链滴

LINUX 脚本使用实战，持续更新

作者：cuijianzhe
原文链接：https://ld246.com/article/1552705279634
来源网站：链滴

1.自动备份系统文件

```
#!/bin/bash
#cd /data/backup
#find ./ -name "*.tar.gz" -mtime +3 -exec rm -f {} \;
#tar zcf /data/backup/`date +%F`_data.tar.gz /data
cd /data/rsyslog
tar zcf /data/rsyslog/`date +%F`_rsyslog.tar.gz /data/rsyslog    ##备份日志文件
find ./ -name "*.tar.gz" -mtime +2 -exec rm -rf {} \;        ##筛选出2天以前的文件并且删除
```

1.1统计文本出现字符的频率：

```
$ awk -F' ' '{for(i=1;i<=NF;i=i+1){print $i}}' word.txt |sort|uniq -c|sort -nr|awk -F' ' '{printf("%s
s\n",$2,$1)}'

the 4
is 3
sunny 2
day 1
```

2.URL检测脚本

```
[root@zabbix shell]# vim checkurl.sh
#!/bin/sh
[ -f /etc/init.d/functions ]&& . /etc/init.d/functions

usages(){
      echo "USAGE:$0 url"
      exit 1

}
RETVAL=0
CheckUrl(){
   wget -T 10 --spider -t 2 $1 &>/dev/null
   RETVAL=$?
   if [ $RETVAL -eq 0 ];then
        action "$1 url" /bin/true
   else
       action "$1 url" /bin/false

   fi
   return $RETVAL
}
main(){
  if [ $# -ne 1 ];then
     usage
  fi

  CheckUrl $1

}
main $*
```

结果测试

```
[root@zabbix shell]# sh checkurl.sh www.baidu.com www.baidu.com url
    [ 确定 ]
```

3.expect脚本

```
#!/usr/bin/expect -f
set ip [lindex $argv 0]
#set password [lindex $argv 1]
set timeout -1
spawn ssh cuijianzhe@$ip

expect {
        "*assword" {send "qwe*123456\r";}
        "(yes/no)" {send "yes\r";exp_continue}
}
expect "*>" {send "ftp 192.168.51.202\r"}
send "ftp01\r"
send "598941324\r"
send "put vrpcfg.zip\r"
send "quit\r"
send "quit\r"

expect eof
```

