Setting up a Raspberry Pi as a WiFi Access Point

Created by lady ada

https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point

Last updated on 2024-06-03 01:20:27 PM EDT

Table of Contents

Overview

Would you like to use your Pi as a WiFi router? Or maybe have it as a special filtering access point? Setting up a Pi as an access point (AP) is a bit more advanced than using it as a client, but still only a half hour of typing to configure. This tutorial will make it so the Pi broadcasts a WiFi service and then routes internet traffic to an Ethernet cable. Since its all Linux you can go in and update or configure it however you like.

This guide is a streamlined version of instructions from the official Raspberry Pi Documentation (https://adafru.it/YJC). If you encounter problems with our directions, check that link, some steps may have changed since this was written. We'll update periodically!

These directions create a routed wireless access point: wireless devices connecting to the Raspberry Pi can access the internet and other devices on the network, but the reverse isn't necessarily true. This is the most common "hotspot" WiFi network approach. If you specifically require a bridged access point instead—where the Pi extends your current network and address assignment to wireless devices, and connections can work either way—these alternate directions from the Raspberry Pi Documentation (https://adafru.it/YJC) will get you going.

What you'll need

The following are **required** for this tutorial:

- Raspberry Pi (https://adafru.it/10jc) computer, preferably one with both **Ethernet** and **WiFi** built-in. The classic form-factor Pi 3 B+ or Pi 4 are ideal. Other Pi models can certainly work with the addition of dongles or IO boards, but this guide focuses on the basics.
- Ethernet internet connectivity is required; that's where we'll be routing wireless traffic.
- Ethernet cable (http://adafru.it/730)
- microSD card (4GB or larger) for Raspberry Pi OS. Early Pi models use a full-size SD card or a microSD adapter.
- Power supply for your Pi. Some also require a USB cable if not attached, either USB-C or Micro-B depending on Pi model.

The following are **optional** or required only in particular circumstances:

- A USB WiFi adapter (http://adafru.it/814) is required for earlier Pi models that don't have WiFi built-in. **Not all WiFi adapters work, we know for sure the ones in the Adafruit shop are compatible!**
- A USB Console cable (http://adafru.it/954) can be used for logging in and configuring the system, or you can remote log in via ssh over Ethernet, whichever you prefer.
- Case for your Pi (https://adafru.it/10C4).
- A SD or MicroSD card reader (http://adafru.it/939).

Our Pi starter packs (https://adafru.it/sSb) have all you need and come with more fun stuff you can play with.

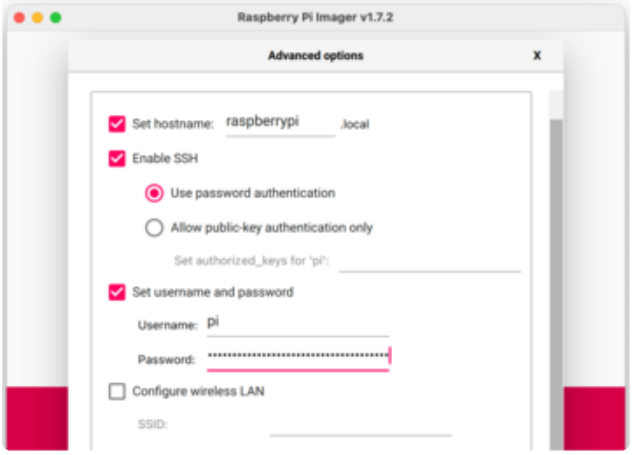
Preparation

Let's set up the SD card using Raspberry Pi Imager (downloadable here) (https://adafru.it/XMd). There are other options for this, but we really like Imager these days... you can configure most system essentials before the card is even written; no need for a post-boot setup round.

The "**CHOOSE OS**" button pops up a menu of choices. Select "**Raspberry Pi OS (other)**" and then download one of the "**Lite**" versions: 32-bit works across every Pi model, or optionally (Pi 3 and later) can use 64-bit, though don't expect any dramatic difference for this particular use. Make sure to download the OS **Lite**, not Full...we'll be using the system "headless" without a display.

Insert the **SD card** in a reader and click "**CHOOSE STORAGE**" to select it.

