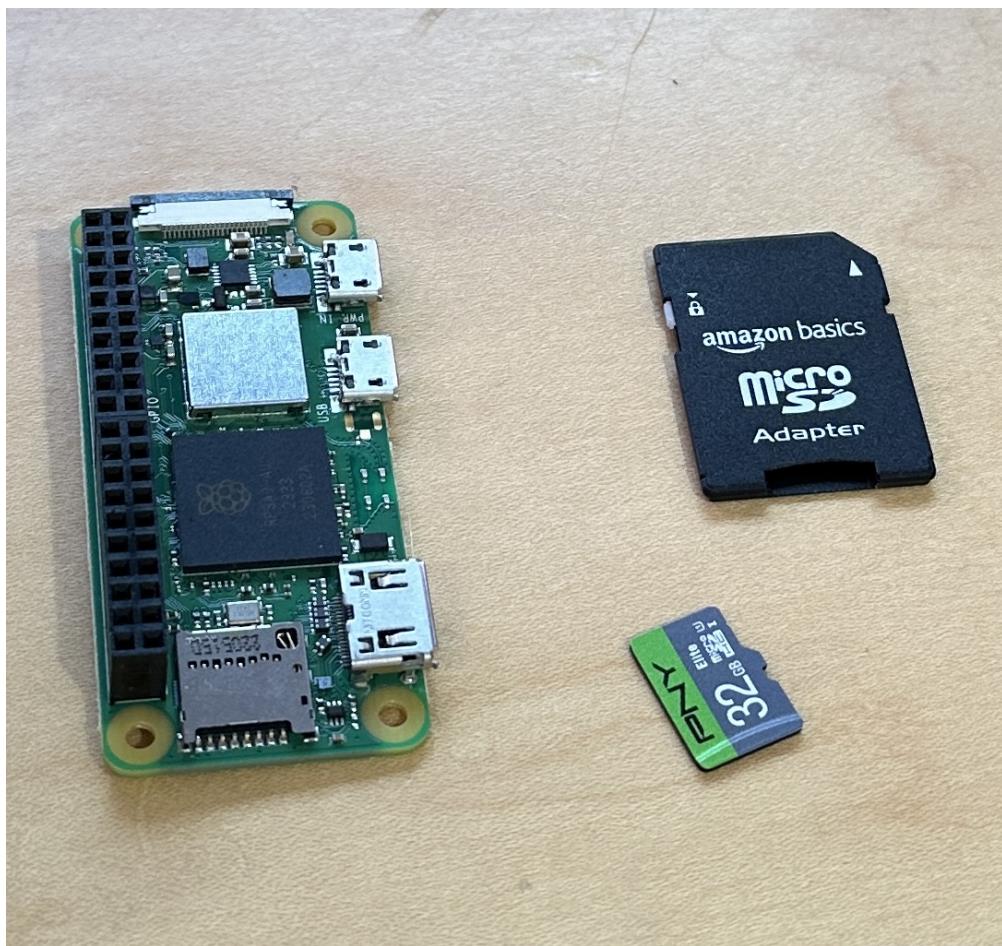


Raspberry Pi Setup Guide

April 12, 2025

Step 1: Gather Your Materials

Snag a Raspberry Pi (any model with Wi-Fi will do), an SD card (at least 8 GB, bigger is always better!), and an SD card adapter or USB reader for your computer. You'll also need a power supply for your Pi, typically a 5V adapter. Connections vary, so check if you need USB-C or micro-USB.

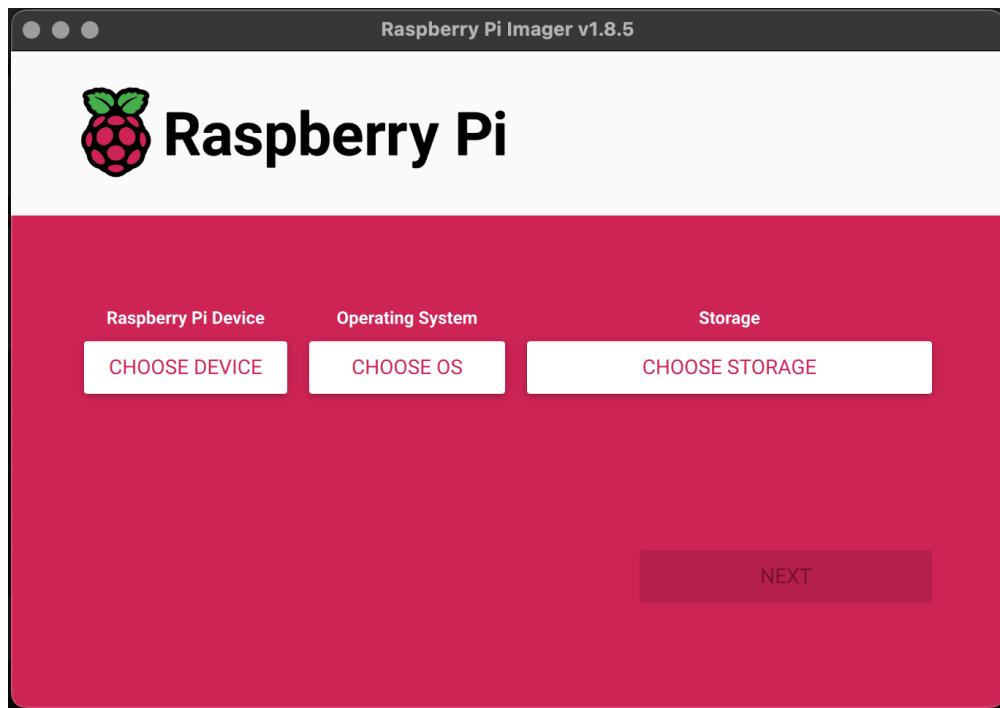


Step 2: Download the Raspberry Pi Imager

Fire up your browser and search for "Raspberry Pi Imager download." Grab the official link from raspberrypi.com.

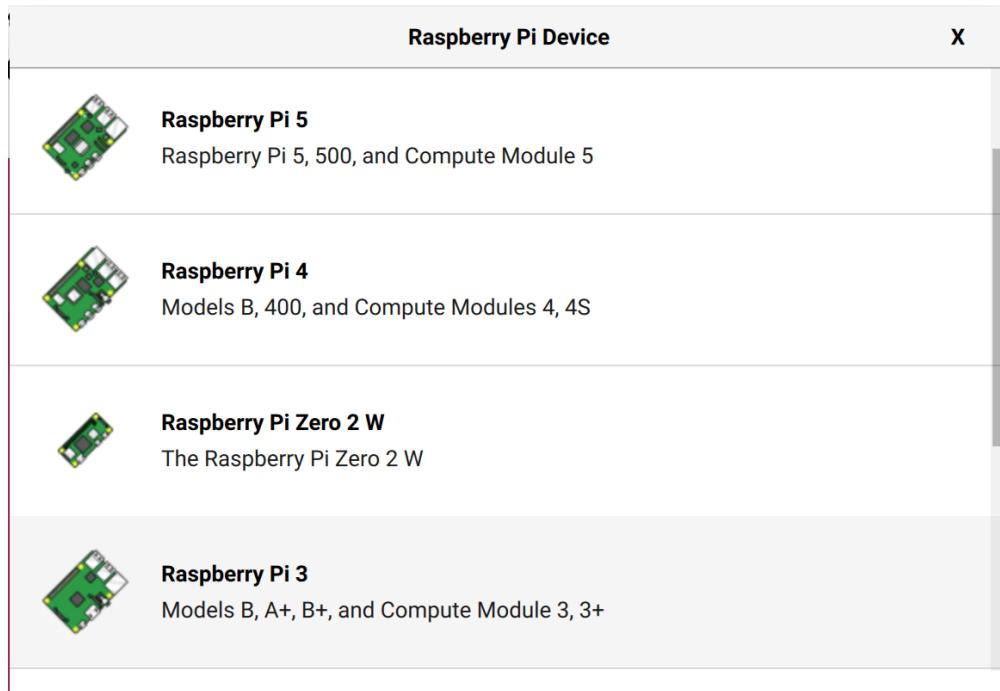


Download the Imager for your operating system and launch it after installation.

