

DIY Web Hosting

Ever thought about creating your own website? Whether you are looking to create a custom blog, host files, work on your programming and scripting skills, or someone who just wants a custom email address, creating a domain and hosting your own web services is a fantastic way to flex your creative muscles. While sites like Wix can help make cookie cutter websites, self-hosting allows complete creative freedom. During this guide, we will cover everything needed to host your very own web services. From setting up your host machine, choosing your hosting software, handling router rules, setting up a domain, to adding extra security with Cloudflare. By the end of this guide, you will have your very own website running completely on your own personal hardware. During this guide, we will be using a Raspberry Pi device.



Raspberry Pi Running Webservice



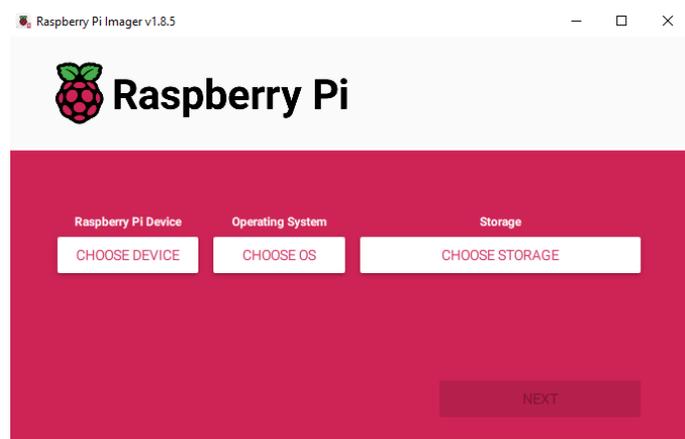
Basic Raspberry Pi Kit

1. Gathering Physical Materials:

- Raspberry Pi
- MicroSD Card
- Power Adapter
- Ethernet cable OR wireless adapter
- Computer with MicroSD reader

2. Setting Up Raspberry Pi

This section will focus on getting the Raspberry Pi or device of your choosing ready for hosting. After these steps, a default webpage will be available from the device's local IP address.



