深入浅出Zabbix 3.0 -- 第三章 Zabbix 监控方式-大白小白一起学-51CTO博客

第三章 Zabbix 监控方式

有人说通过Zabbix可以完成任何监控任务，只有你想不到的，但没有监控不了的，真是这样吗？当你通过本章了解了Zabbix提供的多种监控方式后，你就会发现此言非虚。

这里所说的监控方式，实际上就是Zabbix中的配置监控项时选择的监控项类型。为了适应各种应用场景的需要，Zabbix提供了多种方法帮助你更好的完成监控任务。

Zabbix 3.0中提供的监控方式包括：

- Active agents
- Passive agents
- Extending agents
- Simple checks
- SNMP agents
- Zabbix Internal checks
- Zabbix trapper
- IPMI agents
- JMX agents
- External checks
- Database monitoring
- SSH agents
- Telnet agents
- SNMP Traps
- Aggregate checks
- Calculated checks

3.1 Active agents

使用Zabbix agent创建监控项时有两种方式，即Active（主动式）agent和Passive （被动式）agent。

在Active agent模式下，Zabbix agent启动后，由agent端初始化和Zabbix server之间的通信，向Zabbix server发出获取监控项清单的请求，server端收到请求后响应agent发出的请求，并将监控项清单发送给agent。agent端定期和Zabbix server通信，保证获得最新的监控项清单。agent则根据监控项清单查询监控项的数据并将结果发送给Zabbixserver。流程如下图3-1所示。

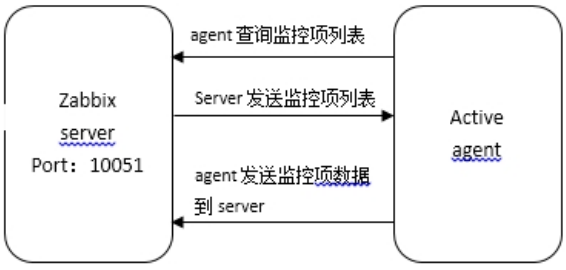

图 3-1

为了启用Active agent模式，需要在zabbix_agentd.conf文件中配置ServerActive参数，告诉agent可以联系到哪些服务器（默认端口是10051）。通过配置RefreshActiveChecks参数，可以设置agent端多长时间向server询问一次监控项清单，默认是120秒。在默认设置下server端改变active agent监控项有关的一些设置后，server端需要1分钟刷新配置缓存（通过server配置文件中的参数CacheUpdateFrequency设置，默认是60秒），agent需要等待2分钟才能够知道监控项的变化。如果从server查询监控项清单失败（网络问题或其他原因），agent端会等待1分钟后重新向server发出查询请求。Active agent也有自己的缓存，可以通过BufferSend或BufferSize进行设置， BufferSend参数设置监控项数据在缓存中保留的时间，默认是5秒（可以设置到3600）。BufferSize参数设置保留监控项数据的缓存大小，默认是100（可以设置到65535）

配置Active agent监控项的步骤：

1、Zabbix agent安装完成后，打开配置文件zabbix_agentd.conf。

2、设置ServerActive参数，格式为IP:port 或DNS主机名:port。在这里我们可以设置多个server或proxy的DNS主机名或IP地址，用逗号分隔。

3、设置Hostname参数，这个名字必须是唯一的并和Zabbixserver中Configuration -->Hosts页面中添加的主机名称相同。

4、验证Zabbix server的10051端口能够访问。

5、重启zabbix_agent（systemctlrestart zabbix-agent.service）。

6、检查agent日志（tail -f/var/log/zabbix/zabbix_agentd.log）。

7、在主机中添加主动式监控项（Configuration --> Hosts --> Items --> Create item）。选择监控项的Type（类型）为Zabbix agent（active）。

3.2 Passive agents

Passive agent为我们提供了一种简单易行的方法，Zabbixserver或proxy根据监控项中配置的Update interval（数据更新间隔），定期向agent端发出查询请求，如CPU负载、磁盘使用空间等等。agent根据请求收集监控项数据并返回给server或proxy。整个过程就是简单的一问一答，你要什么值我给你什么值，从agent角度来看是被动的回答。如下图3-2所示。

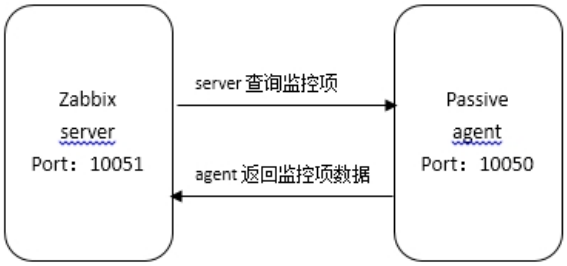

图 3-2

配置Passive agent监控项的步骤：

1、安装Zabbix agent，打开配置文件zabbix_agentd.conf。

2、设置Server参数，格式为IP 或DNS主机名。在这里我们可以设置多个server或proxy的DNS主机名或IP地址，用逗号分隔。

3、注释掉ServerActive和Hostname这两个参数，在Passive agent模式中不需要这两个参数，如果你想同时使用active agent，这两个参数必须配置。

4、验证agent端的10050端口能够访问。

5、重启zabbix_agent（systemctlrestart zabbix-agent.service）。

6、检查agent日志（tail -f/var/log/zabbix/zabbix_agentd.log）。

7、在主机中添加被动式监控项（Configuration --> Hosts --> Items --> Create item）。选择监控项的Type（类型）为Zabbix agent。

3.3 Extending agents

Zabbix中提供了一些标准的监控项可以使用Key，当你添加Zabbix agent监控项时可以选择使用，但在实际环境中这些标准的Key并不能满足特定的监控需求，在Zabbix中可以使用多种方法进行扩展，其中一个方法就是在agent配置文件中使用UserParameter进行扩展。

通过UserParameter参数扩展监控项的key，可以灵活的实现多种监控需求。Zabbix中定义UserParameter的格式为UserParameter=,。

配置UserParameter的步骤：

1、打开agent配置文件zabbix_agentd.conf。

2、设置UserParameter参数。例如：UserParameter=mysql.threads,mysqladmin -u root –p status|cut -f3 -d":"|cut -f1-d"Q" ，该参数返回MySQL线程的数量给自定义的key：mysql.threads。

3、保存配置文件，重新启动Zabbix agent服务。

4、Web前端Configuration --> Hosts--> Items页面中添加监控项。

uName字段中设置监控项名称，例如 MySQL Threads。

uType字段中选择Zabbix agent或者Zabbix agent（active）。

uKey字段中填写mysql.threads，这里填写的内容必须和UserParameter中定义的一样。

uType of Information字段中选择Numeric（unsigned）。
uData type中选择Decimal。
u其他配置参数保持不变。点击Add按钮保存。
5、Monitoring --> Latest data页面查看监控项MySQL Threads。

3.4 Simple checks

Zabbix 中simple checks是基于ICMP ping或者端口检测来确定主机是否在线或服务端口能否正常连接。这种方式下主机中不需要安装Zabbix agent，当我们检测主机或端口的可用性时simplechecks返回的值为1或者0，当我们检测性能时返回的是浮点数（如检测ping的响应时间时返回值0.02秒），如果检测失败则返回0。
为了降低网络流量，更高效的进行ICMP检测，Zabbix执行icmppingsec、icmpping 和 icmppingloss检测时使用了一个第三方的工具fping和fping6，依赖linux不同的发行版安装的版本各有不同，建议使用fping 3.0以上的版本。在CentOS系统中需要安装fping时可以通过命令yum install fping完成安装。
Zabbix中默认定义了3个用于ICMP检测的监控项和2个用于TCP/UDP连接检测的监控项，分别是：