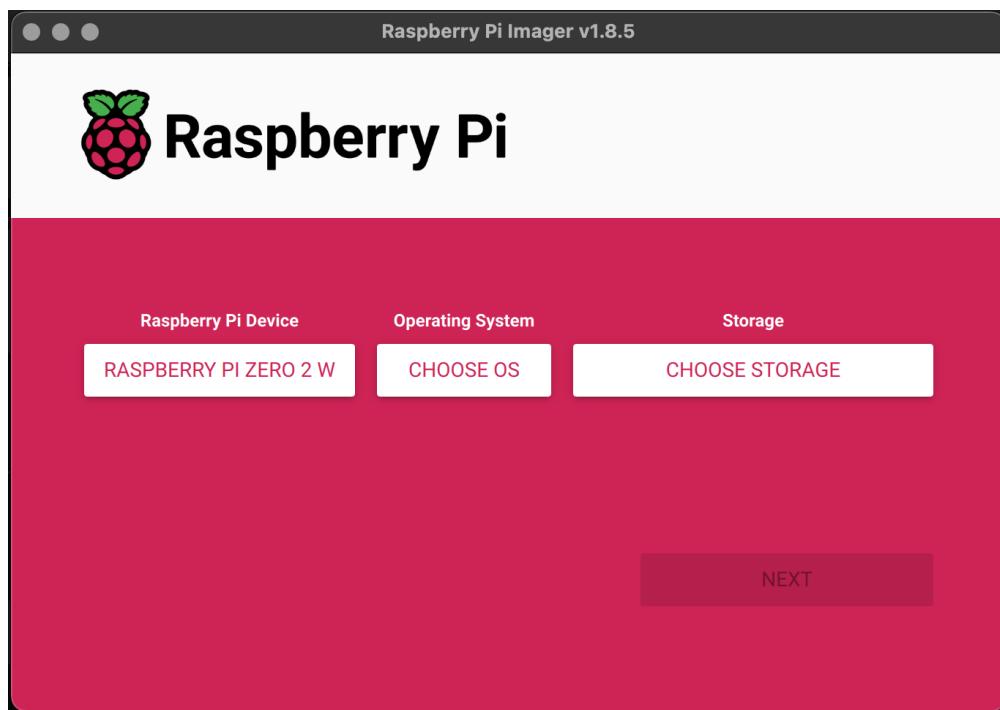


Step 3: Flashing the SD Card with Pi OS

Click on “Choose Device” and select your Raspberry Pi model.



Click “Choose OS.”



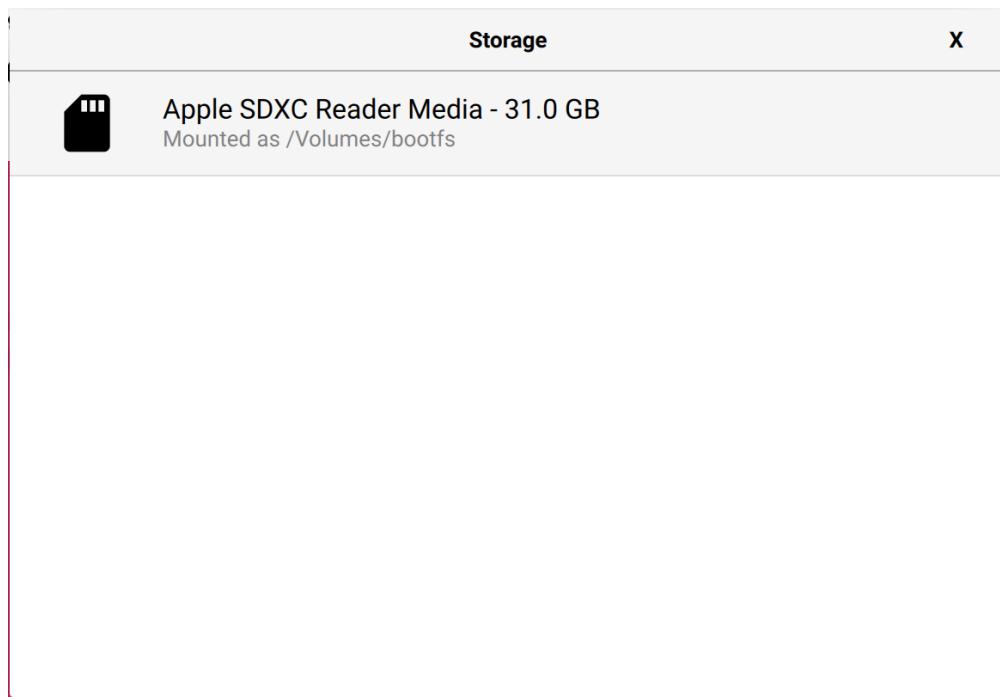
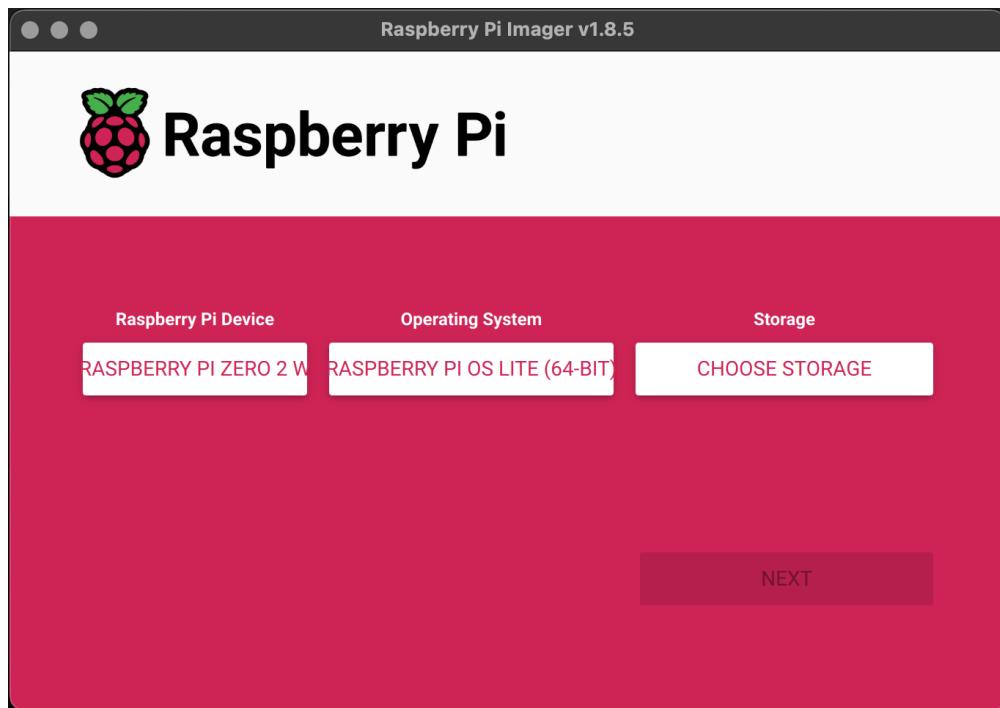
Choose “Raspberry Pi OS (other).”

Operating System	
	A port of Debian Bookworm with the Raspberry Pi Desktop (Recommended) (Recommended) Released: 2024-11-19 Online - 1.2 GB download
	Raspberry Pi OS (32-bit) A port of Debian Bookworm with the Raspberry Pi Desktop Released: 2024-11-19 Online - 1.1 GB download
	Raspberry Pi OS (Legacy, 32-bit) A port of Debian Bullseye with security updates and desktop environment Released: 2024-10-22 Online - 0.9 GB download
	Raspberry Pi OS (other) Other Raspberry Pi OS based images >

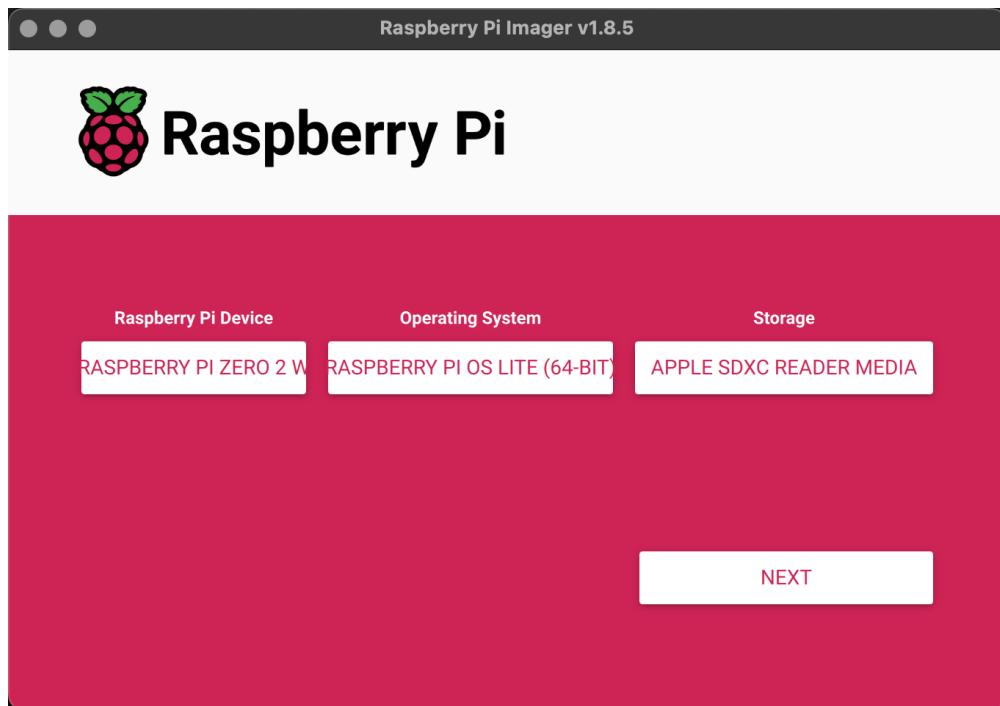
Select "Raspberry Pi OS Lite (64-bit)." (Go Lite if you're planning to SSH, it saves space and headache!)

