Zabbix监控mysql存活，及性能

### **角色：Zabbix Agent, Zabbix Server, MySQL, 模板**

### **第一步，监控规划**

在创建监控项之前要尽量考虑清楚要监控什么，怎么监控，监控数据如何存储，监控数据如何展现，如何处理报警等。要进行监控的系统规划需要对Zabbix很了解，这里只是提出监控的需求。

需求一：监控MySQL的状态，当状态发生异常，发出报警；

需求二：监控MySQL的操作，并用图表展现；

第二步，使用自定义脚本监控扩展Agent。

Zabbix Server与Agent之间监控数据的采集主要是通过Zabbix Server主动向Agent询问某个Key的值，Agent会根据Key去调用相应的函数去获取这个值并返回给Server端。Zabbix 2.4.7的Agent本并没有内置MySQL的监控功能（但是Server端提供了相应的Template配置），所以我们需要使用Zabbix的User Parameters功能，为MySQL添加监控脚本。

### **Trapper工作原理:**

被监控主机根据用户设定的时间间隔定期将数据push到Zabbix Server.这里主要介绍Agent.

### **Agent工作原理:**

* Agent 安装在被监控主机上，定期主动的监控本机的资源和应用,然后将数据进行处理发送给ZabbixServer. Agent工作方式又分为Passive Check 和 Active Check。
* Passive Check: Zabbix Server 发起数据索取请求，Agent响应对应的数据.
* Active Check: Agent首先从Zabbix Server 检索监控项列表,然后定期将对应的数据主动的发送到.Zabbix ServerZabbix Agent 本身预定义了一些监控类型,而对于没有预定义的需要管理员自行定义.因此，Zabbix提供了”UserParameter”参数,以方便用户根据自己的需求自定义想要获取的数据.

### **“UserParameter” 语法:**

UserParameter=<key>,<command>

用户自定义一个key; 为命令，该命令用来获取用户想要监控的数据,也就是key的值;  
定好UserParameter参数后，在为主机或者模板配置监控项的时候，在”key”中输入上面自定义的key的  
名字就可以了.

假如我要获取Mysql Server的版本，我可以这样定义”UserParameter”:  
打开 Zabbix Agent安装路径下的 ../etc/zabbix\_agentd.conf 配置文件，翻页到最后页面，键入如下  
行:

UserParameter=mysql.version,mysql -V

这里我们自定义的key名就是"mysql.version",命令"mysql -V"用来获取Mysql 版本号,其实就是key对  
应的值.

UserParameter参数实现的原理通俗来讲，就是我们先要熟悉Mysql命令，通过Mysql的命令获取想要的  
数据，然后赋值给自定义的key，最后通过Zabbix Server获取这个值通过图像等方式展示出来.

下面利用Agent来实现对mysql性能的监控。

**授权mysql登录用户（agent端）：**

mysql> grant usage on \*.\* to zabbix@127.0.0.1 identified by '123456';

mysql> flush privileges;

### **agent端：**

1.利用UserParameter参数自定义Agent Key。  
对于需求一 ，我们采用mysqladmin这个工具来实现，命令如下：

linux:/etc/zabbix/zabbix-agentd.conf.d # mysqladmin -uzabbix -p'123456' -h127.0.0.1 ping

mysqld is alive

如果MySQL状态正常，会显示mysqld is alive，否则会提示连接不上。对于服务器端，mysqld is alive这样的句子不好理解，服务器端最好只接收1和0，1表示服务可用，0表示服务不可用。那么再改进一下这个命令，如下：

linux:/etc/zabbix/zabbix-agentd.conf.d # mysqladmin -uzabbix -p'123456' -h127.0.0.1 ping | grep -c alive1

用户名和密码放在命令中对于以后的维护不好，所以我们在/etc/zabbix/下创建一个包含MySQL用户名和密码的配置文件“.my.cnf”，如下：(**我这里没有这么做**)

[client]

user=zabbix

host=192.168.2.6

password=123456

有了这个文件后的命令变更为

linux:/etc/zabbix/zabbix-agentd.conf.d # HOME=/etc/zabbix/ mysqladmin ping | grep -c alive

做完这一步后需要做的就是，将这个监控命令添加到Zabbix Agent中，并与一个Key对应，这样Zabbox Server就能通过这个Key获取MySQL的状态了。我们使用mysql.ping作为MySQL状态的Key。