- Icmpping：主机响应ICMP ping返回1，否则返回0。
- Icmppingloss：返回丢失ICMP ping数据包的百分比。
- Icmppingsec：返回ICMP ping的响应时间，单位是秒。如果主机没有响应（timeout reached）则返回0。如果返回值小于0.0001秒时返回值将被设置为0.0001。
- Net.tcp.service / net.udp.service：主机上指定的服务正常运行并能建立TCP / UDP连接时返回1，否则返回0。
- Net.tcp.service.perf / net.udp.service.perf：返回连接到指定TCP / UDP端口的服务所使用的时间，单位是秒。如果服务没有运行则返回0.000000。

net.tcp.service 和net.tcp.service.perf支持我们知道的大部分协议，如SSH、FTP、HTTP等。这两个项目是非常有用的，我们可以对特定的IP和端口进行简单的TCP握手连接完成可用性的检测，同时对主机或应用的性能不会有任何影响。
下面一起来看看这几个项目的用法。

- Icmpping

格式：Icmpping[,,,,]。其中target是主机IP或DNS主机名；packets是数据包的数量；interval是数据包之间的间隔时间，单位是微秒；size是数据包的大小，单位是bytes；timeout是超时时间，单位是微秒。例如icmpping[,20,50,256,100]，如果你不想定义target、packet等值，可以不填写任何参数，可以写成icmpping[,4]，只要4个数据包有1个响应，监控项就会返回1。

- Icmppingloss

格式：icmppingloss[,,,,]。其中target是主机IP或DNS主机名；packets是数据包的数量；interval是数据包之间的间隔时间，单位是微秒；size是数据包的大小，单位是bytes；timeout是超时时间，单位是微秒。

- Icmppingsec

格式：icmppingsec[,,,,,]。其中target是主机IP或DNS主机名；packets是数据包的数量；interval是数据包之间的间隔时间，单位是微秒；size是数据包的大小，单位是bytes；timeout是超时时间，单位是微秒。

- net.tcp.servic

格式：net.tcp.service[service,,]。其中service是TCP协议的名称，如：ssh、ldap、smtp、ftp、http、pop、nntp、imap、Telnet等等；IP是IP地址或DNS主机名（默认使用定义监控项的主机的IP或DNS主机名）；port是端口号（默认使用服务标准的端口号）。例如：net.tcp.service[ftp,45]。当前不支持加密协议的检测，像IMAP的993端口或POP的995端口，但我们可以用net.tcp.service[tcp,,port]来完成对它们的检测。Zabbix 从v2.0起支持https和telnet。

- net.tcp.service.perf

格式：net.tcp.service.perf[service,,]。其中service是TCP协议的名称，
如：ssh、ldap、smtp、ftp、http、pop、nntp、imap、Telnet等等；IP是主机的IP地址或DNS主机名（默认使用定义该监控项主机的IP或DNS主机名）；port是端口号（默认使用服务标准的端口号）。例如：net.tcp.service.perf [ssh]。当前不支持加密协议的检测，像IMAP在993端口或POP在995端口上，但我们可以用net.tcp.service.perf [tcp,,port]来完成对它们的检测。

- net.udp.service

格式：net.udp.service[service,,]。其中service是UDP协议的名称，如：ntp；IP是主机的IP地址或DNS主机名（默认使用定义该监控项主机的IP或DNS主机名）；port是端口号（默认使用服务标准的端口号）。例如：net.udp.service[ntp,45]，在UDP端口45上检测NTP服务的可用性。

- net.udp.service.perf

格式：net.udp.service.perf[service,,]。其中service是UDP协议的名称，如：ntp；IP是主机的IP地址或DNS主机名（默认使用定义该监控项主机的IP或DNS主机名）；port是端口号（默认使用服务标准的端口号）。例如：net.udp.service.perf [ntp]，可以检测NTP服务的响应时间。

Simple checks是一种简单而高效的监控方式，由于其不需要传输复杂的监控数据，因此在监控成百上千的主机和服务的可用性时，对整体的网络流量产生的影响是最小的。

配置Simple checks的步骤：

1、创建一个新的监控项（Configuration --> Template --> Items--> Create item 或Configuration --> Host --> Items --> Create item）。

2、在监控项配置页面中：

- 填写Name，例如：Check SSH port $3。（$3是key中的第三个参数{$SSH_PORT}）。
- 选择Type为Simple check。
- 填写Key，例如：net.tcp.service[ssh,,{$SSH_PORT}]，{$SSH_PORT}是定义的macro。
- Type of information选择Numeric。
- Data type选择Decimal。
- 如果需要可以在NewApplication中填写一个监控项组的名称，如：ssh check。
- 其他配置参数可以保持不变，点击Add按钮保存。

3、Monitoring --> Latest data页面查看监控项。

4.5 SNMP agents

监控交换机、路由器、UPS等设备时，你是没有办法通过Zabbixagent进行监控的，原因是这些设备中没有办法安装agent程序，但这并不代表Zabbix不能对这些设备进行监控，利用标准的SNMP协议，可以轻松实现Zabbix对这些设备的监控。

SNMP（简单网络管理协议）是TCP/IP协议簇的一个应用层协议，是一种广泛用于监测网络设备（例如：交换机、路由器、UPS等）的网络协议。在每个被监控的设备中都会运行设备自带的SNMP agent， Zabbix使用SNMP协议向被监控设备的SNMP agent发出查询指令，并由SNMP agent返回查询的值。

在设置SNMP agent监控项之前，我们先要确定OID（SNMP对象标识符）。SNMP将被管理对象用一个树来组织，被管理对象用OID表示，如：1.3.6.1.2.1.1.3 代表sysUpTime。实际环境中会使用很多厂商的产品，每个产品中定义的OID不尽相同，所以准备使用SNMP agent 监控设备前，需要厂商提供设备的MIB文件。MIB文件是基于SMI语法定义的说明某个OID在OID树中的位置、数据类型、描述等信息的文本文件，如果没有MIB文件，你很难理解一串数字代表的含义是什么。

配置SNMP agent的步骤：

1、确定已经安装了net-snmp-utils，如果没有安装，可通过以下命令安装：

 # yum install net-snmp-utils

如果你是编译源码安装的Zabbix server，一定要使用 --with-net-snmp选项。

2、Zabbix server中添加新的主机并填写SNMPInterface 信息，如下图3-3所示。

图 3-3

3、用snmpwalk检查能否从被监控设备获取到OID的值。

 # snmpwalk -v 2c -cpublic 192.168.10.1 | more

其中 -v 2c 是SNMP协议的版本，Zabbix支持SNMP v1、2c、和3。-c public是community字符串。执行上面的命令有结果返回，说明被监控主机可以连接，可以使用SNMP进行监控。我们可以进一步确定OID的值。

 # snmpget -v 2c -cpublic 192.168.10.1 -On IF-MIB::ifInOctets.1

返回信息：.1.3.6.1.2.1.2.2.1.10.1 = Counter32:1494804。

如果你想获取完整的OID字符串，可以执行下面的命令。

 # snmpget -v 2c -cpublic 192.168.10.1 -Of IF-MIB::ifInOctets.1

返回信息：.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifInOctets.1= Counter32: 1566936。

4、现在创建新的监控项：

- Name中输入ifInOctets $1（$1是Key中第一个参数 Port1）。
- Type选择正确的SNMP版本，例如SNMPv2 agent。
- Key中输入任何你想使用的名称，如：ifInOctets[Port1]。
- Host Interface使用主机中定义的SNMP接口。
- SNMP OID中输入第3步中用snmpget命令返回的OID。这里无论使用哪种格式的OID都是可以的。
- SNMP community中填写community字符串。
- Type of information中选择 Numeric（float）。
- Units中填写Bytes。
- Store value中选择Delta（speed per seconds）。
- 其他参数可以保持不变，单击Add按钮保存。