Operating System	
< Back	Go back to main menu
	Raspberry Pi OS Lite (64-bit) A port of Debian Bookworm with no desktop environment (Compatible with Raspberry Pi 3/4/400/5) Released: 2024-11-19 Cached on your computer
	Raspberry Pi OS Full (64-bit) A port of Debian Bookworm with desktop environment and recommended applications Released: 2024-11-19 Online - 2.9 GB download
	Raspberry Pi OS Lite (32-bit) A port of Debian Bookworm with no desktop environment Released: 2024-11-19

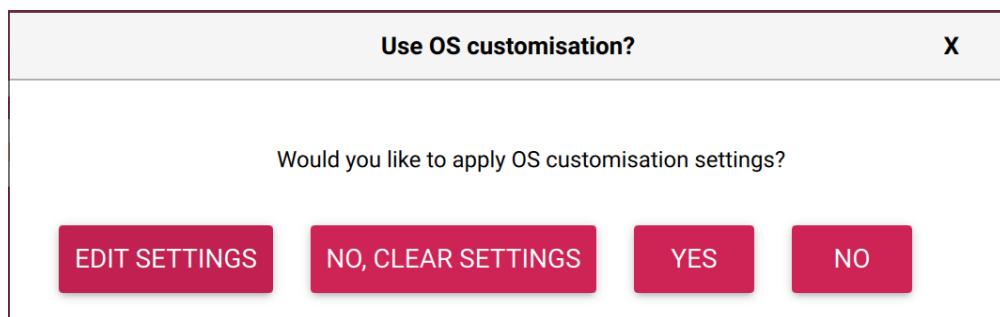
Click "Choose Storage" and select your SD card.



Click "Next."



This prompt will appear. Click "EDIT SETTINGS."



Now, set your username and password. These are your login credentials for the Pi. You can also set the hostname, which is the name you'll use to access your Pi wirelessly. Enable wireless LAN and enter your network name (SSID) and password. Once you're done, click "SAVE."

OS Customisation

GENERAL **SERVICES** **OPTIONS**

Set hostname: `raspberrypi`.local

Set username and password

Username: `pi`

Password: `*****`

Configure wireless LAN

SSID: `FreePublicWifi`

Password: `*****`

Show password Hidden SSID

Wireless LAN country: `GB`

Set locale settings

Time zone: `America/New_York`

Keyboard layout: `us`

SAVE

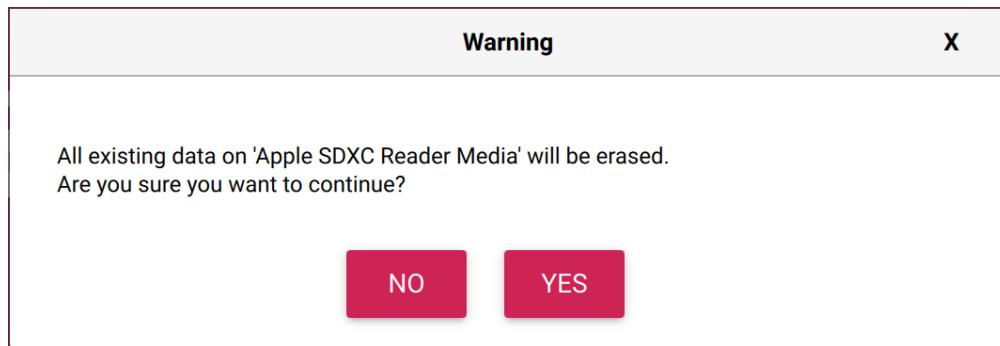
Select "YES."

Use OS customisation? X

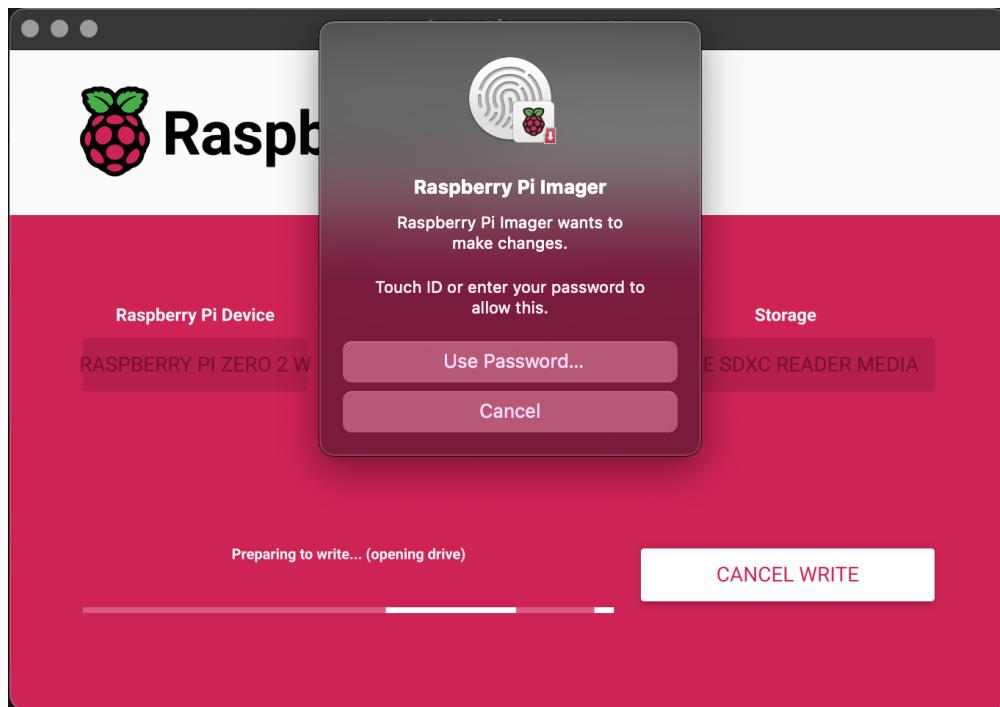
Would you like to apply OS customisation settings?

EDIT SETTINGS **NO, CLEAR SETTINGS** **YES** **NO**

Press "YES" ... again. (Because why not?)



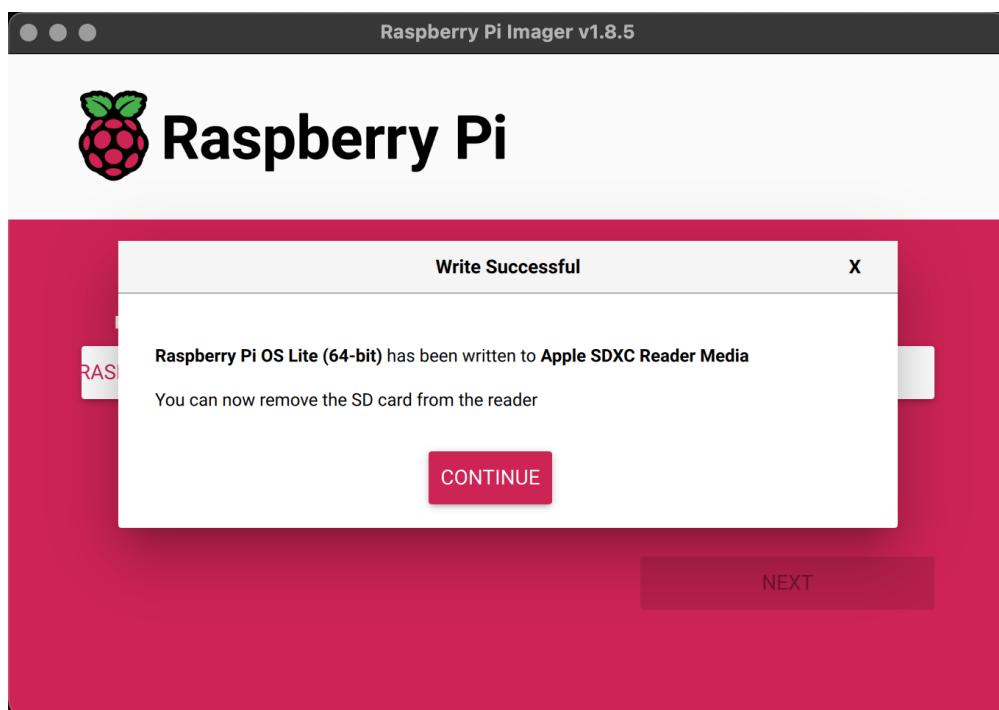
Allow the program to make changes to the SD card if you're on a Mac.

