paramiko建立SSH连接

paramiko是基于Python实现的SSHv2远程安全连接, 它支持认证及密钥的方式, 可以实现远程命令执行 文件传输 中间SSH代理功能.

安装方法:

pip install paramiko --trusted-host pypi.org 或 pip install paramiko -i http://pypi.douban.com/simple (http://pypi.douban.com/simple) --trusted-host pypi.douban.com ##使用豆瓣源来安装

一.SSHClient类

ssh = paramiko.SSHClient()

常用方法:

1.connect #连接ssh服务器

connect(self, hostname, port=22, username=None, password=None, pkey=None, key_filename=None, timeout=None, allow_agent=True, look_for_keys=True, compress=False, sock=None, gss_auth=False, gss_kex=False, gss_deleg_creds=True, gss_host=None, banner_timeout=None, auth_timeout=None, gss_trust_dns=True, passphrase=None)

2.exec_command #在远程服务器上执行系统命令

| exec_command(self, command, bufsize=-1, timeout=None, get_pty=False, environment=None)

| Execute a command on the SSH server. A new `.Channel` is opened and

| the requested command is executed. The command's input and output

| streams are returned as Python `file` -like objects representing

| stdin, stdout, and stderr.

|
| :param str command: the command to execute

| :param int bufsize:

| interpreted the same way as by the built-in `file()` function in

| Python

| :param int timeout:

| set command's channel timeout. See `.Channel.settimeout`

| :param dict environment:

| a dict of shell environment variables, to be merged into the

| default environment that the remote command executes within.

|

| .. warning::

| Servers may silently reject some environment variables; see the

| warning in `.Channel.set_environment_variable` for details.

|

| :return:

| the stdin, stdout, and stderr of the executing command, as a

| 3-tuple

二.SFTPClient类

作为SFTP客户端对象, 根据SSH传输协议的sftp会话, 实现远程文件操作; 如:文件上传 下载 权限 状态 等操作

1. t = paramiko.Transport(('172.16.70.81',22)) #传入一个socket

In [3]: help(paramiko.Transport)

Help on class Transport in module paramiko.transport:

class Transport(threading.Thread, paramiko.util.ClosingContextManager)

```
    __init__(self, sock, default_window_size=2097152, default_max_packet_size=32768, gss_kex=False, gss_deleg_creds=True)
     |      Create a new SSH session over an existing socket, or socket-like

     |      object.  This only creates the `.Transport` object; it doesn't begin

     |      the SSH session yet.  Use `connect` or `start_client` to begin a client

     |      session, or `start_server` to begin a server session.
```

2. t.connect(username='root', password='meizu.com') #输入帐号密码连接
3. sftp = paramiko.SFTPClient.from_transport(t) #创建sftp

GET

In [5]: help(sftp.get) Help on method get in module paramiko.sftp_client:

get(self, remotepath, localpath, callback=None) method of paramiko.sftp_client.SFTPClient instance Copy a remote file (`remotepath`) from the SFTP server to the local host as `localpath` . Any exception raised by operations will be passed through. This method is primarily provided as a convenience.

```
    :param str remotepath: the remote file to copy
    :param str localpath: the destination path on the local host
    :param callable callback:
        optional callback function (form: ``func(int, int)``) that accepts
        the bytes transferred so far and the total bytes to be transferred

    .. versionadded:: 1.4
    .. versionchanged:: 1.7.4
        Added the ``callback`` param
```

(END)

PUT

In [6]: help(sftp.put) Help on method put in module paramiko.sftp_client:

put(self, localpath, remotepath, callback=None, confirm=True) method of paramiko.sftp_client.SFTPClient instance Copy a local file (`localpath`) to the SFTP server as `remotepath` . Any exception raised by operations will be passed through. This method is primarily provided as a convenience.

```
    The SFTP operations use pipelining for speed.

    :param str localpath: the local file to copy
    :param str remotepath: the destination path on the SFTP server. Note
        that the filename should be included. Only specifying a directory
        may result in an error.
    :param callable callback:
        optional callback function (form: ``func(int, int)``) that accepts
        the bytes transferred so far and the total bytes to be transferred
    :param bool confirm:
        whether to do a stat() on the file afterwards to confirm the file
        size

    :return: an `.SFTPAttributes` object containing attributes about the
        given file

    .. versionadded:: 1.4
    .. versionchanged:: 1.7.4
        ``callback`` and rich attribute return value added.
    .. versionchanged:: 1.7.7
        ``confirm`` param added.
```

(END)

