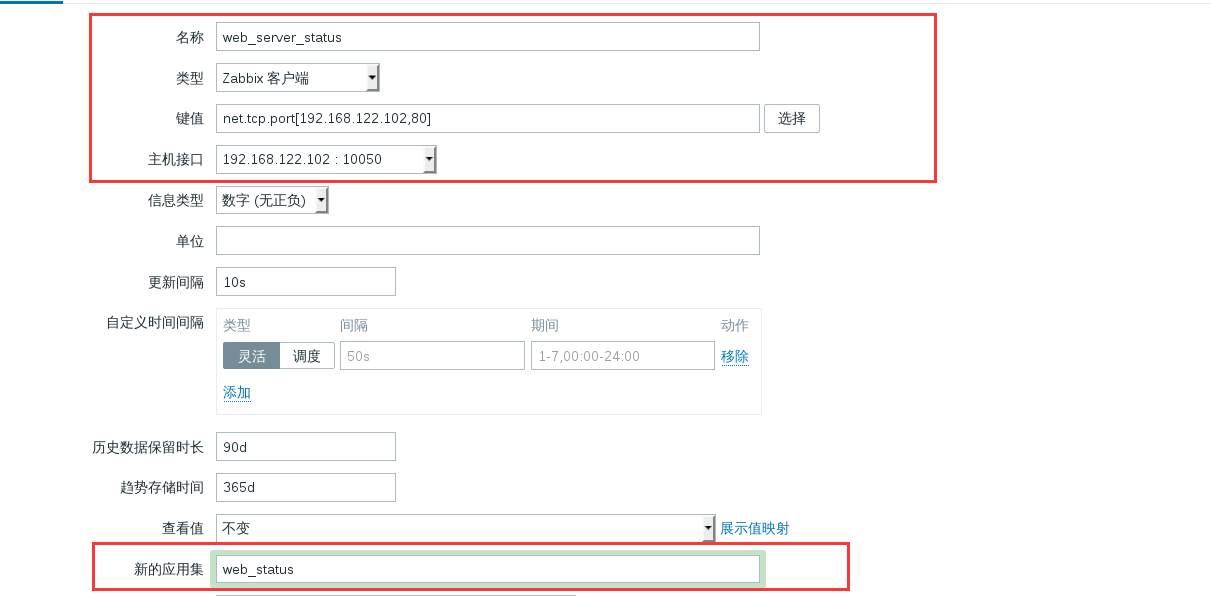
**监控远程主机、自定义监控项**

**监控node01的web服务**



## 自定义监控项

示例01：监控node01的内存剩余量

1) 在被监控机启用自定义监控的功能

[root@node01 ~]# vim /etc/zabbix/zabbix\_agentd.conf

UnsafeUserParameters=1 //自定义监控项可以使用一些特殊字符

UserParameter=memory.free,free -m | awk '/^Mem/{print $4}' //自定义键值memory.free

[root@node01 ~]# systemctl restart zabbix-agent.service

说明: 自定义监控的语法

UserParameter=<key>,<shell command>

<key>：用户自定义的监控项名称

<shell command>：获取对应的值的命令或者脚本，只能是唯一的一个值

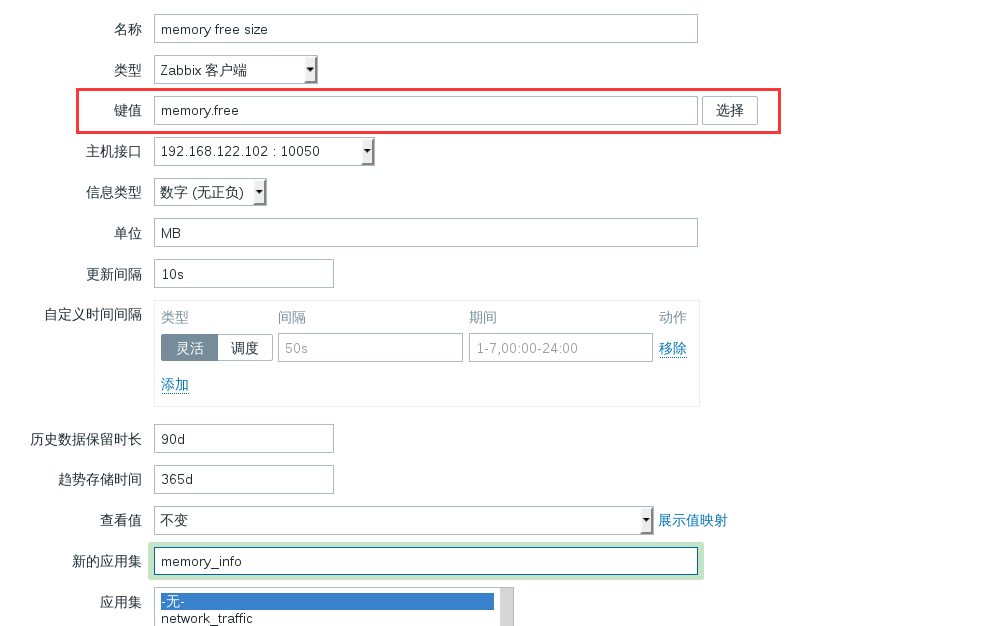
