

Raspberry Pi Setup, Ad Blocker and Website Hosting on DarkNet

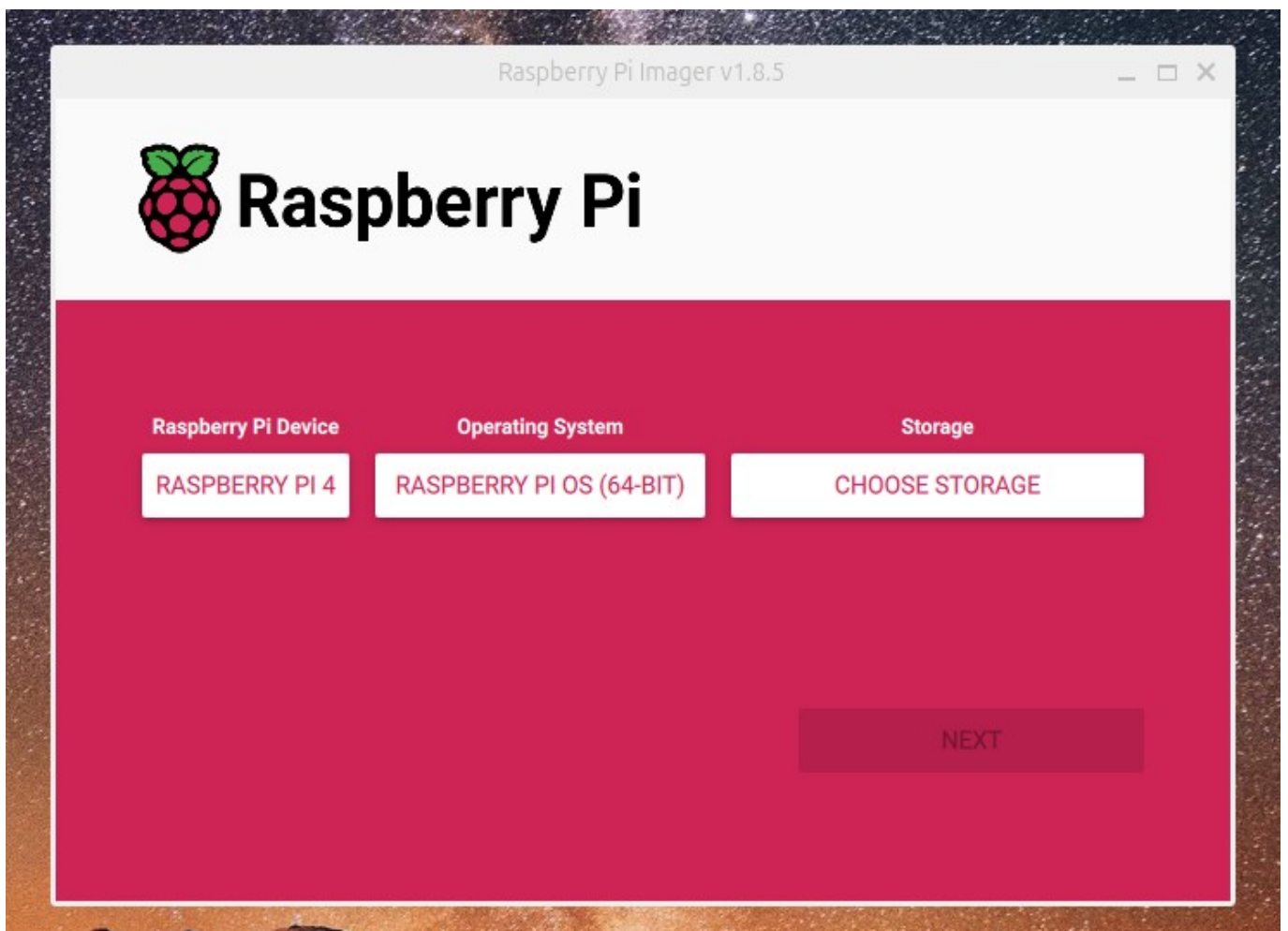
Lia Potikyan

Raspberry Pi Setup

First, download the Raspberry Pi imager from the official website:

<https://www.raspberrypi.com/software/>

Install the Raspberry Pi OS onto your MicroSD card, don't forget to enable SSH and enter the SSID and your wifi's password in custom settings!



