The World's Greatest Pi-hole (and Unbound) Tutorial 2023

Welcome! In this article, I am going to detail for you what I consider to be the perfect Pi-hole setup instructions for 2023 (yes, I know – as of the writing of this article, it's still 2022, but we're close enough).

Table of Contents

Video Version of this Tutorial

World's Greatest Pi-hole Tutorial - Easy Raspberry Pi Project!

WTF is a Pi-hole?

First of all though – what is Pi-hole? Pi-hole is a network-wide ad blocker designed to be run on a Raspberry Pi single-board computer. When Pi-hole is installed, and your computers and devices are configured to use it for their DNS queries, ads and malware are blocked automatically in order to reduce the chances of being tracked, or having malware installed. This increases both security and privacy on your network overall.

“I cAn’T gEt a RaSpBeRrY pI!!”

> Yes yes – supply chain issues make it extremely difficult to get your hands on a Raspberry Pi at the moment. But don’t let that stop you! You can also run Pi-Hole in a Docker container (for instance if you have a QNAP or Synology NAS with the Docker app installed), or any one of a number of other operating systems. Don’t let lack of Raspberry Pi availability stop you from doing this project – it’s extremely lightweight, and can be installed on a virtual machine with very little resources allocated.
>
> Another excellent resource is to follow RPilocator on Twitter, or keep an eye on their web page which can send alerts when a Raspberry Pi comes into stock in your geographical location. I used RPilocator to get an 8GB Raspberry Pi 4 recently, and would have been able to get even MORE Raspberry Pi’s if not for stock limits. (Damn you Adafruit! <shakes fist>).

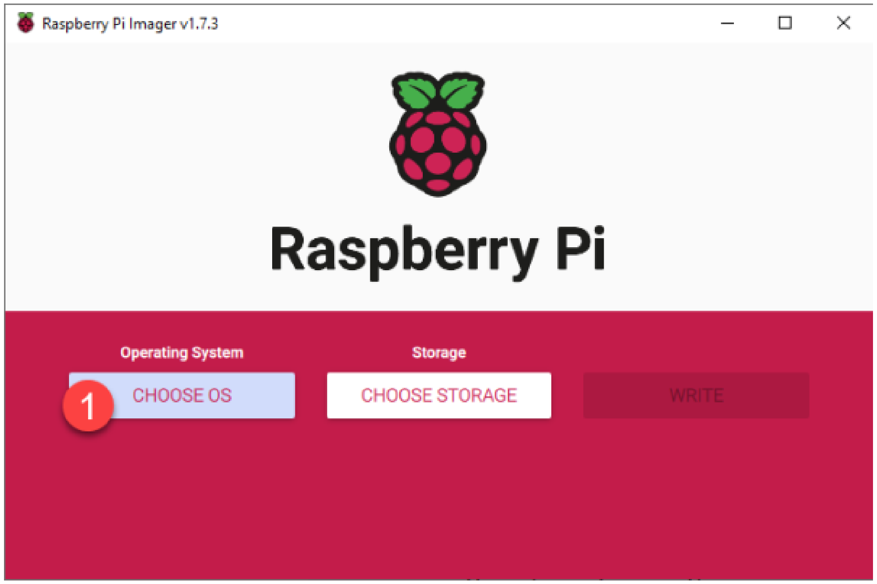
Let’s get started!

Here’s what you need to get started:

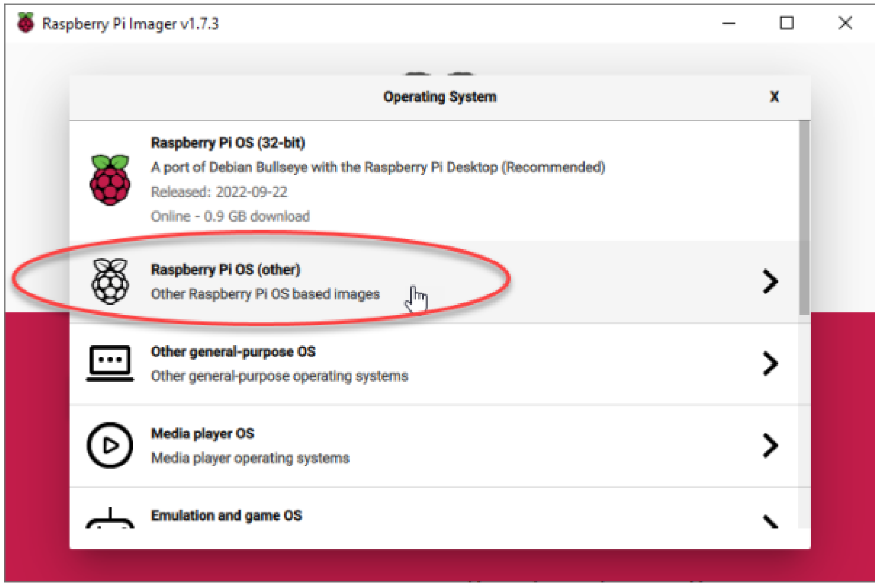
- Raspberry Pi – I’m using a Raspberry Pi 2GB for this project, but even that version is total overkill – Pi-hole works great on older units as well if you have any Raspberry Pi 2’s or 3’s lying around. I wouldn’t recommend trying this on any of the Pi Zeros though since it’s much better to have this hard-wired. We won’t be using any of the wireless capabilities of the Pi.
- A microSD card – Does not have to be a big card – 8GB is totally fine. When finished, Raspberry Pi OS + Pi-hole will end up taking up about 1.6GB on the microSD card. I would recommend a decent quality card though – don’t want anything to slow down those DNS lookups!
- Raspberry Pi Imager – This software formats the microSD card and installs your Raspberry Pi OS of choice. It also allows you to set some of the Raspberry Pi defaults before you ever boot it up. Raspberry Pi Imager is available for download on Windows, macOS, and Ubuntu.

Raspberry Pi Imager

Download and run Raspberry Pi Imager on your desktop computer. You'll also want to insert the microSD card into a microSD card reader. Take note of the drive letter of the microSD card – but no need to format it ahead of time as Raspberry Pi imager will do this for you.



Underneath 'Operating System' click 'CHOOSE OS.' Then click on Raspberry Pi OS (other) and choose 'Raspberry Pi OS Lite (64-bit).'



