开发巡检数据库备份的SHELL脚本 backupCheck.sh

```
[root@vm-recovery-mysql01 rc.d]# cat backupCheck.sh
#!/bin/bash
# Define log
set -x

TIMESTAMP=`date +%Y-%m-%d_%H:%m:%S`
LOG=/var/log/backupCheck/backupCheck_$TIMESTAMP.log
LOG_PATH="/var/log/autocheck/autoCheck_${DAY_TIME%_*}.log"
BACKUPNOPERDAY=14
HOST=`hostname`
BACKUPDIRFORTAOPU='/remote_backup'
BACKUPDIRFORJIULOU='/mysql'
BACKUPDIR='/remote_backup'

# checkBackupeckBackup1IPfunction checkBackup(){
    echo -e "\n\n\n\n\n" >> $LOG
    echo "executing backup check at $TIMESTAMP " >> $LOG
    dir1=($(ls | grep mysqlbackup_))
    count=0             # count=0 eckBackup1

                                                 read -p "Please enter an IP " IP
    if [ x$IP != x ]; then
        i='mysqlbackup_'$IP''
        echo checking backup for $i >> $LOG
        checkBackup1;
        echo -e "\n\n\n\n\n" >> $LOG
    else
        for i in ${dir1[@]};
        do
            checkBackup1;
        done
        echo -e "\n\n\n\n\n" >> $LOG
    fi
}

        #       read -p "Please enter a date " date
        #       if [ -z $date ]; then
        #               j=$date
        #               checkBackup2;
        #       else
        #       fi
# 90%

      function checkSpace()
{
        spaceused=`df -h | grep backup | awk '{print $5}'`
        if [ $spaceused  -gt 90% ]; then
                echo Space need to be cleaned up! >> $LOG
        else
                echo Space is normal! >> $LOG
        fi
}

# checkBackup1eckBackup2unction checkBackup1() {
        if [ -d $BACKUPDIR/$i ]; then
```

```
            echo -e "\n" >> $LOG
            echo "Excuting backupCheck for $i" >> $LOG
            cd $BACKUPDIRFORTAOPU/$i
            dir2=($(ls | grep ibm-))
            dir2no=`ls | grep ibm | wc -l`
            datemin=`ls -l | awk '{print $9}' | awk -F '_' '{print $2}' | awk -F '-' '{print $1}' | sed '/^$/d' | sort | head -n 1`
            datemax=`ls -l | awk '{print $9}' | awk -F '_' '{print $2}' | awk -F '-' '{print $1}' | sed '/^$/d' | sort | tail -n 1`
            startDate=`date -d "${datemin}" +%s`
            endDate=`date -d "${datemax}" +%s`
            stampDiff=`expr $endDate - $startDate`
            dayDiff=`expr $stampDiff / 86400`
            space=`du -s $BACKUPDIR/$i | awk '{print $1}'`
            if [ "$space" != "0" ]; then
                if [ "$dayDiff" != "0" ]; then
                        if [ $dir2no -lt $dayDiff+1 ];then
                                echo Attention! Your backup for $i is missing for one or more days! >> $LOG
                        fi
                else echo attention! You only have one day backup for $i. >> $LOG
                fi
                if [ y$date = y ] && [ $count -eq 0 ]; then
                        read -p "Please enter a date" date
                        count=1         # count=1 heckBackup1   fi
                if [ x$date != x ]; then
                        j=`ls | grep $date`
                        echo Executing backup check for $j >> $LOG
                        checkBackup2;
                else
                for j in ${dir2[@]};
                do
                        #echo Executing backup check for $j >> $LOG
                        checkBackup2;
                done
                fi
            else echo Error! Directory $i is empty! >> $LOG
            fi
            else echo Error! $BACKUPDIR/$i is not a directory
        fi
}

# checkBackup2 unction checkBackup2()
{
        if [ -d $BACKUPDIR/$i/$j ]; then
                cd $BACKUPDIRFORTAOPU/$i/$j
                numtar=`ls | grep -E "(gz|tar)" | wc -l`
                numsuccess=`cat $BACKUPDIR/$i/$j/*.log | grep success | wc -l`
                numfail=`cat $BACKUPDIR/$i/$j/*.log | grep fail | wc -l`
                if [ $numfail -eq 0 ]; then
                        if [ $numtar -lt $BACKUPNOPERDAY ]; then
                                if [ $numtar -lt $numsuccess ]; then
                                        echo Attention! Your backup for $i is missing for day $j! >> $LOG
                                elif [ $numtar -eq $numsuccess ]; then
                                        echo Attention! Check why some of your backup is not taking place. >> $LOG
                                else
                                        echo Warning! backup infomation for $i $j is : Tar package No. is $numtar, Success No. is $numsuccess. >> $LOG
                                fi
                        else
                                for k in `du -ah . | grep tar | awk '{ print $1 }'`;
                                do
                                        if [ "$k" = "0" ]; then
                                                echo Attention! Your backup $j for $i is fake because there is a tar package whose size is 0 >> $LOG
                                        fi
                                done
                        fi
                else echo Not good! Your backup $j for $i  has failed at sometime. Please check for more details.
                fi
        else echo Error! $BACKUPDIR/$i/$j is not a directory!
        fi
}

#unction CheckIPAddr()
{
echo $1|grep "^[0-9]\{1,3\}\.\([0-9]\{1,3\}\.\)\{2\}[0-9]\{1,3\}$" > /dev/null;
#IP

            if [ $? -ne 0 ]
        then
                return 1
        fi
        ipaddr=$1
        a=`echo $ipaddr|awk -F . '{print $1}'`  #        b=`echo $ipaddr|awk -F . '{print $2}'`
        c=`echo $ipaddr|awk -F . '{print $3}'`
        d=`echo $ipaddr|awk -F . '{print $4}'`
        for num in $a $b $c $d
        do
                if [ $num -gt 255 ] || [ $num -lt 0 ]    #-255                then
                        return 1
                fi
        done
                return 0
}
if [ $# -gt 1 ];then             #        echo "Usage: $0 ipaddress."
        exit
elif [ $# -eq 1 ]; then
CheckIPAddr $1
fi

#
#function checkDate()
#{
#}

checkSpace;
cd $BACKUPDIR
checkBackup;




#1qlbackup_... C 0
#2qlbackup_...-...datemin=datemax
```