zabbix监控

1. [案例1：常用系统监控命令](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html" \l "case1)
2. [案例2：部署Zabbix监控平台](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html" \l "case2)
3. [案例3：配置及使用Zabbix监控系统](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html" \l "case3)
4. [案例4：自定义Zabbix监控项目](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html" \l "case4)

## **1 案例1：常用系统监控命令**

### **1.1 问题**

本案例要求熟悉查看Linux系统状态的常用命令，为进一步执行具体的监控任务做准备：

* 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

### **1.2 方案**

一般企业做监控的目的：实时报告系统状态，提前发现系统的问题。

监控的资源可以分为：共有数据（HTTP、FTP等）和私有数据（CPU、内存、进程数等）。

监控软件可以使用：系统自带的命令、Cacti监控系统、Nagios监控系统、Zabbix监控系统。

### **1.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：使用命令查看计算机状态数据

1）查看内存与交换分区信息

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@proxy ~]# free                                     //查看内存信息
2. total used free shared buff/cache available
3. Mem: 16166888 8017696 720016 106504 7429176 7731740
4. Swap: 4194300 218268 3976032
5. [root@proxy ~]# free | awk '/Mem/{print $4}'                //查看剩余内存容量
6. 720928
7. [root@proxy ~]# swapon -s                                //查看交换分区信息
8. 文件名                类型            大小        已用        权限
9. /dev/sda3 partition        4194300    218268    -1

[root@proxy ~]# free //查看内存信息

total used free shared buff/cache available

Mem: 16166888 8017696 720016 106504 7429176 7731740

Swap: 4194300 218268 3976032

[root@proxy ~]# free | awk '/Mem/{print $4}' //查看剩余内存容量

720928

[root@proxy ~]# swapon -s //查看交换分区信息

文件名 类型 大小 已用 权限

/dev/sda3 partition 4194300 218268 -1

步骤二：查看磁盘与CPU利用率

1）查看磁盘信息

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@proxy ~]# df                                     //查看所有磁盘的使用率
2. 文件系统 1K-块 已用 可用         已用% 挂载点
3. /dev/sda2 476254208 116879624 335159084 26%         /
4. /dev/sda1 198174 133897 49737 73%         /boot
5. [root@proxy ~]# df | awk '/\/$/{print $5}'            //查看根分区的利用率

[root@proxy ~]# df //查看所有磁盘的使用率

文件系统 1K-块 已用 可用 已用% 挂载点

/dev/sda2 476254208 116879624 335159084 26% /

/dev/sda1 198174 133897 49737 73% /boot

[root@proxy ~]# df | awk '/\/$/{print $5}' //查看根分区的利用率

2）查看CPU平均负载

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@proxy ~]# uptime                             //查看CPU负载（1，5，15分钟）
2. 23:54:12 up 38 days, 14:54, 9 users, load average: 0.00, 0.04, 0.05
3. [root@proxy ~]# uptime |awk '{print $NF}'            //仅查看CPU的15分钟平均负载
4. 0.05

[root@proxy ~]# uptime //查看CPU负载（1，5，15分钟）

23:54:12 up 38 days, 14:54, 9 users, load average: 0.00, 0.04, 0.05

[root@proxy ~]# uptime |awk '{print $NF}' //仅查看CPU的15分钟平均负载

0.05

步骤二：查看网卡信息、端口信息、网络连接信息

1）查看网卡信息

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@proxy ~]# ifconfig eth0
2. eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3. inet 192.168.4.5 netmask 255.255.255.0 broadcast 172.25.0.255
4. inet6 fe80::5054:ff:fe00:b prefixlen 64 scopeid 0x20<link>
5. ether 52:54:00:00:00:0b txqueuelen 1000 (Ethernet)
6. RX packets 62429 bytes 10612049 (10.1 MiB)
7. RX errors 0 dropped 0 overruns 0 frame 0
8. TX packets 5674 bytes 4121143 (3.9 MiB)
9. TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10. [root@proxy ~]# ifconfig eth0 |awk '/inet /{print $2}'        //查看IP地址信息
11. 192.168.4.5
12. [root@proxy ~]# ifconfig eth0 |awk '/RX p/{print $5}'        //网卡接受数据包流量
13. 10625295
14. [root@proxy ~]# ifconfig eth0 |awk '/TX p/{print $5}'        //网卡发送数据包流量
15. 4130821

[root@proxy ~]# ifconfig eth0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.4.5 netmask 255.255.255.0 broadcast 172.25.0.255

inet6 fe80::5054:ff:fe00:b prefixlen 64 scopeid 0x20<link>

ether 52:54:00:00:00:0b txqueuelen 1000 (Ethernet)

RX packets 62429 bytes 10612049 (10.1 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 5674 bytes 4121143 (3.9 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@proxy ~]# ifconfig eth0 |awk '/inet /{print $2}' //查看IP地址信息

192.168.4.5

[root@proxy ~]# ifconfig eth0 |awk '/RX p/{print $5}' //网卡接受数据包流量

10625295

[root@proxy ~]# ifconfig eth0 |awk '/TX p/{print $5}' //网卡发送数据包流量

4130821

2）查看端口信息

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@proxy ~]# ss -ntulp                                //查看本机监听的所有端口
2. //-n以数字显示端口号
3. //-t显示tcp连接
4. //-u显示udp连接
5. //-p显示监听端口对应的程序名称

