Linux 自动化运维实战

1. Shell企业编程基础

1.4

if条件判断语句通常以if开头，以fi结尾，也可加入else或elif进行多条件的判断。if表达式如下：

if (表达式)

语句1

else

语句2

fi

if语句Shell脚本编程案例如下。

（1）比较两个整数大小。

#!/bin/bash

#By author jfedu.net 2021

NUM=100

if (( $NUM > 4 )) ;then

echo "The Num $NUM more than 4."

else

echo "The Num $NUM less than 4."

fi

（2）判断系统目录是否存在。

#!/bin/bash

#judge DIR or Files

#By author jfedu.net 2021

if [ ! -d /data/20210515 -a ! -d /tmp/2021/ ];then

mkdir -p /data/20210515

fi

if常见判断逻辑运算符详解如下：

-f #判断文件是否存在 eg: if [ -f filename ]

-d #判断目录是否存在 eg: if [ -d dir ]

-eq #等于,应用于整型比较 equal

-ne #不等于,应用于整型比较 not equal

-lt #小于,应用于整型比较 letter

-gt #大于,应用于整型比较 greater

-le #小于或等于,应用于整型比较

-ge #大于或等于,应用于整型比较

-a #双方都成立(and) 逻辑表达式 –a 逻辑表达式

-o #单方成立(or) 逻辑表达式 –o 逻辑表达式

-z #空字符串

|| #单方成立

&& #双方都成立表达式

（3）if多个条件测试分数判断。

#!/bin/bash

#By author jfedu.net 2021

scores=$1

if [[ $scores -eq 100 ]]; then

    echo "very good!";

elif [[ $scores -gt 85 ]]; then

    echo "good!";

elif [[ $scores -gt 60 ]]; then

    echo "pass!";

elif [[ $scores -lt 60 ]]; then

    echo "no pass!"

fi

1.5.1

Shell编程中，尤其是使用if语句时，经常会使用( )、(( ))、[ ]、[[ ]]、{ }等括号，以下为几种括号的简单对比。

( )

#用于多个命令组、命令替换及初始化数组,多用于Shell命令组,例如JF=(jf1 jf2 jf3),其

#中括号左右不保留空格

(( ))

#整数扩展、运算符、重定义变量值,算术运算比较,例如((i++))、((i<=100)),其中括号左右

#不保留空格

[ ]

#Bash内部命令,[ ]与test是等同的,正则字符范围、引用数组元素编号,不支持+-\*/数学运

#算符,逻辑测试使用-a、-o,通常用于字符串比较、整数比较以及数组索引,其中括号左右要保留

#空格

[[ ]]

#Bash程序语言的关键字,不是一个命令,[[ ]]结构比[ ]结构更加通用,不支持+-\*/数学运算

#符,逻辑测试使用&&、||,通常用于字符串比较、逻辑运算符等,其中括号左右要保留空格

{}

#主要用于命令集合或者范围,例如mkdir -p /data/201{7,8}/,其中括号左右不保留空格

1.6

MySQL数据库备份是运维工程师的工作之一，以下为自动备份MySQL数据库脚本。

#!/bin/bash

#auto backup mysql

#By author jfedu.net 2021

#Define PATH定义变量

BAK\_DIR=/data/backup/mysql/'date +%Y-%m-%d'

MYSQLDB=webapp

MYSQLPW=backup

MYSQLUSR=backup

#must use root user run scripts 必须使用root用户运行,$UID为系统变量

if

[ $UID -ne 0 ];then

echo This script must use the root user!!!

sleep 2

exit 0

fi

#Define DIR and mkdir DIR 判断目录是否存在,不存在则新建

if

[ ！ -d $BAKDIR ];then

mkdir -p $BAKDIR

fi

#Use mysqldump backup Databases

/usr/bin/mysqldump -u$MYSQLUSR -p$MYSQLPW -d $MYSQLDB >$BAKDIR/

webapp\_db.sql

echo "The mysql backup successfully"

1.7

（2）编写LNMP一键部署脚本，前提要熟练手动方式部署LNMP架构。auto\_install\_lnmp\_ v1.sh脚本代码如下：

#!/bin/bash

#2021年5月30日19:28:15

#auto install LNMP Web

#by author www.jfedu.net

######################

#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 --prefix=/usr/local/nginx --user=www --group=www --with-http\_

stub\_status\_module

make && make install

/usr/local/nginx/sbin/nginx

setenforce 0

systemctl stop firewalld.service

#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-5.6.51.tar.gz

tar -xzf mysql-5.6.51.tar.gz

cd mysql-5.6.51

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

#Install PHP Web 2021

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

#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

cat>/usr/local/nginx/conf/nginx.conf<<EOF

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 / {

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name

include fastcgi\_params;

}

location ~ .\*\.(php|jsp|cgi)$

{

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

location ~ .\*\.(htm|html|png|jpeg|gif|txt|js|css|doc)$

{

root html;

expires 30d;

}

}

}

EOF

echo "

<?php

phpinfo();

?>">/usr/local/nginx/html/index.php

/usr/local/nginx/sbin/nginx -s reload

1.8

for循环语句主要用于对某个数据域进行循环读取、对文件进行遍历，通常用于循环某个文件或者列表。其语法格式以for…do开头，以done结尾。语法格式如下：

for var in （表达式）

do

语句1

done

for循环语句Shell脚本编程案例如下。

（1）循环打印百度、淘宝和腾讯企业官网。

#!/bin/bash

#By author jfedu.net 2021

for website in www.baidu.com www.taobao.com www.qq.com

do

echo $website

done

（2）循环打印1～100的数字。

#!/bin/bash

#By author jfedu.net 2021

for i in 'seq 1 100' #seq表示列出数据范围

do

echo "NUM is $i"

done

（3）求1～100的总和。

#!/bin/bash

#By author jfedu.net 2021

#auto sum 1 100

j=0

for ((i=1;i<=100;i++))

do

j='expr $i + $j'

done

echo $j

（4）对系统日志文件进行分组打包。

#!/bin/bash

#By author jfedu.net 2021

for i in 'find /var/log -name "\*.log"'

do

tar -czf 2021\_log$i.tgz $i

done

（5）批量远程主机文件传输。

#!/bin/bash

#auto scp files for client

#By author jfedu.net 2021

for i in 'seq 100 200'

do

scp -r /tmp/jfedu.txt root@192.168.1.$i:/data/webapps/www

done

（6）批量远程主机执行命令。

#!/bin/bash

#auto scp files for client

#By author jfedu.net 2021

for i in 'seq 100 200'

do

ssh -l root 192.168.1.$i 'ls /tmp'

done

（7）打印10s等待提示。

for ((j=0;j<=10;j++))

do

echo -ne "\033[32m-\033[0m"

sleep 1

done

echo

1.9

while循环语句与for循环功能类似，主要用于对某个数据域进行循环读取、对文件进行遍历，通常用于循环某个文件或者列表，满足循环条件会一直循环，不满足则退出循环。其语法格式以while…do开头，以done结尾。语法格式如下：

while (表达式)

do

语句1

done

while循环语句Shell脚本编程案例如下。

（1）循环打印百度、淘宝、腾讯企业官网。

#!/bin/bash

#By author jfedu.net 2021

while read line #read指令用于读取行或读取变量

do

echo $line

done <jfedu.txt

其中jfedu.txt内容为：

www.baidu.com

www.taobao.com

www.qq.com

（2）循环每秒输出Hello World。

#!/bin/bash

#By author jfedu.net 2021

while sleep 1

do

echo -e "\033[32mHello World.\033[0m"

done

其中jfedu.txt内容为：

www.baidu.com

www.taobao.com

www.qq.com

（3）打印1～100的数字。

#!/bin/bash

#By author jfedu.net 2021

i=0

while ((i<=100)) #此处只打印1～100,并没有一直循环,当i≤100时结束

do

echo $i

i='expr $i + 1' #expr用于运算逻辑工具

done

（4）求1～100的总和。

#!/bin/bash

#By author jfedu.net 2021

#auto sum 1 100

j=0

i=1

while ((i<=100))

do

j='expr $i + $j'

((i++))

done

echo $j

（5）逐行读取文件。

#!/bin/bash

#By author jfedu.net 2021

while read line

do

echo $line;

done < /etc/hosts

（6）判断输入IP是否正确。

#!/bin/bash

#By author jfedu.net 2021

#Check IP Address

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}"

while [ $? -ne 0 ]

do

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}"

done

（7）每5s循环一次判断/etc/passwd是否被非法修改。

#!/bin/bash

#Check File to change

#By author jfedu.net 2021

FILES="/etc/passwd"

while true

do

echo "The Time is 'date +%F-%T'"