4.zabbix数据库备份脚本

```
#!/bin/bash
#-------------------------------------------------------------------------------#
  #Shell Command For Backup MySQL Databaes Everyday Automatically By Crontab
  #Author     : cuijianzhe
  #Email      : 598941324@qq.com
  #Create date: 2019-3-19
#-------------------------------------------------------------------------------#
#-----------------------Back up MySQL every day---------------------------------#
#-------------------------------------------------------------------------------#
User=root                                   #数据库用户
Password=XXXXXXXXX                                  #用户登录密码
DataBase=zabbix                                 #数据库名
Backup_DIR=/data/mysql_bak/mysqlbackup/      #存放备份数据库文件路径
LogFile_DIR=/data/mysql_bak/mysqlbackup_log/ #存放备份数据库日志路径
LogFile="$LogFile_DIR"mysql_backup.log
Date1=`date +%Y-%m-%d`
start_time=`date +'%Y-%m-%d %H:%M:%S'`                  #获取时间
DumpFile="$Date1"-"$DataBase".sql
Archive="$Date1"-"$DataBase"_sql.tar.gz       #打包后名称
SaveTime=2
DelFile=`find $Backup_DIR -type f -mtime +$SaveTime -exec ls {} \;`  #查找大于2天的备份文件
start_time=`date +'%Y-%m-%d %H:%M:%S'`
systemctl stop zabbix-server.service
-------------------------------------------------------------------------------#
  if [ ! -d $Backup_DIR ];                    #判断路径是否存在，若没有则创建
```

```
    then
      mkdir -p "$Backup_DIR"
  fi
  [ ! -d $LogFile_DIR ] && mkdir -p $LogFile_DIR
  if [ ! -f "$LogFile" ];
    then
      touch $LogFile
  fi
#-------------------------------------------------------------------------------#
#  echo -e "\n" >> $LogFile
  echo "#-------------------------------------------------------#" >> $LogFile
  echo "#------------Backup Date:$start_time !!!--------------#" >> $LogFile
  echo "#-------------------------------------------------------#" >> $LogFile

#-------------------------------------------------------------------------------#
  cd $Backup_DIR
  mysqldump -u$User -p$Password $DataBase > $DumpFile   #mysqldump备份
  if [[ $? == 0 ]];
    then
      tar czvf $Archive $DumpFile >> $LogFile 2>&1                #判断是否成功，成功则打包，未
功则些入到日志文件
      echo -e "$User :$Archive Backup Successul !!!" >> $LogFile
      rm -rf $DumpFile                                      #打包后删除sql文件
  else
      echo -e "$User :$Archive Backup Fail !!!" >> $LogFile
  fi
#-------------------------------------------------------------------------------#
#--------------------Delete MySQL backup 2 days ago----------------------------#
#-------------------------------------------------------------------------------#
#  echo -e "\n" >> $LogFile
  echo "#-------------------------------------------------------#" >> $LogFile
  echo "#------------Delete Date:$start_time !!!--------------#" >> $LogFile
  echo "#-------------------------------------------------------#" >> $LogFile
#-------------------------------------------------------------------------------#
  if [[ $? == 0 ]];
    then
      for delfile in ${DelFile}
        do
          rm -rf $delfile                        #删除2天前备份
      done
    echo -e "$User :Successfully deleted 3 days before backup !!!" >> $LogFile
  else
    echo -e "$User :Unsuccessful deletion of backup 2 days ago !!!" >> $LogFile
  fi
systemctl start zabbix-server.service
#------------------------------打印备份所需时间-------------------------------------------#
end_time=`date +'%Y-%m-%d %H:%M:%S'`
start_seconds=$(date --date="$start_time" +%s);
end_seconds=$(date --date="$end_time" +%s);
time_total=$((end_seconds-start_seconds))

if [[ $time_total -lt 60 ]];then
   echo -e  "备份所用时间为："$(($time_total))"s"  >>$LogFile
else
```

```
  echo -e "备份所用时间为：$(($time_total/60|bc))minutes" >>$LogFile
  echo -e "备份开始时间为: $start_time ,结束时间为：$end_time" >> $LogFile
fi
```

5.系统日志文件清除

```
#!/bin/bash
#clear /var/log/message
#确定当前是不是root用户
if [ $USER != "root" ];then
   echo "你必须使用root用户才能执行这个脚本"
   exit 10
fi
#判断日志文件在不在
if [ ! -f /var/log/messages ];then
   echo "文件不存在"
   exit 11
fi
#保留最近100行的日志内容
tail -100 /var/log/messages > /var/log/mesg.tmp
#日志清理
>/var/log/messages
cat /var/log/mesg.tmp >> /var/log/messages
mv /var/log/mesg.tmp /var/log/messages
echo "Logs clean up"
```

6.nginx日志切割脚本

```
#!/bin/bash
#!/bin/bash
date=$(date +%F -d -1day)
log_dir=/usr/local/nginx/logs
cd $log_dir
if [ ! -d cut  ]; then
       mkdir cut
fi

if [ ! -d logs_history ]; then
       mkdir logs_history
fi

mv access.log cut/access_$(date +%F -d -1day).log
mv error.log cut/error_$(date +%F -d -1day).log
/usr/local/nginx/sbin/nginx -s reload
tar  -zcvf cut/$date.tar.gz cut/
rm -rf cut/access* && rm -rf cut/error*
mv cut/$date.tar.gz logs_history
cat >>/var/spool/cron/root<<eof
00 00 * * * /bin/sh /usr/local/nginx/logs/cut_nginx_log.sh >/dev/null 2>&1
eof
find -type f -mtime +5 | xargs rm -rf
```

目录下的日志切割（传入值）

```
#!/bin/sh

function rotate() {
  logs_path= /alidata/logs/paas-cmdb-backend/uwsgi_emperor/$1

  echo Rotating Log: $1
  cp ${logs_path} ${logs_path}_$(date -d "yesterday" +"%Y%m%d").log > ${logs_path}
  rm -f ${logs_path}_$(date -d "7 days ago" +"%Y%m%d").log
}

for i in $*
do
  rotate $i
done
```