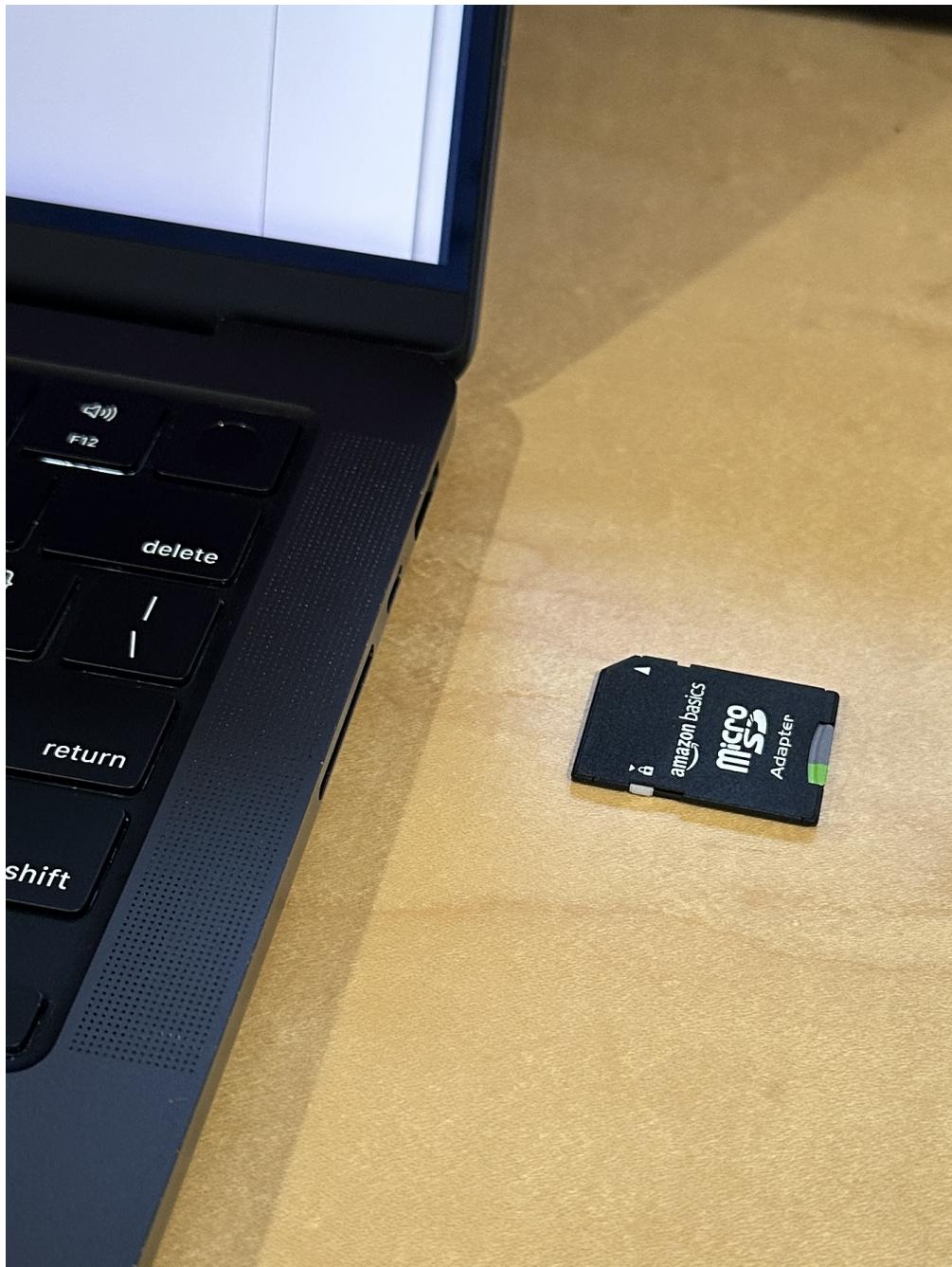


The OS will now flash onto the SD card. This will take a few minutes, so grab a snack or something.



When it's done, this pop-up will appear, and you can safely remove the SD card adapter.





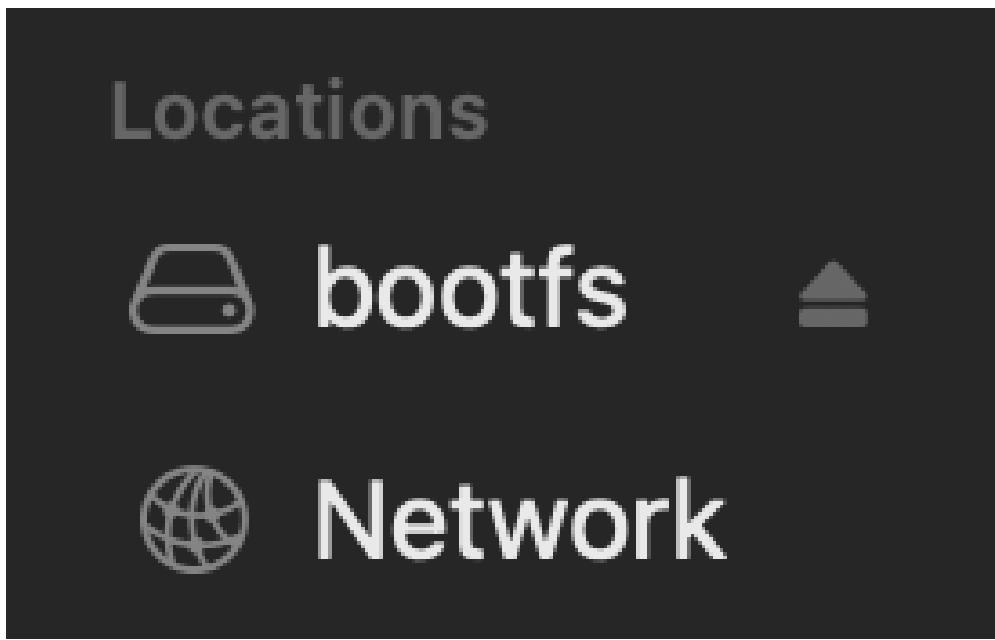
Step 4: Adding WiFi Configuration to the Boot SD

To connect to the Pi wirelessly, we need to get it on the same network as your laptop. I'm using my phone's hotspot because the building network is an enterprise-grade beast that requires more than just a simple password and SSID.

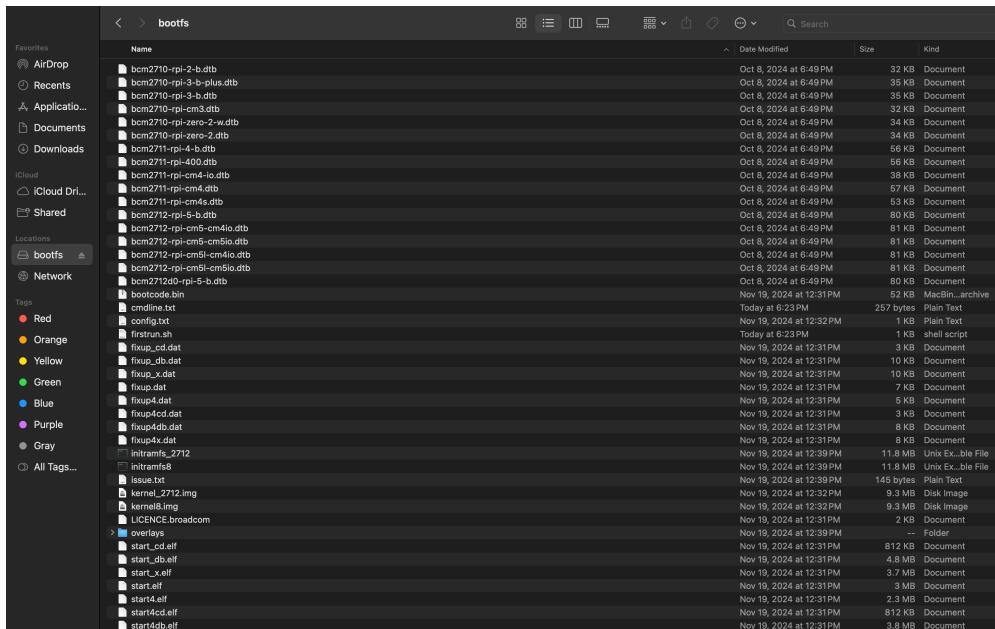
Insert the SD card reader back into your laptop, so the SD card shows up in your file manager.



