conf
#Server=127.0.0.1,192.168.2.5
//注释该行，允许谁监控本机
StartAgents=0
//被动监控时启动多个进程
//设置为 0，则禁止被动监控，不启动 zabbix_agentd 服务
ServerActive=192.168.2.5
//允许哪些主机监控本机（主动模式），一定要取消 127.0.0.1
Hostname=zabbixclient_web2
//告诉监控服务器，是谁发的数据信息
//一定要和 zabbix 服务器配置的监控主机名称一致（后面设置）
RefreshActiveChecks=120
//默认 120 秒检测一次
UnsafeUserParameters=1
//允许自定义 key
Include=/usr/local/etc/zabbix_agentd.conf.d/
[root@zabbixclient_web2 ~]# killall zabbix_agentd                    //关闭服务
[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

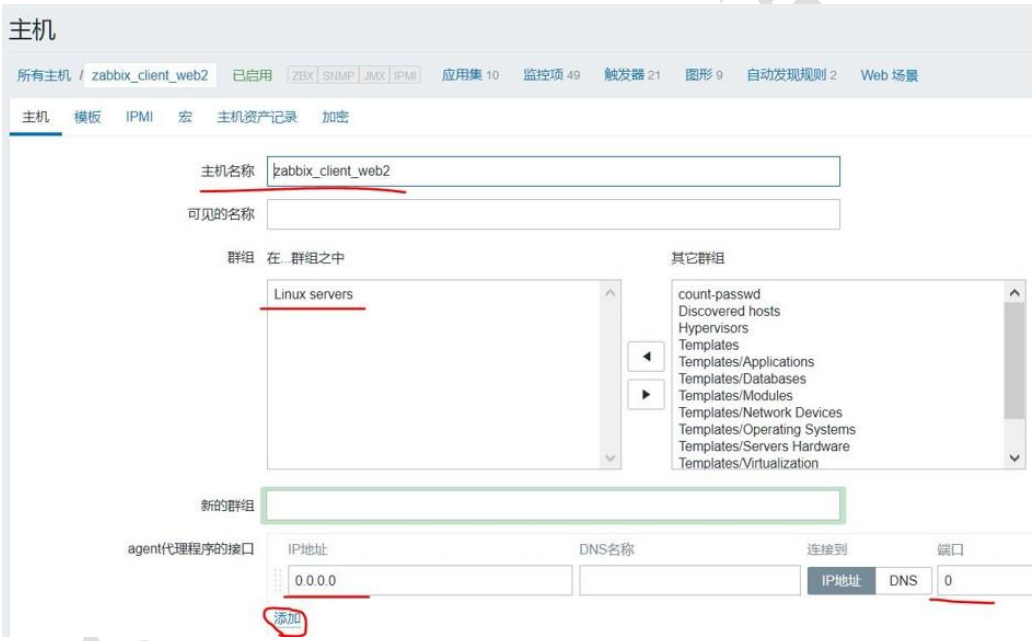

图-28

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。

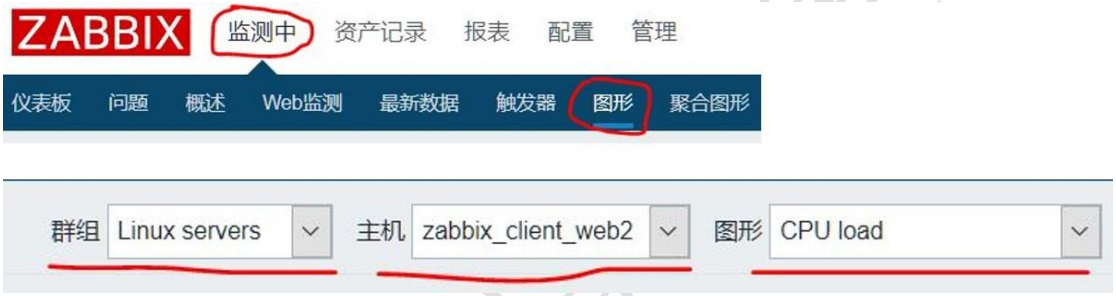

图-30

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

- **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：

4) 创建修改拓扑图

5) 创建聚合图形

- **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。

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。

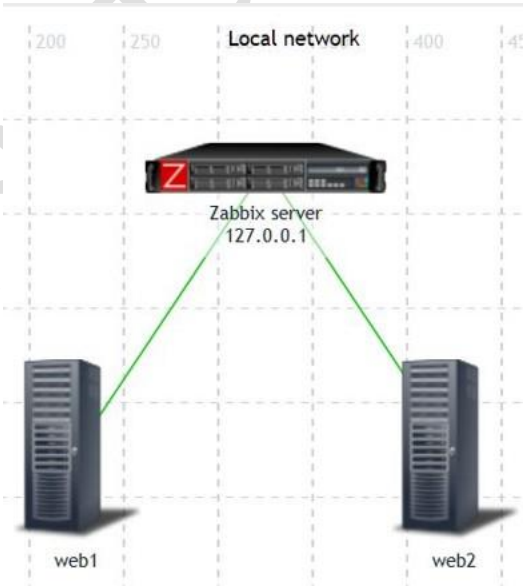

图-33

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。

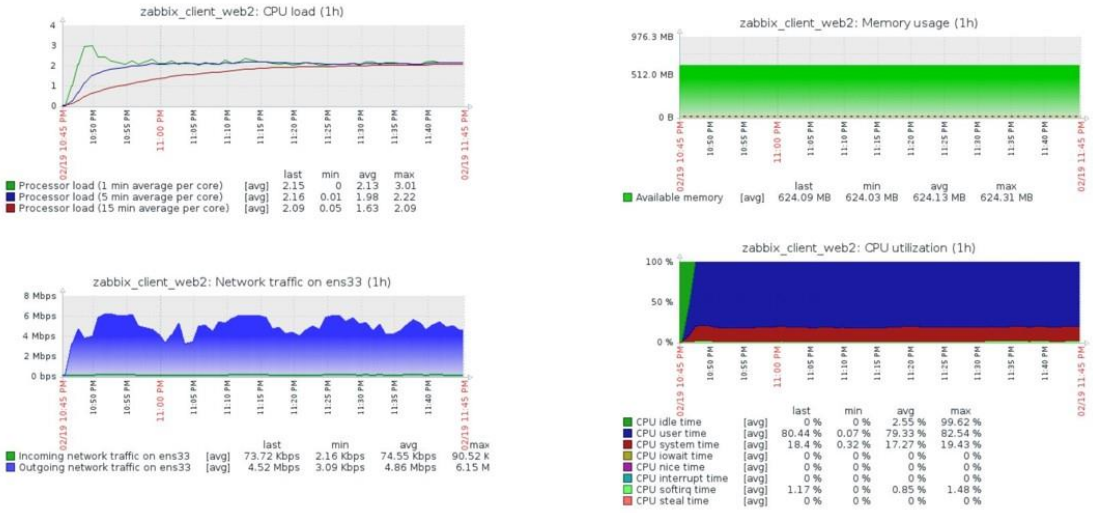

图-35

5. 案例 5：自定义监控案例

- **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

- **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

```
