第 2 章 Shell 编程高级企业实战

企业生产环境中，服务器规模成百上千，如果依靠人工管理和维护，将非常吃力。Shell 编程脚本使管理和维护服务器变得简单、从容，对企业自动化运维之路的建设起到极大的推动作用。

本章将介绍企业生产环境 Shell 编程案例、自动化备份 MySQL 数据、服务器信息收集、防止恶意 IP 访问、LAMP+MySQL 主从实战、千台服务器 IP 修改、Nginx+Tomcat 高级自动化部署脚本、Nginx 虚拟主机配置、Docker 管理平台等。

2.1 Shell 编程 Linux 系统备份脚本

日常企业运维中，需要备份 Linux 操作系统中重要的文件，例如/etc、/boot 分区，重要网站数据等。在备份时，由于数据量非常大，需要指定高效的备份方案，以下为常用的备份数据方案：

（1）每周日进行完整备份，周一～周六使用增量备份。

（2）每周六进行完整备份，周日～周五使用增量备份。

企业备份数据的工具主要有 tar、cp、rsync、scp、sersync、dd 等。以下为基于开源 tar 工具实现系统数据备份方案。

tar 工具手动全备份网站，–g 参数指定新的快照文件。

```
tar -g  /tmp/snapshot  -czvf  /tmp/2021_full_system_data.tar.gz  /data/sh/
```

tar 工具手动增量备份网站，–g 参数指定全备已生成的快照文件，后续增量备份基于上一个增量备份快照文件。

```
tar -g  /tmp/snapshot  -czvf  /tmp/2021_add01_system_data.tar.gz /data/sh/
```

tar 工具全备份、增量备份网站，Shell 脚本实现自动打包备份编写思路如下：

（1）系统备份数据按每天存放。

（2）创建完整备份函数块。

（3）创建增量备份函数块。

（4）根据星期数判断完整或增量。

（5）将脚本加入 Crontab 实现自动备份。

tar 工具全备份、增量备份网站，Shell 脚本实现自动打包备份，相关代码如下：

```
#!/bin/bash
#Auto Backup Linux System Files
#By author jfedu.net 2021
#Define Path variables
SOURCE_DIR=(
    $*
)
TARGET_DIR=/data/backup/
YEAR='date +%Y'
MONTH='date +%m'
DAY='date +%d'
WEEK='date +%u'
A_NAME='date +%H%M'
FILES=system_backup.tgz
CODE=$?
if
    [ -z "$*" ];then
    echo -e "\033[32mUsage:\nPlease Enter Your Backup Files or Directories\
n-------------------------------------------\n\nUsage: { $0 /boot /etc}\
033[0m"
    exit
fi
#Determine Whether the Target Directory Exists
if
    [ ! -d $TARGET_DIR/$YEAR/$MONTH/$DAY ];then
    mkdir -p $TARGET_DIR/$YEAR/$MONTH/$DAY
    echo -e "\033[32mThe $TARGET_DIR Created Successfully !\033[0m"
fi
#EXEC Full_Backup Function Command
Full_Backup()
```

```
{
if
    [ "$WEEK" -eq "7" ];then
    rm -rf $TARGET_DIR/snapshot
    cd $TARGET_DIR/$YEAR/$MONTH/$DAY ;tar -g $TARGET_DIR/snapshot -czvf
$FILES ${SOURCE_DIR[@]}
    [ "$CODE" == "0" ]&&echo -e  "----------------------------------------
-----\n\033[32mThese Full_Backup System Files Backup Successfully !\033[0m"
fi
}
#Perform incremental BACKUP Function Command
Add_Backup()
{
   if
        [ $WEEK -ne "7" ];then
        cd $TARGET_DIR/$YEAR/$MONTH/$DAY ;tar -g $TARGET_DIR/snapshot -czvf
$A_NAME$FILES ${SOURCE_DIR[@]}
        [ "$CODE" == "0" ]&&echo -e  "----------------------------------------
-----\n\033[32mThese Add_Backup System Files $TARGET_DIR/$YEAR/$MONTH/
$DAY/${YEAR}_$A_NAME$FILES Backup Successfully !\033[0m"
   fi
}
sleep 3
Full_Backup;Add_Backup
```

在 Crontab 任务计划中添加如下语句，每天凌晨 1 点整执行备份脚本即可。

```
0  1  *  *  * /bin/sh /data/sh/auto_backup.sh /boot /etc/ >> /tmp/back.log
2>&1
```

2.2　Shell 编程收集服务器信息脚本

在企业生产环境中，经常需要对服务器资产进行统计存档，单台服务器可以手动统计服务器的 CPU 型号、内存大小、硬盘容量、网卡流量等，但如果服务器数量超过百台、千台，使用手动方式将变得非常吃力。

Shell 脚本实现自动化服务器硬件信息的收集，并将收集的内容存放在数据库，能更快、更高效地实现对服务器资产信息的管理。Shell 脚本实现服务器信息自动收集的编写思路如下：

（1）创建数据库和表存储服务器信息。

（2）Shell“四剑客”awk、find、sed、grep 获取服务器信息。

（3）将获取的信息写成 SQL 语句。

（4）定期对 SQL 数据进行备份。

（5）将脚本加入 Crontab 实现自动备份。

创建数据库表，创建 SQL 语句如下：

```
CREATE TABLE 'audit_system' (
  'id' int(11) NOT NULL AUTO_INCREMENT,
  'ip_info' varchar(50) NOT NULL,
  'serv_info' varchar(50) NOT NULL,
  'cpu_info' varchar(50) NOT NULL,
  'disk_info' varchar(50) NOT NULL,
  'mem_info' varchar(50) NOT NULL,
  'load_info' varchar(50) NOT NULL,
  'mark_info' varchar(50) NOT NULL,
  PRIMARY KEY ('id'),
  UNIQUE KEY 'ip_info' ('ip_info'),
  UNIQUE KEY 'ip_info_2' ('ip_info')
  );
```

Shell 脚本实现服务器信息自动收集，相关代码如下：

```
#!/bin/bash
#Auto get system info
#By author jfedu.net 2021
#Define Path variables
echo -e "\033[34m \033[1m"
cat <<EOF
+++++++++++++++++++++++++++++++++++++++++++++
++++++++Welcome to use system Collect+++++++++
+++++++++++++++++++++++++++++++++++++++++++++
EOF
ip_info='ifconfig |grep "Bcast"|tail -1 |awk '{print $2}'|cut -d: -f 2'
cpu_info1='cat /proc/cpuinfo |grep 'model name'|tail -1 |awk -F: '{print
$2}'|sed 's/^ //g'|awk '{print $1,$3,$4,$NF}''
cpu_info2='cat /proc/cpuinfo |grep "physical id"|sort |uniq -c|wc -l'
serv_info='hostname |tail -1'
disk_info='fdisk -l|grep "Disk"|grep -v "identifier"|awk '{print $2,$3,
$4}'|sed 's/,//g''
mem_info='free -m |grep "Mem"|awk '{print "Total",$1,$2"M"}'
load_info='uptime |awk '{print "Current Load: "$(NF-2)}'|sed 's/\,//g''
```

```
mark_info='BeiJing_IDC'
echo -e "\033[32m-------------------------------------------\033[1m"
echo IPADDR:${ip_info}
echo HOSTNAME:$serv_info
echo CPU_INFO:${cpu_info1} X${cpu_info2}
echo DISK_INFO:$disk_info
echo MEM_INFO:$mem_info
echo LOAD_INFO:$load_info
echo -e "\033[32m-------------------------------------------\033[0m"
echo -e -n "\033[36mYou want to write the data to the databases? \033[1m" ;
read ensure
if [ "$ensure" == "yes" -o "$ensure" == "y" -o "$ensure" == "Y" ];then
   echo "-------------------------------------------"
   echo -e  '\033[31mmysql -uaudit -p123456 -D audit -e ''' "insert into
audit_system values('','${ip_info}','$serv_info','${cpu_info1} X${cpu_info2}',
'$disk_info','$mem_info','$load_info','$mark_info')" ''' \033[0m '
   mysql -uroot -p123456 -D test -e "insert into audit_system values
('','${ip_info}','$serv_info','${cpu_info1} X${cpu_info2}','$disk_info',
'$mem_info','$load_info','$mark_info')"
else
   echo "Please wait,exit......"
   exit
fi
```

手动读取数据库服务器信息命令：

```
mysql -uroot -p123 -e 'use wugk1 ;select * from audit_audit_system;'|sed
's/-//g'|grep -v "id"
```

2.3　Shell 编程拒绝恶意 IP 登录脚本

企业服务器暴露在外网，每天会有大量的人使用各种用户名和密码尝试登录服务器，如果用户一直尝试，难免会猜出密码。通过开发 Shell 脚本，可以自动将尝试登录服务器错误密码一定次数的 IP 列表加入防火墙配置。

Shell 脚本实现服务器拒绝恶意 IP 登录，编写思路如下：

（1）登录服务器日志/var/log/secure。

（2）检查日志中认证失败一定次数的行并打印其 IP 地址。

（3）将 IP 地址写入防火墙。

（4）禁止该 IP 访问服务器 SSH 22 端口。

（5）将脚本加入 Crontab 实现自动禁止恶意 IP。

Shell 脚本实现服务器拒绝恶意 IP 登录，相关代码如下：

```
#!/bin/bash
#Auto drop ssh failed IP address
#By author jfedu.net 2021
#Define Path variables
SEC_FILE=/var/log/secure
IP_ADDR='awk '{print $0}'  /var/log/secure|grep -i  "fail"| egrep -o "
([0-9]{1,3}\.){3}[0-9]{1,3}" | sort -nr | uniq -c |awk '$1>=15 {print $2}''
IPTABLE_CONF=/etc/sysconfig/iptables
echo
cat <<EOF
++++++++++++++welcome to use ssh login drop failed ip+++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++-----------------------------------------+++++++++++++++++
EOF
echo
for ((j=0;j<=6;j++)) ;do echo -n "-";sleep 1 ;done
echo
for i in 'echo $IP_ADDR'
do
    cat $IPTABLE_CONF |grep $i >/dev/null
if
    [ $? -ne 0 ];then
    sed -i "/lo/a -A INPUT -s $i -m state --state NEW -m tcp -p tcp --dport 22
-j DROP" $IPTABLE_CONF
fi
done
NUM='find /etc/sysconfig/  -name iptables -a -mmin -1|wc -l'
       if [ $NUM -eq 1 ];then
              /etc/init.d/iptables restart
       fi
```

2.4 Shell 编程 LAMP 部署脚本

LAMP 是目前互联网主流 Web 网站架构，通过源码安装、维护和管理单台服务器很轻松，如果服务器数量较多，手动管理就非常困难，Shell 脚本可以更快速地维护 LAMP 架构。

Shell 脚本实现服务器 LAMP 一键源码安装配置，编写思路如下：

（1）利用脚本安装 LAMP 环境。

（2）Apache 安装配置，MySQL、PHP 安装。

（3）源码 LAMP 整合配置。

（4）启动数据库，创建数据库并授权。

（5）重启 LAMP 所有服务，验证访问。

利用 Shell 脚本实现服务器 LAMP 一键源码安装配置，相关代码如下：

```
#!/bin/bash
#Auto install LAMP
#By author jfedu.net 2021
#Define Path variables
#Httpd define path variable
H_FILES=httpd-2.2.32.tar.bz2
H_FILES_DIR=httpd-2.2.32
H_URL=http://mirrors.cnnic.cn/apache/httpd/
H_PREFIX=/usr/local/apache2/
#MySQL define path variable
M_FILES=mysql-5.5.20.tar.gz
M_FILES_DIR=mysql-5.5.20
M_URL=http://down1.chinaunix.net/distfiles/
M_PREFIX=/usr/local/mysql/
#PHP define path variable
P_FILES=php-5.3.28.tar.bz2
P_FILES_DIR=php-5.3.28
P_URL=http://mirrors.sohu.com/php/
P_PREFIX=/usr/local/php5/
function httpd_install(){
if [[ "$1" -eq "1" ]];then
   wget -c $H_URL/$H_FILES &&  tar -jxvf $H_FILES && cd $H_FILES_DIR
&&./configure --prefix=$H_PREFIX
   if [ $? -eq 0 ];then
      make && make install
   fi
fi
}
function mysql_install(){
if [[ "$1" -eq "2" ]];then
wget -c $M_URL/$M_FILES &&  tar -xzvf $M_FILES && cd $M_FILES_DIR &&yum
```

```
install cmake ncurses-devel -y ;cmake . -DCMAKE_INSTALL_PREFIX=$M_PREFIX \
-DMYSQL_UNIX_ADDR=/tmp/mysql.sock \
-DMYSQL_DATADIR=/data/mysql \
-DSYSCONFDIR=/etc \
-DMYSQL_USER=mysql \
-DMYSQL_TCP_PORT=3306 \
-DWITH_XTRADB_STORAGE_ENGINE=1 \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DWITH_PARTITION_STORAGE_ENGINE=1 \
-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \
-DWITH_MYISAM_STORAGE_ENGINE=1 \
-DWITH_READLINE=1 \
-DENABLED_LOCAL_INFILE=1 \
-DWITH_EXTRA_CHARSETS=1 \
-DDEFAULT_CHARSET=utf8 \
-DDEFAULT_COLLATION=utf8_general_ci \
-DEXTRA_CHARSETS=all \
-DWITH_BIG_TABLES=1 \
-DWITH_DEBUG=0
if [ $? -eq 0 ];then
   make && make install
   echo -e "\n\033[32m-----------------------------------------------\033
[0m"
             echo -e "\033[32mThe $M_FILES_DIR Server Install Success !\
033[0m"
      else
             echo -e "\033[32mThe $M_FILES_DIR Make or Make install
ERROR,Please Check......"
             exit 0
fi
/bin/cp support-files/my-small.cnf  /etc/my.cnf
/bin/cp support-files/mysql.server /etc/init.d/mysqld
chmod +x /etc/init.d/mysqld
chkconfig --add mysqld
chkconfig mysqld on
fi
}
function php_install(){
if [[ "$1" -eq "3" ]];then
      yum install  libxml2-devel  perl-devel  perl  libtool* -y
      wget -c $P_URL/$P_FILES &&  tar -jxvf $P_FILES && cd $P_FILES_DIR
```

```
&&./configure --prefix=$P_PREFIX --with-config-file-path=$P_PREFIX/etc
--with-mysql=$M_PREFIX --with-apxs2=$H_PREFIX/bin/apxs
       if [ $? -eq 0 ];then
              make ZEND_EXTRA_LIBS='-liconv' && make install
              echo -e "\n\033[32m----------------------------------------
-------\033[0m"
              echo -e "\033[32mThe $P_FILES_DIR Server Install Success !\
033[0m"
       else
              echo -e "\033[32mThe $P_FILES_DIR Make or Make install
ERROR,Please Check......"
              exit 0
       fi
fi
}
function lamp_config(){
if [[ "$1" -eq "4" ]];then
   sed -i '/DirectoryIndex/s/index.html/index.php index.html/g' $H_PREFIX/
conf/httpd.conf
   $H_PREFIX/bin/apachectl restart
   echo "AddType    application/x-httpd-php .php" >>$H_PREFIX/conf/httpd.
conf
   IP='ifconfig eth0|grep "Bcast"|awk '{print $2}'|cut -d: -f2'
   echo "You can to access http://$IP/"

cat >$H_PREFIX/htdocs/index.php<<EOF
<?php
phpinfo();
?>
EOF
fi
}
PS3="Please enter you select install menu:"
select i in http mysql php config quit
do

case $i in
   http)
   httpd_install 1
   ;;
   mysql)
```

```
    mysql_install 2
    ;;
    php)
    php_install 3
    ;;
    config)
    lamp_config 4
    ;;
    quit)
    exit
esac
done
```