如下图3-4所示。

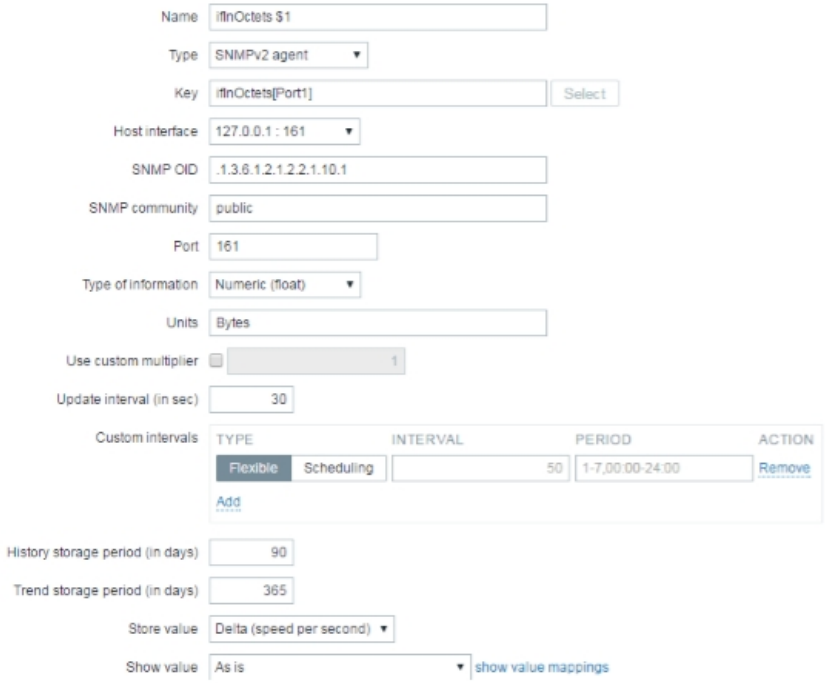

图 3-4

5、Monitoring --> Latest data页面查看监控项。

3.6 Zabbix Internal checks

Zabbix Internal主要是用来监测Zabbixserver 或 proxy server自身的性能。Zabbix Internal由Zabbix server 或proxy server进行计算，从Zabbix 2.4版本开始，即使在主机维护状态下也会处理Zabbix Internal。Zabbix server处理Zabbix Internal时不依赖agent，你只需要拥有超级管理员的权限即可。

Zabbix在系统中已经预设了针对Zabbix server和 proxy server的模板，模板的名称是Template App Zabbix Server 和 Template App Zabbix Proxy。

配置Zabbix Internal的步骤：

1、创建一个新的监控项（Configuration --> Template --> Items --> Create item 或Configuration --> Host --> Items --> Create item）。

- Name中填写监控项的名称。
- Type中选择Zabbix internal。
- Key中选择（单击右侧的Select按钮）zabbix[process,,,]，在这里我们使用zabbix[process,poller,avg,busy]。

- Type of information选择Numeric (float)。
- Units填写%。
- 其他参数可以保持原状，单击Add按钮保持。

如下图3-5所示。

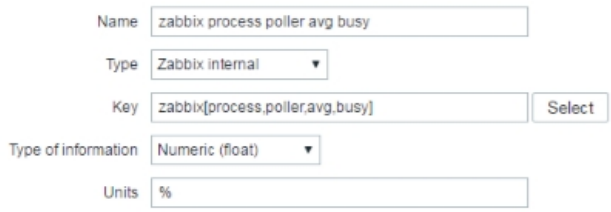

图 3-5

2、Monitoring --> Latest data页面查看监控项。

3.7 Zabbix trapper

Zabbix使用agent、IPMI或SNMP的方式收集监控数据时，有时候会因为监控项Key中使用的脚本执行时间过长而超时，从而无法获取数据。因此Zabbix 提供了trapper的监控方式，利用zabbix-sender工具可以主动将数据从被监控主机发送到Zabbix server，这种方式中不需要在被监控主机中安装Zabbix agent。

配置Zabbix trapper的步骤：

1、首先我们要在被监控主机中安装zabbix-sender程序，安装命令如下：

 # yum installzabbix-sender

2、创建监控项（Configuration --> Template --> Items --> Create item或Configuration --> Host --> Items --> Create item）。

- Name中填写监控项名称。
- Type中选择Zabbix trapper。
- Key中填写你想使用的key名称（例如：trapper.key）。
- Type of information和Data type中选择发送到Zabbixserver的数据类型。
- Allowed hosts中可以填写被监控主机的IP地址或DNS主机名，如果填写，trapper只接收来自这些主机的数据。这里可以设置一个IP地址或DNS主机名，例如：192.168.10.22或testtrapper.Zabbix.com。也可以设置多个IP地址或DNS主机名（使用逗号分隔），例如：192.168.10.22,192.168.10.23,192.168.10.24或testtrapper1.Zabbix.com, testtrapper2.Zabbix.com, testtrapper3.Zabbix.com。
- 其他参数可以保持不变，单击Add按钮保存。

如下图3-6所示。

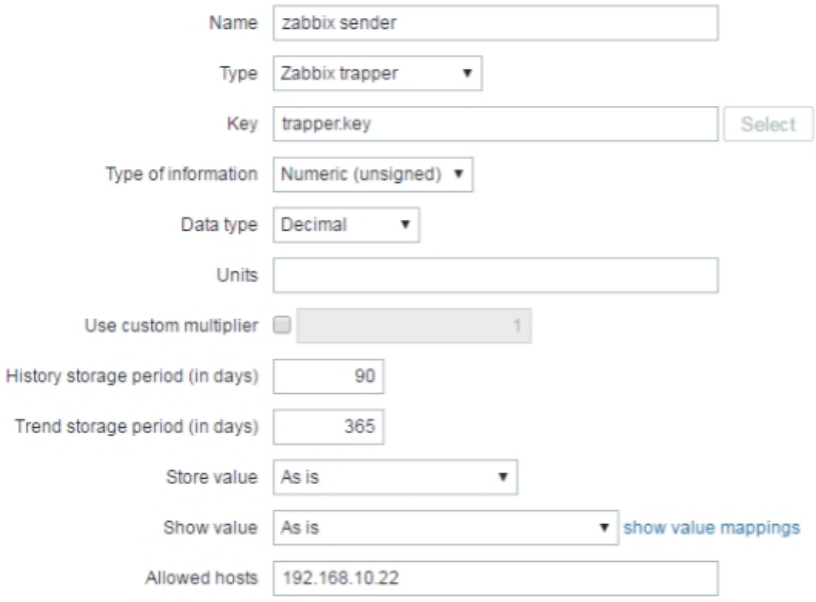

图 3-6

3、在被监控主机中运行 zabbix_sender -z -s -k -o。例如：

 # zabbix_sender -z 192.168.10.102 –s "trapper host" -k trapper.key -o 20

4、Monitoring --> Latest data页面查看监控项。

3.8 IPMI agents

IPMI（Intelligent PlatformManagement Interface）是一个开放标准的硬件管理接口规范，定义了嵌入式管理子系统进行通信的特定方法。现在主流的服务器使用远程控制卡（例如Dell的DRAC、HP的ILO等）都可以进行远程控制管理，通过IPMI你可以远程开机、关机、重启，远程查看服务器当前的运行状态，可以安装操作系统，实现带外管理。Zabbix server通过IPMI可以直接监控服务器硬件，即使是服务器的电源处在关闭的状态下也是没有问题的。

如果你是编译安装Zabbix，并且需要在Zabbix中使用IPMI agent，那么在编译时需要带上 --with-openipmi参数。

在使用IPMI agent之前，我们需要在主机中安装IPMI相关工具。

 # yum install ipmitoolOpenIPMI OpenIPMI-libs

安装完成后，我们可以检测下温度：

 # ipmitool sdr list | grepTemp

Ambient Temp | 23degrees C | ok

CPU 1 Temp | 45degrees C | ok

CPU 2 Temp |disabled | ns

CPU 3 Temp |disabled | ns

CPU 4 Temp |disabled | ns

进一步可以看看CPU 1 Temp的详细情况：

 # ipmitool event “CPU 1Temp” list

Finding sensor CPU 1 Temp... ok

Sensor States:

lnr : Lower Non-Recoverable

lcr : Lower Critical

lnc : Lower Non-Critical

unc : Upper Non-Critical

ucr : Upper Critical

unr : Upper Non-Recoverable

接下来我们配置IPMI的管理账号，这一步可以通过服务器的远程控制卡管理界面完成，也可以通过OpenIPMI工具完成。

重新设置管理员密码，2为 root用户的ID。

 # ipmitool user setpassword 2

设置Zabbix用户账号，例子中用户ID是3，在设置前要先确认ID 3 没有其他用户使用。

 # ipmitool user set name 3 zabbix

设置Zabbix用户密码。

 # ipmitool user setpassword 3

Password for user 3:

Password for user 3:

为Zabbix用户授权。

 # ipmitool channelsetaccess 1 3 link=on ipmi=on callin=on privilege=2

启用Zabbix用户。

 # ipmitool user enable 3

验证设置是否正确。

 # ipmitool channelgetaccess 1 3

Maximum User IDs : 15

Enabled User IDs :2

User ID :3

User Name :zabbix

Fixed Name :No

Access Available :call-in / callback

Link Authentication :enabled

IPMI Messaging :enabled

Privilege Level :USER

看到上面的信息时，说明Zabbix用户已经创建，Privilege Level是USER。但这个账号还不能通过网络访问，我们需要授权访问。

 # ipmitool lan set 1 authUSER MD5

验证设置是否正确。

 # ipmitool lan print 1

Set in Progress : Set Complete

Auth Type Support :NONE MD5 PASSWORD

Auth Type Enable : Callback :

: User : MD5

: Operator :

: Admin : MD5

: OEM :

现在我们使用Zabbix用户远程查询服务器的状态。

 # ipmitool –U Zabbix –H192.168.10.22 -I lanplus sdr list | grep Temp

Ambient Temp | 23degrees C | ok

CPU 1 Temp | 45degrees C | ok

CPU 2 Temp |disabled | ns

CPU 3 Temp |disabled | ns

CPU 4 Temp |disabled | ns

修改zabbix_server.conf配置文件。

 # vi zabbix_server.conf

StartIPMIPollers=5

重启Zabbix server服务。

 # systemctl restartzabbix-server

配置IPMI checks的步骤：

1、在Configuration --> Host页面中，单击Name列中的主机名称，例如Zabbix server，在主机配置页面Host标签下添加IPMI接口配置，如下图3-7所示。

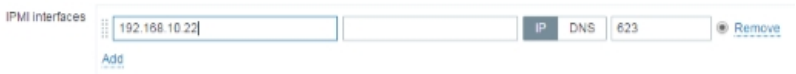

图 3-7

2、在IPMI标签下配置身份验证算法、权限级别、用户名及密码。如下图3-8所示。

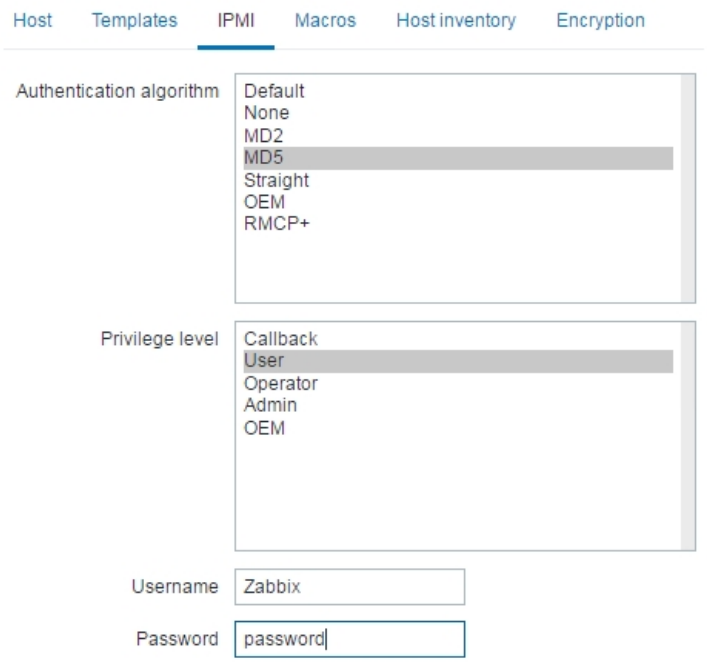

图 3-8

3、单击Update按钮保存主机的配置。

4、创建一个新的监控项。

- Name中填写监控项名称。
- Type中选择IPMI agent。
- Key中填写key名称，例如：CPU 1 Temp。
- Host Interface中选择IPMI接口。
- IPMI Sensor中填写CPU 1 Temp。
- Type of information选择Numeric（float）
- Units中填写°C。
- 其他参数可以保持不变，单击Add按钮保存。

如下图3-9所示。

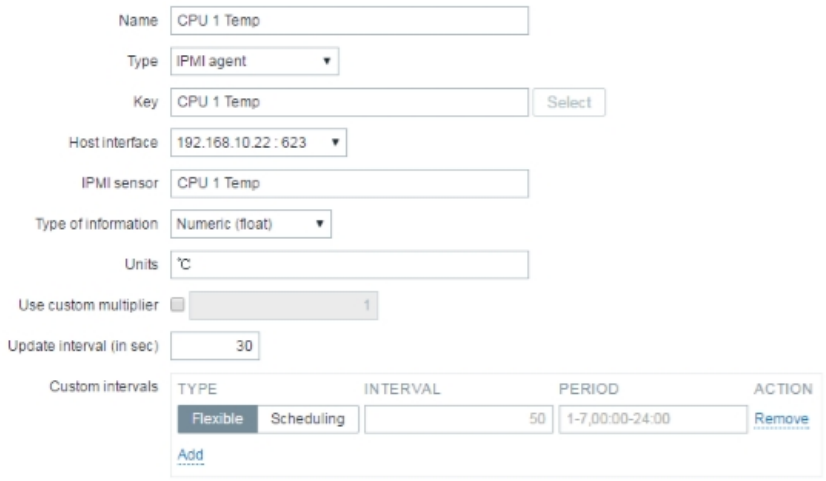

图 3-9

5、Monitoring --> Latest data页面查看监控项。

3.9 JMX agents

Zabbix通过JMX（Java Management Extensions）可以对Java Application进行监控，Zabbix利用原生的Zabbix Java gateway，一个Java守护进程监控JMX应用。当Zabbix想要知道某个JMX counter当前的数据时，它只去询问ZabbixJava gateway，而gateway会去查询需要的数据，所有这些查询都是通过JMX管理API完成的。

使用时，一个Java应用不需要额外安装任何其他的软件，也不需要实现或扩展新的代码来处理Zabbix的查询，仅仅需要在Java 应用的配置文件中设置一些参数，支持远程JMX的监控。

