Remote Software Development on a Raspberry Pi:
VS Code "Tunneling"

by este-este
(github.com/este-este/)

Based on this post: https://learn.arm.com/install-guides/vscode-tunnels/
Do not use https://code.visualstudio.com/docs/remote/tunnels#_using-the-code-cli since this is not for the raspi arm processor.

These instructions assume you are using the full VS Code installed on a laptop computer to remotely develop code using the VS Code CLI (CLI = "Command Line Interface" i.e. the "tunnel-only" VS Code server) installed on a Raspberry Pi (a "raspi"). Make sure you have installed the full VS Code, with the "Remote Development" extension pack, on the laptop. Note that you also need a github account.

Setting Up The Initial Tunnel

On the laptop, start with VS Code closed and a web browser (Chrome preferred) open.

On the raspi, in a terminal window (direct or via ssh), use the following commands:

Go to /home/pi/

$ CD /home/pi/

Determine what Arm processor the raspi has:

$ uname -m
[The output should be either "aarch64" (Armv8 64 bit) or "armv7l" (Armv7 32 bit)]

For the 64 bit "aarch64" raspi processor, use this command:

$ wget -O vscode.tgz 'https://code.visualstudio.com/sha/download?build=stable&os=cli-alpine-arm64'

For the 32 bit "armv7l" raspi processor, use this command:

$ wget -O vscode.tgz 'https://code.visualstudio.com/sha/download?build=stable&os=cli-linux-armhf'

Extract to create the executable file "code" then delete delete .tgz file

$ tar xvf vscode.tgz

Creates "code" file

$ rm vscode.tgz

Start the initial tunnel (the first time you must "accept the server license terms"

$ ./code tunnel --name <tunnel-name> --accept-server-license-terms
where <tunnel-name> is a tunnel name (<= 20 characters) of your choosing:
e.g. for <tunnel-name> = pi-tunnel

If you get this message:

```
./code: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.29' not found (required by ./code)
./code: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.28' not found (required by ./code)
```

It means that a later version of GLIBC is needed. This can happen with Ubuntu 18.04 and, since GLIBC is also used by the OS, upgrading GLIBC may cause OS problems and is not recommended. Use Ubuntu 20.04 or higher.

The VS Code CLI will be installed and begin creating a tunnel. The response will be something like:

https://github.com/login/device and use code ####-####
(where ####-#### is one-time alphanumeric code)

On the laptop, in the web browser, navigate to this, VS Code CLI provided, URL. Login with your github account and enter the identifying, VS Code CLI provided, code. This sequence associates your github account with the raspi VS Code CLI and the tunnel it provides. This is a one time setup association.

On the raspi, after you use this code, the VS Code CLI will respond with something like:

https://vscode.dev/tunnel/<tunnel-name>
where <tunnel-name> is the same name you provided above

Your initial tunnel is ready! Make note of this URL because you will need it every time you want to connect VS Code on your laptop to the tunnel that VS Code CLI provides on the raspi.

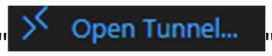
Subsequent Tunneling

Once the raspi VS Code CLI and your github account are associated, you can create and use the subsequent tunnels as follows:

1. On the raspi, in a terminal window (direct or via ssh), use the following command:

   $ ./code tunnel --name <tunnel-name>
   where <tunnel-name> is the same tunnel name you used above

2. On the laptop, in the web browser, navigate to the URL that the VS Code CLI provides.
   Should be "https://vscode.dev/tunnel/<tunnel-name>"
   where <tunnel-name> is the same tunnel name you provided

   The VS Code Dev browser window will open and act as client to connect to the tunnel that the raspi VS Code CLI server provides. When you see this option: " 

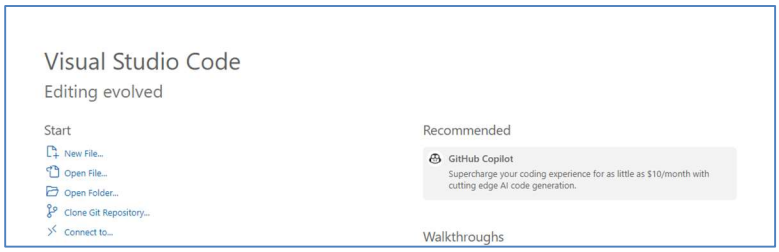
" that should mean the tunnel is in place. (You can click that item if you want to work in the VS Code Dev browser window.)