2.5 Shell 编程 LNMP 部署脚本

Shell 脚本实现 LNMP 部署安装，编写思路如下：

（1）利用脚本的功能，实现安装 LNMP 环境。

（2）Nginx 安装配置，MySQL、PHP 安装。

（3）源码 LNMP 整合配置。

（4）启动数据库，创建数据库并授权。

（5）重启 LNMP 所有服务，验证访问。

相关代码如下：

```
#!/bin/bash
#2019年10月30日19:28:15
#auto install lnmp web.
#by author www.jfedu.net
######################
if [ $1 -eq 1 ];then
    #Install Nginx WEB.
    yum install -y wget gzip tar make gcc
    yum install -y pcre pcre-devel zlib-devel
    wget -c http://nginx.org/download/nginx-1.16.0.tar.gz
    tar zxf nginx-1.16.0.tar.gz
    cd nginx-1.16.0
    useradd -s /sbin/nologin www -M
    ./configure --user=www --group=www --prefix=/usr/local/nginx
    make && make install
```

```
    /usr/local/nginx/sbin/nginx
    setenforce 0
    systemctl stop firewalld.service
fi

if [ $1 -eq 2 ];then
    #Install MySQL Database.
    cd ../
    yum install -y gcc-c++ ncurses-devel cmake make perl gcc autoconf
    yum install -y automake zlib libxml2 libxml2-devel libgcrypt libtool bison
    wget -c http://mirrors.163.com/mysql/Downloads/MySQL-5.6/mysql-5.6.45.
tar.gz
    tar -xzf mysql-5.6.45.tar.gz
    cd mysql-5.6.45
    cmake  .  -DCMAKE_INSTALL_PREFIX=/usr/local/mysql56/ \
    -DMYSQL_UNIX_ADDR=/tmp/mysql.sock \
    -DMYSQL_DATADIR=/data/mysql \
    -DSYSCONFDIR=/etc \
    -DMYSQL_USER=mysql \
    -DMYSQL_TCP_PORT=3306 \
    -DWITH_XTRADB_STORAGE_ENGINE=1 \
    -DWITH_INNOBASE_STORAGE_ENGINE=1 \
    -DWITH_PARTITION_STORAGE_ENGINE=1 \
    -DWITH_BLACKHOLE_STORAGE_ENGINE=1 \
    -DWITH_MYISAM_STORAGE_ENGINE=1 \
    -DWITH_READLINE=1 \
    -DENABLED_LOCAL_INFILE=1 \
    -DWITH_EXTRA_CHARSETS=1 \
    -DDEFAULT_CHARSET=utf8 \
    -DDEFAULT_COLLATION=utf8_general_ci \
    -DEXTRA_CHARSETS=all \
    -DWITH_BIG_TABLES=1 \
    -DWITH_DEBUG=0
    make
    make install
    #Config MySQL Set System Service
    cd /usr/local/mysql56/
    \cp support-files/my-large.cnf /etc/my.cnf
    \cp support-files/mysql.server /etc/init.d/mysqld
    chkconfig --add mysqld
    chkconfig --level 35 mysqld on
```

```
    mkdir  -p  /data/mysql
    useradd  mysql
    /usr/local/mysql56/scripts/mysql_install_db --user=mysql --datadir=/
data/mysql/ --basedir=/usr/local/mysql56/
    ln  -s  /usr/local/mysql56/bin/* /usr/bin/
    service  mysqld  restart
fi

if [ $1 -eq 3 ];then
    #Install PHP WEB 2018
    cd ../../
    yum install libxml2 libxml2-devel -y
    wget http://mirrors.sohu.com/php/php-5.6.28.tar.bz2
    tar jxf  php-5.6.28.tar.bz2
    cd php-5.6.28
    ./configure --prefix=/usr/local/php5 --with-config-file-path=/usr/
local/php5/etc --with-mysql=/usr/local/mysql56/ --enable-fpm
    make
    make install
fi

if [ $1 -eq 4 ];then
#Config LNMP WEB and Start Server.
cp  php.ini-development   /usr/local/php5/etc/php.ini
cp  /usr/local/php5/etc/php-fpm.conf.default  /usr/local/php5/etc/php-
fpm.conf
cp sapi/fpm/init.d.php-fpm /etc/init.d/php-fpm
chmod o+x /etc/init.d/php-fpm
/etc/init.d/php-fpm start
echo "
worker_processes  1;
events {
    worker_connections  1024;
}
http {
    include      mime.types;
    default_type  application/octet-stream;
    sendfile       on;
    keepalive_timeout  65;
    server {
        listen      80;
```

```
        server_name  localhost;
        location / {
            root   html;
            fastcgi_pass   127.0.0.1:9000;
            fastcgi_index  index.php;
            fastcgi_param  SCRIPT_FILENAME  \$document_root\$fastcgi_script_
name;
            include       fastcgi_params;
        }
    }
}" >/usr/local/nginx/conf/nginx.conf

echo "
<?php
phpinfo();
?>">/usr/local/nginx/html/index.php
/usr/local/nginx/sbin/nginx -s reload
fi
```

2.6　Shell 编程 MySQL 主从复制脚本

MySQL 数据库服务器主要应用于动态网站，存放网站必要的数据，例如订单、交易、员工表、薪资等记录。为了实现数据备份，需引入 MySQL 主从架构，MySQL 主从架构脚本可以实现自动化安装、配置和管理。

Shell 脚本实现服务器 MySQL 一键 YUM 安装配置，编写思路如下：

（1）MySQL 主库的操作。

① 在主库上安装 MySQL，并设置参数 server-id、bin-log。

② 授权复制同步的用户，对客户端授权。

③ 确认 bin-log 文件名及 position 位置点。

（2）MySQL 从库的操作。

① 在从库上安装 MySQL，设置参数 server-id。

② change master：指定主库和 bin-log 名和 position。

③ start slave：启动从库 I/O 线程。

④ show slave status\G：查看主从的状态。

Shell 脚本实现服务器 MySQL 一键 YUM 安装配置，需要提前手动授权主库免密码登录从库服务器，相关代码如下：

```
#!/bin/bash
#Auto install Mysql AB Replication
#By author jfedu.net 2021
#Define Path variables
MYSQL_SOFT="mysql mysql-server mysql-devel php-mysql mysql-libs"
NUM='rpm -qa |grep -i mysql |wc -l'
INIT="/etc/init.d/mysqld"
CODE=$?
#Mysql To Install 2021
if [ $NUM -ne 0 -a -f $INIT ];then
    echo -e "\033[32mThis Server Mysql already Install.\033[0m"
    read -p "Please ensure yum remove Mysql Server,YES or NO": INPUT
    if [ $INPUT == "y" -o $INPUT == "yes" ];then
        yum remove $MYSQL_SOFT -y ;rm -rf /var/lib/mysql /etc/my.cnf
        yum install $MYSQL_SOFT -y
    else
        echo
    fi
else
    yum remove $MYSQL_SOFT -y ;rm -rf /var/lib/mysql /etc/my.cnf
    yum install $MYSQL_SOFT -y
    if [ $CODE -eq 0 ];then
        echo -e "\033[32mThe Mysql Install Successfully.\033[0m"
    else
        echo -e "\033[32mThe Mysql Install Failed.\033[0m"
        exit 1
    fi
fi
my_config(){
cat >/etc/my.cnf<<EOF
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0
log-bin=mysql-bin
server-id = 1
auto_increment_offset=1
```

```
auto_increment_increment=2
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
EOF
}
my_config
/etc/init.d/mysqld restart
ps -ef |grep mysql
MYSQL_CONFIG(){
#Master Config Mysql
mysql -e "grant  replication  slave  on *.* to  'tongbu'@'%'  identified by
'123456';"
MASTER_FILE='mysql -e "show master status;"|tail -1|awk '{print $1}''
MASTER_POS='mysql -e "show master status;"|tail -1|awk '{print $2}''
MASTER_IPADDR='ifconfig eth0|grep "Bcast"|awk '{print $2}'|cut -d: -f2'
read -p "Please Input Slave IPaddr: " SLAVE_IPADDR
#Slave Config Mysql
ssh -l root $SLAVE_IPADDR "yum remove $MYSQL_SOFT -y ;rm -rf /var/lib/mysql
/etc/my.cnf ;yum install $MYSQL_SOFT -y"
ssh -l root $SLAVE_IPADDR "$my_config"
#scp -r /etc/my.cnf root@192.168.111.129:/etc/
ssh -l root $SLAVE_IPADDR "sed -i 's#server-id = 1#server-id = 2#g'
/etc/my.cnf"
ssh -l root $SLAVE_IPADDR "sed -i '/log-bin=mysql-bin/d' /etc/my.cnf"
ssh -l root $SLAVE_IPADDR "/etc/init.d/mysqld restart"
ssh -l root $SLAVE_IPADDR "mysql -e \"change master to master_host=
'$MASTER_IPADDR',master_user='tongbu',master_password='123456',master_
log_file='$MASTER_FILE',master_log_pos=$MASTER_POS;\""
ssh -l root $SLAVE_IPADDR "mysql -e \"slave start;\""
ssh -l root $SLAVE_IPADDR "mysql -e \"show slave status\G;\""
}

read -p "Please ensure your Server is Master and you will config mysql
Replication?yes or no": INPUT
if [ $INPUT == "y" -o $INPUT == "yes" ];then
   MYSQL_CONFIG
else
   exit 0
fi
```

2.7 Shell 编程修改 IP 及主机名脚本

在企业中，服务器 IP 地址系统通过自动化工具安装，IP 均是自动获取的，而服务器要求固定的静态 IP，手动配置上百台服务器的静态 IP 是不可取的，可以利用 Shell 脚本自动修改 IP、主机名等信息。

Shell 脚本可实现服务器 IP、主机名自动修改及配置，编写思路如下：

（1）静态 IP 修改。

（2）动态 IP 修改。

（3）根据 IP 生成主机名并配置。

（4）修改 DNS 域名解析。

相关代码如下：

```
#!/bin/bash
#Auto Change ip netmask gateway scripts
#By author jfedu.net 2021
#Define Path variables
ETHCONF=/etc/sysconfig/network-scripts/ifcfg-eth0
HOSTS=/etc/hosts
NETWORK=/etc/sysconfig/network
DIR=/data/backup/'date +%Y%m%d'
NETMASK=255.255.255.0
echo "----------------------------"
judge_ip(){
    read -p "Please enter ip Address,example 192.168.0.11 ip": IPADDR
    echo $IPADDR|grep -v "[Aa-Zz]"|grep --color -E "([0-9]{1,3}\.){3}
[0-9]{1,3}"
}
count_ip(){
    count=('echo $IPADDR|awk -F. '{print $1,$2,$3,$4}'')
    IP1=${count[0]}
    IP2=${count[1]}
    IP3=${count[2]}
    IP4=${count[3]}
}
ip_check()
{
judge_ip
```

```
while [ $? -ne 0 ]
do
   judge_ip
done
count_ip
while [ "$IP1" -lt 0 -o "$IP1" -ge 255 -o "$IP2" -ge 255 -o "$IP3" -ge 255
-o "$IP4" -ge 255 ]
do
   judge_ip
   while [ $? -ne 0 ]
   do
      judge_ip
   done
   count_ip
done
}
change_ip()
{
if [ ! -d $DIR ];then
    mkdir -p $DIR
fi
echo "The Change ip address to Backup Interface eth0"
cp $ETHCONF  $DIR
grep "dhcp"  $ETHCONF
if [ $? -eq 0 ];then
   read -p "Please enter ip Address:" IPADDR
   sed -i 's/dhcp/static/g' $ETHCONF
   echo -e "IPADDR=$IPADDR\nNETMASK=$NETMASK\nGATEWAY='echo $IPADDR|awk
-F. '{print $1"."$2"."$3}''.2" >>$ETHCONF
   echo "The IP configuration success. !"
else
   echo -n  "Static IP has been configured,please confirm whether to
modify,yes or No":
   read i
fi
if  [ "$i" == "y" -o "$i" == "yes" ];then
   ip_check
   sed -i -e '/IPADDR/d' -e '/NETMASK/d' -e '/GATEWAY/d' $ETHCONF
   echo -e "IPADDR=$IPADDR\nNETMASK=$NETMASK\nGATEWAY='echo $IPADDR|awk
-F. '{print $1"."$2"."$3}''.2" >>$ETHCONF
   echo "The IP configuration success. !"
```

```
    echo
else
    echo "Static IP already exists,please exit."
    exit $?
fi
}
change_hosts()
{

if [ ! -d $DIR ];then
    mkdir -p $DIR
fi
cp $HOSTS $DIR
ip_check
host=' echo $IPADDR|sed 's/\./-/g'|awk '{print "BJ-IDC-"$0"-jfedu.net"}''
cat $HOSTS |grep "$host"
if [ $? -ne 0 ];then
    echo "$IPADDR    $host" >> $HOSTS
    echo "The hosts modify success "
fi
grep "$host" $NETWORK
if [ $? -ne 0 ];then
    sed -i "s/^HOSTNAME/#HOSTNAME/g" $NETWORK
    echo "NETWORK=$host" >>$NETWORK
    hostname $host;su
fi
}
PS3="Please Select configuration ip or configuration host:"
select i in  "modify_ip" "modify_hosts" "exit"
do
    case $i in
            modify_ip)
            change_ip
       ;;
            modify_hosts)
            change_hosts
       ;;
            exit)
            exit
       ;;
            *)
```

```
            echo -e "1) modify_ip\n2) modify_ip\n3)exit"
    esac

done
```

2.8　Shell 编程 Zabbix 安装配置脚本

Zabbix 是一款分布式监控系统，基于 C/S 模式，需在服务器安装 Zabbix_server，在客户端安装 Zabbix_agent。通过 Shell 脚本可以更快速地实现该需求。

Shell 脚本可实现 Zabbix 服务器端和客户端自动安装，编写思路如下：

（1）确定 Zabbix 软件的版本源码安装路径，启用服务器和代理服务器。

（2）cp zabbix_agentd 启动进程，–/etc/init.d/zabbix_agentd 给执行权限。

（3）配置 zabbix_agentd.conf 文件，指定 server IP 变量。

（4）指定客户端的 Hostname 可以等于客户端 IP 地址。

（5）启动 zabbix_agentd 服务，创建 zabbix user。

相关代码如下：

```
#!/bin/bash
#Auto install zabbix server and client
#By author jfedu.net 2021
#Define Path variables
ZABBIX_SOFT="zabbix-4.0.26.tar.gz"
INSTALL_DIR="/usr/local/zabbix/"
SERVER_IP="192.168.111.128"
IP='ifconfig|grep Bcast|awk '{print $2}'|sed 's/addr://g''
SERVER_INSTALL(){
yum -y install curl curl-devel net-snmp net-snmp-devel perl-DBI
groupadd zabbix ;useradd -g zabbix zabbix;usermod -s /sbin/nologin zabbix
tar -xzf $ZABBIX_SOFT;cd 'echo $ZABBIX_SOFT|sed 's/.tar.*//g''
./configure  --prefix=/usr/local/zabbix --enable-server --enable-agent
--with-mysql --enable-ipv6 --with-net-snmp --with-libcurl &&make install
if [ $? -eq 0 ];then
    ln -s /usr/local/zabbix/sbin/zabbix_* /usr/local/sbin/
fi
cd - ;cd zabbix-4.0.26
cp  misc/init.d/tru64/{zabbix_agentd,zabbix_server}  /etc/init.d/ ;chmod
o+x /etc/init.d/zabbix_*
mkdir -p /var/www/html/zabbix/;cp -a  frontends/php/*  /var/www/html/zabbix/
```

```
#config zabbix server
cat >$INSTALL_DIR/etc/zabbix_server.conf<<EOF
LogFile=/tmp/zabbix_server.log
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=123456
EOF
#config zabbix agentd
cat >$INSTALL_DIR/etc/zabbix_agentd.conf<<EOF
LogFile=/tmp/zabbix_agentd.log
Server=$SERVER_IP
ServerActive=$SERVER_IP
Hostname = $IP
EOF
#start zabbix agentd
/etc/init.d/zabbix_server restart
/etc/init.d/zabbix_agentd restart
/etc/init.d/iptables stop
setenforce 0
}
AGENT_INSTALL(){
yum -y install curl curl-devel net-snmp net-snmp-devel perl-DBI
groupadd zabbix ;useradd -g zabbix zabbix;usermod -s /sbin/nologin zabbix

tar -xzf $ZABBIX_SOFT;cd 'echo $ZABBIX_SOFT|sed 's/.tar.*//g''
./configure  --prefix=/usr/local/zabbix  --enable-agent&&make install
if [ $? -eq 0 ];then
   ln -s /usr/local/zabbix/sbin/zabbix_* /usr/local/sbin/
fi
cd - ;cd zabbix-4.0.26
cp  misc/init.d/tru64/zabbix_agentd  /etc/init.d/zabbix_agentd ;chmod o+x
/etc/init.d/zabbix_agentd
#config zabbix agentd
cat >$INSTALL_DIR/etc/zabbix_agentd.conf<<EOF
LogFile=/tmp/zabbix_agentd.log
Server=$SERVER_IP
ServerActive=$SERVER_IP
Hostname = $IP
EOF
#start zabbix agentd
/etc/init.d/zabbix_agentd restart
/etc/init.d/iptables stop
```

```
setenforce 0
}

read -p "Please confirm whether to install Zabbix Server,yes or no? " INPUT
if [ $INPUT == "yes" -o $INPUT == "y" ];then
    SERVER_INSTALL
else
    AGENT_INSTALL
fi
```