Raspberry Pi Imager v1.7.3

Operating System

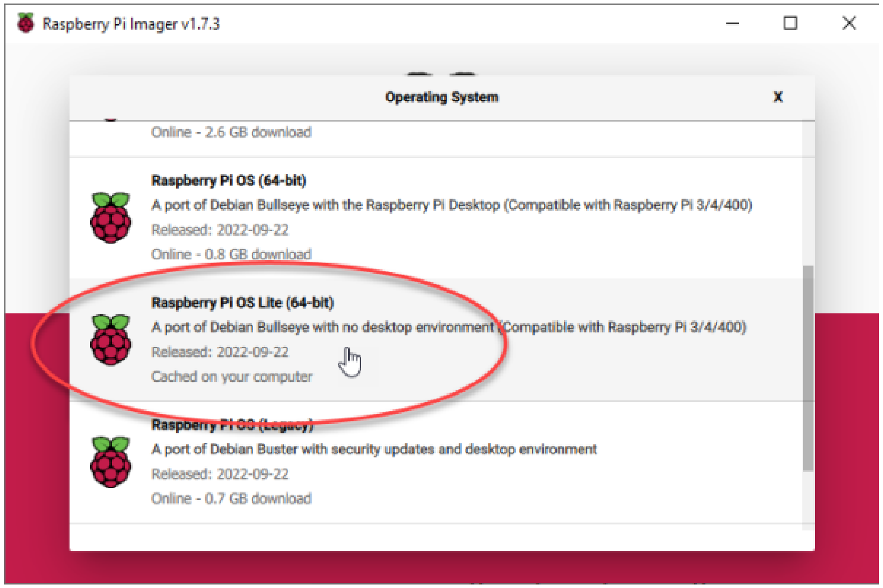
Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Released: 2022-09-22
Online - 0.9 GB download

Raspberry Pi OS (other)
Other Raspberry Pi OS based images

Other general-purpose OS
Other general-purpose operating systems

Media player OS
Media player operating systems

Emulation and game OS



Raspberry Pi Imager v1.7.3

Operating System

Online - 2.6 GB download

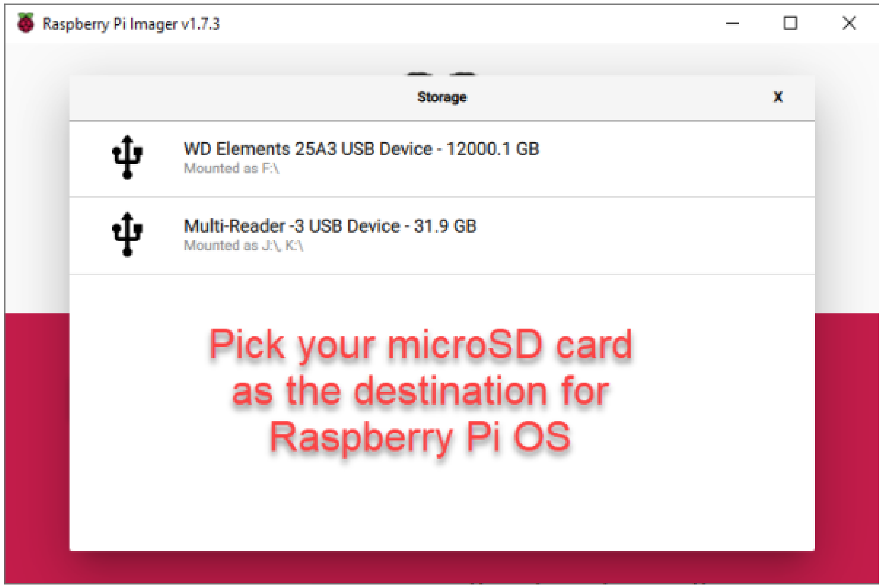
Raspberry Pi OS (64-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Compatible with Raspberry Pi 3/4/400)
Released: 2022-09-22
Online - 0.8 GB download

Raspberry Pi OS Lite (64-bit)
A port of Debian Bullseye with no desktop environment (Compatible with Raspberry Pi 3/4/400)
Released: 2022-09-22
Cached on your computer

Raspberry Pi OS (Legacy)
A port of Debian Buster with security updates and desktop environment
Released: 2022-09-22
Online - 0.7 GB download

We don't need any sort of GUI or extra software here, so the Lite version of the Raspberry Pi OS is going to be just fine.

Next, click 'CHOOSE STORAGE.' You'll want to select your microSD card.



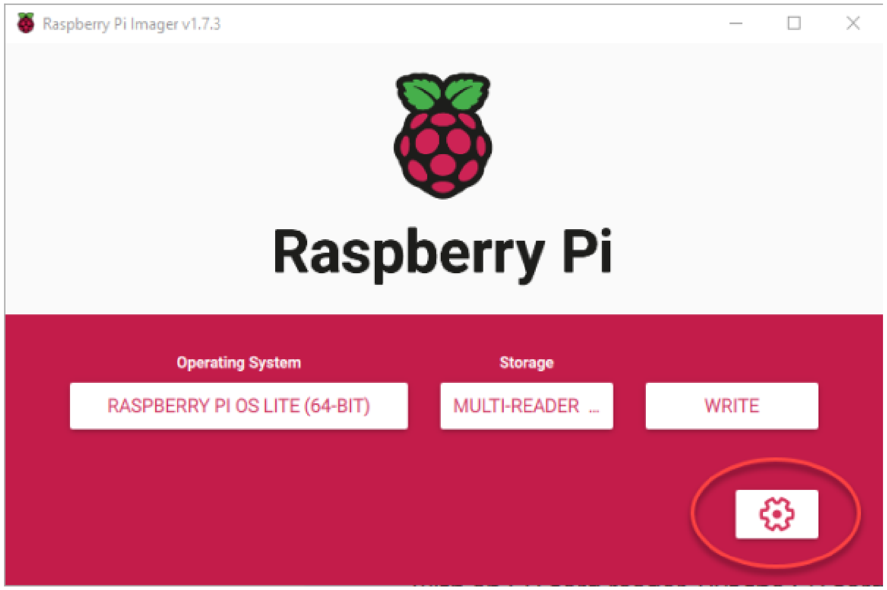
WD Elements 25A3 USB Device - 12000.1 GB
Mounted as F:\

Multi-Reader -3 USB Device - 31.9 GB
Mounted as J:\, K:\

Pick your microSD card as the destination for Raspberry Pi OS

MAKE SURE you don't overwrite any drive that ISN'T the microSD card!! This is the only warning! If you pick the wrong drive and erase Grandma's 90th birthday pictures, you will never get them back.

