MySQL 探测检查

目 录

1. 制定说明

1.1. 适用范围

本文提供了数据库上线前所需要检查的各种项目，主要在于数据库安全和默认参数两个方面。

2. 安全检查

2.1. 启动用户检查

数据库启动用户权限设置为无法登陆，非 root 权限最佳，能够避免许多权限和数据的安全问题。

[x21@VM_0_7_centos ~]# ps -ef|grep mysqld |awk 'NR>1'| awk '{print $1}'

```
[x21@VM_0_7_centos ~]$ ps -ef|grep mysqld |awk 'NR>1'| awk '{print $1}'
x21
mysql
```

#说明
上述命令打印出当前多个数据库启动的用户名，x21 用户为个人用户，属于可登陆用户，而 mysql 属于不可登陆用户，是最佳设置。

2.2. 数据磁盘空间检查

数据磁盘检查能够保证数据库服务受制于磁盘空间，及时增大或者清理磁盘空间，在另一个方面也能保证备份脚本的稳定。

[root@VM_0_7_centos ~]# df -h |awk 'NR>1'|awk '{print $1,int($5)}'

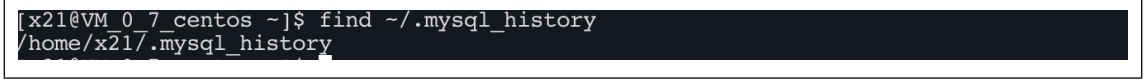
```
[x21@VM_0_7_centos ~]$ df -h |awk 'NR>1'|awk '{print $1,int($5)}'
/dev/vda1 74
devtmpfs 0
tmpfs 1
tmpfs 1
tmpfs 0
tmpfs 0
```

#说明
第一列为磁盘空间，第二列为磁盘使用率，目前设置 80 为上限。

2.3.MySQL 历史记录检查

MySQL 会将 shell 用户的客户端操作记录在家目录下的.mysql_history 文件中，上线前应该删除该文件或者注意它的权限。

```
[x21@VM_0_7_centos ~]# find ~/.mysql_history
```



2.4.客户端用户安全检查

1.检查是否使用了 root 的用户名称，应确保不使用 root 作为业务用户，以免造成安全问题。

```
mysql> SELECT 1 from mysql.user where user='root';
```

2.检查具有修改数据权限的用户，在上线前应该明确权限范围。

```
mysql> SELECT
            user, host,Update_priv,Delete_priv,Drop_priv,Grant_priv
            FROM mysql.user
            WHERE
            (Update_priv = 'Y')    OR
            (Delete_priv = 'Y')    OR
            (Drop_priv = 'Y')       OR
            (Grant_priv = 'Y') ;
```

3.检查空密码的用户。

```
mysql> SELECT User,host FROM mysql.user WHERE authentication_string='';
```

4.检查空用户名的用户。

```
mysql> SELECT user,host FROM mysql.user WHERE user = '';
```

5.检查用户可以访问 IP 范围，确保用户仅能在业务服务节点登入。

```
mysql> SELECT user, host FROM mysql.user WHERE host = '%';
```

6.检查数据库服务是否使用默认端口 3306。

```
mysql> show variables like 'port';
```

7.检查数据库是否启用了 ssl 连接，如果确保用户登入足够安全，可以关闭以提高性能。

```
mysql> show variables like 'have_ssl';
```

8.检查客户端是否具有 load 本地数据的权限，确保远程不可 load 客户端数据。

```
mysql> show variables like ' local_infile';
```

3. 数据库参数检查

数据参数检查能够做到保证业务使用 MySQL 数据服务的最佳实践，调整参数以提供一个稳定的数据库服务。

3.1. 数据库版本检查

MySQL 5.7 数据库建议使用 5.7.26 及以后的版本，MySQL 8.0 数据库建议使用 8.0.19 及以上的版本。

```
mysql> show variables like 'version';
```

3.2. 二进制日志检查

1.必须开启二进制日志

```
mysql> show variables like 'log_bin';
```

2.必须使用 row 模式

```
mysql> show variables like 'binlog_format';
```

3.建议设置过期时间

```
mysql> show variables like 'expire_logs_days';
```

4.必须设置 binlog 安全落盘，sync_binlog 为 1

```
mysql> show variables like 'sync_binlog';
```