2.9　Shell 编程 Nginx 虚拟主机脚本

Nginx Web 服务器的最大特点在于 Nginx 常被用于负载均衡、反向代理，单台 Nginx 服务器配置多个虚拟主机时，使用 Shell 脚本更加高效。

Shell 脚本实现 Nginx 自动安装及虚拟主机的维护，编写思路如下：

（1）脚本指定参数 v1.jfedu.net。

（2）创建 v1.jfedu.net 同时创建目录/var/www/v1。

（3）将 Nginx 虚拟主机配置定向到新的目录。

（4）重复虚拟主机不再添加。

相关代码如下：

```
#!/bin/bash
#Auto config Nginx virtual Hosts
#By author jfedu.net 2021
#Define Path variables
NGINX_CONF="/usr/local/nginx/conf/"
NGINX_MAKE="--user=www --group=www --prefix=/usr/local/nginx --with-http_
stub_status_module --with-http_ssl_module"
NGINX_SBIN="/usr/local/nginx/sbin/nginx"
NGINX_INSTALL(){
#Install Nginx server
NGINX_FILE=nginx-1.16.0.tar.gz
NGINX_DIR='echo $NGINX_FILE|sed 's/.tar*.*//g''
if [ ! -e /usr/local/nginx/ -a ! -e /etc/nginx/ ];then
    pkill nginx
    wget -c http://nginx.org/download/$NGINX_FILE
    yum install pcre-devel pcre -y
    rm -rf $NGINX_DIR ;tar xf $NGINX_FILE
```

```
    cd $NGINX_DIR;useradd www;./configure $NGINX_MAKE
    make &&make install
    grep -vE "#|^$" $NGINX_CONF/nginx.conf >$NGINX_CONF/nginx.conf.swp
    \mv  $NGINX_CONF/nginx.conf.swp $NGINX_CONF/nginx.conf
    for i in 'seq 1 6';do sed -i '$d' $NGINX_CONF/nginx.conf;done
    echo "}" >>$NGINX_CONF/nginx.conf
    cd ../
fi
}
NGINX_CONFIG(){
#config tomcat nginx vhosts
grep "include domains" $NGINX_CONF/nginx.conf >>/dev/null
if [ $? -ne 0 ];then
    #sed -i '$d' $NGINX_CONF/nginx.conf
    echo -e "\ninclude domains/*;\n}" >>$NGINX_CONF/nginx.conf
    mkdir -p $NGINX_CONF/domains/
fi
VHOSTS=$1
ls $NGINX_CONF/domains/$VHOSTS>>/dev/null 2>&1
if [ $? -ne 0 ];then
    #cp -r xxx.jfedu.net $NGINX_CONF/domains/$VHOSTS
    #sed -i "s/xxx/$VHOSTS/g" $NGINX_CONF/domains/$VHOSTS
    cat>$NGINX_CONF/domains/$VHOSTS<<EOF
    #vhost server $VHOSTS
    server {
        listen      80;
        server_name  $VHOSTS;
        location / {
            root   /data/www/$VHOSTS/;
            index  index.html index.htm;
        }
      }
EOF
    mkdir -p /data/www/$VHOSTS/
    cat>/data/www/$VHOSTS/index.html<<EOF
    <html>
    <h1><center>The First Test Nginx page.</center></h1>
    <hr color="red">
    <h2><center>$VHOSTS</center></h2>
    </html>
EOF
```

```
    echo -e "\033[32mThe $VHOSTS Config success,You can to access http://
$VHOSTS/\033[0m"
    NUM='ps -ef |grep nginx|grep -v grep|grep -v auto|wc -l'
    $NGINX_SBIN -t >>/dev/null 2>&1
    if [ $? -eq 0 -a $NUM -eq 0 ];then
        $NGINX_SBIN
    else
        $NGINX_SBIN -t >>/dev/null 2>&1
        if [ $? -eq 0 ];then
            $NGINX_SBIN -s reload
        fi
    fi
else
    echo -e "\033[32mThe $VHOSTS has been config,Please exit.\033[0m"
fi
}
if [ -z $1 ];then
    echo -e "\033[32m--------------------\033[0m"
    echo -e "\033[32mPlease enter sh $0 xx.jf.com.\033[0m"
    exit 0
fi
NGINX_INSTALL
NGINX_CONFIG $1
```

2.10　Shell 编程 Nginx、Tomcat 脚本

Tomcat 用于发布 JSP Web 页面，根据企业实际需求，会在单台服务器配置多个 Tomcat 实例，同时手动将 Tomcat 创建后的实例加入 Nginx 虚拟主机，同时重启 Nginx。开发 Nginx、Tomcat 自动创建 Tomcat 实例及 Nginx 虚拟主机管理脚本能大大减轻人工的干预，实现快速交付。

Shell 脚本实现 Nginx 自动安装、虚拟主机及自动将 Tomcat 加入虚拟主机，编写思路如下：

（1）手动将 Tomcat 复制到脚本同一目录下（可自动修改）。

（2）手动修改 Tomcat 端口为 6001、7001、8001（可自动修改）。

（3）脚本指定参数 v1.jfedu.net。

（4）创建 v1.jfedu.net Tomcat 实例。

（5）修改 Tomcat 实例端口，保证 Port 唯一。

（6）将 Tomcat 实例加入 Nginx 虚拟主机。

（7）重复创建 Tomcat 实例，端口自动增加，并加入原 Nginx 虚拟主机，实现负载均衡。

Shell 脚本实现 Nginx 自动安装、虚拟主机及自动将 Tomcat 加入虚拟主机，相关代码如下：

```
#!/bin/bash
#Auto config Nginx and tomcat cluster
#By author jfedu.net 2021
#Define Path variables
NGINX_CONF="/usr/local/nginx/conf/"
install_nginx(){
    NGINX_FILE=nginx-1.10.2.tar.gz
    NGINX_DIR='echo $NGINX_FILE|sed 's/.tar*.*//g''
    wget -c http://nginx.org/download/$NGINX_FILE
    yum install pcre-devel pcre -y
    rm -rf $NGINX_DIR ;tar xf $NGINX_FILE
    cd $NGINX_DIR;useradd www;./configure --user=www --group=www --prefix=
/usr/local/nginx2 --with-http_stub_status_module --with-http_ssl_module
    make &&make install
    cd ../
}
install_tomcat(){
    JDK_FILE="jdk1.8.0_131.tar.gz"
    JDK_DIR='echo $JDK_FILE|sed 's/.tar.*//g''
    tar -xzf $JDK_FILE  ;mkdir -p /usr/java/ ;mv $JDK_DIR /usr/java/
    sed -i '/JAVA_HOME/d;/JAVA_BIN/d;/JAVA_OPTS/d' /etc/profile
    cat >> /etc/profile <<EOF
    export JAVA_HOME=/export/servers/$JAVA_DIR
    export JAVA_BIN=/export/servers/$JAVA_DIR/bin
    export PATH=\$JAVA_HOME/bin:\$PATH
    export CLASSPATH=.:\$JAVA_HOME/lib/dt.jar:\$JAVA_HOME/lib/tools.jar
    export JAVA_HOME JAVA_BIN PATH CLASSPATH
EOF
    source /etc/profile;java -version
    #install tomcat start
    ls tomcat
}
config_tomcat_nginx(){
    #config tomcat nginx vhosts
    grep "include domains" $NGINX_CONF/nginx.conf >>/dev/null
    if [ $? -ne 0 ];then
        sed -i '$d' $NGINX_CONF/nginx.conf
        echo -e "\ninclude domains/*;\n}" >>$NGINX_CONF/nginx.conf
        mkdir -p $NGINX_CONF/domains/
    fi
```

```
    VHOSTS=$1
    NUM='ls /usr/local/|grep -c tomcat'
    if [ $NUM -eq 0 ];then
       cp -r tomcat /usr/local/tomcat_$VHOSTS
       cp -r xxx.jfedu.net $NGINX_CONF/domains/$VHOSTS
       #sed -i "s/VHOSTS/$VHOSTS/g" $NGINX_CONF/domains/$VHOSTS
       sed -i "s/xxx/$VHOSTS/g" $NGINX_CONF/domains/$VHOSTS
       exit 0
    fi
    #--------------------------------
    #VHOSTS=$1
    VHOSTS_NUM='ls $NGINX_CONF/domains/|grep -c $VHOSTS'
    SERVER_NUM='grep -c "127" $NGINX_CONF/domains/$VHOSTS'
    SERVER_NUM_1='expr $SERVER_NUM + 1'
    rm -rf /tmp/.port.txt
    for i in 'find /usr/local/ -maxdepth 1 -name "tomcat*"';do
       grep "port" $i/conf/server.xml |egrep -v "\--|8080|SSLEnabled"|awk
'{print $2}'|sed 's/port=//g;s/\"//g'|sort -nr >>/tmp/.port.txt
    done
    MAX_PORT='cat /tmp/.port.txt|grep -v 8443|sort -nr|head -1'
    PORT_1='expr $MAX_PORT - 2000 + 1'
    PORT_2='expr $MAX_PORT - 1000 + 1'
    PORT_3='expr $MAX_PORT + 1'
    if [ $VHOSTS_NUM -eq 1 ];then
       read -p "The $VHOSTS is exists,You sure create mulit Tomcat for the
$VHOSTS? yes or no " INPUT
       if [ $INPUT == "YES" -o $INPUT == "Y" -o $INPUT == "yes" ];then
          cp -r tomcat /usr/local/tomcat_${VHOSTS}_${SERVER_NUM_1}
          sed -i "s/6001/$PORT_1/g" /usr/local/tomcat_${VHOSTS}_${SERVER_
NUM_1}/conf/server.xml
          sed -i "s/7001/$PORT_2/g" /usr/local/tomcat_${VHOSTS}_${SERVER_
NUM_1}/conf/server.xml
          sed -i "s/8001/$PORT_3/g" /usr/local/tomcat_${VHOSTS}_${SERVER_
NUM_1}/conf/server.xml
          sed -i "/^upstream/a     server 127.0.0.1:${PORT_2} weight=1
max_fails=2 fail_timeout=30s;" $NGINX_CONF/domains/$VHOSTS
          exit 0
       fi
       exit
    fi
    cp -r tomcat /usr/local/tomcat_$VHOSTS
       cp -r xxx.jfedu.net $NGINX_CONF/domains/$VHOSTS
       sed -i "s/VHOSTS/$VHOSTS/g" $NGINX_CONF/domains/$VHOSTS
```

```
        sed -i "s/xxx/$VHOSTS/g" $NGINX_CONF/domains/$VHOSTS
    sed -i "s/7001/${PORT_2}/g" $NGINX_CONF/domains/$VHOSTS
    #######config tomcat
        sed -i "s/6001/$PORT_1/g" /usr/local/tomcat_${VHOSTS}/conf/server.
xml
        sed -i "s/7001/$PORT_2/g" /usr/local/tomcat_${VHOSTS}/conf/server.
xml
        sed -i "s/8001/$PORT_3/g" /usr/local/tomcat_${VHOSTS}/conf/server.
xml
}
if [ ! -d $NGINX_CONF -o ! -d  /usr/java/$JDK_DIR ];then
    install_nginx
    install_tomcat
fi
config_tomcat_nginx $1
```

2.11 Shell 编程管理 Linux 用户和组脚本

Shell 脚本实现编程管理 Linux 用户和组脚本，编写思路如下：

（1）脚本支持创建普通用户。

（2）支持创建多个用户或者列表用户添加。

（3）支持 Linux 系统用户删除。

（4）支持 Linux 系统组删除。

（5）支持对某个用户修改密码。

相关代码如下：

```
#!/bin/bash
#2021 年 7 月 29 日 15:54:58
#auto manager linux user
#by author www.jfedu.net
########################
USR="$*"
if [ $UID -ne 0 ];then
    echo -e "\033[32m-----------------\033[0m"
    echo -e "\033[32mThe script must be executed using the root user.\033[0m"
    exit 1
fi
add_user(){
    read -p "Please enter the user name you need to create? " USR
```

```
    for USR in $USR
    do
       id $USR
       if [ $? -ne 0 ];then
          useradd -s /bin/bash $USR -d /home/$USR
          echo ${USR}_123456|passwd --stdin $USR
          if [ $? -eq 0 ];then
             echo -e "\033[32m-----------------\033[0m"
             echo -e "\033[32mThe $USR user created successfully\033[0m"
             echo -e "User,Password"
             echo -e "$USR,${USR}_123"
             echo
             tail -n 5 /etc/passwd
          fi
       else
          echo -e "\033[32m-----------------\033[0m"
          echo -e "\033[32mThis $USR user already exists, please exit\033[0m"
          exit 1
       fi
    done
}
add_user_list(){
    G
          useradd -s /bin/bash $USR -d /home/$USR
          echo ${USR}_123456|passwd --stdin $USR
          if [ $? -eq 0 ];then
             echo -e "\033[32m-----------------\033[0m"
             echo -e "\033[32mThe $USR user created successfully\033[0m"
             echo -e "User,Password"
             echo -e "$USR,${USR}_123"
             echo
             tail -n 5 /etc/passwd
          fi
       done

    else
       echo -e "\033[32m-----------------\033[0m"
       echo -e "\033[32mThe user list file must be entered. The reference
content format is as follows:\033[0m"
       echo "jfedu1"
       echo "jfedu2"
```

```
        echo "jfedu3"
        echo "jfedu4"
        echo "......"
    fi
}

remove_user(){
        for USR in $USR
        do
        userdel -r $USR
        groupdel $USR
                if [ $? -eq 0 ];then
                        echo -e "\033[32m-----------------\033[0m"
                        echo -e "\033[32mThe $USR user delete successfully\
033[0m"
                        echo
                        tail -n 5 /etc/passwd
                fi
        done
}

remove_group(){
        for USR in $USR
        do
        groupdel $USR
                if [ $? -eq 0 ];then
                        echo -e "\033[32m-----------------\033[0m"
                        echo -e "\033[32mThe $USR group delete successfully\
033[0m"
                        echo
                        tail -n 5 /etc/passwd
        else
           grep "$USR" /etc/group
           if [ $? -eq 0 ];then
              echo -e "\033[32m-----------------\033[0m"
              echo -e "\033[32mThe $USR group delete falied,cannot remove the
primary group of user $USR\033[0m"
              read -p "Are you sure you want to delete the $USR user? yes or
no " INPUT
              if [ $INPUT == "y" -o $INPUT == "Y" -o $INPUT == "yes" -o $INPUT
== "YES" ];then
                  userdel -r $USR
```

```
                groupdel $USR >>/dev/null 2>&1
                echo -e "\033[32m-----------------\033[0m"
                echo -e "\033[32mThe $USR user delete successfully\033[0m"
                echo -e "\033[32mThe $USR group delete successfully\033[0m"
            fi
        fi
            fi
    done
}

change_user_passwd(){
    read -p "Please enter your user name and new password: username password: "
INPUT
    if ['echo $INPUT|sed 's/ /\n/g'|wc -l' -eq 2 ];then
        USR='echo $INPUT|awk '{print $1}''
        PAS='echo $INPUT|awk '{print $2}''
            for USR in $USR
            do
            echo $PAS|passwd --stdin $USR
                if [ $? -eq 0 ];then
                        echo -e "\033[32m-----------------\033[0m"
            echo -e "\033[32mThe password of $USR user was modified
successfully\033[0m"
                        echo -e "User,Password"
                        echo -e "$USR,$PAS"
                        echo
                fi
            done
    fi
}

case $1 in
    1)
    add_user
    ;;
    2)
    add_user_list
    ;;
    3)
    remove_user
    ;;
```

```
    4)
    remove_group
    ;;
    5)
    change_user_passwd
    ;;
    *)
    echo "--------------------------------------------"
    echo -e "\033[34mWelcome to system user management scripts:\033[0m"
    echo -e "\033[32m1) add user\033[0m"
    echo -e "\033[32m2) add_user_list\033[0m"
    echo -e "\033[32m3) remove_user\033[0m"
    echo -e "\033[32m4) remove_group\033[0m"
    echo -e "\033[32m5) change_user_passwd\033[0m"
    echo -e "\033[32mUsage:{/bin/sh $0 1|2|3|4|5|help}\033[0m"
    echo "--------------------------------------------"
esac
```