Insert the MicroSD card into your Raspberry Pi and power it on(5W/3A).

sudo netdiscover -r [your ip range]

```
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 3 hosts. Total size: 186
-----
IP                At MAC Address      Count    Len  MAC Vendor / Hostname
-----
192.168.11.254    f8:22:29:c8:e4:40    2       102  Nokia Shanghai Bell Co., Ltd
192.168.11.50     e6:be:eb:2a:f3:3f    1        42  Unknown vendor
192.168.11.54     d8:3a:dd:7d:5c:45    1        42  Raspberry Pi Trading Ltd
```

Now when we have it's IP address, let's log in via SSH.

ssh pi@192.168.11.54

```
heisenberg@192.168.11.54's password:
Linux raspberrypi 6.6.20+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.20-1+rpt1 (2024
03-07) 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.
Last login: Wed Jun 12 10:04:10 2024
heisenberg@raspberrypi:~ $
```

sudo apt update

sudo apt install realvnc-vnc-server

sudo systemctl enable vncserver-x11-serviced

sudo systemctl start vncserver-x11-serviced

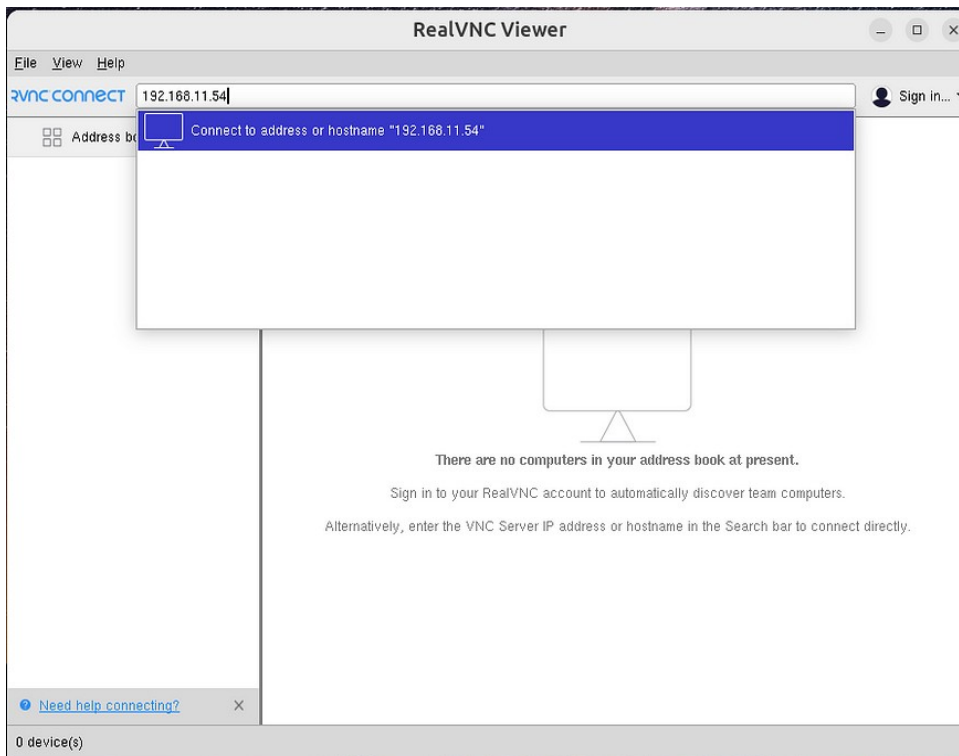
After this:

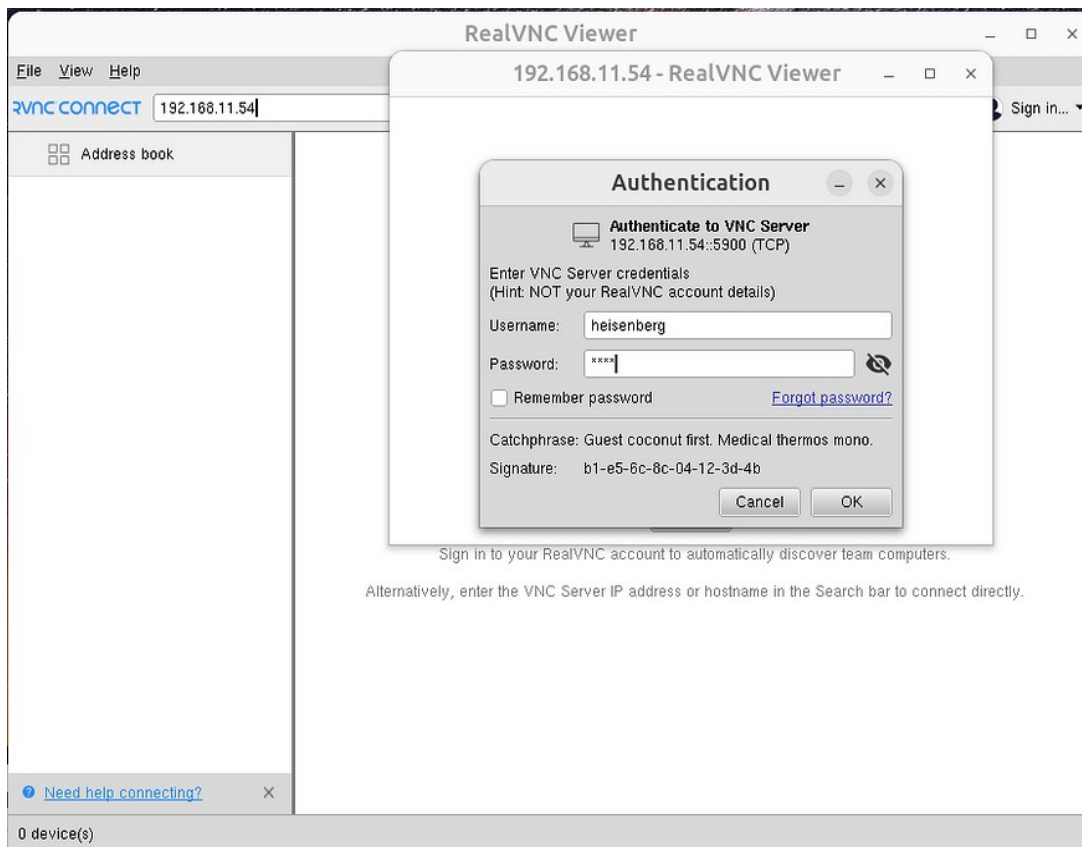
sudo raspi-config

Go to **Interface Options** → **VNC** → **Yes**

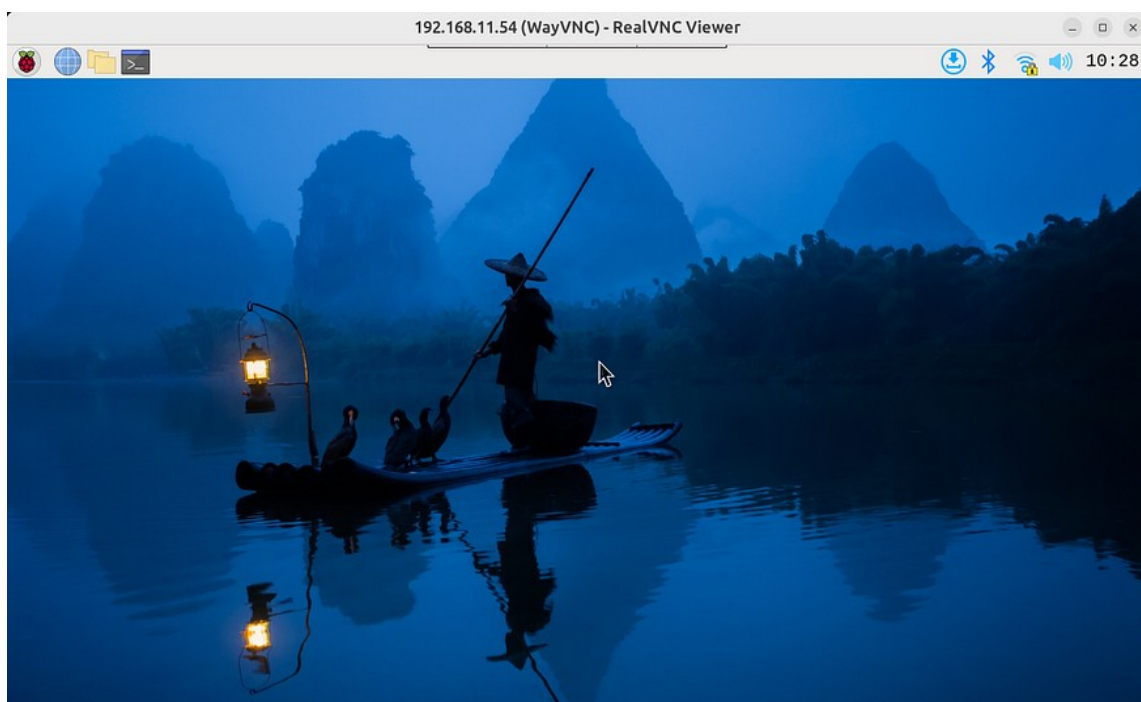
Exit and Reboot.

Open VNC Viewer on your laptop and enter the IP address of your Raspberry Pi:





Enter your Username and Password:



Ad Blocker

This command will start the automatic installer:

The image displays two sequential terminal window screenshots from a Raspberry Pi, showing the installation of Pi-hole. The terminal window has a title bar with the text "heisenberg@raspberrypi: ~" and a menu bar with "File Edit Tabs Help".

Top Screenshot:

```
heisenberg@raspberrypi:~$ curl -sSL https://install.pi-hole.net | bash
```

[i] Root user check
 [i] **Script called with non-root privileges**
 The Pi-hole requires elevated privileges to install and run
 Please check the installer for any concerns regarding this requirement
 Make sure to download this script from a trusted source

[✓] Sudo utility check
 [✓] Root user check

Progress bar (approx. 10% complete):

```

      .;.;.;
    .cccccc;.
    :cccclll:.
    :ccccclll. ;ooode
    'ccll;ll .oooodc
    .;ccl.;;looo:.
  
```

Bottom Screenshot:

```
heisenberg@raspberrypi:~$
```

[✓] FTL is listening on port
 [✓] UDP (IPv4)
 [✓] TCP (IPv4)
 [✓] UDP (IPv6)
 [✓] TCP (IPv6)

[i] Pi-hole blocking will be enabled
 [i] Enabling blocking
 [✓] Reloading DNS lists
 [✓] Pi-hole Enabled
 [i] Web Interface password: **VWJA7qZr**
 [i] This can be changed using 'pihole -a -p'