这些参数主要有：

-Dcom.sun.management.jmxremote

-Dcom.sun.management.jmxremote.port=<你要设置的端口号>

-Dcom.sun.management.jmxremote.authenticate=false

-Dcom.sun.management.jmxremote.ssl=false

例如：

```
java \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-jar/usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

上面的配置定义了Java程序使用12345端口监听来自本地的JMX的连接，并不需要身份验证和加密。如果你想从其他主机访问，需要配置 -Djava.rmi.server.hostname 参数。

在实际环境中从安全的角度考虑，需要设置身份验证和加密。具体设置步骤如下：

1、启用身份验证并指定一个包含密码的文件。

-Dcom.sun.management.jmxremote.authenticate=true

-Dcom.sun.management.jmxremote.password.file=/etc/java-6-penjdk/management/jmxremote.password

jmxremote.password文件内容：

monitorRole

controlRole

2、指定用户的配置文件。

-Dcom.sun.management.jmxremote.access.file=/etc/java-6-penjdk/management/jmxremote.access

jmxremote.access文件内容：

monitorRole readonly

controlRole readwrite

3、启用SSL。

-Dcom.sun.management.jmxremote.ssl=true

绑定下面的参数：

-Djavax.net.ssl.keyStore=<你的keyStore >

-Djavax.net.ssl.keyStorePassword=<你的 keyStorePassword >

-Djavax.net.ssl.trustStore=<你的trustStore >

-Djavax.net.ssl.trustStorePassword=<你的trustStorePassword >

-Dcom.sun.management.jmxremote.ssl.need.client.auth=true

完整的例子如下：

java \

-Djava.rmi.server.hostname=192.168.3.14 \

-Dcom.sun.management.jmxremote \

-Dcom.sun.management.jmxremote.port=12345 \

-Dcom.sun.management.jmxremote.authenticate=true \

-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password\

-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access\

-Dcom.sun.management.jmxremote.ssl=true \

-Djavax.net.ssl.keyStore=<你的KeyStore> \

-Djavax.net.ssl.keyStorePassword=<你的KeyStorePassword> \

-Djavax.net.ssl.trustStore=<你的trustStore > \

-Djavax.net.ssl.trustStorePassword=<你的trustStorePassword > \

-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \

-jar/usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar

-D参数需要写到你的应用或应用服务器的启动文件中，在完成上述的各项配置后，你的启动文件中将包含一些敏感的内容（keyStore和trustStore密码），因此需要保护好你的启动文件。

为了在Zabbix server使用JMX agent，还需要在Zabbix server中安装Java gateway。

 # yum installzabbix-java-gateway

 # systemctl enablezabbix-java-gateway

 # systemctl startzabbix-java-gateway

如果你是编译源码安装的Zabbix server，一定要使用 --enable-java选项。

在zabbix_server.conf文件中进行配置。

 # vi zabbix_server.conf

Java gateway = 127.0.0.1

Java Gateway Port = 10052

Start Java pollers = 5

不要忘记修改zabbix_server.conf文件后要重启Zabbixserver服务。

 # systemctl restartzabbix-server

下面我们就可以配置一个JMX agent监控项，步骤如下：

1、在Configuration --> Host页面中，单击Name列中的主机名称，例如Zabbix server，在主机配置页面Host标签下添加JMX接口配置，如下图3-10所示。

图 3-10

2、单击Update按钮保存主机的配置。

3、创建一个新的监控项。

- Name中填写监控项名称。
- Type中选择JMX agent。
- Key中填写你要监控的项目，格式为jmx[

jmx["java.lang:type=Memory","HeapMemoryUsage.used"]

- Host interface中选择JMX接口。
- 安装JMX控制台后为了安全，通常我们会设置一个登录控制台的用户名和密码，Zabbix中支持这种方式，在User name和Password中填写就可以了。
- Type of information选择Numeric（unsigned）。
- Data type中选择Decimal。
- Unit中设置单位，如：B。
- 其他参数可以保持不变，单击Add按钮保存。

如下图3-11所示。

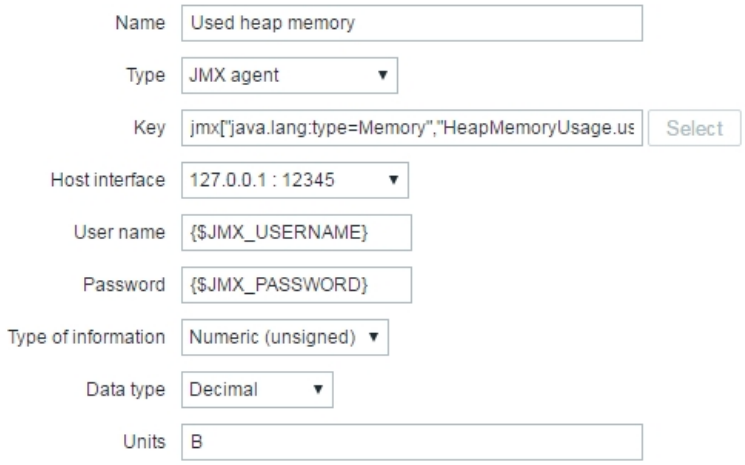

图 3-11

4、Monitoring --> Latest data页面查看监控项。

如果你想监控返回值为true或false的Boolean类型的监控项，你可以指定Typeof information为Numeric（unsigned），Data type为Boolean，Zabbix server将保存这个Boolean的值为1或0。

在Zabbix server中只能安装一个Javagateway，如果你需要的话可以在每个proxyserver中安装一个Java gateway。

如果遇到问题可以检查JMX 的日志文件：/var/log/zabbix_java_gateway.log。

在设置JMX agent的Key时，需要注意下面的一些情况：

- 属性名称中包含有点分隔符（.），比如说all.pen，你需要用 \ 转义，例如：jmx[com.example:Type=Test,all\.pen.color]
- 属性名称中包含有反斜杠（\），你需要使用两个 \\，例如：jmx[com.example:Type=Test,c:\\utility]
- 对象名称或属性名称中包含有空格或逗号，它需要用两个双引号（”）括起来，例如：jmx[com.example:type=Hello,"c:\\documentsand settings"]

通过External checks监控方式可以在Zabbixserver上执行脚本或二进制程序收集监控数据，这种方式不需要在被监控设备中安装agent。

使用这种方式之前，需要在zabbix-server.conf配置文件中定义脚本或程序的路径，设置正确的执行权限，保证Zabbix用户能

够执行脚本或程序。

 # vi /etc/zabbix/zabbix-server.conf

ExternalScripts=/usr/lib/zabbix/externalscripts

ExternalScripts 参数中可以使用系统默认的路径，你也可以指定其他路径，需要执行的脚本或程序必须在ExternalScripts配置的目录中，并设置相应的执行权限。以监测系统CPU内核数量的脚本为列子，看一下设置的过程。

1、创建一个检测系统CPU内核数量的脚本。

 # touch cores.sh

2、设置可执行权限。

 # chmod +x cores.sh

3、编辑cores.sh脚本内容。

 # vi cores.sh

#! /bin/bash

nproc

4、设置cores.sh脚本的所有者和所属组为zabbix。

 # chown zabbix:zabbixcores.sh

在External checks监控方式中，定义监控项的key必须遵循特定的语法，语法格式为：script[,,...] 。其中script是脚本/可执行程序的名称，parameter是可选的命令行参数。脚本不带参数时，可以写成script[] 或者 script。

配置External checks监控项的步骤：

1、在Configuration--> Host页面，单击Items列中的链接，例如主机Zabbix server的Items链接，在Items页面右上角点击Create item按钮创建新的监控项。

- Name中输入监控项名称，例如Number of CPU Core。
- Type中选择External check。
- Key中填写脚本的名字，例如：cores.sh[]。
- Host Interface中选择agent接口。
- Type of information中选择Numeric（unsigned）。
- Data Type选择Decimal。
- 其他参数可以保持不变，点击Add按钮保存。

如下图3-12所示。

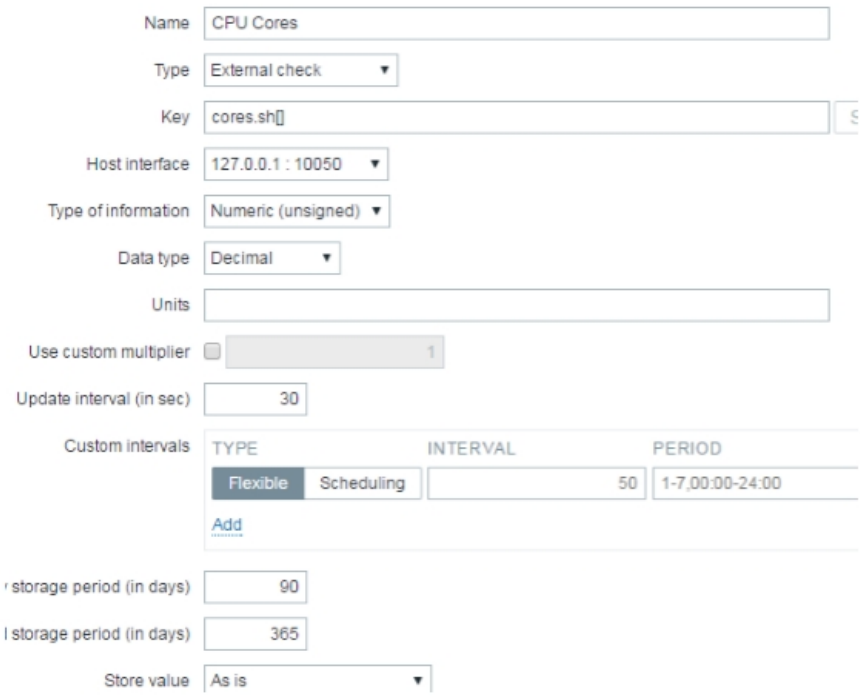

图 3-12

2、Monitoring --> Latest data页面查看监控项。

如果脚本或可执行程序需要传递参数时，可以把这些参数放到名称后面的方括号中，例

如：myscript.sh[“var1”,”var2]”,”var3”]。系统中定义的宏变量也可以传递给脚本或可执行程序，例
如：myscript.sh[“{HOST.IP}”,”var1”]。
这里需要注意脚本或可执行程序的执行时间不能太长，如果超过3秒钟，Zabbix 会标记这个item为unsupported，脚本进程将被杀掉。Zabbix server的配置文件中默认配置的Timeout参数是3秒，你可以调整这个参数，最大可以设置为30秒。

3.11 Database monitor

对于数据库Zabbix也提供了监控方法，可以通过ODBC（Open Database Connectivity）接口对数据库进行数据查询，ODBC做为Zabbix和数据库系统之间的中间件，能够让Zabbix查询不同的数据库收集数据，当然这个数据库需要unixODBC或Independent Open DataBase Connectivity（iODBC）的支持。
使用ODBC监控数据库之前，你要确认Zabbix支持ODBC，通过yum包管理器安装的Zabbix server默认支持ODBC，如果你使用源码编译安装Zabbixserver，在编译时需要使用选项 --with-unixODBC，
首先需要安装unixODBC。

 # yum -y install unixODBC

 # yum -y installunixODBC-devel（源码编译安装Zabbix server时需要）
安装数据库的连接器，我们以MySQL为例子。

 # yum -y install mysql-connector-odbc

接下来配置odbcinst.ini文件，使用下面的命令查找ODBC数据库驱动的位置。

 # odbcinst -j

 # odbcinst –j