[root@proxy ~]# ss -ntulp //查看本机监听的所有端口

//-n以数字显示端口号

//-t显示tcp连接

//-u显示udp连接

//-p显示监听端口对应的程序名称

3）查看网络连接信息

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@proxy ~]# ss -antup                                //查看所有的网络连接信息
2. //-a查看所有连接状态信息

[root@proxy ~]# ss -antup //查看所有的网络连接信息

//-a查看所有连接状态信息

## **2 案例2：部署Zabbix监控平台**

### **2.1 问题**

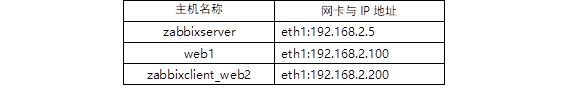
本案例要求部署一台Zabbix监控服务器，一台被监控主机，为进一步执行具体的监控任务做准备：

1. 安装LNMP环境
2. 源码安装Zabbix
3. 安装监控端主机，修改基本配置
4. 初始化Zabbix监控Web页面
5. 修改PHP配置文件，满足Zabbix需求
6. 安装被监控端主机，修改基本配置

### **2.2 方案**

使用1台RHEL7虚拟机，安装部署LNMP环境、Zabbix及相关的依赖包，配置数据库并对Zabbix监控平台进行初始化操作。使用2台被监控端，源码安装Zabbix Agent。完成Zabbix实验需要我们搭建一个实验环境，拓扑结构如表-1所示。

表-1 实验拓扑结构



### **2.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：部署监控服务器

1）安装LNMP环境

Zabbix监控管理控制台需要通过Web页面展示出来，并且还需要使用MySQL来存储数据，因此需要先为Zabbix准备基础LNMP环境。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# yum -y install gcc pcre-devel openssl-devel
2. [root@zabbixserver ~]# tar -xf nginx-1.12.2.tar.gz
3. [root@zabbixserver ~]# cd nginx-1.12.2
4. [root@zabbixserver nginx-1.12.2]# ./configure --with-http\_ssl\_module
5. [root@zabbixserver nginx-1.12.2]# make && make install
6. [root@zabbixserver ~]# yum -y install php php-mysql \
7. > mariadb mariadb-devel mariadb-server
8. [root@zabbixserver ~]# yum -y install php-fpm-5.4.16-42.el7.x86\_64.rpm
9. //注意，php-fpm这个软件包在lnmp\_soft/目录下

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

[root@zabbixserver ~]# tar -xf nginx-1.12.2.tar.gz

[root@zabbixserver ~]# cd nginx-1.12.2

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

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

[root@zabbixserver ~]# yum -y install php php-mysql \

> mariadb mariadb-devel mariadb-server

[root@zabbixserver ~]# yum -y install php-fpm-5.4.16-42.el7.x86\_64.rpm

//注意，php-fpm这个软件包在lnmp\_soft/目录下

2）修改Nginx配置文件

配置Nginx支持PHP动态网站，因为有大量PHP脚本需要执行，因此还需要开启Nginx的各种fastcgi缓存，加速PHP脚本的执行速度。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# vim /usr/local/nginx/conf/nginx.conf
2. … …
3. http{
4. … …
5. fastcgi\_buffers 8 16k;                 #缓存php生成的页面内容，8个16k
6. fastcgi\_buffer\_size 32k;                 #缓存php生产的头部信息
7. fastcgi\_connect\_timeout 300;             #连接PHP的超时时间
8. fastcgi\_send\_timeout 300;                 #发送请求的超时时间
9. fastcgi\_read\_timeout 300;                     #读取请求的超时时间
10. location ~ \.php$ {
11. root html;
12. fastcgi\_pass 127.0.0.1:9000;
13. fastcgi\_index index.php;
14. include fastcgi.conf;
15. }
16. … …

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

… …

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

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf;

}

… …

3）启动服务

启动Nginx、PHP-FPM、MariaDB服务，关闭SELinux与防火墙。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# systemctl start mariadb
2. [root@zabbixserver ~]# systemctl start php-fpm
3. [root@zabbixserver ~]# ln -s /usr/local/nginx/sbin/nginx /sbin/nginx
4. [root@zabbixserver ~]# nginx
6. [root@zabbixserver ~]# firewall-cmd --set-default-zone=trusted
7. [root@zabbixserver ~]# setenforce 0

[root@zabbixserver ~]# systemctl start mariadb

[root@zabbixserver ~]# systemctl start php-fpm

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

[root@zabbixserver ~]# nginx

[root@zabbixserver ~]# firewall-cmd --set-default-zone=trusted

[root@zabbixserver ~]# setenforce 0

4）客户端测试LNMP环境

服务器创建PHP测试页面，浏览器访问页面测试网页连通性。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# cat /usr/local/nginx/html/test.php
2. <?php
3. $i=33;
4. echo $i;
5. ?>
6. [root@zabbixserver ~]# curl http://192.168.2.5/test.php

[root@zabbixserver ~]# cat /usr/local/nginx/html/test.php

<?php

$i=33;

echo $i;