Raspberry Pi Imager

- a. Download latest version of Raspberry Pi Imager
- b. Choose your Raspberry pi device
- c. Choose 32-bit or 64-bit OS depending on Pi
- d. Continue following the Imager
- e. Insert the microSD into the Raspberry Pi
- f. Power up the Raspberry Pi
- g. Connect the Pi to the internet (Wired or Wireless)
- h. Log into the system
RECOMMENDED: TO ADD A LAYER OF SECURITY, CHANGE DEFAULT PASSWORD FROM "raspberry" TO SOMETHING ELSE USING "passwd"
- i. Update the system with "sudo apt-get update && sudo apt-get upgrade"

```
micah@raspberrypi:~ $ micah@raspberrypi:~ $ passwd
Changing password for micah.
Current password:
New password:
Retype new password:
passwd: password updated successfully
micah@raspberrypi:~ $ sudo apt-get update && sudo apt-get upgrade
Get:1 http://archive.raspberrypi.org/debian bullseye InRelease [23.6 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf Packages [13.2 MB]
Get:4 http://archive.raspberrypi.org/debian bullseye/main armhf Packages [313 kB]
Fetched 13.6 MB in 10s (1,399 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  base-files bind9-host bind9-libs chromium-browser chromium-codecs-ffmpeg-extra cifs-utils
  distro-info-data gstreamer1.0-plugins-bad libde265-0 libglib2.0-0 libglib2.0-bin libglib2.0-data libgnutls30
  libgstreamer-plugins-bad1.0-0 libmariadb3 libnftables1 libperl5.32 libzbar0 libzmq5 mariadb-client-10.5
  mariadb-client-core-10.5 mariadb-common mariadb-server mariadb-server-10.5 mariadb-server-core-10.5 nftables perl
  perl-base perl-modules-5.32 tar tzdata usb.ids
33 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 153 MB of archives.
After this operation, 8,293 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Terminal View of updating system

- j. Choose webhosting software
Note: Apache and Nginx are two popular servers. For this guide, we will be using Nginx, but Apache is great as well.

k. Install software with “sudo apt-get install nginx”

Note: At this point, you should be able to connect to your web server by entering the device’s local IP address in a URL bar. There will be a default page created by Nginx.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Default NGINX page

3. Port Forwarding

Now that we have set up the web server, you might start to wonder how to access it outside of your own home network. Port forwarding will allow you to access your site from outside your local network.

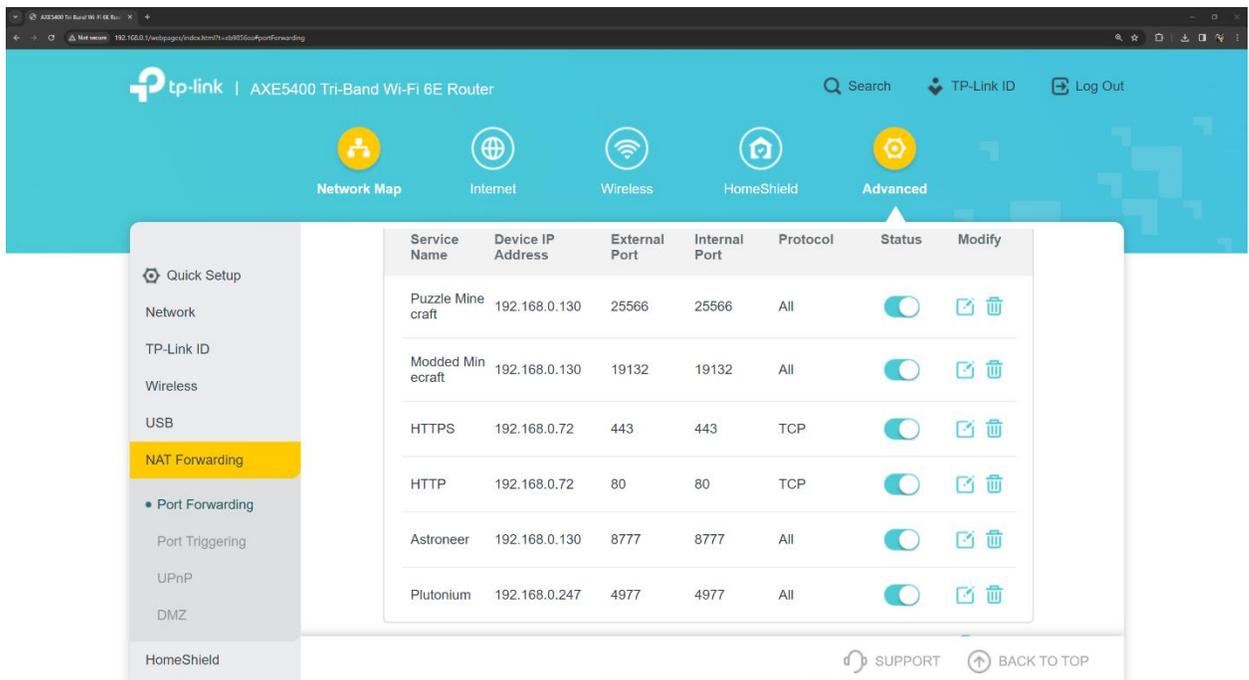
a. Find your router’s IP address and access its online settings page.

Note: A common address for router’s would be “192.168.0.1”

b. Locate the port forwarding section. It will usually be under advanced settings

c. Forward both Port 80 and port 443 to the local address of your hosting device using TCP

WARNING: FORWARDING NETWORK PORTS CAN ALLOW UNSOLICITED REQUESTS TO YOUR NETWORK. NEVER OPEN MORE PORTS THAN NEEDED.



Possible layout of Port Forwarding Page

4. Buying a Domain

From here, you will now be able to access your site from an external network. A potentially disappointing aspect of its current state, however, is that you must type your public IP instead of having a normal “.com” site. Setting up will address this potential problem.

- a. Choose a domain registrar and select an available domain
- b. Purchase selected domain name
- c. Wait for DNS propagation

Note: Use a site such as “whatmydns.net” to verify your domain name is active.

5. Using Cloudflare

Now you have a domain name, you have a web server, and you’ve port forwarded your hosting device. It is time to put everything together. Cloudflare is a service that adds extra security. It allows you to securely encrypt the traffic that flows through your web services. It provides a security certificate that allows you to access a site as HTTPS as opposed to HTTP.

- a. Sign up for a Cloudflare account
- b. From the Cloudflare dashboard, select “Add a site.”
- c. Enter the domain name you selected from before.
- d. Once the domain has been added, find the DNS options.
- e. Find the Cloudflare name servers.
- f. Go back to your domain registrar and add Cloudflare’s name servers.
- g. Finally, add CNAME or A DNS record which points to your public IP address

Type ▲	Name	Content	Proxy status	TTL	Actions
④ CNAME	██████████	████████████████████	🍷 Proxied	Auto	Edit ▶

CNAME Record Pointing to Public IP Address or DDNS

RECOMMENDED: Under “SSL/TLS” settings, set encryption mode to “Flexible.” Additionally, under “Edge Certificates”, enable the “Always Use HTTPS.” Both of these options will increase security.

NOTE: It can take up to 24 hours for certificate to be assigned



1Example of HTTPS Icon

6. What's Next?

Congratulations on successfully setting up your own web services. Now that your webservices are up and running, the possibilities are endless. From hosting blogs to sharing files, development programming projects, or even creating a personalized email address, you have the freedom to explore your online presence. From here, it's up to you as to what you want to use it for. A popular use of domains is to make a custom email. This can be done by adding MX record in Cloudflare and then configuring email routing rules. This allows you to have an email address that looks like your domain. Ex. `yourname@yourdomain.com`.

Custom address	Action	Destination	Status
micah@[REDACTED]	Send to an email	micahpucchioball@gmail.com	<input checked="" type="checkbox"/> Active Edit