2.12 Shell 编程 Vsftpd 虚拟用户管理脚本

Shell 脚本实现编程 Vsftpd 虚拟用户管理脚本，编写思路如下：

（1）实现单个用户随机添加。

（2）实现多个用户随机添加。

（3）实现文件列表批量用户添加。

（4）实现单个用户或者多个用户删除。

相关代码如下：

```
#!/bin/bash
#2021 年 8 月 18 日 21:32:13
#auto create vsftpd for virtual user
#by author www.jfedu.net
#########################
CONF_DIR="/etc/vsftpd"
VIR_USER="$*"
SYS_USER="ftpuser"
LOGIN_DB="vsftpd_login"

if [ $# -eq 0 ];then
```

```
    echo -e "\033[32m---------------------\033[0m"
    echo -e "\033[32mUsage:{/bin/sh $0 jfedu001 jfedu002|jfedu003}\033[0m"
    exit 0
fi

if [ ! -f $CONF_DIR/vsftpd.conf ];then
    yum install vsftpd* db4* -y
else
    continue
fi

#for i in 'echo $VIR_USER'
echo $VIR_USER|sed 's/ /\n/g' >list.txt
while read i
do
grep "$i" $CONF_DIR/${SYS_USER}s.txt
if [ $? -ne 0 ];then
cat>>$CONF_DIR/${SYS_USER}s.txt<<EOF
$i
pwd_$i
EOF
fi
done <list.txt

db_load -T -t hash -f $CONF_DIR/${SYS_USER}s.txt $CONF_DIR/$LOGIN_DB.db
chmod 700 $CONF_DIR/${SYS_USER}s.txt
chmod 700 $CONF_DIR/$LOGIN_DB.db

cat>/etc/pam.d/vsftpd<<EOF
auth    sufficient     /lib64/security/pam_userdb.so     db=$CONF_DIR/
$LOGIN_DB
account sufficient     /lib64/security/pam_userdb.so     db=$CONF_DIR/
$LOGIN_DB
EOF
useradd -s /sbin/nologin $SYS_USER

grep "guest_"  $CONF_DIR/vsftpd.conf
if [ $? -ne 0 ];then
cat>>$CONF_DIR/vsftpd.conf<<EOF
guest_enable=YES
guest_username=$SYS_USER
```

```
pam_service_name=vsftpd
user_config_dir=$CONF_DIR/vsftpd_user_conf
virtual_use_local_privs=YES
EOF
fi

#for j in 'echo $VIR_USER'
while read j
do
mkdir -p $CONF_DIR/vsftpd_user_conf/
cat>$CONF_DIR/vsftpd_user_conf/$j <<EOF
local_root=/home/$SYS_USER/$j
write_enable=YES
anon_world_readable_only=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
EOF
mkdir -p /home/$SYS_USER/$j/
done < list.txt
chown -R $SYS_USER.$SYS_USER /home/$SYS_USER
service vsftpd restart
```

2.13 Shell 编程 Apache 多版本软件安装脚本

Shell 实现编程 Apache 多版本软件安装脚本，编写思路如下：

（1）安装不同的 Apache 版本。

（2）检测系统是否已经存在，是否可以覆盖版本。

（3）启动 Apache，并且测试访问。

相关代码如下：

```
#!/bin/bash
#2021 年 6 月 16 日 10:33:13
#by author jfedu.net
#auto install apache and vhosts
##################
H_URL="http://mirror.bit.edu.cn/apache/httpd/"
APR_URL="http://mirrors.hust.edu.cn/apache/apr/"
H_SOFT="httpd-2.4.25.tar.bz2"
APR_SOFT="apr-1.6.2.tar.bz2"
```

```
APR_UTIL_SOFT="apr-util-1.6.0.tar.bz2"
APACHE_DIR="/usr/local/apache2/"
VHOST_FILES="httpd-vhosts.conf"
DOMAINS="$1"
NUM1='grep -c "^Include conf/extra/httpd-vhosts.conf" $APACHE_DIR/conf/
httpd.conf'
NUM2=$(grep -c "$DOMAINS" $APACHE_DIR/conf/extra/httpd-vhosts.conf)

if [ $# -eq 0 ];then
    echo -e "\033[32m-------------------------\033[0m"
    echo -e "\033[32mUsage:{Please Enter $0 www.jf1.com|www.jf2.com}\033[0m"
    exit 0
fi

if [ ! -d $APACHE_DIR ];then
    wget -c  $H_URL/$H_SOFT
    wget -c  $APR_URL/$APR_SOFT
    wget -c  $APR_URL/$APR_UTIL_SOFT
    #Install apr for apache for
    tar -jxvf $APR_SOFT
    cd apr-1.6.2
    ./configure --prefix=/usr/local/apr
    make
    make install
    #Install apr-util for apache for
    cd ..
    tar -jxvf $APR_UTIL_SOFT
    cd apr-util-1.6.0
    ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr/
    make
    make install

    #Install apache
    cd ..
    tar -jxvf $H_SOFT
    cd httpd-2.4.25
    ./configure --prefix=$APACHE_DIR/ --with-apr=/usr/local/apr --with-apr-
util=/usr/local/apr-util
    make
    make install
    pkill httpd
```

```
    pkill nginx
    $APACHE_DIR/bin/apachectl start
fi

#config vhosts for apache
if [ $NUM1 -eq 0 ];then
    echo "Include conf/extra/$VHOST_FILES" >>$APACHE_DIR/conf/httpd.conf
fi
touch $APACHE_DIR/conf/extra/$VHOST_FILES


if [ $NUM2 -eq 0 ];then
cat >>$APACHE_DIR/conf/extra/$VHOST_FILES<<EOF
<VirtualHost *:80>
    ServerAdmin support@jfedu.net
    DocumentRoot "$APACHE_DIR/htdocs/$DOMAINS"
    ServerName $DOMAINS
    ErrorLog "logs/${DOMAINS}_error_log"
    CustomLog "logs/${DOMAINS}_access_log" common
</VirtualHost>
EOF

mkdir -p $APACHE_DIR/htdocs/$DOMAINS
touch $APACHE_DIR/htdocs/$DOMAINS/index.html
cat >$APACHE_DIR/htdocs/$DOMAINS/index.html<<EOF
<html><body>
<h1>$DOMAINS It works!</h1>
<h1><font color=\"red\">$DOMAINS</font></h1>
</body></html>
EOF
$APACHE_DIR/bin/apachectl restart
fi
```

2.14 Shell 编程局域网 IP 探活脚本

Shell 实现编程局域网 IP 探活脚本，编写思路如下：

（1）Shell 脚本支持指定特定的网段。

（2）对特定的网段进行探活。

（3）将存活的 IP 地址写入存活的列表。

（4）将不存活的 IP 地址写入不存活的列表。

相关代码如下：

```
#!/bin/bash
#2021 年 7 月 29 日 15:54:58
#auto ping check IP
#by author www.jfedu.net
########################
INPUT="0"
IP_LIST="$*"
RES_FILE1="/tmp/available.txt"
RES_FILE2="/tmp/unavailable.txt"
#Define check function 2021
check_lan(){
    read -p "Please enter the LAN segment,example 192.168.1.0 (Netmask/24):
" INPUT
       if ['echo $INPUT|sed 's/ /\n/g'|wc -l' -ne 0 ];then
              for IP in $(seq 1 254)
              do
          IP_PREFIX=$(echo $INPUT|awk -F\. '{print $1"."$2"."$3"."}')
                    ping -c 2 -W1 ${IP_PREFIX}$IP >/dev/null 2>1
                    if [ $? -eq 0 ];then
                           echo "${IP_PREFIX}$IP is up."
                           echo "${IP_PREFIX}$IP" >> $RES_FILE1
                    else
                           echo "${IP_PREFIX}$IP is down."
                           echo "${IP_PREFIX}$IP" >> $RES_FILE2
                    fi
              done
              echo -e "\033[32m-------------------------\033[0m"
              echo -e "\033[32mPlease check the following files:\033[0m"
              echo "Available IP addresses: $RES_FILE1"
              echo "Unavailable IP addresses: $RES_FILE2"
              echo
       fi
}

check_list()
{
    read -p "Please enter the IP list to be checked,example list.txt: " INPUT
```

```
    if [ ! -z $INPUT ];then
        for IP in $(cat $INPUT)
        do
               ping -c 2 -W1 $IP >/dev/null 2>1
               if [ $? -eq 0 ];then
                  echo "$IP is up."
                     echo $IP >> $RES_FILE1
               else
                  echo "$IP is down."
                     echo $IP >> $RES_FILE2
              fi
        done
        echo -e "\033[32m-------------------------\033[0m"
        echo -e "\033[32mPlease check the following files:\033[0m"
        echo "Available IP addresses: $RES_FILE1"
        echo "Unavailable IP addresses: $RES_FILE2"
        echo
    fi
}

check_ip(){
read -p "Please enter the IP to be checked,example 1.1.1.1 | 1.1.1.2: " INPUT
for INPUT in 'echo $INPUT'
do
    while true
    do
        echo $INPUT|grep -E "\<([0-9]{1,3}\.){3}[0-9]{1,3}\>"
        if [ $? -eq 0 ];then
            IP=('echo $INPUT|sed 's/\./ /g'')
            IP1='echo ${IP[0]}'
            IP2='echo ${IP[1]}'
            IP3='echo ${IP[2]}'
            IP4='echo ${IP[3]}'
            if [ $IP1 -gt 0 -a $IP1 -le 255 -a $IP2 -ge 0 -a $IP2 -le 255 -a
$IP3 -ge 0 -a $IP3 -le 255 -a $IP4 -ge 0 -a $IP4 -lt 255 ];then
                    if ['echo $INPUT|sed 's/ /\n/g'|wc -l' -ne 0 ];then
                           for IP in $(echo $INPUT)
                           do
                                  ping -c 2 -W1 $IP >/dev/null 2>1
                                  if [ $? -eq 0 ];then
                                         echo "$IP is up."
                                         echo $IP >> $RES_FILE1
```

```
                                    else
                                           echo "$IP is down."
                                           echo $IP >> $RES_FILE2
                                    fi
                            done
                            echo -e "\033[32m------------------------\033[0m"
                            echo -e "\033[32mPlease check the following
files:\033[0m"
                            echo "Available IP addresses: $RES_FILE1"
                            echo "Unavailable IP addresses: $RES_FILE2"
                            echo
                    fi
            break;
            else
                read -p "Please Enter server IP address:" INPUT
            fi
        else
            read -p "Please Enter server IP address:" INPUT
        fi
    done
done
}

case $1 in
    1)
    check_lan
    ;;
    2)
    check_ip
    ;;
    3)
    check_list
    ;;
    *)
    echo "-------------------------------------------"
    echo -e "\033[34mWelcome to LAN live scripts:\033[0m"
    echo -e "\033[32m1) check_lan\033[0m"
    echo -e "\033[32m2) check_ip\033[0m"
    echo -e "\033[32m3) check_list\033[0m"
    echo -e "\033[32mUsage:{/bin/sh $0 1|2|3|4|5|help}\033[0m"
    echo "-------------------------------------------"
esac
```

2.15 Shell 编程 Apache 虚拟主机管理脚本

Shell 实现编程 Apache 虚拟主机管理脚本，编写思路如下：

（1）检测 Apache 是否安装，并测试访问。

（2）如果已经安装，则直接添加虚拟主机。

（3）支持单个虚拟主机添加。

（4）支持多个虚拟主机添加。

（5）支持单个或多个虚拟主机删除。

相关代码如下：

```
#!/bin/bash
#2017 年 6 月 14 日 21:27:31
#auto config httpd vhosts
#by author jfedu.net
#####################
APACHE_SOFT="httpd httpd-devel httpd-tools"
BACK_DIR=/data/backup/'date +%F'
HTTP_DIR="/etc/httpd/conf"
HTTP_FILES="httpd.conf"
VHOSTS_CONF="vhosts.conf"
NUM1=$(grep -c "$VHOSTS_CONF" $HTTP_DIR/$HTTP_FILES)
NUM2=$(grep -c "NameVirtualHost" $HTTP_DIR/$VHOSTS_CONF)
DOMAIN="$1"

if [ $# -eq 0 ];then
   echo -e "\033[32m-----------------\033[0m"
   echo -e "\033[32mUsage:{Please Enter sh $0 www.jf1.com|www.jf2.com}\
033[0m"
   exit 0
fi

yum install $APACHE_SOFT -y
mkdir  -p  $BACK_DIR
cp  -a  $HTTP_DIR/$HTTP_FILES $BACK_DIR
touch $HTTP_DIR/$VHOSTS_CONF

if [ -z $NUM1 ];then
```

```
    NUM1=0
fi
if [ $NUM1 -eq 0 ];then
    echo "Include  conf/$VHOSTS_CONF" >>$HTTP_DIR/$HTTP_FILES
fi

if [ -z $NUM2 ];then
    NUM2=0
fi
if [ $NUM2 -eq 0 ];then
    echo "NameVirtualHost *:80" >>$HTTP_DIR/$VHOSTS_CONF
fi

NUM3='grep -c "$DOMAIN" /etc/httpd/conf/vhosts.conf'
if [ $NUM3 -eq 0 ];then
echo "
<VirtualHost *:80>
    ServerAdmin wgkgood@163.com
    DocumentRoot  \"/data/webapps/$DOMAIN\"
    ServerName  $DOMAIN
  <Directory \"/data/webapps/$DOMAIN\">
    AllowOverride All
    Options -Indexes FollowSymLinks
    Order allow,deny
    Allow from all
  </Directory>
    ErrorLog  logs/error_log
    CustomLog logs/access_log common
</VirtualHost>
" >>$HTTP_DIR/$VHOSTS_CONF
fi
```