7.通过snmp时间间隔计算网络设备带宽使用百分比

```
#!/bin/bash
RX_pre=$(snmpwalk -v 2c -c limi@2018 10.200.0.1 1.3.6.1.2.1.31.1.1.1.6.6 | awk '{print $4}' && sleep 1)
RX_next=$(snmpwalk -v 2c -c limi@2018 10.200.0.1 1.3.6.1.2.1.31.1.1.1.6.6 | awk '{print $4}')

RX=$((${RX_next}-${RX_pre}))
awk  'BEGIN{printf"%0.2f",('${RX}'/'20971520')*100}'
```

另一种计算方式：

```
#!/bin/bash

RX_pre=$(snmpwalk -v 2c -c limi@2018 10.200.0.1 1.3.6.1.2.1.31.1.1.1.10.6 | awk '{print $4}' & sleep 1)
RX_next=$(snmpwalk -v 2c -c limi@2018 10.200.0.1 1.3.6.1.2.1.31.1.1.1.10.6 | awk '{print $4}')
RX=$((${RX_next}-${RX_pre}))
echo "scale=2;$RX/20971520*100"|bc
```

此脚本：带宽20M专线，转化bps为20x1024x1024=20971520bps

通过snmp监测L2TP登陆用户名称和ip：

```
#!/bin/bash

date=`date +'%Y-%m-%d %H:%M:%S'`
echo -e "-------------------------------------------------------------------------------------------------------------------"
echo -e "-------------------------------------$date-----------------------------------------------------------------"
user=$(snmpwalk -v 2c -c limi@2018 192.168.100.1 1.3.6.1.4.1.2011.6.122.2.4.1.5 |awk '{print 4}' | sed 's/"//g' | xargs )
userip=$(snmpwalk -v 2c -c limi@2018 192.168.100.1 1.3.6.1.4.1.2011.6.122.2.3.1.5 |awk '{print $4}' | xargs)
echo -e "当前登录用户名：$user"
echo -e  "当前登录IP：” $userip"
echo -e "==========================================================================================================================="
```

测试：

```
[root@zabbix ~]# /shell/user_monitor.sh

--------------------------------------------------------------------------------------------------------------------
-------------------------------------2019-04-21 18:54:27-------------------------------------------------------------
当前登录用户名：huamao@limi xutaoran@limikeji.com cuijianzhe@limikeji.com
当前登录IP：” 103.219.187.194 111.194.46.165 103.219.187.194
==========================================================================================================
```

8.zabbix自动发现之拼接json格式

```
#!/bin/bash

id=$(snmpwalk -v 2c -c limi@2018  10.200.252.8 1.3.6.1.4.1.2011.6.139.13.3.10.1.5 | cut -f1 -d
=" | cut -f10 -d ".")
id_array=(${id})
sum=$(snmpwalk -v 2c -c limi@2018  10.200.252.8 1.3.6.1.4.1.2011.6.139.13.3.10.1.5 | awk  '{pr
nt $4}' | sed 's/"//g' | wc -l)
name=$(snmpwalk -v 2c -c limi@2018  10.200.252.8 enterprises.2011.6.139.13.3.10.1.5 | awk  '
print $4}' | sed 's/"//g')
name_array=($name)

printf '{\"data\":[ '
for ((i=0;i<$sum;i++))
do
    printf "{\"{#APID}\":\"${id_array[$i]}\",\"{#APNAME}\":\"${name_array[$i]}\" }"

    if [ $i -lt $[ $sum-1 ] ];then
        printf ','
        fi
done
printf " ]}"
```