Open your file manager and look for "bootfs," the boot partition we just created.



Open the boot drive. We'll need to create two files: an "ssh" file and a "wpa-supplicant.conf" file. The "ssh" file enables the Pi's SSH feature, which stands for Secure Shell, a network protocol for securely logging into and managing remote devices. The "wpa-supplicant.conf" file will store the Wi-Fi information, allowing your Pi to automatically connect to the network. This lets you SSH into your Pi from any device on the same network.



I'll use a terminal to create these files, but you can also just create a file called "ssh" (no extension!) and "wpa-supplicant.conf" using your file manager.

```

antonvalov@Antons-MacBook-Pro ~ % cd ..
antonvalov@Antons-MacBook-Pro /Users % cd ..
antonvalov@Antons-MacBook-Pro / % ls
Application Support  Volumes cores      etc      opt      sbin      usr
System             bin       dev      home     private    tmp      var
antonvalov@Antons-MacBook-Pro /Volumes % ls
Macintosh HD  bootfs
antonvalov@Antons-MacBook-Pro /Volumes % cd bootfs
antonvalov@Antons-MacBook-Pro bootfs % ls
LICENCE.brcmcon  bcm2711-rpi-400.dtb  bcm2712-rpi-cm5l-cm5io.dtb  fixup4cd.dat  issue.txt  start4x.elf
bcm2719-rpi-2-b.dtb  bcm2711-rpi-cm4.dtb  bcm2712dr9-rpi-5-b.dtb  fixup4db.dat  kernel18.img  start_cd.elf
bcm2710-rpi-3-b-plus.dtb  bcm2711-rpi-cm4_dtb  bootcode.bin  fixup4x.dat  kernel_2712.img  start_db.elf
bcm2710-rpi-3-b.dtb  bcm2712-rpi-5-b.dtb  cmdline.txt  fixup_cd.dat  overlays   start_x.elf
bcm2710-rpi-3-w.dtb  bcm2711-rpi-cm4io.dtb  config.txt   fixup_db.dat
bcm2710-rpi-2-w.dtb  bcm2712-rpi-cm5-cm5io.dtb  fixup_dt.dat  fixup_cd.dat
bcm2710-rpi-2-r.dtb  bcm2711-rpi-cm5_dtb  fixup_dtsh   fixup_db.dat
bcm2710-rpi-2-r.dtb  bcm2712-rpi-cm5-cm5io.dtb  fixup_dtsh  initramfs8
bcm2711-rpi-4-dtb  bcm2711-rpi-cm5l-cm5io.dtb  fixup4.dat   initramfs_2712
antonvalov@Antons-MacBook-Pro bootfs % touch shh

```

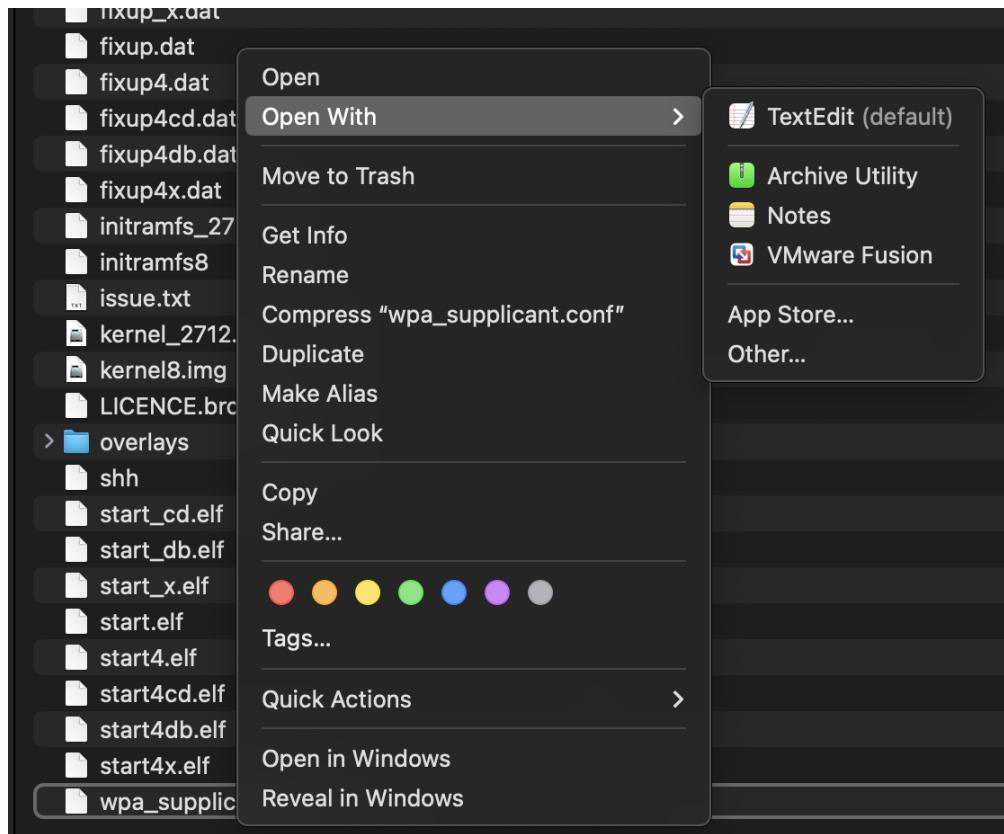
Create "ssh" and "wpa-supplicant.conf."

```
antonvalov@Antons-MacBook-Pro bootfs % touch wpa_supplicant.conf
```

As you can see, "bootfs" now contains these files.



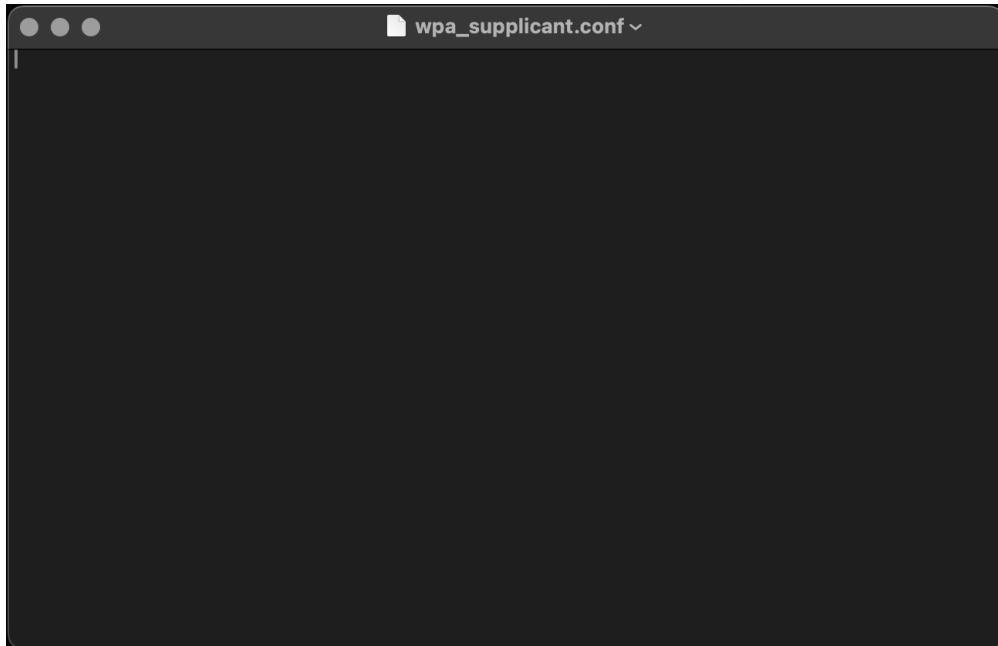
Right-click on "wpa-supplicant.conf" and open it with your favorite text editor.



