Basic SSH tunneling to Codio

David Croft

October 10, 2017

Abstract

This sheet looks at how to configure your Codio boxes for SSH access and how to connect to services running on those boxes.

1 What is SSH

Secure SHell (SSH) is a means of securely connecting to a remote computer so that you can interact with it. One of it's additional features is that we can tunnel network communications from one machine to another over an SSH connection in order to protect them and/pr bypass network issues/firewalls.

Tunnelling allows use to connect a port of a local machine is a port on a remote machine via an encrypted connection. Any network packets that go in one end of the tunnel will emerge from the other end of the tunnel. We are using tunnels in this case only as a means of connecting our local machine to the ports on our remote Codio box without having to worry about the configuration of the networks in the middle.

2 Codio SSH access

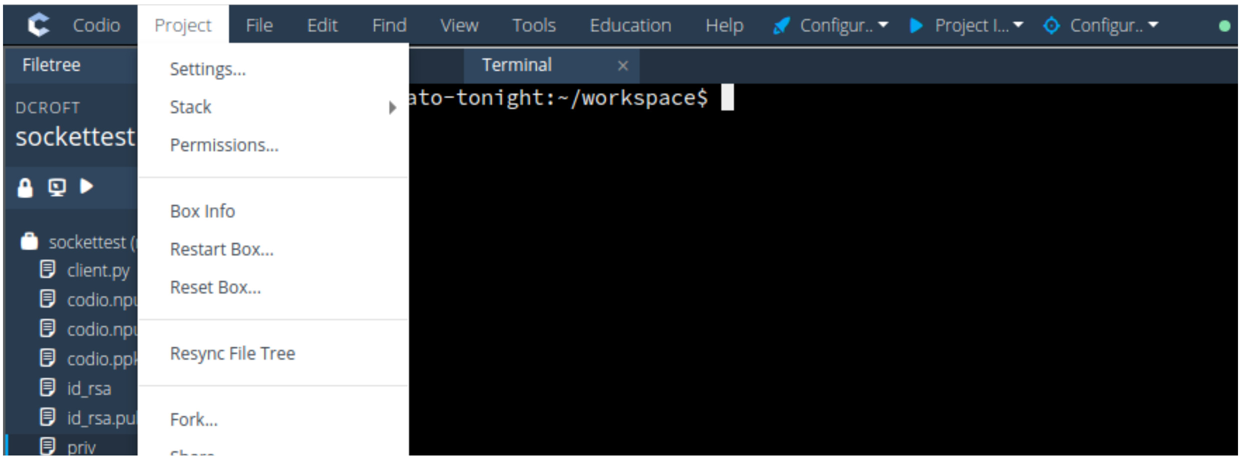
2.1 Enable SSH server

Lab work: Settings

1. Load your Codio box and then go to...

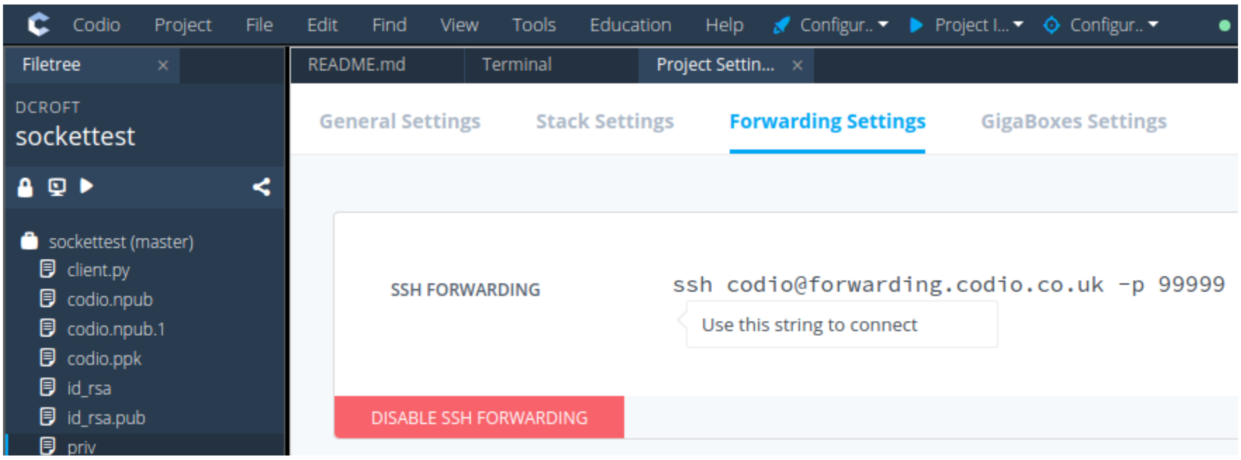
   Project → Settings... → Forwarding Settings → Enable SSH Forwarding

   ...as shown in figures 1 and 2 on the following page. Make sure to record the port number shown. In figure 2 the port shown is 99999 but yours will be different.



```
ato-tonight:~/workspace$
```

Figure 1: Enable SSH 1/2.



SSH FORWARDING `ssh codio@forwarding.codio.co.uk -p 99999`

Use this string to connect

Figure 2: Enable SSH 2/2.

2.2 Add public key

The next step is to a public key to your codio box so that we can login without using a password.

Lab work: Add key

2. On your Codio box enter the following line into the terminal, I strongly recommend that you copy and paste it to ensure that you get it exactly right.

```
$ wget -qO- http://tinyurl.com/codiossh >> ~/.ssh/authorized_keys
```

⚠ *Everyone that follows these instructions will be using the same SSH keys. This means that it is possible for other people to connect to your Codio machine if they know your connection port number. If you are concerned about security you may wish to investigate creating your own SSH keys.*

2.3 Tunneling ports

The final step is to connect to your Codio box via SSH and to tunnel ports from your local machine to the remote Codio box. This will allow programs than we run on our local machine to communicate with programs on our remote Codio machine.

You can specify ports that you want to tunnel manually, however I am providing you with a program that will connect and configure everything automatically.

Lab work: Tunneling

3. Run the putty.py file by double clicking on it.

   ⚠ *This will only work on Windows, if you are using Mac or Linux talk to one of the staff to see how to achieve the same thing.*

4. The program will ask what your Codio connection port that you noted earlier, enter it now.

2.4 Connecting

Ports 1234 to 1250 are automatically tunnelled to your Codio box, make sure that you use one of these ports to run your processes on. If you want/need additional or different ports then talk to a staff member or investigate tunneling in your own time.

Lab work: Connect

5. You are now able to connect to services on your remote Codio box.
   - In order to connect you need to connect to localhost:PORT, where PORT is the correct port number. The communication will then travel via the tunnel to the Codio Box.