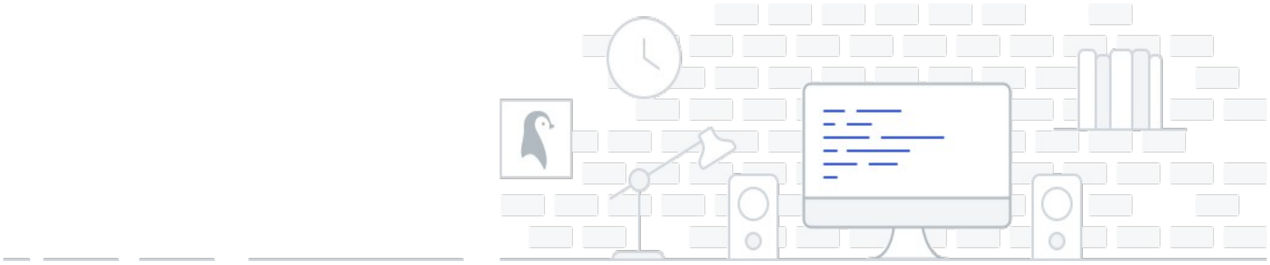
nixCraft

Linux and Unix tutorials for new and seasoned sysadmin

https://www.cyberciti.biz/faq/linux-unix-ssh-proxycommand-passing-through-one-host-gateway-server/



SSH ProxyCommand example: Going through one host to reach another server

How do I use and jump through one server to reach another using ssh on a Linux or Unix-like systems? Is it possible to connect to another host via an intermediary so that the client can act as if the connection were direct using ssh? Can you give me SSH ProxyCommand and ProxyJump example?

You can jump host using ProxyCommand. [donotprint][/donotprint] Some times you can only access a remote server via ssh by first login into an intermediary server (or firewall/jump host). So you first login into to the intermediary server and then ssh to another server. You need to authenticate twice and the chain can be long and is not limited to just two hosts. This page provide SSH ProxyCommand example

Our SSH ProxyCommand example setup

```
+-------+       +----------+      +-----------+
| Laptop| <---> | Jumphost | <--> | FooServer |
+-------+       +----------+      +-----------+


          OR

+-------+       +----------+      +-----------+
| Laptop| <---> | Firewall | <--> | FooServer |
+-------+       +----------+      +-----------+
   192.168.1.5      121.1.2.3         10.10.29.68
```

I can can only access a remote server named ‘FooServer’ via ssh by first login into an intermediary server called ‘Jumphost’. First, login to Jumphost:

```
$ ssh vivek@Jumphost
```

Next, I must ssh through the intermediary system as follows:

```
$ ssh vivek@FooServer
```

Passing through a gateway or two

I can connect to the target host named www.nixcraft.com by first making a ssh connection to the jump host called hello.vpn.cyberciti.biz and then establishing a TCP forwarding to the ultimate destination from there:

```
ssh -J Jumphost FooServer
ssh -J Jumphost vivek@FooServer
ssh -J hello.vpn.cyberciti.biz www.nixcraft.com
ssh -J hello.vpn.cyberciti.biz:22 vivek@www.nixcraft.com
```

Command to use when the `-J` option not available

In older versions of openssh the `-J` is not available. So use the following syntax:

```
$ ssh -o ProxyCommand="ssh -W %h:%p Jumphost" FooServer
$ ssh -o ProxyCommand="ssh -W %h:%p vivek@hello.vpn.cyberciti.biz"
vivek@www.nixcraft.com

$ ssh -o ProxyCommand="ssh -W %h:%p vivek@hello.vpn.cyberciti.biz"
root@wp-admin-server
```

The oldest clients don’t support the `-W` option

In this case the `ssh -tt` command. Instead of typing two ssh command, I can type the following all-in-one command. This is useful for connecting to FooServer via firewall called ‘Jumphost’ as the jump host:

```
$ ssh -tt Jumphost ssh -tt FooServer
$ ssh -tt vivek@Jumphost ssh -tt vivek@FooServer
$ ssh -tt vivek@Jumphost ssh -tt vivek@FooServer command1 arg1 arg2
$ ssh -tt vivek@Jumphost ssh -tt vivek@FooServer htop
$ ssh -tt vivek@Jumphost ssh -tt vivek@FooServer screen -dR
```

Where,

- The `-t` option passed to the ssh command force pseudo-tty allocation. This can be used to execute arbitrary screen-based programs on a remote machine. Multiple `-tt` options force tty allocation, even if ssh has no local tty.

How to pass through a gateway using stdio forwarding

The syntax is simple as explained above. You no **longer** need nc installed due `ProxyCommand` syntax:

```
ssh -o ProxyCommand="ssh -W %h:%p jumphost.nixcraft.com"
server1.cyberciti.biz
```

Say the user accounts names are different on the two Unix or Linux server. Here, ‘tom’ is the account on the second machine which is the final target. The user ‘jerry’ is the account on the intermediary or jump host:

```
ssh -l tom \
      -o 'ProxyCommand ssh -l jerry %h nc server2.nixcraft.com 22' \
      -o 'HostKeyAlias server2.nixcraft.com' \
      server1.cyberciti.biz
```

An updated version of my ~/.ssh/config file:

```
Host webserver
        Hostname www42.cyberciti.biz
        ProxyCommand ssh jumphost.nixcraft.com -W %h:%p

Host mysftpserver
        HostName sftpserver.cyberciti.biz
        HostKeyAlias sftpserver.cyberciti.biz
        ProxyCommand ssh jumphost.nixcraft.com -W %h:%p
```

Now all I have to do is type the following ssh command or sftp command:

```
ssh webserver
```

How to pass through One or more gateways/firewall using ProxyJump

OpenSSH version 7.3 or above includes simple syntax for ~/.ssh/config file:

```
Host forum
        HostName www.nixcraft.com
        ProxyJump vivek@jumhost.nixcraft.com:22
        User vivek
```

One can set multiple jump host using a comma-separated list and the servers will be visited in the order listed:

```
Host www-admin-box
        HostName www.cyberciti.biz
        ProxyJump tom@jumphost1.cyberciti.biz:22,jerry@jumphost2.cyberciti.biz:42
        User vivek
```

How do I recursively chain gateways using stdio forwarding?

Try the following `ProxyJump` syntax hich is available starting with OpenSSH version 7.3 or above in your ~/.ssh/config file:

```
Host nixcraftserver1
        Hostname hello.vpn.cyberciti.biz
        User vivek
        IdentityFile /home/vivek/.ssh/nixcraftserver1_e25519
        Port 22

Host nixcraftserver2
        Hostname 192.168.2.25
        User vivek
        IdentityFile /home/vivek/.ssh/nixcraftserver2_e25519
        Port 22
        ProxyCommand ssh -W %h:%p nixcraftserver1

Host nixcraftserver3
        Hostname 10.8.0.5
        User fred
        IdentityFile /home/vivek/.ssh/nixcraftserver3_e25519
        Port 22
        ProxyCommand ssh -W %h:%p nixcraftserver2
```

Say hello to the ProxyCommand with netcat (older method)

The syntax is as follows and works with all clients

```
$ ssh -o ProxyCommand='ssh firewall nc remote_server1 22'
remote_server1
$ ssh -o ProxyCommand='ssh vivek@Jumphost nc FooServer 22'
vivek@FooServer
##########################################
## -t option is needed to run commands ###
##########################################
$ ssh -t -o ProxyCommand='ssh vivek@Jumphost nc FooServer 22'
vivek@FooServer htop
```

The netcat (nc) command is needed to set and establish a TCP pipe between Jumphost (or firewall) and FooServer. Now, my laptop (local system) is connected to Jumphost it now connected FooServer. In this example, the utility netcat (nc) is for reading and writing network connections directly. It can be used to pass connections to a 2nd server such as FooServer.

Update ~/.ssh/config file (older method with netcat [nc])

Edit the $HOME/.ssh/config file using a text editor such as vi, enter:

```
$ vi ~/.ssh/config
```

Append the following configuration:

```
Host fooserver
HostName FooServer
User vivek
ProxyCommand ssh vivek@Jumphost nc %h %p
```

Save and close the file. Where,

1. **`Host fooserver`** : Set nickname of your choice.
2. **`HostName FooServer`** : Set the real remote server/host name.
3. **`User vivek`** : Set the real user name for remote server/host.
4. **`ProxyCommand ssh vivek@Jumphost nc %h %p`** : Specifies the command to use to connect to the server. In this example, I'm using nc command. Any occurrence of `%h` will be substituted by the host name to connect, `%p` by the port, and `%r` by the remote user name.

To test enter:

```
$ ssh fooserver
```

To see the details, pass the `-v` option to the ssh command. Here is another snippet:

```
Host server1
HostName v.server1
User root
Port 22
ProxyCommand ssh root@v.backup2 nc %h %p %r
```

Now, run:

```
$ ssh -v server1
```

Sample outputs:

```
OpenSSH_6.2p2, OSSLShim 0.9.8r 8 Dec 2011
debug1: Reading configuration data /Users/veryv/.ssh/config
debug1: /Users/veryv/.ssh/config line 1: Applying options for server1
debug1: Reading configuration data /etc/ssh_config
debug1: /etc/ssh_config line 20: Applying options for *
debug1: /etc/ssh_config line 102: Applying options for *
debug1: Executing proxy command: exec ssh root@v.backup2 nc v.server1 22 root
debug1: permanently_drop_suid: 501
debug1: identity file /Users/veryv/.ssh/id_rsa type 1
debug1: identity file /Users/veryv/.ssh/id_rsa-cert type -1
debug1: identity file /Users/veryv/.ssh/id_dsa type -1
debug1: identity file /Users/veryv/.ssh/id_dsa-cert type -1
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_6.2
debug1: Remote protocol version 2.0, remote software version OpenSSH_6.6.1p1
Ubuntu-2ubuntu2
debug1: match: OpenSSH_6.6.1p1 Ubuntu-2ubuntu2 pat OpenSSH*
```

```
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-ctr hmac-md5-etm@openssh.com none
debug1: kex: client->server aes128-ctr hmac-md5-etm@openssh.com none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(1024<1024<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Server host key: RSA d2:07:84:79:21:a7:84:84:14:ef:f1:7a:84:a5:a1:7s
debug1: Host 'v.server1' is known and matches the RSA host key.
debug1: Found key in /Users/veryv/.ssh/known_hosts:37
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password,keyboard-interactive
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /Users/veryv/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-rsa blen 279
debug1: Authentication succeeded (publickey).
Authenticated to v.server1 (via proxy).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: Sending environment.
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Sun May 17 15:41:26 2015 from 10.70.203.66
```

