Expect自动化交互程序应用实践

# Expect简介

## 什么是Expect

Expect是一个用来实现自动化交互功能的软件套件，是基于TCL的脚本编程工具语言，方便学习，功能强大。

## 为什么要使用Expect

在现在的企业运维中，自动化运维已经成为运维的主流趋势，但是在很多情况下，执行系统命令或程序时，系统需要以交互式的形式要求运维人员输入指定的字符串，之后才能继续执行命令。例如，为用户设置密码时，一般情况下需要手工输入2次密码。ssh远程连接服务器时需要输入yes和密码信息，才能连接。

expect自动化交互工作流程简单说明，以此执行如下操作：

spawn 启动指定进程------>expect获取期待的关键字--->send向指定进程发送指定字符---->进程执行完毕，退出结束。

# Expect安装

rpm -qa expect查看机器是否已安装。

yum -y install expect

# 小试牛刀：实现expect自动化功能

此次准备的是3台虚拟机，IP和主机名如下

|  |  |
| --- | --- |
| IP地址 | 主机名 |
| 192.168.132.20 | salt-master |
| 192.168.132.11 | salt-minion01 |
| 192.168.132.10 | salt-minion02 |

expect脚本如下：

[root@salt-master tmp]# cat tuwei.exp

#!/usr/bin/env expect ------解析器，使用expect解析

spawn ssh root@192.168.132.11 uptime---执行ssh命令

expect {

"yes/no" {send "yes\r";exp\_continue}

"\*password" {send "x9i86wrz\r"}

}

expect eof

执行时使用expect tuwei.exp

结果如下：

[root@salt-master tmp]# expect tuwei.exp

spawn ssh root@192.168.132.11 uptime

The authenticity of host '192.168.132.11 (192.168.132.11)' can't be established.

RSA key fingerprint is 14:7f:3a:76:0b:db:11:eb:03:62:9e:0f:f8:b8:e4:45.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.132.11' (RSA) to the list of known hosts.

root@192.168.132.11's password:

21:25:26 up 1:07, 1 user, load average: 0.00, 0.00, 0.00

# expect常用命令

spawn：通过spawn命令执行一个命令或者程序，之后所有的expect操作都会在这个执行过的命令或程序中进行，包括自动交互功能。

spawn 选项 命令或程序

如 spawn ssh [root@192.168.132.11](mailto:root@192.168.132.11) uptime

expect：获取spawn命令执行后的信息，看是否和事先定义的相匹配，一旦匹配就执行后面的动作。如：

expect "\*password" {send "123456\r"}或者放在不同行

expect "\*password"

send "123456\r"

send：expect匹配指定的字符串后，发送指定字符串给系统。

exp\_continue:表示让程序继续匹配（用于多次匹配字符串并执行不同的动作）

send\_user:打印expect脚本信息，类似shell里的echo命令，而且有echo -e的功能。

exit：退出脚本，还可以利用该命令对脚本做一些关闭前的清理和提示等工作。

如

exit -onexit {

send\_user "Good bye.\n"

}

# 变量

## 普通变量

基础语法如下：

set 变量名 变量值

如 set password "123456"

打印变量的基础语法：

puts $变量名

## 特殊参数变量

类似与shell中的$0,$1,$2,用于接收及控制expect脚本传参。

$argv 表示参数数组

[lindex $argv n]接收expect脚本传参，n从0开始。

如set file [lindex $argv 0]

set host [lindex $argv 1]

$argc表示传参的个数，$argv0 表示脚本的名字。

# if条件语句

基础语法为：

if {条件表达式} {

指令

}

或

if {条件表达式} {

指令

} else

{

指令

}

例如：

if {$argc != 2} {

send\_user "usage:expect $argv0 file host"

exit

}

# Expect中的关键字

关键字用于匹配过程，一般用于expect命令中。

eof（end of file）用于匹配结束符

timeout：控制时间的关键字变量。可以通过为该变量赋值来规定整个expect操作的时间，服务于expect全局的。

set timeout 30 -----设置30s超时

# 企业生产场景expect案例

此次准备的是3台虚拟机，IP和主机名如下

|  |  |
| --- | --- |
| IP地址 | 主机名 |
| 192.168.132.20 | salt-master |
| 192.168.132.11 | salt-minion01 |
| 192.168.132.10 | salt-minion02 |

## 批量执行命令

expect自动交互脚本test.exp

#!/usr/bin/env expect

if { $argc != 2 } {

send\_user "usage:expect $argv0 ip command"

exit

}

set host [lindex $argv 0]

set cmd [lindex $argv 1]

set password "x9i86wrz"

spawn ssh root@$host $cmd

expect {

"yes/no" {send "yes\r";exp\_continue}

"\*password" {send "$password\r"}

}

expect eof

shell循环执行expect脚本 test.sh

#!/bin/sh

if [ $# -ne 1 ];then

echo "usage:$0 cmd"

exit 1

fi

cmd=$1

cat /tmp/iplist |while read ip

do

expect test.exp $ip "$cmd"

done

查看系统磁盘使用情况 sh test.sh "df -h"

[root@salt-master tmp]# sh test.sh "df -h"

spawn ssh root@192.168.132.11 df -h

root@192.168.132.11's password:

Filesystem Size Used Avail Use% Mounted on

/dev/sda3 48G 4.0G 41G 9% /

tmpfs 285M 12K 285M 1% /dev/shm

/dev/sda1 194M 29M 155M 16% /boot

spawn ssh root@192.168.132.10 df -h

root@192.168.132.10's password:

Filesystem Size Used Avail Use% Mounted on

/dev/sda5 16G 4.7G 10G 33% /

tmpfs 334M 12K 334M 1% /dev/shm

/dev/sda1 190M 54M 127M 30% /boot

/dev/sda2 2.0G 3.2M 1.9G 1% /home

## 批量发送文件

expect自动交互脚本fenfa.exp

#!/usr/bin/env expect

if { $argc != 3 } {

send\_user "usage:expect $argv0 file ip dir"

exit

}

set file [lindex $argv 0]

set host [lindex $argv 1]

set dir [lindex $argv 2]

set password "x9i86wrz"

spawn scp -P22 -r -p $file root@$host:$dir

expect {

"yes/no" {send "yes\r";exp\_continue}

"\*password" {send "$password\r"}

}

expect eof

利用shell循环执行expect脚本

[root@salt-master tmp]# cat fenfa.sh

#!/bin/sh

. /etc/init.d/functions

if [ $# -ne 2 ];then

echo "usage:$0 file dir"

exit 1

fi

file=$1

dir=$2

cat /tmp/iplist |while read ip

do

expect fenfa.exp $file $ip $dir >/dev/null 2>&1

if [ $? -eq 0 ];then

action "$ip" /bin/true

else

action "$ip" /bin/false

fi

done

脚本中添加了function 系统函数库，是为了使用action，打印结果信息。

使脚本显得更专业。

将/etc/hosts文件批量分发到所有服务器的/tmp目录

[root@salt-master tmp]# sh fenfa.sh /etc/hosts /tmp

192.168.132.11 [ OK ]

192.168.132.10 [ OK ]

分发后查看

[root@salt-minion01 tmp]# ls -l

total 4

-rw-r--r-- 1 root root 243 Jun 21 20:48 hosts

[root@salt-minion02 tmp]# ls -l

total 4

-rw-r--r-- 1 root root 243 Jun 21 20:48 hosts

通过发送文件方式可以批量部署ssh免秘钥。