   Note: You may be prompted again to enter your github credentials. To avoid this, click on the Gear, select "Backup and Sync Settings...", check all of the sync boxes then Sign-In to github.

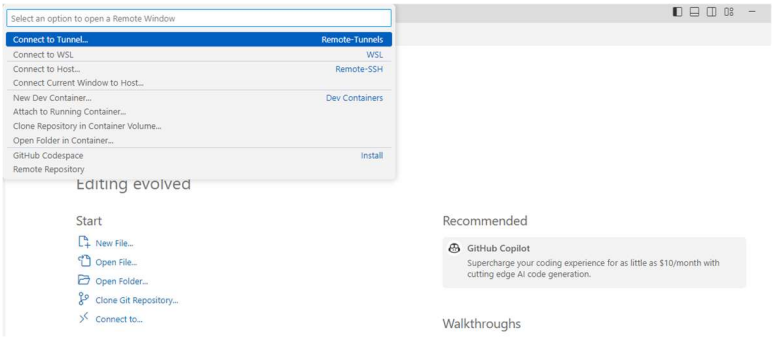
Start the full VS Code that is installed on your laptop and follow this sequence to use it as the tunnel client:

Click on the Gear "  " and make sure that syncing is on.

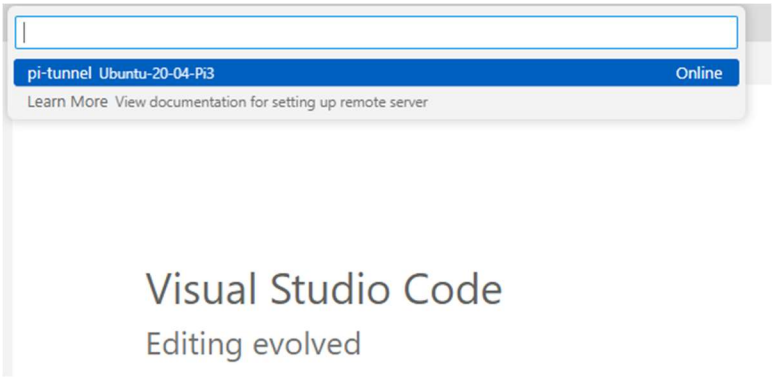
Click "Connect to..."



Select Connect to Tunnel...

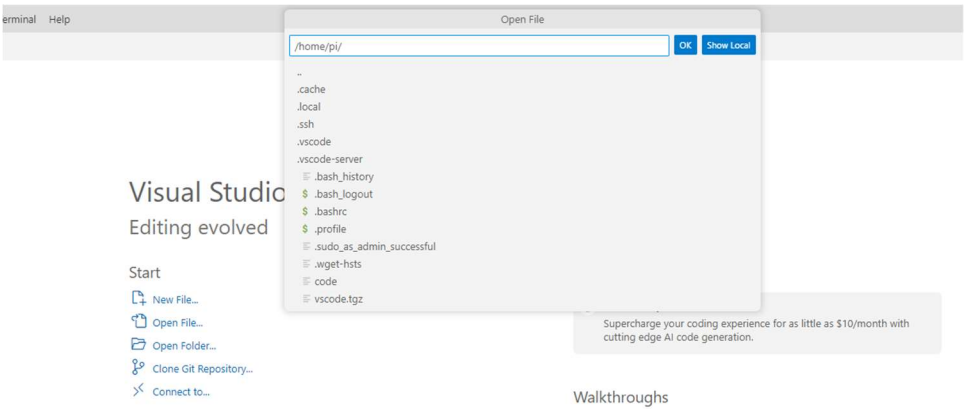


You will see the "*<tunnel_name> <raspi_name>*" and it should say "Online" because the tunnel is "up" - click on it.



The window will reset and now the Open File... and Open Folder... items point to the $HOME directory on the raspi.

Click on the " Open Folder... " option and accept the default (/home/pi). Click OK. You can then use the VS Code Explorer button ( ) to remotely navigate the folders in your raspi home directory.



Finally, you must leave open whatever terminal window you issue the "./code tunnel --name <tunnel-name>" in. That could be an ssh window but you could also log into the raspi and issue the command from a terminal window there. The latter allows you to reboot the laptop without completely bringing down the tunnel!