

Pre-setup

1. Download [copSSH](#)
2. Download [msysgit](#)
3. Download [TortiseGIT](#)
4. Download [PuTTY Installer](#)

Step 1 – Installing copSSH

1. You've got copSSH – now go on and install it.
2. Install copSSH into the c:\SSH directory
3. Write down the SvcCOPSSH password – you will probably never need it – but just in case.
4. It will install and create a bunch of directories – *(Aside: I have found that you don't need to create a password on your Windows User account – it's not necessary if you use the settings I have later)*
5. Goto Start -> Programs -> copSSH -> Activate a User
6. You can only activate existing users on Windows – so you can choose *Your Own Account* or *Administrator* (or indeed another account but I've used the primary)
7. Uncheck the option – *Create keys for public key authentication* – we will do this ourselves.
8. Finish the Installation Process.

Step 2 – Setup copSSH

1. Open Windows Explorer – go into *C:\SSH\etc* and open BOTH your *ssh_config* and *sshd_config* WordPad (*Aside: Note – one has a "d" and one doesn't – it's an important difference*)
2. *ssh_config* – Delete the # (pound) key next to "Port" and change this to something like 4837 (or whatever Port you want to use)
3. *sshd_config* – Delete the # (pound) key next to "Port" and change this the same as that in *ssh_config* – *i.e. in this case 4837*
4. *sshd_config* – Delete the # (pound) key next to *MaxAuthTries* and make this 2
5. *sshd_config* – Delete the # (pound) key next to *RSAAuthentication* & also next to the *PubkeyAuthentication*
6. *sshd_config* – Delete the # (pound) key next to *PasswordAuthentication* and make this *no*.
7. Save all the changes to *ssh_config* and *sshd_config*.
8. Open your Router IP Address (or however you access your router) and open the Port you are using – *i.e. in this case 4837 (Aside: If you don't know it – c:\cmd -> ipconfig /all – will tell you)*
9. Save these Port changes in your router (& restart your router if needed)
10. Open *Windows Firewall with Advanced Security Settings* and create a rule which allows the Port you set in both *ssh_config* and *sshd_config* – *in this case 4837* (also do this in your Firewall if your firewall requires it – maybe/maybe not)
11. Restart Your PC

Step 3 – Installing PuTTY

1. Welcome back after the Restart – now Open the PuTTY Installer you downloaded.
2. Install Everything
3. Navigate to the Putty Installed Directory (usually *c:\Program Files\PuTTY*)
4. Open *PuttyGen.exe*
5. Enter in *Number of bits in a generated key: 4096*
6. Move your mouse around randomly as instructed until finished – **don't close PuttyGen!**
7. Open Windows Explorer and Navigate to *c:\SSH\Home\<user>\.ssh* – i.e in our case *c:\SSH\Home\Administrator\.ssh* (*Aside: If .ssh doesn't exist – you can only create it via a cmd.exe prompt – open cmd.exe and enter `cd C:\SSH\Home\<user>\` then enter `mkdir .ssh`*)
8. Create a new file called *authorized_keys* (*Aside: if it's not there – Right Click – New Text Document – Delete everything (including the *.txt) – name the file authorized_keys*)
9. Open *PuttyGen* and Copy/Paste the *Public Key* for pasting into *OpenSSH authorized_keys* filecode.
10. Still in *PuttyGen* – save the *Private Key* as *private_key.ppk* in the same *c:\SSH\Home\<user>\.ssh* – i.e. i.e in our case *c:\SSH\Home\Administrator\.ssh\private_key.ppk*
11. You should now have 2 files in this directory – *authorized_keys* and *private_key.ppk*
12. To test our connection – load up *putty.exe* from the same *c:\Program Files\PuTTY*
13. Enter your *IP Address* and *Port* in the *Session* window (*Aside: If you don't know your IP – `c:\cmd -> ipconfig /all` – will tell you*)
14. Click on *SSH* in the left hand menu and select – *Auth* – then navigate to your private key *c:\SSH\Home\<user>\.ssh\private_key.ppk*
15. Hit *Open* and a terminal will open asking you to *Enter Login Name:* – enter your *<user>* – i.e. in our case *Administrator*
16. You may get a message about *Accept Public Key* – type *yes* & if you get logged in great!