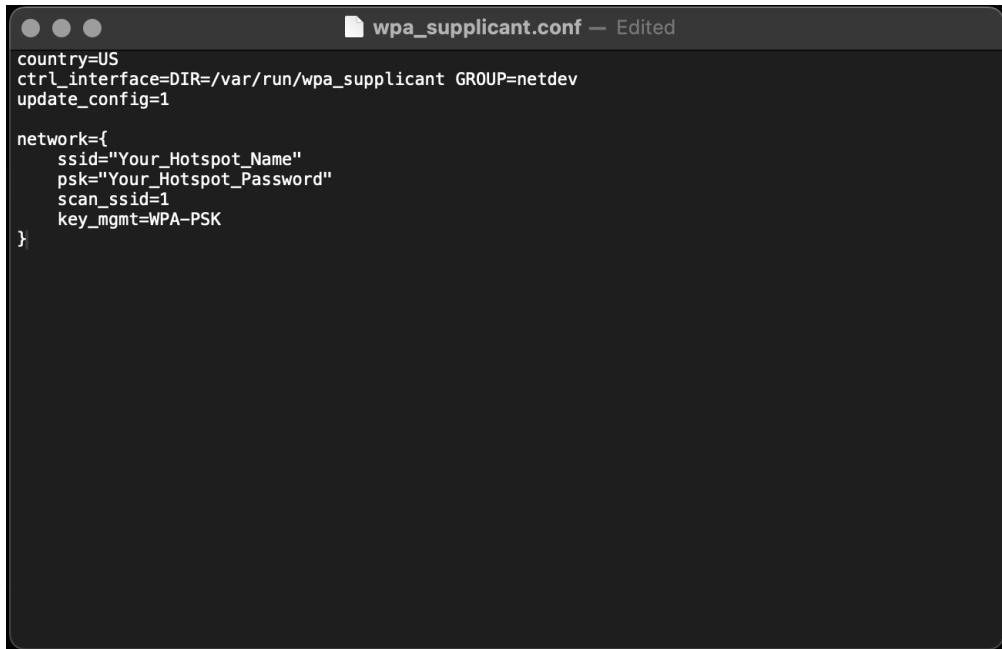
Now, copy and paste this into the file:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="Your_Network_Name"
    psk="Your_Network_Password"
    scan_ssid=1
    key_mgmt=WPA-PSK
}
```



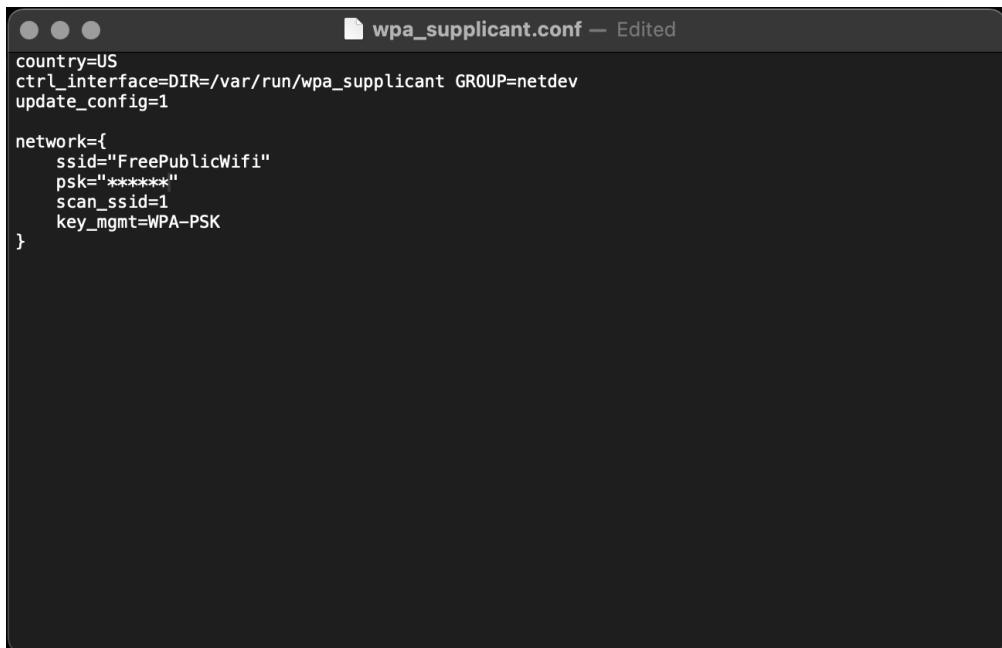
Paste the code above.