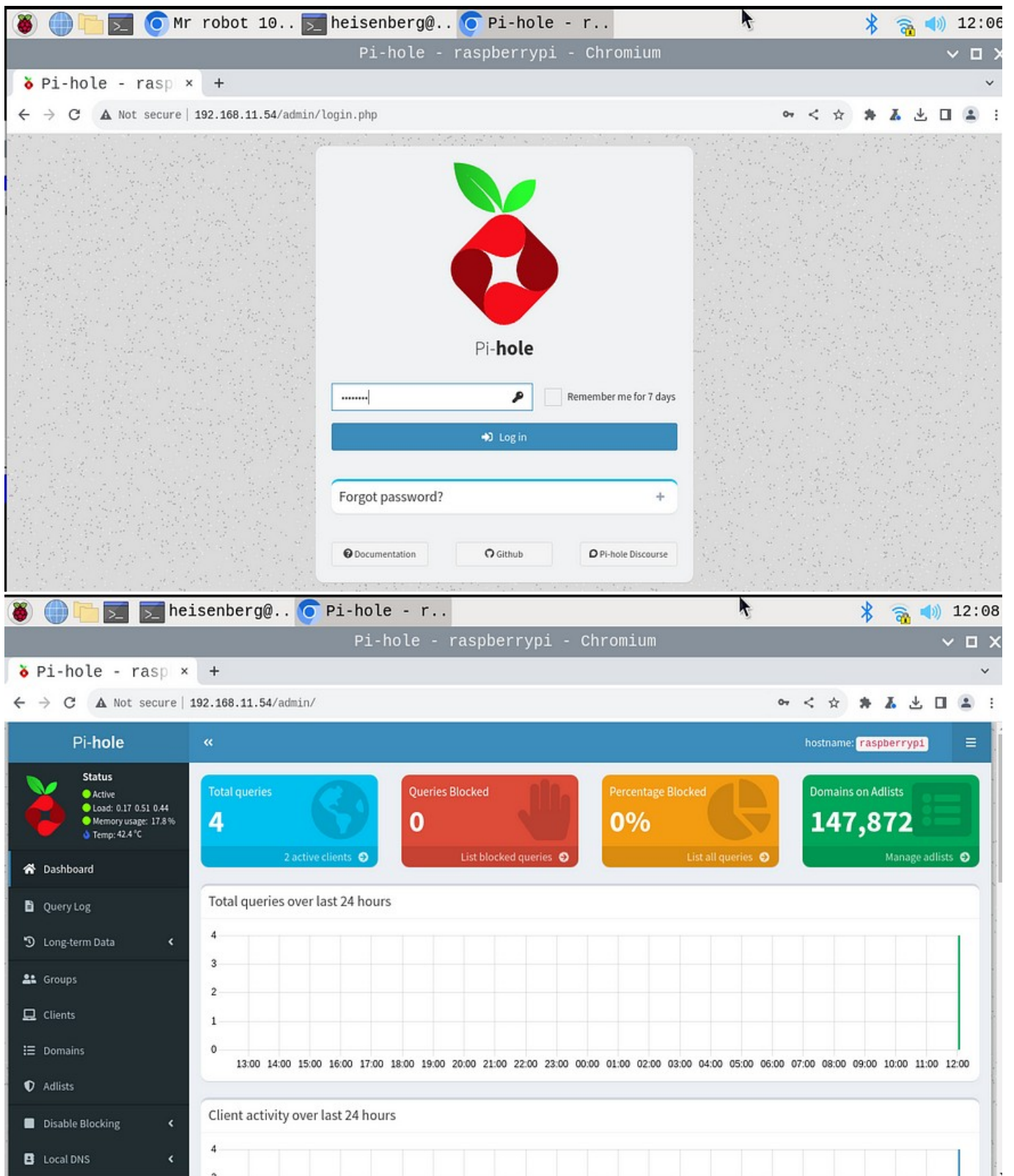
[i] View the web interface at <http://pi.hole/admin> or <http://192.168.11.54/admin>

[i] You may now configure your devices to use the Pi-hole as their DNS server
 [i] Pi-hole DNS (IPv4): 192.168.11.54
 [i] Pi-hole DNS (IPv6): fd11:5ee:bad:c0de::a91:2901
 [i] If you have not done so already, the above IP should be set to static.

[i] The install log is located at: /etc/pihole/install.log
 [✓] **Installation complete!**

heisenberg@raspberrypi:~\$

```
<ip>/admin
```



And here is the admin web interface!

Now let's configure our laptop to use PiHole.

