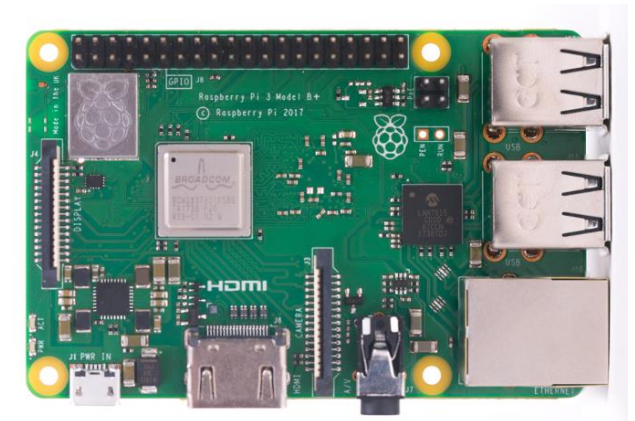




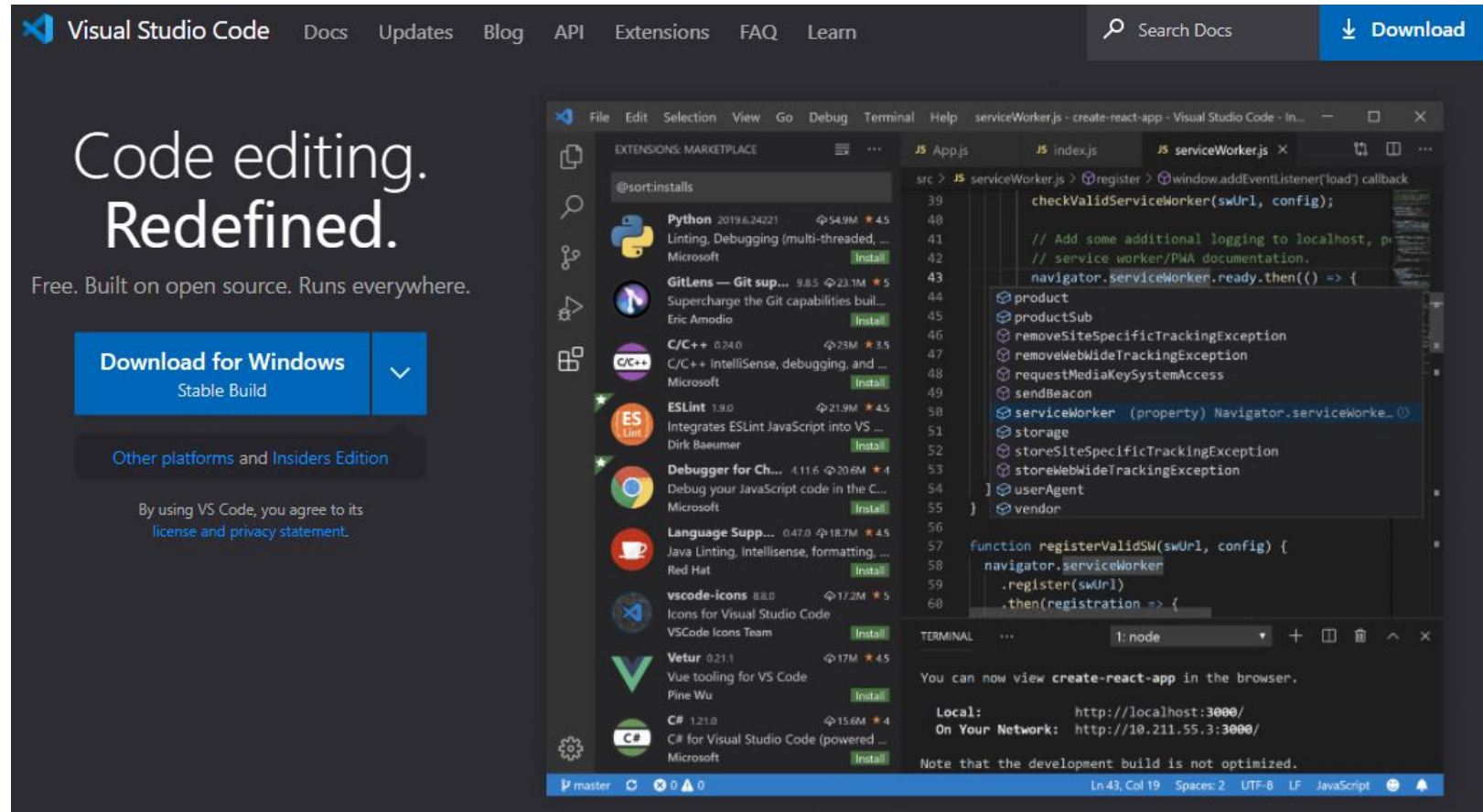
# OPTIONAL: VS Code & Remote SSH

For developing on the Raspberry Pi



# VS Code

- Editor that's Extension based

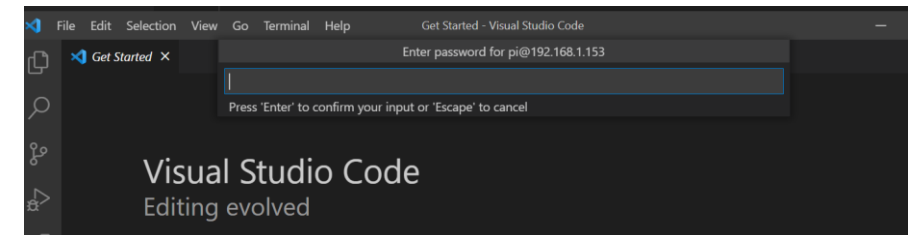
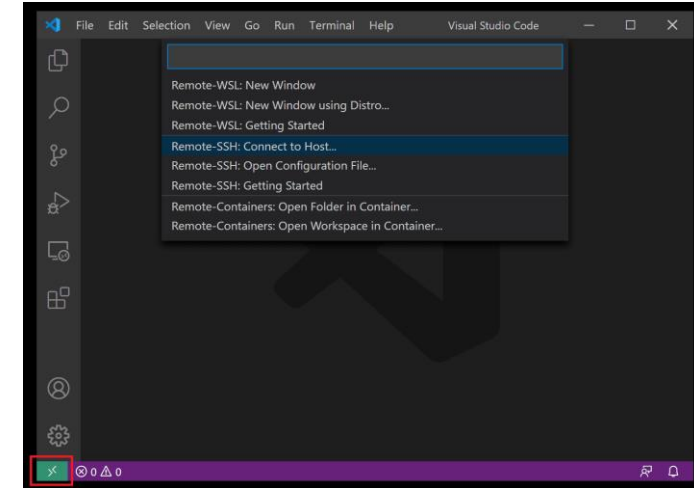


# Visual Studio Code: Remote - SSH

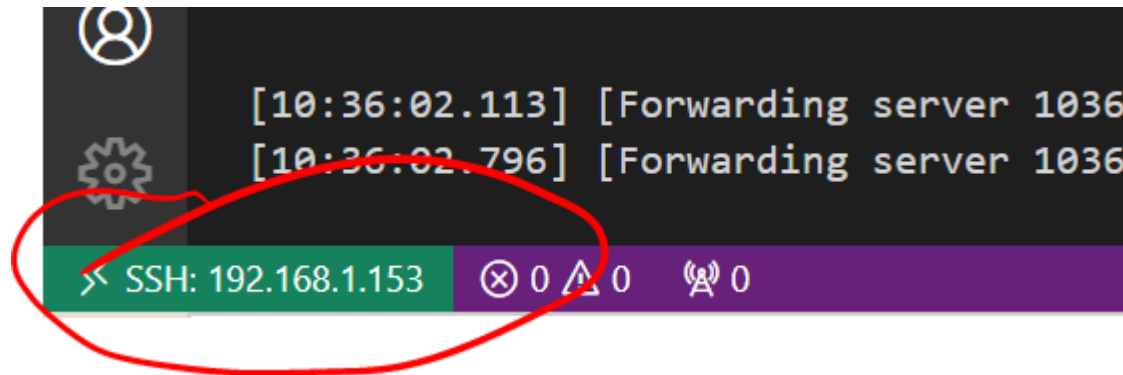
- The **Remote - SSH** extension lets you use any remote machine with a SSH server as your development environment.
- This can greatly simplify development and troubleshooting in a wide variety of situations. You can:
  - Develop on the same operating system you deploy to or use larger, faster, or more specialized hardware than your local machine.
  - Quickly swap between different, remote development environments and safely make updates without worrying about impacting your local machine.
  - Access an existing development environment from multiple machines or locations.
  - Debug an application running somewhere else such as the cloud.
- Official Tutorial Here: <https://code.visualstudio.com/docs/remote/ssh-tutorial>
- **INSTALL IT IF YOU WANT TO USE IT!**