OLD='md5sum $FILES|cut -d" " -f 1'

sleep 5

NEW='md5sum $FILES|cut -d" " -f 1'

if [[ $OLD != $NEW ]];then

echo "The $FILES has been modified."

fi

done

（8）每10s循环一次判断jfedu用户是否登录系统。

#!/bin/bash

#Check File to change

#By author jfedu.net 2021

USERS="jfedu"

while true

do

echo "The Time is 'date +%F-%T'"

sleep 10

NUM='who|grep "$USERS"|wc -l'

if [[ $NUM -ge 1 ]];then

echo "The $USERS is login in system."

fi

done

1.10

case选择语句主要用于对多个选择条件进行匹配输出，与if…elif语句结构类似，通常用于脚本传递输入参数，打印出输出结果及内容，其语法格式以case…in开头，以esac结尾。语法格式如下：

#!/bin/bash

#By author jfedu.net 2021

case $1 in

Pattern1)

语句1

;;

Pattern2)

语句2

;;

Pattern3)

语句3

;;

esac

case条件语句Shell脚本编程案例如下。

（1）打印monitor及archive选择菜单。

#!/bin/bash

#By author jfedu.net 2021

case $1 in

monitor)

monitor\_log

;;

archive)

archive\_log

;;

help)

echo –e "\033[32mUsage:{$0 monitor | archive |help }\033[0m"

;;

\*)

echo –e "\033[32mUsage:{$0 monitor | archive |help }\033[0m "

esac

（2）自动修改IP脚本菜单。

#!/bin/bash

#By author jfedu.net 2021

case $i in

modify\_ip)

change\_ip

;;

modify\_hosts)

change\_hosts

;;

exit)

exit

;;

\*)

echo -e "1) modify\_ip\n2) modify\_ip\n3)exit"

esac

1.11

select语句一般用于选择，常用于选择菜单的创建，可以配合PS3做打印菜单的输出信息，其语法格式以select…in do开头，以done结尾。

select i in （表达式）

do

语句

done

select选择语句Shell脚本编程案例如下。

（1）打印开源操作系统选择。

#!/bin/bash

#By author jfedu.net 2021

PS3="What you like most of the open source system?"

select i in CentOS RedHat Ubuntu

do

echo "Your Select System: "$i

done

（2）打印LAMP选择菜单。

#!/bin/bash

#By author jfedu.net 2021

PS3="Please enter you select install menu:"

select i in http php mysql quit

do

case $i in

http)

echo Test Httpd.

;;

php)

echo Test PHP.

;;

mysql)

echo Test MySQL.

;;

quit)

echo The System exit.

exit

esac

done

1.12

Shell编程函数默认不能将参数传入“( )”内部，Shell函数参数跟随函数名称传递，例如name args1 args2。

function name (){

command1

command2

........

}

name args1 args2

（1）创建Apache软件安装函数，给函数Apache\_install传递参数1。

#!/bin/bash

#auto install LAMP

#By author jfedu.net 2021

#Httpd define path variable

H\_FILES=httpd-2.2.31.tar.bz2

H\_FILES\_DIR=httpd-2.2.31

H\_URL=http://mirrors.cnnic.cn/apache/httpd/

H\_PREFIX=/usr/local/apache2/

function Apache\_install()

{

#Install httpd Web Server

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

echo -e "\n\033[32m-----------------------------------------------\

033[0m"

echo -e "\033[32mThe $H\_FILES\_DIR Server Install Success!\033[0m"

else

echo -e "\033[32mThe $H\_FILES\_DIR Make or Make install ERROR,Please

Check......"

exit 0

fi

fi

}

Apache\_install 1

（2）创建judge\_ip函数判断IP函数。

#!/bin/bash

#By author jfedu.net 2021

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}"

}

judge\_ip

1.13.1

find工具主要用于操作系统文件和目录的查找，其语法参数格式如下：

find path -option [ -print ] [ -exec -ok command ] { } \;

其中option常用参数详解如下：

-name filename #查找名为filename的文件

-type b/d/c/p/l/f #查是块设备、目录、字符设备、管道、符号链接、普通文件

-size n[c] #查长度为n块[或n字节]的文件

-perm #按执行权限查找

-user username #按文件属主查找

-group groupname #按组查找

-mtime -n +n #按文件更改时间查找文件,-n指n天以内,+n指n天以前

-atime -n +n #按文件访问时间查找文件

-ctime -n +n #按文件创建时间查找文件

-mmin -n +n #按文件更改时间查找文件,-n指n min以内,+n指n min以前

-amin -n +n #按文件访问时间查找文件

-cmin -n +n #按文件创建时间查找文件

-nogroup #查无有效属组的文件

-nouser #查无有效属主的文件

-newer f1 !f2 #找文件,-n指n天以内,+n指n天以前

-depth #使查找在进入子目录前先行查找完本目录

-fstype #查更改时间比f1新但比f2旧的文件

-mount #查文件时不跨越文件系统mount点

-follow #如果遇到符号链接文件,就跟踪链接所指的文件

-cpio #查位于某一类型文件系统中的文件

-prune #忽略某个目录

-maxdepth #查找目录级别深度

（1）find工具-name参数案例如下：

find /data/ -name "\*.txt" #查找/data/目录以.txt结尾的文件

find /data/ -name "[A-Z]\*" #查找/data/目录以大写字母开头的文件

find /data/ -name "test\*" #查找/data/目录以test开头的文件

（2）find工具-type参数案例如下：

find /data/ -type d #查找/data/目录下的文件夹

find /data/ ! -type d #查找/data/目录下的非文件夹

find /data/ -type l #查找/data/目录下的链接文件

find /data/ -type d|xargs chmod 755 –R #查目录类型并将权限设置为755

find /data/ -type f|xargs chmod 644 –R #查文件类型并将权限设置为644

（3）find工具-size参数案例如下：

find /data/ -size +1M #查找大于1MB的文件

find /data/ -size 10M #查找大小为10MB的文件

find /data/ -size -1M #查找小于1MB的文件

（4）find工具-perm参数案例如下：

find /data/ -perm 755 #查找/data/目录权限为755的文件或目录

find /data/ -perm -007 #与-perm 777相同,表示所有权限

find /data/ -perm +644 #文件权限符号644以上

（5）find工具-mtime参数案例如下：

atime,access time #文件被读取或者执行的时间

ctime,change time #文件状态改变时间

mtime,modify time #文件内容被修改的时间

find /data/ -mtime +30 -name "\*.log" #查找30天以前的log文件

find /data/ -mtime -30 -name "\*.txt" #查找30天以内的log文件

find /data/ -mtime 30 -name "\*.txt" #查找第30天的log文件

find /data/ -mmin +30 -name "\*.log" #查找30min以前修改的log文件

find /data/ -amin -30 -name "\*.txt" #查找30min以内被访问的log文件

find /data/ -cmin 30 -name "\*.txt" #查找第30min改变的log文件

（6）find工具参数综合案例如下：

#查找/data目录以.log结尾,文件大于10KB的文件,同时复制到/tmp目录

find /data/ -name "\*.log" –type f -size +10k -exec cp {} /tmp/ \;

#查找/data目录以.txt结尾,文件大于10KB的文件,权限为644并删除该文件

find /data/ -name "\*.log" –type f -size +10k -m perm 644 -exec rm –rf {} \;

#查找/data目录以.log结尾、30天以前的、大于10MB的文件并移动到/tmp目录

find /data/ -name "\*.log" –type f -mtime +30 –size +10M -exec mv {} /tmp/ \;

1.13.2

逐行处理直到文件末尾，然而如果打印在屏幕上，实质文件内容并没有改变，除非使用重定向存储输出或写入文件。其语法参数格式如下：

sed [-Options] ['Commands'] filename;