?>

[root@zabbixserver ~]# curl http://192.168.2.5/test.php

步骤二：部署监控服务器Zabbix Server

1）源码安装Zabbix Server

多数源码包都是需要依赖包的，zabbix也一样，源码编译前需要先安装相关依赖包。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver lnmp\_soft]# yum -y install net-snmp-devel \
2. > curl-devel autoconf
3. //安装相关依赖包
4. [root@zabbixserver lnmp\_soft]# yum -y install \
5. > libevent-devel-2.0.21-4.el7.x86\_64.rpm
6. //注意libevent-devel这个软件包在lnmp\_soft目录下有提供
7. [root@zabbixserver lnmp\_soft]# tar -xf zabbix-3.4.4.tar.gz
8. [root@zabbixserver lnmp\_soft]# cd zabbix-3.4.4/
9. [root@zabbixserver zabbix-3.4.4]# ./configure --enable-server \
10. > --enable-proxy --enable-agent --with-mysql=/usr/bin/mysql\_config \
11. > --with-net-snmp --with-libcurl
12. // --enable-server安装部署zabbix服务器端软件
13. // --enable-agent安装部署zabbix被监控端软件
14. // --enable-proxy安装部署zabbix代理相关软件
15. // --with-mysql配置mysql\_config路径
16. // --with-net-snmp允许zabbix通过snmp协议监控其他设备
17. // --with-libcurl安装相关curl库文件，这样zabbix就可以通过curl连接http等服务，测试被监控主机服务的状态
18. [root@zabbixserver zabbix-3.4.4]# make && make install

[root@zabbixserver lnmp\_soft]# yum -y install net-snmp-devel \

> curl-devel autoconf

//安装相关依赖包

[root@zabbixserver lnmp\_soft]# yum -y install \

> libevent-devel-2.0.21-4.el7.x86\_64.rpm

//注意libevent-devel这个软件包在lnmp\_soft目录下有提供

[root@zabbixserver lnmp\_soft]# tar -xf zabbix-3.4.4.tar.gz

[root@zabbixserver lnmp\_soft]# cd zabbix-3.4.4/

[root@zabbixserver zabbix-3.4.4]# ./configure --enable-server \

> --enable-proxy --enable-agent --with-mysql=/usr/bin/mysql\_config \

> --with-net-snmp --with-libcurl

// --enable-server安装部署zabbix服务器端软件

// --enable-agent安装部署zabbix被监控端软件

// --enable-proxy安装部署zabbix代理相关软件

// --with-mysql配置mysql\_config路径

// --with-net-snmp允许zabbix通过snmp协议监控其他设备

// --with-libcurl安装相关curl库文件，这样zabbix就可以通过curl连接http等服务，测试被监控主机服务的状态

[root@zabbixserver zabbix-3.4.4]# make && make install

2）创建并初始化数据库

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# mysql
2. mysql> create database zabbix character set utf8;
3. #创建数据库，支持中文字符集
4. mysql> grant all on zabbix.\* to zabbix@'localhost' identified by 'zabbix';
5. #创建可以访问数据库的账户与密码
6. [root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/database/mysql/
7. [root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < schema.sql
8. [root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < images.sql
9. [root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < data.sql
10. //刚刚创建是空数据库，zabbix源码包目录下，有提前准备好的数据
11. //使用mysql导入这些数据即可（注意导入顺序）

[root@zabbixserver ~]# mysql

mysql> create database zabbix character set utf8;

#创建数据库，支持中文字符集

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

#创建可以访问数据库的账户与密码

[root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/database/mysql/

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

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

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

//刚刚创建是空数据库，zabbix源码包目录下，有提前准备好的数据

//使用mysql导入这些数据即可（注意导入顺序）

3）修改zabbix\_server配置并启动监控服务

修改Zabbix\_server配置文件，设置数据库相关参数，启动Zabbix\_server服务

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# vim /usr/local/etc/zabbix\_server.conf
2. DBHost=localhost
3. #数据库主机，默认该行被注释
4. DBName=zabbix
5. #设置数据库名称
6. DBUser=zabbix
7. #设置数据库账户
8. DBPassword=zabbix
9. #设置数据库密码，默认该行被注释
10. LogFile=/tmp/zabbix\_server.log
11. #设置日志，仅查看以下即可
12. [root@zabbixserver ~]# useradd -s /sbin/nologin zabbix
13. #不创建用户无法启动服务
14. [root@zabbixserver ~]# zabbix\_server                 #启动服务
16. [root@zabbixserver ~]# echo zabbix\_server >> /etc/rc.local #设置开机自启
17. [root@zabbixserver ~]# chmod +x /etc/rc.local
18. [root@zabbixserver ~]# ss -ntulp |grep zabbix\_server #确认连接状态，端口10051
19. tcp LISTEN 0 128 \*:10051 \*:\* users:(("zabbix\_server",pid=23275,fd=4),("zabbix\_server",pid=23274,fd=4)

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

DBHost=localhost

#数据库主机，默认该行被注释

DBName=zabbix

#设置数据库名称

DBUser=zabbix

#设置数据库账户

DBPassword=zabbix

#设置数据库密码，默认该行被注释

LogFile=/tmp/zabbix\_server.log

#设置日志，仅查看以下即可

[root@zabbixserver ~]# useradd -s /sbin/nologin zabbix

#不创建用户无法启动服务

[root@zabbixserver ~]# zabbix\_server #启动服务

[root@zabbixserver ~]# echo zabbix\_server >> /etc/rc.local #设置开机自启

[root@zabbixserver ~]# chmod +x /etc/rc.local

[root@zabbixserver ~]# ss -ntulp |grep zabbix\_server #确认连接状态，端口10051

tcp LISTEN 0 128 \*:10051 \*:\* users:(("zabbix\_server",pid=23275,fd=4),("zabbix\_server",pid=23274,fd=4)