5.建议从库也记录二进制日志到本地

```
mysql> show variables like 'log_slave_updates';
```

3.3. 通用日志检查

通用日志通常在测试环境可以用于临时获取 SQL 具体执行，上线时应确保关闭，减少性能损失。

```
mysql> show variables like 'general_log';
```

3.4. 存储引擎检查

MySQL 数据库存储引擎 InnoDB 在绝大多数情况下已经成为性能最强、适用性最好的事务引擎，所以必须默认使用 InnoDB。

```
mysql> show variables like 'InnoDB';
```

3.5. 事务隔离级别检查

InnoDB 引擎支持事务，提供四种事务隔离界别，默认隔离级别应设置为 RC，特殊时使用可重复读 RR。

```
mysql> show variables like 'transaction_isolation';
```

3.6. 查询缓存检查

查询缓存在业务 OLTP 事务类型下必须关闭。

```
mysql> show variables like 'query_cache_type';
```

3.7. 独立空间检查

1.必须开启独立表空间。

```
mysql> show variables like 'innodb_file_per_table';
```

2.必须开启独立 undo 空间。

```
mysql> show variables like 'innodb_undo_tablespaces';
```

3.设置临时表空间上限

```
mysql> show variables like 'innodb_temp_data_file_path';
```

3.8. 域名解析检查

建议关闭域名解析以降低连接性能损耗。

```
mysql> show variables like ' skip_name_resolve';
```

3.9. 慢 SQL 阈值定义检查

MySQL 慢查询定义阈值为 10 秒，建议改为 2 秒以统计出慢查询。

```
mysql> show variables like 'long_query_time';
```

3.10. 外键检查

外键在使用不当时常常引起性能问题，建议去除外键。

```
mysql>SELECT table_name,
      column_name,
      constraint_name,
      REFERENCED_TABLE_NAME,
      REFERENCED_COLUMN_NAME
from INFORMATION_SCHEMA.KEY_COLUMN_USAGE
where REFERENCED_TABLE_NAME is not null;
```

