

NSD SECURITY DAY05

1. [案例1：常用系统监控命令](#)
2. [案例2：部署Zabbix监控平台](#)
3. [案例3：配置及使用Zabbix监控系统](#)
4. [案例4：自定义Zabbix监控项目](#)

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：

1. 查看内存信息
2. 查看交换分区信息
3. 查看磁盘信息
4. 查看CPU信息
5. 查看网卡信息
6. 查看端口信息
7. 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

```

01. [root@proxy ~]# free                                //查看内存信息
02.      total      used      free   shared  buff/cache   available
03. Mem:    16166888  8017696   720016    106504   7429176   7731740
04. Swap:    4194300   218268    3976032
05. [root@proxy ~]# free | awk '/Mem/{ print $4}'        //查看剩余内存容量
06. 720928
07. [root@proxy ~]# swapon -s                            //查看交换分区信息
08. 文件名          类型      大小    已用    权限
09. /dev/sda3         partition 4194300 218268 - 1

```

[Top](#)

步骤二：查看磁盘与CPU利用率

1) 查看磁盘信息

```

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

```

2) 查看CPU平均负载

```

01. [root@proxy ~] # uptime //查看CPU负载 (1, 5, 15分钟)
02. 23: 54: 12 up 38 days, 14: 54, 9 users, load average: 0.00, 0.04, 0.05
03. [root@proxy ~] # uptime | awk '{ print $NF}' //仅查看CPU的15分钟平均负载
04. 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

```

01. [root@proxy ~] # ifconfig eth0
02. eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
03.      inet 192.168.4.5 netmask 255.255.255.0 broadcast 172.25.0.255
04.      inet6 fe80::5054:ff:fe00:b prefixlen 64 scopeid 0x20<link>
05.      ether 52:54:00:00:00:0b txqueuelen 1000 (Ethernet)
06.      RX packets 62429 bytes 10612049 (10.1 MB)
07.      RX errors 0 dropped 0 overruns 0 frame 0
08.      TX packets 5674 bytes 4121143 (3.9 MB)
09.      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

```

2) 查看端口信息

[Top](#)

```

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

```



```

01. [ root@zabbixserver ~] # yum -y install gcc pcre-devel openssl-devel
02. [ root@zabbixserver ~] # tar -xvf nginx-1.12.2.tar.gz
03. [ root@zabbixserver ~] # cd nginx-1.12.2
04. [ root@zabbixserver nginx-1.12.2] # ./configure --with-http_ssl_module
05. [ root@zabbixserver nginx-1.12.2] # make && make install
06. [ root@zabbixserver ~] # yum -y install php php-mysql \
07. > mariadb mariadb-devel mariadb-server
08. [ root@zabbixserver ~] # yum -y install php-fpm-5.4.16-42.el7.x86_64.rpm
09. //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。

```

01. [ root@zabbixserver ~] # vim /usr/local/nginx/conf/nginx.conf
02. ... ..
03. http{
04. ... ..
05.     fastcgi_buffers 8 16k;           //缓存php生成的页面内容，8个16k
06.     fastcgi_buffer_size 32k;         //缓存php生产的头部信息
07.     fastcgi_connect_timeout 300;     //连接PHP的超时时间
08.     fastcgi_send_timeout 300;        //发送请求的超时时间
09.     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. ... ..

```

3) 启动服务

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

```

01. [ root@zabbixserver ~] # systemctl start mariadb
02. [ root@zabbixserver ~] # systemctl start php-fpm
03. [ root@zabbixserver ~] # ln -s /usr/local/nginx/sbin/nginx /sbin/nginx
04. [ root@zabbixserver ~] # nginx

```

[Top](#)

```

05.
06. [ root@zabbixserver ~] # firewall-cmd --set-default-zone=trusted
07. [ 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。