首先在去除/etc/zabbix/zabbix-agentd.conf中 包含子配置文件路径的注释，并修改

Include=/etc/zabbix/zabbix-agentd.conf.d/ （这里根据自己情况取名就行，别忘了创建这个目录）

其次，在etc/zabbix/zabbix\_agentd.conf.d/目录下创建userparameter\_mysql.conf文件。在文件中添加如下命令：

linux:/etc/zabbix/zabbix-agentd.conf.d # cat userparameter\_mysql.conf

UserParameter=mysql.ping,mysqladmin -uzabbix -p'123456' -h127.0.0.1 ping | grep -c alive

这个命令中”UserParameter”表示这是一个用户自定义的脚本；“=”号后是脚本的内容；“mysql.ping”是Key，“，”号后的命令会在Zabbix Server向Agent发起获取“mysql.ping”这个key的请求时被调用，并将返回值返回给Server。  
保存并退出后可以使用下面的命令测试是否正常工作。

linux:/etc/zabbix # /usr/sbin/zabbix-agentd -t mysql.ping -c /etc/zabbix/zabbix-agentd.conf.d/userparameter\_mysql.conf 　　（正常工作）

mysql.ping [t|1]

这里zabbix\_agentd使用方法可参考：

[http://www.ttlsa.com/zabbix/zabbix-command-zabbix\_agentd/](http://www.ttlsa.com/zabbix/zabbix-command-zabbix_agentd/" \t "https://www.cnblogs.com/shenjianyu/p/_blank)

同时，在Server端也可以使用使用zabbix\_get命令来测试从Server端获取指定的Client端的数据，如下：

[root@Zabbixserver alertscripts]# zabbix\_get -s192.168.2.6 -p 10050 -k mysql.ping1

这里如果操作是跟我一样，但你是有错误的

可能跟你安装版本不统一有问题。

这里zabbix\_get使用方法可参考:[http://www.ttlsa.com/zabbix/zabbix-zabbix\_get-get-items/](https://www.cnblogs.com/shenjianyu/p/ http://www.ttlsa.com/zabbix/zabbix-zabbix_get-get-items/" \t "https://www.cnblogs.com/shenjianyu/p/_blank)

也可以参考官网方法：

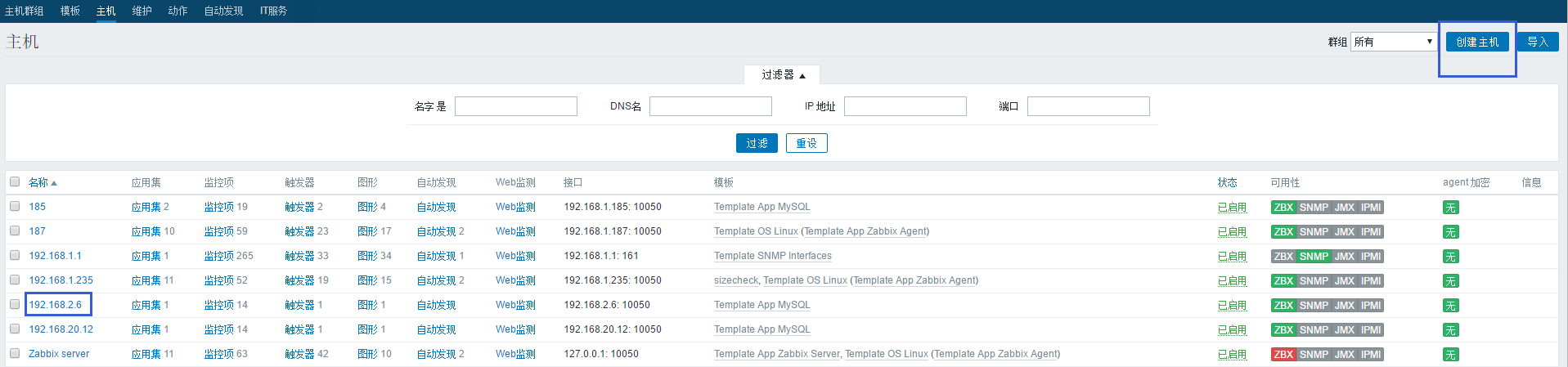
[https://www.zabbix.com/documentation/3.0/manual/config/items/userparameters/extending\_agent](https://www.zabbix.com/documentation/3.0/manual/config/items/userparameters/extending_agent" \t "https://www.cnblogs.com/shenjianyu/p/_blank)