Set the DNS Server to your Raspberry Pi's IP- replace the "YOUR_CONNECTION_NAME" with your wifi's SSID and "YOUR_PIHOLE_IP" to your Pi's IP address:

```
nmcli con mod "YOUR_CONNECTION_NAME" ipv4.dns  
"YOUR_PIHOLE_IP"  
nmcli con mod "YOUR_CONNECTION_NAME"  
ipv4.ignore-auto-dns yes  
nmcli con up "YOUR_CONNECTION_NAME"
```

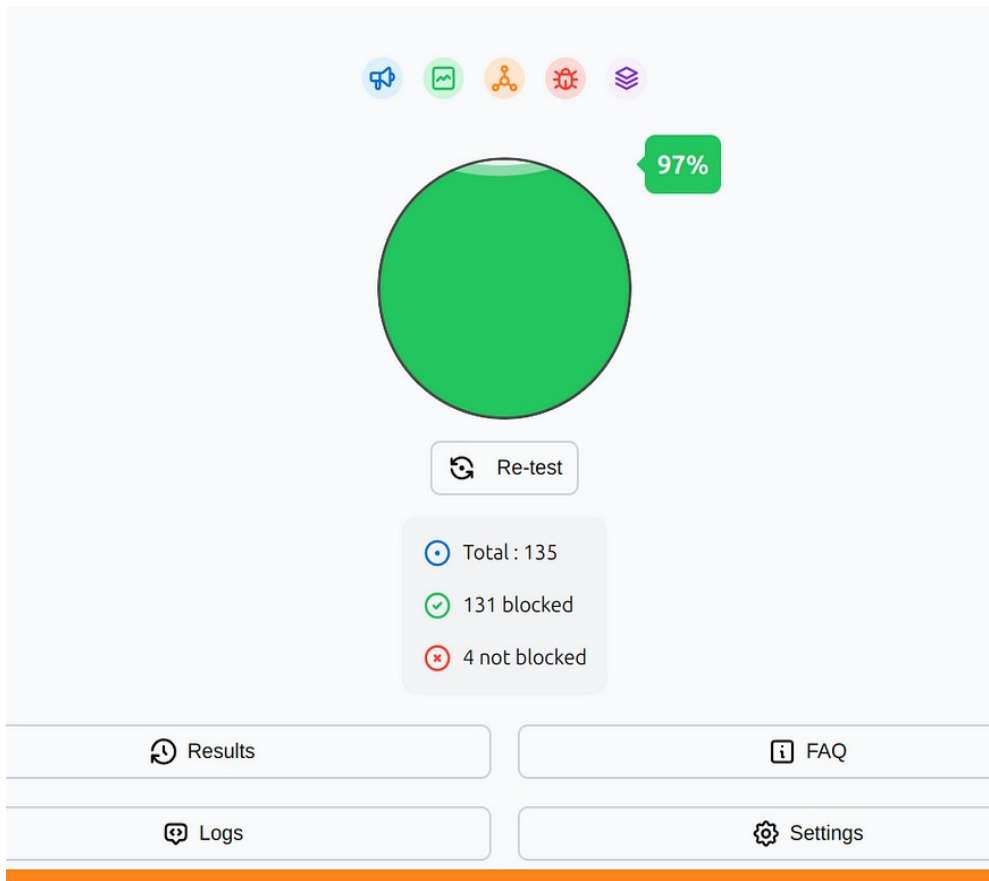
Enter this command and you should see the IP of your Raspberry:

```
nmcli dev show | grep DNS
```

A terminal window with a dark background and green text. The prompt is 'heisenberg@MSI:~\$'. The command 'nmcli dev show | grep DNS' has been entered. The output shows 'IP4.DNS[1]: 192.168.11.54' and 'IP6.DNS[1]: fe80::1'. The prompt 'heisenberg@MSI:~\$' is repeated at the bottom.

```
heisenberg@MSI:~$ nmcli dev show | grep DNS  
IP4.DNS[1]: 192.168.11.54  
IP6.DNS[1]: fe80::1  
heisenberg@MSI:~$
```

Now let's test the Ad Bloacker:



Raspberry Pi Website Hosting on DarkNet

First, we need to install Apache and Tor.