2) 在zabbix server上测试获取被监控机上的自定义键值

[root@zabbix\_server ~]# yum install -y zabbix-get

[root@zabbix\_server ~]# zabbix\_get -s 192.168.122.102 -k "memory.free"

725

3) 在zabbix web页面上添加监控项



示例02：监控node01 httpd的进程数

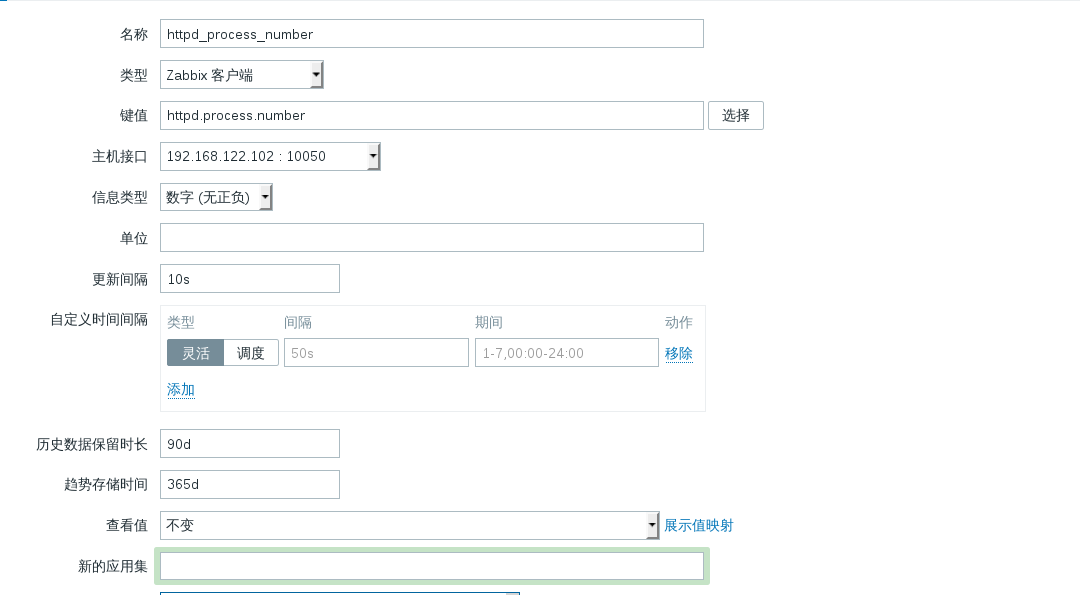
[root@node01 ~]# vim /etc/zabbix/zabbix\_agentd.conf

UserParameter=httpd.process.number,ps aux | grep httpd | grep -v "grep" | wc -l

[root@node01 ~]# systemctl restart zabbix-agent

[root@zabbix\_server ~]# zabbix\_get -s 192.168.122.102 -k "httpd.process.number"

6



**创建带有参数的自定义监控项**

**1) 在被监控机创建自定义键值**

**[root@node01 ~]# vim /etc/zabbix/zabbix\_agentd.conf**

**UserParameter=memory.size[\*],cat /proc/meminfo | awk '/^$1/{print $$2}'**

**[\*]:允许键值接受参数**

**$1:用户传递的第一个参数名称;**

**由于$1的写法和awk中冲突,awk中对应的写法要换为$$1**

**[root@node01 ~]# systemctl restart zabbix-agent.service**

**2) 在zabbix server测试获取数据**

**[root@zabbix\_server ~]# zabbix\_get -s 192.168.122.102 -k "memory.size[Buffers]"**

**948**

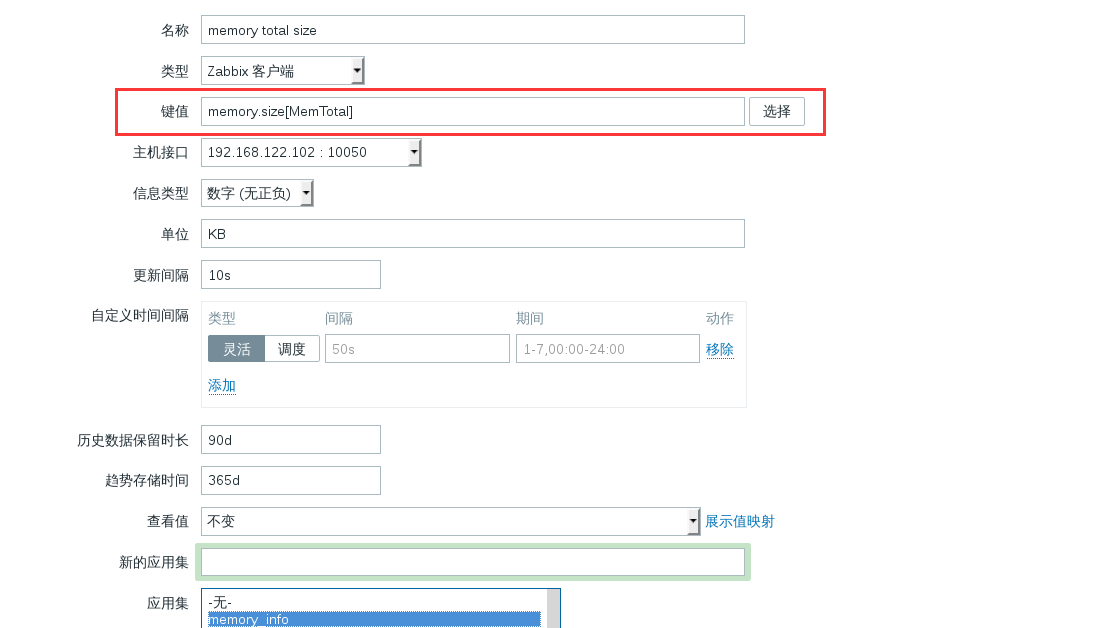
**[root@zabbix\_server ~]# zabbix\_get -s 192.168.122.102 -k "memory.size[MemFree]"**

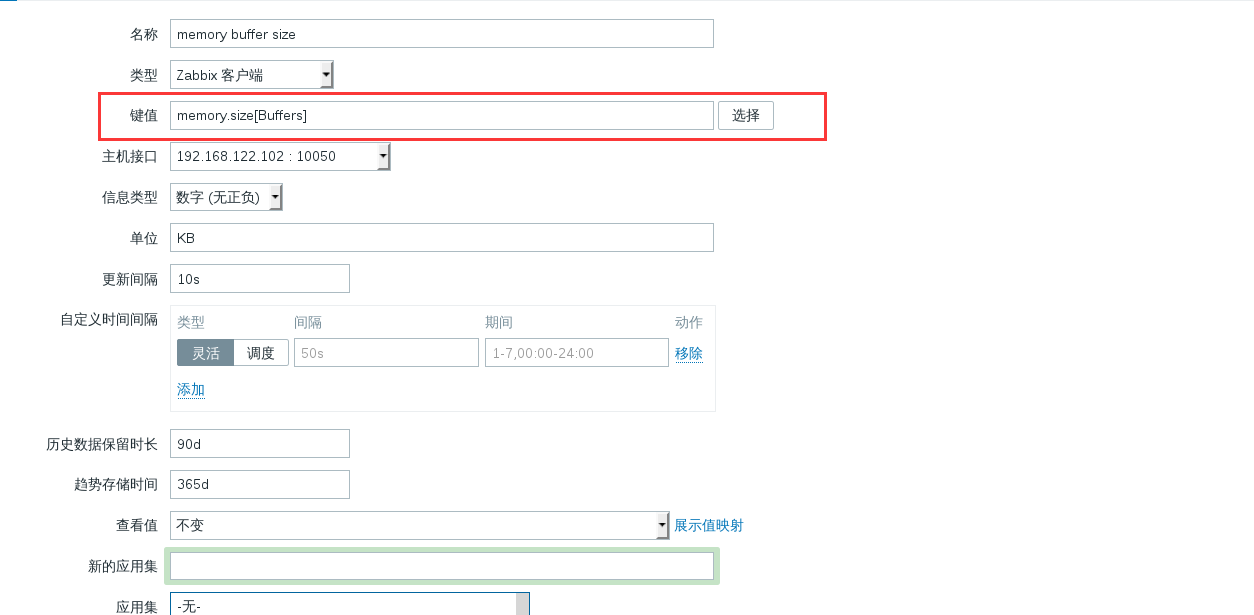
**589128**

**[root@zabbix\_server ~]# zabbix\_get -s 192.168.122.102 -k "memory.size[MemTotal]"**

**1016916**

**3) 添加监控项**





**磁盘**

[root@node01 ~]# cat /proc/diskstats

1 0 sda 505235 28461 7421219 6259625 2397331 3804539 49818500 25885599 0 6122393 32152888

设备号 编号 设备 读完成次数rps 合并完成次数rrps 读扇区次数 读操作花费毫秒数 写完成次数 合并写完成次数 写扇区次数 写操作花费的毫秒数 正在处理的输入/输出请求数 输入/输出操作花费的毫秒数 输入/输出操作花费的加权毫秒数。

**网卡**

[root@node01 ~]# cat /proc/net/dev

Inter-| Receive 接收 | Transmit 发送

face |**bytes packets errs drop** fifo frame compressed multicast|bytes packets errs drop fifo colls carrier compressed

eth0: 59135654 665686 0 4 0 0 0 0 46293653 622032 0 0 0 0 0 0

lo: 24377640 446596 0 0 0 0 0 0 24377640 446596 0 0 0 0 0 0

MySQL：

1) 进程状态、数据目录大小、二进制日志目录大小

2) 连接数

[root@node01 ~]# mysql -uroot -predhat -e "show processlist" | sed '1d' | wc -l

3) 查看MySQL数据库状态

[root@node01 ~]# mysql -uroot -predhat -e "status"

Uptime: 6 min 34 sec //数据库的运行时间

Threads: 1 Questions: 52 Slow queries: 0 Opens: 0 Flush tables: 2 Open tables: 26 Queries per second avg: 0.131

Threads 线程数

Slow queries 慢查询数量

Queries per second avg: 平均秒数

4) 扩展状态/所有变量的值

[root@node01 ~]# mysqladmin -uroot -predhat extended-status

Com\_commit // 每秒提交事务的次数

Com\_rollback // 每秒回滚事务的次数

Com\_delete // 每秒执行delete的次数

Com\_insert

Com\_select

Com\_update

Slow\_queries //慢查询次数

max\_connections //最大连接数

Queries // 每秒执行的所有查询数

5) 主从复制

# mysql -uroot -predhat “show slave status”

IO thread

SQL thread

延迟时间 Seconds\_Behind\_Master: 0

nginx, http, tomcat, lnmp