Once the above steps are done, this **gear icon** appears, which brings up the **Advanced options** menu...



The first four **Advanced options** settings are essential:

Set hostname to something unique for your network (prior versions of Raspberry Pi OS used raspberrypi.local by default, but for security they recommend against that now).
Enable SSH if using **Ethernet** for later setup. This is not strictly required if using a **serial console cable**, but you might have other use for it later.
Set **username and password**. These used to default to pi and raspberry, but again they recommend against this now. Pick something distinct.
DO NOT select "Configure wireless LAN." After some software installation, **we'll configure WiFi manually later...**selecting this now would interfere.
Other settings like time zone and keyboard layout are optional...you can set these as you like, or not.

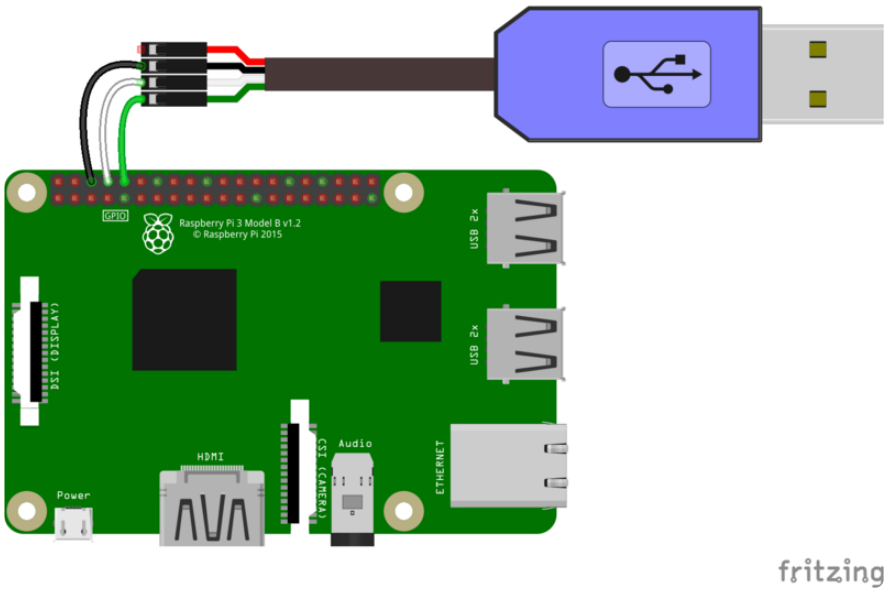
Click "**SAVE**" when done, then the "**WRITE**" button to write the OS to the SD card.

While Imager works, get the Pi ready by connecting an Ethernet cable and...if using an earlier Pi (without WiFi)...insert a USB WiFi adapter into any available USB port. This is best done now when the Pi is powered off.

After the card is **written** and **verified**, safely **eject** it, move it over to the Pi and **connect power** to the system. **Allow a few minutes for first boot** as the filesystem is expanded and ssh keys are generated.

If your Pi is connected to the Ethernet network and **ssh** is enabled, you can log in remotely from another computer with an ssh client, providing the hostname, user name and password you set earlier.

Or, if using a **serial console cable**: connect it as in the following diagram. Notice the red wire is not connected. You can now log in from another computer with a serial terminal program, providing the username and password you set earlier.

