Linux常用脚本集合

## 批量添加删除用户

|  |
| --- |
| ###添加用户  #bin/bash  read -p "please input user name:" -t 30 name  read -p "please input the number of users :" -t 30 num  read -p "please input the password of users:" -t 30 pass  if [ ! -z "$name" -a ! -z "$num" -a ! -z "$pass" ];then  y=$(echo $num|sed 's/[0-9]//g')  if [ -z "$y" ];then  for((i=1;i<=$num;i=i+1))  do  /usr/sbin/useradd ${name}${i} &>/dev/null  echo $pass | /usr/bin/passwd --stdin "$name$i" &>/dev/null  done  fi  fi  ###批量删除用户  grep zxx\* /etc/passwd | awk -F : '{system("userdel -r "$1"")}'  注意：在awk中使用linux命令要通过system来调用。里面的域标记要通过“”引起来使用。 |

## 批量解压缩脚本

|  |
| --- |
| cd /lamp（find . –name \*.tar.gz -print | xargs tar –zxvf 替代）  ls \*.tar.gz >ls.log  for i in $(cat ls.log)  do  tar -zvf $i &>/dev/null  done  ###可以通过文件来保留住命令输出的内容 |

## 磁盘监控

|  |
| --- |
| #/bin/bash  rate=$( df -h | grep "/$" | awk '{print $(NF-1)}' |cut -d "%" -f 1)  if [ ${rate} -gt 80] ;then  mutt  fi  mussh -H /usr/local/hosts/all\_hosts -C /home/k12/diskCheck | mutt taochen@hengtiansoft.com xiaoxiazhang1@hengtiansoft.com -s "磁盘使用率" |

## 监控常见服务

|  |
| --- |
| status=nmap -sT ip | grep tcp | grep http | awk '{print $2}'  if [ ${port} == "open"];then  echo "$(date) httpd is ok" >> /tmp/autostart-acc.log  else  /etc/rc.d/init.d/httpd restart &>dev/null  echo "$(data) restart httpd !!" >>/tmp/autostart-err.log  fi |

注意：有些时候进程开着的，但是服务不一定正常服务，应该使用nmap

yum –y install nmap

## mysql的备份

|  |
| --- |
| #!/bin/bash  DATE=$(date +%Y-%d-%m)  DES=/usr/src/mysql\_bak  MYSQL\_U="root"  MYSQL\_P=""  MYSQL\_ALL=""  MYSQL\_SC=  if [ ! -d "$DES" ] ; then  mkdir -p "$DES"  fi  DB=$(mysql -u $MYSQL\_U -h $MYSQL\_H -p$MYSQL\_P -Bse 'show databases')  for database in $DB  do  if [ $database == "information" -o ...]  mysqldump -u $MYSQL\_U -h $MYSQL\_H -p$MYSQL\_P $database \  | bzip2> "$DES/${DATE}\_mysql.gz"  fi  done |
| --分表  #!/bin/bash  #MYSQL\_H="127.0.0.1"  MYSQL\_U="root"  MYSQL\_P="123"  #kpids=$(mysql -u $MYSQL\_U -h -p$MYSQL\_P -e 'SELECT DISTINCT knowledge\_point\_id FROM `k12\_tiku2.0`.question\_similar\_ref where knowledge\_point\_id is not null')  kpids=$(mysql -u $MYSQL\_U -p$MYSQL\_P -e 'SELECT DISTINCT knowledge\_point\_id FROM `k12\_tiku2.0`.question\_similar\_ref where knowledge\_point\_id is not null')  start=$(date +%s)  for kpid in $kpids  do  mysql -u $MYSQL\_U -p$MYSQL\_P test -e " drop table if exists question\_similar\_ref\_$kpid"  mysql -u $MYSQL\_U -p$MYSQL\_P test -e " create table question\_similar\_ref\_$kpid like `k12\_tiku2.0`.question\_similar\_ref"  mysql -u $MYSQL\_U -p$MYSQL\_P test -e " insert into question\_similar\_ref\_$kpid select \* from `k12\_tiku2.0`.question\_similar\_ref qs where qs.knowledge\_point\_id=$kpid"  done  end=$(date +%s)  echo "消耗的时间是:$(($end-$start))s" |