In [3]:

```
#例1: 创建SSH连接并在远程服务器上执行系统命令

import paramiko

ssh = paramiko.SSHClient()
ssh.load_system_host_keys()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy()) ##允许不在knows_host的主机可以连接 常用
ssh.connect('172.16.70.203',username='root', password='meizu.com')
stdin, stdout, stderr  = ssh.exec_command('ls -la /app_shell')
##返回一个元组 stdin stdout strerr
print stdout.read()  ##返回命令输出
print stderr.read()  ##命令如果执行未出错, 返回空
```

```
total 29828
drwxr-xr-x  11 root root      4096 Oct 17 11:46 .
dr-xr-xr-x. 20 root root      4096 Aug 12 11:08 ..
drwxr-xr-x   3 root root        66 Oct 10 22:34 blog
drwxrwxr-x  11 1000  1000     4096 Oct 10 22:07 Django-1.9.13
-rwxr-xr-x   1 root root   7498364 Sep 16 11:34 Django-1.9.13.tar.gz
drwxr-xr-x   9 root root      4096 Oct 17 18:37 jupyter
drwxr-xr-x   8 root root       108 Sep  9 11:51 PA_cloud
drwxr-xr-x  10 root root       133 Oct 17 11:46 project1
-rw-r--r--   1 root root  21102229 Jul 16 23:37 project1.tar.gz
drwxr-xr-x   2 root root         6 Jul 16 23:30 python
drwxr-xr-x  10  501 games     4096 Oct 10 22:02 virtualenv-15.1.0
-rwxr-xr-x   1 root root   1863951 Sep 16 11:34 virtualenv-15.1.0.tar.gz
drwxr-xr-x   5 root root        77 Oct 10 22:03 web
drwxr-xr-x   7 root root       103 Jul 14 10:50 webapps
-rw-r--r--   1 root root     49558 Jul 16 09:20 webapps.tar.gz
```

In [23]:

```
#例2. 创建sftp通道, 将本地文件传输到远程服务器

t = paramiko.Transport(('172.16.70.180',22))
t.connect(username='root', password='meizu.com')
sftp = paramiko.SFTPClient.from_transport(t)

sftp.put('/app_shell/jupyter/Python自动化运维/demo1.py', '/tmp/demo1.py') #将本地路径下文件推送到远程服务器
sftp.get('/tmp/demo1.py', '/app_shell/demo1.py') #将远程服务器上的指定文件拉取到本地目录
```

