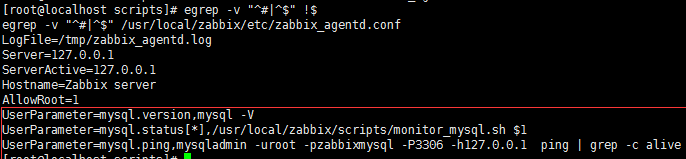
**Zabbix监控Mysql数据库状态性能**

**[1] 配置文件zabbix\_agentd.conf里面添加mysql监控信息：**

|  |
| --- |
| LogFile=/tmp/zabbix\_agentd.log  Server=127.0.0.1  ServerActive=127.0.0.1  Hostname=Zabbix server  AllowRoot=1  UserParameter=mysql.version,mysql -V  UserParameter=mysql.status[\*],/usr/local/zabbix/scripts/chk\_mysql.sh $1  UserParameter=mysql.ping,mysqladmin -uroot -pzabbixmysql -P3306 -h127.0.0.1 ping | grep -c alive |



**[2]添加mysql帐号：**

|  |
| --- |
| GRANT PROCESS,SUPER,REPLICATION CLIENT ON \*.\* TO zabbix@'127.0.0.1' IDENTIFIED BY '123456'; |

**[3]添加check脚本权限：**

|  |
| --- |
| chmod u+x /usr/local/zabbix/ scripts/chk\_mysql.sh  chown -R zabbix.zabbix /usr/local/zabbix/ scripts/chk\_mysql.sh |

**[4]到zabbix服务器上面检查基本mysql服务器信息是否能正常获取到，/usr/local/zabbix/bin/zabbix\_get -s 127.0.0.1 -p 10050 -k mysql.status[Com\_insert]**



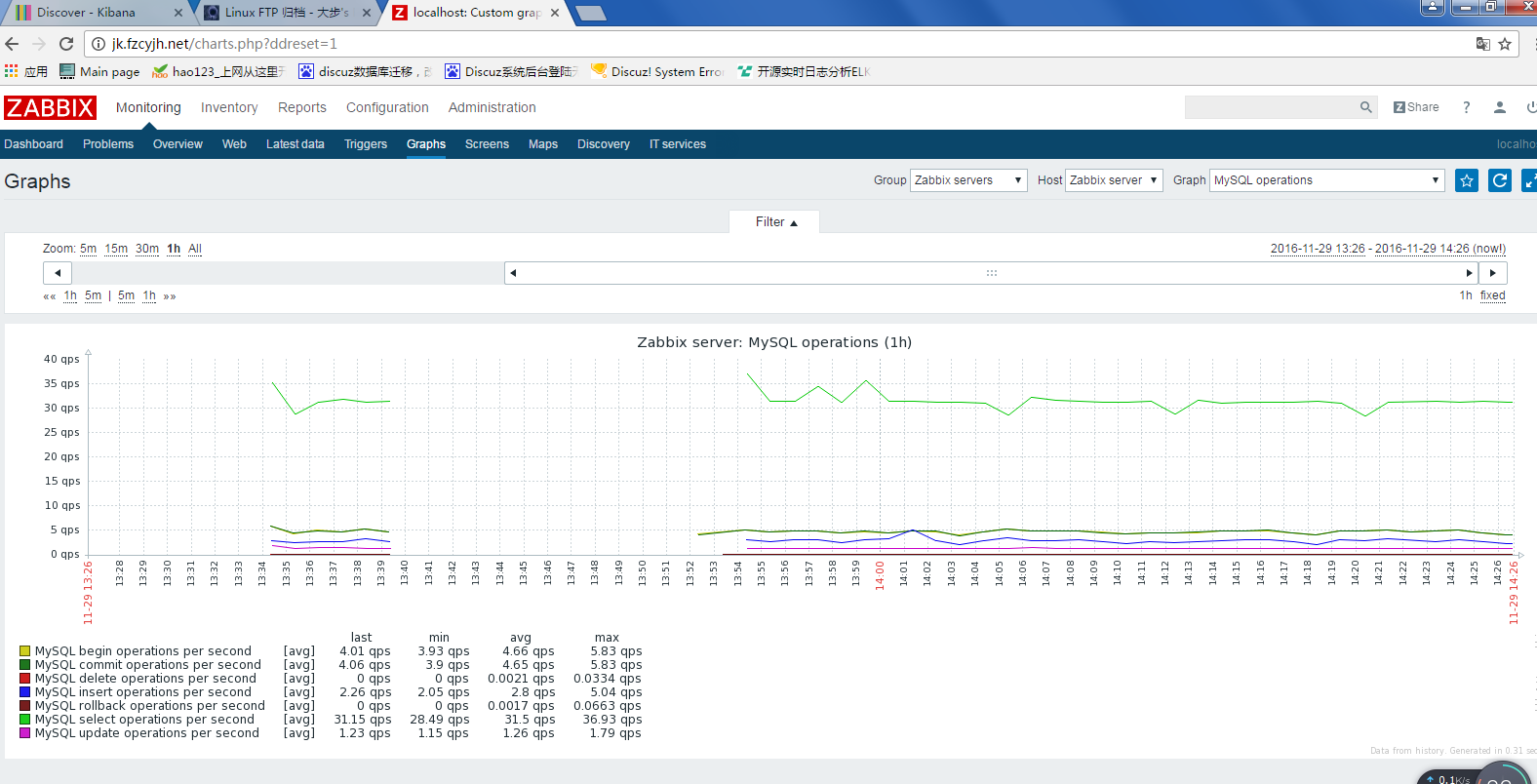
**[5]接下来准备mysql的运行信息监控，需要准备新的mysql监控脚本chk\_mysql.sh,/usr/local/zabbix/chk\_mysql.sh：**