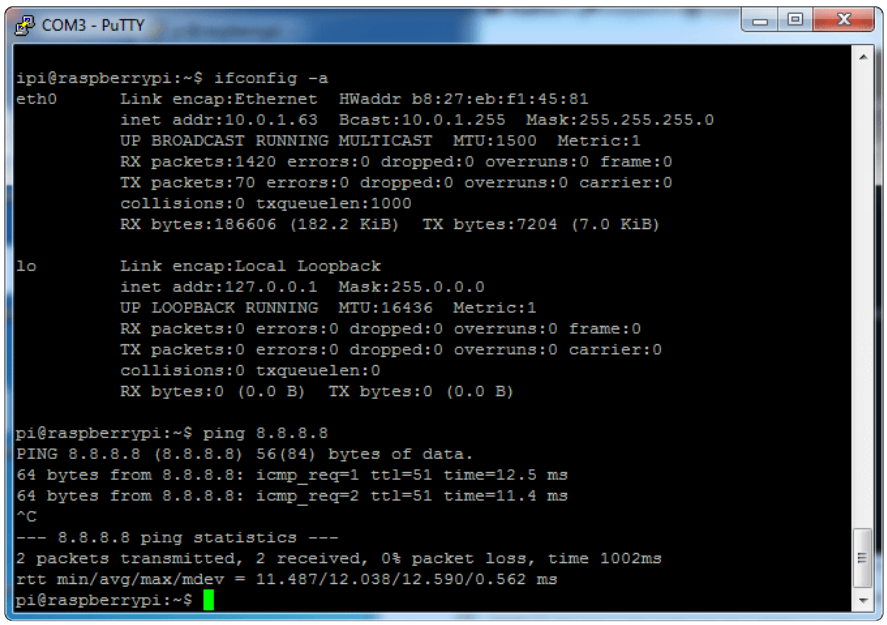


Check Ethernet & Wifi

At this point, the Pi should be booted and you should be logged in via ssh or serial console.

Confirm you can "**ping**" out from the Pi to the internet at large:

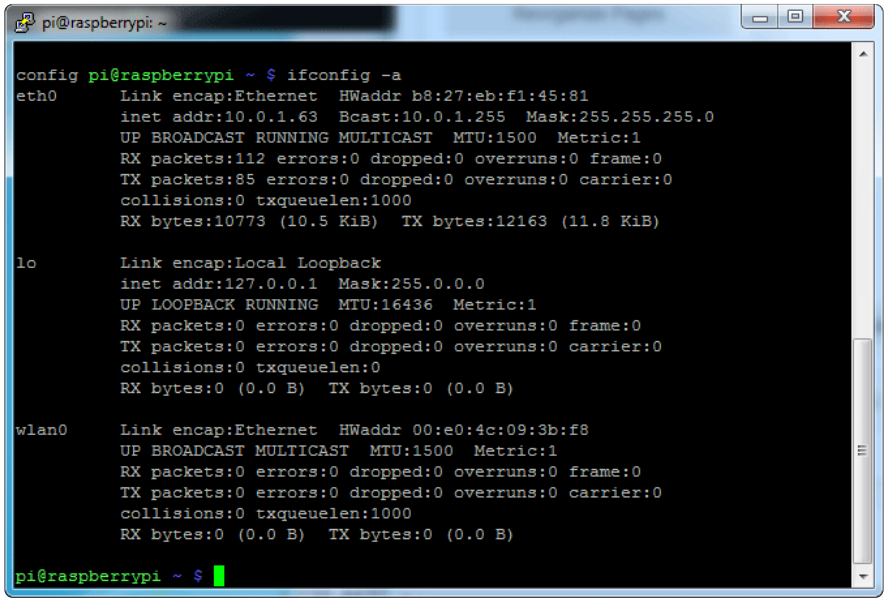
```
ping 8.8.8.8
```



If using the **serial cable**, and the ping does not respond, check that the Ethernet cable is properly seated at both ends, and that the router (or cable or DSL modem, etc.) are powered on and working for any other devices on the network.

Although WiFi is not yet configured, we can check for the existence of that hardware (e.g. if using a USB WiFi adapter):

```
ifconfig -a
```



If you see “wlan0” in the output, you’re in good shape.

Install software

A lot of persnickety typing follows, so you’ll probably want to use the “copy code” button to get these commands and settings exact.

Lets make sure the Pi’s operating system and utilities are fresh and up-to-date:

```
sudo apt update
sudo apt -y upgrade
```

The second line in particular may take a little while; perhaps 10 minutes. Take a moment to stretch or get a drink of water. Then:

```
sudo apt install -y hostapd dnsmasq
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo DEBIAN_FRONTEND=noninteractive apt install -y netfilter-persistent iptables-
persistent
sudo reboot
```

The official Pi directions (https://adafru.it/YJC) omit the reboot step, but it does seem to be necessary before certain commands later. The system will kick you off at this point. Allow a minute or so for the reboot, then log back in for more commands...

```
sudo nano /etc/dhcpcd.conf
```

Here we're using the nano editor, but you can substitute another editor of your choice. That goes for other files being edited later as well.

