* 1. paramiko的安装
  2. SSHclient方法
  3. sftpclient方法
  4. sftpclient应用实例

【考核要求】

* + - 1. paramiko的安装

1、安装pyCrypto  
安装这个比较麻烦，需要本地编译，要装vs或gcc还有一堆配置，还不一定能编译成功。（网上能搜到安装步骤)  
建议直接下载已编译版： http://www.voidspace.org.uk/python/modules.shtml#pycrypto  
直接下载并安装既可。  
（注：crypto有java和C++版）  
2、安装ecdsa  
看很多博客没有提到这个库，但我执行paramiko时，提示找不到ecdsa模块。  
下载：https://pypi.python.org/pypi/ecdsa/0.9 ,解压到一个目录，目录中有一个setup.py。  
安装比较简单,windows下直接在刚才解压后的目录执行：python setup.py install

3.安装paramiko

与安装ecdsa类型，只是打开下载页面很慢。。。

下载： https://github.com/paramiko/paramiko#，

安装步骤同ecdsa

* 安装paramiko有两个先决条件，

python和另外一个名为PyCrypto的模块。

　　通常安装标准的python模块，只需要在模块的根目录下运行：

python setup.py build

python setup.py install

备注：安装前先检查是否安装gcc（yum -y install gcc）

* PyCrypto安装

wget http://ftp.dlitz.net/pub/dlitz/crypto/pycrypto/pycrypto-2.6.tar.gz

tar -zxvf pycrypto-2.6.tar.gz

cd pycrypto-2.6/

python setup.py build && python setup.py install

　　测试：

python>> import Crypto

　　(编译时报错：error: command 'gcc' failed with exit status 1；这是因为缺少python-dev的软件包，所yum -y install python-devel)

* paramiko安装

wget http://www.lag.net/paramiko/download/paramiko-1.7.7.1.tar.gz

tar xvzf paramiko-1.7.7.1.tar.gz

cd paramiko-1.7.7.1/

python setup.py build && python setup.py install

Crypto error: 'module' object has no attribute 'HAVE\_DECL\_MPZ\_POWM\_SEC'

* 测试：

python>> import paramiko

使linux支持拖放文件rpm -ivh lrzsz-0.12.20-27.1.el6.x86\_64.rpm

测试代码：

#!/usr/bin/python

# -\*- coding:utf-8 -\*-

# cp@chenpeng.info

import paramiko

def MAIN():

host = “10.1.1.1”

port = 22

user = “root”

pswd = “111222333”

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(host, port, user, pswd)

stdin, stdout, stderr = ssh.exec\_command(‘ifconfig’)

print stdout.read()

ssh.close()

#

if \_\_name\_\_==’\_\_main\_\_':

try:

MAIN()

except Exception,e:

print e

#

输出如下：

image

目前主要用于批量执行多个服务器的ssh命令，批量维护比较方便。

下面是两种使用paramiko连接到linux服务器的代码

方式一：

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect("某IP地址",22,"用户名", "口令")

上面的第二行代码的作用是允许连接不在know\_hosts文件中的主机

方式二：

t = paramiko.Transport(("主机","端口"))

t.connect(username = "用户名", password = "口令")

如果连接远程主机需要提供密钥，上面第二行代码可改成：

t.connect(username = "用户名", password = "口令", hostkey="密钥")

例子：

**使用Key 连接远程**

**import** paramiko  
ssh = paramiko.SSHClient()  
ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())  
pkey\_file = **'/root/.ssh/id\_rsa'**key = paramiko.RSAKey.from\_private\_key\_file(pkey\_file)  
ssh.connect(**"10.1.1.137"**,22,**"root"**,pkey=key,timeout=5)  
stdin,stdout,stderr = ssh.exec\_command(**"ls"**)  
**print** stdout.read()

#!/usr/bin/python

import paramiko

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect("某IP地址",22,"用户名", "口令")

stdin, stdout, stderr = ssh.exec\_command("你的命令")print stdout.readlines()

ssh.close()