Finally, click the gear icon so that we can set some of the Raspberry Pi default settings.



Operating System: RASPBERRY PI OS LITE (64-BIT)

Storage: MULTI-READER ...

Check the box next to 'Set hostname:' and pick a name. I'm going to call mine pihole3.local (since I already have 2 Pi-holes on my network).

Check the box next to 'Enable SSH' and then leave the radio button on 'Use password authentication.'

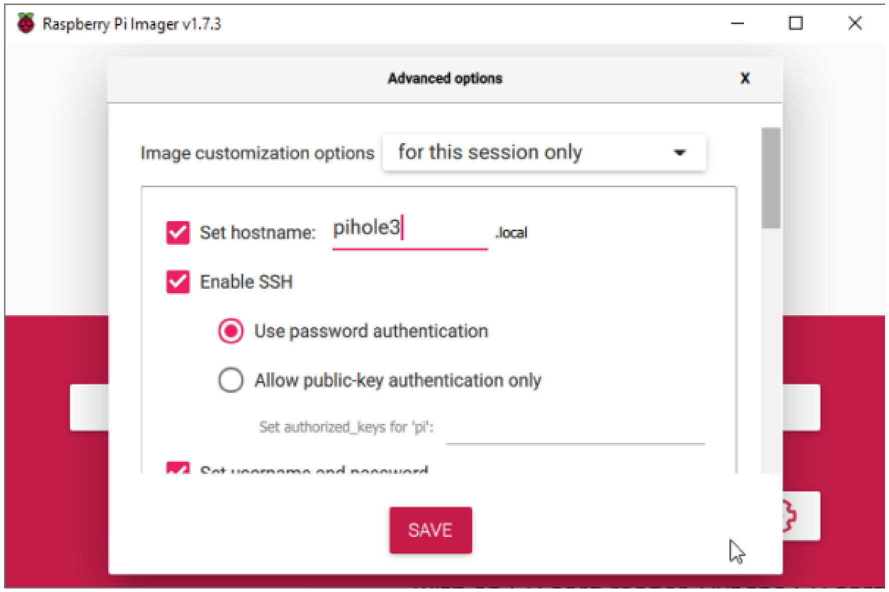


Image customization options: for this session only

☑ Set hostname: pihole3 .local

☑ Enable SSH

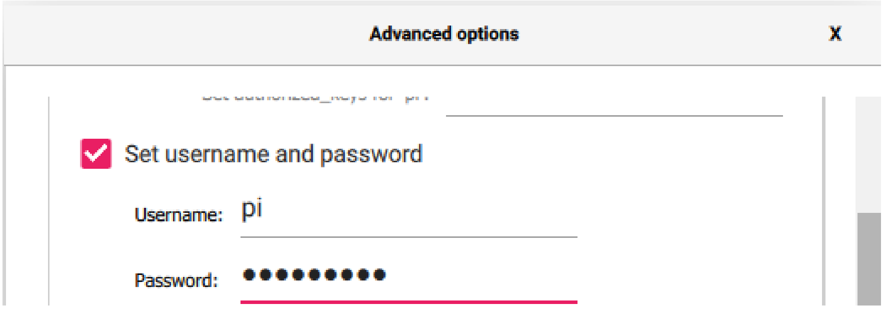
● Use password authentication

○ Allow public-key authentication only

Set authorized_keys for 'pi':

☑ Set username and password

Scroll down and set a username and password for your Pi-hole SSH. I'm going to leave 'pi' as the username, but set a strong password.



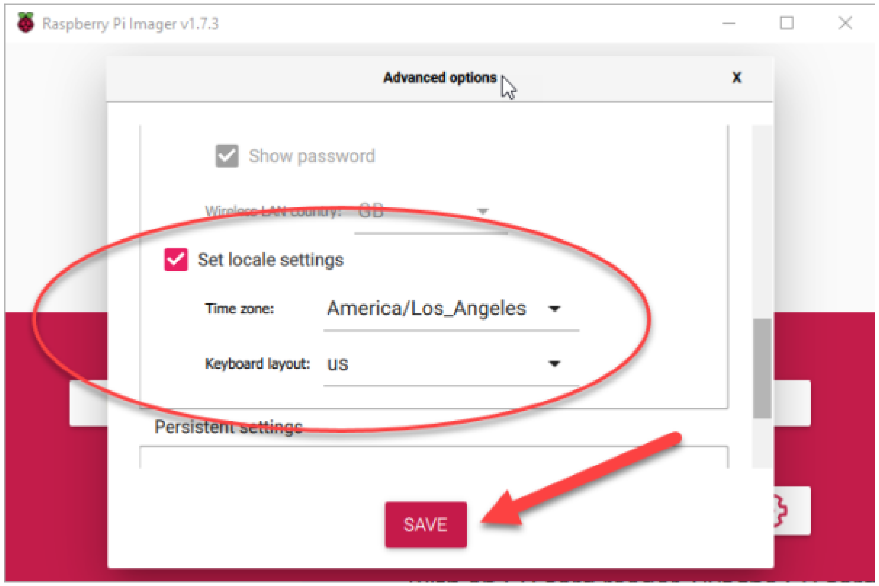
Set authorized_keys for 'pi':

☑ Set username and password

Username: pi

Password: ●●●●●●●●●●

Optionally, you can scroll further down and check the box next to 'Set locale settings' and then pick your time zone and keyboard layout. When finished, click 'SAVE.'