## 初始化服务器的脚本

|  |
| --- |
| #!/bin/bash  chmod +x /etc/rc.d/rc.local  #调整系统时间  systemctl stop ntpd.service  systemctl disable ntpd.service  cat >> /etc/crontab <<eof  0 0 \* \* \* /usr/sbin/ntpdate ntp.api.bz &>/dev/null  eof  echo "/usr/sbin/ntpdate ntp.api.bz" >> /etc/rc.local  #调整最大文件打开数  echo "ulimit -SHn 65535" >> /etc/rc.local  #调整内核tcp参数  cat >> /etc/sysctl.conf <<eof  net.ipv4.tcp\_fin\_timeout = 30  net.ipv4.tcp\_keepalive\_time = 1200  net.ipv4.tcp\_syncookies = 1  net.ipv4.tcp\_max\_tw\_buckets = 5000  net.ipv4.tcp\_tw\_reuse = 1  net.ipv4.tcp\_tw\_recycle = 1  net.ipv4.ip\_local\_port\_range = 1024 65000  net.ipv4.tcp\_max\_syn\_backlog = 8192    eof  /sbin/sysctl -p  #设置iptables  cat > /etc/sysconfig/iptables <<EOF  # Firewall configuration written by system-config-firewall  # Manual customization of this file is not recommended.  \*filter  :INPUT ACCEPT [0:0]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [0:0]  :synflood - [0:0]  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  -A INPUT -p icmp -j ACCEPT  -A INPUT -i lo -j ACCEPT  #-A INPUT -m state --state NEW -m tcp -p tcp -m multiport --dports 110,80,25,8080,8888,88,3306,11211,11311,11411,27017 -j ACCEPT  -A INPUT -s 10.25.51.74 -j ACCEPT  -A INPUT -s 10.25.248.168 -j ACCEPT  -A INPUT -s 10.25.248.125 -j ACCEPT  -A INPUT -s 122.225.227.160/28 -j ACCEPT  #防止外部的SYN洪水攻击  -A INPUT -p tcp -m state --state NEW -j synflood  -A synflood -m limit --limit 10/sec --limit-burst 100 -j RETURN  -A synflood -p tcp -j REJECT --reject-with tcp-reset  -A INPUT -j REJECT --reject-with icmp-host-prohibited  #防止ping洪水攻击，限制每秒的ping包不超过5个  -A INPUT -p icmp -m icmp --icmp-type 8 -m limit --limit 1/sec -j ACCEPT  -A FORWARD -j REJECT --reject-with icmp-host-prohibited  COMMIT  EOF  service iptables restart  #设置selinux  sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config  #重启机器  shutdown -r now |

## jdk安装

|  |
| --- |
| #!/bin/bash  JDK\_INSTALL=jdk-8u101-linux-x64.rpm  PAC\_DIR=/usr/local/src  JDK\_PCK=$(rpm -qa| grep jdk)  wget --no-cookies --header "Cookie: oraclelicense=accept-securebackup-cookie;" http://download.oracle.com/otn-pub/java/jdk/8u101-b13/jdk-8u101-linux-x64.rpm -P $PAC\_DIR  #wget ftp://kobe@172.16.129.212/jdk-8u101-linux-x64.rpm --ftp-password=k12\_prod -P $PAC\_DIR  if [ $? -ne 0 ] ;then  echo "-------获取jdk包失败-------"  exit  fi  if [ ! -z $JDK\_PCK ] ;then  echo "-------JDK已经安装--------"  exit  fi  rpm -ivh $PAC\_DIR/$JDK\_INSTALL &>/dev/null  cat >> /etc/profile <<EOF  export JAVA\_HOME=/usr/java/$(rpm -qa | grep jdk | cut -d - -f 1)  export PATH=\$JAVA\_HOME/bin:\$PATH  export CLASSPATH=.:\$JAVA\_HOME/lib/dt.jar:\$JAVA\_HOME/lib/tools.jar  EOF  source /etc/profile  echo "---安装JDK成功---" |