[root@zabbixclient_web1 nginx-1.12.2]# ./configure \
> --with-http_stub_status_module
[root@zabbixclient_web1 nginx-1.12.2]# make && make install
[root@zabbixclient_web1 ~]# cat /usr/local/nginx/conf/nginx.conf
… …
location /status {
               stub_status on;
       }
… …
[root@zabbixclient_web1 ~]# curl  http://192.168.2.100/status
Active connections: 1
server accepts handled requests
10 10 3
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

```
[root@zabbixclient_web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/nginx.status
UserParameter=nginx.status[*],/usr/local/bin/nginx_status.sh $1
[root@zabbixclient_web1 ~]# killall zabbix_agentd
[root@zabbixclient_web1 ~]# zabbix_agentd
```

自定义监控脚本（仅供参考，未检测完整状态）

```
[root@zabbixclient_web1 ~]# vim /usr/local/bin/nginx_status.sh
#!/bin/bash
case $1 in
active)
    curl -s http://192.168.2.100/status |awk '/Active/{print $NF}';;
waiting)
    curl -s http://192.168.2.100/status |awk '/Waiting/{print $NF}';;
accepts)
    curl -s http://192.168.2.100/status |awk 'NR==3{print $2}';;
esac
[root@zabbixclient_web1 ~]# chmod +x  /usr/local/bin/nginx_status.sh
```

测试效果：

```
[root@zabbixclient_web1 ~]# zabbix_get  -s 127.0.0.1 \
 -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。