☑ Show password

Wireless LAN country: GB

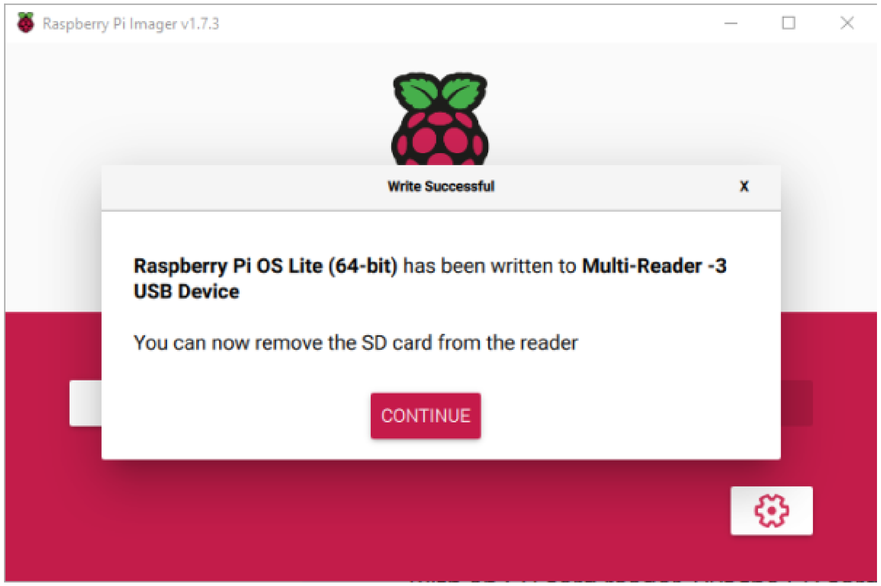
☑ Set locale settings

Time zone: America/Los_Angeles

Keyboard layout: US

Persistent settings

Finally, click 'WRITE' and your brand new Raspberry Pi OS will be written to the microSD card. This takes about 3-4 minutes. Once complete, remove the microSD card and insert it into your Raspberry Pi.



Raspberry Pi OS Lite (64-bit) has been written to **Multi-Reader -3 USB Device**

You can now remove the SD card from the reader

Boot your Pi-hole

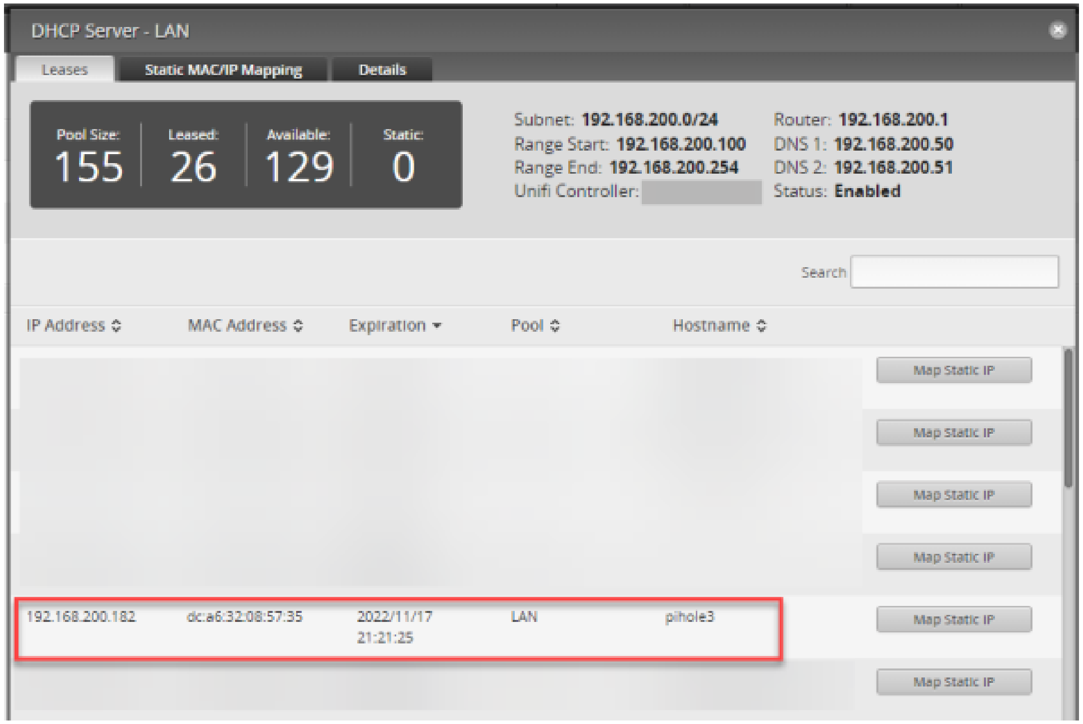
Plug power into your Raspberry Pi to boot it up – the boot process should only take a minute or so. Other than the power cord, you should also have the Pi's Ethernet port plugged into a network switch.

> PRO TIP: I have found that when working with Raspberry Pi's it's nice to have an on/off switch. This product works great for being able to easily switch the power on and off of the Raspberry Pi without having to disconnect the USB-C or MicroUSB cable.

PISwitch (USB-C)

Once your Raspberry Pi has booted, it will likely grab an IP address from your network's DHCP server – but how will you know which IP address it grabbed? There are two main ways to do this.

Option 1: You can check your DHCP lease table in your router. I have an EdgeRouter 4, so I log into the router and navigate to the Services –> DHCP tab. From there, I can drop down the 'Actions' box to view DHCP leases for my LAN network. If I sort by the most recent leases, I can see Hostname 'pihole3' (exactly as I set in the Raspberry Pi Imager above). This tells me that the 'pihole3' host received 192.168.200.182 from the DHCP server.



DHCP Server - LAN

Pool Size: 155 | Leased: 26 | Available: 129 | Static: 0

Subnet: 192.168.200.0/24
Range Start: 192.168.200.100
Range End: 192.168.200.254
Unifi Controller: [redacted]
Router: 192.168.200.1
DNS 1: 192.168.200.50
DNS 2: 192.168.200.51
Status: **Enabled**

IP Address	MAC Address	Expiration	Pool	Hostname	
[redacted]					Map Static IP Map Static IP Map Static IP Map Static IP
192.168.200.182	dc:a6:32:08:57:35	2022/11/17 21:21:25	LAN	pihole3	Map Static IP
[redacted]					Map Static IP

