

ssh hangs when command invoked directly, but exits cleanly when run interactive

Asked 16 years, 3 months ago Modified 11 years, 7 months ago Viewed 4k times



2



I need to launch a server on the remote machine and retrieve the port number that the server process is listening on. When invoked, the server will listen on a random port and output the port number on stderr.

I want to automate the process of logging on to the remote machine, launching the process, and retrieving the port number. I wrote a Python script called "`invokejob.py`" that lives on the remote machine to act as a wrapper that invokes the job and then returns the port number, it looks like this:

```
import re, subprocess
executable = ... # Name of executable
regex = ... # Regex to extract the port number from the output
p = subprocess.Popen(executable,
    bufsize=1, # line buffered
    stderr=subprocess.PIPE
)
s = p.stderr.readline()
port = re.match(regex).groups()[0]
print port
```

If I log in interactively, this script works:

```
$ ssh remotehost.example.com
Last login: Thu Aug 28 17:31:18 2008 from localhost
$ ./invokejob.py
63409
$ exit
logout
Connection to remotehost.example.com closed.
```

(Note: successful logout, it did not hang).

However, if I try to invoke it from the command-line, it just hangs:

```
$ ssh remotehost.example.com invokejob.py
```

Does anybody know why it hangs in the second case, and what I can do to avoid this?

Note that I need to retrieve the output of the program, so I can't just use the ssh "-f" flag or redirect standard output.

python ssh

Share Improve this question Follow

asked Aug 28, 2008 at 21:48



Lorin Hochstein

59k 31 109 144

2 Answers

Sorted by: Highest score (default)



3

```
s = p.stderr.readline()
```



I suspect it's the above line. When you invoke a command directly through ssh, you don't get your full pty (assuming Linux), and thus no stderr to read from.



When you log in interactively, stdin, stdout, and stderr are set up for you, and so your script works.



Share Improve this answer Follow

answered Aug 28, 2008 at 21:53



Ben Collins

20.7k 18 128 190

it is possible to get full pty when running a command through SSH, or is interacting with stdin on a running process through SSH impossible? – [cdated](#) Jun 11, 2010 at 14:49

`stderr=subprocess.PIPE` makes the subprocess's stderr a pipe, not a pty. So it should make no difference. But it can be a difference what the subprocess expects as stdin and stdout. – [glglgl](#) May 27, 2013 at 7:46



0

what if you do the following:

```
ssh <remote host> '<your command> ;<your regexp using awk or something>'
```



For example



```
ssh <remote host> '<your program>; ps aux | awk \'/root/ {print $2}\''
```

This will connect to , execute and then print each PSID for any user root or any process with root in its description.

I have used this method for running all kinds of commands on remote machines. The catch is to wrap the command(s) you wish to execute in single quotation marks (') and to separate each command with a semi-colon (;).

Share

Improve this answer

Follow

edited May 27, 2013 at 7:43



g!g!g!

90.9k ● 13 ● 154 ● 228

answered Aug 28, 2008 at 23:50



Misha M

11.3k ● 17 ● 55 ● 67