```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="Your_Hotspot_Name"
    psk="Your_Hotspot_Password"
    scan_ssid=1
    key_mgmt=WPA-PSK
}
```

Replace "Your_Network_Name" with the actual name of your network (hotspot or home Wi-Fi), and "Your_Network_Password" with the network's password.



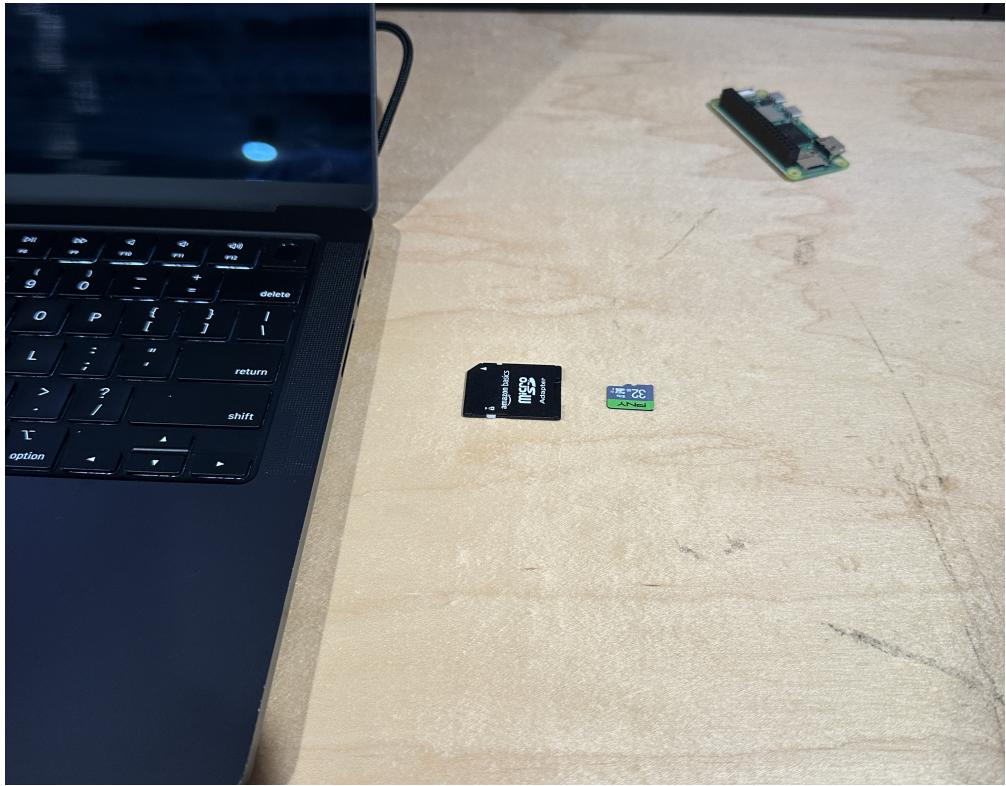
```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="FreePublicWifi"
    psk="*****"
    scan_ssid=1
    key_mgmt=WPA-PSK
}
```

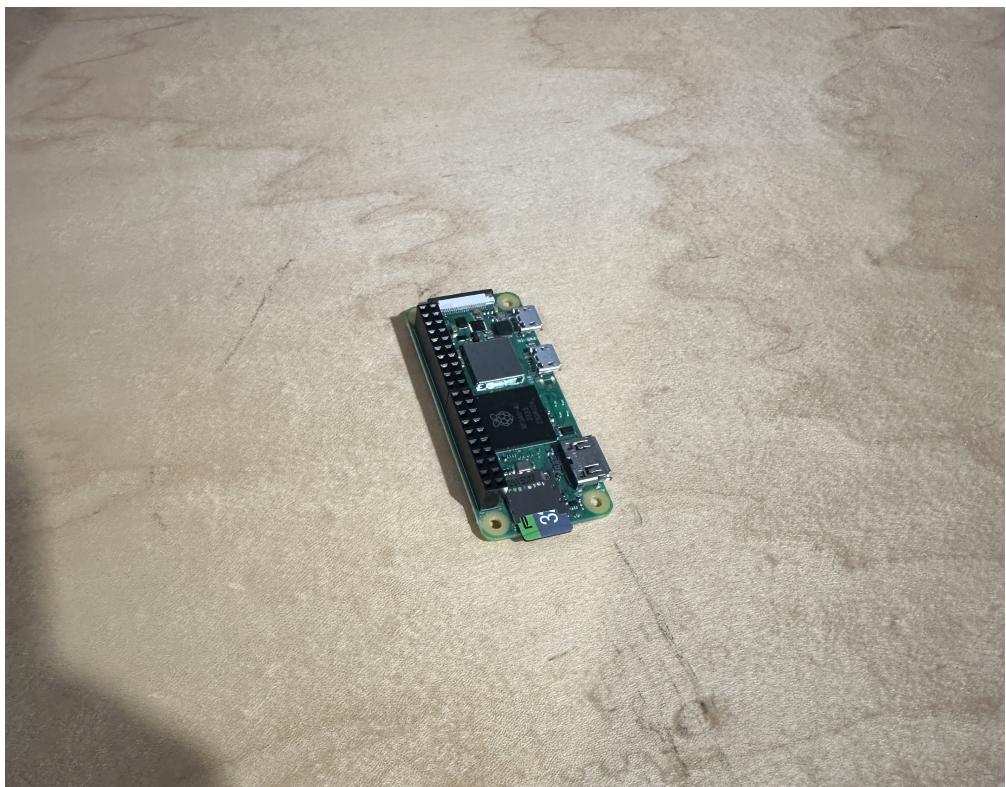
DON'T FORGET TO SAVE IT! (Ctrl+S on Windows, Command+S on Mac)

Step 5: Connecting to the Pi

Eject the SD card from your computer and remove it from the reader.

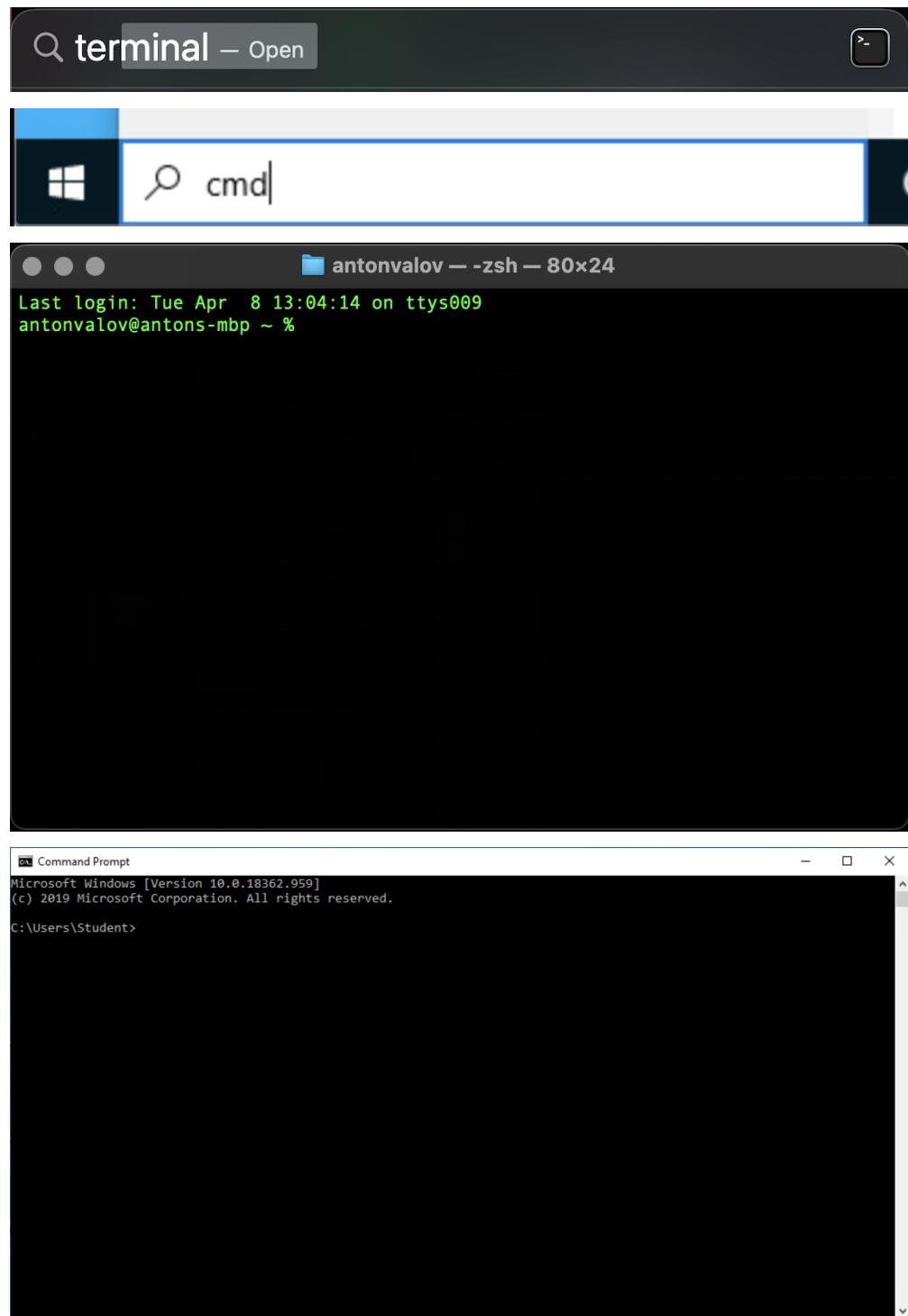


Insert the SD card into your Pi.

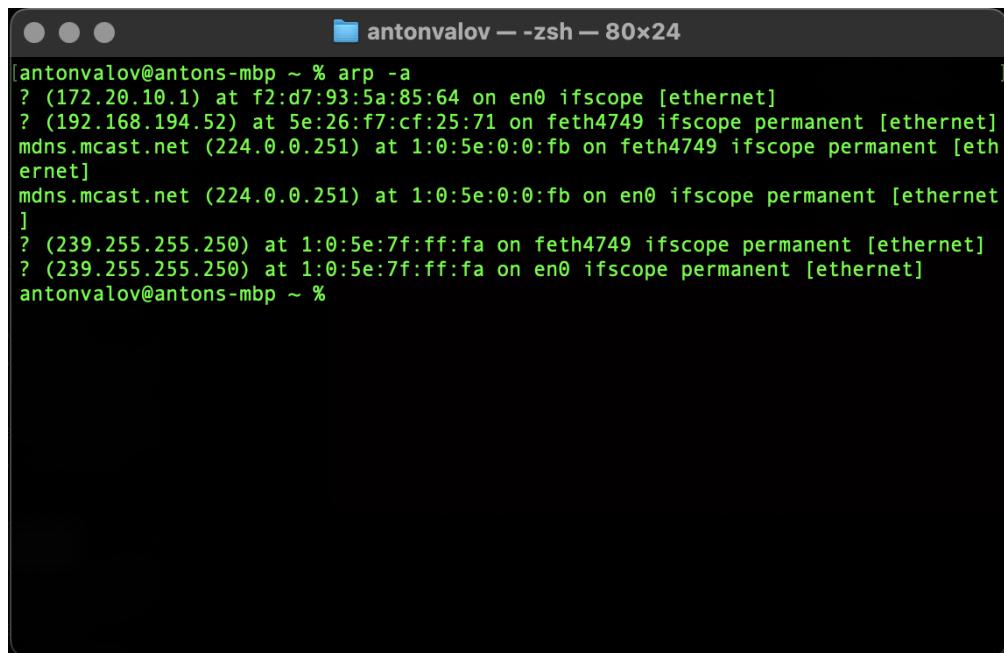


Time to open your computer's terminal: "Terminal" on Mac, or "cmd" on Windows. Search for and open

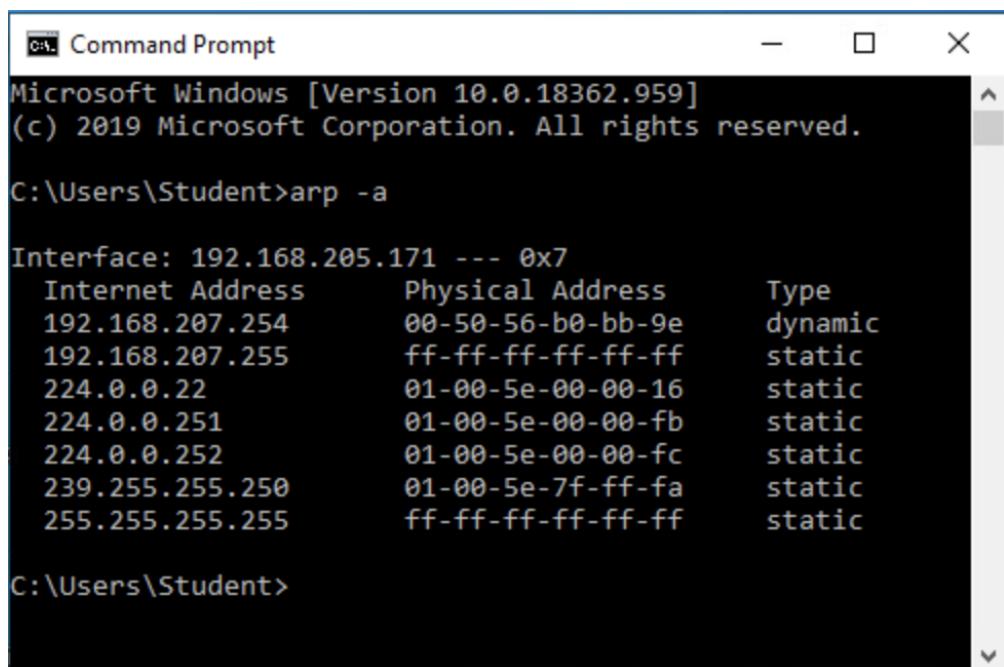
the appropriate program.



Now, type "arp -a." This shows you the IP addresses on your local network. Since your Pi isn't powered on yet, it won't appear. We'll run this command again after powering it on to find its IP address.