Apache is a web server software that we will use to host are website, and we employ Tor to make our website accessible on the DarkNet.

```
sudo apt install apache2
```

```
sudo apt install tor
```

Open the Tor configuration file:

```
sudo nano /etc/tor/torrc
```

Add the following lines:

```
HiddenServiceDir /var/lib/tor/hidden_service/
```

```
HiddenServicePort 80 127.0.0.1:80
```



```
192.168.11.54 (WayVNC) - RealVNC Viewer
heisenberg@..
heisenberg@raspberrypi: /etc/ssl/certs
File Edit Tabs Help
GNU nano 7.2 /etc/tor/torrc
##
#ExitPolicy accept *:6660-6667,reject *:.* # allow irc ports but no more
#ExitPolicy accept *:119 # accept nntp as well as default exit policy
#ExitPolicy reject *:.* # no exits allowed

## Bridge relays (or "bridges") are Tor relays that aren't listed in the
## main directory. Since there is no complete public list of them, even an
## ISP that filters connections to all the known Tor relays probably
## won't be able to block all the bridges. Also, websites won't treat you
## differently because they won't know you're running Tor. If you can
## be a real relay, please do; but if not, be a bridge!
#BridgeRelay 1
## By default, Tor will advertise your bridge to users through various
## mechanisms like https://bridges.torproject.org/. If you want to run
## a private bridge, for example because you'll give out your bridge
## address manually to your friends, uncomment this line:
#PublishServerDescriptor 0

HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:80

```

Restart Tor:

```
sudo systemctl restart tor
```

And get your onion address:

```
sudo cat /var/lib/tor/hidden_service/hostname
```

```
192.168.11.54 (WayVNC) - RealVNC Viewer
heisenberg@..
heisenberg@raspberrypi: /etc/ssl/certs
File Edit Tabs Help
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for man-db (2.11.2-2) ...
heisenberg@raspberrypi:/etc/ssl/certs $ sudo nano /etc/tor/torrc
heisenberg@raspberrypi:/etc/ssl/certs $ sudo cat /var/lib/tor/hidden_service/hostname
cb3dby6zdgqtauhnschpw6rov4jsdvotmpndf6wxrlz7bxtjiczzpqyd.onion
heisenberg@raspberrypi:/etc/ssl/certs $
```

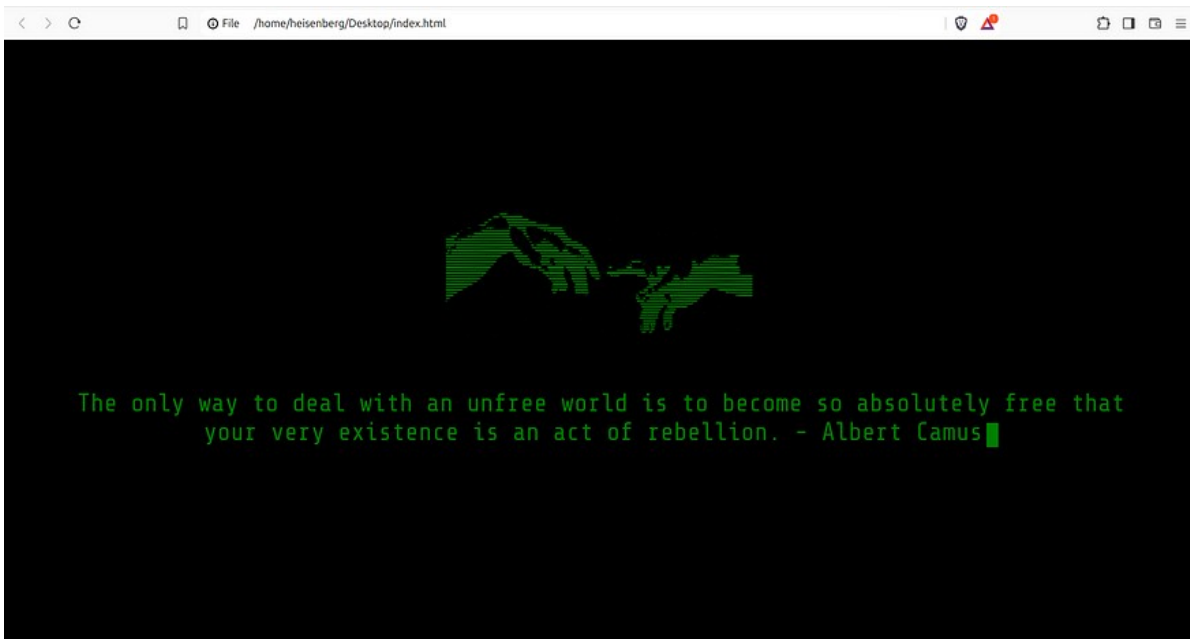
Now let's enable and start our Apache:

```
sudo systemctl enable apache2
```