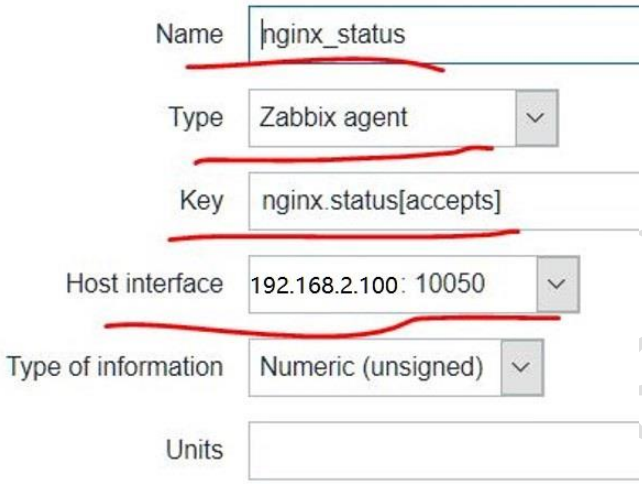

图-36

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。

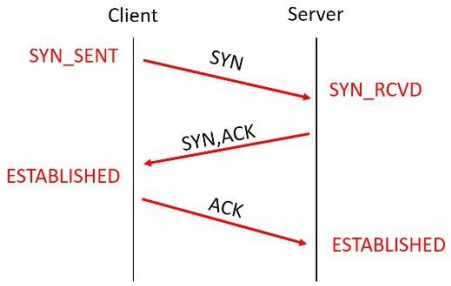

图-37

熟悉 TCP 连接的四次断开，参考图-38。

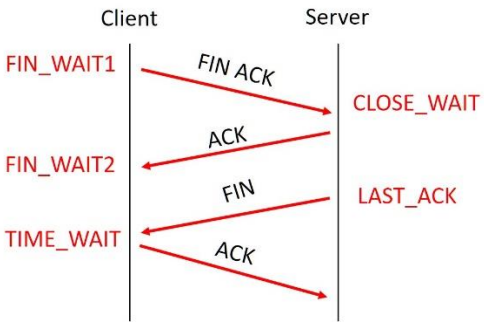

图-38

2）查看网络连接状态

模拟多人并发连接

```
[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

```
[root@zabbixclient_web1 ~]# 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
```

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。

```
[root@zabbixclient_web1 ~]# vim /usr/local/etc/zabbix_agentd.conf.d/net.status
UserParameter=net.status[*],/usr/local/bin/net_status.sh $1

[root@zabbixclient_web1 ~]# killall zabbix_agentd
[root@zabbixclient_web1 ~]# zabbix_agentd
```

自定义监控脚本（仅供参考，未检测完整状态）

```
[root@zabbixclient_web1 ~]# vim /usr/local/bin/net_status.sh
#!/bin/bash
case $1 in
estab)
    ss -antp |awk '/^ESTAB/{x++} END{print x}';;
close_wait)
    ss -antp |awk '/^CLOSE-WAIT/{x++} END{print x}';;
time_wait)
    ss -antp |awk '/^TIME-WAIT/{x++} END{print x}';;
esac
[root@zabbixclient_web1 ~]# chmod +x  /usr/local/bin/net_status.sh
```

测试效果：

```
[root@zabbixclient_web1 ~]# zabbix_get  -s 127.0.0.1 \
 -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。

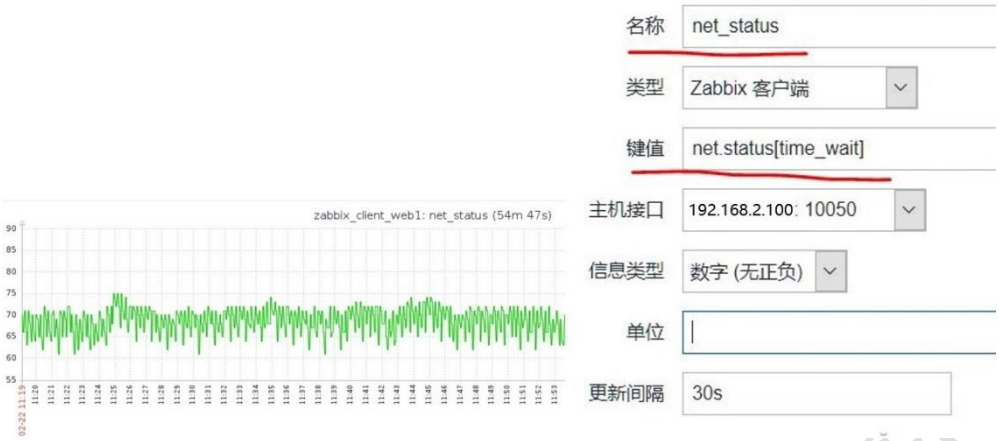

图-39