```
unixODBC 2.3.1
DRIVERS.............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
USER DATA SOURCES..: /root/.odbc.ini
SQLULEN Size.......: 8
SQLLEN Size........: 8
SQLSETPOSIROW Size.: 8
```

编辑配置文件。

 # vi /etc/odbcinst.ini

```
# Driver from the mysql-connector-odbc package
# Setup from the unixODBC package
[MySQL]
Description  = ODBC for MySQL
Driver       = /usr/lib/libmyodbc5.so
Setup        =/usr/lib/libodbcmyS.so
Driver64     =/usr/lib64/libmyodbc5.so
Setup64      =/usr/lib64/libodbcmyS.so
FileUsage    = 1
```

创建或编辑odbc.ini文件，设置dsn（data source），添加数据库的配置。

 # vi /etc/odbc.ini

```
[mysql-test]                                #dsn 名称
Description = Mysql test DB                 #描述
Driver = mysql
Server = 127.0.0.1
User = root
Password =              #密码
Port = 3306
Database =               #zabbix 数据库
```

完成上面的配置后，我们可以测试能否连接到数据库。

 # isql mysql-test

```
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL>
SQL> select count(*) from items
+----------------------+
| count(*)             |
+----------------------+
| 708                  |
+----------------------+
SQLRowCount returns 1
1 rows fetched
```

当你看到上面的输出内容，并能执行SQL查询语句就说明配置没有问题。接下来就可以添加监控项了。

配置ODBC监控项的步骤：

1、创建新的监控项（Configuration--> Host --> Items --> Ctreate item）。

- Name中输入监控项名称，例如item counts。
- Type中选择Database monitor。
- Key中选择（点击右侧Select按钮）db.odbc.select[,]，其中是我们自己定义的唯一的key的名字，是odbc.ini中定义的DSN名称。
- 如果在odbc.ini中设置了数据库的用户名和密码，Username 和 Password就不需要填写了。
- SQL query中填写我们想要执行的SQL查询语句，例如：select count(*)from items。
- Type of information和Data type中选择相应的数据类型
- 其他配置可以保持不变，点击Add按钮保存。

如下图3-13所示

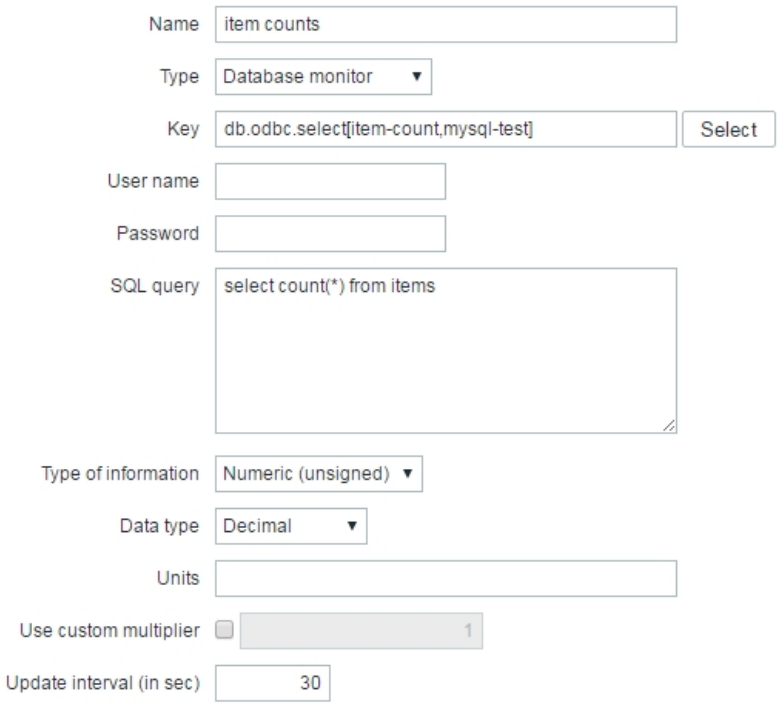

图 3-13

2、Monitoring --> Latest data页面查看监控项。

3.12 SSH agents

Zabbix中通过SSH协议也可以实现监控目标，通过SSH agent监控方式，需要对服务器进行监控但又不能安装Zabbix agent的环境中非常有用。Zabbix中使用SSh agent时要求libssh2的最低版本是 1.0.0。

SSH agent支持两种身份认证的方式：基于用户名密码的方式和基于秘钥的方式，使用用户名密码的方式不需要任何特殊的配置，添加监控项时需要在页面中输入明文的用户名和密码，因此在实际环境中建议使用基于秘钥的方式，但这个方式需要做些额外的配置。下面我们来看看基于秘钥的方式如何配置的。

首先，检查zabbix用户的设置，使用下面的命令。

 # grep zabbix /etc/passwd

zabbix:x:996:994:Zabbix Monitoring System:/var/lib/zabbix:/sbin/nologin

可以看到系统中zabbix用户的home目录是/var/lib/zabbix，确认该目录是否存在，如果不存在，使用下面的命令创建目录。

 # mkdir –p /var/lib/zabbix/.ssh

 # chown –R zabbix:zabbix /var/lib/zabbix

接下来我们需要修改zabbix-server.conf文件，配置SSH Key文件的存储路径。

 # vi /etc/zabbix/zabbix-server.conf

SSHKeyLocation=/ var/lib/zabbix/.ssh

修改完zabbix-server.conf配置文件后重启Zabbixserver。

 # systemctl restartzabbix-server

现在，我们生成zabbix用户的SSH Key，询问passphrase时直接回车就可以。

 # sudo -u zabbixssh-keygen -t rsa -b 2048

```
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/zabbix/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/zabbix/.ssh/id_rsa.
Your public key has been saved in /var/lib/zabbix/.ssh/id_rsa.pub.
The key fingerprint is:
15:3e:d5:61:ed:16:b3:0a:67:9d:35:f0:35:55:0b:7e zabbix@zbxserver
The key's randomart image is:
+--[ RSA 2048]- -----+
|         . .+++*|
|         . o..++*|
|         +  ..EB|
|         . o o.+o|
|        S   + .. |
|            .   |
|                |
|                |
|                |
+----------------------------+
# ll
total 8
-rw------- 1 zabbix zabbix 1675 May 31 13:29 id_rsa
-rw-r--r-- 1 zabbix zabbix 398 May 31 13:29 id_rsa.pub
```

接下来拷贝秘钥文件到被监控主机中，假设被监控主机的IP 地址是192.168.10.112。

 # sudo -u zabbixssh-copy-id root@192.168.10.112

```
The authenticity of host '192.168.10.112 (192.168.10.112)' can't beestablished.
ECDSA key fingerprint is0d:33:e5:5c:43:c3:5b:c4:da:e4:f0:6d:0c:fb:4a:6e.
```

Are you sure you want to continue connecting (yes/no)? yes

/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), tofilter out any that are already installed

/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if youare prompted now it is to install the new keys

root@192.168.10.112's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.10.112'"

and check to make sure that only the key(s) you wanted were added.

现在我们测试一下能否登录成功。

 # sudo -u zabbix ssh root@192.168.10.112

当完成上面的配置后，就可以创建SSH agent监控方式的监控项了。

配置SSH agent监控项的步骤：

1、创建一个新主机（Configuration --> Host --> Create host）。在主机配置页面的Host标签下添加Agentinterfaces接口配置，如下图3-14所示。

图 3-14

2、在主机中创建新的监控项。

- Name中输入监控项名称，例如Check uname。
- Type中选择SSH agent。
- Key中内容替换成ssh.run[uname]。
- Host interface 中选择agent接口。
- Authentication method中选择Public key。
- User name中填写root ，Public key file中填写id_rsa_pub，Private keyfile中填写 id_rsa。
- Key passphrase留空，如果生成密钥时你输入了passphrase，就需要在这里输入相同的passphrase。
- Executed script中输入uname -a。
- Type of information中选择Text类型。
- 其他参数可以保持不变，点击Add按钮保存。

如下图3—15所示。

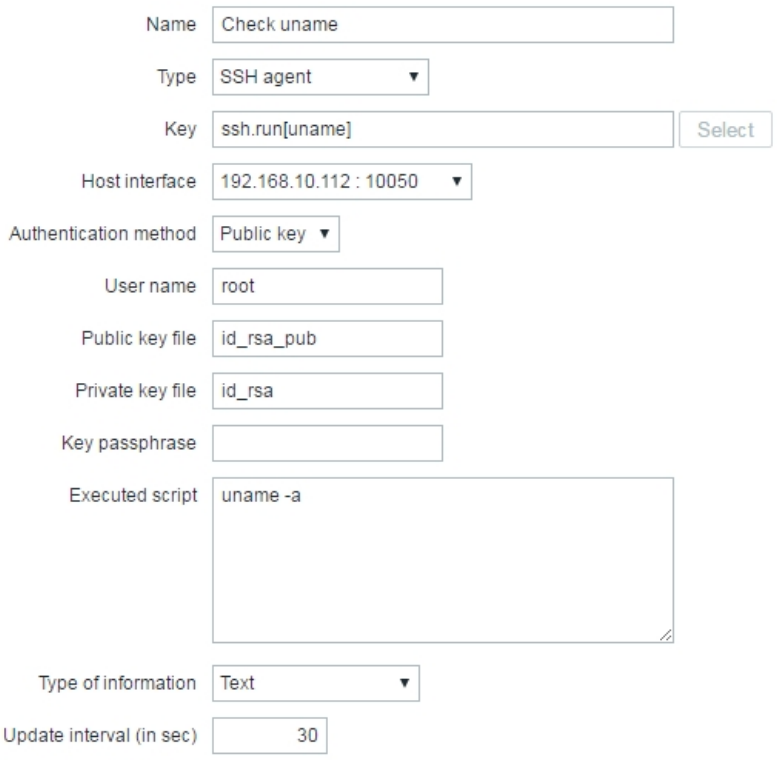

图 3-15

3、Monitoring --> Latest data页面查看监控项。

使用SSH agent监控方式需要注意的是libssh2可能会把可执行脚本的输出截断到32KB，另外在脚本中最好使用命令的全路径。

3.13 Telnet agents

Telnet agent监控方式和SSH agent方式一样，可以在无法安装agent软件的环境中使用。但和SSH 不同的是，Telnet不是一个加密的协议，只支持用户名密码的身份认证方式。除非是只能使用Telnet agent方式而别无选择时可以使用Telnetagent，一般情况下不建议使用，使用时最好将用户账户设置成只读的权限。

使用Telnet agent方式前，需要确认被监控设备中已经安装并启动了Telnet server，设置了使用Telnet 登录系统的用户账户。

如果被监控设备中没有安装Telnet server，可以使用下面的命令安装。

 # yum –y installtelnet-server

编辑xinet.d配置文件，配置telnetserver的参数，将disabled设置从yes变为no。

 # vi /etc/xinetd.d/telnet