#sed工具默认处理文本,文本内容输出屏幕已经修改,但是文件内容其实没有修改,需要加-i参

#数对文件彻底修改

x #x为指定行号

x,y #指定从x到y的行号范围

/pattern/ #查询包含模式的行

/pattern/pattern/ #查询包含两个模式的行

/pattern/,x #从与pattern的匹配行到x号行之间的行

x,/pattern/ #从x号行到与pattern的匹配行之间的行

x,y! #查询不包括x和y行号的行

r #从另一个文件中读文件

w #将文本写入到一个文件

y #变换字符

q #第一个模式匹配完成后退出

l #显示与八进制ASCII码等价的控制字符

{} #在定位行执行的命令组

p #打印匹配行

= #打印文件行号

a\ #在定位行号之后追加文本信息

i\ #在定位行号之前插入文本信息

d #删除定位行

c\ #用新文本替换定位文本

s #使用替换模式替换相应模式

n #读取下一个输入行,用下一个命令处理新的行

N #将当前读入行的下一行读取到当前模式空间

h #将模式缓冲区的文本复制到保持缓冲区

H #将模式缓冲区的文本追加到保持缓冲区

x #互换模式缓冲区和保持缓冲区的内容

g #将保持缓冲区的内容复制到模式缓冲区

G #将保持缓冲区的内容追加到模式缓冲区

常用sed工具企业演练案例如下。

（1）替换jfedu.txt文本中的old为new。

sed 's/old/new/g' jfedu.txt

（2）打印jfedu.txt文本中的第1～3行。

sed -n '1,3p' jfedu.txt

（3）打印jfedu.txt文本中的第1行与最后一行。

sed -n '1p;$p' jfedu.txt

（4）删除jfedu.txt第1～3行，删除匹配行至最后一行。

sed '1,3d' jfedu.txt

sed '/jfedu/,$d' jfedu.txt

（5）删除jfedu.txt最后6行及删除最后一行。

for i in 'seq 1 6';do sed -i '$d' jfedu.txt ;done

sed '$d' jfedu.txt

（6）删除jfedu.txt最后一行。

sed '$d' jfedu.txt

（7）在jfedu.txt查找jfedu所在行，并在其下一行添加word字符，a表示在其下一行添加字符串。

sed '/jfedu/aword' jfedu.txt

（8）在jfedu.txt查找jfedu所在行，并在其上一行添加word字符，i表示在其上一行添加字符串。

sed '/jfedu/iword' jfedu.txt

（9）在jfedu.txt查找以test结尾的行尾添加字符串word，$表示结尾标识，&在sed中表示添加。

sed 's/test$/&word/g' jfedu.txt

（10）在jfedu.txt查找www的行，在其行首添加字符串word，^表示起始标识，&在sed中表示添加。

sed '/www/s/^/&word/' jfedu.txt

（11）多个sed命令组合，使用-e参数。

sed -e '/www.jd.com/s/^/&1./' -e 's/www.jd.com$/&./g' jfedu.txt

（12）多个sed命令组合，使用分号“；”分隔。

sed -e '/www.jd.com/s/^/&1./;s/www.jd.com$/&./g' jfedu.txt

（13）sed读取系统变量，变量替换。

WEBSITE=WWW.JFEDU.NET

sed "s/www.jd.com/$WEBSITE/g" jfedu.txt

（14）修改Selinux策略enforcing为disabled，查找/SELINUX/行，然后将其行enforcing值改成disabled、!s表示不包括SELINUX行。

sed -i '/SELINUX/s/enforcing/disabled/g' /etc/selinux/config

sed -i '/SELINUX/!s/enforcing/disabled/g' /etc/selinux/config

通常而言，sed将待处理的行读入模式空间，脚本中的命令逐行进行处理，直到脚本执行完毕，然后该行被输出，模式空间清空；重复上述动作，文件中的新的一行被读入，直到文件处理完毕。

如果希望在某个条件下脚本中的某个命令被执行，或者希望模式空间得到保留以便下一次处理，都可以使sed在处理文件时不按照正常的流程来进行。这时可以使用sed高级语法来满足用户需求。sed高级命令可以分为3种功能。

（1）N、D、P：处理多行模式空间的问题。

（2）H、h、G、g、x：将模式空间的内容放入存储空间以便接下来的编辑。

（3）:、b、t：在脚本中实现分支与条件结构。

① 在jfedu.txt每行后加入空行，每行后插入一行空行。

sed '/^$/d;G' jfedu.txt

② 将jfedu.txt偶数行删除及隔两行删除一行。

sed 'n;d' jfedu.txt

sed 'n;n;d' jfedu.txt

③ 在jfedu.txt匹配行前一行、后一行插入空行，同时在匹配前后插入空行。

sed '/jfedu/{x;p;x;}' jfedu.txt

sed '/jfedu/G' jfedu.txt

sed '/jfedu/{x;p;x;G;}' jfedu.txt

④ 在jfedu.txt每行后加入空行，即每行占据两行空间，每一行后插入空行。

sed '/^$/d;G' jfedu.txt

⑤ 在jfedu.txt每行前加入顺序数字序号，加上制表符\t及“.”符号。

sed = jfedu.txt| sed 'N;s/\n/ /'

sed = jfedu.txt| sed 'N;s/\n/\t/'

sed = jfedu.txt| sed 'N;s/\n/\./'

⑥ 删除jfedu.txt行前和行尾的任意空格。

sed 's/^[ \t]\*//;s/[ \t]\*$//' jfedu.txt

⑦ 打印jfedu.txt关键词old与new之间的内容。

sed -n '/old/,/new/'p jfedu.txt

⑧ 打印及删除jfedu.txt最后两行。

sed '$!N;$!D' jfedu.txt

sed 'N;$!P;$!D;$d' jfedu.txt

⑨ 合并上下两行，即两行合并。

sed '$!N;s/\n/ /' jfedu.txt

sed 'N;s/\n/ /' jfedu.txt

1.13.3

awk的基本原理是逐行处理文件中的数据，查找与命令行中所给定内容相匹配的模式，如果发现匹配内容，则进行下一个编程步骤；如果找不到匹配内容，则继续处理下一行。awk常用参数、变量、函数详解如下：

awk 'pattern + {action}' file

（1）awk基本语法参数详解。

① 单引号' '是为了和Shell命令区分开。

② 大括号{ }表示一个命令分组。

③ pattern是一个过滤器，表示匹配pattern条件的行才进行action处理。

④ action是处理动作，常见动作为print。

⑤ 使用#作为注释，pattern和action可以只有其一，但不能二者都没有。

（2）awk内置变量详解。

FS #分隔符,默认是空格

OFS #输出分隔符

NR #当前行数,从1开始

NF #当前记录字段个数

$0 #当前记录

$1~$n #当前记录第n个字段(列)

（3）awk内置函数详解。

gsub(r,s) #在$0中用s代替r

index(s,t) #返回s中t的第一个位置

length(s) #s的长度

match(s,r) #s是否匹配r

split(s,a,fs) #在fs上将s分成序列a

substr(s,p) #返回s从p开始的子串

（4）awk常用操作符、运算符及判断符详解。

++ -- #增加与减少(前置或后置)

^ \*\* #指数(右结合性)

! + - #非、一元(unary) 加号、一元减号

+ - \* / % #加、减、乘、除、余数

< <= == != > >= #数字比较

&& #逻辑and

|| #逻辑or

= += -= \*= /= %= ^= \*\*= #赋值

（5）awk与流程控制语句。

if(condition) { } else { }

while { }

do{ }while(condition)

for(init;condition;step){ }

break/continue

常用awk工具企业演练案例如下。

（1）awk打印硬盘设备名称，默认以空格分隔。

df -h|awk '{print $1}'

（2）awk以空格、冒号、\t、分号分隔。

awk -F '[ :\t;]' '{print $1}' jfedu.txt

（3）awk以冒号分隔，打印第1列，同时将内容追加到/tmp/awk.log下。

awk -F: '{print $1 >>"/tmp/awk.log"}' jfedu.txt

（4）打印jfedu.txt文件中的第3～5行，NR表示打印行，$0表示文本所有域。

awk 'NR==3,NR==5 {print}' jfedu.txt

awk 'NR==3,NR==5 {print $0}' jfedu.txt

（5）打印jfedu.txt文件中的第3～5行的第1列与最后一列。

awk 'NR==3,NR==5 {print $1,$NF}' jfedu.txt

（6）打印jfedu.txt文件中长度大于80的行号。

awk 'length($0)>80 {print NR}' jfedu.txt

（7）awk引用Shell变量，使用-v或双引号+单引号即可。

awk -v STR=hello '{print STR,$NF}' jfedu.txt

STR="hello";echo| awk '{print "'${STR}'";}'

