pssh,pdsh和mussh的使用

# SSH免密码登陆远程服务器

在linux系统中,ssh是远程登录的默认工具,因为该工具的协议使用了RSA/DSA的加密算法.该工具做linux系统的远程管理是非常安全的。telnet,因为其不安全性,在linux系统中被搁置使用了。

" 公私钥"认证方式简单的解释:首先在客户端上创建一对公私钥 （公钥文件：~/.ssh/id\_rsa.pub； 私钥文件：~/.ssh/id\_rsa）。然后把公钥放到服务器上（~/.ssh/authorized\_keys）, 自己保留好私钥.在使用ssh登录时,ssh程序会发送私钥去和服务器上的公钥做匹配.如果匹配成功就可以登录了。

## 确认系统已经安装了SSH

**rpm -qa | grep openssh**

**rpm -qa | grep rsync**

**//没有安装，则使用yum进程安装**

yum install ssh -->安装SSH协议

yum install rsync -->rsync是一个远程数据同步工具，可通过LAN/WAN快速同步多台主机间的文件

service sshd restart -->启动服务

## 生成秘钥对

###id\_rsa和id\_rsa.pub，默认存储在~/.ssh"目录下。

ssh-keygen -t rsa -P ''

###把id\_rsa.pub追加到授权的key里面去

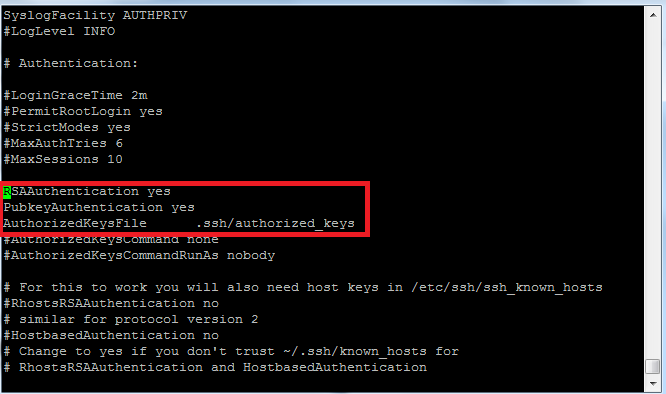
cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

###修改授权key的权限

chmod 600 ~/.ssh/authorized\_keys

###修改SSH配置文件 su - -->登陆root用户修改配置文件

vim /etc/ssh/sshd\_config -->去掉下图中三行的注释



修改本机配置: /etc/ssh/ssh\_config

# StrictHostKeyChecking ask 改成StrictHostKeyChecking no

###测试连接

service sshd restart -->重启ssh服务,

ssh localhost -->连接普通用户测试

## 配置远程服务器

###把本地的公钥传给服务器

scp ~/.ssh/id\_rsa.pub 远程用户名@远程服务器IP:~/

###追加公钥到授权key中

**cat ~/id\_rsa.pub >> ~/.ssh/authorized\_keys**

###修改授权key的权限

chmod 600 ~/.ssh/authorized\_keys

###修改SSH配置文件(可以不配置)

vim /etc/ssh/sshd\_config（同上）,service sshd restart

# ssh自动化登录

第一步：创建公钥和私钥（默认保存在~/.ssh下）

ssh-keygen -t rsa -P ''(回车)

或者ssh-keygen -t rsa -P '' -f /root/.ssh/id\_rsa

第二步：建立/etc/pssh/hosts(通讯主机，用户名，密码)

mkdir –p /etc/pssh

vi /etc/pssh/hosts

[root@vdlbnjavacoe02 test]# cat /etc/pssh/hosts

172.16.129.56 root htYun@pass

172.16.129.212 root htYun@pass

第三步：[root@vdlbnjavacoe02 test]# vi buildSSH

yum install expect -y

|  |
| --- |
| #!/bin/bash  HOST\_DIR=/etc/pssh/hosts  ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa >/dev/null  cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys  chmod 600 ~/.ssh/authorized\_keys  sed -i 's/^#RSAAuthentication/RSAAuthentication yes/' /etc/ssh/sshd\_config  sed -i 's/^#PubkeyAuthentication/PubkeyAuthentication/' /etc/ssh/sshd\_config  sed -i 's/^#AuthorizedKeysFile/AuthorizedKeysFile/' /etc/ssh/sshd\_config  for i in `awk '{print $1}' $HOST\_DIR`  do  j=`awk -v I="$i" '{if(I==$1)print $2}' $HOST\_DIR`  k=`awk -v I="$i" '{if(I==$1)print $3}' $HOST\_DIR`  expect uploadPub $i $j $k  expect editKey $i $j $k  done |