****

|  |
| --- |
| #!/bin/sh  # -------------------------------------------------------------------------------  # FileName: check\_mysql.sh  # Revision: 1.0  # Date: 2016/11/28  export MYSQL\_PWD='zabbixmysql'  mysqladmin="/usr/local/mysql/bin/mysqladmin"  MYSQL\_SOCK="/tmp/mysql.sock"  MYSQL\_USER='root'  #MYSQL\_PWD='zabbixmysql'  MYSQL\_HOST='127.0.0.1'  MYSQL\_PORT='3306'  ARGS=1  if [ $# -ne "$ARGS" ];then  echo "Please input one arguement:"  fi  case $1 in  Uptime)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK status|cut -f2 -d":"|cut -f1 -d"T"`  #echo "$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK status|cut -f2 -d":"|cut -f1 -d"T""  echo $result  ;;  Com\_update)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Com\_update"|cut -d"|" -f3`  echo $result  ;;  Slow\_queries)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK status |cut -f5 -d":"|cut -f1 -d"O"`  echo $result  ;;  Com\_select)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Com\_select"|cut -d"|" -f3`  echo $result  ;;  Com\_rollback)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Com\_rollback"|cut -d"|" -f3`  echo $result  ;;  Questions)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK status|cut -f4 -d":"|cut -f1 -d"S"`  echo $result  ;;  Com\_insert)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Com\_insert"|cut -d"|" -f3`  echo $result  ;;  Com\_delete)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Com\_delete"|cut -d"|" -f3`  echo $result  ;;  Com\_commit)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Com\_commit"|cut -d"|" -f3`  echo $result  ;;  Bytes\_sent)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Bytes\_sent" |cut -d"|" -f3`  echo $result  ;;  Bytes\_received)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Bytes\_received" |cut -d"|" -f3`  echo $result  ;;  Com\_begin)  result=`$mysqladmin -u$MYSQL\_USER -h$MYSQL\_HOST -S $MYSQL\_SOCK extended-status |grep -w "Com\_begin"|cut -d"|" -f3`  echo $result  ;;  \*)  echo "Usage:$0(Uptime|Com\_update|Slow\_queries|Com\_select|Com\_rollback|Questions)"  ;;  esac |

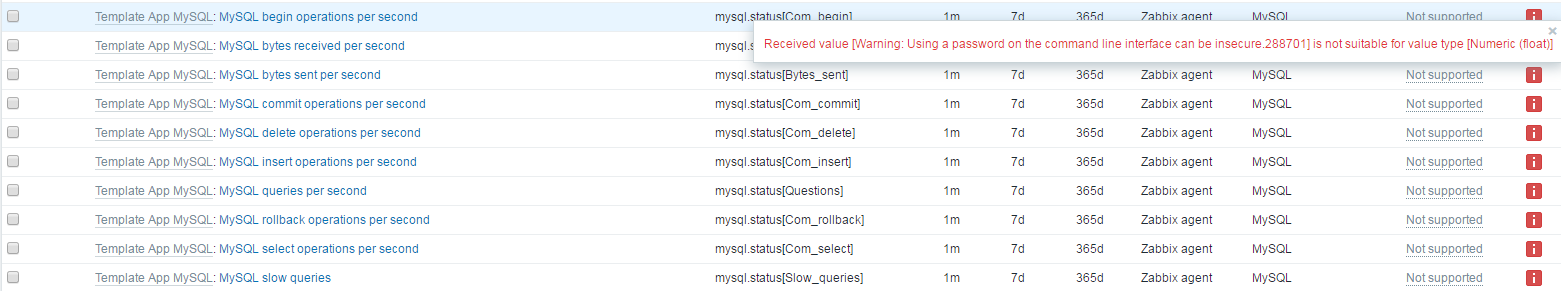
**[6]脚本准备完后，重启agent，然后在zabbix-server服务器上**[**测试**](http://lib.csdn.net/base/softwaretest)**是否可以获取到mysql的status信息：**

|  |
| --- |
| [root@localhost ~]# /usr/local/zabbix/bin/zabbix\_get -s 127.0.0.1 -p 10050 -k mysql.status[Com\_insert]  Warning: Using a password on the command line interface can be insecure.  4713 |



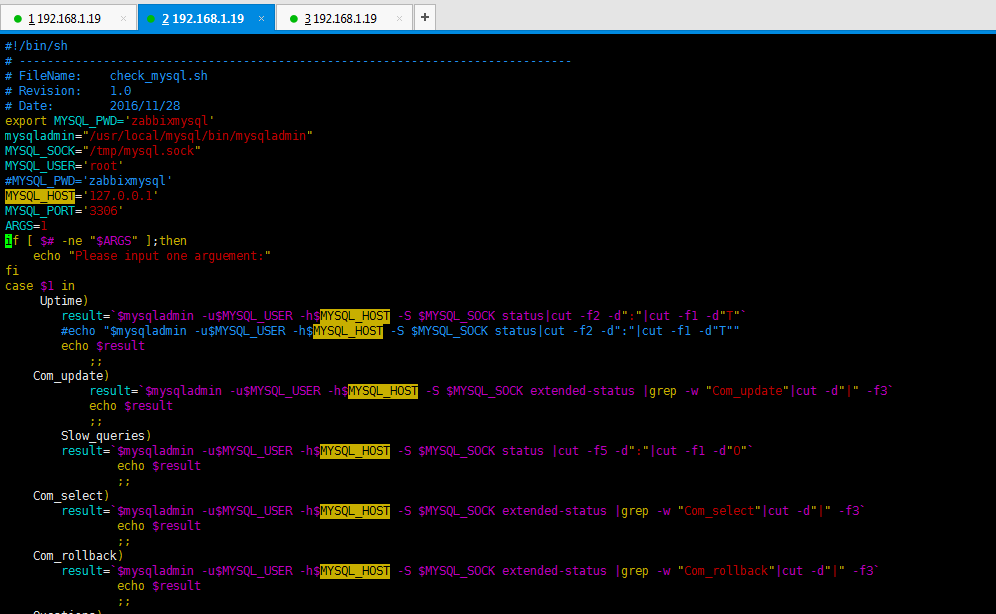
在MySQL中, 如果显示的输入密码去登录的话, 就会有一个Warning显示出来, 提醒这种使用方式会不安全,

[Warning] Using a password on the command line interface can be insecure.



这种方式其实就是利用常量来代替了明文输入密码的这一步骤,   
使用方法也很简单, 在脚本中加入如下命令即可  
---------------------------------------------  
export MYSQL\_PWD=<password>  
---------------------------------------------

之后在脚本中就不用带上 -p参数了, 直接在命令行试试



第一次尝试连接用的是root, 可以发现即使不带上-p参数, 也在后面显示出了Using password: Yes,   
说明设定的这个常量已经被当做密码拿去验证登录了;

参考文档

|  |
| --- |
| <http://blog.csdn.net/mchdba/article/details/51288767>  <http://blog.itpub.net/29510932/viewspace-2098853/> |