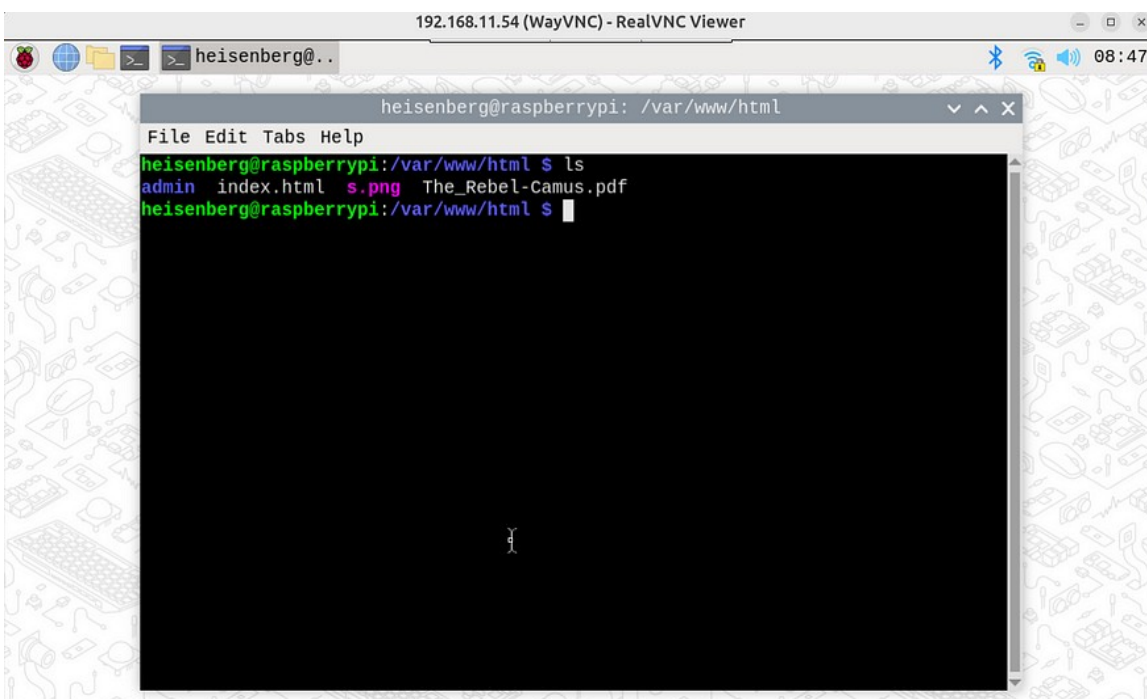
```
sudo systemctl start apache2
```

COOL!

Okay, now it's time for you to make your website you wanna host.



After that we need to get the necessary files in the **/var/www/html** directory of the Raspberry Pi (or your other hosting machine).



Now hit the .onion link into your Tor browser and you should see your website!