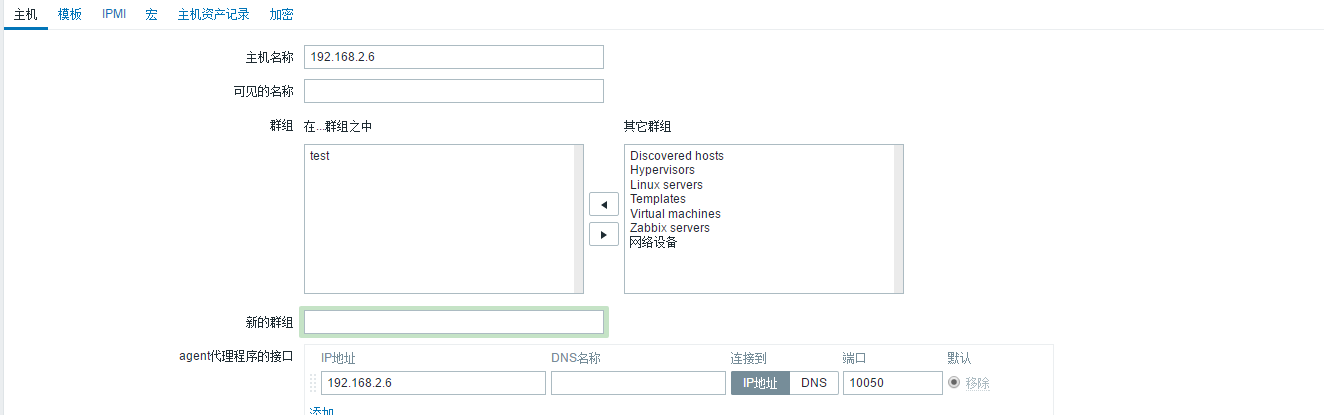
然后下载安装MySQL监控的模板　　（我这里使用脚本监控+zabbix自带MySQL模板）

[https://github.com/yangcvo/zabbix.2.4/](https://github.com/yangcvo/zabbix.2.4/" \t "https://www.cnblogs.com/shenjianyu/p/_blank)

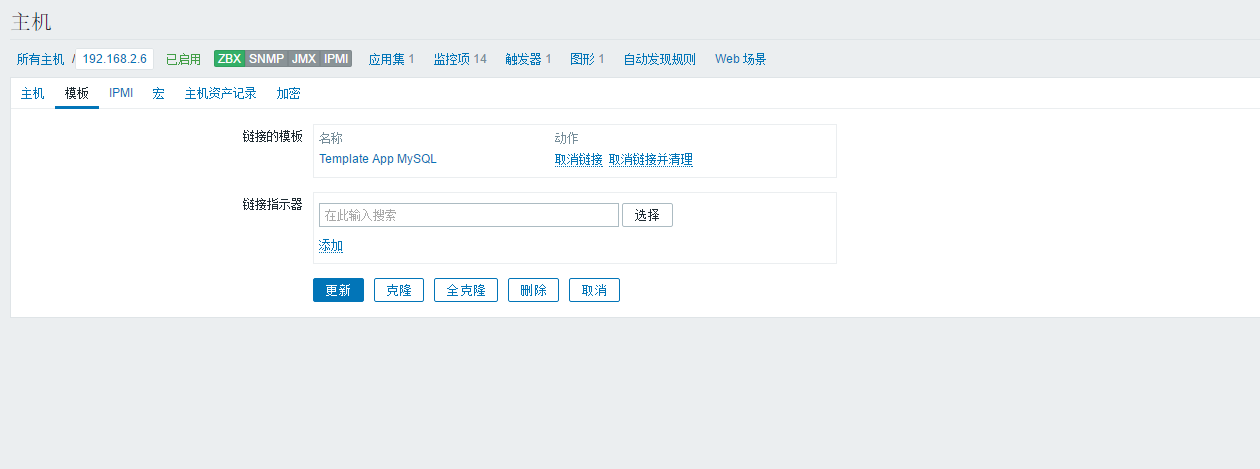
我这里使用的是zabbix 自带模板 [Template App MySQL](http://172.30.1.208:8051/templates.php?form=update&templateid=10073" \t "https://www.cnblogs.com/shenjianyu/p/_blank)

