



myadmin tools



- myadmin是什么？
- myadmin的适用场景
- myadmin的安装步骤
- myadmin的使用概述
- myadmin的命令详解
- myadmin的安全建议
- myadmin的功能扩展

myadmin是一个简单的mysql的管理工具，借鉴了oracle的ora工具的思路，通过执行my命令加相关参数可以完成mysql的简单监控和运维。此小工具由shell脚本编写。脚本里面组装了服务器和数据库的相关命令和查询信息，在平时日常维护中，可以更加方便和快捷的管理mysql数据库。或者在数据库发生故障的时候，更加敏捷的定位问题。这是一个可以帮你提高工作效率的工具，仅此而已。

- mysql服务器较多，日常登录服务器操作数据库频繁的时候
- 发生故障需要迅速的通过日志和相关命令定位问题的时候
- 需要对某个数据库的基本信息监控，而不想敲重复命令的时候

该工具只有一个文件，安装非常简便，下载解压后将文件复制到系统里面赋予权限就可以执行，执行前请记得修改文件里面的配置信息，主要为连接数据库的信息，为了安全，该脚本只允许root账号执行，并且权限最好设置为700.

```
# cp my /usr/local/sbin/  
# chmod 700 /usr/local/sbin/my  
# vim /usr/local/sbin/my  
host="localhost" #mysql server  
port="3306" #mysql port  
username="myuser" #mysql username  
password="123456" #mysql password  
basedir=/data/mysql # if not define. default /usr/local/mysql
```

安装好之后，在系统界面敲入#my --help 即可看到帮助信息，里面有每个命令的使用说明和示例：

```
[root@db-2-68 sbin]# my --help
my help info:
support-site:www.ruzuojun.com  bug-report:ruzuojun@yihaodian.com
=====
--help      Print my tools help info; Command: #my --help
dba         Login mysql server; Command: #my dba
doc         Print mysql refman doc online; Command: #my doc [parameter] ; Example: #my doc 'create table'
ping        Check if mysqld is alive; Command: #my ping
listen      Print the mysql listen port; Command: #my listen
active      Print mysql processlist once ; Command: #my active
top         Print mysql processlist ever and again; Command: #my top
kill        kill session by session id ; Command: #my kill [pid parameter]; Example: my kill 20000
killall     kill all session if time > N seconds; Command: #my killall [N second parameter]; Example: my killall 60
queryavg    Print query number per seconds; Command: #my queryavg
deadlock    Print dead locks info; Command: #my deadlock
errorlog    Print last 50 line error log info; Command: #my errorlog
tailerror   Print last error log info real time; Command: #my tailerror
mycnf       Print my.cnf info; Command: #my mycnf [parameter] ;you can find my.cnf file by #mysql --verbose --help
variables   Print mysql variables info; Command: #my variables [parameter]
status      Print mysql status info; Command: #my status [parameter] ; Example: #my status locks
status-inc  Check mysql status change value; Command: #my status-inc [parameter]; Example: my status-inc Buffer
innodb      Print innodb status info; Command: #my innodb
mutex       Print innodb mutex status info; Command: #my mutex
user        Print mysql user list info; Command: #my user
grants      Print mysql user grant info; Command: #my grants [parameter] ; Example: #my grants root@localhost
history     Print mysql history file info; Command: #my history [parameter] ; Example: #my history update
slave       Print mysql slave info; Command: #my slave
delay       Print mysql slave delay to master time ever and again; Command: #my delay
exec        Execute SQL statement; Command:#my exec [SQL parameter] ; Example: #my exec 'select * from test.user'
admin       Execute mysqladmin Command; Command: #my admin [Command parameter]; Example: #my admin stop-slave
[root@db-2-68 sbin]#
```

my dba

可以快速登录mysql client.

my doc [parameter]

查看mysql在线帮助文档，需要一个参数。示例：

例如要查看创建索引的语法：**#my doc " create index"**

my ping

此命令会向mysql服务器发送一个请求，检查mysql是否存活，如果没有中断，此命令会一直Ping下去.