（8）awk以冒号分隔，打印第1列同时只显示前5行。

cat /etc/passwd|head -5|awk -F: '{print $1}'

awk -F: 'NR>=1&&NR<=5 {print $1}' /etc/passwd

（9）awk指定文件jfedu.txt第1列的总和。

cat jfedu.txt |awk '{sum+=$1}END{print sum}'

（10）awk NR行号除以2余数为0则跳过该行，继续执行下一行，打印至屏幕。

awk -F: 'NR%2==0 {next} {print NR,$1}' /etc/passwd

（11）awk添加自定义字符。

ifconfig eth0|grep "Bcast"|awk '{print "ip\_"$2}'

（12）awk格式化输出passwd内容，printf打印字符串，%格式化输出分隔符，s表示字符串类型，-12表示12个字符，-6表示6个字符。

awk -F: '{printf "%-12s %-6s %-8s\n",$1,$2,$NF}' /etc/passwd

（13）awk OFS输出格式化\t。

netstat -an|awk '$6 ~ /LISTEN/&&NR>=1&&NR<=10 {print NR,$4,$5,$6}' OFS="\t"

（14）awk与if组合实战，判断数字比较。

echo 3 2 1 | awk '{ if(($1>$2)||($1>$3)) { print $2} else {print $1} }'

（15）awk与数组组合实战，统计passwd文件用户数。

awk -F ':' 'BEGIN {count=0;} {name[count] = $1;count++;}; END{for (i = 0;

i < NR; i++) print i, name[i]}' /etc/passwd

（16）awk分析Nginx访问日志的状态码404、502等错误信息页面，统计次数大于20的IP地址。

awk '{if ($9~/502|499|500|503|404/) print $1,$9}' access.log|sort|uniq –

c|sort –nr | awk '{if($1>20) print $2}'

（17）用/etc/shadow文件中的密文部分替换/etc/passwd中的"x"位置，生成新的/tmp/passwd文件。

awk 'BEGIN{OFS=FS=":"} NR==FNR{a[$1]=$2}NR>FNR{$2=a[$1];print >>"/tmp/

passwd"}' /etc/shadow /etc/passwd

（18）awk统计服务器状态连接数。

netstat -an | awk '/tcp/ {s[$NF]++} END {for(a in s) {print a,s[a]}}'

netstat -an | awk '/tcp/ {print $NF}' | sort | uniq -c

1.13.4

目前Linux操作系统默认使用GNU版本的grep。它的功能更强，可以通过-G、-E、-F命令行选项使用egrep和fgrep的功能。其语法格式及常用参数详解如下：

grep -[acinv]   'word'   Filename

grep常用参数详解如下：

-a #以文本文件方式搜索

-c #计算找到符合行的次数

-i #忽略大小写

-n #顺便输出行号

-v #反向选择,即显示不包含匹配文本的所有行

-h #查询多文件时不显示文件名

-l #查询多文件时只输出包含匹配字符的文件名

-s #不显示不存在或无匹配文本的错误信息

-E #允许使用egrep扩展模式匹配

学习grep时，需要了解通配符、正则表达式两个概念。通配符主要用在[Linux](http://lib.csdn.net/base/linux" \t "_blank" \o "Linux知识库)的Shell命令中，常用于文件或文件名称的操作，而正则表达式则用于文本内容中的字符串搜索和替换，常用在awk、grep、sed、VIM工具中对文本的操作。

通配符类型详解如下：

\* #0个或者多个字符、数字

? #匹配任意一个字符

# #表示注解

| #管道符号

; #多个命令连续执行

& #后台运行指令

! #逻辑运算非

[ ] #内容范围,匹配括号中的内容

{ } #命令块,多个命令匹配

正则表达式详解如下：

\* #前一个字符匹配0次或多次

. #匹配除了换行符以外任意一个字符

.\* #代表任意字符

^ #匹配行首,即以某个字符开头

$ #匹配行尾,即以某个字符结尾

\(..\) #标记匹配字符

[] #匹配中括号里的任意指定字符,但只匹配一个字符

[^] #匹配除中括号以外的任意一个字符

\ #转义符,取消特殊含义

\< #锚定单词的开始

\> #锚定单词的结束

{n} #匹配字符出现n次

{n,} #匹配字符出现大于等于n次

{n,m} #匹配字符至少出现n次,最多出现m次

\w #匹配文字和数字字符

\W #\w的反置形式,匹配一个或多个非单词字符

\b #单词锁定符

\s #匹配任何空白字符

\d #匹配一个数字字符,等价于[0-9]

常用grep工具企业演练案例如下：

grep -c "test" jfedu.txt #统计test字符总行数

grep -i "TEST" jfedu.txt #不区分大小写查找test所有的行

grep -n "test" jfedu.txt #打印test的行及行号

grep -v "test" jfedu.txt #不打印test的行

grep "test[53]" jfedu.txt #以字符test开头,接5或3的行

grep "^[^test]" jfedu.txt #显示输出行首不是test的行

grep "[Mm]ay" jfedu.txt #匹配以M或m开头的行

grep "K…D" jfedu.txt #匹配K,3个任意字符,紧接D的行

grep "[A-Z][9]D" jfedu.txt #匹配大写字母,紧跟9D的字符行

grep "T\{2,\}" jfedu.txt #打印字符T字符连续出现2次以上的行

grep "T\{4,6\}" jfedu.txt #打印字符T字符连续出现4次及6次的行

grep -n "^$" jfedu.txt #打印空行的所在的行号

grep -vE "#|^$" jfedu.txt #不匹配文件中的#和空行

grep --color -ra -E "db|config|sql" \* #匹配包含db或config或sql的文件

grep --color -E "\<([0-9]{1,3}\.){3}([0-9]{1,3})\>" #jfedu.txt 匹配

#IPv4地址

1.14

（1）一维数组定义及创建。

JFTEST=(

test1

test2

test3

)

LAMP=(httpd php php-devel php-mysql mysql mysql-server)

（2）数组下标一般从0开始，以下为引用数组的方法。

echo ${JFTEST[0]} #引用第1个数组变量,结果打印test1

echo ${JFTEST[1]} #引用第2个数组变量

echo ${JFTEST[@]} #显示该数组所有参数

echo ${#JFTEST[@]} #显示该数组参数个数

echo ${#JFTEST[0]} #显示test1字符长度

echo ${JFTEST[@]:0} #打印数组所有的值

echo ${JFTEST[@]:1} #打印第2个值开始的所有值

echo ${JFTEST[@]:0:2} #打印第1个值与第2个值

echo ${JFTEST[@]:1:2} #打印第2个值与第3个值

（3）数组替换操作。

JFTEST=( [0]=www1 [1]=www2 [2]=www3 ) #数组赋值

echo ${JFTEST[@]/test/jfedu} #将数组值test替换为jfedu

NEWJFTEST='echo ${JFTEST[@]/test/jfedu}' #将结果赋值新数组

（4）数组删除操作。

unset array[0] #删除数组第1个值

unset array[1] #删除数组第2个值

unset array #删除整个数组

数组Shell脚本企业案例1：网卡绑定脚本。

#!/bin/bash

#Auto Make KVM Virtualization

#Auto config bond scripts

#By author jfedu.net 2021

eth\_bond()

{

NETWORK=(

HWADDR='ifconfig eth0 |egrep "HWaddr|Bcast" |tr "\n" " "|awk '{print

$5,$7,$NF}'|sed -e 's/addr://g' -e 's/Mask://g'|awk '{print $1}''

IPADDR='ifconfig eth0 |egrep "HWaddr|Bcast" |tr "\n" " "|awk '{print

$5,$7,$NF}'|sed -e 's/addr://g' -e 's/Mask://g'|awk '{print $2}''

NETMASK='ifconfig eth0 |egrep "HWaddr|Bcast" |tr "\n" " "|awk '{print

$5,$7,$NF}'|sed -e 's/addr://g' -e 's/Mask://g'|awk '{print $3}''

GATEWAY='route -n|grep "UG"|awk '{print $2}''

)

cat >ifcfg-bond0<<EOF

DEVICE=bond0

BOOTPROTO=static

${NETWORK[1]}

${NETWORK[2]}

${NETWORK[3]}

ONBOOT=yes

TYPE=Ethernet

NM\_CONTROLLED=no

EOF

数组Shell脚本企业案例2：定义IPv4值。

#!/bin/bash

#auto Change ip netmask gateway scripts

#By author jfedu.net 2021

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 "----------------------------"

count\_ip(){

count=('echo $IPADDR|awk -F. '{print $1,$2,$3,$4}'')