4. 附录：实施 SHELL 脚本

```
#!bin/bash
USER='root'
PORT=3306
HOST=127.0.0.1
export MYSQL_PWD='!qazxsw@'
mysqlconnectstring="mysql -u${USER} -h${HOST} -P${PORT} "
number=0
#########################################################################
#主函数
funcmain(){
if [ `${mysqlconnectstring} -e "select 1;" | awk 'NR>1' |awk '{print $1}'` ]
      then

         check_mysqld_user
         check_mysql_user_exist_root
         check_mysql_user_modifydata
         check_mysql_user_0password
         check_mysql_user_0name
         check_mysql_user_anyip
         check_mysqld_port
         check_mysql_connect_ssl
         check_mysqld_datadir
         check_mysql_historyc
         check_mysqld_binlog
    #    check_mysqld_errorlog
         check_mysqld_generalog
    #    check_mysqld_waring
         check_mysql_local_infile
         check_mysql_version
         check_mysqld_dengine
         check_mysqld_trxiso
         check_mysqld_binlogformat
         check_mysqld_binlogexpire
         check_mysqld_qcache
         check_mysqld_pidb
         check_mysqld_pundo
         check_mysqld_ibtmpmax
         check_mysqld_nameresolve
         check_mysqld_slowqt
         check_mysqld_binlogsync
         check_mysqld_log_slave_updates
```

```
fi
}

#########################################################################
#检查函数
check_mysqld_user(){
ps -ef|grep mysqld |awk 'NR>1'| awk '{print $1}'| while read line
do
        if test $line = 'root'
                then
                   let number++
                   echo $number.'MySQL 启动用户须禁止使用 root'
                   echo $number > tmp.pid
        fi
done
if [ -f tmp.pid ]
then
        number=`cat tmp.pid`
        rm -f tmp.pid
fi
}

check_mysql_user_exist_root(){
tag=`${mysqlconnectstring} -e "SELECT count(*) from mysql.user where user='root';" | awk 'NR>1' `
        if [ $tag -gt 0 ]
                then
                   let number++
                   echo $number.'MySQL 客户端用户须禁止使用 root 名称'
        fi
}

check_mysql_user_modifydata(){
tag=`${mysqlconnectstring} -e "SELECT count(*) FROM mysql.user WHERE
        (Update_priv = 'Y')    OR
        (Delete_priv = 'Y')    OR
        (Drop_priv = 'Y')       OR
        (Grant_priv = 'Y') ;" | awk 'NR>1' `
        if [ $tag -gt 0 ]
                then
                   let number++
                   echo $number.'请检查具有修改数据权限的用户!'
                   ${mysqlconnectstring} -e "SELECT
                   user, host,Update_priv,Delete_priv,Drop_priv,Grant_priv
```

```
            FROM mysql.user
            WHERE
            (Update_priv = 'Y')    OR
            (Delete_priv = 'Y')    OR
            (Drop_priv = 'Y')      OR
            (Grant_priv = 'Y') ;"
    fi
}

check_mysql_user_0password(){
tag=`${mysqlconnectstring} -e "SELECT count(*) FROM mysql.user WHERE authentication_string="" | awk 'NR>1' `
    if [ $tag -gt 0 ]
        then
          let number++
          echo $number.'MySQL 客户端用户须禁止使用空密码'
          ${mysqlconnectstring} -e "SELECT User,host
          FROM mysql.user
          WHERE authentication_string="";"
    fi
}
check_mysql_user_0name(){
tag=`${mysqlconnectstring} -e "SELECT count(*) FROM mysql.user WHERE user = '';" | awk 'NR>1' `
    if [ $tag -gt 0 ]
        then
          let number++
          echo $number.'MySQL 客户端用户须禁止使用空名'
          ${mysqlconnectstring} -e "SELECT user,host FROM mysql.user WHERE user = '';"
    fi
}

check_mysql_user_anyip(){
tag=`${mysqlconnectstring} -e "SELECT count(*) FROM mysql.user WHERE host = '%';" | awk 'NR>1' `
    if [ $tag -gt 0 ]
        then
          let number++
          echo $number.'MySQL 数据库须禁止用户从任意 IP 访问'
          ${mysqlconnectstring} -e "SELECT user, host
          FROM mysql.user
          WHERE host = '%';"
    fi
}
```

```
check_mysqld_port(){
if [ `show_variable port` -eq 3306 ]
        then
           let number++
           echo $number.'MySQL 数据库服务须禁止使用默认端口 3306'
fi
}

check_mysql_connect_ssl(){
if [ `show_variable have_ssl` = 'DISABLED' ]
        then
           echo -n ''
        else
           let number++
           echo $number.'MySQL 客户端连接使用 ssl'
fi
}

check_mysqld_datadir(){
df -h |awk 'NR>1'|awk '{print $1,int($5)}' | while read disk userate
do
        if [ $userate -gt 80 ]
                then
                let number++
                echo $number.$disk'磁盘空间使用超过 80%'
                echo $number > tmp.pid
        fi
done
if [ -f tmp.pid ]
        number=`cat tmp.pid`
        rm -f tmp.pid
}

check_mysql_historyc(){
        if [ –f ~/.mysql_history ]