结果测试：

```
sh /usr/lib/zabbix/externalscripts/apdiscovery.sh

{"data":[ {"{#APID}":"0","{#APNAME}":"446a-2e13-01e0" },{"{#APID}":"1","{#APNAME}":"446a-2
13-0900" },{"{#APID}":"2","{#APNAME}":"446a-2e13-01c0" },{"{#APID}":"3","{#APNAME}":"446a
2e13-01a0" },{"{#APID}":"4","{#APNAME}":"446a-2e13-0260" },{"{#APID}":"5","{#APNAME}":"44
a-2e13-1220" },{"{#APID}":"6","{#APNAME}":"446a-2e17-db80" },{"{#APID}":"7","{#APNAME}":"
46a-2e17-dac0" },{"{#APID}":"8","{#APNAME}":"446a-2e17-db00" },{"{#APID}":"9","{#APNAME}"
"3F_DONG" },{"{#APID}":"10","{#APNAME}":"446a-2e17-db40" },{"{#APID}":"11","{#APNAME}":
446a-2e17-da60" },{"{#APID}":"12","{#APNAME}":"446a-2e13-2660" },{"{#APID}":"13","{#APN
ME}":"446a-2e13-2680" },{"{#APID}":"14","{#APNAME}":"446a-2e13-0e20" },{"{#APID}":"15","{#
PNAME}":"446a-2e13-25e0" },{"{#APID}":"16","{#APNAME}":"446a-2e13-2620" },{"{#APID}":"17
,"{#APNAME}":"446a-2e13-0160" },{"{#APID}":"18","{#APNAME}":"446a-2e13-0980" },{"{#APID}
:"19","{#APNAME}":"446a-2e13-2580" },{"{#APID}":"20","{#APNAME}":"446a-2e13-18e0" },{"{#
PID}":"21","{#APNAME}":"446a-2e13-2600" },{"{#APID}":"22","{#APNAME}":"446a-2e20-6780" }
{"{#APID}":"23","{#APNAME}":"446a-2e20-6760" } ]}
```

可用python tools判断是否是json格式。

```
$ sh /usr/lib/zabbix/externalscripts/apdiscovery.sh | python -m json.tool

{
    "data": [
        {
            "{#APID}": "0",
            "{#APNAME}": "446a-2e13-01e0"
        },
        {
            "{#APID}": "1",
            "{#APNAME}": "446a-2e13-0900"
        },
    ]
}
```

8.1 python3写法json

```
#!/bin/env python3
import json
import os
import sys
import subprocess
def discovery():
    CMD_name = '''snmpwalk -v 2c -c limi@2018  10.200.250.5 enterprises.2011.6.139.13.3.10.1
5 | awk  '{print $4}' | sed 's/"//g' '''
    CMD_id = 'snmpwalk -v 2c -c limi@2018  10.200.250.5 1.3.6.1.4.1.2011.6.139.13.3.10.1.5 | cu
 -f1 -d "=" | cut -f10 -d "."'
    ap_id = subprocess.getoutput(CMD_id)
    ap_name = subprocess.getoutput(CMD_name)
   #print(ap_id)
    id_list = ap_id.split("\n") #把AP的id每行数据添加到列表
    name_list = ap_name.split("\n")
    AP_list = list(zip(id_list,name_list))
    ap_dict = {}
    for v in AP_list:
        ap_dict[v[0]] = v[1]
    return ap_dict

#格式化成适合zabbix lld的json数据
if __name__ == "__main__":
    ap_value = discovery()
    ap_list = []
    for key in ap_value:
        ap_list += [{'{#APID}':key,'{#APNAME}':ap_value[key]}]
    #print(ap_list)
    print(json.dumps({'data':ap_list},sort_keys=True,indent=4,separators=(',',':')))
```

9. zabbix监控 端口自动发现

```
#!/bin/env python3
```

```
import subprocess
import json
import re

def PortList():
    CMD = "sudo netstat -pntl | awk '{print $4,$7}'|grep  [0-9] |egrep -vw '%s'"
    Result_str = subprocess.getoutput(CMD)
   #print(Result_str)
    tmp_list = Result_str.split("\n")
    #print(tmp_list)
    port_dict = {}
    for line in tmp_list:
       # print(line)
        PORT_REG = re.search(r"(127.0.0.1:|:::|0.0.0.0:)(\d+).+\d+/(\S+)",line)
#         print(PORT_REG)
        if PORT_REG is not None:
            match_line = (PORT_REG.groups())
            port_dict [ match_line[1]] = match_line[2]
    return port_dict

if __name__ == "__main__":
    Results = PortList()
    ports = []
    for key in Results:
        ports += [{'{#PNAME}':key,'{#PPORT}':Results[key]}]
    print(json.dumps({'data':ports},sort_keys=True,indent=4,separators=(',',':')))
```