## Tomcat安装

|  |
| --- |
| #!/bin/bash  PAC\_DIR=/usr/local/src  TOMCAT\_DIR=/usr/local/tomcat\_admin  TOMCAT\_NAME=apache-tomcat-8.0.36.tar.gz  wget http://mirror.bit.edu.cn/apache/tomcat/tomcat-8/v8.0.36/bin/apache-tomcat-8.0.36.tar.gz -P $PAC\_DIR  #wget ftp://kobe@172.16.129.212/apache-tomcat-8.0.36.tar.gz --ftp-password=k12\_prod -P $PAC\_DIR  if [ $? -ne 0 ] ;then  echo "-------下载tomcat失败-------"  exit  fi  if [ -d $TOMCAT\_DIR ] ;then  echo "-------TOMCAT已经安装--------"  exit  fi  tar -zxvf $PAC\_DIR/$TOMCAT\_NAME -C /usr/local &>/dev/null  mv /usr/local/apache-tomcat\* $TOMCAT\_DIR  #修改tomcat的配置文件conf/server.xml  sed -i '69,71c <Connector port="8080" \n protocol="HTTP/1.1" \n maxThreads="1000" \n minProcessors="100" \n maxProcessors="1000" \n minSpareThreads="100" \n maxSpareThreads="1000" \n enableLookups="false" \n URIEncoding="utf-8" \n acceptCount="1000" \n connectionTimeout="20000" \n disableUploadTimeout="ture" \n redirectPort="8443" />' $TOMCAT\_DIR/conf/server.xml    #修改配置bin/catalina.sh  sed -i '/OS specific support/a JAVA\_OPTS="-Xms1536m -Xmx1536m "' $TOMCAT\_DIR/bin/catalina.sh  #修改配置tomcat-users.xml  /usr/bin/cp -f $TOMCAT\_DIR/conf/tomcat-users.xml $TOMCAT\_DIR/conf/tomcat-users.xml.bak  cat > $TOMCAT\_DIR/conf/tomcat-users.xml <<EOF  <?xml version='1.0' encoding='utf-8'?>  <tomcat-users>  <role rolename="manager-gui"/>  <role rolename="manager-jmx"/>  <role rolename="manager-script"/>  <user username="tomcat" password="p@\$\$w0rd" roles="manager-gui,manager-script,manager-script"/>  </tomcat-users>  EOF  #自启动tomcat  cat >> /etc/rc.local <<EOF  $TOMCAT\_DIR/bin/startup.sh  EOF |

## Memcached安装

|  |
| --- |
| #!/bin/bash  PAC\_DIR=/usr/local/src  yum -y install gcc  wget http://www.memcached.org/files/memcached-1.4.29.tar.gz -P $PAC\_DIR  wget https://github.com/libevent/libevent/releases/download/release-2.0.22-stable/libevent-2.0.22-stable.tar.gz -P $PAC\_DIR  #wget ftp://kobe@172.16.129.212/libevent-2.0.22-stable.tar.gz --ftp-password=k12\_prod -P $PAC\_DIR  #wget ftp://kobe@172.16.129.212/memcached-1.4.29.tar.gz --ftp-password=k12\_prod -P $PAC\_DIR  cd /usr/local/src  tar -zxvf libevent-2.0.22-stable.tar.gz  cd libevent-2.0.22-stable  ./configure --prefix=/usr  make & make install  cd /usr/local/src  tar -zxvf memcached-1.4.29.tar.gz  cd memcached-1.4.29  ./configure --prefix=/usr/local/memcache  make & make install  cat >> /etc/profile <<EOF  export PATH=\$PATH:/usr/local/memcache/bin  EOF  source /etc/profile #使配置立即生效  cat >> /etc/rc.local <<EOF  /usr/local/memcache/bin/memcached -d -m 1536 -u root -p 12000  EOF |