提示：如果是因为配置文件不对，导致服务无法启动时，不要重复执行zabbix\_server，

一定要先使用killall zabbix\_server关闭服务后，再重新启动一次。

4) 修改Zabbix\_agent配置文件，启动Zabbix\_agent服务

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# vim /usr/local/etc/zabbix\_agentd.conf
2. Server=127.0.0.1,192.168.2.5                    //允许哪些主机监控本机
3. ServerActive=127.0.0.1,192.168.2.5                //允许哪些主机通过主动模式监控本机
4. Hostname=zabbix\_server                        //设置本机主机名
5. LogFile=/tmp/zabbix\_server.log                    //设置日志文件
6. UnsafeUserParameters=1                        //是否允许自定义key
7. [root@zabbixserver ~]# zabbix\_agentd            //启动监控agent
9. [root@zabbixserver ~]# echo zabbix\_agentd >> /etc/rc.local #设置开机自启
11. [root@zabbixserver ~]# ss -ntulp |grep zabbix\_agentd //查看端口信息为10050
12. tcp LISTEN 0 128 \*:10050 \*:\* users:(("zabbix\_agentd",pid=23505,fd=4),("zabbix\_agentd",pid=23504,fd=4)

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

Server=127.0.0.1,192.168.2.5 //允许哪些主机监控本机

ServerActive=127.0.0.1,192.168.2.5 //允许哪些主机通过主动模式监控本机

Hostname=zabbix\_server //设置本机主机名

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

UnsafeUserParameters=1 //是否允许自定义key

[root@zabbixserver ~]# zabbix\_agentd //启动监控agent

[root@zabbixserver ~]# echo zabbix\_agentd >> /etc/rc.local #设置开机自启

[root@zabbixserver ~]# ss -ntulp |grep zabbix\_agentd //查看端口信息为10050

tcp LISTEN 0 128 \*:10050 \*:\* users:(("zabbix\_agentd",pid=23505,fd=4),("zabbix\_agentd",pid=23504,fd=4)

提示：如果是因为配置文件不对，导致服务无法启动时，不要重复执行zabbix\_agentd，

一定要先使用killall zabbix\_agentd关闭服务后，再重新启动一次。

5)上线Zabbix的Web页面

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/frontends/php/
2. [root@zabbixserver php]# cp -r \* /usr/local/nginx/html/
3. [root@zabbixserver php]# chown -R apache.apache /usr/local/nginx/html/

[root@zabbixserver ~]# cd lnmp\_soft/zabbix-3.4.4/frontends/php/

[root@zabbixserver php]# cp -r \* /usr/local/nginx/html/

[root@zabbixserver php]# chown -R apache.apache /usr/local/nginx/html/

浏览器访问Zabbix\_server服务器的Web页面

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# firefox http://192.168.2.5/index.php
2. //第一次访问，初始化PHP页面会检查计算机环境是否满足要求，如果不满足会给出修改建议
3. //默认会提示PHP的配置不满足环境要求，需要修改PHP配置文件

[root@zabbixserver ~]# firefox http://192.168.2.5/index.php

//第一次访问，初始化PHP页面会检查计算机环境是否满足要求，如果不满足会给出修改建议

//默认会提示PHP的配置不满足环境要求，需要修改PHP配置文件

根据错误提示，修改PHP配置文件，满足Zabbix\_server的Web环境要求。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@zabbixserver ~]# yum -y install php-gd php-xml
2. [root@zabbixserver ~]# yum install php-bcmath
3. [root@zabbixserver ~]# yum install php-mbstring
4. [root@zabbixserver ~]# vim /etc/php.ini
5. date.timezone = Asia/Shanghai                //设置时区
6. max\_execution\_time = 300                    //最大执行时间，秒
7. post\_max\_size = 32M                        //POST数据最大容量
8. max\_input\_time = 300                        //服务器接收数据的时间限制
9. memory\_limit = 128M                        //内存容量限制
10. [root@zabbixserver ~]# systemctl restart php-fpm