**创建主机 192.168.2.6**





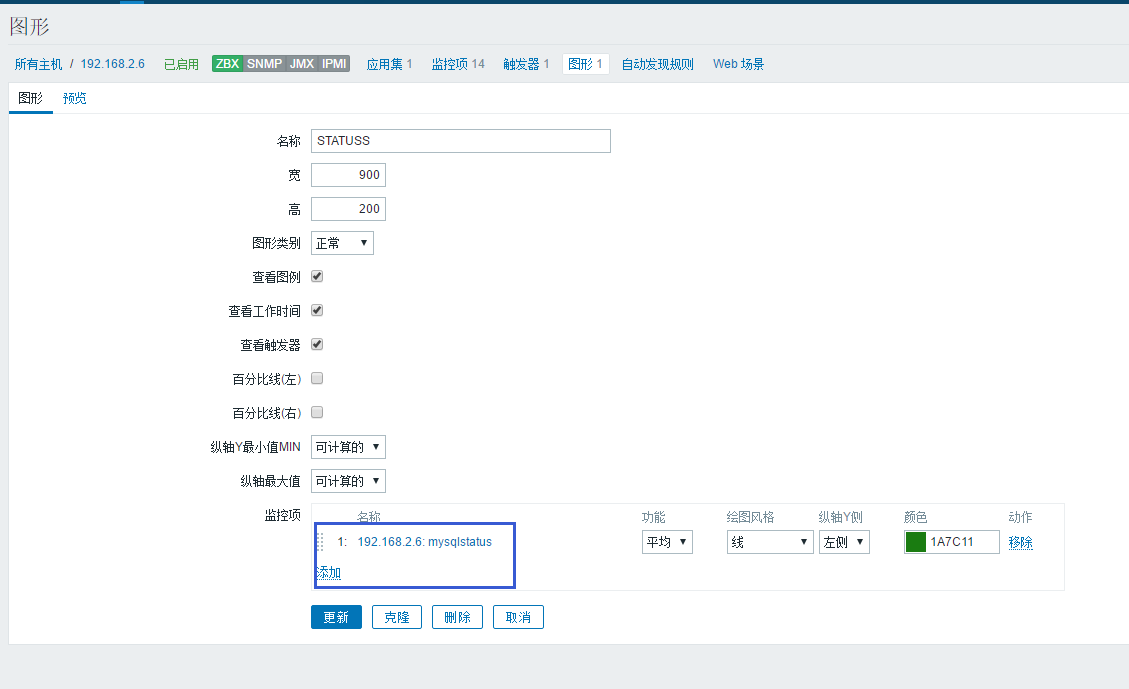
**关联模板**



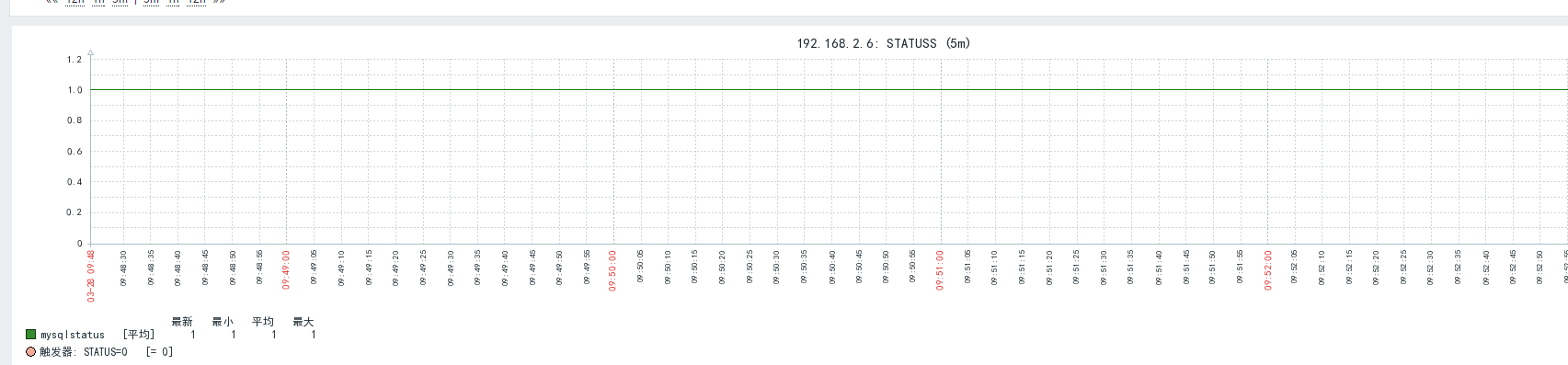
**创建监控项**



**创建图形**



**查看监控图像**



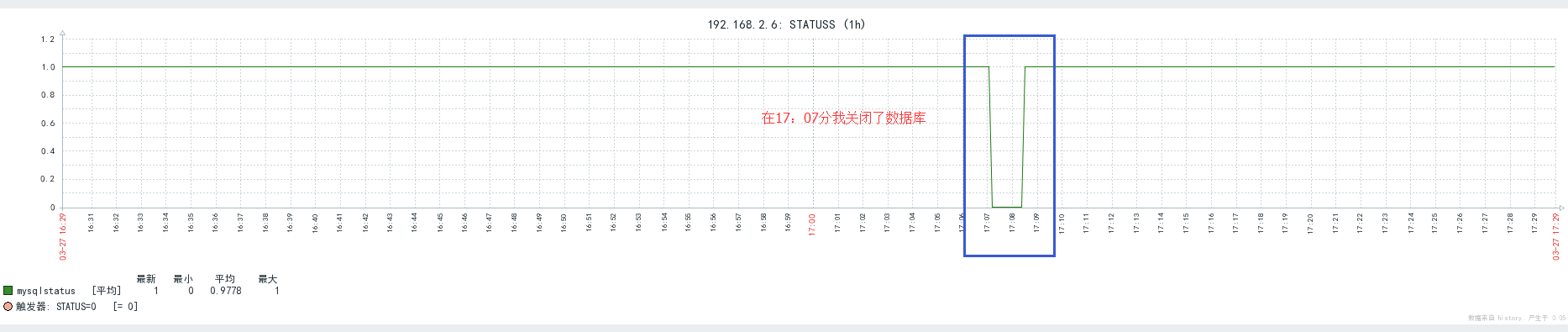
**设置报警，创建触发器，动作（当STATUS=0时，报警）**



**进行测试**

停掉数据库（测试环境）

linux:~ # /etc/init.d/mysql stop

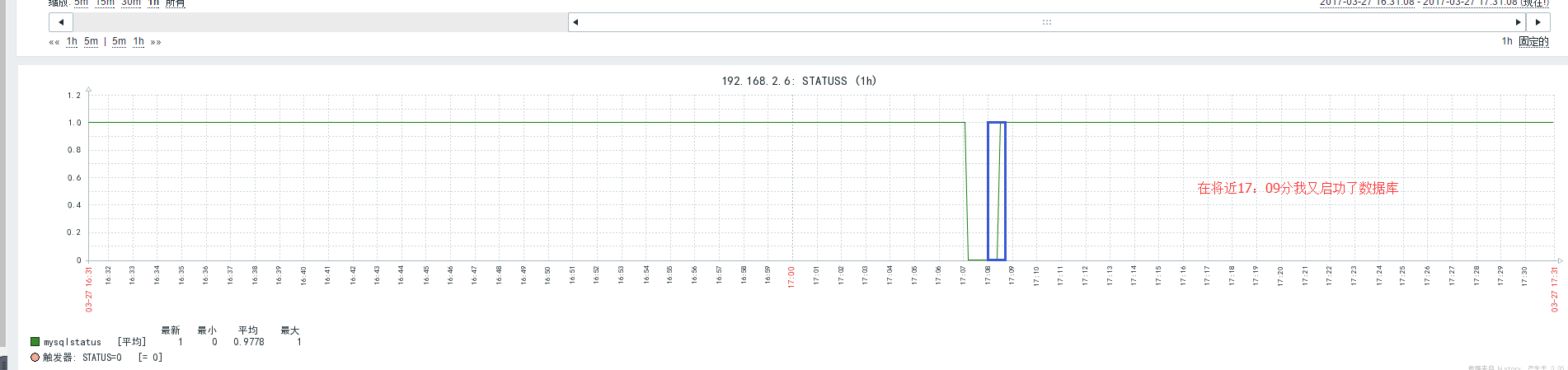


**查看邮件**

邮件提示我数据库STATUS=0 也就是down了

恢复数据库

linux:~ # /etc/init.d/mysql start



邮件提示我数据库STATUS=1也就是UP了

**zabbix监控mysql性能**  
通过获取mysql状态值将这些状态值传递给服务器并绘制成图片，这样可以观察mysql的工作情况，通常需要获得状态变量有以下

Com\_update：mysql执行的更新个数

Com\_select：mysql执行的查询个数

Com\_insert：mysql执行插入的个数

Com\_delete：执行删除的个数

Com\_rollback：执行回滚的操作个数

Bytes\_received:接受的字节数

Bytes\_sent：发送的字节数

Slow\_queries：慢查询语句的个数

监控mysql脚本（chk\_mysql.sh 在客户端编写）

#!/bin/bash

# -------------------------------------------------------------------------------

# FileName: check\_mysql.sh

# Revision: 1.0

# -------------------------------------------------------------------------------

# Copyright:

# License: GPL

# 用户名

MYSQL\_USER='zabbix'

# 密码

MYSQL\_PWD='123456'

# 主机地址/IP

MYSQL\_HOST='127.0.0.1'

# 端口

MYSQL\_PORT='3306'

# 数据连接

MYSQL\_CONN="/usr/bin/mysqladmin -u${MYSQL\_USER} -p${MYSQL\_PWD} -h${MYSQL\_HOST} -P${MYSQL\_PORT}"