|  |
| --- |
| [root@vdlbnjavacoe02 test]# vi uploadPub  #!/usr/bin/expect -f  set ipaddress [lindex $argv 0]  set username [lindex $argv 1]  set passwd [lindex $argv 2]  if { $argc != 3 } {  puts stderr $passwd  puts stderr $username  exit 1  }  set timeout 30  spawn scp -r "/root/.ssh/id\_rsa.pub" $username@$ipaddress:~/  expect {  "yes/no" { send "yes\r";exp\_continue }  "password:" { send "$passwd\r" }  }  send "exit/r"  expect eof |

|  |
| --- |
| [root@vdlbnjavacoe02 test]# vi editKey  #!/usr/bin/expect -f  set ipaddress [lindex $argv 0]  set username [lindex $argv 1]  set passwd [lindex $argv 2]  if { $argc != 3 } {  puts stderr $passwd  puts stderr $username  exit 1  }  set timeout 30  spawn ssh $username@$ipaddress  expect {  "yes/no" { send "yes\r";exp\_continue }  "password:" { send "$passwd\r" }  }  expect "]\*"  send "cat ~/id\_rsa.pub >> ~/.ssh/authorized\_keys \r"  expect "]\*"  send "chmod 600 ~/.ssh/authorized\_keys \r"  send "exit\r "  expect eof {exit 1} |

# pssh的使用

## pssh的安装

|  |
| --- |
| tar xf pssh-2.3.1.tar.gz  cd pssh-2.3.1/  python setup.py install  常用命令   * pscp     传输文件到多个hosts，类似scp * pslurp   从多台远程机器拷贝文件到本地 * pnuke    并行在远程主机杀进程 * prsync   使用rsync协议从本地计算机同步到远程主机 |

## pssh命令的使用

|  |
| --- |
| 参数说明：  -h 执行命令的远程主机列表  或者 -H user@ip:port  文件内容格式[user@]host[:port]         -l 远程机器的用户名         -P 执行时输出执行信息         -p 一次最大允许多少连接        -o 输出内容重定向到一个文件        -e 执行错误重定向到一个文件        -t 设置命令执行的超时时间        -A 提示输入密码并且把密码传递给ssh        -O 设置ssh参数的具体配置，参照ssh\_config配置文件        -x 传递多个SSH 命令，多个命令用空格分开，用引号括起来        -X 同-x 但是一次只能传递一个命令        -i 显示标准输出和标准错误在每台host执行完毕后  常用选项：-O StrictHostKeyChecking=no //设置这个参数可以让远程主机自动接收本地主机的hostkey,而不用每次都要手动输入yes。 |

## 常见应用

批量常看远程主机信息：

pssh -i -h /etc/pssh/hosts StrictHostKeyChecking=no uptime

传输文件到多个hosts，如果是文件夹要使用-r:

pscp -h /etc/pssh/hosts /etc/ssh/ssh\_config /tmp

pscp -h /etc/pssh/hosts -r /usr/local/src /tmp

# pdsh的使用

## 安装

|  |
| --- |
| tar -jxvf pdsh-2.26.tar.bz2  yum install -y gcc cmake  cd pdsh-2.26/  ./configure --with-ssh --with-rsh --with-mrsh --with-mqshell --with-qshell --with-dshgroups --with-machines=/etc/pdsh/machines  make&& make install  测试：  [root@iZ23ef53rrcZ local]# pdsh -V  pdsh-2.26  rcmd modules: ssh,exec (default: ssh)  misc modules: (none) |