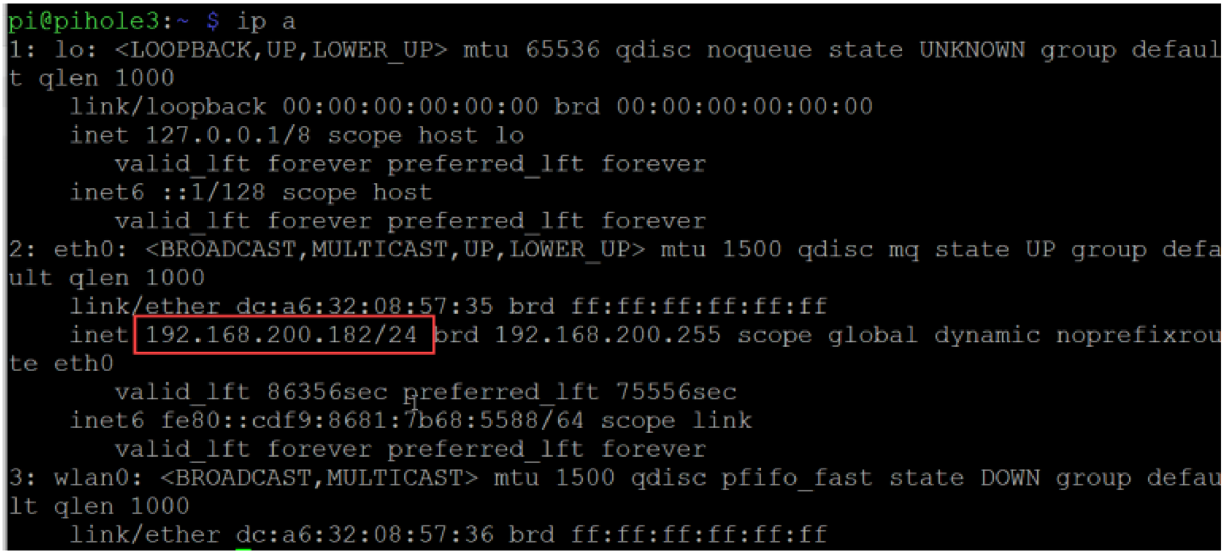
Finding the IP address of my Pi-hole upon first boot.

Now – if you don't have an EdgeRouter, whichever device you DO have that provides DHCP leases to your client devices will likely have a similar way to view these leases and figure out which IP address your Pi-hole has been given. If you're using UniFi or a Dream Machine / Dream Router, you can likely find this information in the 'Clients' section.

Option 2: If you don't have any way to check DHCP leases in your DHCP server, you can always hook up a keyboard and monitor to the Raspberry Pi, log in, and then run:

```
ip a
```

To get the IP address information.



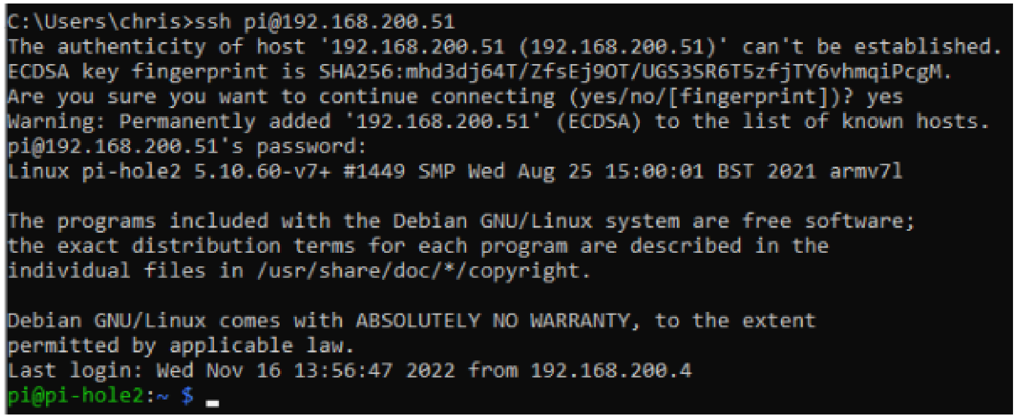
```
pi@pihole3:~ $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group defa
ult qlen 1000
    link/ether dc:a6:32:08:57:35 brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.182/24 brd 192.168.200.255 scope global dynamic noprefixrou
te eth0
       valid_lft 86356sec preferred_lft 75556sec
    inet6 fe80::cdf9:8681:7b68:5588/64 scope link
       valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN group defau
lt qlen 1000
    link/ether dc:a6:32:08:57:36 brd ff:ff:ff:ff:ff:ff
```

SSH into the Pi

Once you have your Raspberry Pi's IP address, use SSH to log in. On Windows, you can SSH from the command prompt by typing:

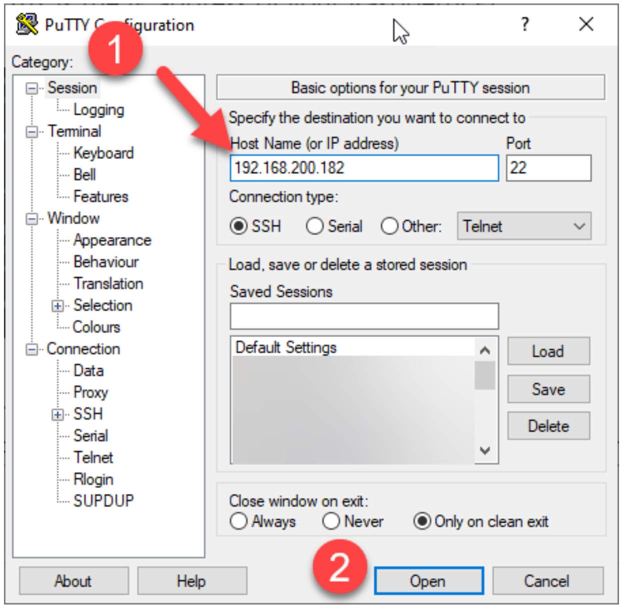
```
ssh pi@192.168.200.182
```

Where 'pi' is your username, and 192.168.200.182 is the IP address of your Raspberry Pi.



```
C:\Users\chris>ssh pi@192.168.200.51
The authenticity of host '192.168.200.51 (192.168.200.51)' can't be established.
ECDSA key fingerprint is SHA256:mhd3dj64T/ZfsEj9OT/UGS3SR6T5zfjTY6vhmqiPcgM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.200.51' (ECDSA) to the list of known hosts.
pi@192.168.200.51's password:
Linux pi-hole2 5.10.60-v7+ #1449 SMP Wed Aug 25 15:00:01 BST 2021 armv7l

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 Nov 16 13:56:47 2022 from 192.168.200.4
pi@pi-hole2:~ $
```

Alternatively, you can use a program such as PuTTY to log in via SSH. Just download and run PuTTY and then type your IP address in the Host Name field and click ‘Open.’



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.200.182

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Use PuTTY to log in.

Finally, on macOS, open ‘Terminal’ and then log in with:

```
ssh pi@192.168.200.182
```

Just like you would with the Windows command prompt.

Log in with username ‘pi’ and the password you set in the Raspberry Pi Imager.

Update Raspberry Pi OS

The first thing we want to do is update all packages. To do so, run this command:

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

Updating the Pi takes 4-5 minutes.