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

[root@zabbixserver ~]# yum install php-bcmath

[root@zabbixserver ~]# yum install php-mbstring

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

date.timezone = Asia/Shanghai //设置时区

max\_execution\_time = 300 //最大执行时间，秒

post\_max\_size = 32M //POST数据最大容量

max\_input\_time = 300 //服务器接收数据的时间限制

memory\_limit = 128M //内存容量限制

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

修改完PHP配置文件后，再次使用浏览器访问服务器，则会提示如图-1和图-2所示的提示信息。

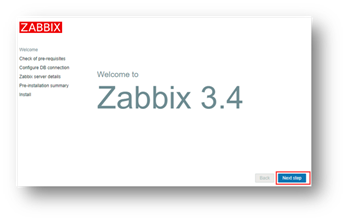


图-1

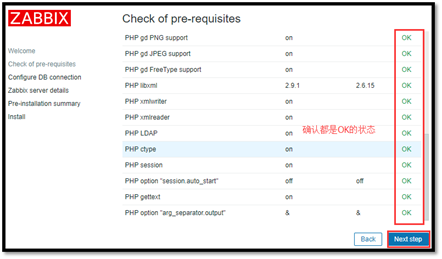


图-2

注意：这里有一个PHP LDAP是warning状态是没有问题的！

在初始化数据库页面，填写数据库相关参数，如图-3所示。

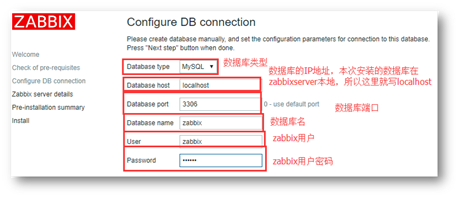


图-3

在登陆页面，使用用户(admin)和密码(zabbix)登陆，登陆后设置语言环境为中文，如图-4和图-5所示。

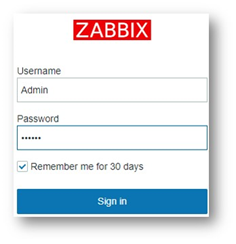


图-4

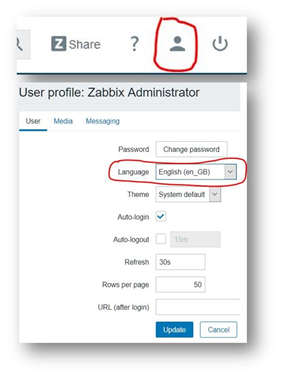


图-5

步骤三：部署被监控主机Zabbix Agent

1）源码安装Zabbix agent软件

在2.100和2.200做相同操作（以web1为例）。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@web1 ~]# useradd -s /sbin/nologin zabbix
2. [root@web1 ~]# yum -y install gcc pcre-devel autoconf
3. [root@web1 ~]# tar -xf zabbix-3.4.4.tar.gz
4. [root@web1 ~]# cd zabbix-3.4.4/
5. [root@web1 zabbix-3.4.4]# ./configure --enable-agent
6. [root@web1 zabbix-3.4.4]# make && make install

[root@web1 ~]# useradd -s /sbin/nologin zabbix

[root@web1 ~]# yum -y install gcc pcre-devel autoconf

[root@web1 ~]# tar -xf zabbix-3.4.4.tar.gz

[root@web1 ~]# cd zabbix-3.4.4/

[root@web1 zabbix-3.4.4]# ./configure --enable-agent

[root@web1 zabbix-3.4.4]# make && make install

2）修改agent配置文件，启动Agent

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf
2. Server=127.0.0.1,192.168.2.5                //谁可以监控本机（被动监控模式）
3. ServerActive=127.0.0.1,192.168.2.5            //谁可以监控本机（主动监控模式）
4. Hostname=web1                 //被监控端自己的主机名
5. EnableRemoteCommands=1
6. //监控异常后，是否允许服务器远程过来执行命令，如重启某个服务
7. UnsafeUserParameters=1                     //是否允许自定义key监控
8. [root@web1 ~]# zabbix\_agentd                //启动agent服务
9. [root@web1 ~]# echo zabbix\_agentd >> /etc/rc.local
10. [root@web1 ~]# chmod +x /etc/rc.local

[root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf

Server=127.0.0.1,192.168.2.5 //谁可以监控本机（被动监控模式）

ServerActive=127.0.0.1,192.168.2.5 //谁可以监控本机（主动监控模式）

Hostname=web1 //被监控端自己的主机名

EnableRemoteCommands=1

//监控异常后，是否允许服务器远程过来执行命令，如重启某个服务

UnsafeUserParameters=1 //是否允许自定义key监控

[root@web1 ~]# zabbix\_agentd //启动agent服务

[root@web1 ~]# echo zabbix\_agentd >> /etc/rc.local

[root@web1 ~]# chmod +x /etc/rc.local

## **3 案例3：配置及使用Zabbix监控系统**

### **3.1 问题**

沿用练习一，使用Zabbix监控平台监控Linux服务器，实现以下目标：

1. 监控CPU
2. 监控内存
3. 监控进程
4. 监控网络流量
5. 监控硬盘

### **3.2 方案**

通过Zabbix监控平台，添加被监控web1主机（192.168.2.100）并链接监控模板即可，Zabbix默认模板就可以监控CPU、内存、进程、网络、磁盘等项目。

### **3.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：添加监控主机

主机是Zabbix监控的基础，Zabbix所有监控都是基于Host主机。

使用火狐浏览器登录http://192.168.2.5，通过Configuration（配置）-->Hosts（主机）-->Create Host（创建主机）添加被监控Linux主机，如图-7所示。