## pdsh参数说明

|  |
| --- |
| -w host,host ：指定远程主机，可以指定多个，每个主机用逗号隔开，host可以是主机名也可以是IP地址。此参数非常灵活，常用的形式有：  pdsh -w ssh:user001,ssh:user002,ssh:user003 "date"  此命令用来查看user001、user002、user003主机上的时间，其中ssh表示在  远程主机上执行命令的形式，默认是rsh。  //此命令用于在user001到user0010上执行date命令。  pdsh -w ssh:user00[1-10] "date"  //此命令在选择远程主机时使用了正则表达式，表示在user010到user031中选  // 择以1结尾的主机名，即在user011、user021、user031执行uptime命令  pdsh -w ssh:user0[10-31],/1$/ "uptime"  -R ：指定使用rcmd的模块名，默认是rsh。如果要选择ssh，可以通过如下方式指 定：pdsh -R ssh -w user00[1-10] "date"  -l： 指定在远程主机上使用的用户名称。例如：  pdsh -R ssh -l opsuser -w user00[1-9] "date"  -x ：此参数用来排除某些或某个主机，例如：  pdsh -R ssh -l opsuser -w user00[1-9] -x user005,user007 "date"    -t ：指定连接远程主机的超时时间，以秒为单位，默认是10秒，可以通过此参数修改默认值，例如：pdsh -R ssh -w slave000[1-9] -t 15 "date"  -u ：设置远程命令执行的超时时间，以秒为单位，以ssh方式连接时，默认时间为无限。 -f ：设置同时连接到远程主机的个数  -N ：此参数用来关闭远程主机所返回结果中的主机名显示  -a ：通过此参数可以指定所有的远程主机，设置此参数后，pdsh默认会查看  /etc/machines文件中的主机列表，要改变此路径，在编译pdsh时通过  “-- with-machines”参数指定即可  -g ：此参数用来指定一组远程主机，在编译pdsh时可以通过  “--with-dshgroups”参数来激活此选项，默认可以将一组主机列表写入一个文件  中并放到本地主机的~/.dsh/group或/etc/dsh/group目录下，这样就可以通过“-g”参数调用了。例如：pdsh -R ssh -g userhosts "date"，其中“userhosts”是一个主机列表文件，可以将此文件放在~/.dsh/group或  /etc/dsh/group目录下  -X ：此参数用来排除指定组内的所有主机，经常与“-a”参数一起使用。  例如：pdsh -R ssh -a -X userhosts "date"  -q ：此参数可以列出pdsh执行时的一些配置信息  -V ：此参数可以查看软件的版本信息以及可用的模块信息 |

## pdsh常见使用

-w支持正则表达式，通过正则表达式来过滤。

远程主机名字不是规范的，可以将所有的主机名写到一个文件中，然后通过-a参数调用。

大规模运维的时候，我们通常对服务器进行分组，将同一个组的服务器写个一个文件中，并把文件放在etc/dsh/group或者~/.dsh/group目录下，通过-g来调用。--dshgroup激活安装。

-a –X：可以排除组内所有的主机

pdsh还支持交互,不指定命令的时候，pdsh会进入交互状态。

# 多主机ssh封装器mussh的使用

## 安装

yum install –y mussh

## 常用使用

|  |
| --- |
| 参数说明  -m：启动mussh的并发执行，管理多个主机的时候使用。  -o：引入ssh参数的具体配置，-o StrictHostKeyChecking=no  -h：指定远程主机，多个主机之间用空格隔开  -H：指定一组主机的列表文件，管理多个主机的时候使用  -c：指定要在远程主机上执行的命令，多个命令用分号隔开  -C：参数用来指定本地脚本文件  -u：参数可以用来除去重复的主机名（默认）  -U：参数不管是否存在重复的主机名  -s：指定要在远程主机上执行的shell路径  -l：指定在远程主机上执行命令的用户 |

例如：

mussh -H /etc/pssh/hosts -m -c uptime

注意：要在mussh中使用绝对路径。另外我们可以使用alias 写入~/.bashrc