Set a Static IP Address

We want to ensure that the IP address of the Raspbery Pi never changes, so we need to set a static IP. There are two ways to do this.

Option 1 – Set a DHCP reservation

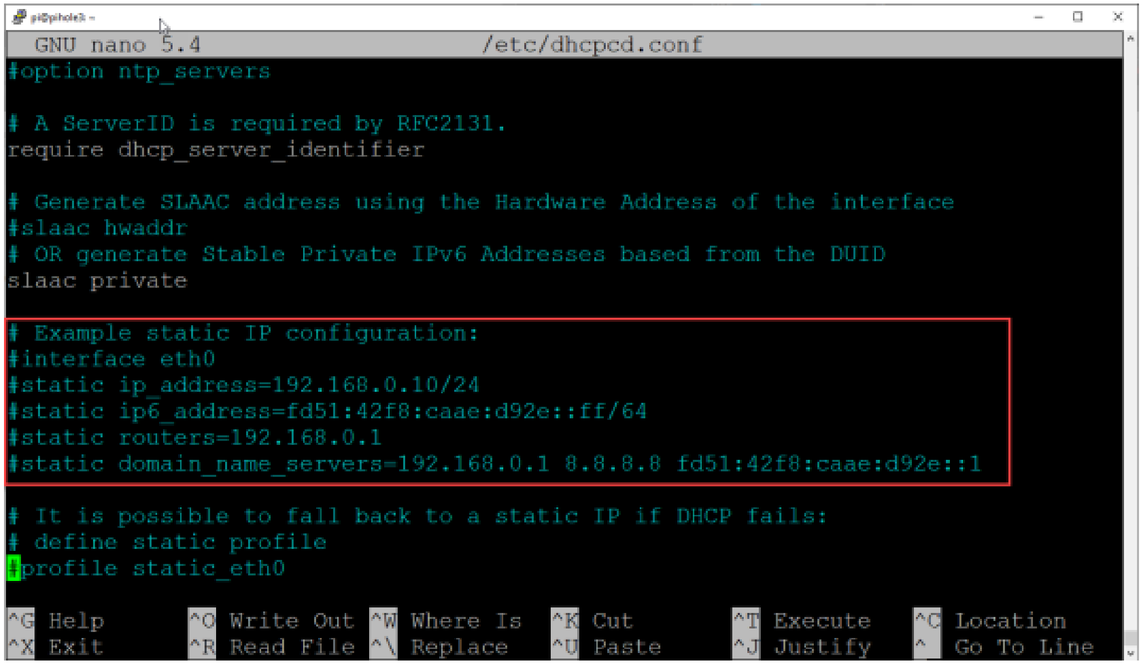
A DHCP reservation is basically just a way for us to tell the DHCP server to only ever give a specific IP address to the MAC address of our Raspberry Pi. In my EdgeRouter 4, this is pretty simple – I just find the DHCP lease for the Raspberry Pi, and then click the 'Map Static IP' button followed by 'Save.' Once I have done that, the DHCP server will only ever give out the IP address I picked to the Raspberry Pi. In the settings of the Pi itself, it's still set to obtain an IP address from DHCP, but my DHCP server knows that it should only ever give the Pi a specific IP…which essentially makes it static.

Option 2 – Set a Static IP on the Raspberry Pi

The other option is to set an IP address statically on the Pi itself – this is the method that I prefer. Open up the dhcpcd.conf file by typing:

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

And then scroll down until you find the section called 'Example Static IP configuration:'



```
  GNU nano 5.4                         /etc/dhcpcd.conf
#option ntp_servers

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0

^G Help         ^O Write Out ^W Where Is  ^K Cut         ^T Execute   ^C Location
^X Exit         ^R Read File ^\ Replace   ^U Paste       ^J Justify   ^_ Go To Line
```

This section is commented out by default, so you will need to uncomment some lines and input your own static IP address and network information. For my Pi-hole, I am going to want to use the following information:

Static IP: 192.168.200.52/24
Gateway: 192.168.200.1
DNS Servers: 192.168.200.1, 1.1.1.1

In my case the configuration will look like this:

```
# Example static IP configuration:
interface eth0
static ip_address=192.168.200.52/24
# static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.200.1
static domain_name_servers=192.168.200.1 1.1.1.1
```

Interface eth0 was uncommented, and then I set a static IPv4 IP address of 192.168.200.52 (don't use my network subnet information – substitute it with your own). Then I set 'static routers' to my gateway IP and set two DNS servers (I removed the IPv6 DNS server).

Once you're happy with your configuration, press CTRL+X to exit followed by 'Y' and ENTER to save.

Now reboot the Pi to take advantage of the new network settings:

```
sudo reboot
```

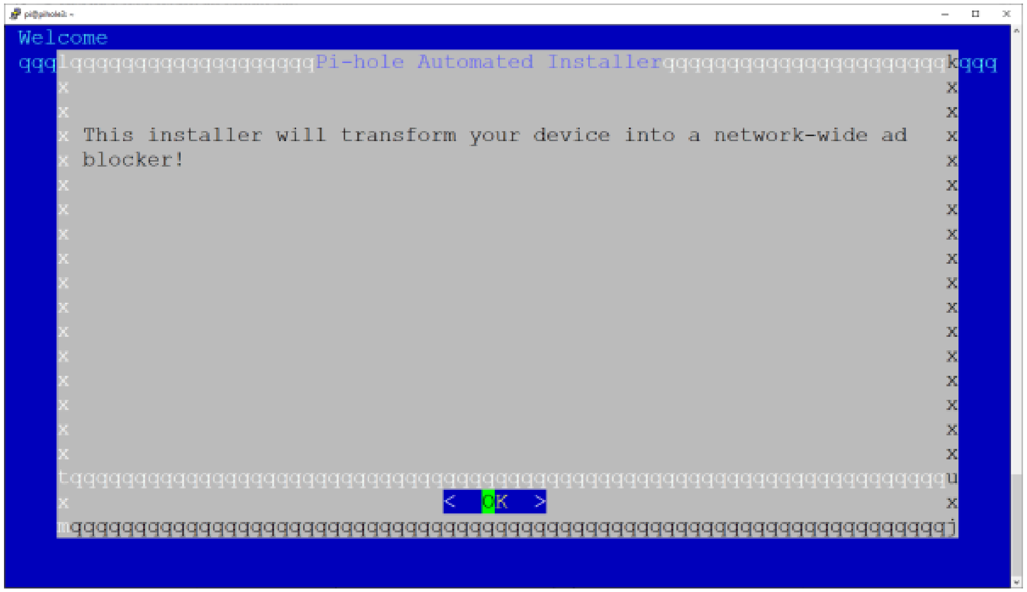
Install Pi-hole

Once you've set your Pi-hole's networking to a static IP address and rebooted (or if you've done a DHCP reservation), it's time to install Pi-hole. Remember to re-connect to the NEW IP address if you set it statically on the Raspberry Pi.