图-7

添加被监控主机时，需要根据提示输入被监控Linux主机的主机名称（最好与电脑的主机名一致，但也允许不一致）、主机组、IP地址等参数，具体参考图-8所示。

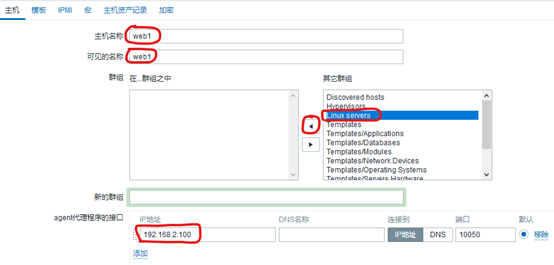


图-8

步骤二：为被监控主机添加监控模板

Zabbix通过监控模板来对监控对象实施具体的监控功能，根据模板来定义需要监控哪些数据，对于Linux服务器的监控，Zabbix已经内置了相关的模板（Template OS Linux），选择模板并链接到主机即可，如图-9所示。



图-9

步骤三：查看监控数据

查看监控数据，登录Zabbix Web控制台，点击Monitoring(监控中)—> Latest data(最新数据)，正过滤器中填写过滤条件，根据监控组和监控主机选择需要查看哪些监控数据，如图-10所示。



图-10

找到需要监控的数据后，可以点击后面的Graph查看监控图形，如图-11所示。

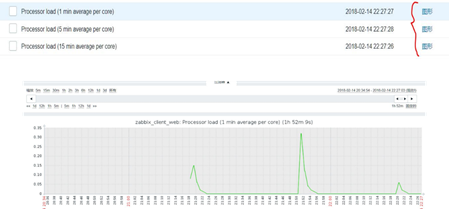


图-11

## **4 案例4：自定义Zabbix监控项目**

### **4.1 问题**

沿用练习二，使用Zabbix实现自定义监控，实现以下目标：监控Linux服务器系统账户的数量。

### **4.2 方案**

需要使用Zabbix自定义key的方式实现自定义监控，参考如下操作步骤：

1. 创建自定义key
2. 创建监控项目
3. 创建监控图形
4. 将监控模板关联到主机

### **4.3 步骤**

实现此案例需要按照如下步骤进行。

步骤一：被监控主机创建自定义key（在192.168.2.100操作）

1）创建自定义key

自定义key语法格式为：UserParameter=自定义key名称,命令。

自定义的key文件一般存储在/usr/local/etc/zabbix\_agentd.conf.d/目录，这里还需要修改zabbix\_agentd.conf文件，允许自定义监控key，来读取该目录下的所有文件 。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf
2. Include=/usr/local/etc/zabbix\_agentd.conf.d/                //加载配置文件目录
3. [root@web1 ~]# cd /usr/local/etc/zabbix\_agentd.conf.d/
4. [root@web1 zabbix\_agentd.conf.d]# vim count.line.passwd
5. UserParameter=count.line.passwd,sed -n '$=' /etc/passwd
6. ////自定义key语法格式:
7. //UserParameter=自定义key名称,命令

[root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf

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

[root@web1 ~]# cd /usr/local/etc/zabbix\_agentd.conf.d/

[root@web1 zabbix\_agentd.conf.d]# vim count.line.passwd

UserParameter=count.line.passwd,sed -n '$=' /etc/passwd

////自定义key语法格式:

//UserParameter=自定义key名称,命令

2）测试自定义key是否正常工作

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@web1 ~]# killall zabbix\_agentd
2. [root@web1 ~]# zabbix\_agentd                                //重启agent服务
3. [root@web1 ~]# zabbix\_get -s 127.0.0.1 -k count.line.passwd
4. 21

[root@web1 ~]# killall zabbix\_agentd

[root@web1 ~]# zabbix\_agentd //重启agent服务

[root@web1 ~]# zabbix\_get -s 127.0.0.1 -k count.line.passwd

21

注意：如zabbix\_get命令执行错误，提示Check access restrictions in Zabbix agent configuration，则需要检查agent配置文件是否正确：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY01/CASE/01/index.html)

1. [root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf
2. Server=127.0.0.1,192.168.2.5
3. ServerActive=127.0.0.1,192.168.2.5

[root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf

Server=127.0.0.1,192.168.2.5

ServerActive=127.0.0.1,192.168.2.5

步骤二：创建监控模板

模板、应用集与监控项目的关系图，参考图-12所示

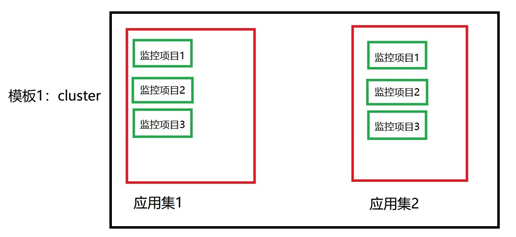


图-12

1）添加监控模板

登录Zabbix Web监控控制台，通过Configuration(配置)-->Template(模板)-->Create template(创建模板)，填写模板名称，新建模板群组，如图-13所示。