测试：

```
$ python3 port_discovery.py
{
    "data":[
        {
            "{#PNAME}":"3306",
            "{#PPORT}":"mysqld"
        },
        {
            "{#PNAME}":"139",
            "{#PPORT}":"smbd"
        },
        {
            "{#PNAME}":"1617",
            "{#PPORT}":"rsyslogd"
        },
        {
            "{#PNAME}":"5203",
            "{#PPORT}":"sshd"
        },
        {
```

端口自动发现之shell脚本：

```
#!/bin/bash
portarray=(`sudo netstat -tnlp|egrep -i "$1"|awk {'print $4'}|awk -F':' '{if ($NF~/^[0-9]*$/) prin
$NF}'|sort|uniq`)
length=${#portarray[@]}
printf "{\n"
printf '\t'"\"data\":["
for ((i=0;i<$length;i++))
```

```
    do
        printf '\n\t\t{'
    printf "\"{#TCP_PORT}\":\"${portarray[$i]}\"}"
    if [ $i -lt $[$length-1] ];then
    printf ','
    fi
  done
printf "\n\t]\n"
printf "}\n"
```

监控磁盘信息：

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-

import sys
import commands


# firmware_state = ["Failed",
#                   "Online, Spun Up",
#                   "Online, Spun Down",
#                   "Unconfigured(bad)",
#                   "Unconfigured(good), Spun down",
#                   "Hotspare, Spun down",
#                   "Hotspare, Spun up",
#                   "not Online",
#                   "JBOD"]


(_, result) = commands.getstatusoutput("/opt/MegaRAID/MegaCli/MegaCli64 -PDList -aALL |
rep 'Firmware state' | awk -F ':' '{print $2}' | sed 's/^[ \t]*//g'")

if result == '':
    sys.exit(1)

result = result.split('\n')
result = list(set(result))

if ("Failed" in result) or ("Online, Spun Down" in result) or ("Unconfigured(bad)" in result) or ("
nconfigured(good), Spun down" in result) or ("Hotspare, Spun down" in result) or ("not Onlin
" in result):
    print 100
else:
    print 200
```

10. 查看nginx流量

```
#!/bin/bash

exec < access.log

while read line
```

```
do
        i=`echo $line|awk '{print $10}'`
        expr $1 + 1 &>/dev/null
        if [ $? -ne 0 ];then
           continue
        fi
        ((sum+=i))

done
[ -n "$sum" ] && echo $su
```

11. nmap扫描存活主机以及端口

```
#!/bin/bash

ip="192.168.51.200-214"
CMD="nmap -sP"
CMD2="nmap -sS"
CMD3="nmap"
funSecond(){
        $CMD $ip|awk '/Nmap scan report for/ {print $NF}'
}

funThird(){
        $CMD2 $ip|grep "Nmap scan report for"|awk -F "[ ()]+" '{print $6}'
}

Port(){

        $CMD3 $ip -p1-65535|grep "Nmap scan report for"|awk -F "[ ()]+" '{print $6}' >/tmp/host
ist.txt
        hosts=`cat /tmp/hostlist.txt|xargs`
        for i in ${hosts[*]}
                do
                  port=`nmap $i |awk '{print $1}'| grep "/tcp"| awk -F "/" '{print $1}'|xargs`
                  echo  "主机ip：$i 对应监听端口：  $port" >/tmp/ports.txt
        done
}


#funSecond
#funThird
Port
```

遍历目录里面的文件并操作

```
#! /bin/bash
function read_dir(){
for file in `ls $1` #注意此处这是两个反引号，表示运行系统命令
do
 if [ -d $1"/"$file ] #注意此处之间一定要加上空格，否则会报错
 then
  read_dir $1"/"$file
```

```
 else
  echo $1"/"$file #在此处处理文件即可
  sed -ri "s#(.*)(proxy_pass http://)(.*)(-limikeji;)#\1\2\3.limikeji.com;#g" $1"/"$file
  sed -ri "s#(.*)(proxy_pass http://)(.*)(-limixuexi;)#\1\2\3.limixuexi.com;#g" $1"/"$file
  sed -ri "s#(.*)(-limikeji \{)#\1.limikeji.com {#g" $1"/"$file
  sed -ri "s#(.*)(-limixuexi \{)#\1.limixuexi.com {#g" $1"/"$file
 fi
done
 }
 #读取第一个参数
read_dir $1
```