Step 4 – Installing msysgit

1. Open the *msysgit* installer package.
2. Ensure its *C:\Program Files\Git (x32 bit)* or *C:\Program Files (x86)\Git (x64 bit)*
3. Set *Use Git Bash Only* (*Aside: I wanted to only Use Git Bash console but if you want to Run Git from the Windows Command Prompt then you have to select this option – I would recommend only options 1 & 2 unless you really know what you are doing*)
4. Select *Use (Tortoise)Plink* available via *C:\Program Files\TortoiseSVN\bin\TortoisePlink.exe* (*Aside: I set this as using just PuTTY Plink.exe as opposed to TortoisePlink.exe – would recommend settings this instead to `c:\Program Files\PuTTY\plink.exe` but it's up to you*)
5. Let it install and [have a laugh in the meantime](#).
6. Once it's installed – you now have 2 windows which I will name 1. *Git Bash* (*Right Click Mouse on a file/folder in Explorer*) and 2. *Start -> Programs -> copSSH -> Start a Unix Bash Shell – Unix Bash*.
7. Finally, goto your *Git-Core* folder in the *GIT* installed directory - *C:\Program Files\Git\libexec\git-core* (x32 bit) or *C:\Program Files (x86)\Git\libexec\git-core* (x64 bit) – and copy the *files**git.exe*, *git-receive-pack.exe*, *git-upload-archive.exe* and *git-upload-pack.exe* and paste these into your *C:\SSH\Bin*.

Step 5 – Modify the User Environment

1. The problem with copSSH is that it sets its \$HOME environment to the `c:\users\ variable – and GIT looks for authorized_keys in this folder. Of course, we don't want this – we setup our Server Environment in the C:\SSH\Home\ and so we want GIT to look for keys in there.`
2. Open Windows Explorer – go into `C:\SSH\Home\ and open the .bashrc file using a text editor (Aside: Be sure not to attempt to format this file as it has Unix Encodings NOT Dos encodings – i.e. just open it in a text editor and don't use another viewer unless you want encoding errors – you don't need to read the text in this file if you are following these instructions to a tee.)`
3. Put the Cursor at the beginning of the `.bashrc` file and use CTRL+F to find the following text - `bashrc file`.
4. After the text `.bashrc file#` (after the pound with a space) – paste in - `export HOME=/c/SSH/home/<user>` - i.e. in our case `export HOME=/c/SSH/home/Administrator`
5. Ensure there is a “space” before and after this paste in – i.e. `.bashrc file#<space>export HOME=/c/SSH/home/Administrator<space>Shell Options#`
6. Save this file and close it.
7. Copy this file and navigate to your windows assigned home directory (*Aside: GIT will still be looking for your path in this directory so we need to set to refer to our `C:\SSH\Home\ directory instead`*)
8. Paste this in your `c:\users\ or c:\Documents and Settings\ – so it now has c:\users\ for example.`
9. Open both a *Git Bash (Right Click mouse in Explorer)* and a *Start -> Programs -> copSSH -> Start a Unix Bash Shell – Unix Bash shell*
10. Type `echo $HOME` – into both – they should both spit out – `/c/SSH/Home/<user>`
11. If one does not – you need to set it via - `export HOME=/c/SSH/home/<user>`

Step 6 – Install TortiseGIT

1. Install TortiseGIT per the standard installer using `plink.exe` (*Aside: This should be the same as that setting at Step 4.4*)
2. Once the install has completed, open Windows Explorer.
3. Right Click on anything and goto *TortiseGIT -> Settings*
4. Select *General* and ensure the MSysGit path is setting to `C:\Program Files\Git\bin (x32)` or `C:\Program Files (x86)\Git\bin (x64)`
5. Select *Network* and ensure that this is set to the same `plink.exe` path that we set in Step 6.1.
6. Select *Save* then *OK*.