```
{
flags = REUSE
socket_type = stream
wait = no
user = root
server = /usr/sbin/in.telnetd
log_on_failure += USERID
disable = no
}
```

设置telnet服务在设备重启后能自动启动。

 # systemctl starttelnet.socket

 # systemctl enabletelnet.socket

在telnet服务器中添加用户。

 # useradd zabbix

 # passwd zabbix

如果Zabbix server中没有安装Telnet客户端，可以使用下面的命令进行安装。

 # yum -y install telnet

接下来我们就可以创建使用Telnet agnet方式的监控项了。

配置Telnet agent监控项的步骤：

1、在主机中创建新的监控项。

- 在Name中输入监控项的名称，例如：Check uname。
- Type中选择TELNET agent。
- Key中输入telnet.run[uname]。
- Host Interface中选择agent接口。
- User name中输入telnet用户名，如：zabbix。
- Password中输入telnet用户密码，如：zabbix。
- Executed script中输入命令，如：uanme -a。
- Type of information中选择Text。
- 其他参数可以保持不变，点击Add按钮保存。

如下图3—16所示。

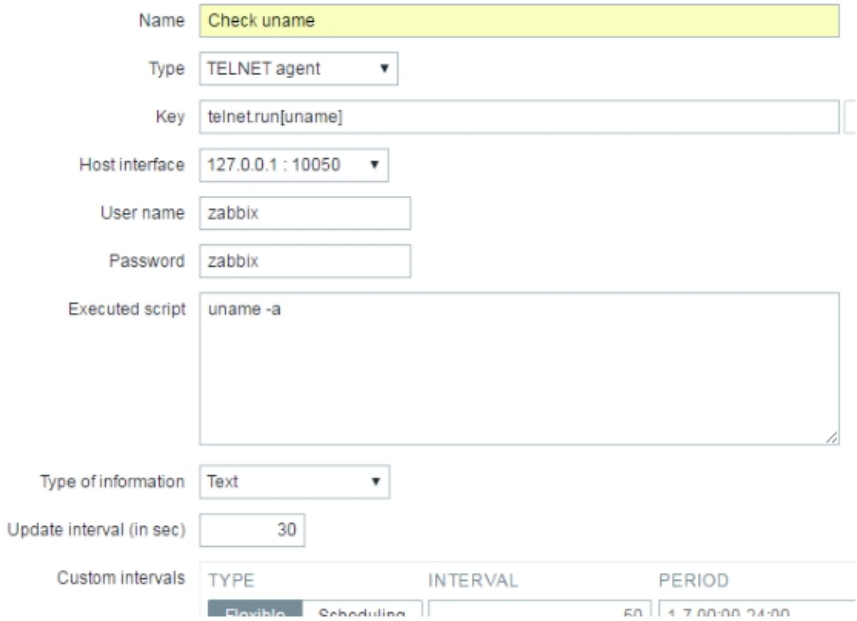

图 3-16

2、Monitoring --> Latest data页面查看监控项。

3.14 SNMP Traps

SNMP Traps监控方式和前面介绍的监控方式有所不同，在这种方式中被监控设备能够主动发送信息到Zabbix server。被监控设备发生一些特定事件时，例如设备重启、网络接口宕掉或磁盘损坏等，被监控设备中的SNMP agent会给Zabbix server发送事件的状态信息。

Zabbix中并没有定义一种简单的监控项指标对应到特定的事件，为了能让Zabbix有效的管理SNMP Traps，Zabbix额外使用snmptrapd守护进程帮助处理来自被监控设备中SNMP agent发送的traps信息，通过脚本进行格式化处理后将结果交给Zabbix server。

配置SNMP Traps监控项的步骤：

1. 创建一个新主机（Configuration--> Host --> Create host）。在主机配置页面的Host标签下添加SNMP interfaces接口配置，如下图3-17所示。

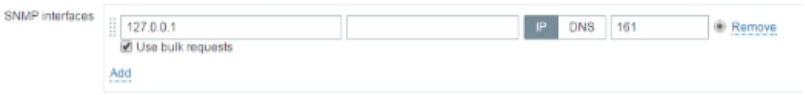

图 3-17

1. 在主机中创建新的监控项。
2. uName中输入Link Down trap。
3. uType选择SNMP trap。
4. uKey中选择snmptrap[]，将替换为linkDown。在这里主要有两个选项，snmptrap.fallback是捕获那些在该接口上没有被snmptrap[]监控项捕获的SNMP traps。snmptrap[]是捕获所有和指定的regexp匹配的SNMP traps，如果没有指定regexp则捕获任意trap。
5. uHost Interface中选择SNMP 接口。
6. uType of information中选择Log。如果想使用其他类型像Numeric也是可以的，但需要我们自定义trap handler处理程序。
7. u其他参数可以保持不变，点击 Add按钮保存。

如下图3-18所示。

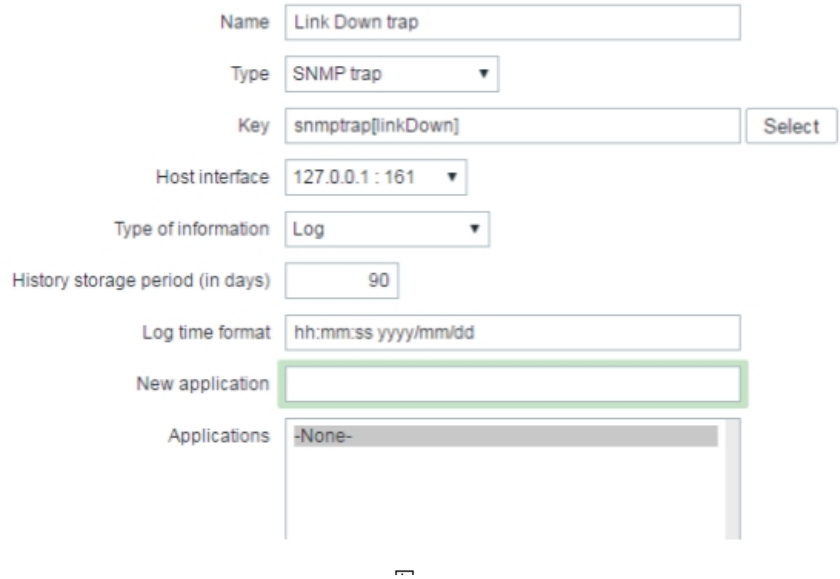

图 3-18

1. Monitoring --> Latest data页面查看监控项。

3.15 Aggregate checks

直到现在，我们看到Zabbix提供的监控方式都是用不同的方法获得监控项的原始数据，但在实际环境中，当我们想了解同一主机组中所有主机CPU负载的总计时，就需要用到Aggregatechecks监控方式。Aggregate我们可以理解为汇总的意思，这种方式不需要从被监控主机中直接收集原始数据，而是从数据库中获取监控项的数据进行计算，因此Aggregate checks仅对Numeric类型的监控项有效。

配置Aggregate 监控项的步骤：

1、创建一个新的主机组，例如aggregated（Configuration --> Host groups --> Create host group），并添加两个或多个Linux 主机到该组中。

2、创建一个新的模板（Configuration--> Templates --> Create template），在这个模板中创建一个监控项，key为system.cpu.load。将这个模板链接到主机组Aggregated中所有的Linux主机上。

3、创建一个新主机（Configuration--> Host --> Create host）。在主机配置页面的Host标签下添加Agent interfaces接口，IP地址可以填写127.0.0.1或者0.0.0.0都可以。

4、在新建主机上添加一个监控项。

- Name中填写监控项名称，例如avg cpu load。
- Type中选择Zabbix aggregate。
- Key设置为grpavg["aggregated","system.cpu.load","last"]。其中aggregated是前面创建的主机组，system.cpu.load是该组中主机上的监控项。
- Type of information中选择Numeric（float）。
- Date type中选择Decimal。
- 其他参数可以保持不变，点击Add按钮保存。

如下图3-19所示。

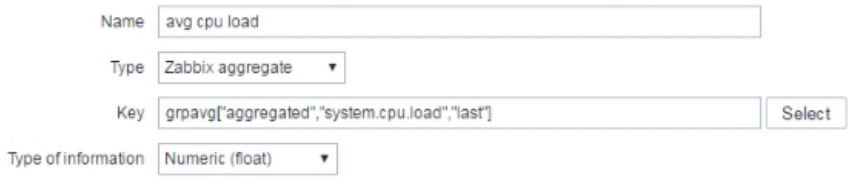

图 3-19

5、Monitoring --> Latest data页面查看监控项。

3.16 Calculated checks

Calculated checks和Aggregatechecks类似，也是从数据库中获取已经存在的数据进行计算，但和Aggregate checks不同的是Calculatedchecks不会限制到特定的主机组。Calculated 监控项的计算结果会保存到数据库中，可以在触发器表达式中使

用Calculated 监控项，也可以在宏变量、图形和动作中像其他类型的监控项一样引用，
Calculated 监控项的定义包含在一个formula（公式）中，根据你的需要可以很简单也可以很复杂。一个简单公式的语法如下。
func(|,,,...)
其中func是触发器表达式中支持的函数，包括：last、min、max、avg和count。Key或hostname:key是你想使用的其他监控项。建议将整个key放到双引号中，以避免由于key中含有空格或逗号造成的解析错误，key中含有双引号时要用斜杠（\）转义。Parameter为所需的参数。
Calculated 监控项遇到下面的情况时可能会变成unsupported：

- 没有发现引用的监控项
- 没有数据供函数计算
- 被零除
- 使用不正确的语法

在Calculated 监控项公式中引用宏变量时，可以对参数和常量进行扩展，不能对函数、主机名、监控项的key或运算符进行扩展。Zabbix处理Calculated监控项时使用的数据是依靠监控项定期收集的监控数据，不像触发器表达式那样接收到新的数据时马上进行计算。
配置Calculated监控项的步骤：
1、创建一个新主机（Configuration--> Host --> Create host）。在主机配置页面的Host标签下添加Agent interfaces接口，IP地址可以填写127.0.0.1或者0.0.0.0都可以。
2、在新建主机上添加一个监控项。

- Name中填写监控项名称，例如% free on root。
- Type中选择Calculated。
- Key中填写free.root，这个名字和其他监控项一样必须是唯一的。
- Formula中输入100*last("vfs.fs.size[/,free]",0)/last("vfs.fs.size[/,total]",0)。
- Type of information中选择Numeric（float）。
- Units中填写%。
- 其他参数可以保持不变，点击Add按钮保存

如下图3-20所示。

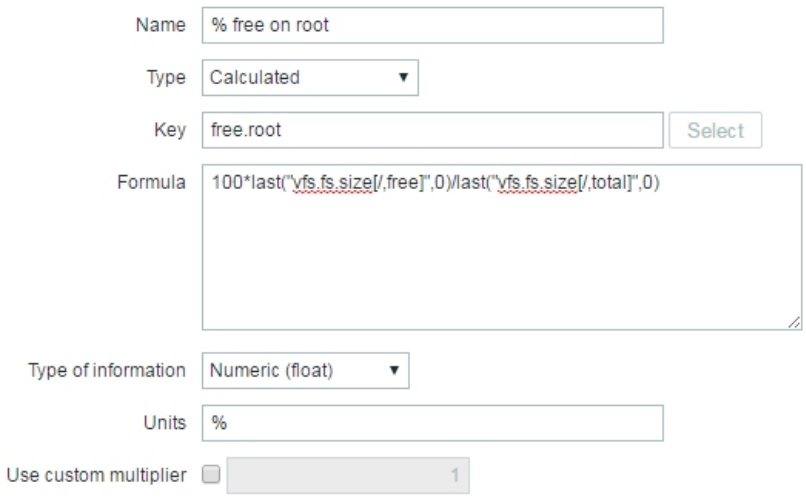

图3-20
3、Monitoring --> Latest data页面查看监控项。