## nginx安装

|  |
| --- |
| #!/bin/bash  PAC\_DIR=/usr/local/src  NGINX\_DIR=/usr/local/nginx  NGINX\_NAME=nginx-1.10.1  #安装依赖库  yum -y install gcc pcre-devel zlib-devel  #下载软件包  #wget http://nginx.org/download/nginx-1.10.1.tar.gz -P $PAC\_DIR  wget ftp://kobe@172.16.129.212/nginx-1.10.1.tar.gz --ftp-password=k12\_prod -P $PAC\_DIR  if [ $? -ne 0 ] ;then  echo "-------下载nginx失败-------"  exit  fi  if [ -d $NGINX\_DIR ] ;then  echo "-------nginx已经安装--------"  exit  fi  useradd -s /sbin/nologin www  #编译安装nginx  tar -zxvf $PAC\_DIR/$NGINX\_NAME.tar.gz -C $PAC\_DIR  cd $PAC\_DIR/$NGINX\_NAME  ./configure --user=www --group=www --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_gzip\_static\_module  make && make install  #配置环境变量  cat >> /etc/profile<<EOF  export NGINX\_HOME=/usr/local/nginx  export PATH=\$PATH:\$NGINX\_HOME/sbin  EOF  source /etc/profile  #备份和修改配置  mkdir /data  /usr/bin/cp -f /usr/local/nginx/conf/nginx.conf /usr/local/nginx/conf/nginx.bak  cat > /usr/local/nginx/conf/nginx.conf <<EOF  user www;  worker\_processes 4;  error\_log logs/error.log notice;  pid /var/run/nginx.pid;  worker\_rlimit\_nofile 65536;  events {  use epoll;  worker\_connections 65536;  }  http {  include mime.types;  default\_type application/octet-stream;  charset UTF-8;  limit\_conn\_zone \$binary\_remote\_addr zone=one:10m;  log\_format main '\$remote\_addr - \$remote\_user [\$time\_local] '  '"\$request" \$status \$bytes\_sent '  '"\$http\_referer" "\$http\_user\_agent" '  '"\$gzip\_ratio"';  log\_format download '\$remote\_addr - \$remote\_user [\$time\_local] '  '"\$request" \$status \$bytes\_sent '  '"\$http\_referer" "\$http\_user\_agent" '  '"\$http\_range" "\$sent\_http\_content\_range"';  access\_log logs/access.log main;  client\_max\_body\_size 20m;  client\_header\_buffer\_size 32k;  large\_client\_header\_buffers 4 128k;  sendfile on;  tcp\_nopush on;  tcp\_nodelay on;  keepalive\_timeout 60;  client\_header\_timeout 10;  client\_body\_timeout 10;  send\_timeout 10;  client\_body\_buffer\_size 512k;  proxy\_connect\_timeout 5;  proxy\_read\_timeout 60;  proxy\_send\_timeout 5;  proxy\_buffer\_size 16k;  proxy\_buffers 4 64k;  proxy\_busy\_buffers\_size 128k;  proxy\_temp\_file\_write\_size 64k;  proxy\_redirect off;  proxy\_set\_header HOST \$host;  proxy\_set\_header X-Real-IP \$remote\_addr;  proxy\_set\_header X-Forwarded-For \$proxy\_add\_x\_forwarded\_for;  gzip on;  gzip\_min\_length 1k;  gzip\_buffers 4 16k;  gzip\_http\_version 1.1;  gzip\_comp\_level 4;  gzip\_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript;  gzip\_vary on;  proxy\_cache\_path /data/cache levels=1:2 keys\_zone=cache\_one:2048m max\_size=1g inactive=600m;  proxy\_temp\_path /data/temp;  open\_file\_cache max=65536 inactive=20s;  open\_file\_cache\_valid 30s;  open\_file\_cache\_min\_uses 2;  open\_file\_cache\_errors on;      # upstream webServer{  # server 10.161.171.195:80 weight=3 max\_fails=3 fail\_timeout=20s;  # server 10.161.158.22:80 weight=3 max\_fails=3 fail\_timeout=20s;  # server 10.161.132.101:80 weight=3 max\_fails=3 fail\_timeout=20s;  # }  server{  listen 80;  server\_name 10.25.51.74;  index index.html index.htm ;  root /var/ftp/3tbetter;  charset UTF-8;  access\_log logs/access.log main;  #错误页面处理  error\_page 404 /404.html;  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }    location = / {  proxy\_pass http://10.25.248.168:8080/3t-admin;  }  #静态请求处理  #location ~ .\*\.(htm|html|js|css|less|json|gif|jpg|jpeg|png|bmp|swf|ioc|rar|zip|txt|flv|mid|doc|ppt|pdf|xls|mp3|wma|apk|zip|rar)\$ {  # root ...;  # expires 30d;  #}    location /download/ {  limit\_conn one 1;  valid\_referers none blocked server\_names \*.example.com;  if (\$invalid\_referer) {  return 403;  }  rewrite ^/(download/.\*)/mp3/(.\*)\..\*\$ /\$1/mp3/\$2.mp3 break;  #root /home/k12;  access\_log logs/download.log download;  }  #新旧地址交替  #location /old\_stuff/ {  # rewrite ^/old\_stuff/(.\*)\$ /new\_stuff/\$1 permanent;  #}  location /NginxStatus {  stub\_status on;  access\_log off;  #allow可以有多个  allow 122.225.227.162;  deny all;  }    location ^~ /3t-admin/ {  proxy\_cache cache\_one;  proxy\_cache\_key \$host\$uri\$is\_args\$args;  proxy\_pass http://10.25.248.168:8080;  proxy\_next\_upstream http\_500 http\_502 http\_503 error timeout invalid\_header;    }  location ^~ /3t-wechat/ {  proxy\_cache cache\_one;  proxy\_cache\_key \$host\$uri\$is\_args\$args;  proxy\_pass http://10.25.248.168:80;  proxy\_next\_upstream http\_500 http\_502 http\_503 error timeout invalid\_header;    }  location ^~ /3t-mobile-emp/ {  proxy\_cache cache\_one;  proxy\_cache\_key \$host\$uri\$is\_args\$args;  proxy\_pass http://10.25.248.125:80;  proxy\_next\_upstream http\_500 http\_502 http\_503 error timeout invalid\_header;    }  location ^~ /3t-mobile-user/ {  proxy\_cache cache\_one;  proxy\_cache\_key \$host\$uri\$is\_args\$args;  proxy\_pass http://10.25.248.125:8080;  proxy\_next\_upstream http\_500 http\_502 http\_503 error timeout invalid\_header;    }    location ^~ /3t-job/ {  proxy\_cache cache\_one;  proxy\_cache\_key \$host\$uri\$is\_args\$args;  proxy\_pass http://10.25.248.125:8888;  proxy\_next\_upstream http\_500 http\_502 http\_503 error timeout invalid\_header;    }    location /{  proxy\_cache cache\_one;  proxy\_cache\_key \$host\$uri\$is\_args\$args;  proxy\_pass http://10.25.248.168:8080/3t-admin;  proxy\_next\_upstream http\_500 http\_502 http\_503 error timeout invalid\_header;  }      }  }  EOF  #启动nginx  nginx -c /usr/local/nginx/conf/nginx.conf  #自启动  echo " /usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf" >> /etc/rc.local |
| #加入openssl  wget http://www.openssl.org/source/openssl-1.0.2.tar.gz  tar zxf openssl-1.0.2.tar.gz -C /usr/local  ./configure --user=www --group=www --prefix=/usr/local/nginx1 --with-http\_stub\_status\_module --with-http\_gzip\_static\_module --with-http\_ssl\_module --with-openssl=/usr/local/openssl-1.0.2  make && make install |