图-13

创建模板后，默认模板中没有任何应用、项目、触发器、图形等，如图-14所示。

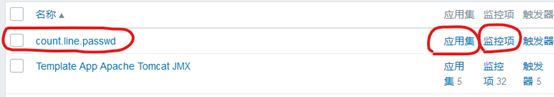


图-14

2）创建应用

创建完成模板后，默认模板中没有任何应用、项目、触发器、图形等资源。这里需要点击模板后面的Application（应用集）链接打开创建应用的页面，如图-15所示。

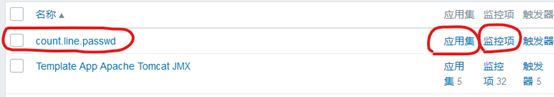


图-15

点击Application（应用集）后，会刷新出图-16所示页面，在该页面中点击Create application（创建应用集）按钮。

IMG_271

图-16

设置应用名称如图-17所示。



图-17

3）创建监控项目item（监控项）

与创建应用一样，在模板中还需要创建监控项目，如图-18所示，并在刷新出的新页面中选择Create items（创建监控项）创建项目，如图-19所示。

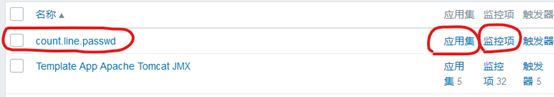


图-18

IMG_274

图-19

接下来，还需要给项目设置名称及对应的自定义key，如图-20所示。



图-20

4）创建图形

为了后期可以通过图形的方式展示监控数据，还需要在模板中创建图形，设置方法与前面的步骤一致，在监控模板后面点击Graph（图形）即可创建图形，设置监控图形基于什么监控数据，如图-21所示。



图-21

5）将模板链接到被监控主机

将完整的监控模板制作完成后，就可以将模板链接到主机实现监控功能了。首先找到被监控主机Configuration（配置）-->Hosts（主机），如图-22所示。



图-22

点击需要的被监控主机链接，打开监控主机设置页面，在Template（模板）页面中选择需要链接到该主机的模板，在此选择刚刚创建的模板count\_line.passwd添加即可，如图-23所示。



图-23

6）查看监控数据图形

点击Monitoring（监控中）-->Craphs（图形），根据需要选择条件，查看监控图形，如图-24和图-25所示。





图

# NSD SECURITY DAY02

1. [案例1：实现Zabbix报警功能](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html" \l "case1)
2. [案例2：Zabbix自动发现](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html" \l "case2)
3. [案例3：Zabbix主动监控](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html" \l "case3)
4. [案例4：拓扑图与聚合图形](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html" \l "case4)
5. [案例5：自定义监控案例](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html" \l "case5)

## **1 案例1：实现Zabbix报警功能**

### **1.1 问题**

沿用前面的Zabbix练习环境，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35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。

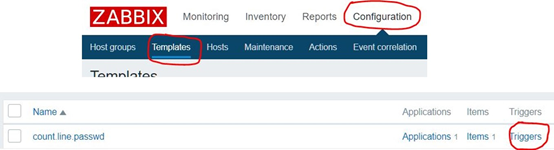


图-1

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，最近账户数量大于35（根据系统账户数量实际填写），效果如图-4所示。

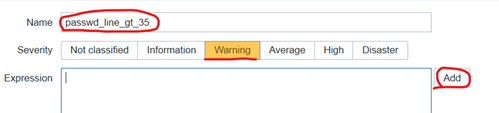


图-3

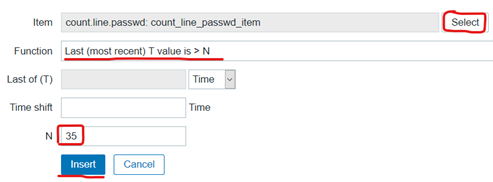


图-4

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。

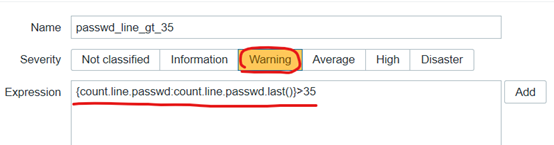


图-5

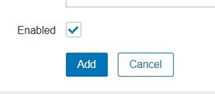


图-6

**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。

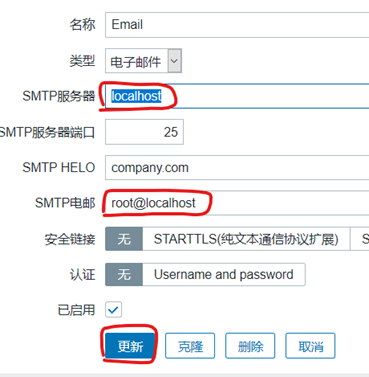


图-8

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35），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。

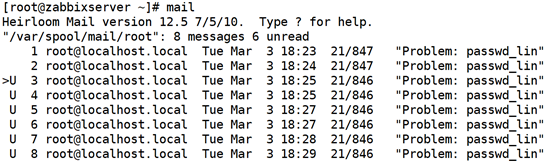


图-17

## **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。

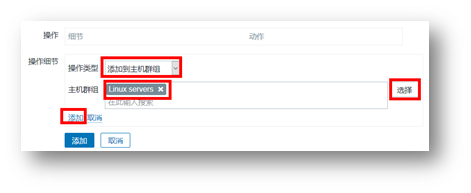


图-23

**步骤二：添加新的虚拟机**

1）创建新的虚拟机

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，则先在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：前面的实验如果我们已经在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。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@web2 ~]# yum -y install gcc pcre-devel autoconf
2. [root@web2 ~]# tar -xf zabbix-3.4.4.tar.gz
3. [root@web2 ~]# cd zabbix-3.4.4/
4. [root@web2 ~]#./configure --enable-agent
5. [root@web2 ~]# make && make install