IP1=${count[0]}

IP2=${count[1]}

IP3=${count[2]}

IP4=${count[3]}

}

1. Shell编程高级企业实战

2.1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.19

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

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

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

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

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

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

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

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

第四章Puppet自动运维企业实战

4.3

（1）Puppet服务器端安装。

由于Puppet主要是基于hostname检测的，所以Puppet服务器端需修改主机名称为192-168-149-128-jfedu.net，并在hosts文件中添加主机名和本机IP的对应关系，如果本地局域网有DNS服务器，则无须修改hosts文件。修改主机名及配置hosts代码如下：

hostname 'ifconfig eth0 |grep Bcast|awk '{print $2}'|cut -d: -f 2 |sed

's/\./\-/g''-jfedu.net

cat >>/etc/hosts<<EOF

192.168.149.128 192-168-149-128-jfedu.net

192.168.149.130 192-168-149-130-jfedu.net

EOF

Puppet服务器端除了需要安装Puppet外，还需要Ruby语言的支持，需要安装Ruby相关软件包。默认YUM安装Puppet，会自动下载并安装Ruby相关软件。相关代码如下，运行结果如图4-2所示。

rpm -Uvh http://yum.puppetlabs.com/el/6/products/x86\_64/puppetlabs-

release-6-1.noarch.rpm

yum install puppet-server -y

/etc/init.d/puppetmaster start

/etc/init.d/iptables stop

sed -i '/SELINUX/S/enforce/disabled/' /etc/selinux/config

setenforce 0

（2）Puppet客户端安装。

Puppet主要是基于hostname检测的，所以Puppet客户端也需要修改主机名称为192-168-149-130-jfedu.net，并在hosts文件中添加主机名和本机IP的对应关系，如果本地局域网有DNS服务器，则无须修改hosts文件。修改主机名及配置hosts代码如下：

hostname 'ifconfig eth0 |grep Bcast|awk '{print $2}'|cut -d: -f 2 |sed

's/\./\-/g''-jfedu.net

cat >>/etc/hosts<<EOF

192.168.149.128 192-168-149-128-jfedu.net

192.168.149.130 192-168-149-130-jfedu.net

EOF

Puppet客户端除了需要安装Puppet外，还需要Ruby语言的支持，需要安装Ruby相关软件包。默认YUM安装Puppet，会自动下载并安装Ruby相关软件。相关代码如下，运行结果如图4-3所示。

rpm -Uvh http://yum.puppetlabs.com/el/6/products/x86\_64/puppetlabs-

release-6-1.noarch.rpm

yum install puppet -y

/etc/init.d/puppetmaster start

/etc/init.d/iptables stop

sed -i '/SELINUX/S/enforce/disabled/' /etc/selinux/config

setenforce 0

（3）Puppet客户端申请证书。

由于Puppet客户端与Puppet服务器端是通过SSL隧道通信的，客户端安装完成后，首次使用需向服务器端申请Puppet通信证书。Puppet客户端第一次连接服务器端会发起证书申请，在Puppet客户端执行命令如下，运行结果如图4-4所示。

puppet agent --server 192-168-149-128-jfedu.net --test

（4）Puppet服务器端颁发证书。

Puppet客户端向服务器发起证书申请，服务器端必须审核证书，如果不审核，客户端与服务器端将无法进行后续正常通信。Puppet服务器端颁发证书命令代码如下，运行结果如图4-5所示。

puppet cert --list #查看申请证书的客户端主机名

puppet cert -s 192-168-149-130-jfedu.net #颁发证书给客户端

puppet cert -s #为特定的主机颁发证书

puppet cert -s and -a #给所有的主机颁发证书

puppet cert --list --all #查看已经颁发的所有证书

4.4.

在Puppet客户端创建test.txt文件，并在该文件中写入测试内容，操作方法如下。

（1）Puppet服务器端创建node代码，创建或编辑/etc/puppet/manifests/site.pp文件，在文件中加入以下代码：

node default {

file {

"/tmp/test.txt":

content => "Hello World,jfedu.net 2021";

}

}

site.pp配置文件代码详解如下：

node default #新建node节点,default表示所有主机,可修改为特定主机名

file #基于file资源模块管理客户端文件或者目录操作

"/tmp/test.txt": #需在客户端文件创建的文件名

content #客户端服务器文件内容

（2）客户端执行同步命令，获取Puppet服务器端node配置，代码如下，运行结果如图4-6所示。

puppet agent --server=192-168-149-128-jfedu.net --test

报错原因是服务器端与客户端时间不同步，需要同步时间，代码如下，然后再次执行puppet agent命令，如图4-7所示。

ntpdate pool.ntp.org

puppet agent --server=192-168-149-128-jfedu.net --test

Puppet客户端执行同步，执行日志如下，会在/tmp/目录创建test.txt文件，内容为“Hello World，jfedu.net”，即证明Puppet客户端成功获取服务器端Node配置。

Info: Caching certificate\_revocation\_list for ca

Warning: Unable to fetch my node definition, but the agent run will continue:

Warning: undefined method 'include?' for nil:NilClass

Info: Retrieving pluginfacts

Info: Retrieving plugin

Info: Caching catalog for 192-168-149-130-jfedu.net

Info: Applying configuration version '1496805041'

Notice: /Stage[main]/Main/Node[default]/File[/tmp/test.txt]/ensure:

defined content as '{md5}d1c2906ad0b249a330e936e3bc1d38d9'

Info: Creating state file /var/lib/puppet/state/state.yaml

Notice: Finished catalog run in 0.04 seconds

4.6

（1）从Puppet服务器下载nginx.conf文件至客户端/tmp目录，首先需要将nginx.conf文件复制至/etc/puppet/files目录，然后在/etc/puppet/fileserver.conf中添加如下3行代码，并重启puppet master即可。

[files]

path /etc/puppet/files/

allow \*

创建site.pp文件，文件代码如下：

node default {

file {

'/tmp/nginx.conf':

mode => '644',

owner => 'root',

group => 'root',

source => 'puppet://192-168-149-128-jfedu.net/files/nginx.conf',

}

}

（2）从Puppet服务器下载sysctl.conf，如果客户端该文件存在，则备份为sysctl.conf.bak，然后再覆盖原文件。site.pp代码如下，运行结果如图4-11所示。

node default {

file {

"/etc/sysctl.conf":

source => "puppet://192-168-149-128-jfedu.net/files/sysctl.conf",

backup => ".bak\_$uptime\_seconds",

}

}

（3）在Agent上创建/export/docker的软链接为/var/lib/docker/，site.pp代码如下，运行结果如图4-12所示。

node default {

file {

"/var/lib/docker":

ensure => link,

target => "/export/docker",

}

（4）在Agent上创建目录/tmp/20501212，site.pp代码如下，运行结果如图4-13所示。

node default {

file {

"/tmp/20501212":

ensure => directory;

}

}

4.7

（1）客户端安装ntpdate及screen软件，代码如下，运行结果如图4-14所示。

node default {

package {

["screen","ntp"]:

ensure => "installed";

}

（2）客户端卸载ntpdate及screen软件，代码如下，运行结果如图4-15所示。

node default {

package {

["screen","ntp"]:

ensure => "absent";

}

4.8

（1）启动Agent httpd服务，停止nfs服务，代码如下，运行结果如图4-16所示。

node default {

service {

"httpd":

ensure => running;

"nfs":

ensure => stopped;

}

（2）启动Agent httpd服务并设置开机启动；停止nfs服务，设置为开机不启动，代码如下，运行结果如图4-17所示。

node default {

service {

"httpd":

ensure => running,

enable => true;

"nfs":

ensure => stopped,

enable => false;