## jenkins部署脚本

|  |
| --- |
| JENKIN\_HOME=/root/.jenkins  TOMCAT\_HOME=/usr/local/tomcat\_admin  JENKINS\_PROJECT\_NAME=3t\_prod\_admin  HOST=139.224.28.17  PORT=8080  PROJECT=3t-admin  TIME1=$(date +%s)  INTERVAL=60  /usr/bin/mussh -m -h $HOST -c "kill -9 $(ps -ef | grep $TOMCAT\_HOME | grep -v -E 'grep|tail' | awk '{ print $2 }')"  /usr/bin/mussh -m -h $HOST -c "rm -rf $TOMCAT\_HOME/webapps/$PROJECT\*"  scp $JENKIN\_HOME/jobs/$JENKINS\_PROJECT\_NAME/workspace/$PROJECT/target/$PROJECT-0.0.1-SNAPSHOT.war root@$HOST:$TOMCAT\_HOME/webapps/$PROJECT.war  /usr/bin/mussh -m -h $HOST -c "$TOMCAT\_HOME/bin/startup.sh"  while true  do  mytime=$(($(date +%s)-$TIME1))    if [ mytime -gt $ ];then  echo "部署超时"  exit  fi  HTTP\_CODE=$(curl -I -m 10 -o /dev/null -s -w %{http\_code} $HOST:$PORT)  if [ $HTTP\_CODE -lt 400 ];then  break  fi  sleep 3  done  echo "部署成功" |
| Maven命名  clean install -pl 3t-admin -am -Pprod -DskipTests=true  JENKIN\_HOME=/root/.jenkins  TOMCAT\_HOME=/usr/local/tomcat\_admin  TOMCAT\_BAK\_HOME=/usr/local/tomcat8  JENKINS\_PROJECT\_NAME=3t\_prod\_admin  PROJECT=3t-admin  HOST=139.224.28.17  HOST\_BAK=139.224.33.13  PORT\_1=8005  PORT\_2=8080  PORT\_3=8009  /usr/bin/mussh -m -h $HOST -c "$TOMCAT\_HOME/bin/shutdown.sh"  /usr/bin/mussh -m -h $HOST -c "fuser -k $PORT\_1/tcp"  /usr/bin/mussh -m -h $HOST -c "fuser -k $PORT\_2/tcp"  /usr/bin/mussh -m -h $HOST -c "fuser -k $PORT\_3/tcp"  /usr/bin/mussh -m -h $HOST -c "rm -rf $TOMCAT\_HOME/webapps/$PROJECT\*"  scp $JENKIN\_HOME/jobs/$JENKINS\_PROJECT\_NAME/workspace/$PROJECT/target/$PROJECT-0.0.1-SNAPSHOT.war root@$HOST:$TOMCAT\_HOME/webapps/$PROJECT.war  /usr/bin/mussh -m -h $HOST -c "$TOMCAT\_HOME/bin/startup.sh"  sleep 40  /usr/bin/mussh -m -h $HOST\_BAK -c "rm -rf $TOMCAT\_BAK\_HOME/webapps/$PROJECT\*"  scp $JENKIN\_HOME/jobs/$JENKINS\_PROJECT\_NAME/workspace/$PROJECT/target/$PROJECT-0.0.1-SNAPSHOT.war root@$HOST\_BAK:$TOMCAT\_BAK\_HOME/webapps/$PROJECT.war  sleep 40  echo "部署成功" |