There's a few dozen lines of stuff in **dhcpcd.conf**. Go to the **end of the file** and add the following at the bottom:

```
interface wlan0
    static ip_address=192.168.4.1/24
    nohook wpa_supplicant
```

(The latter two lines are indented by four spaces. This is correct and normal; do not "clean up.")

Save the changes and exit nano (if that's what you're using) by pressing CTRL-O and CTRL-X. Then we'll edit another:

```
sudo nano /etc/sysctl.d/routed-ap.conf
```

This file is likely empty, so you'll simply be filling it with:

```
# Enable IPv4 routing
net.ipv4.ip_forward=1
```

Save and exit, then edit another:

```
sudo nano /etc/dnsmasq.conf
```

Everything in this file is commented out by default, so you can insert the following lines anywhere...beginning, end, or delete everything there and replace with just these lines, it's all good:

```
interface=wlan0 # Listening interface
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
                # Pool of IP addresses served via DHCP
domain=wlan     # Local wireless DNS domain
address=/gw.wlan/192.168.4.1
                # Alias for this router
```

Save and exit that, then there's one more file to edit:

```
sudo nano /etc/hostapd/hostapd.conf
```

This one's likely empty too. Begin with the following:

```
country_code=US
interface=wlan0
ssid=WirelessNetworkName
hw_mode=a
channel=48
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=WirelessNetworkPassword
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

This is just a starting point, and several lines should be changed:

WirelessNetworkName is the name (or "SSID") one sees in the WiFi network selection menu on your computer or phone; replace with something descriptive.

WirelessNetworkPassword is the corresponding password for the above.

country_code=US must be edited with the 2-letter country code for your location; the default here is for the United States. 5 GHz WiFi will not work correctly without it! Here is a list of country codes on Wikipedia (https://adafru.it/10Ca).

The lines **hw_mode=a** and **channel=48** probably will need to be edited…

hw_mode can be **a**, **b** or **g**, corresponding to different 802.11 WiFi generations. 802.11a is 5 GHz (Pi 3 B+ onwards), the others are 2.4 GHz; for USB WiFi dongles, try **g**.

channel will depend on **hw_mode** and your country. If it's a 2.4 GHz mode (**b** or **g**), this is 1–11 (and up to 14 in some places) (List of 2.4 GHz channels (https://adafru.it/10Cb)). If 5 GHz (**hw_mode=a** only), the range is much larger and a bit confusing (List of 5 GHz channels (https://adafru.it/10Cb) — look for ones with a "Yes" in your country/region's column). **48** worked for me in the **US**, but you might need to try a few different channels before one sticks, rebooting each time.

If using a non-Adafruit USB WiFi adapter, you might need to add a line: **driver=nl80211** or similar, depends on the WiFi chip used within. Check documentation for the item, also there's no guarantee all such dongles can work with the Pi.

Save and exit, then enter the following commands to finalize everything and restart the system, hopefully bringing up the WiFi network with it.

If outside the United States, change the "US" on the first line here to the 2-letter country code you previously entered in hostapd.conf.

```
sudo raspi-config nonint do_wifi_country US
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo netfilter-persistent save
sudo reboot
```

These directions are in a slightly different sequence than the canonical Pi documentation, to minimize going back-and-forth between editing files and entering commands. End result should be the same though.

Test and Troubleshooting

After rebooting the Pi, allow a minute or two for it to start up and manifest the wireless network. Look for it using another computer, a phone or a tablet...it should appear under the WiFi network list, with the name you assigned earlier.

If the Pi's wireless network does not appear...

- If using 5 GHz mode (802.11**a**), edit the **hostapd.conf** file and try a different channel, using the list mentioned earlier for recommendations. Reboot, wait and check for it again.
- Also if using 5 GHz: does the 2-letter **country_code** value match what you passed to **raspi-config nonint**? If not, fix one or the other.
- If the device you're testing from only supports the earlier 2.4 GHz WiFi modes, a 5 GHz network won't be visible at all. You may need to go back in and set **hw_mode=g** and **channel=11** or something else.