```
[antonvalov@antons-mbp ~ % arp -a
? (172.20.10.1) at f2:d7:93:5a:85:64 on en0 ifscope [ethernet]
? (192.168.194.52) at 5e:26:f7:cf:25:71 on feth4749 ifscope permanent [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on feth4749 ifscope permanent [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
? (239.255.255.250) at 1:0:5e:7f:ff:fa on feth4749 ifscope permanent [ethernet]
? (239.255.255.250) at 1:0:5e:7f:ff:fa on en0 ifscope permanent [ethernet]
antonvalov@antons-mbp ~ %
```



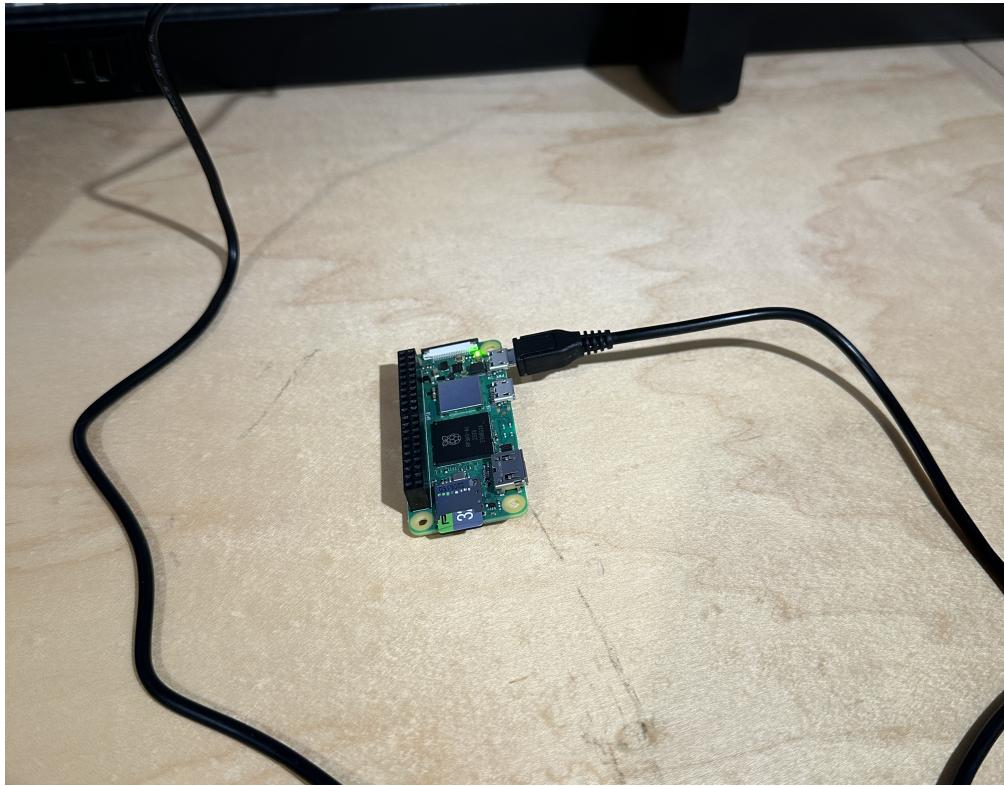
```
Command Prompt
Microsoft Windows [Version 10.0.18362.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Student>arp -a

Interface: 192.168.205.171 --- 0x7
  Internet Address          Physical Address          Type
  192.168.207.254          00-50-56-b0-bb-9e        dynamic
  192.168.207.255          ff-ff-ff-ff-ff-ff        static
  224.0.0.22                01-00-5e-00-00-16        static
  224.0.0.251              01-00-5e-00-00-fb        static
  224.0.0.252              01-00-5e-00-00-fc        static
  239.255.255.250          01-00-5e-7f-ff-fa        static
  255.255.255.255          ff-ff-ff-ff-ff-ff        static

C:\Users\Student>
```

Plug your Pi into the power supply. It'll start setting up its OS, which takes a few minutes. Be patient! Once it's done, it'll try to connect to your network. If you've done everything correctly, it'll succeed.



Run "arp -a" again.

```
[antonvalov@antons-mbp ~ % arp -a
? (172.20.10.1) at f2:d7:93:5a:85:64 on en0 ifscope [ethernet]
? (192.168.194.255) at ff:ff:ff:ff:ff:ff on feth4749 ifscope [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
?
? (239.255.255.250) at 1:0:5e:7f:ff:fa on feth4749 ifscope permanent [ethernet]
? (239.255.255.250) at 1:0:5e:7f:ff:fa on en0 ifscope permanent [ethernet]
antonvalov@antons-mbp ~ %

[antonvalov@Antons-MacBook-Pro ~ % arp -a
? (172.20.10.1) at f2:d7:93:5a:85:64 on en0 ifscope [ethernet]
? (172.20.10.2) at d8:3a:dd:bd:3:67 on en0 ifscope [ethernet]
? (172.20.10.15) at ff:ff:ff:ff:ff:ff on en0 ifscope [ethernet]
? (192.168.194.255) at ff:ff:ff:ff:ff:ff on feth4749 ifscope [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
?
? (239.255.255.250) at 1:0:5e:7f:ff:fa on feth4749 ifscope permanent [ethernet]
? (239.255.255.250) at 1:0:5e:7f:ff:fa on en0 ifscope permanent [ethernet]
antonvalov@Antons-MacBook-Pro ~ %
```

As you can see, there's a new IP address: 172.20.10.2. That's your Pi's local IP.

To connect to it via SSH, use the following command:

```
ssh {pi_name}@{local_ip}
```

In my case, that's "ssh pi@172.20.10.2."

Hmm... an error. This happens because there was a different Pi on the same IP address before, and your computer is confused. This won't happen unless you've SSHed into another Pi on the same IP. If you re-flash the SD card, you'll get this error again because it's technically a "new" Pi. To fix it, use:

```
ssh-keygen -R {ip}
```

```
antonvalov@Antons-MacBook-Pro ~ % ssh-keygen -R 172.20.10.2
# Host 172.20.10.2 found: line 5
# Host 172.20.10.2 found: line 6
# Host 172.20.10.2 found: line 7
/Users/antonvalov/.ssh/known_hosts updated.
Original contents retained as /Users/antonvalov/.ssh/known_hosts.old
antonvalov@Antons-MacBook-Pro ~ %
```

Try SSHing again.

```
[antonvalov@Antons-MacBook-Pro ~ % ssh pi@172.20.10.2
The authenticity of host '172.20.10.2 (172.20.10.2)' can't be established.
ED25519 key fingerprint is SHA256:EhsIWXX00FUT3Aa0oyguAj+7E5etXV6uI85F8dV3fAU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Type "yes."

```
[antonvalov@Antons-MacBook-Pro ~ % ssh pi@172.20.10.2
The authenticity of host '172.20.10.2 (172.20.10.2)' can't be established.
ED25519 key fingerprint is SHA256:EhsIWXx00FUT3Aa0oyguAj+7E5etXV6uI85F8dV3fAU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added '172.20.10.2' (ED25519) to the list of known hosts.
pi@172.20.10.2's password:
```

Enter the password you set during the SD card configuration.

```
pi@172.20.10.2's password:  
Linux raspberrypi 6.6.51+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.51-1+rpt3 (2024-  
10-08) aarch64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
pi@raspberrypi:~ $
```

Congratulations! You've successfully connected to your Pi's CLI (Command Line Interface) and can now control it remotely!