[root@web2 ~]# yum -y install gcc pcre-devel autoconf

[root@web2 ~]# tar -xf zabbix-3.4.4.tar.gz

[root@web2 ~]# cd zabbix-3.4.4/

[root@web2 ~]#./configure --enable-agent

[root@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。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@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=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@web2 ~]# killall zabbix\_agentd                    //关闭服务
18. [root@web2 ~]# zabbix\_agentd                            //启动服务

[root@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=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@web2 ~]# killall zabbix\_agentd //关闭服务

[root@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 Server Active。



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：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

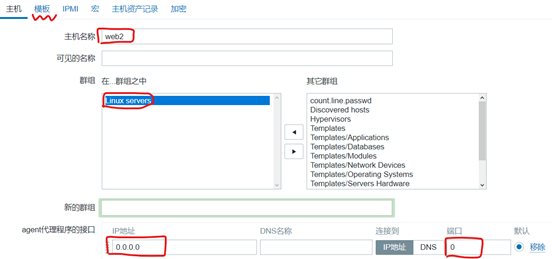


图-28

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。

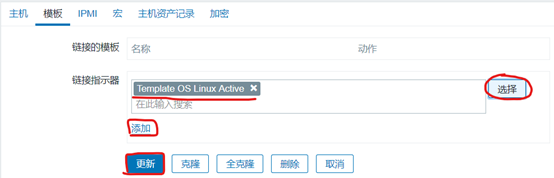


图-29

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 Server Active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。

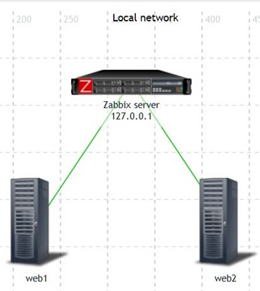


图-33

**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（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。

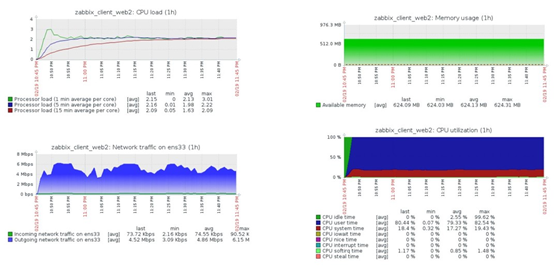


图-35

## **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

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@web1 nginx-1.12.2]# ./configure \
2. > --with-http\_stub\_status\_module
3. [root@web1 nginx-1.12.2]# make && make install
4. [root@web1 ~]# cat /usr/local/nginx/conf/nginx.conf
5. … …
6. location /status {
7. stub\_status on;
8. }
9. … …
10. [root@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

[root@web1 nginx-1.12.2]# ./configure \

> --with-http\_stub\_status\_module

[root@web1 nginx-1.12.2]# make && make install

[root@web1 ~]# cat /usr/local/nginx/conf/nginx.conf

… …

location /status {

stub\_status on;

}

… …

[root@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

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf.d/nginx.status
2. UserParameter=nginx.status[\*],/usr/local/bin/nginx\_status.sh $1
3. [root@web1 ~]# killall zabbix\_agentd
4. [root@web1 ~]# zabbix\_agentd

[root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf.d/nginx.status

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

[root@web1 ~]# killall zabbix\_agentd

[root@web1 ~]# zabbix\_agentd

自定义监控脚本（仅供参考，未检测完整状态）

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@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@web1 ~]# chmod +x /usr/local/bin/nginx\_status.sh

[root@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@web1 ~]# chmod +x /usr/local/bin/nginx\_status.sh

测试效果：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@web1 ~]# zabbix\_get -s 127.0.0.1 \
2. -k 'nginx.status[accepts]'

[root@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。

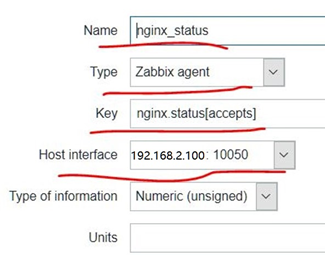


图-36

**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。

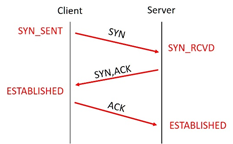


图-37

熟悉TCP连接的四次断开，参考图-38。

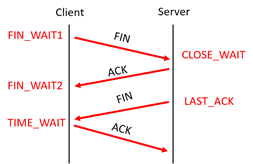


图-38

2）查看网络连接状态

模拟多人并发连接

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@web1 ~]# ab -c 1000 -n 100000 http://192.168.2.100/

[root@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

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@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

[root@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。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf.d/net.status
2. UserParameter=net.status[\*],/usr/local/bin/net\_status.sh $1
4. [root@web1 ~]# killall zabbix\_agentd
5. [root@web1 ~]# zabbix\_agentd

[root@web1 ~]# vim /usr/local/etc/zabbix\_agentd.conf.d/net.status

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

[root@web1 ~]# killall zabbix\_agentd

[root@web1 ~]# zabbix\_agentd

自定义监控脚本（仅供参考，未检测完整状态）

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@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@web1 ~]# chmod +x /usr/local/bin/net\_status.sh

[root@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@web1 ~]# chmod +x /usr/local/bin/net\_status.sh

测试效果：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY02/CASE/01/index.html)

1. [root@web1 ~]# zabbix\_get -s 127.0.0.1 \
2. -k 'net.status[time\_wait]'

[root@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