## 系统监控

|  |
| --- |
| #!/bin/bash  . /etc/profile  OUTPUT\_FILE=/home/kobe/sys\_status.txt  NGINX\_STATUS=$(nmap -sT 139.224.33.13 -p80 | grep 80)  FTP\_STATUS=$(nmap -sT 139.224.33.13 -p21 | grep 21)  MEMCACHED\_STATUS=$(nmap -sT 139.224.33.13 -p12000 | grep 12000)  ADMIN\_STATUS=$(nmap -sT 139.224.28.17 -p8080 | grep 8080)  WECHAT\_STATUS=$(nmap -sT 139.224.28.17 -p80 | grep 80)  JOB\_STATUS=$(nmap -sT 139.224.28.40 -p8888 | grep 8888)  EMP\_STATUS=$(nmap -sT 139.224.28.40 -p80 | grep 80)  USER\_STATUS=$(nmap -sT 139.224.28.40 -p8080 | grep 8080)  echo "创建时间：$(date +'%Y-%m-%d %H:%M:%S') 创建人：xiaoxiazhang" >$OUTPUT\_FILE  ips="139.224.33.13 139.224.28.17 139.224.28.40"  for ip in $ips  do  echo "================================================================================" >>$OUTPUT\_FILE  echo "服务器$ip系统信息" >>$OUTPUT\_FILE  echo "================================================================================" >>$OUTPUT\_FILE  DISK\_RATE=$(mussh -h $ip -c "df -h | grep '/$' | awk '{print \$(NF-1)}' |cut -d '%' -f 1" | cut -d ':' -f 2)  MEM\_RATE=$(mussh -h $ip -c "free -m | sed -n '2p' | awk '{print \$3/\$2\*100}'" | cut -d ':' -f 2)  echo "CPU平均负载: $(mussh -m -h $ip -c "uptime" | grep ":" | cut -d ':' -f 6)" >>$OUTPUT\_FILE  if [ ${DISK\_RATE} -gt 85 ];then  echo "磁盘占用率:${DISK\_RATE}%" >>$OUTPUT\_FILE  else  echo "磁盘占用率:${DISK\_RATE}%" >>$OUTPUT\_FILE  fi  #if [ ${MEM\_RATE} -gt 90 ];then  # echo "内存使用率:${MEM\_RATE}%" >>$OUTPUT\_FILE  #else  echo "内存使用率:${MEM\_RATE}%" >>$OUTPUT\_FILE  #fi    echo "================================================================================" >>$OUTPUT\_FILE  echo "服务器$ip系统信息" >>$OUTPUT\_FILE  echo "================================================================================" >>$OUTPUT\_FILE  echo "" >>$OUTPUT\_FILE  echo "" >>$OUTPUT\_FILE  done  echo "================================================================================" >>$OUTPUT\_FILE  echo "服务器应用信息" >>$OUTPUT\_FILE  echo "================================================================================" >>$OUTPUT\_FILE  echo "nginx服务状态: $NGINX\_STATUS" >>$OUTPUT\_FILE  echo "ftp服务状态: $FTP\_STATUS" >>$OUTPUT\_FILE  echo "memcache服务状态: $MEMCACHED\_STATUS" >>$OUTPUT\_FILE  echo "3t-admin服务状态: $ADMIN\_STATUS" >>$OUTPUT\_FILE  echo "3t-wechat服务状态: $WECHAT\_STATUS" >>$OUTPUT\_FILE  echo "3t-job服务状态: $JOB\_STATUS" >>$OUTPUT\_FILE  echo "3t-mobile-emp服务状态: $EMP\_STATUS" >>$OUTPUT\_FILE  echo "3t-mobile-user服务状态: $USER\_STATUS" >>$OUTPUT\_FILE  echo "" >>$OUTPUT\_FILE  echo "================================================================================" >>$OUTPUT\_FILE  echo "服务器应用信息" >>$OUTPUT\_FILE  echo "================================================================================" >>$OUTPUT\_FILE  #发送邮件  mail -s "3t生产服务器状态" xiaoxiazhang1@hengtiansoft.com dongdongliu@hengtiansoft.com taochen@hengtiansoft.com yulangwang@hengtiansoft.com < sys\_status.txt |