# 参数是否正确if [ $# -ne "1" ];then

echo "arg error!"

fi

# 获取数据case $1 in

Uptime)

result=`${MYSQL\_CONN} status|cut -f2 -d":"|cut -f1 -d"T"`

echo $result

;;

Com\_update)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_update"|cut -d"|" -f3`

echo $result

;;

Slow\_queries)

result=`${MYSQL\_CONN} status |cut -f5 -d":"|cut -f1 -d"O"`

echo $result

;;

Com\_select)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_select"|cut -d"|" -f3`

echo $result

;;

Com\_rollback)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_rollback"|cut -d"|" -f3`

echo $result

;;

Questions)

result=`${MYSQL\_CONN} status|cut -f4 -d":"|cut -f1 -d"S"`

echo $result

;;

Com\_insert)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_insert"|cut -d"|" -f3`

echo $result

;;

Com\_delete)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_delete"|cut -d"|" -f3`

echo $result

;;

Com\_commit)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_commit"|cut -d"|" -f3`

echo $result

;;

Bytes\_sent)

result=`${MYSQL\_CONN} extended-status |grep -w "Bytes\_sent" |cut -d"|" -f3`

echo $result

;;

Bytes\_received)

result=`${MYSQL\_CONN} extended-status |grep -w "Bytes\_received" |cut -d"|" -f3`

echo $result

;;

Com\_begin)