In [26]:

```
#示例代码1：连接Centos服务器

#!/usr/bin/python
#coding:utf8
import os
import sys
import commands
import paramiko


class ConnectError(Exception):
    pass

class controlHost:

    def __init__(self, ip_address, username, password, port, key_file='/root/.ssh/id_rsa'):

        self.ip_address = ip_address
        self.pkey = paramiko.RSAKey.from_private_key_file(key_file)
        self.ssh = controlHost.__sshConn(ip_address, username, password, self.pkey, port)
        self.sftp = controlHost.__sftpConn(ip_address, username, password, self.pkey, port)

    @staticmethod
    def __sshConn(ip_address, username, password, pkey, port):
        ssh = paramiko.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        try:
            ssh.connect(hostname=ip_address, port=port, username=username, pkey=pkey)
        except:
            try:
                ssh.connect(hostname=ip_address, port=port, username=username, password=password)
            except Exception as e:
                raise ConnectError("paramiko ssh connect %s error! %s" %(ip_address, e))
            else:
                return ssh
        else:
            return ssh

    @staticmethod
    def __sftpConn(ip_address, username, password, pkey, port):
        try:
            sftp_conn = paramiko.Transport((ip_address, port))
            sftp_conn.connect(username=username, pkey=pkey)
        except:
            try:
                sftp_conn = paramiko.Transport((ip_address, port))
                sftp_conn.connect(username=username, password=password)
            except Exception as e:
                raise ConnectError("paramiko sftp connect %s error! %s" % (ip_address, e))
            else:
                sftp = paramiko.SFTPClient.from_transport(sftp_conn)
                return sftp
        else:
            sftp = paramiko.SFTPClient.from_transport(sftp_conn)
            return sftp

    def exeCommand(self, cmd, timeout=300):
        try:
            _, stdout, stderr = self.ssh.exec_command(cmd, timeout=timeout)
            channel = stdout.channel
            exit_code = channel.recv_exit_status()
            return {"status": 1, "stdout": stdout.read(), "stderr": stderr.read(), 'exit_code': exit_code}
        except Exception as e:
            return {"status": 0, "stdout": stdout.read() + e, "stderr": stderr.read(), 'exit_code': exit_code}

    def sftpFile(self, localpath, remotepath, action):
        try:
            if action == 'push':
                dirname = os.path.dirname(remotepath)
                self.exeCommand("mkdir -p %s" %dirname)
                self.sftp.put(localpath, remotepath)
                return {"status": 1, "stdout": 'sftp %s %s success!'%(self.host, action), 'stderr': ''}
            elif action == "pull":
                dirname = os.path.dirname(localpath)
                if not os.path.exists(dirname):
                    os.makedirs(dirname)
                if os.path.exists(localpath):
                    os.remove(localpath)
                self.sftp.get(remotepath, localpath)
                return {"status": 1, "stdout": 'sftp %s %s success!'%(self.host, action), "stderr": ""}
        except Exception as e:
            return {"status": 0, "stderr": 'sftp %s %s failed %s'%(self.host, action, e), "stdout": ""}

if __name__  == '__main__':
    import time
    beg = time.time()
    ssh = controlHost('172.16.70.180', 'root', 'meizu.com', 22)
    print ssh.exeCommand('ls /tmp')
    print ssh.sftpFile("/tmp/abc123.log", '/tmp/cc/dd/123123.log', 'put')
    print ssh.sftpFile('/tmp/abc/bcd/abc123.log', '/tmp/c/d/123123.log', 'get')
```

```
{'status': 1, 'stderr': '', 'stdout': '172.16.70.180_full\ndemo1.py\ngitaly-ruby006448575\ngitaly-ruby131046893\ngitaly-ruby189560474\ngitaly-ruby383912889\ngitaly-r
uby856919948\ngitaly-ruby980979619\ngnome-system-monitor.root.3690914916\nkeyring-BYmRj4\nkeyring-EjeBBD\nkeyring-IEivPz\nkeyring-LAZkDM\nkeyring-XqGuhi\nkeyring-Ykq
44V\nnotebook-5.6.0.tar.gz\norbit-gdm\norbit-root\npercona-version-check\npip-ephem-wheel-cache-WyJmFA\npip-install-14R1Zi\npip-req-build-62hN5w\npip-req-tracker-fYZ
QoI\npip-unpack-MSUgFK\npulse-5A7k17Hqntw3\npulse-x1nvveXaJ7YU\ntest\ntimer_create3kPsnQ.c\ntimer_createQxaxjH.c\ntimer_createVA49BM.c\ntools1536769948\ntools1536769
949\nvirtual-root.dzLoTA\n'}
```

In [1]:

```
#示例代码2：连接思科网络设备(Python3版)

#coding:utf8
import paramiko
import os
import time
import asyncio

class ConnectError(Exception):
    pass


class sshCisco:

    def __init__(self, host, username, password, pripwd, port):
        self.host = host
        self.conn = sshCisco.__sshConn(host, username, password, pripwd, port)

    @staticmethod
    def __sshConn(host, username, password, pripwd, port):
        ssh = paramiko.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        try:
            ssh.connect(host, port, username, password, look_for_keys=False)
        except Exception as e:
            raise ConnectError("paramiko ssh connect %s error! %s" % (host, e))
        else:
            ssh_shell = ssh.invoke_shell()  # 用户模式
            ssh_shell.send(bytes('enable\n', encoding='utf8'))
            time.sleep(1)
            ssh_shell.send(bytes(pripwd + '\n', encoding='utf8'))  # 进入特权模式
            time.sleep(1)
            line = str(ssh_shell.recv(1024), encoding='utf8')
            if line.endswith("#"):
                return ssh_shell
            else:
                raise ConnectError("paramiko login cisco privilege level error! %s" % host)


    def executeCmd(self, cmd):
        self.conn.send(bytes(cmd + '\n', encoding='utf8'))
        time.sleep(2)
        result = str(self.conn.recv(65535), encoding='utf8')
        return result

    def close(self):
        print('%s Connection interrupt!!' %self.host)
        self.conn.close()


if __name__ == '__main__':
    s = [("172.16.70.121", "ccna", "ccna", "ccna","22"), ("172.16.70.122", "ccna", "ccna", "ccna","22")]
    for j in s:
        x = sshCisco(*j)
        x.executeCmd('terminal length 0')
        print(x.executeCmd("show run"))
        x.close()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-412de2ecc102> in <module>
      2 
      3 #coding:utf8
----> 4 import paramiko
      5 import os
      6 import time

ModuleNotFoundError: No module named 'paramiko'
```

In [ ]: