Accessing VirtualBox Guests from Host using SSH, WinSCP and Tunnelling

By Steven Gordon on Thu, 15/08/2013 - 3:49pm

Previously I described how to setup a virtual network [3] amongst several Linux virtual machines using VirtualBox. VirtualBox provides terminal access to those virtual machines via its own GUI. However I find the VirtualBox GUI is sometimes a bit slow and not as convenient as other terminals (e.g. Gnome Terminal on Ubuntu). In this article I describe some alternatives for connecting to your virtual machines. The include:

- Using a terminal on the host to access the guest
- Viewing and copy files to guest using a file explorer (Linux only)
- Copying files on the command line in Linux/OSX and Windows
- Copying files using a GUI in Windows
- Using host web browser to access a guest web server

1. Assumptions

I assume you have one (or more) Ubuntu Linux virtual guests installed as per my instructions [3] for setting up a virtual network. Of particular importance, Port Forwarding has been enabled in VirtualBox (see towards bottom of step 4 [4]), forwarding a host port (e.g. 2201) to the guest port 22. This means that when an application on the host connects to port 2201 (and destination address `localhost` or `127.0.0.1`), VirtualBox will forward the original application traffic to port 22 on the corresponding guest. Recall port 22 is for a SSH server (which should already be installed on each virtual machine as "OpenSSH server" was selecting during the install).

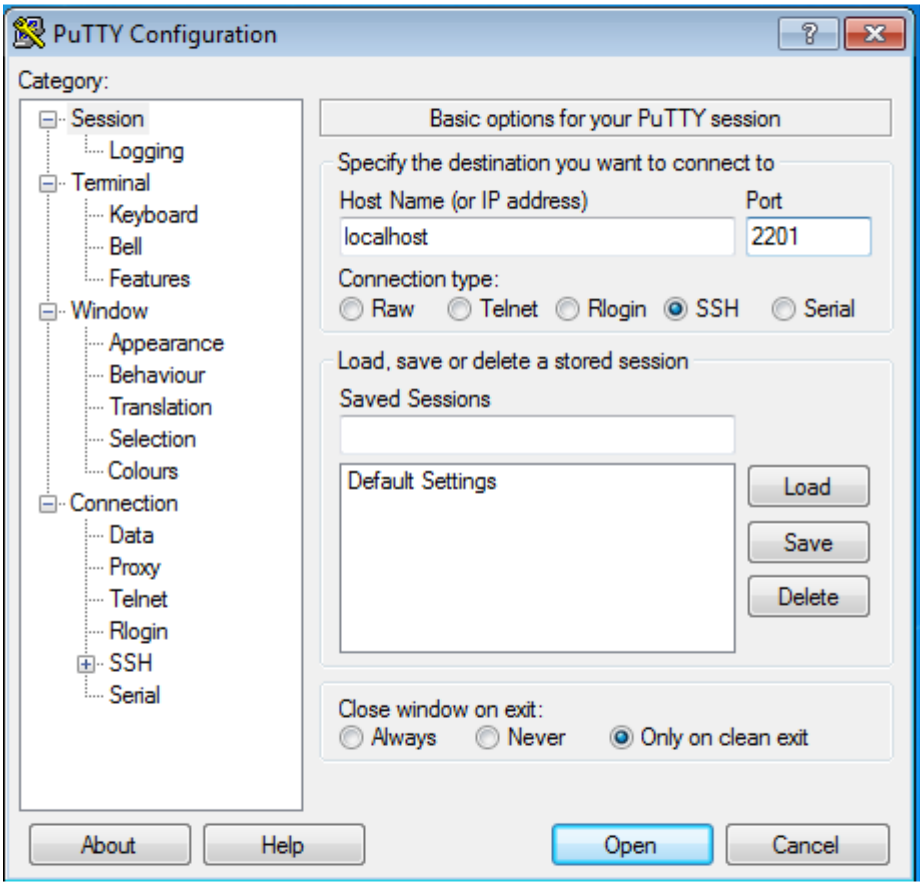
2. Accessing Virtual Machine using SSH via a Terminal

Open up a terminal in your host system. This is usually the application called *Terminal* in Linux or OSX; start PuTTY [5] if in Windows.

Now connect using the SSH client to `localhost` and port 2201 (or whichever you choose in Port Forwarding). On the command line that is performed using the *`-p`* option. The *`-l`* option may also be used to specific the username for the guest login:

```
sgordon@lime:~$ ssh -p 2201 -l network localhost
network@localhost's password:
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-23-virtual i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Fri Jun 21 14:23:35 ICT 2013

  System load:  0.16              Processes:           76
  Usage of /:   14.4% of 7.47GB   Users logged in:     1
  Memory usage: 33%               IP address for eth0: 10.0.2.15
  Swap usage:   0%                IP address for eth1: 192.168.1.10
  Graph this data and manage this system at https://landscape.canonical.com/

Last login: Fri Jun 21 13:01:01 2013 from 10.0.2.2
network@client:~$
```

In Windows using PuTTY when connecting make sure you set the *Host Name* to `localhost` and change the *Port* to 2201. It looks similar to this:



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): localhost

Port: 2201

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Create session in PuTTY to connect to virtual guest SSH server

Once connected you are now accessing the virtual guest.

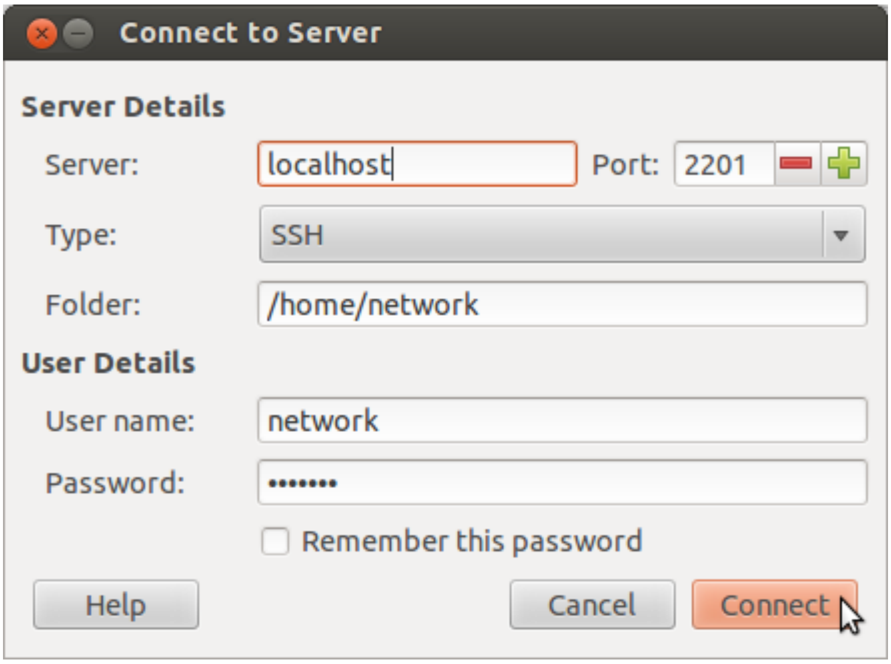
3. Exploring and Sharing Files on the Virtual Machine (Linux only)

In Linux and OSX you can use `scp` in the Terminal to copies files to/from the guest. PuTTY in Windows also supports command line based file copying with `pscp`. Another alternative is to use WinSCP to copy files in Windows. However it would be nicer to be able to view and copy files using the standard file explorer on the host. The following describes how to do this in Ubuntu Linux using the standard file explorer called Nautilus (it should be similar with other Linux distributions and file explorers). The alternative techniques - scp, pscp and WinSCP - are described later.

In Ubuntu Linux (on the host computer) you can also use Nautilus, the default file explorer, to connect to the SSH server on the virtual machine.

From the *File* menu, select *Connect to server ...*. Change the *Type* to *SSH* and then set the following:

- Server: `localhost`
- Port: `2201`
- Folder: `/home/network`
- Username: `network`
- Password: `network` (or whatever you chose)



Connect to Server

Server Details

Server: localhost

Port: 2201

Type: SSH

Folder: /home/network

User Details

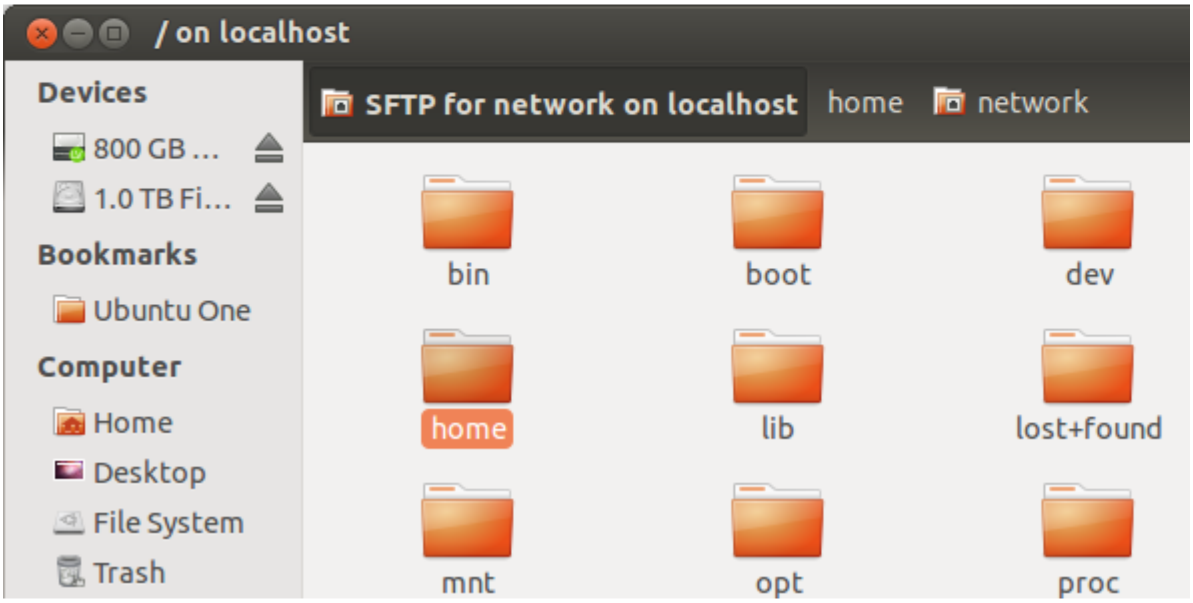
User name: network

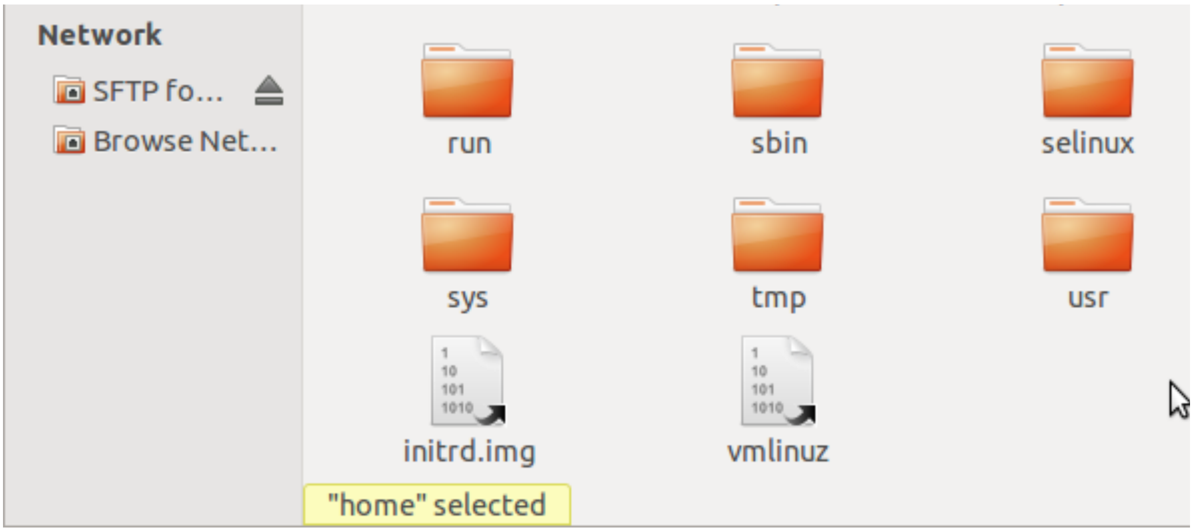
Password: •••••••

□ Remember this password

Connect to server using Nautilus

Once connected you can manage files on the virtual machine, including copy-and-paste to/from your host machine using the GUI provided by Nautilus.





Example of Nautilus for viewing folders on a virtual machine

4. Copying Files Between Host and Guest using Command Line (Linux, OSX)

If you can't or don't want to use the file explorer in your host to copy files, then you can use the command line. On most Linux/Unix systems (including OSX), `scp` can be used in a terminal on the host. The general format is to specified the source and destination, where one of them (depending which direction you are copying) includes the remote address.

To copy a file `example.txt` in your current directory on the host to the guest use the following. Note that you indicate the port using the `-P` option and the guest username is given before the `@` character.

```
$ scp -P 2201 example.txt network@localhost:/home/network/
network@localhost's password:
example.txt                                   100%    6     0.0KB/s   00:00
```

And to copy a file `index.html` in the users `tmp` diretory on the guest to the host:

```
$ scp -P 2201 network@localhost:/home/network/tmp/index.html .
network@localhost's password:
index.html                                    100%  177     0.2KB/s   00:00
```

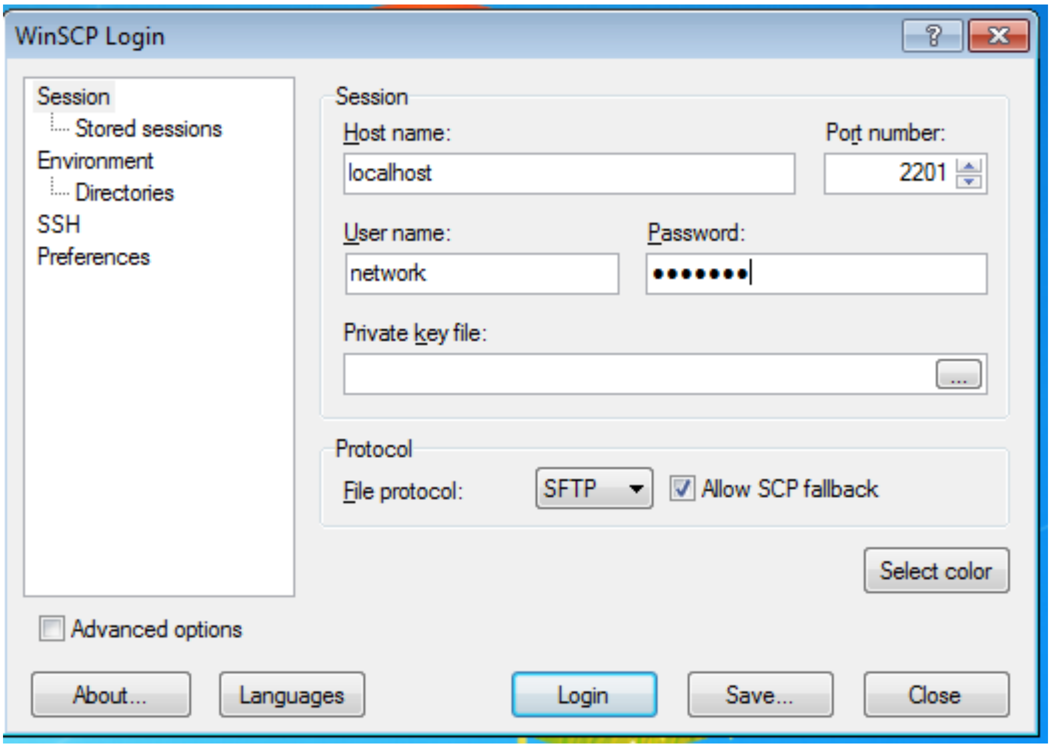
5. Copying Files Between Host and Guest using Command Line (Windows: PuTTY)

In Windows, you can use `pscp.exe` (which is included with PuTTY) in a similar manner to `scp` (see above). If you expect to use it in a regular basis, then set the path. Instructions for setting the path in Windows, and for using `pscp.exe` are given in the PuTTY manual [6].

See below if you'd prefer to use a GUI in Windows, instead of the command line.

6. Copying Files Between Host and Guest using WinSCP (Windows only)

WinSCP [7] is a free, open source Windows application that provides a GUI to transfer files to/from a SSH server, i.e. between host and guest. Download, install and the connect to the server as below:



Host name: localhost

Port number: 2201

User name: network

Password: •••••••

Private key file:

File protocol: SFTP

☑ Allow SCP fallback

☐ Advanced options

Connect to SSH server on guest with WinSCP

7. Using the Host Web Browser to Access a Guest Web Server

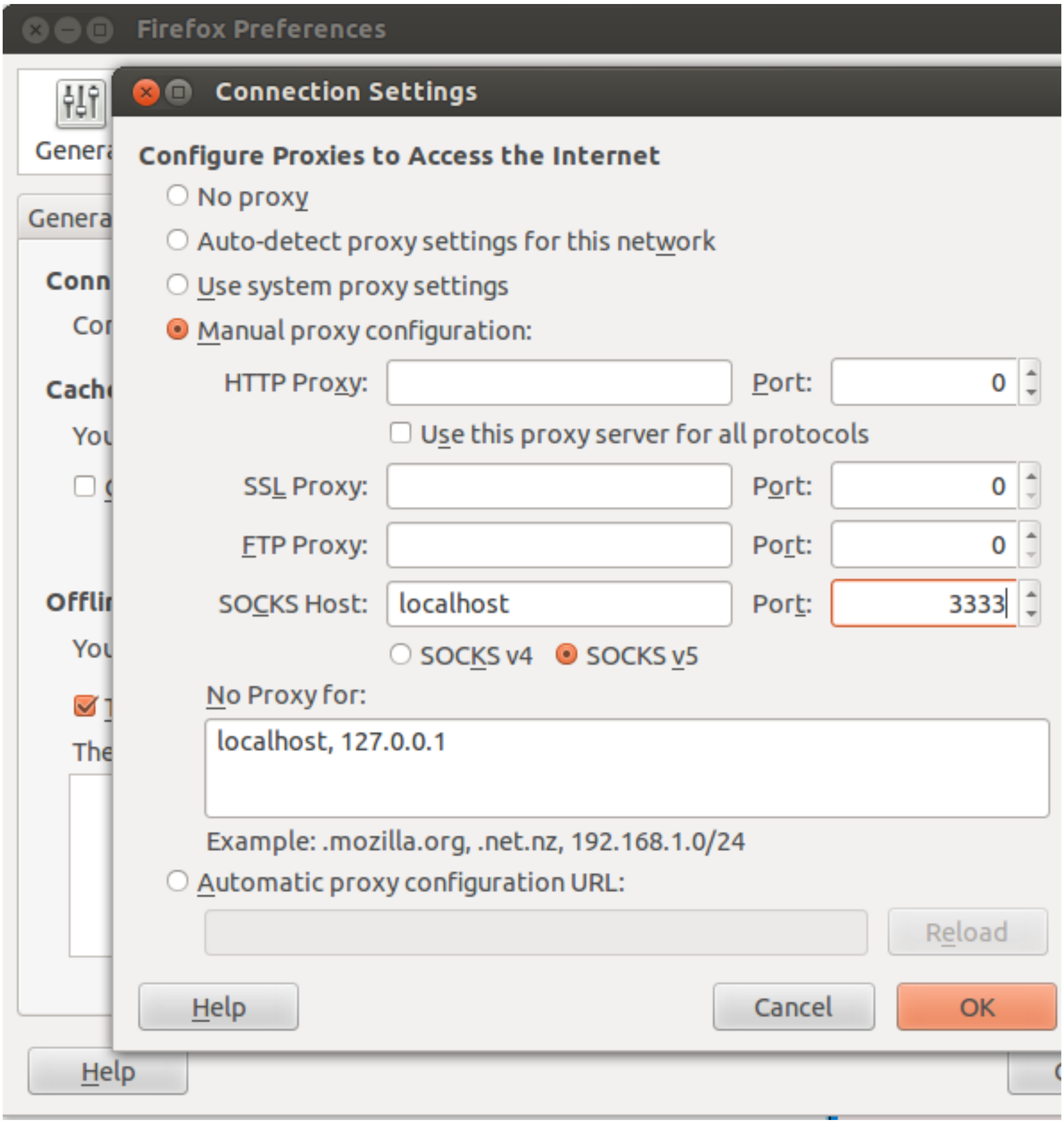
The default setup of the virtual network uses only terminal access on the guests. There is no GUI or window manager on the guests. That makes using a web browser on a guest difficult: you are restricted to a terminal based text browser like Lynx. If you want to run a full web browser, like Firefox, on a guest then a full desktop environment needs to be installed on that guest. This takes up a lot of disk space and may also require more RAM for the guest. An alternative is to use the web browser on your *host* to access the web servers on your guests. It involves SOCKS [8] tunnelling. Lets go straight into how to do it, with an explanation of how it works later.

On your host, connect to the guest using SSH and a special tunnelling option. With `ssh` on the command line in Linux/OSX, this is performed using the *-N* and *-D* options:

```
$ ssh -ND 3333 -p 2201 -l network localhost
network@localhost's password:
```

After entering the password there will be no visual response - leave this terminal open.

Now in your web browser preferences/settings, you need to enable proxying specifically using SOCKS. In Ubuntu Firefox, go to *Edit* menu and select *Preferences*. From the *Advanced* icon select the *Network* tab and press the *Settings* button. This lets you to set a proxy. Choose *Manual proxy configuration* and set the *SOCKS Host* to `localhost` and the *Port* to `3333` (or whatever you selected when starting `ssh` with the *-ND* option). It should look similar to the following:



Configure Proxies to Access the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration:

HTTP Proxy: Port: 0
□ Use this proxy server for all protocols
SSL Proxy: Port: 0
FTP Proxy: Port: 0
SOCKS Host: localhost Port: 3333
○ SOCKS v4 ● SOCKS v5

No Proxy for:
localhost, 127.0.0.1

Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:

Firefox proxy settings to tunnel to virtual guest

Now your host web browser will connect to the SSH client on the host, which in turn is connected to the SSH server on the guest. All of your host web browser traffic will go via the guest. In your web browser address bar, type in the IP address of the guest, e.g. `192.168.1.1`. You should see the web page offered by that guests web server.

This should work with most host operating systems and browsers, so long as the browser supports SOCKS proxies. If using Windows, since `ssh` is not available on the command line, you need to use PuTTY. Some instructions on using PuTTY for the SOCKS tunnelling are here [9].

Content: Howto [10]
Interest: Firefox [11]
Ubuntu Linux [12]
Virtual Machines [13]
VirtualBox [14]

Source URL: http://sandilands.info/sgordon/accessing-virtualbox-guests-using-ssh-winscp-tunnelling

Links:
[1] http://sandilands.info/sgordon/accessing-virtualbox-guests-using-ssh-winscp-tunnelling
[2] http://sandilands.info/sgordon/user/2
[3] http://sandilands.info/sgordon/creating-a-virtual-network-of-linux-guests-using-virtualbox
[4] http://sandilands.info/sgordon/creating-a-virtual-network-of-linux-guests-using-virtualbox#portforwarding
[5] http://www.chiark.greenend.org.uk/~sgtatham/putty/
[6] http://the.earth.li/~sgtatham/putty/0.63/htmldoc/Chapter5.html#pscp
[7] http://winscp.net/
[8] http://en.wikipedia.org/wiki/SOCKS
[9] http://gbar.dtu.dk/wiki/SOCKS_Proxy_with_PuTTY
[10] http://sandilands.info/sgordon/taxonomy/term/212
[11] http://sandilands.info/sgordon/taxonomy/term/318
[12] http://sandilands.info/sgordon/taxonomy/term/302
[13] http://sandilands.info/sgordon/taxonomy/term/326
[14] http://sandilands.info/sgordon/taxonomy/term/328