2.16　Shell 编程实现 Apache 高可用脚本

Shell 编程实现 Apache 高可用脚本，编写思路如下：

（1）部署两台 Apache 服务器，发布 Web 测试页面。

（2）通过两台 IP 均可以实现访问 Web 网页。

（3）增加第三个 IP，称为 VIP，可以绑定至某一台服务器。

（4）实现访问 VIP 即可访问一台 Apache Web 服务器。

（5）当该 Apache Web 服务器宕机，VIP 自动切换至另外一台 Web 服务器。

（6）时刻保证不管哪台 Apache Web 服务器宕机，VIP 均可以访问 Web 服务器。

相关代码如下：

```
#!/bin/bash
#2021 年 11 月 7 日 20:42:50
#auto change service VIP
#by author www.jfedu.net
########################
ETH_NAME="ens33:1"
APA_VIP="192.168.1.188"
APA_MASK="255.255.255.0"
ETH_DIR="/etc/sysconfig/network-scripts"
APA_NUM='ps -ef|grep httpd|grep -v grep|grep -v check|wc -l'
start(){
while sleep 4
do
if [ $APA_NUM -eq 0 ];then
    ifdown $ETH_NAME
    exit 0
else
    ping -c 2 $APA_VIP >/dev/null 2>&1
    if [ $? -ne 0 ];then
cat>$ETH_DIR/ifcfg-$ETH_NAME<<EOF
TYPE="Ethernet"
BOOTPROTO="static"
DEVICE="$ETH_NAME"
IPADDR=$APA_VIP
NETMASK=$APA_MASK
ONBOOT="yes"
EOF
    ifup $ETH_NAME
    fi
fi
date
done
}

stop(){
```

```
    ifdown $ETH_NAME
    rm -rf $ETH_DIR/ifcfg-$ETH_NAME
}

case $1 in
    start)
    start
    ;;
    stop)
    stop
    ;;
    *)
    echo -e "\033[32m------------------\033[0m"
    echo -e "\033[32mUsage: /bin/sh $0 {start|stop|help}\033[0m"
       exit 1
esac
```

2.17　Shell 编程拒绝黑客攻击 Linux 脚本

Shell 编程拒绝黑客攻击 Linux 脚本，编写思路如下：

（1）登录服务器日志/var/log/secure。

（2）检查日志中认证失败的行并打印其 IP 地址。

（3）将 IP 地址写入 Linux 服务器黑名单文件或防火墙。

（4）禁止该 IP 访问服务器 SSH 22 端口。

（5）将脚本加入 Crontab 实现自动禁止恶意 IP。

相关代码如下：

```
#!/bin/bash
#Auto drop ssh failed IP address
#By author jfedu.net 2021
#Define Path variables
SEC_FILE=/var/log/secure
IP_ADDR='awk '{print $0}'  /var/log/secure|grep -i  "fail"| egrep -o
"([0-9]{1,3}\.){3}[0-9]{1,3}" | sort -nr | uniq -c |awk '$1>=1 {print $2}''
DENY_CONF=/etc/hosts.deny
TM1='date +%Y%m%d%H%M'
DENY_IP="/tmp/2h_deny_ip.txt"
```

```
echo
cat <<EOF
++++++++++++++welcome to use ssh login drop failed ip+++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++----------------------------------------+++++++++++++++++
EOF
echo
for ((j=0;j<=2;j++)) ;do echo -n "-";sleep 1 ;done
echo
for i in 'echo $IP_ADDR'
do
    cat $DENY_CONF |grep $i >/dev/null 2>&1
    if  [ $? -ne 0 ];then
        grep "$i" $DENY_IP>>/dev/null 2>&1
        if  [ $? -eq 0 ];then
           TM3='date +%Y%m%d%H%M'
            IP1='awk -F"[#:]" '/'$i'/ {print $2,$4}' $DENY_IP|awk '{if
('$TM3'>=$2+2) print $1}''
           if [ ! -z $IP1 ];then
              echo "sshd:$IP1:deny #$TM1" >>$DENY_CONF
              sed -i "/$IP1/d"  $DENY_IP
           fi
       else
              echo "sshd:$i:deny #$TM1" >>$DENY_CONF
       fi
   fi
done

#Allow IP to access
TM2='date +%Y%m%d%H%M'
IP2='awk -F"[#:]" '/sshd/ {print $2,$4}' $DENY_CONF|awk '{if('$TM2'>=$2+2)
print $1}''
for k in 'echo $IP2'
do
   echo $k
   sed -i "/$k/d"  $DENY_CONF
   echo "sshd:$k:deny #$TM2" >>$DENY_IP
done
```

2.18　Shell 编程 mysqldump 数据库自动备份脚本

Shell 编程 mysqldump 数据库自动备份脚本，编写思路如下：

（1）支持 MySQL 单个库备份。

（2）支持 MySQL 多个库备份。

（3）支持 MySQL 全数据库备份。

（4）支持 MySQL 数据库定期删除数据。

相关代码如下：

```
#!/bin/bash
#2020 年 7 月 6 日 22:21:18
#auto backup mysql database
#by author www.jfedu.net
########################
SQL_DB="$*"
SQL_USR="backup"
SQL_PWD="bak123456"
SQL_CMD="/usr/bin/mysqldump"
BAK_DIR="/data/backup/'date +%F'"
if [ $# -eq 0 ];then
    echo -e "\033[32m----------------\033[0m"
    echo -e "\033[32mUsage:{/bin/bash $0 jfedu001|jfedu002|all|help}\033[0m"
    exit
fi

if [ $UID -ne 0 ];then
    echo "Exec backup scripts,must to be use root."
    exit
fi
if [ ! -d $BAK_DIR ];then
    mkdir -p $BAK_DIR
fi

if [ $SQL_DB == "all" ];then
    for SQL_DB in $(/usr/bin/mysql -u$SQL_USR -p$SQL_PWD -e "show
databases;")
    do
        $SQL_CMD -u$SQL_USR -p$SQL_PWD $SQL_DB >$BAK_DIR/${SQL_DB}.sql
```

```
        if [ $? -eq 0 ];then
            echo -e "\033[32m----------------\033[0m"
            echo -e "\033[32mThe mysql database $SQL_DB backup successfully.\
033[0m"
            echo
            echo "ls -l $BAK_DIR/"
            ls -l $BAK_DIR/
        else
            echo "The mysql database $SQL_DB backup falied."
            rm -rf $BAK_DIR/${SQL_DB}.sql

        fi
    done
    exit
fi

for SQL_DB in $SQL_DB
do
    $SQL_CMD -u$SQL_USR -p$SQL_PWD $SQL_DB >$BAK_DIR/$SQL_DB.sql
    if [ $? -eq 0 ];then
        echo -e "\033[32m----------------\033[0m"
        echo -e "\033[32mThe mysql database $SQL_DB backup successfully.\
033[0m"
        echo
        echo "ls -l $BAK_DIR/"
        ls -l $BAK_DIR/
    else
        echo "The mysql database $SQL_DB backup falied."
        rm -rf $BAK_DIR/${SQL_DB}.sql
        while true
        do
            echo
            read -p "Please retry enter database name: " INPUT
            /usr/bin/mysql -u$SQL_USR -p$SQL_PWD -e "show databases;"|grep -ai
"$INPUT"
            if [ $? -eq 0 ];then
                break
            fi
    done
```

```
    fi
done
```

2.19　Shell 编程 MySQL 主从自动配置脚本

Shell 编程 MySQL 主从自动配置脚本，编写思路如下：

（1）主库上安装 MySQL，设置 server-id、bin-log。

（2）授权复制同步的用户，对客户端授权。

（3）确认 bin-log 文件名、position 位置点。

（4）从库上安装 MySQL，设置 server-id。

（5）change master：指定主库、bin-log 名和 position。

（6）start slave：启动从库 I/O 线程。

（7）show slave status\G：查看主从的状态。

相关代码如下：

```
#!/bin/bash
#auto make install Mysql AB Repliation
#by author jfudu.net wugk
#2015-11-16 change
MYSQL_SOFT="mariadb mariadb-server mariadb-devel mariadb-libs"
NUM='rpm -qa |grep -i mariadb |wc -l'
INIT="mariadb.service"
CODE=$?

#Mysql To Install 2015
if [ $NUM -ne 0 -a -f /usr/lib/systemd/system/$INIT ];then
    echo -e "\033[32mThis Server Mysql already Install.\033[0m"
    read -p "Please ensure yum remove Mysql Server,YES or NO": INPUT

    if [ $INPUT == "y" -o $INPUT == "yes" ];then
        yum remove $MYSQL_SOFT -y ;rm -rf /var/lib/mysql /etc/my.cnf
        yum install $MYSQL_SOFT -y
    else
        echo
    fi
else
    yum remove $MYSQL_SOFT -y ;rm -rf /var/lib/mysql /etc/my.cnf
```

```
    yum install $MYSQL_SOFT -y
    if [ $CODE -eq 0 ];then
       echo -e "\033[32mThe Mysql Install Successfully.\033[0m"
    else
       echo -e "\033[32mThe Mysql Install Failed.\033[0m"
       exit 1
    fi
fi


cat >/etc/my.cnf<<EOF
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0
log-bin=mysql-bin
server-id = 1
auto_increment_offset=1
auto_increment_increment=2
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
EOF

chown -R mysql.mysql /var/log/
mkdir -p /var/run/mysqld
chown -R mysql.mysql /var/run/mysqld
systemctl restart mariadb.service
ps -ef |grep mysql

MYSQL_CONFIG(){

#Master Config Mysql
mysql -e "grant  replication  slave  on *.* to  'tongbu'@'%'  identified by
'123456';"
MASTER_FILE='mysql -e "show master status;"|tail -1|awk '{print $1}''
MASTER_POS='mysql -e "show master status;"|tail -1|awk '{print $2}''

#MASTER_IPADDR='ifconfig eth0|grep "Bcast"|awk '{print $2}'|cut -d: -f2'
MASTER_IPADDR=$(ifconfig|grep "broadcast"|cut -d" " -f10)
```

```
read -p "Please Input Slave IPaddr: " SLAVE_IPADDR

#Slave Config Mysql
ssh -l root $SLAVE_IPADDR "yum remove $MYSQL_SOFT -y ;rm -rf /var/lib/mysql
/etc/my.cnf ;yum install $MYSQL_SOFT -y"
#ssh -l root $SLAVE_IPADDR "$my_config"
scp -r /etc/my.cnf root@$SLAVE_IPADDR:/etc/
ssh -l root $SLAVE_IPADDR "sed -i 's#server-id = 1#server-id = 2#g'
/etc/my.cnf"
ssh -l root $SLAVE_IPADDR "sed -i '/log-bin=mysql-bin/d' /etc/my.cnf"

ssh -l root $SLAVE_IPADDR "chown -R mysql.mysql /var/log/"
ssh -l root $SLAVE_IPADDR "mkdir -p /var/run/mysqld"
ssh -l root $SLAVE_IPADDR "chown -R mysql.mysql /var/run/mysqld"

ssh -l root $SLAVE_IPADDR "systemctl restart mariadb.service"
ssh -l root $SLAVE_IPADDR "mysql -e \"change master to master_host='$MASTER_
IPADDR',master_user='tongbu',master_password='123456',master_log_file=
'$MASTER_FILE',master_log_pos=$MASTER_POS;\""
ssh -l root $SLAVE_IPADDR "mysql -e \"slave start;\""
ssh -l root $SLAVE_IPADDR "mysql -e \"show slave status\G;\""
}

read -p "Please ensure your Server is Master and you will config mysql
Replication?yes or no": INPUT
if [ $INPUT == "y" -o $INPUT == "yes" ];then
   MYSQL_CONFIG
else
   exit 0
fi
```

2.20　Shell 编程部署 Tomcat 多实例脚本

Shell 编程部署 Tomcat 多实例脚本，编写思路如下：

（1）检测服务器是否部署 JDK 和 Tomcat。

（2）部署 Tomcat 实例至/usr/local/目录。

（3）Shell 脚本支持任意单个 Tomcat 实例添加并启动。

（4）Shell 脚本支持多个 Tomcat 实例添加并启动。

相关代码如下：

```
function config_tomcat_nginx(){
    #Install JAVA JDK
    TOMCAT_VER="8.0.50"
    JAVA_VER="1.8.0_131"
    JAVA_DIR="/usr/java"
    TOMCAT_DIR="/usr/local"
    JAVA_SOFT="jdk${JAVA_VER}.tar.gz"
    TOMCAT_SOFT="apache-tomcat-${TOMCAT_VER}.tar.gz"
    grep -ai "^export" /etc/profile|grep -ai "JAVA_HOME" >/dev/null
    if [ $? -ne 0 ];then
    #Install JAVA JDK
    ls -l $JAVA_SOFT
    tar -xzvf $JAVA_SOFT
    mkdir -p $JAVA_DIR/
    \mv jdk$JAVA_VER $JAVA_DIR/
    ls -l $JAVA_DIR/jdk$JAVA_VER/
    $JAVA_DIR/jdk$JAVA_VER/bin/java -version
    cat>>/etc/profile<<-EOF
    export JAVA_HOME=$JAVA_DIR/jdk$JAVA_VER
    export CLASSPATH=\$CLASSPATH:\$JAVA_HOME/lib:\$JAVA_HOME/jre/lib
    EOF
    source /etc/profile
    fi

    shift 1
    NUM='ls /usr/local/|grep -c tomcat'
    if [ $NUM -eq 0 ];then
        cp -r tomcat /usr/local/tomcat_$*
        exit 0
    fi
    #--------------------------------
    #VHOSTS=$1
    VHOSTS_NUM='ls $NGINX_CONF/domains/|grep -c $*'
    SERVER_NUM='grep -c "127" $NGINX_CONF/domains/$*'
    SERVER_NUM_1='expr $SERVER_NUM + 1'
    rm -rf /tmp/.port.txt
    for i in 'find /usr/local/ -maxdepth 1 -name "tomcat*"';do
        grep "port" $i/conf/server.xml |egrep -v "\--|8080|SSLEnabled"|awk
'{print $2}'|sed 's/port=//g;s/\"//g'|sort -nr >>/tmp/.port.txt
    done
    MAX_PORT='cat /tmp/.port.txt|grep -v 8443|sort -nr|head -1'
```

```
    PORT_1='expr $MAX_PORT - 2000 + 1'
    PORT_2='expr $MAX_PORT - 1000 + 1'
    PORT_3='expr $MAX_PORT + 1'
    if [ $*_NUM -eq 1 ];then
       read -p "The $* is exists,You sure create mulit Tomcat for the $*?
yes or no " INPUT
       if [ $INPUT == "YES" -o $INPUT == "Y" -o $INPUT == "yes" ];then
          cp -r tomcat /usr/local/tomcat_${VHOSTS}_${SERVER_NUM_1}
          sed -i "s/6001/$PORT_1/g" /usr/local/tomcat_${VHOSTS}_${SERVER_
NUM_1}/conf/server.xml
          sed -i "s/7001/$PORT_2/g" /usr/local/tomcat_${VHOSTS}_${SERVER_
NUM_1}/conf/server.xml
          sed -i "s/8001/$PORT_3/g" /usr/local/tomcat_${VHOSTS}_${SERVER_
NUM_1}/conf/server.xml
          sed -i "/^upstream/a     server 127.0.0.1:${PORT_2} weight=1
max_fails=2 fail_timeout=30s;" $NGINX_CONF/domains/$*
          exit 0
       fi
       exit
    fi
       cp -r tomcat /usr/local/tomcat_$*
       cp -r xxx.jfedu.net $NGINX_CONF/domains/$*
       sed -i "s/VHOSTS/$*/g" $NGINX_CONF/domains/$*
       sed -i "s/xxx/$*/g" $NGINX_CONF/domains/$*
       sed -i "s/7001/${PORT_2}/g" $NGINX_CONF/domains/$*
       #######config tomcat
       sed -i "s/6001/$PORT_1/g" /usr/local/tomcat_${VHOSTS}/conf/server.
xml
       sed -i "s/7001/$PORT_2/g" /usr/local/tomcat_${VHOSTS}/conf/server.
xml
       sed -i "s/8001/$PORT_3/g" /usr/local/tomcat_${VHOSTS}/conf/server.
xml

}
if [ ! -d $NGINX_CONF -o ! -d  /usr/java/$JDK_DIR ];then
    install_nginx
    install_tomcat
fi
config_tomcat_nginx $1
```