```

01. [ root@zabbixserver ~] # cat /usr/local/nginx/html/test.php
02. <?php
03. $i=33;
04. echo $i;
05. ?>
06. [ 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。

```

01. [ root@zabbixserver lnmp_soft] # yum -y install net-snmp-devel \
02. > curl-devel
03. //安装相关依赖包
04. [ root@zabbixserver lnmp_soft] # yum -y install \
05. > libevent-devel-2.0.21-4.el7.x86_64.rpm
06. //注意libevent-devel这个软件包在lnmp_soft目录下有提供
07. [ root@zabbixserver lnmp_soft] # tar -xzf zabbix-3.4.4.tar.gz
08. [ root@zabbixserver lnmp_soft] # cd zabbix-3.4.4/
09. [ 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
18. [ root@zabbixserver zabbix-3.4.4] # make && make install

```

[Top](#)

2) 初始化Zabbix

创建数据库，上线Zabbix的Web页面

```

01. [ root@zabbixserver ~] # mysql
02. mysql> create database zabbix character set utf8;
03. //创建数据库，支持中文字符集
04. mysql> grant all on zabbix.* to zabbix@'localhost' identified by 'zabbix';
05. //创建可以访问数据库的账户与密码
06. [ root@zabbixserver ~] # cd /tmp/soft/zabbix-3.4.4/database/mysql/
07. [ root@zabbixserver mysql] # mysql - uzabbix - pzabbix zabbix < schema.sql
08. [ root@zabbixserver mysql] # mysql - uzabbix - pzabbix zabbix < images.sql
09. [ 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）

```

上线Zabbix的Web页面

```

01. [ root@zabbixserver ~] # cd /tmp/soft/zabbix-3.4.4/frontend/php/
02. [ root@zabbixserver php] # cp -r * /usr/local/nginx/html/
03. [ root@zabbixserver php] # chmod -R 777 /usr/local/nginx/html/*

```

修改Zabbix_server配置文件，设置数据库相关参数，启动Zabbix_server服务

```

01. [ root@zabbixserver ~] # vim /usr/local/etc/zabbix_server.conf
02. DBHost=localhost
03. //数据库主机，默认该行被注释
04. DBName=zabbix
05. //设置数据库名称
06. DBUser=zabbix
07. //设置数据库账户
08. DBPassword=zabbix
09. //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
15.

```

[Top](#)

```

16. [ root@zabbixserver ~] # ss - ntulp | grep zabbix_server //确认连接状态，端口10051
17. tcp LISTEN 0 128 *:10051 *: * users: ( ( "zabbix_server",pid=23275,fd=4), ( "zabbix_server",

```

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。

修改Zabbix_agent配置文件，启动Zabbix_agent服务

```

01. [ root@zabbixserver ~] # vim /usr/local/etc/zabbix_agentd.conf
02. Server=127.0.0.1,192.168.2.5 //允许哪些主机监控本机
03. 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
04. Hostname=zabbix_server //设置本机主机名
05. LogFile=/tmp/zabbix_server.log //设置日志文件
06. UnsafeUserParameters=1 //是否允许自定义key
07. [ root@zabbixserver ~] # zabbix_agentd //启动监控agent
08.
09. [ root@zabbixserver ~] # ss - ntulp | grep zabbix_agentd //查看端口信息为10050
10. tcp LISTEN 0 128 *:10050 *: * users: ( ( "zabbix_agentd",

```

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。

浏览器访问Zabbix_server服务器的Web页面

```

01. [ root@zabbixserver ~] # firefox http://192.168.2.5/index.php
02. //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
03. //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
php-bcmath和php-mbstring都在lnmp_soft目录下有提供软件包。