## 日志监控

|  |
| --- |
| #!/usr/bin/env python  # coding=utf-8  #---------------------------------------------------------  # Name: Tomcat错误日志发送邮件脚本  # Purpose: 收集Tomcat异常日志并发送邮件  # Created: 2016-08-22  # Python： 2.7/2.4 皆可使用  #--------------------------------------------------------  from smtplib import SMTP  from email import MIMEText  from email import Header  from os.path import getsize  from sys import exit  from re import compile, IGNORECASE  #定义主机 帐号 密码 收件人 邮件主题  smtpserver = 'smtp.163.com'  sender = 'h861336327@163.com'  password = 'zh19931006ZH'  receiver = ('xiaoxiazhang1@hengtiansoft.com')  subject = u'Web服务器Tomcat日志错误信息'  From = u'xxx Web服务器'  To = u'服务器管理员'  #定义tomcat日志文件位置  tomcat\_log = '/usr/local/tomcat\_admin/logs/catalina.out'  #该文件是用于记录上次读取日志文件的位置,执行脚本的用户要有创建该文件的权限  last\_position\_logfile = '/root/last\_position.txt'  #匹配的错误信息关键字的正则表达式  pattern = compile(r'Exception|^\s+\bat\b',IGNORECASE)  #发送邮件函数  def send\_mail(error):  #定义邮件的头部信息  header = Header.Header  msg = MIMEText.MIMEText(error,'plain','utf-8')  msg['From'] = header(From)  msg['To'] = header(To)  msg['Subject'] = header(subject+'\n')  #连接SMTP服务器，然后发送信息  smtp = SMTP(smtpserver)  smtp.login(sender, password)  smtp.sendmail(sender, receiver, msg.as\_string())  smtp.close()  #读取上一次日志文件的读取位置  def get\_last\_position(file):  try:  data = open(file,'r')  last\_position = data.readline()  if last\_position:  last\_position = int(last\_position)  else:  last\_position = 0  except:  last\_position = 0  return last\_position  #写入本次日志文件的本次位置  def write\_this\_position(file,last\_positon):  try:  data = open(file,'w')  data.write(str(last\_positon))  data.write('\n' + "Don't Delete This File,It is Very important for Looking Tomcat Error Log !! \n")  data.close()  except:  print "Can't Create File !" + file  exit()  #分析文件找出异常的行  def analysis\_log(file):  error\_list = [] #定义一个列表，用于存放错误信息.  try:  data = open(file,'r')  except:  exit()  last\_position = get\_last\_position(last\_position\_logfile) #得到上一次文件指针在日志文件中的位置  this\_postion = getsize(tomcat\_log) #得到现在文件的大小，相当于得到了文件指针在末尾的位置  if this\_postion < last\_position: #如果这次的位置 小于 上次的位置说明 日志文件轮换过了，那么就从头开始  data.seek(0)  elif this\_postion == last\_position: #如果这次的位置 等于 上次的位置 说明 还没有新的日志产生  exit()  elif this\_postion > last\_position: #如果是大于上一次的位置，就移动文件指针到上次的位置  data.seek(last\_position)  for line in data:  if pattern.search(line):  error\_list.append(line)  write\_this\_position(last\_position\_logfile,data.tell()) #写入本次读取的位置  data.close()  return ''.join(error\_list) #形成一个字符串  #调用发送邮件函数发送邮件  error\_info = analysis\_log(tomcat\_log)  if error\_info:  send\_mail(error\_info) |