Run this command to install Pi-hole:

```
curl -sSL https://install.pi-hole.net | bash
```

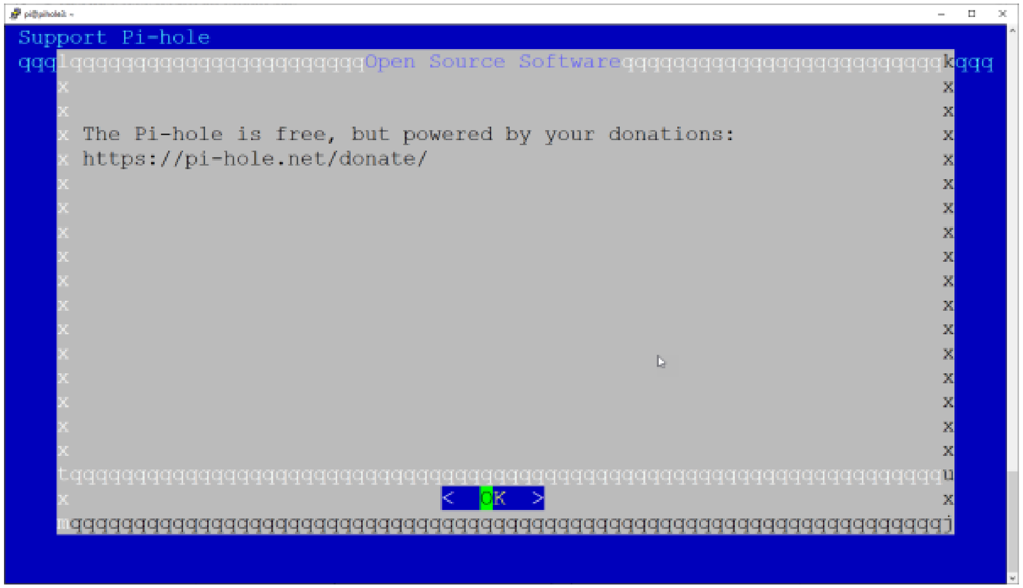
This will fire off the installation wizard. When the screen below appears, press ENTER (OK).



```
Welcome
qqqlqqqqqqqqqqqqqqqqqqqPi-hole Automated Installerqqqqqqqqqqqqqqqqqqqqqkqqq

    This installer will transform your device into a network-wide ad
    blocker!

                              < OK >
```

Pi-hole installation wizard – press OK.

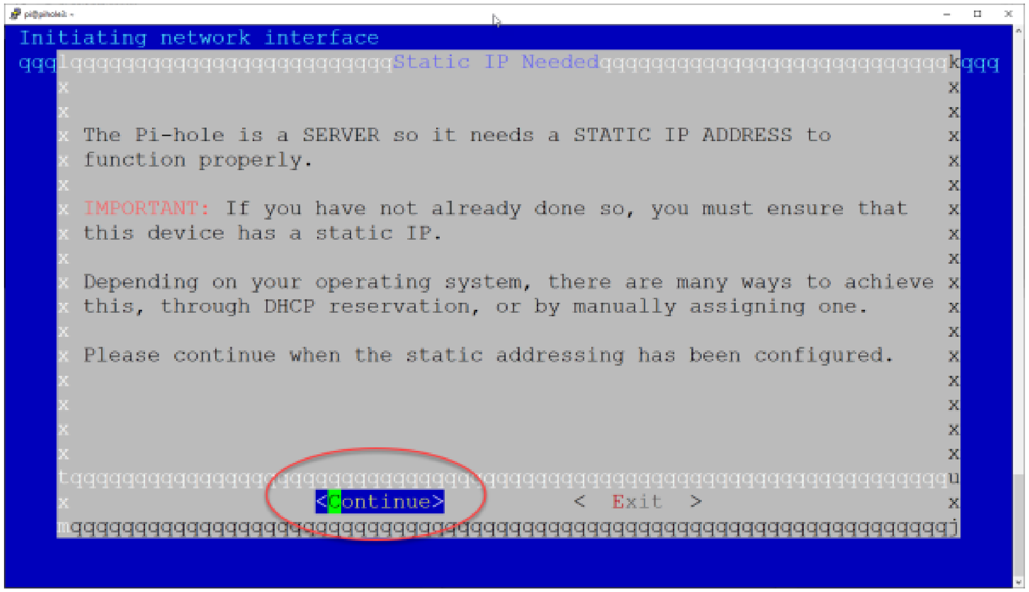
The next screen recommends that you DONATE to help the Pi-hole project – I recommend it as well! Press ENTER (OK).



```
Support Pi-hole
qqqlqqqqqqqqqqqqqqqqqqqqqqqOpen Source Softwareqqqqqqqqqqqqqqqqqqqqqqqqkqqq

    The Pi-hole is free, but powered by your donations:
    https://pi-hole.net/donate/

                              < OK >
```

Donate to this worth project! Press OK.

