How to SSH client Login to server with password provided from command line as a script argument - Running same commands to many Linux servers

Author : admin

Usually admins like me who casuanlly need to administer "forests" (thousands of identicallyconfigured services Linux servers) are generating and using RSA / DSA key authentication for passwordless login, however this is not always possible as ***some client environments does prohibit the use of RSA / DSA non-pass authentication, thus in such environments to make routine server basic package rpm / deb upgrades or do other maintanance patching its necessery to use normal ssh user / pass login but as ssh client doesn't allow password to be provided from prompt for security*** reasons and therefore using some custom bash loop to issue single command to many servers (such as explained in my previous article) *requires you to copy / paste password on password prompt multiple times.* This works its pretty annoying so if you want to run single command on all your 500 servers with specifying the password from password prompt use sshpass tool (**for non-interactive ssh password auth**).

SSHPASS *official site description*:

> ***sshpass is a utility designed for running ssh using the mode referred to as "keyboard-interactive" password authentication, but in non-interactive mode.***

Install sshpass on Debian / Ubuntu (deb based) Linux

sshpass is installable right out of regular repositories so to install run:

```
apt-get install ---yes sshpass
```

Install sshpass on CentOS / Fedora (RPM based) Linux

sshpass is available also across most RPM based distros too so just use **yum package manager**

```
yum -y install sshpass
```

If its not available across standard RPM distro provided repositories, there should be RPM on the net for distro just download latest one and *use wget and rpm to install*:

wget -q http://dl.fedoraproject.org/pub/epel/6/x86_64/sshpass-1.05-1.el6.x86_64.rpm

rpm -ivh sshpass-1.05-1.el6.x86_64.rpm

How Does SshPass Works?

Normally **openssh (ssh)** client binary uses direct ***TTY (/dev/tty)= an abbreviation***

for PhyTeleTYpewriter or (the admin jargon call *Physical Console access*) instead of standard remotely

defined ***/dev/pts*** = *Virtual PTY.*

To get around this **Sshpass** runs ssh in a *dedicated TTY* to *emulate the password is indeed issues by*

interactive keyboard user thus fooling remote sshd server to thinking password

is provided by interactive user.

SSHPass use

Very basic standard use which allows you to pass the password from command line is like this:

```
sshpass -p 'Your_Password_Goes_here123' ssh username@server.your-server.com
```

Note that the server you're working is shared with other developers they might be able to steal your username / password by using a simple process list command such as:

```
ps auxwwef
...
```

In my case security is not a hot issue, as I'm the only user on the server (and only concern might be if *someone hacks into the server* :)

Then assuming that you have a plain text file with all your administered servers, you can easily use sshpass in a Bash Script loop in order to run, lets say a package upgrade across all identical Linux version machines:

```
while read line; do
sshpass -p 'Your_Password_Goes_here123' ssh username@$line "apt-get update && apt-get upgrade && apt-get dist-upgrade"  done
```

Change the command you like to issue across all machines with the string **"apt-get ..."**
Above command can be used to keep up2date all Debian stable server packages. What you will do on servers is up to your imaginations, very common use of above line would be if you want to see **uptime /netstat** command output across all your network servers.

```
while read line; do
sshpass -p 'Your_Password_Goes_here123' ssh username@$line "uptime; who; netstat -tunlp; "  done
```

As you can guess SshPass is *swiss army knife tool for admins who***need to automate things with scripts simultaneously across number of servers.**

Happy SSH-ing :)