2.21 Shell 编程 Nginx 日志切割脚本

Shell 编程 Nginx 日志切割脚本，编写思路如下：

（1）支持指定日志文件或多个文件切割。

（2）日志文件支持按天切割。

（3）日志文件支持按小时切割。

（4）日志文件切割之后打包并上传至 FTP 服务器。

相关代码如下：

```
#!/bin/bash
#auto mv nginx log shell
#by author jfedu.net
NUM=$(date +%H%M%S)
echo 'date'
if [ $NUM == "000000" ];then
       LOG_DIR="/data/logs/linux_web/"
       TIME='date -d "-1 day" +%Y%m%d'
       echo -e "\033[32mPlease wait start cut shell scripts...\033[1m"
       sleep 2
       cd $LOG_DIR
       mv access.log access_${TIME}.log
       kill  -USR1  'cat /usr/local/nginx/nginx.pid'
       echo "----------------------------------------"
       echo "The Nginx log Cutting Successfully!"
fi
```

2.22 Shell 编程 Tomcat 实例和 Nginx 均衡脚本

Shell 编程 Tomcat 实例和 Nginx 均衡脚本，编写思路如下：

（1）检测服务器是否部署 Nginx、JDK 和 Tomcat。

（2）部署 Tomcat 实例至/usr/local/目录。

（3）Shell 脚本支持单个 Tomcat 实例添加并启动。

（4）Shell 脚本支持多个 Tomcat 实例添加并启动。

（5）Shell 脚本除了实现单个和多个 Tomcat 之外，将其 Tomcat 实例的端口加入 Nginx 虚拟

主机均衡。

（6）实现 Nginx 均衡多个虚拟主机域名，分别对应后端不同 Tomcat。

（7）Shell 脚本支持删除 Nginx 均衡和删除 Tomcat 实例（单个和多个）。

相关代码如下：

```
#!/bin/bash
#2020 年 11 月 19 日 15:50:38
#auto config nginx virtual
#by author www.jfedu.net
#########################
NGX_CNF="nginx.conf"
NGX_DIR="/usr/local/nginx"
NGX_YUM="yum install -y"
NGX_URL="http://nginx.org/download"
NGX_ARG="--user=www --group=www --with-http_stub_status_module"
function nginx_help(){
    echo -e "\033[33mNginx VIrtual Manager SHELL Scripts\033[0m"
    echo -e "\033[33m------------------------------\033[0m"
       echo -e "\033[33m1)-I New Install Nginx WEB Server.\033[0m"
       echo -e "\033[33m2)-U Update Install Nginx WEB Server.\033[0m"
       echo -e "\033[33m3)-A v1.jfedu.net|v2.jfedu.net v3.jfedu.net\033[0m"
       echo -e "\033[33m4)-D v1.jfedu.net|v2.jfedu.net v3.jfedu.net\033[0m"
       echo -e "\033[33m5)-T v1.jfedu.net|v2.jfedu.net v3.jfedu.net\033[0m"
       echo -e "\033[35mUsage:{/bin/bash $0 -I(Install) | -U(Update)| -A(Add)
| -D(Del) | -H(Help) -T(Tomcat)\033[0m"
       exit 0
}

function nginx_install(){
#Nginx Install Config
if [ $# -le 1 ];then
       nginx_help
fi
if [ ! -d ${NGX_DIR} ];then
    shift 1
    NGX_VER=$(echo $*)
    NGX_SOFT="nginx-${NGX_VER}.tar.gz"
    NGX_SRC=$(echo $NGX_SOFT|sed 's/.tar.*//g')
    NGX_CODE="src/core/nginx.h"
    echo -e "\033[33m------------------------------\033[0m"
```

```
    echo -e "\033[33mStart Nginx install Proccess...\033[0m"
    $NGX_YUM wget make gzip tar gcc gcc-c++ >>/dev/null 2>&1
    $NGX_YUM pcre pcre-devel zlib zlib-devel >>/dev/null 2>&1
    wget -c $NGX_URL/$NGX_SOFT
    tar -xzf $NGX_SOFT
    cd $NGX_SRC
    sed -i "s/$NGX_VER//g" $NGX_CODE
    sed -i 's/nginx\//JWS/g' $NGX_CODE
    sed -i 's/"NGX"/"JWS"/g' $NGX_CODE
    useradd -s /sbin/nologin www -M
    ./configure --prefix=${NGX_DIR}/ $NGX_ARG
    make -j4
    make -j4 install
    ${NGX_DIR}/sbin/nginx
    ps -ef|grep -aiwE nginx
    netstat -tnlp|grep -aiwE 80
    setenforce 0
    sed -i '/SELINUX/s/enforcing/disabled/g' /etc/sysconfig/selinux
    if [ $(uname -r|awk -F"[-|.]" '{print $1}') -ge "3" ];then
       firewall-cmd --add-port=80/tcp --permanent
       systemctl reload firewalld.service
    else
       iptables -t filter -A INPUT -m tcp -p tcp --dport 80 -j ACCEPT
       service iptables save
    fi
else
    echo -e "\033[32mThe Nginx WEB Already Install,Please Exit.\033[0m"
    echo -e "\033[33m----------------------\033[0m"
    echo "ls -l $NGX_DIR/"
    ls -l $NGX_DIR/

    echo -e "\033[33m----------------------\033[0m"
    while true
    do
    echo -e -n "\033[33mPlease ensure to retry Nginx WEB service,yes or no ?
\033[0m"
    read INPUT
    if [ -z $INPUT ];then
       continue
    fi
    if [ $INPUT == "yes" -o $INPUT == "YES" -o $INPUT == "y" ];then
```

```
        echo -e "--------------------------------"
        echo -e "Backup nginx to ${NGX_DIR}.bak,\mv $NGX_DIR ${NGX_DIR}.bak"
        \mv $NGX_DIR ${NGX_DIR}.bak
        shift 1
            NGX_VER=$(echo $*)
            NGX_SOFT="nginx-${NGX_VER}.tar.gz"
            NGX_SRC=$(echo $NGX_SOFT|sed 's/.tar.*//g')
            NGX_CODE="src/core/nginx.h"
            echo -e "\033[33m--------------------------------\033[0m"
            echo -e "\033[33mStart Nginx install Proccess...\033[0m"
            $NGX_YUM wget make gzip tar gcc gcc-c++ >>/dev/null 2>&1
            $NGX_YUM pcre pcre-devel zlib zlib-devel >>/dev/null 2>&1
            wget -c $NGX_URL/$NGX_SOFT
            tar -xzf $NGX_SOFT
            cd $NGX_SRC
            sed -i "s/$NGX_VER//g" $NGX_CODE
            sed -i 's/nginx\//JWS/g' $NGX_CODE
            sed -i 's/"NGX"/"JWS"/g' $NGX_CODE
            useradd -s /sbin/nologin www -M
            ./configure --prefix=${NGX_DIR}/ $NGX_ARG
            make -j4
            make -j4 install
            ${NGX_DIR}/sbin/nginx
            ps -ef|grep -aiwE nginx
            netstat -tnlp|grep -aiwE 80
            setenforce 0
            sed -i '/SELINUX/s/enforcing/disabled/g' /etc/sysconfig/selinux
            if [ $(uname -r|awk -F"[-|.]" '{print $1}') -ge "3" ];then
                   firewall-cmd --add-port=80/tcp --permanent
                   systemctl reload firewalld.service
            else
                   iptables -t filter -A INPUT -m tcp -p tcp --dport 80 -j ACCEPT
                   service iptables save
            fi
        break

    fi
    done
fi
}
```

```
function nginx_update(){
#Nginx Install Config
if [ $# -le 1 ];then
       nginx_help
fi
if [ ! -d ${NGX_DIR} ];then
    shift 1
    NGX_VER=$(echo $*)
    NGX_SOFT="nginx-${NGX_VER}.tar.gz"
    NGX_SRC=$(echo $NGX_SOFT|sed 's/.tar.*//g')
    NGX_CODE="src/core/nginx.h"
    echo -e "\033[33m--------------------------------\033[0m"
    echo -e "\033[33mStart Nginx install Proccess...\033[0m"
    $NGX_YUM wget make gzip tar gcc gcc-c++ >>/dev/null 2>&1
    $NGX_YUM pcre pcre-devel zlib zlib-devel >>/dev/null 2>&1
    wget -c $NGX_URL/$NGX_SOFT
    tar -xzf $NGX_SOFT
    cd $NGX_SRC
    sed -i "s/$NGX_VER//g" $NGX_CODE
    sed -i 's/nginx\//JWS/g' $NGX_CODE
    sed -i 's/"NGX"/"JWS"/g' $NGX_CODE
    useradd -s /sbin/nologin www -M
    ./configure --prefix=${NGX_DIR}/ $NGX_ARG
    make -j4
    make -j4 install
    ${NGX_DIR}/sbin/nginx
    ps -ef|grep -aiwE nginx
    netstat -tnlp|grep -aiwE 80
    setenforce 0
    sed -i '/SELINUX/s/enforcing/disabled/g' /etc/sysconfig/selinux
    if [ $(uname -r|awk -F"[-|.]" '{print $1}') -ge "3" ];then
       firewall-cmd --add-port=80/tcp --permanent
       systemctl reload firewalld.service
    else
       iptables -t filter -A INPUT -m tcp -p tcp --dport 80 -j ACCEPT
       service iptables save
    fi
else
    echo -e "\033[32mThe Nginx WEB Already Install,Please Exit.\033[0m"
    echo -e "\033[33m-----------------------\033[0m"
    echo "ls -l $NGX_DIR/"
```

```
    ls -l $NGX_DIR/

    echo -e "\033[33m----------------------\033[0m"
    while true
    do
    echo -e -n "\033[33mPlease ensure to Update Nginx WEB service,yes or no ?
\033[0m"
    read INPUT
    if [ -z $INPUT ];then
       continue
    fi
    if [ $INPUT == "yes" -o $INPUT == "YES" -o $INPUT == "y" ];then
       echo -e "------------------------------"
       echo -e "Backup nginx to ${NGX_DIR}.bak,\mv $NGX_DIR ${NGX_DIR}.bak"
       \mv $NGX_DIR ${NGX_DIR}.bak
       shift 1
          NGX_VER=$(echo $*)
          NGX_SOFT="nginx-${NGX_VER}.tar.gz"
          NGX_SRC=$(echo $NGX_SOFT|sed 's/.tar.*//g')
          NGX_CODE="src/core/nginx.h"
          echo -e "\033[33m------------------------------\033[0m"
          echo -e "\033[33mStart Nginx install Proccess...\033[0m"
          $NGX_YUM wget make gzip tar gcc gcc-c++ >>/dev/null 2>&1
          $NGX_YUM pcre pcre-devel zlib zlib-devel >>/dev/null 2>&1
          wget -c $NGX_URL/$NGX_SOFT
          tar -xzf $NGX_SOFT
          cd $NGX_SRC
          sed -i "s/$NGX_VER//g" $NGX_CODE
          sed -i 's/nginx\//JWS/g' $NGX_CODE
          sed -i 's/"NGX"/"JWS"/g' $NGX_CODE
          useradd -s /sbin/nologin www -M
          ./configure --prefix=${NGX_DIR}/ $NGX_ARG
          make -j4
          \mv ${NGX_DIR}/sbin/nginx ${NGX_DIR}/sbin/nginx.old
       \cp objs/nginx ${NGX_DIR}/sbin/
          ${NGX_DIR}/sbin/nginx
          ps -ef|grep -aiwE nginx
          netstat -tnlp|grep -aiwE 80
          setenforce 0
          sed -i '/SELINUX/s/enforcing/disabled/g' /etc/sysconfig/selinux
          if [ $(uname -r|awk -F"[-|.]" '{print $1}') -ge "3" ];then
```

```
                    firewall-cmd --add-port=80/tcp --permanent
                    systemctl reload firewalld.service
            else
                    iptables -t filter -A INPUT -m tcp -p tcp --dport 80 -j ACCEPT
                    service iptables save
            fi
        break

    fi
    done
fi
}

function virtual_add(){
    #Nginx Config Virtual Host
    if [ $# -le 1 ];then
        nginx_help
    fi
    cd ${NGX_DIR}/conf/
    grep -aiE "include domains" ${NGX_CNF} >>/dev/null 2>&1
    if [ $? -ne 0 ];then
        grep -aiE -vE "^$|#" ${NGX_CNF} > ${NGX_CNF}.swp
        \cp ${NGX_CNF}.swp ${NGX_CNF}
        sed -i '/server/,$d' ${NGX_CNF}
        echo -e -e "    include domains/*;\n}" >>${NGX_CNF}
        ${NGX_DIR}/sbin/nginx -t
        mkdir domains -p
    fi
    shift 1
    for NGX_VHOSTS in $*
    do
        CHECK_NGX_NUM='ls domains/|grep -aiE -c $NGX_VHOSTS'
        if [ $CHECK_NGX_NUM -eq 0 ];then
        cat>domains/$NGX_VHOSTS<<-EOF
        server {
               listen      80;
               server_name  $NGX_VHOSTS;
               location / {
                   root   html/$NGX_VHOSTS;
                   index  index.html index.htm;
               }
```

```
        }
        EOF
        mkdir -p ${NGX_DIR}/html/$NGX_VHOSTS
        cat>${NGX_DIR}/html/$NGX_VHOSTS/index.html<<-EOF
        <h1>$* Welcome to nginx!</h1>
        <hr color=red>
        EOF
        echo -e "\033[32m-----------------------\033[0m"
        echo -e "\033[32mThe Nginx $NGX_VHOSTS ADD Success.\033[0m"
        cat domains/$NGX_VHOSTS
        echo -e "\033[32m-----------------------\033[0m"
        $NGX_DIR/sbin/nginx -t
        $NGX_DIR/sbin/nginx -s reload
        echo
        else
            echo -e "\033[32m-----------------------\033[0m"
            echo -e "\033[32mThe Nginx $NGX_VHOSTS Already Exist,Please
Exit.\033[0m"
            cat domains/$NGX_VHOSTS
        fi
    done
}

function virtual_del(){
    if [ $# -le 1 ];then
            nginx_help
    fi
    shift 1
    for NGX_VHOSTS in $*
    do
        cd ${NGX_DIR}/conf/domains/ >/dev/null 2>&1
        if [ $? -eq 0 ];then
            ls -l|grep -aiE "$NGX_VHOSTS" >/dev/null 2>&1
            if [ $? -eq 0 ];then
                cat $NGX_VHOSTS
                if [ $? -eq 0 ];then
                    mkdir -p /data/backup/'date +%F'
                    \cp -a $NGX_VHOSTS /data/backup/'date +%F'
                    rm -rf $NGX_VHOSTS
                    $NGX_DIR/sbin/nginx -s reload
                    echo -e "\033[32m-----------------------\033[0m"
```

```
                echo -e "\033[32mThe Nginx $NGX_VHOSTS Already remove,
reload nginx...\033[0m"
            fi
        else
            shift 1
                  echo -e "\033[31m-----------------------\033[0m"
                  echo -e "\033[31mNginx $NGX_VHOSTS Virtual hosts does
not exist.please check.\033[0m"
                  ls -l $NGX_DIR/conf/ |head -10
        fi
    else
        shift 1
        echo -e "\033[31m-----------------------\033[0m"
        echo -e "\033[31mNginx $NGX_VHOSTS Virtual hosts does not exist.
please check.\033[0m"
        ls -l $NGX_DIR/conf/ |head -10
    fi
done
}

function tomcat_install(){
    #auto config tomcat web
    #change tomcat port :6001 7001 8001
    #upload jdk and tomcat for shell dir
    TOMCAT_VER="8.0.50"
    JAVA_VER="1.8.0_131"
    JAVA_DIR="/usr/java"
    TOMCAT_DIR="/usr/local"
    JAVA_SOFT="jdk${JAVA_VER}.tar.gz"
    TOMCAT_SOFT="apache-tomcat-${TOMCAT_VER}.tar.gz"
    if [ $# -le 1 ];then
            nginx_help
    fi
    shift 1
    #Install JAVA JDK
    grep -ai "^export" /etc/profile|grep -ai "JAVA_HOME" >/dev/null
    if [ $? -ne 0 ];then
        ls -l $JAVA_SOFT
        tar -xzvf $JAVA_SOFT
        mkdir -p $JAVA_DIR/
        \mv jdk$JAVA_VER $JAVA_DIR/
```

```
        ls -l $JAVA_DIR/jdk$JAVA_VER/
        $JAVA_DIR/jdk$JAVA_VER/bin/java -version
        cat>>/etc/profile<<-EOF
        export JAVA_HOME=$JAVA_DIR/jdk$JAVA_VER
        export CLASSPATH=\$CLASSPATH:\$JAVA_HOME/lib:\$JAVA_HOME/jre/lib
        export PATH=\$PATH:\$JAVA_HOME/bin/
        EOF
        java -version
        source /etc/profile
    fi
    source /etc/profile
    #Install Tomcat WEB
    MAX_PORT=$(for i in $(find /usr/local/ -name "server.xml");do grep -ai
"port=" $i;done|awk -F"=" '{print $2}'|awk '{print $1}'|sed 's/\"//g'|grep
-aivE "8443"|sort -nr|head -1)
    if [ -z $MAX_PORT ];then
        for TOMCAT_DOMAINS in $*
        do
            MAX_PORT=$(for i in $(find /usr/local/ -name "server.xml");do grep
-ai "port=" $i;done|awk -F"=" '{print $2}'|awk '{print $1}'|sed 's/\"//g'
|grep -aivE "8443"|sort -nr|head -1)
            if [ -z $MAX_PORT ];then
                #Install Tomcat WEB
                ls -l $TOMCAT_SOFT
                tar -xzvf $TOMCAT_SOFT >/dev/null
                mkdir -p $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
                \mv apache-tomcat-$TOMCAT_VER/* $TOMCAT_DIR/tomcat_$TOMCAT_
DOMAINS/ >/dev/null 2>&1
                $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/bin/startup.sh
                echo -e "\033[32m-----------------------\033[0m"
                       echo -e "\033[32mThe Nginx $TOMCAT_DOMAINS ADD Success.\
033[0m"
                sleep 5
                ps -ef|grep "$TOMCAT_DOMAINS"|grep -v "$0"
                netstat -tnlp|grep -aiwE "6001|7001|8001"
                setenforce 0
                systemctl stop firewalld.service
                service iptables stop
            else
                ls -l $TOMCAT_DIR/ |grep "$TOMCAT_DOMAINS" >>/dev/null 2>&1
                if [ $? -ne 0 ];then
```

```
                #Install Tomcat WEB
                        PORT1=$(expr $MAX_PORT - 2000 + 1)
                        PORT2=$(expr $MAX_PORT - 1000 + 1)
                        PORT3=$(expr $MAX_PORT + 1)
                        ls -l $TOMCAT_SOFT
                        tar -xzvf $TOMCAT_SOFT >/dev/null
                        mkdir -p $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
                        \mv apache-tomcat-$TOMCAT_VER/* $TOMCAT_DIR/
tomcat_$TOMCAT_DOMAINS/ >/dev/null 2>&1
                        sed -i "s/6001/$PORT1/g" $TOMCAT_DIR/tomcat_
$TOMCAT_DOMAINS/conf/server.xml
                        sed -i "s/7001/$PORT2/g" $TOMCAT_DIR/tomcat_
$TOMCAT_DOMAINS/conf/server.xml
                        sed -i "s/8001/$PORT3/g" $TOMCAT_DIR/tomcat_
$TOMCAT_DOMAINS/conf/server.xml
                        $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/bin/startup.sh
                echo -e "\033[32m-----------------------\033[0m"
                        echo -e "\033[32mThe Nginx $TOMCAT_DOMAINS ADD
Success.\033[0m"
                        sleep 5
                ps -ef|grep "$TOMCAT_DOMAINS"|grep -v "$0"
                        netstat -tnlp|grep -aiwE "$PORT1|$PORT2|$PORT3"
                        setenforce 0
                        systemctl stop firewalld.service
                        service iptables stop
            else
                echo -e "\033[32m-----------------------\033[0m"
                            echo -e "\033[32mThe Tomcat $TOMCAT_DOMAINS
Already Exist,Please Exit.\033[0m"
                echo "ls -l $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/"
                            ls -l $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
            fi
        fi
    done
else
    for TOMCAT_DOMAINS in $*
    do
        ls -l $TOMCAT_DIR/ |grep "$TOMCAT_DOMAINS" >>/dev/null 2>&1
        if [ $? -ne 0 ];then
            MAX_PORT=$(for i in $(find /usr/local/ -name "server.xml");do
grep -ai "port=" $i;done|awk -F"=" '{print $2}'|awk '{print $1}'|sed
```

```
's/\"//g'|grep -aivE "8443"|sort -nr|head -1)
              #Install Tomcat WEB
              PORT1=$(expr $MAX_PORT - 2000 + 1)
              PORT2=$(expr $MAX_PORT - 1000 + 1)
              PORT3=$(expr $MAX_PORT + 1)
              ls -l $TOMCAT_SOFT
              tar -xzvf $TOMCAT_SOFT >/dev/null
              mkdir -p $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
              \mv apache-tomcat-$TOMCAT_VER/* $TOMCAT_DIR/tomcat_$TOMCAT_
DOMAINS/ >/dev/null 2>&1
              sed -i "s/6001/$PORT1/g" $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
conf/server.xml
              sed -i "s/7001/$PORT2/g" $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
conf/server.xml
              sed -i "s/8001/$PORT3/g" $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
conf/server.xml
              $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/bin/startup.sh
              echo -e "\033[32m----------------------\033[0m"
                    echo -e "\033[32mThe Tomcat $TOMCAT_DOMAINS ADD
Success.\033[0m"
              sleep 5
              ps -ef|grep "$TOMCAT_DOMAINS"|grep -v "$0"
              netstat -tnlp|grep -aiwE "$PORT1|$PORT2|$PORT3"
              setenforce 0
              systemctl stop firewalld.service
              service iptables stop
          else
                          echo -e "\033[32m----------------------\033[0m"
                          echo -e "\033[32mThe Tomcat $TOMCAT_DOMAINS
Already Exist,Please Exit.\033[0m"
              echo "ls -l $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/"
                          ls -l $TOMCAT_DIR/tomcat_$TOMCAT_DOMAINS/
          fi
       done
    fi
}
case $1 in
    -i|-I)
    nginx_install $*
    ;;
    -u|-U)
```

```
    nginx_update $*
    ;;
    -a|-A)
    virtual_add $*
    ;;
    -d|-D)
    virtual_del $*
    ;;
    -t|-T)
    tomcat_install $*
    ;;
    * )
    nginx_help
    ;;
esac
```

2.23 Shell 编程密码远程执行命令脚本

Shell 编程密码远程执行命令脚本，编写思路如下：

（1）自动将服务器 IP 和用户名、密码表保存至文件 list.txt。

（2）自动安装 expect 远程交互工具。

（3）自动编写 expect 远程执行命令文件。

（4）支持任意命令远程执行。

（5）支持列表循环操作多台服务器。

相关代码如下：

```
#!/bin/sh
#auto exec expect shell scripts
#by author www.jfedu.net 2021
if
     [ ! -e /usr/bin/expect ];then
     yum  install expect -y
fi
#Judge passwd.txt exist
if
     [ ! -e ./passwd.txt ];then
     echo -e "The passwd.txt is not exist......Please touch ./passwd.txt ,
Content Example:\n192.168.1.11 passwd1\n192.168.1.12 passwd2"
```

```
     sleep 2 &&exit 0
fi
#Auto Touch login.exp File
cat>login.exp <<EOF
#!/usr/bin/expect -f
set ip [lindex \$argv 0 ]
set passwd [lindex \$argv 1 ]
set command [lindex \$argv 2]
set timeout -1
spawn ssh root@\$ip
expect {
"yes/no" { send "yes\r";exp_continue }
"password:" { send "\$passwd\r" }
}
expect "*#*" { send "\$command\r" }
expect  "#*"  { send "exit\r" }
expect eof
EOF
##Auto exec shell scripts
CMD="$*"
if
   [ "$1" == "" ];then
   echo ============================================================
   echo "Please insert your command ,Example {/bin/sh  $0 'mkdir -p
/tmp'} ,waiting exit ..... "
   sleep 2
   exit 1
fi
for i in 'awk '{print $1}' passwd.txt'
do
   j='awk -v I="$i" '{if(I==$1)print $2}' passwd.txt'
   expect ./login.exp  $i  $j  "$CMD"
done
```

2.24　Shell 编程密码远程复制文件脚本

Shell 编程密码远程复制文件脚本，编写思路如下：

（1）自动将服务器 IP 和用户名、密码表保存至文件 list.txt。

（2）自动安装 expect 远程交互工具。

（3）自动编写 expect 远程执行复制文件。

（4）支持任意文件远程复制操作。

（5）支持列表循环操作多台服务器。

相关代码如下：

```
#!/bin/sh
#auto exec expect shell scripts
#by author www.jfedu.net 2021
if
     [ ! -e /usr/bin/expect ];then
     yum  install expect -y
fi
#Judge passwd.txt exist
if
     [ ! -e ./passwd.txt ];then
     echo -e "The passwd.txt is not exist......Please touch ./passwd.txt ,
Content Example:\n192.168.1.11 passwd1\n192.168.1.12 passwd2"
     sleep 2 &&exit 0
fi
#Auto Touch login.exp File
cat>login.exp <<EOF
#!/usr/bin/expect -f
set ip [lindex \$argv 0]
set passwd [lindex \$argv 1]
set src_file [lindex \$argv 2]
set des_dir [lindex \$argv 3]
set timeout -1
spawn scp -r \$src_file root@\$ip:\$des_dir
expect {
"yes/no"    { send "yes\r"; exp_continue }
"password:"  { send "\$passwd\r" }
}
expect "100%"
expect eof
EOF
##Auto exec shell scripts
if
   [ "$1" == "" ];then
   echo ============================================================
   echo "Please insert your are command ,Example {/bin/sh  $0 /src
/des } ,waiting exit ..... "
   sleep 2
```

```
    exit 1
fi
for i in 'awk '{print $1}'  passwd.txt'
do
     j='awk -v I="$i" '{if(I==$1)print $2}' passwd.txt'
     expect ./login.exp  $i  $j $1 $2
done
```

2.25　Shell 编程 Bind DNS 管理脚本

Bind 主要应用于企业 DNS 构建平台，而 DNS 用于解析域名与 IP 地址，用户在浏览器只需输入域名，即可访问服务器对应 IP 地址的虚拟主机网站。

Bind 难点在于创建各种记录，例如 A 记录、mail 记录、反向记录、资源记录等，Shell 脚本可以减轻人工的操作，节省大量的时间成本。

Shell 脚本实现 Bind 自动安装、初始化 Bind 环境、自动添加 A 记录、反向记录、批量添加 A 记录，编写思路如下：

（1）YUM 方式自动安装 Bind。

（2）自动初始化 Bind 配置。

（3）创建安装，初始化，添加记录函数。

（4）自动添加单个 A 记录及批量添加 A 记录和反向记录。

相关代码如下：

```
#!/bin/bash
#Auto install config bind server
#By author jfedu.net 2021
#Define Path variables
BND_ETC=/var/named/chroot/etc
BND_VAR=/var/named/chroot/var/named
BAK_DIR=/data/backup/dns_'date +%Y%m%d-%H%M'
##Backup named server
if
     [ ! -d  $BAK_DIR ];then
     echo "Please waiting  Backup Named Config ....."
     mkdir   -p  $BAK_DIR
     cp -a  /var/named/chroot/{etc,var}   $BAK_DIR
     cp -a  /etc/named.* $BAK_DIR
```

```
fi
##Define Shell Install Function
Install ()
{
  if
     [ ! -e /etc/init.d/named ];then
     yum install bind* -y
else
     echo ------------------------------------------------
     echo "The Named Server is exists ,Please exit ........."
     sleep 1
 fi
}
##Define Shell Init Function
Init_Config ()
{
      sed  -i -e 's/localhost;/any;/g' -e '/port/s/127.0.0.1/any/g'
/etc/named.conf
      echo ------------------------------------------------
      sleep 2
      echo "The named.conf config Init success !"
}
##Define Shell Add Name Function
Add_named ()
{
##DNS name
      read -p  "Please  Insert Into Your Add Name ,Example 51cto.com :" NAME
      echo $NAME |grep -E "com|cn|net|org"
      while
       [ "$?" -ne 0 ]
        do
       read -p  "Please  reInsert Into Your Add Name ,Example 51cto.com :"
NAME
       echo $NAME |grep -E "com|cn|net|org"
     done
## IP address
      read -p  "Please  Insert Into Your Name Server IP ADDress:" IP
      echo $IP |egrep -o "([0-9]{1,3}\.){3}[0-9]{1,3}"
      while
      [ "$?" -ne "0" ]
       do
```

```
        read -p  "Please  reInsert Into Your Name Server IP ADDress:" IP
       echo $IP |egrep -o "([0-9]{1,3}\.){3}[0-9]{1,3}"
      done
       ARPA_IP='echo $IP|awk -F. '{print $3"."$2"."$1}''
       ARPA_IP1='echo $IP|awk -F. '{print $4}''
       cd  $BND_ETC
       grep  "$NAME" named.rfc1912.zones
if
        [ $? -eq 0 ];then
        echo "The $NAME IS exist named.rfc1912.zones conf ,please exit ..."
        exit
else
       read -p  "Please  Insert Into SLAVE Name Server IP ADDress:" SLAVE
       echo $SLAVE |egrep -o "([0-9]{1,3}\.){3}[0-9]{1,3}"
       while
       [ "$?" -ne "0" ]
       do
          read -p  "Please  Insert Into SLAVE Name Server IP ADDress:" SLAVE
          echo $SLAVE |egrep -o "([0-9]{1,3}\.){3}[0-9]{1,3}"
       done
          grep  "rev" named.rfc1912.zones
      if
        [ $? -ne 0 ];then
          cat >>named.rfc1912.zones <<EOF
#'date +%Y-%m-%d' Add $NAME CONFIG
zone "$NAME" IN {
       type master;
       file "$NAME.zone";
       allow-update { none; };
};
zone "$ARPA_IP.in-addr.arpa" IN {
       type master;
       file "$ARPA_IP.rev";
       allow-update { none; };
};
EOF
     else
       cat >>named.rfc1912.zones <<EOF

#'date +%Y-%m-%d' Add $NAME CONFIG
zone "$NAME" IN {
```

```
        type master;
        file "$NAME.zone";
        allow-update { none; };
};
EOF
   fi
fi
      [ $? -eq 0 ]&& echo "The $NAME config name.rfc1912.zones success !"
      sleep 3 ;echo "Please waiting config $NAME zone File ............."
      cd  $BND_VAR
      read -p "Please insert Name DNS A HOST ,EXample  www or mail :" HOST
      read -p "Please insert Name DNS A NS IP ADDR ,EXample 192.168.111.130 :"
IP_HOST
      echo $IP_HOST |egrep -o "([0-9]{1,3}\.){3}[0-9]{1,3}"
      ARPA_IP2='echo $IP_HOST|awk -F. '{print $3"."$2"."$1}''
      ARPA_IP3='echo $IP_HOST|awk -F. '{print $4}''
      while
      [ "$?" -ne "0" ]
do
      read -p "Please Reinsert Name DNS A IPADDRESS ,EXample 192.168.111.
130 :" IP_HOST
      echo $IP_HOST |egrep -o "([0-9]{1,3}\.){3}[0-9]{1,3}"
done
      cat >$NAME.zone <<EOF
\$TTL   86400
@            IN SOA  localhost.     root.localhost. (
                                  43            ; serial (d. adams)
                                  1H            ; refresh
                                  15M           ; retry
                                  1W            ; expiry
                                  1D )          ; minimum

             IN  NS         $NAME.
EOF
      REV='ls  *.rev'
      ls  *.rev >>/dev/null
if
      [ $? -ne 0 ];then
      cat >>$ARPA_IP.rev <<EOF
\$TTL   86400
@      IN     SOA    localhost.    root.localhost.  (
```

```
                                1997022703 ; Serial
                                28800     ; Refresh
                                14400     ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum

          IN  NS  $NAME.
EOF
       echo  "$HOST           IN  A          $IP_HOST" >>$NAME.zone
       echo  "$ARPA_IP3        IN  PTR        $HOST.$NAME." >>$ARPA_IP.rev
       [ $? -eq 0 ]&& echo -e "The $NAME config success:\n$HOST      IN  A
$IP_HOST\n$ARPA_IP3        IN  PTR        $HOST.$NAME."
else
       sed -i  "9a IN  NS  $NAME." $REV
       echo  "$HOST           IN  A          $IP_HOST" >>$NAME.zone
       echo  "$ARPA_IP3        IN  PTR        $HOST.$NAME." >>$REV
       [ $? -eq 0 ]&& echo -e "The $NAME config success1:\n$HOST      IN  A
$IP_HOST\n$ARPA_IP3        IN  PTR        $HOST.$NAME."
fi
}
##Define Shell List A Function
Add_A_List ()
{
if
      cd  $BND_VAR
      REV='ls  *.rev'
      read -p  "Please  Insert Into Your Add Name ,Example 51cto.com :" NAME
      [ ! -e "$NAME.zone" ];then
      echo "The $NAME.zone File is not exist ,Please ADD $NAME.zone File :"
      Add_named ;
else
      read -p "Please Enter List Name A NS File ,Example /tmp/name_list.txt:
" FILE
   if
      [ -e $FILE ];then
      for i in  'cat $FILE|awk '{print $2}'|sed "s/$NAME//g"|sed 's/\.$//g''
      #for i in  'cat $FILE|awk '{print $1}'|sed "s/$NAME//g"|sed 's/\.$//g''
do
      j='awk -v I="$i.$NAME" '{if(I==$2)print $1}' $FILE'
      echo ------------------------------------------------------------
```

```
      echo "The $NAME.zone File is exist ,Please Enter insert NAME HOST ...."
      sleep 1
      ARPA_IP='echo $j|awk -F. '{print $3"."$2"."$1}''
      ARPA_IP2='echo $j|awk -F. '{print $4}''
      echo  "$i           IN  A         $j" >>$NAME.zone
      echo  "$ARPA_IP2     IN  PTR     $i.$NAME." >>$REV
      [ $? -eq 0 ]&& echo -e "The $NAME config success:\n$i     IN  A
$j\n$ARPA_IP2        IN  PTR        $i.$NAME."
done
    else
      echo "The $FILE List File IS Not Exist .......,Please exit ..."
    fi
fi
}
##Define Shell Select Menu
PS3="Please select Menu Name Config: "
select i in "自动安装 Bind 服务"  "自动初始化 Bind 配置" "添加解析域名"  "批量添加 A
记录"
do
case   $i   in
      "自动安装 Bind 服务")
      Install
;;
      "自动初始化 Bind 配置")
      Init_Config
;;
      "添加解析域名")
      Add_named
;;
      "批量添加 A 记录")
      Add_A_List
;;
      * )
      echo ----------------------------------------------------
      sleep 1
      echo "Please exec: sh  $0  { Install(1)  or Init_Config(2) or
Add_named(3) or Add_config_A(4) }"
;;
esac
done
```

2.26　Shell 编程 Docker 虚拟化管理脚本

Docker 虚拟化是目前主流的虚拟化解决方案，越来越多的企业在使用 Docker 轻量级虚拟化，构建、维护和管理 Docker 虚拟化平台是非常重要的环节，开发 Docker Shell 脚本可以在命令行界面快速管理和维护 Docker。

Shell 脚本实现 Docker 自动安装、自动导入镜像、创建虚拟机、指定 IP 地址、将创建的 Docker 虚拟机加入 Excel 存档或加入 MySQL 数据库，编写思路如下：

（1）基于 CentOS 6.5+或 7.x YUM 安装 Docker。

（2）Docker 脚本参数指定 CPU、内存和硬盘容量。

（3）Docker 自动检测局域网 IP 并赋予 Docker 虚拟机。

（4）Docker 基于 pipework 指定 IP。

（5）将创建的 Docker 虚拟机加入 CSV（Excel）或 MySQL 数据库。

相关代码如下：

```
#!/bin/bash
#Auto install docker and Create VM
#By author jfedu.net 2021
#Define Path variables
IPADDR='ifconfig|grep -E "\<inet\>"|awk '{print $2}'|grep "192.168"|head -1'
GATEWAY='route -n|grep "UG"|awk '{print $2}'|grep "192.168"|head -1'
IPADDR_NET='ifconfig|grep -E "\<inet\>"|awk '{print $2}'|grep "192.168"
|head -1|awk -F. '{print $1"."$2"."$3"."}''
LIST="/root/docker_vmlist.csv"
if [ ! -f /usr/sbin/ifconfig ];then
    yum install net-tools* -y
fi
for i in 'seq 1 253';do ping -c 1 ${IPADDR_NET}${i} ;[ $? -ne 0 ]&&
DOCKER_IPADDR="${IPADDR_NET}${i}" &&break;done >>/dev/null 2>&1
echo "###################"
echo -e "Dynamic get docker IP,The Docker IP address\n\n$DOCKER_IPADDR"
NETWORK=(
    HWADDR='ifconfig eth0|grep ether|awk '{print $2}''
    IPADDR='ifconfig eth0|grep -E "\<inet\>"|awk '{print $2}''
    NETMASK='ifconfig eth0|grep -E "\<inet\>"|awk '{print $4}''
    GATEWAY='route -n|grep "UG"|awk '{print $2}''
)
```

```
if [ -z "$1" -o -z "$2" ];then
    echo -e "\033[32m---------------------------------\033[0m"
    echo -e "\033[32mPlease exec $0 CPU(C) MEM(G),example $0 4 8\033[0m"
    exit 0
fi
#CPU='expr $2 - 1'
if [ ! -e /usr/bin/bc ];then
    yum install bc -y >>/dev/null 2>&1
fi
CPU_ALL='cat /proc/cpuinfo |grep processor|wc -l'
if [ ! -f $LIST ];then
    CPU_COUNT=$1
    CPU_1="0"
    CPU1='expr $CPU_1 + 0'
    CPU2='expr $CPU1 + $CPU_COUNT - 1'
    if [ $CPU2 -gt $CPU_ALL ];then
        echo -e "\033[32mThe System CPU count is $CPU_ALL,not more than
it.\033[0m"
        exit
    fi
else
    CPU_COUNT=$1
    CPU_1='cat $LIST|tail -1|awk -F"," '{print $4}'|awk -F"-" '{print $2}''
    CPU1='expr $CPU_1 + 1'
    CPU2='expr $CPU1 + $CPU_COUNT - 1'
    if [ $CPU2 -gt $CPU_ALL ];then
        echo -e "\033[32mThe System CPU count is $CPU_ALL,not more than
it.\033[0m"
        exit
    fi
fi
MEM_F='echo $2 \* 1024|bc'
MEM='printf "%.0f\n" $MEM_F'
DISK=20
USER=$3
REMARK=$4
ping $DOCKER_IPADDR -c 1 >>/dev/null 2>&1
if [ $? -eq 0 ];then
    echo -e "\033[32m---------------------------------\033[0m"
    echo -e "\033[32mThe IP address to be used,Please change other
IP,exit.\033[0m"
```

```
    exit 0
fi
if [ ! -e /usr/bin/docker ];then
    yum install docker* device-mapper*  -y
    mkdir -p /export/docker/
    cd /var/lib/ ;rm -rf docker ;ln -s /export/docker/ .
    mkdir -p /var/lib/docker/devicemapper/devicemapper
    dd if=/dev/zero of=/var/lib/docker/devicemapper/devicemapper/data bs=1G
count=0 seek=2000
    service docker start
    if [ $? -ne 0 ];then
       echo "Docker install error ,please check."
       exit
    fi
fi
cd  /etc/sysconfig/network-scripts/
    mkdir -p /data/backup/'date +%Y%m%d-%H%M'
    yes|cp ifcfg-eth* /data/backup/'date +%Y%m%d-%H%M'/
if
    [ -e /etc/sysconfig/network-scripts/ifcfg-br0 ];then
    echo
else
    cat >ifcfg-eth0 <<EOF
    DEVICE=eth0
    BOOTPROTO=none
    ${NETWORK[0]}
    NM_CONTROLLED=no
    ONBOOT=yes
    TYPE=Ethernet
    BRIDGE="br0"
    ${NETWORK[1]}
    ${NETWORK[2]}
    ${NETWORK[3]}
    USERCTL=no
EOF
    cat >ifcfg-br0 <<EOF
    DEVICE="br0"
    BOOTPROTO=none
    ${NETWORK[0]}
    IPV6INIT=no
    NM_CONTROLLED=no
```

```
    ONBOOT=yes
    TYPE="Bridge"
    ${NETWORK[1]}
    ${NETWORK[2]}
    ${NETWORK[3]}
    USERCTL=no
EOF
    /etc/init.d/network restart
fi
echo 'Your can restart Ethernet Service: /etc/init.d/network restart !'
echo '------------------------------------------------------------'

cd -
#######create docker container
service docker status >>/dev/null
if [ $? -ne 0 ];then
    service docker restart
fi
NAME="Docker_'echo $DOCKER_IPADDR|awk -F"." '{print $(NF-1)"_"$NF}'"
IMAGES='docker images|grep -v "REPOSITORY"|grep -v "none"|grep "jfedu"|head
-1|awk '{print $1}''
if [ -z $IMAGES ];then
    echo "Plesae Download Docker Centos Images,you can to be use docker search
centos,and docker pull centos6.5-ssh,exit 0"
    if [ ! -f jfedu_centos68.tar ];then
        echo "Please upload jfedu_centos68.tar for docker server."
        exit
    fi
    cat jfedu_centos68.tar|docker import - jfedu_centos6.8
fi
IMAGES='docker images|grep -v "REPOSITORY"|grep -v "none"|grep "jfedu"|head
-1|awk '{print $1}''
CID=$(docker run -itd --privileged --cpuset-cpus=${CPU1}-${CPU2} -m ${MEM}m
--net=none --name=$NAME $IMAGES /bin/bash)
echo $CID
docker ps -a |grep "$NAME"
pipework br0 $NAME  $DOCKER_IPADDR/24@$IPADDR
docker exec $NAME /etc/init.d/sshd start
if [ ! -e $LIST ];then
    echo "编号,容器 ID,容器名称,CPU,内存,硬盘,容器 IP,宿主机 IP,使用人,备注" >$LIST
fi
```

```
###################
NUM='cat $LIST |grep -v CPU|tail -1|awk -F, '{print $1}''
if [[ $NUM -eq "" ]];then
       NUM="1"
else
       NUM='expr $NUM + 1'
fi
##################
echo -e "\033[32mCreate virtual client Successfully.\n$NUM 'echo $CID|cut
-b 1-12',$NAME,$CPU1-$CPU2,${MEM}M,${DISK}G,$DOCKER_IPADDR,$IPADDR,$USER,
$REMARK\033[0m"
if [ -z $USER ];then
   USER="NULL"
   REMARK="NULL"
fi
echo $NUM, 'echo $CID|cut -b 1-12',$NAME,$CPU1-$CPU2,${MEM}M,${DISK}G,
$DOCKER_IPADDR,$IPADDR,$USER,$REMARK >>$LIST
rm -rf /root/docker_vmlist_*
iconv -c -f utf-8 -t gb2312 $LIST  -o /root/docker_vmlist_'date +%H%M'.csv
```

2.27　Shell 编程脚本

2.27.1　Shell 编程采集服务器硬件信息脚本

Shell 编程采集服务器硬件信息脚本，编写思路如下：

（1）创建数据库和表存储服务器信息。

（2）基于获取硬件服务器 CPU、内存、硬盘、网卡等相关信息。

（3）将获取的信息写成 SQL 语句，并插入数据库。

（4）定期对 SQL 数据进行备份。

（5）将脚本加入 Crontab 实现自动备份。

2.27.2　Shell 编程 Linux 系统初始化脚本

Shell 编程 Linux 系统初始化脚本，编写思路如下：

（1）Linux 系统安装完成，自动初始化系统。

（2）关闭不必要的端口。

（3）关闭不必要的服务。

（4）添加同步时间任务计划。

（5）优化相关 Linux 内核参数。

2.27.3 Shell 编程 Xtrabackup 数据库自动备份脚本

Shell 编程 Xtrabackup 数据库自动备份脚本，编写思路如下：

（1）支持 MySQL 单个库备份。

（2）支持 MySQL 多个库备份。

（3）支持 MySQL 全数据库备份。

（4）支持 MySQL 指定数据库增量备份。

（5）支持 MySQL 指定多个数据库增量备份。

（6）支持 MySQL 数据库定期删除。

2.27.4 Shell 编程 Linux 服务器免密钥分发脚本

Shell 编程 Linux 服务器免密钥分发脚本，编写思路如下：

（1）基于 ssh-keygen 自动生成公钥和私钥。

（2）给定所有客户端的 IP、用户名和密码信息。

（3）基于命令工具将公钥自动复制到远程机器。

（4）给定的客户端 ip.txt 信息如下：

```
#ip    user    password
192.168.1.100  root  123456
192.168.1.101  root  1qaz@WSX
192.168.1.102  root  123
192.168.1.103  root  456
```

2.27.5 Shell 编程 Nginx 多版本软件安装脚本

Shell 编程 Nginx 多版本软件安装脚本，编写思路如下：

（1）安装不同的 Nginx 版本。

（2）检测系统是否已经存在，是否可以覆盖版本。

（3）启动 Nginx 并测试访问。

（4）支持多个版本指定目录安装和启动。

2.27.6　Shell 编程自动收集软件、端口、进程脚本

Shell 编程自动收集软件、端口、进程脚本，编写思路如下：

（1）收集服务器所有端口和对应的服务。

（2）收集服务器对应服务 Nginx 发布目录、虚拟主机。

（3）收集服务器对应服务 MySQL 数据目录、配置文件。

2.27.7　Shell 编程 LVS 负载均衡管理脚本

Shell 编程 LVS 负载均衡管理脚本，编写思路如下：

（1）Shell 脚本自动安装 LVS 负载均衡。

（2）支持添加 VIP 地址。

（3）支持在 VIP 上添加后端 Realserver IP。

（4）支持删除后端真实 IP 和 VIP。

2.27.8　Shell 编程 Keepalived 管理脚本

Shell 编程 Keepalived 管理脚本，编写思路如下：

（1）Shell 脚本支持自动配置 Keepalived 服务。

（2）自动添加负载均衡 VIP。

（3）能够在已有的 VIP 均衡中添加 Realserver IP。

（4）支持删除某个 VIP 中指定的 Realserver IP。

2.27.9　Shell 编程 Discuz 门户网站自动部署脚本

Shell 编程 Discuz 门户网站自动部署脚本，编写思路如下：

（1）Shell 脚本支持从 SVN 获取网站代码。

（2）实现创建备份目录&备份源网站。

（3）将新的文件更新至 Discuz 对应的目录。

2.27.10 Shell 编程监控 Linux 磁盘分区容量脚本

Shell 编程监控 Linux 磁盘分区容量脚本，编写思路如下：

（1）Shell 脚本实现 Linux 多个分区监控。

（2）打印磁盘使用率超过 85%的分区。

（3）将分区使用率超过 85%的磁盘发送 SA 邮件报警。

（4）将分区使用率超过 85%的磁盘发送 SA 微信报警。