# Connect using SSH

- When installed, an indicator will appear on the bottom-left corner of the Status bar. This indicator tells you in which context VS Code is running (local or remote). Click on the indicator to bring up a list of Remote extension commands.
- Choose the **Remote-SSH: Connect to Host** command and connect to the host by entering connection information for your Raspberry Pi in the following format: [pi@192.168.1.234](https://www.raspberrypi.com/documentation/computers/getting-started.html#ssh) (use your pi's IP address).
- You'll be prompted for the password (and asked if you trust the cert)

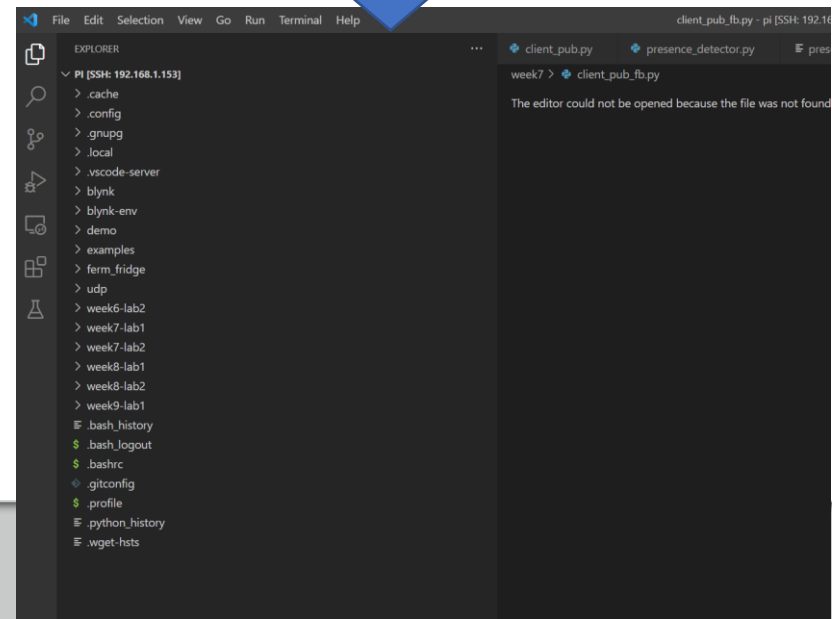
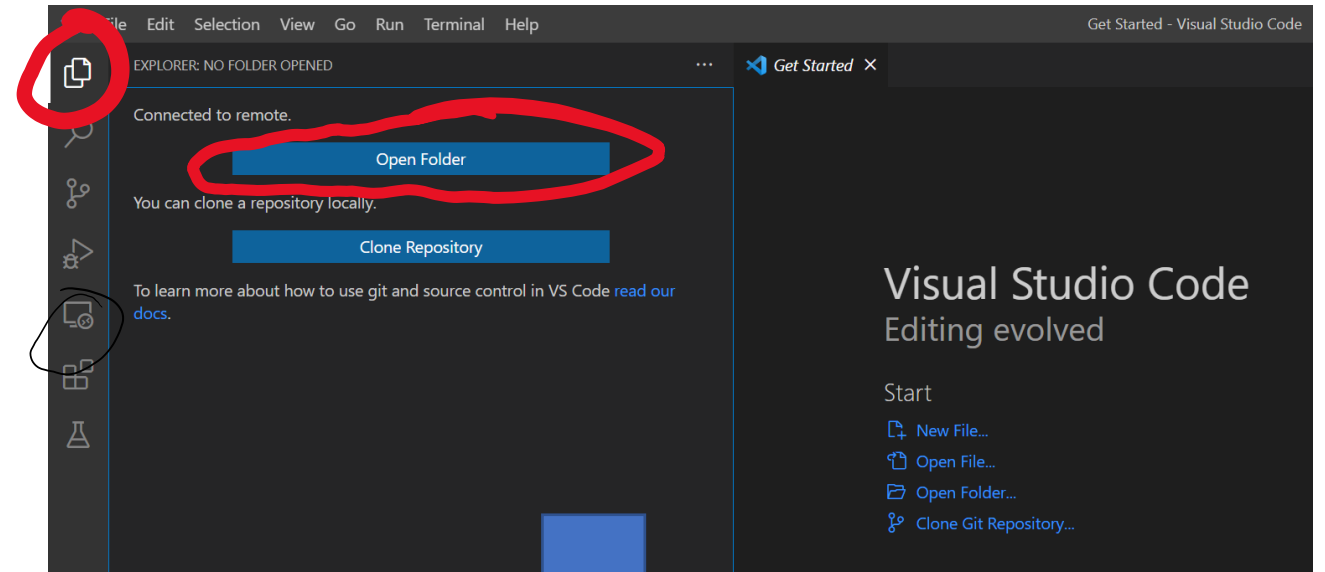