SSH免密码登陆远程服务器

背景：搭建环境需要设置无密码登陆，所谓无密码登陆其实是指通过证书认证的方式登陆，使用一种被称为"公私钥"认证的方式来进行ssh登录。

 　　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。

工具/原料

* linux系统

方法/步骤

1. 确认系统已经安装了SSH。

**rpm –qa | grep openssh**

**rpm –qa | grep rsync**

     -->出现如下图的信息表示已安装

假设没有安装ssh和rsync，可以通过下面命令进行安装。

**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

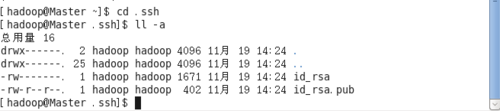
**service sshd restart** -->启动服务

[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=1)

1. 生成秘钥对

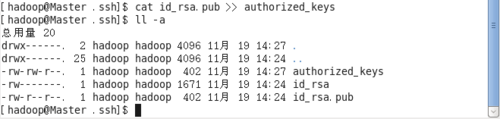
**ssh-keygen –t rsa –P ''** -->直接回车生成的密钥对：id\_rsa和id\_rsa.pub，默认存储在"/home/hadoop/.ssh"目录下。

[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=2)

[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=3)

1. 把id\_rsa.pub追加到授权的key里面去。

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

[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=4)

1. 修改授权key的权限

**chmod 600 ~/.ssh/authorized\_keys**

[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=5)

1. 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

[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=6)

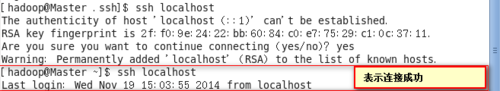
1. 测试连接

**service sshd restart** -->重启ssh服务,

**exit** -->退出root用户，回到普通用户

**ssh localhost** -->连接普通用户测试

这只是配置好了单机环境上的SSH服务，要远程连接其它的服务器，接着看下面。

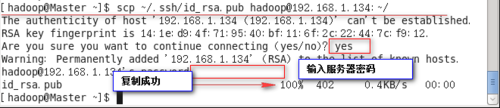
[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=7)

1. 现在秘钥对已经生成好了，客户端SSH服务也已经配置好了，现在就把我们的钥匙（公钥）送给服务器。

**scp ~/.ssh/id\_rsa.pub 远程用户名@远程服务器IP:~/**  -->将公钥复制到远程服务器的~/目录下

如: scp ~/.ssh/id\_rsa.pub hadoop@192.168.1.134:~/

可以看到我们复制的时候需要我们输入服务器的密码，等我们把SSH配置好之后这些步骤就可以不用输入密码了。

[](http://jingyan.baidu.com/album/2fb0ba4043124a00f2ec5f0f.html?picindex=8)

1. 上 一步把公钥发送到192.168.1.134服务器上去了，我们去134机器上把公钥追加到授权key中去。（注意：如果是第一次运行SSH，那 么.ssh目录需要手动创建，或者使用命令ssh-keygen -t rsa生成秘钥，它会自动在用户目录下生成.ssh目录。特别注意的是.ssh目录的权限问题，记得运行下chmod 700 .ssh命令）

在134机器上使用命令：

**cat ~/id\_rsa.pub >> ~/.ssh/authorized\_keys**  -->追加公钥到授权key中

**rm ~/id\_rsa.pub**  -->保险起见，删除公钥

同样在134机器上重复第四步和第五步，

**service sshd restart** -->重启ssh服务

1. 回到客户机来，输入:

**ssh 192.168.1.134   -->**应该就能直接连接服务器咯。

下载linux服务器上的文件

#!/usr/bin/python

import paramiko

t = paramiko.Transport(("主机","端口"))

t.connect(username = "用户名", password = "口令")

sftp = paramiko.SFTPClient.from\_transport(t)

remotepath=’/var/log/system.log’

localpath=’/tmp/system.log’

sftp.get(remotepath, localpath)

t.close()

上传文件到linux服务器

#!/usr/bin/python

import paramiko

t = paramiko.Transport(("主机","端口"))

t.connect(username = "用户名", password = "口令")

sftp = paramiko.SFTPClient.from\_transport(t)

remotepath=’/var/log/system.log’

localpath=’/tmp/system.log’

sftp.put(localpath,remotepath)

t.close()