}

4.9

（1）Agent服务器执行命令tar解压nginx软件包，代码如下，运行结果如图4-18所示。

node default {

exec {

'Agent tar xzf nginx-1.12.0.tar.gz':

path => ["/usr/bin","/bin"],

user => 'root',

group => 'root',

timeout => '10',

command => 'tar -xzf /tmp/nginx-1.12.0.tar.gz',

}

}

（2）Agent服务器远程执行auto\_install\_nginx.sh脚本，代码如下，运行结果如图4-19所示。

node default {

file {

"/tmp/auto\_install\_nginx.sh":

source =>"puppet://192-168-149-128-jfedu.net/files/auto\_install\_

nginx.sh",

owner => "root",

group => "root",

mode => 755,

}

exec {

"/tmp/auto\_install\_nginx.sh":

cwd => "/tmp",

user => root,

path => ["/usr/bin","/usr/sbin","/bin","/bin/sh"],

}

（3）Agent服务器更新sysctl.conf，如果该文件发生改变，则执行命令sysctl -p，代码如下，运行结果如图4-20所示。

node default {

file {

"/etc/sysctl.conf":

source =>"puppet://192-168-149-128-jfedu.net/files/sysctl.conf",

owner => "root",

group => "root",

mode => 644,

}

exec {

"sysctl refresh kernel config":

path => ["/usr/bin", "/usr/sbin", "/bin", "/sbin"],

command => "/sbin/sysctl -p",

subscribe => File["/etc/sysctl.conf"],

refreshonly => true

}

}

4.10

（1）Agent服务器添加ntpdate时间同步任务，代码如下，结果如图4-21所示。

node default {

cron{

"ntpdate":

command => "/usr/sbin/ntpdate pool.ntp.org",

user => root,

hour => 0,

minute => 0,

}

}

（2）Agent服务器删除ntpdate时间同步任务，代码如下，结果如图4-22所示。

node default {

cron{

"ntpdate":

command => "/usr/sbin/ntpdate pool.ntp.org",

user => root,

hour => 0,

minute => 0,

ensure => absent,

}

}

4.11.1

自动颁发证书的前提是服务器端与客户端能ping通彼此的主机名，配置自动颁发证书需在Puppet服务器端的puppet.conf配置文件main段加入如下代码，如图4-23所示。

[main]

autosign = true

重启puppetmaster服务，并删除192.168.149.130证书。

/etc/init.d/puppetmaster restart

puppet cert --clean 192-168-149-130-jfedu.net

删除Puppet客户端SSL文件，重新生成SSL文件，执行如下命令自动申请证书。

rm -rf /var/lib/puppet/ssl/

puppet agent --server=192-168-149-128-jfedu.net --test

4.11.2

Puppet客户端配置相关参数和同步时间，修改/etc/sysconfig/puppet配置文件，最终代码如下：

# The puppetmaster server

PUPPET\_SERVER=192-168-149-128-jfedu.net

# If you wish to specify the port to connect to do so here

PUPPET\_PORT=8140

# Where to log to. Specify syslog to send log messages to the system log.

PUPPET\_LOG=/var/log/puppet/puppet.log

# You may specify other parameters to the puppet client here

PUPPET\_EXTRA\_OPTS=--waitforcert=500

/etc/sysconfig/puppet配置文件参数详解如下：

PUPPET\_SERVER=192-168-149-128-jfedu.net #指定Puppet Master主机名

PUPPET\_PORT=8140 #指定Puppet Master端口

PUPPET\_LOG=/var/log/puppet/puppet.log #Puppet客户端日志路径

PUPPET\_EXTRA\_OPTS=--waitforcert=500 #获取Puppet Master证书返回等待时间

重启Puppet客户端服务，客户端会每半小时与服务器同步一次配置信息。

/etc/init.d/puppet restart

可以修改与服务器端同步配置信息的时间，修改vi /etc/puppet/puppet.conf文件，在[agent]段加入如下语句，表示60s与Puppet Master同步一次配置信息。重启Puppet，同步结果如图4-25所示。

[agent]

runinterval = 60

4.11.3

修改Puppet客户端配置文件/etc/puppet/puppet.conf，在agent段加入如下代码：

[agent]

listen = true

修改Puppet客户端配置文件/etc/sysconfig/puppet，指定Puppet Master端主机名。

PUPPET\_SERVER=192-168-149-128-jfedu.net

创建Puppet客户端配置文件namespaceauth.conf，写入如下代码：

[puppetrunner]

allow \*

修改Puppet客户端配置文件auth.conf，在“path /”前添加如下代码：

path /run

method save

allow \*

重启Puppet客户端。

/etc/init.d/puppet restart

Puppet服务器端执行如下命令，通知客户端同步配置，也可以批量通知其他客户端，只需将客户端的主机名写入host.txt文件，如图4-26所示。

puppet kick -d 192-168-149-130-jfedu.net

#puppet kick -d 'cat host.txt'

4.12.1

（1）修改IP为静态IP的Shell脚本代码如下：

#!/bin/bash

#auto Change ip netmask gateway scripts

#By author jfedu.net 2021

#Define Path variables

ETHCONF=/etc/sysconfig/network-scripts/ifcfg-eth0

DIR=/data/backup/'date +%Y%m%d'

IPADDR='ifconfig|grep inet|grep 192|head -1|cut -d: -f2|awk '{print $1}''

NETMASK=255.255.255.0

grep dhcp $ETHCONF

if [ $? -eq 0 ];then

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. !"

service network restart

fi

（2）Puppet Master执行kick命令推送配置至Agent服务器远程，Puppet客户端修改IP脚本代码如下，结果如图4-27所示。

node default {

file {

"/tmp/auto\_change\_ip.sh":

source =>"puppet://192-168-149-128-jfedu.net/files/auto\_change\_

ip.sh",

owner => "root",

group => "root",

mode => 755,

}

exec {

"/tmp/auto\_change\_ip.sh":

cwd => "/tmp/",

user => root,

path => ["/usr/bin","/usr/sbin","/bin","/bin/sh"],

}

}

4.12.2

（1）Puppet Master上创建客户端node配置。可以编写NTP模块，使用class可以定义模块分组，对不同业务进行分组管理，/etc/puppet/modules/ntp/manifests/init.pp配置文件代码如下，将原ntpdate同步时间从0:0分改成每5min同步一次时间，并修改原pool.ntp.org服务器为本地局域网NTP时间服务器的IP地址。

class ntp {

Exec { path =>"/bin:/sbin:/bin/sh:/usr/bin:/usr/sbin:/usr/local/bin:/usr/

local/sbin"}

exec {

"auto change crontab ntp config":

command =>"sed -i -e '/ntpdate/s/0/\*\/5 /2' -e 's/pool.ntp.org/10.1.1.21/'

/var/spool/cron/root",

}

}

（2）在/etc/puppet/manifests目录创建两个文件，分别为modules.pp和nodes.pp，即模块入口文件以及node配置段。

modules.pp配置文件内容如下：

import "ntp"

nodes.pp配置文件内容如下：

node default {

include ntp

}

（3）在site.pp 中加载导入modules.pp 和nodes.pp名称，site.pp代码如下：

import "modules.pp"

import "nodes.pp"

（4）Puppet Master执行kick命令推送配置至Agent服务器远程，Puppet客户端最终结果如图4-28所示。

当服务器分组之后，可以使用正则表达式进行定义node，在定义一个node节点时，要指定节点的名称，并使用单引号将名称引起来，然后在大括号中指定需要应用的配置。

客户端节点名称可以是主机名也可以是客户端的正式域名，目前Puppet版本还不能使用通配符来指定节点，例如不能用\*.jfedu.net，但可以使用正则表达式。相关代码如下：

node /^Beijing-IDC-web0\d+\-jfedu\.net {

    include ntp

}

4.12.3

（1）Puppet服务器端配置，/etc/puppet/modules/www/manifests/init.pp代码如下：

class www {

Exec { path =>"/bin:/sbin:/bin/sh:/usr/bin:/usr/sbin:/usr/local/bin:/usr/

local/sbin"}

file {

"/data/sh/rsync\_www\_client.sh":

source =>"puppet://192-9-11-162-tdt.com/files/www/rsync\_www\_client.sh",

owner =>"root",

group =>"root",

mode =>"755",

}

file {

"/etc/rsync.pas":

source =>"puppet://192-9-11-162-tdt.com/files/www/rsync.pas",

owner =>"root",

group =>"root",

mode =>"600",

}

exec {

"auto backup www data":

command =>"mkdir -p /data/backup/'date +%Y%m%d';mv /data/index /data/

backup/www/'date +%Y%m%d' ; /bin/sh /data/sh/rsync\_www\_client.sh ",

user =>"root",

subscribe =>File["/data/sh/rsync\_www\_client.sh"],

refreshonly =>"true",

}

}

（2）在/etc/puppet/manifests目录下创建两个文件，分别为modules.pp和nodes.pp，即模块入口文件以及node配置段。

modules.pp配置文件内容如下：

import "www"

nodes.pp配置文件内容如下：

node /^Beijing-IDC-web0\d+\-jfedu\.net {

include www

}

（3）在site.pp中加载导入modules.pp和nodes.pp名称，site.pp代码如下：

import "modules.pp"

import "nodes.pp"