```

01. [ root@zabbixserver ~] # yum - y install php- gd php- xml
02. [ root@zabbixserver ~] # yum install php- bcmath- 5.4.16- 42.el7.x86_64.rpm
03. [ root@zabbixserver ~] # yum install php- mbstring- 5.4.16- 42.el7.x86_64.rpm
04. [ root@zabbixserver ~] # vim /etc/php.ini
05. date.timezone = Asia/Shanghai //设置时区
06. max_execution_time = 300 //最大执行时间，秒

```

[Top](#)

07. `post_max_size = 32M` //POST数据最大容量
08. `max_input_time = 300` //服务器接收数据的时间限制
09. `memory_limit = 128M` //内存容量限制
10. `[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

[Top](#)

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

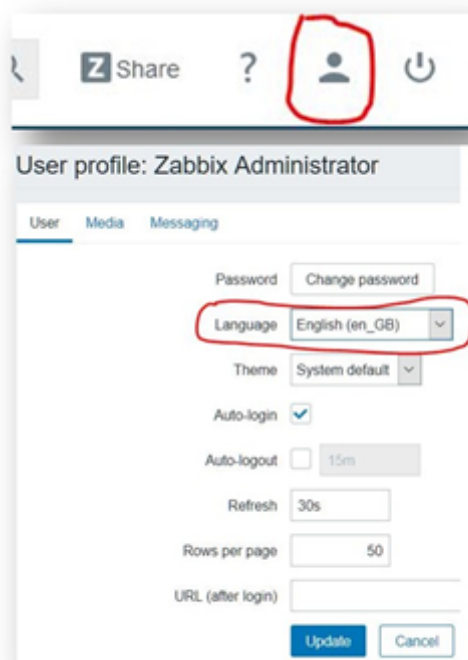


图-5

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zabbixclient_web1为例）。

01. [root@zabbixclient_web1 ~] # useradd -s /sbin/nologin zabbix
02. [root@zabbixclient_web1 ~] # yum -y install gcc pcre-devel
03. [root@zabbixclient_web1 ~] # tar -xf zabbix-3.4.4.tar.gz
04. [root@zabbixclient_web1 ~] # cd zabbix-3.4.4/
05. [root@zabbixclient_web1 zabbix-3.4.4] # ./configure --enable-agent

[Top](#)

```
06. [ root@zabbixclient_web1 zabbix- 3.4.4 ] # make && make install
```

2) 修改agent配置文件，启动Agent

```
01. [ root@zabbixclient_web1 ~ ] # vim /usr/local/etc/zabbix_agentd.conf
02. Server=127.0.0.1,192.168.2.5           //谁可以监控本机（被动监控模式）
03. ServerActive=127.0.0.1,192.168.2.5     //谁可以监控本机（主动监控模式）
04. Hostname=zabbixclient_web1             //被监控端自己的主机名
05. EnableRemoteCommands=1
06. //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
07. UnsafeUserParameters=1                //是否允许自定义key 监控
08. [ root@zabbixclient_web1 ~ ] # zabbix_agentd           //启动agent服务
```

3) 拷贝启动脚本（非必须操作，可选做），有启动脚本可以方便管理服务，启动与关闭服务。启动脚本位于zabbix源码目录下。

```
01. [ root@zabbixclient_web1 zabbix- 3.4.4 ] # cd misc/init.d/fedora/core
02. [ root@zabbixclient_web1 zabbix- 3.4.4 ] # cp zabbix_agentd /etc/init.d/
03. [ root@zabbixclient_web1 zabbix- 3.4.4 ] # /etc/init.d/zabbix_agentd start
04. [ root@zabbixclient_web1 zabbix- 3.4.4 ] # /etc/init.d/zabbix_agentd stop
05. [ root@zabbixclient_web1 zabbix- 3.4.4 ] # /etc/init.d/zabbix_agentd status
06. [ root@zabbixclient_web1 zabbix- 3.4.4 ] # /etc/init.d/zabbix_agentd restart
```

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zabbixclient_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

[Top](#)

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。

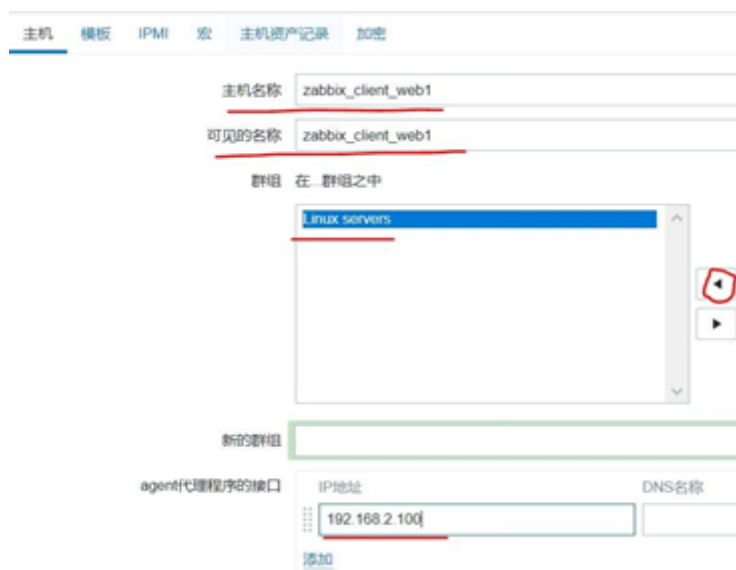


图-8

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

[Top](#)

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。

[Top](#)

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。