Step 7 – Using GIT and Plink

1. Navigate to `c:\SSH\Home\<user>` and create a new directory called `<myapp>.git` – i.e. `c:\SSH\Home\Administrator\myapp.git`
2. Right click on this new directory and select *Git Bash*
3. When the window loads enter `git –bare init`
4. You’ll now see a *(BARE: master)* appear and can close the window.
5. Navigate to `c:\Program Files\PuTTY` and open `pageant.exe` - a small icon will appear in your system tray at the bottom right hand corner of your screen (*a PC with a Hat on it*) – open this and add your key from `c:\SSH\Home\Administrator\.ssh\private_key.ppk` (*Aside: I recommend adding pageant.exe to your Start → Startup folder so this will load each time you start your PC. To do this, goto Start → PuTTY and right click on pageant.exe. Then change the target path to `c:\<path-to-dir>\pageant.exe` dir:\<path_to_private_key>\private_key.ppk. Then drag pageant.exe to your StartUp folder – this will then load both pageant.exe and your private key each time the system loads. Refer [here if confused.](#))*
6. Now it’s time to clone this repository to our local development environment and finally start using GIT
7. Navigate to where you want to have your local repository – in my case `D:\Git\`
8. Right click inside this Directory and hit *Git Clone*
9. Enter your URL as `ssh://<user>@<ip_address>:<port number>/SSH/Home/<user>/<git directory>` – i.e. in our case `ssh://administrator@127.0.0.1:4837/SSH/Home/administrator/myapp.git`
10. The Directory should have the correct path – i.e. in my case `D:\Git\myapp`
11. There is no need to *Load Putty Key* as we have already done this using `pageant` in Step 7.5.
12. Hit OK and with a bit of luck (a lot of luck) you will get a successful clone.

Step 8 – The Real Test

1. Cloning is one thing – the real test is pushing a new commit.
2. Go to your newly cloned directory - `D:\Git\myapp`
3. Add a *New Text Document.txt*
4. Go back to just `D:\Git\` → *Right Click on the Folder* → *Git Commit* → “*master*” ...
5. Enter a new Commit message “*Test*” & tick the checkbox for *Not Versioned* commit on the file `New Text Document.txt`.
6. Hit *OK* & then hit *Push*.
7. Ensure *Local: master* to *Remote: master* (for this test) & *Remote: origin*
8. Hit *OK* and you should get below

```
Counting objects: 1, done.
Compressing objects: 100% (1/1)
Writing objects: 100% (1/1)
Writing objects: 100% (1/1), 244 bytes, done.
Total 1 (delta 0), reused 0 (delta 0)
To ssh://administrator@127.0.0.1:4837/SSH/Home/administrator/myapp.git
0526eba..1b1f4a4  master -> master
```

If yes (get a beer), if no (see **Problem Guide** and get ready for tears, profane language and keyboard bashing)

Problem Guide

copSSH

1. I don't understand how to install copSSH – can't you add some pictures? No, [but this guy can](#).
2. I want to add another directory instead of installing my GIT repo in the home account ? OK, [see here](#) which must be done via Unix Bash.
3. When I check services.msc – I can see that the service has stopped :(– what can I do ? You can setup a [dependent copSSH service](#) if you are brave. (*tip: if you don't know what you are doing, don't stop the service ever – when you make changes to c:\SSH\etc\ directory restart your PC instead*)

PuTTY

1. *Can I create keys with less encryption than 4096 from Step 3.5?* You can never have enough encryption so no (*well at least it's my opinion!*)
2. *I cant connect via PuTTY?* You may have to regenerate your keys via *puttygen.exe* and put them back into your */.ssh/authorized_keys* file per the *Step 3 Instructions* above.

GIT

1. *I keep getting “fatal: the remote end hung up unexpectedly”?* Open a Git Bash window and type *echo \$HOME* – ensure it is set to */c/SSH/Home/<user>/*. If it is not – enter *export HOME=/c/SSH/home/<user>*
2. *I get “fatal: connection refused”?* Check that you have correctly opened the port you have set on the service per Step 1.
3. *I get “fatal: no authorized methods accepted”?* You will have to regenerate your keys with *puttygen.exe* and follow steps per 3.
4. *I get “git-upload-pack:command not found”?* You must ensure that you have completed Step 4.7 and copied the relevant files into your *c:\SSH\Bin* directory.

Other

1. *Sorry cannot read your mind?* But these www.stackoverflow.com