Puppet Master端批量执行通知客户端同步配置，命令如下：

puppet kick -d --host 'cat hosts.txt'

（4）cat hosts.txt内容为需要同步的客户端的主机名。

Beijing-IDC-web01-jfedu.net

Beijing-IDC-web02-jfedu.net

Beijing-IDC-web03-jfedu.net

Beijing-IDC-web04-jfedu.net

1. Ansible自动运维企业实战

5.3

Red Hat、CentOS操作系统可以直接基于YUM工具自动安装Ansible，CentOS 6.x或CentOS 7.x安装前，需先安装epel扩展源，代码如下：

rpm -Uvh http://mirrors.ustc.edu.cn/fedora/epel/6/x86\_64/epel-release-

6-8.noarch.rpm

yum install epel-release -y

yum install ansible -y

5.6

Ansible command模块企业常用案例如下。

（1）Ansible command模块远程执行date命令如下，执行结果如图5-7所示。

ansible -k -i /etc/ansible/hosts all -m command -a "date"

（2）Ansible command模块远程执行ping命令如下，执行结果如图5-8所示。

ansible -k all -m command -a "ping -c 1 www.baidu.com"

（3）Ansible Hosts正则模式远程执行df -h命令如下，执行结果如图5-9所示。

ansible -k 192.168.149.13\* -m command -a "df -h"

5.7

（1）操作代码如下，其中src表示源文件，dest表示目标目录或者文件，owner指定拥有者，执行结果如图5-10所示。

ansible -k all -m copy -a 'src=/etc/passwd dest=/tmp/ mode=755 owner=

root'

（2）操作代码如下，其中content为文件内容，dest为目标文件，owner指定拥有者，执行结果如图5-11所示。

ansible -k all -m copy -a 'content="Hello World" dest=/tmp/jfedu.txt

mode=755 owner=root'

（3）操作代码如下，其中content为文件内容，dest为目标文件，owner指定拥有者，backup= yes开启备份，执行结果如图5-12所示。

ansible -k all -m copy -a 'content="Hello World" dest=/tmp/jfedu.txt

backup=yes mode=755 owner=root'

5.8

Ansible YUM模块企业常用案例如下。

（1）操作代码如下，其中name表示需安装的软件名称，state表示状态，state=installed表示安装软件，执行结果如图5-13所示。

ansible all -k -m yum -a "name=sysstat,screen state=installed"

（2）操作代码如下，其中name表示需安装的软件名称，state表示状态，state=absent表示安装软件，执行结果如图5-14所示。

ansible all -k -m yum -a "name=sysstat,screen state=absent"

（3）操作代码如下，其中name表示需安装的软件名称，state表示状态，state=installed表示安装软件，disable\_gpg\_check=no表示不检查key，执行结果如图5-15所示。

ansible 192.168.149.129 -k -m yum -a "name=sysstat,screen state=

installed disable\_gpg\_check=no"

5.9

Ansible file模块企业常用案例如下。

（1）操作代码如下，其中path表示目录的名称和路径，state=directory表示创建目录，执行结果如图5-16所示。

ansible -k 192.168.\* -m file -a "path=/tmp/'date +%F' state=directory

mode=755"

（2）操作代码如下，其中path表示目录的名称和路径，state=touch表示创建文件，执行结果如图5-17所示。

ansible -k 192.168.\* -m file -a "path=/tmp/jfedu.txt state=touch

mode=755"

5.10

Ansible user模块企业常用案例如下。

（1）操作代码如下，其中name表示用户名称，home表示其主目录，执行结果如图5-18所示。

ansible -k 192.168.149.\* -m user -a "name=jfedu home=/tmp/"

（2）操作代码如下，其中name表示用户名称，home表示其主目录，执行结果如图5-19所示。

vansible -k 192.168.149.\* -m user -a "name=jfedu home=/tmp/

shell=/sbin/nologin"

（3）操作代码如下，其中name表示用户名称，state=absent表示删除用户，执行结果如图5-20所示。

ansible -k 192.168.149.\* -m user -a "name=jfedu state=absent

force=yes"

5.11

Ansible cron模块企业常用案例如下。

（1）创建crontab任务计划的代码如下，执行结果如图5-21所示。

ansible -k all -m cron -a "minute=0 hour=0 day=\* month=\* weekday=

\* name='Ntpdate server for sync time' job='/usr/sbin/ntpdate 139.224.

227.121'"

（2）备份crontab任务计划，其中backup=yes表示开启备份，备份文件存放于客户端/tmp/，代码如下，执行结果如图5-22所示。

ansible -k all -m cron -a "minute=0 hour=0 day=\* month=\* weekday=

name='Ntpdate server for sync time' backup=yes job='/usr/sbin/ntpdate

pool.ntp.org'"

（3）删除crontab任务计划的代码如下，执行结果如图5-23所示。

ansible -k all -m cron -a "name='Ntpdate server for sync time'

state=absent"

5.12

Ansible synchronize模块企业常用案例如下。

（1）操作代码如下，其中src表示源目录，dest表示目标目录，执行结果如图5-24所示。

ansible -k all -m synchronize -a 'src=/tmp/ dest=/tmp/'

（2）操作代码如下，其中src表示源目录，dest表示目标目录，compress=yes表示开启压缩，delete=yes表示数据一致，rsync\_opts表示同步参数，--exclude表示排除文件，执行结果如图5-25所示。

ansible -k all -m synchronize -a 'src=/tmp/ dest=/tmp/ compress=yes

delete=yes rsync\_opts=--no-motd,--exclude=.txt'

5.13

Ansible Shell模块企业常用案例如下。

（1）操作代码如下，其中-m shell指定模块为Shell，远程执行Shell脚本。远程执行脚本也可采用script模块。把执行结果追加至客户端服务器/tmp/var.log文件，执行结果如图5-26所示。

ansible -k all -m shell -a "/bin/sh /tmp/variables.sh >>/tmp/var.log"

（2）远程执行创建目录命令，执行之前切换至/tmp目录，屏蔽警告信息，代码如下，执行结果如图5-27所示。

ansible -k all -m shell -a "mkdir -p 'date +%F' chdir=/tmp/ state=directory

warn=no"

（3）操作代码如下，其中-m shell指定模块为Shell，远程客户端查看http进程是否启动，执行结果如图5-28所示。

ansible -k all -m shell -a "ps -ef |grep http"

5.14

Ansible service模块企业常用案例如下。

（1）远程重启httpd服务，代码如下，执行结果如图5-30所示。

ansible -k all -m service -a "name=httpd state=restarted"

（2）远程重启网卡服务，指定参数eth0，代码如下，执行结果如图5-31所示。

ansible -k all -m service -a "name=network args=eth0 state=restarted"

（3）远程开机启动nfs服务，设置3,5级别自动启动，代码如下，执行结果如图5-32所示。

ansible -k all -m service -a "name=nfs enabled=yes runlevel=3,5"

5.15

Ansible Playbook案例演示如下。

（1）远程主机安装Nginx Web服务，Playbook代码如下，执行结果如图5-33所示。

- hosts: all

remote\_user: root

tasks:

- name: Jfedu Pcre-devel and Zlib LIB Install.

yum: name=pcre-devel,pcre,zlib-devel state=installed

- name: Jfedu Nginx WEB Server Install Process.

shell: cd /tmp;rm -rf nginx-1.12.0.tar.gz;wget http://nginx.org/

download/nginx-1.12.0.tar.gz;tar xzf nginx-1.12.0.tar.gz;cd nginx-1.12.