#my listen

查看当前服务器mysql监听端口

my active

输出mysql当前的活动进程。进程上方会统计当前活动连接数和sleep状态的连接数，只输出一次。

my top

动态的输出mysql进程信息，2秒自动刷新一次，直至退出。输出结果类似于my active.

my kill [parameter]

杀掉单个mysql连接,需要一个参数。 示例：#my kill 12345, 可结合my active使用

my killall [parameter]

批量杀掉查询超过N秒的查询.需要一个参数。 示例：#my killall 60

my queryavg

显示mysql当前查询数

#my deadlock

查看当前mysql 死锁情况

my errorlog

输出mysql最新的错误日志信息和日志的路径.默认为最新的**50**行日志

my tailerror

查询mysql时时错误日志，**1**秒刷新一次

my mycnf [parameter]

输出mysql配置文件. 参数为可选，没有参数会输出所有参数。

示例：**#my mycnf datadir**

my variables [parameter]

输出mysql变量信息. 参数为可选，没有参数会输出所有变量。

示例：**#my variables lock**

#my status [parameter]

输出mysql服务器状态信息. 参数为可选，没有参数会输出所有状态值。

示例：**# my status cache**

#my status-inc [parameter]

输出mysql状态值的变化量，参数可选，3秒刷新一次

示例：# my status-inc Queries

my innodb

输出mysql innodb存储引擎信息。

my mutex

输出innodb 互斥体信息。

my user

输出mysql服务器的所有用户列表。

#my grants [parameter]

输出某一个用户的授权信息，需要一个参数。

示例：#my grants root@localhost

#my history [parameter]

查询mysql历史命令，参数为可选。示例：**# my history drop**

my slave

输出mysql slave同步信息。

my delay

输出mysql slave延时，动态输出，1秒刷新一次。

my exec [parameter]

执行mysql命令和SQL语句，需要一个参数。

示例：**#my exec 'select * from test.user'**

#my admin [parameter]

执行mysqladmin的相关命令，其中参数为mysqladmin的参数。

示例：**#my admin stop-slave**

和Oracle数据库一样，只要有系统权限，就可以以管理员身份对数据库进行操作。这个工具也是如此，只要有root权限，我们就认为你有mysql的所有权限，可以使用此工具进行mysql操作，但是该工具对非root用户是禁止的。对系统账号和数据库账号严格划分的服务器，这个工具并不适用，因为这个工具里面写入了mysql的数据库账号密码。一旦你使用了该工具，为了安全，请参考如下建议：

- 1.在root用户下将脚本权限改为700
- 2.连接账号请设置一个本地账号，请勿用%作为主机名。
- 3.只允许root用户使用，这个是工具里的一个限制。如果你是非root用户，即使有可执行权限也无法正常允许该工具。如果你认为需要非root用户操作，那么请删除程序里的如下代码：

```
#check privileges, This script must be run as root.
#####
#####
if [ $EUID -ne 0 ]; then
    echo "[ERROR]: Please run this script with root privileges."
    exit 1
fi
```

myadmin由于由**shell**编写，只有一个文件，代码量也不多，所有功能扩展非常容易，如果你要加入自己的监控代码，请参考如下步骤：

- 1.加入自己的命令分支：在代码的**case**分支内加入自己的分支，如下所示为**delay**分支：**delay**为命令，中间写入你的命令。将你的分支写在分支的后面即可。如下图所示：
- 2.编写一个帮助信息：将帮助信息复制一行，加在帮助信息的末尾，大约在**323**行，修改为自己所加部分的描述即可。

```
'delay')
#Print mysql slave delay to master time ever and again; Command: #my delay
for((;;))
do
c=`$bindir/mysql $login_param -e "show slave status;" |wc -l`
if [ "$c" == "0" ];then
echo "Note:This mysql server is not a slave."
else
$bindir/mysql $login_param -e "show slave status\G" |grep Seconds_Behind_Master
echo "*****"
fi
sleep 1
done
;;
```

Thanks

工具在线说明和下载支持网站:

<http://www.ruzuojun.com>

<http://www.yhddba.com>

Bug report : ruzuojun@yihaodian.com