                then
                let number++
                echo $number.'MySQL 历史命令文件存在，请注意清除'
        fi
}

check_mysqld_binlog(){
        if [ `show_variable log_bin` = 'OFF' ]
```

```
            then
            let number++
            echo $number.'MySQL 须开启二进制日志'
        fi
}

check_mysqld_errorlog(){
tag=$(grep 'ERROR' `${mysqlconnectstring} -e "show global variables like 'log_error';" | awk 'NR>1' |awk '{print $2}'` | sed -n "/`date -d "5 day ago" +"%Y-%m-%d"`/","/`date +"%Y-%m-%d"`/p" |wc -l)
        if [ $tag -gt 0 ]
            then
            let number++
            echo $number.'MySQL 服务器存在近 5 天的错误记录，共'$tag'个，"请处理'
        fi

}

check_mysqld_generalog(){
        if [ `show_variable general_log` = 'ON' ]
            then
            let number++
            echo $number.'MySQL 须关闭通用日志'
        fi
}

check_mysqld_waring(){
        if [ `show_variable log_warnings` -ne 2 ]
            then
            let number++
            echo $number.'MySQL 须设置告警等级 log_warnings 为 2'
        fi
}

check_mysql_local_infile(){
        if [ `show_variable local_infile` = 'ON' ]
            then
            let number++
            echo $number.'MySQL 须禁止远程数据 load 到本地'
        fi
}

check_mysql_version(){
        if [[ `show_variable version` < '5.7.26' ]]
```

```
            then
            let number++
            echo $number.'MySQL 版本建议使用 5.7.26 以上'
            elif [[ `show_variable version` > '8.0.0' ]] && [[ `show_variable version` < '8.0.19' ]]
              then
              let number++
              echo $number.'MySQL 版本建议使用 8.0.19 以上'

    fi
}

check_mysqld_dengine(){
    if [ `show_variable default_storage_engine` != 'InnoDB' ]
        then
        let number++
        echo $number.'MySQL 须使用 InnoDB 引擎'
    fi
}

check_mysqld_trxiso(){
    if [ `show_variable transaction_isolation` != 'READ-COMMITTED' ]
        then
        let number++
        echo $number.'MySQL 事务隔离级别须使用 READ-COMMITTED'
    fi
}

check_mysqld_binlogformat(){
    if [ `show_variable binlog_format` != 'ROW' ]
        then
        let number++
        echo $number.'MySQL 二进制日志格式须设置为 ROW 模式'
    fi
}

check_mysqld_binlogexpire(){
    if [ `show_variable expire_logs_days` -eq 0 ]
        then
        let number++
        echo $number.'MySQL 二进制日志须设置过期时间'
    fi

}
```

```
check_mysqld_qcache(){
    if [ `show_variable query_cache_type` = 'ON' ]
        then
        let number++
        echo $number.'MySQL 查询缓存在无特殊情况须关闭'
    fi
}

check_mysqld_pidb(){
    if [ `show_variable innodb_file_per_table` = 'OFF' ]
        then
        let number++
        echo $number.'MySQL 服务器须开启独立表空间'
    fi
}

check_mysqld_pundo(){
    if [ `show_variable innodb_undo_tablespaces` -eq 0 ]
        then
        let number++
        echo $number.'MySQL 服务器须开启独立 undo 空间'
    fi

}
check_mysqld_ibtmpmax(){
    if [ `show_variable innodb_temp_data_file_path` = 'ibtmp1:12M:autoextend' ]
        then
        let number++
        echo $number.'MySQL 服务器须设置 ibtmp 上限'
    fi
}

check_mysqld_nameresolve(){
    if [ `show_variable skip_name_resolve` = 'OFF' ]
        then
        let number++
        echo $number.'MySQL 服务器须启用 skip_name_resolve'
    fi

}
check_mysqld_slowqt(){
    if [ `show_variable long_query_time` -gt 2 ]
        then
        let number++
```

```
            echo $number.'MySQL 慢查询定义建议设为 2 秒'
        fi
}
check_mysqld_tables_fk(){
tag=`${mysqlconnectstring} -e "SELECT count(*) from INFORMATION_SCHEMA.KEY_COLUMN_USAGE where REFERENCED_TABLE_NAME is not null;" | awk 'NR>1' `
        if [ $tag -gt 0 ]
            then
              let number++
              echo $number.'MySQL 外键建议不使用'
              ${mysqlconnectstring} -e "SELECT table_name,column_name,constraint_name,REFERENCED_TABLE_NAME,REFERENCED_COLUMN_NAME
              from INFORMATION_SCHEMA.KEY_COLUMN_USAGE
              where REFERENCED_TABLE_NAME is not null;"
        fi
}
check_mysqld_binlogsync(){
        if [ `show_variable sync_binlog` -ne 1 ]
            then
            let number++
            echo $number.'MySQL 服务器二进制日志刷盘须设置为 1'
        fi
}

check_mysqld_log_slave_updates(){
        if [ `show_variable log_slave_updates` = 'OFF' ]
            then
            let number++
            echo $number.'MySQL 从库须记录二进制日志到本地'
        fi
}

show_variable(){
${mysqlconnectstring} -e "show global variables like '${1}';" | awk 'NR>1' |awk '{printf $2}'
}

show_status(){
${mysqlconnectstring} -e "show global status like '${1}';" | awk 'NR>1' |awk '{printf "%-20s:%30s\n",$1,$2}'
}

#执行
```

funcmain