- VS Code will open a new window (instance).
- You'll then see a notification that the "VS Code Server" is initializing on the SSH Host. This might take a few minutes
- You'll know you're connected to the Rpi by looking at the indicator in the Status bar. It'll show the hostname/IP address of your Rpi when connected.



# Remote File Explorer

- Click on the Explorer icon and select “Open Folder” to connect the explorer to the Rpi
- You should then be able to see/interact with your file system on the RPi



# Finally!!!

File System on Rpi.  
Navigate folders  
Create/Delete files  
Select Files to edit

Edit Remote Files

Remote Terminal  
on the RPi

```
File Edit Selection View Go Run Terminal Help
blynk.py - pi [SSH: 192.168.1.153] - Visual Studio Code

EXPLORER
PI [SSH: 192.168.1.153]
> .cache
> .config
> .gnupg
> .local
> .vscode-server
> blynk
> blynk-env
> demo
> examples
> ferm_fridge
> udp
> week6-lab2
> week7-lab1
> week7-lab2
> week8-lab1
> week8-lab2
> week9-lab1
  blynk.py
  blynk1.py
  blynk2.py
  .bash_history
  .bash_logout
  .bashrc
  .gitconfig
  .profile
  .python_history
  .wget-hsts

blynk.py
1 import BlynkLib
2 from sense_hat import SenseHat
3
4 BLYNK_AUTH = 'P5ziNDMnaR-Cn6yx-ep5foS4KYAbRAGG'
5
6 # initialize Blynk
7 blynk = BlynkLib.Blynk(BLYNK_AUTH)
8
9 # register handler for virtual pin V1 write event
10 @blynk.on("V0")
11 def v3_write_handler(value):
12     buttonValue=value[0]
13     print(f'Current button value: {buttonValue}')
14
15 # infinite loop that waits for event
16 while True:
17     blynk.run()
18     t = time.time()
19     if t - tmr_start_time > 1:
20         print("1 sec elapsed, sending data to the server...")
21         blynk.virtual_write(2, random.randint(15, 25))
22         blynk.virtual_write(4, random.randint(15, 25))
23         blynk.virtual_write(3, target)
24         blynk.virtual_write(1, state[random.randint(0, 2)])
25         blynk.virtual_write(1, state[random.randint(0, 2)])
26         x = datetime.datetime.now()
27         blynk.virtual_write(5, x.strftime("%H:%M"))
28
29     tmr_start_time += 5
30
```

```
pi@sensePi:~ $
```