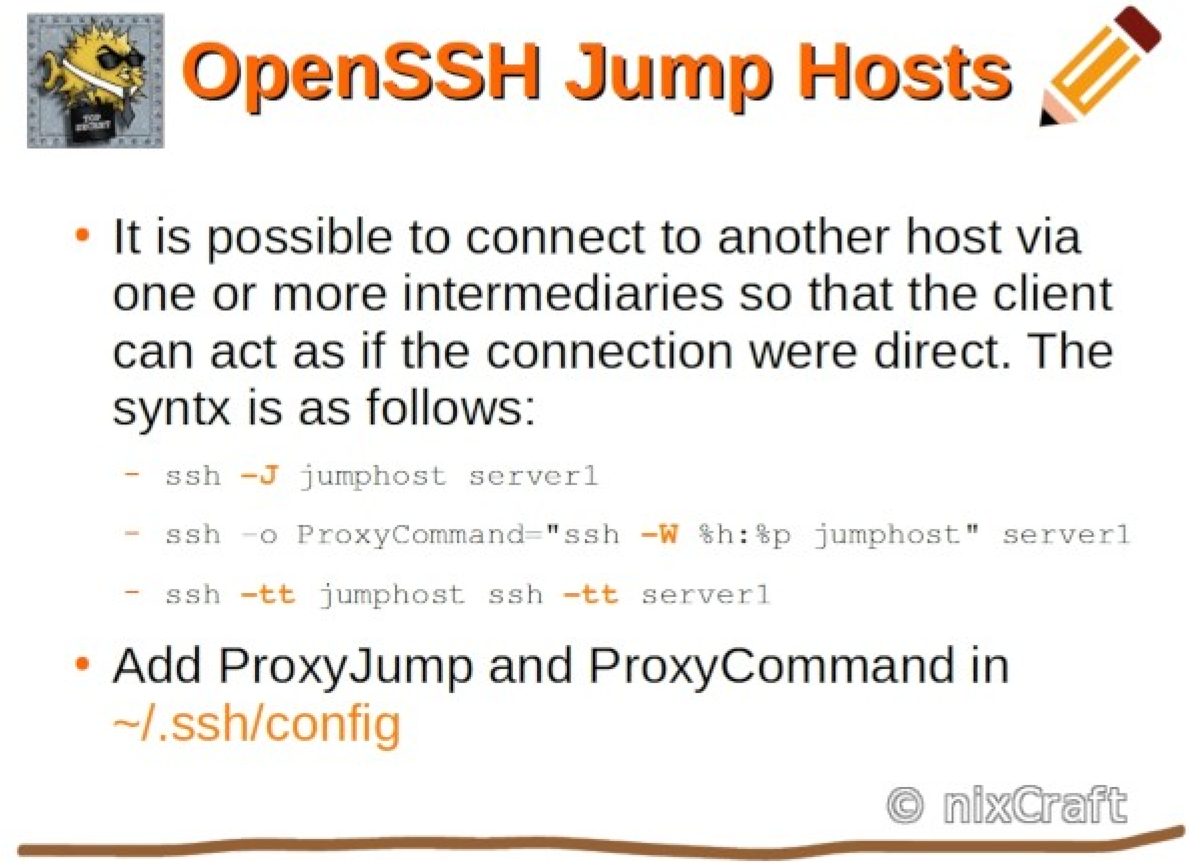
The sftp syntax

The syntax is as follows:

```
sftp -o 'ProxyCommand=ssh %h nc firewall.nixcraft.net.in 22' \
       -o 'HostKeyAlias=firewall.nixcraft.net.in' \
       vivek@server1.nixcraft.net.in
```

Conclusion

You learned about SSH ProxyCommand and ProxyJump command with examples. Here is a quick summary:



OpenSSH Jump Hosts

- It is possible to connect to another host via one or more intermediaries so that the client can act as if the connection were direct. The syntx is as follows:
  - `ssh -J jumphost server1`
  - `ssh -o ProxyCommand="ssh -W %h:%p jumphost" server1`
  - `ssh -tt jumphost ssh -tt server1`
- Add ProxyJump and ProxyCommand in ~/.ssh/config

© nixCraft

See openssh docs here for more info.

Share on Facebook Twitter

ADVERTISEMENTS

Posted by: Vivek Gite

The author is the creator of nixCraft and a seasoned sysadmin, DevOps engineer, and a trainer for the Linux operating system/Unix shell scripting. Get the **latest tutorials on SysAdmin, Linux/Unix and open source topics via RSS/XML feed** or weekly email newsletter.

- How To Reuse SSH Connection To Speed Up Remote Login Process Using Multiplexing
- OpenSSH Config File Examples
- Run SSH In The background After Running a GUI Linux Application
- Linux / UNIX: Generate SSH Keys
- Access Any Remote Server Port Without Modifying Firewall Settings
- Linux or UNIX securely copy files across a network computer
- How To Run Multiple SSH Command On Remote Machine And Exit Safely
- PostgreSQL psql: could not connect to server: Connection refused
- Use ssh-copy-id with an OpenSSH Server Listening On a Different Port