result=`${MYSQL\_CONN} extended-status |grep -w "Com\_begin"|cut -d"|" -f3`

echo $result

;;

\*)

echo "Usage:$0(Uptime|Com\_update|Slow\_queries|Com\_select|Com\_rollback|Questions|Com\_insert|Com\_delete|Com\_commit|Bytes\_sent|Bytes\_received|Com\_begin)"

;;

esac

这是针对5.6及以前的，5.7之后，如果密码暴露了会有警告，故

Bytes\_sent)

result=`${MYSQL\_CONN} extended-status 2>/etc/null |grep -w "Bytes\_sent" |cut -d"|" -f3`

echo $result

;;

重启zabbix客户端

linux:/etc/zabbix # /etc/init.d/zabbix-agentd restart

**在userparameter\_mysql.conf** **增加自定义key**

linux:/etc/zabbix # cat zabbix-agentd.conf.d/userparameter\_mysql.conf

UserParameter=mysql.ping,mysqladmin -uzabbix -p'123456' -h127.0.0.1 ping | grep -c alive

UserParameter=mysql.version,mysql -V

UserParameter=mysql.status[\*],/etc/zabbix/chk\_mysql.sh $1

**在zabbix服务器端测试**

[root@Zabbixserver alertscripts]# zabbix\_get -s 192.168.2.6 -p10050 -k mysql.status[Com\_insert]121354

[root@Zabbixserver alertscripts]# zabbix\_get -s 192.168.2.6 -p10050 -k mysql.status[Uptime] 62602

**在zabbix客户端测试是否能获取到信息**

linux:/etc/zabbix/zabbix-agentd.conf.d # zabbix-agentd -tmysql.status[Uptime] -c /etc/zabbix/zabbix-agentd.conf.d/userparameter\_mysql.conf

mysql.status[Uptime] [t|62325]

 创建图形（步骤就不演示了）

**查看图形**