```
01. [ root@zabbixclient_web1 ~] # vim /usr/local/etc/zabbix_agentd.conf
02. Include=/usr/local/etc/zabbix_agentd.conf.d/           //加载配置文件目录
03. [ root@zabbixclient_web1 ~] # cd /usr/local/etc/zabbix_agentd.conf.d/
04. [ root@zabbixclient_web1 zabbix_agentd.conf.d] # vim count.line.passwd
05. UserParameter=count.line.passwd,wc -l /etc/passwd | awk ' { print $1} '
06.      //自定义key语法格式:
07.      //UserParameter=自定义key名称,命令
```

2) 测试自定义key是否正常工作

```
01. [ root@zabbixclient_web1 ~] # killall zabbix_agentd
02. [ root@zabbixclient_web1 ~] # zabbix_agentd           //重启agent服务
03. [ root@zabbixclient_web1 ~] # zabbix_get -s 127.0.0.1 -k count.line.passwd
04. 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：

```
01. [ root@zabbixclient_web1 ~] # vim /usr/local/etc/zabbix_agentd.conf
02. Server=127.0.0.1,192.168.2.5
03. 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

[Top](#)

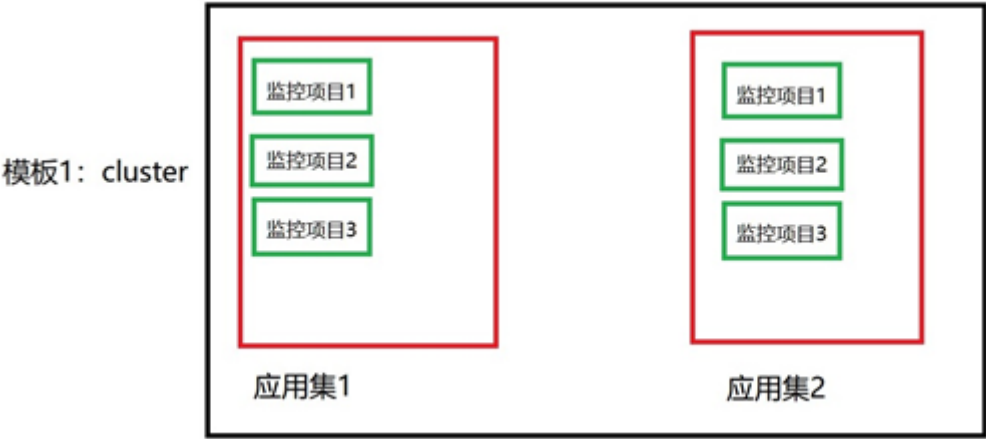


图-12

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。

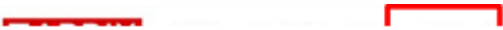


图-13

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。

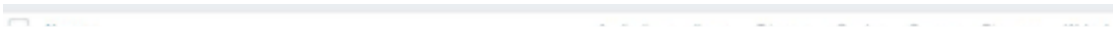


图-14

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。

Templates	Applications	Items	Triggers	Graphs	Screens	Di
count_line_passwd	Applications (0)	Items (0)	Triggers (0)	Graphs (0)	Screens (0)	Di

[Top](#)

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。



图-16

设置应用名称如图-17所示。

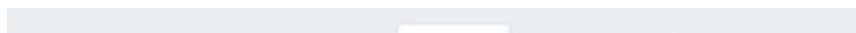


图-17

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。

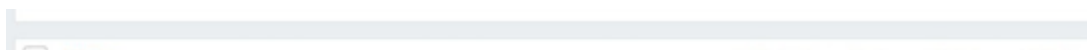


图-18

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-25

[Top](#)