0;./configure --prefix=/usr/local/nginx;make;make install

（2）检测远程主机Nginx目录是否存在，不存在则安装Nginx Web服务，安装完成后启动Nginx，Playbook代码如下，执行结果如图5-34所示。

- hosts: all

remote\_user: root

tasks:

- name: Nginx server Install 2021

file: path=/usr/local/nginx/ state=directory

notify:

- nginx install

- nginx start

handlers:

- name: nginx install

shell: cd /tmp;rm -rf nginx-1.12.0.tar.gz;wget http://nginx.org/

download/nginx-1.12.0.tar.gz;tar xzf nginx-1.12.0

.tar.gz;cd nginx-1.12.0;./configure --prefix=/usr/local/nginx;make;make

install

- name: nginx start

shell: /usr/local/nginx/sbin/nginx

（3）检测远程主机内核参数配置文件是否更新，如果更新，则执行命令sysctl -p使内核参数生效。Playbook代码如下，执行结果如图5-35所示。

- hosts: all

remote\_user: root

tasks:

- name: Linux kernel config 2021

copy: src=/data/sh/sysctl.conf dest=/etc/

notify:

- source sysctl

handlers:

- name: source sysctl

shell: sysctl -p

（4）基于列表items多个值创建用户，代码如下，通过{{}}定义列表变量，with\_items选项传入变量的值，执行结果如图5-36所示。

- hosts: all

remote\_user: root

tasks:

- name: Linux system Add User list.

user: name={{ item }} state=present

with\_items:

- jfedu1

- jfedu2

- jfedu3

- jfedu4

（5）Ansible Playbook可以自定义template（模板文件），模板文件主要用于服务器需求不一致的情况，需要独立定义，如两台服务器安装了Nginx，安装完毕之后需将服务器A的HTTP端口改成80，服务器B的HTTP端口改成81。

① Ansible hosts文件指定不同服务器不同httpd\_port端口，代码如下：

[web]

192.168.149.128 httpd\_port=80

192.168.149.129 httpd\_port=81

② Ansible创建nginx.conf jinja2模板文件，复制文件nginx.conf nginx.conf.j2，修改listen 80为listen {{httpd\_port}}，Nginx其他配置项不变，代码如下：

cp nginx.conf nginx.conf.j2

listen {{httpd\_port}};

③ Ansible Playbook剧本YAML文件创建，代码如下：

- hosts: all

remote\_user: root

tasks:

- name: Nginx server Install 2021

file: path=/usr/local/nginx/ state=directory

notify:

- nginx install

- nginx config

handlers:

- name: nginx install

shell: cd /tmp;rm -rf nginx-1.12.0.tar.gz;wget http://nginx.org/

download/nginx-1.12.0.tar.gz;tar xzf nginx-1.12.0

.tar.gz;cd nginx-1.12.0;./configure --prefix=/usr/local/nginx;make;make

install

- name: nginx config

template: src=/data/sh/nginx.conf.j2 dest=/usr/local/nginx/conf/

nginx.conf

1. SaltStack自动运维企业实战

6.5

Master1、node1（minion）节点进行如下配置：

#添加hosts解析

cat >/etc/hosts<<EOF

127.0.0.1 localhost localhost.localdomain

192.168.1.145 master

192.168.1.146 node1

EOF

#临时关闭selinux和防火墙

sed -i '/SELINUX/s/enforcing/disabled/g' /etc/sysconfig/selinux

setenforce 0

systemctl stop firewalld.service

systemctl disable firewalld.service

#firewall-cmd --permanent --zone=public --add-port=4505-4506/tcp

#同步节点时间

yum install ntpdate -y

ntpdate pool.ntp.org

#修改对应节点主机名

hostname 'cat /etc/hosts|grep $(ifconfig|grep broadcast|awk '{print $2}'

|tail -1)|awk '{print $2}'';su

6.6

（1）Red Hat、CentOS操作系统可以直接基于YUM工具自动安装SaltStack，CentOS 6.x或CentOS 7.x安装前，需先安装SaltStack源，代码如下：

yum install -y https://repo.saltstack.com/py3/redhat/salt-py3-repo-latest.

el7.noarch.rpm

yum clean all

yum install -y salt-master salt-minion zeromq\*

systemctl enable salt-master.service

systemctl enable salt-minion.service

systemctl restart salt-master.service

systemctl restart salt-minion.service

6.14

（1）创建/srv/salt/top.sls通过正则进行匹配的示例如下：

base:

  '\*':

    - webserver

（2）通过分组名进行匹配的示例，必须要有- match: nodegroup。

base:

group1:

- match: nodegroup

- webserver

（3）通过grain模块匹配的示例，必须有- match: grain。

base:

'os:Fedora':

- match: grain

- webserver

（4）准备好top.sls文件后，编写一个state文件， /srv/salt/webserver.sls，如下所示：

apache: #标签定义

pkg: #state declaration

- installed #function declaration

注：第一行称为标签定义（ID declaration），在这里被定义为安装包的名。注意：在不同发行版软件包命名不同。

第二行称为状态定义（state declaration），在这里定义使用（pkg state module）。

第三行称为函数定义（function declaration），在这里定义使用（pkg state module）调用installed函数。

（5）最后可以在终端中执行如下命令，结果如图6-15所示。

salt '\*' state.highstate

6.14.1

在企业生产环境中，Linux系统安装完成之后，通常会将DNS设置为本地局域网的服务器IP，操作的方法和代码如下：

cat>dns.sls<<EOF

dns-config:

file.managed:

- name: /etc/resolv.conf

- source: salt://init/files/resolv.conf

- user: root

- group: root

- mode: 644

EOF

cp /etc/resolv.conf files/

6.14.2

在企业生产环境中，Linux系统安装完成之后，通常会设置时间同步策略，安装ntpd服务，操作方法和代码如下：

cat>ntp.sls<<EOF

ntp-install:

pkg.installed:

- name: ntpdate

cron-ntpdate:

cron.present:

- name: ntpdate poo.ntp.org

- user: root

- minute: 5

EOF

6.14.3

在企业生产环境中，Linux系统安装完成之后，通常会设置Selinux策略，一般会关闭Selinux，操作方法和代码如下：

mkdir -p /srv/salt/

mkdir -p init/files/

cd /srv/salt/init/

cat>selinux.sls<<EOF

selinux-config:

file.managed:

- name: /etc/selinux/config

- source: salt://init/files/selinux-config

- user: root

- group: root

- mode: 0644

EOF

cp /etc/selinux/config files/selinux-config

6.14.4

在企业生产环境中，Linux系统安装完成之后，通常会安装常见的软件包和工具，例如net-tools、gzip等，操作方法和代码如下：

cat>pkg-base.sls<<EOF

include:

- init.yum-repo

base-install:

pkg.installed:

- pkgs:

- screen

- lrzsz

- telnet

- iftop

- iotop

- sysstat

- wget

- dos2unix

- lsof

- net-tools

- unzip

- zip

- vim

- require:

- file: /etc/yum.repos.d/epel.repo

EOF

6.14.5

在企业生产环境中，Linux系统安装完成之后，通常会进行SSHD服务优化，例如关闭DNS反向查找等，操作方法和代码如下：

cat>sshd.sls<<EOF

sshd-config:

file.managed:

- name: /etc/ssh/sshd\_config

- source: salt://init/files/sshd\_config

- user: root

- gourp: root

- mode: 0600

service.running:

- name: sshd

- enable: True

- reload: True

- watch:

- file: sshd-config

EOF

cp /etc/ssh/sshd\_config files/

vim files/sshd\_config

Port 6022

UseDNS no

PermitRootLogin no

PermitEmptyPasswords no

GSSAPIAuthentication no

6.14.6

在企业生产环境中，Linux系统安装完成之后，通常会设置内核参数策略，例如修改Linux系统最大文件数等，操作方法和代码如下：

cat>limit.sls<<EOF

limit-config:

file.managed:

- name: /etc/security/limits.conf

- source: salt://init/files/limits.conf

- user: root

- group: root

- mode: 0644

EOF

cp /etc/security/limits.conf files/

echo "\* - nofile 65535" >> files/limits.conf

6.14.7

在企业生产环境中，Linux系统安装完成之后，通常会进行内核优化，例如设置相关内核参数，操作方法和代码如下：

cat>sysctl.sls<<EOF

net.ipv4.tcp\_fin\_timeout:

sysctl.present:

- value: 2

net.ipv4.tcp\_tw\_reuse:

sysctl.present:

- value: 1

net.ipv4.tcp\_tw\_recycle:

sysctl.present:

- value: 1

net.ipv4.tcp\_syncookies:

sysctl.present:

- value: 1

net.ipv4.tcp\_keepalive\_time:

sysctl.present:

- value: 600

EOF

6.14.8

在企业生产环境中，Linux系统安装完成之后，通常会设置防火墙策略，例如可以关闭Firewalld，操作方法和代码如下：

cat>firewalld.sls<<EOF

firewall-stop:

service.dead:

- name: firewalld.service

- enable: False

EOF