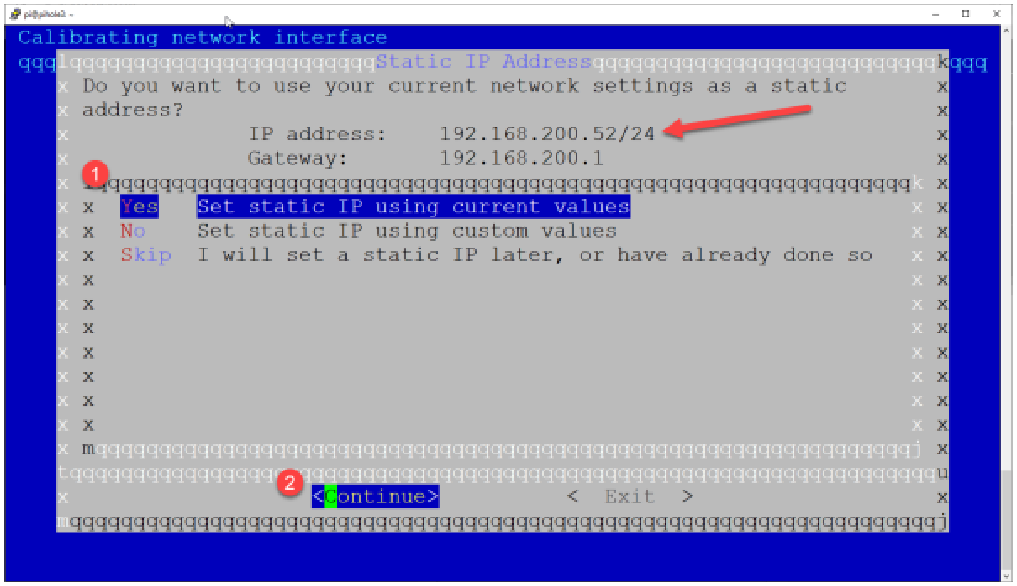
Next, the wizard gives us a warning about using a static IP address – we've already covered this, to move the cursor over to Continue and press ENTER.



```
Initiating network interface
                         Static IP Needed

 The Pi-hole is a SERVER so it needs a STATIC IP ADDRESS to
 function properly.

 IMPORTANT: If you have not already done so, you must ensure that
 this device has a static IP.

 Depending on your operating system, there are many ways to achieve
 this, through DHCP reservation, or by manually assigning one.

 Please continue when the static addressing has been configured.

                    <Continue>          <  Exit  >
```

Choose Continue and press OK.

On this screen, double-check that the IP address shown is the IP address that you want to use for the Pi-hole and then press ENTER (select Yes – Set static IP using current values).

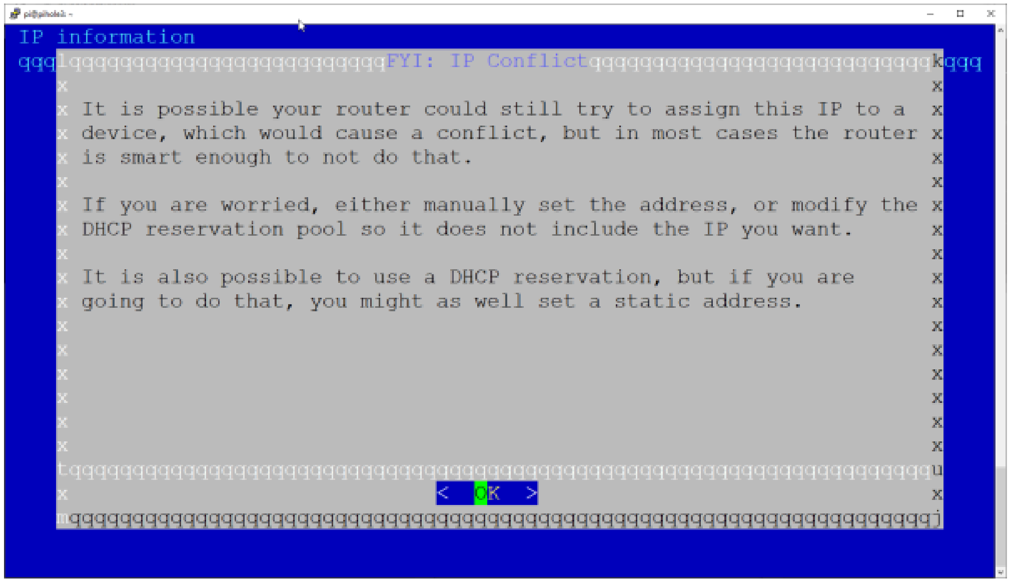


```
Calibrating network interface
                         Static IP Address
 Do you want to use your current network settings as a static
 address?
              IP address:    192.168.200.52/24
              Gateway:       192.168.200.1

   Yes    Set static IP using current values
   No     Set static IP using custom values
   Skip   I will set a static IP later, or have already done so

                    <Continue>          <  Exit  >
```

Choose Yes and press OK.

NOTE: If you set a specific static IP, but that is not displayed on this screen, you probably forgot to reboot the Raspberry Pi. Cancel out – reboot, and then re-start the installation wizard.

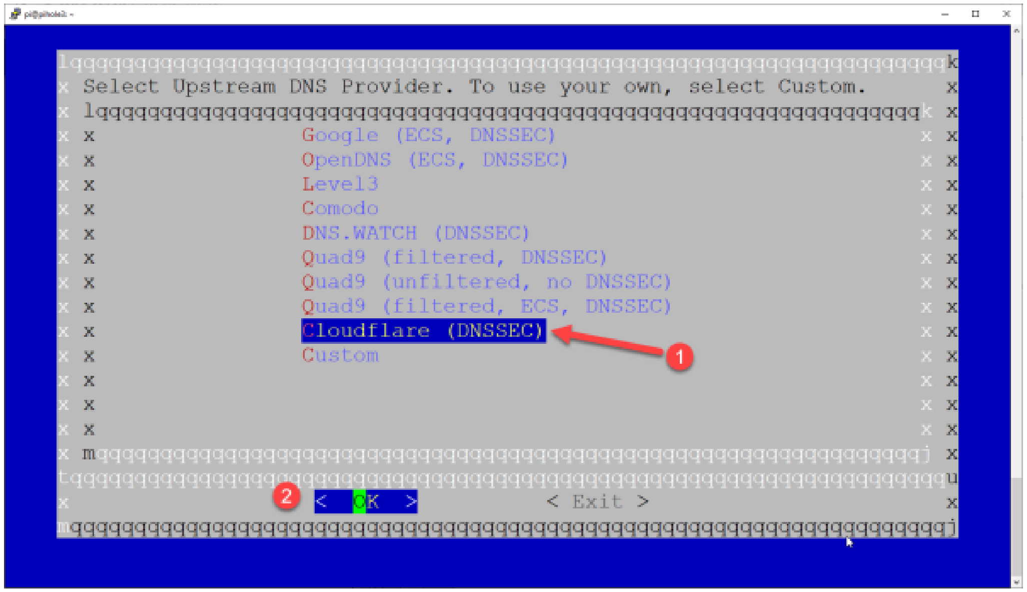
The next screen is another warning about networking – they REALLY REALLY want you to have a static IP address. Press ENTER (OK).



```
IP information
qqqlqqqqqqqqqqqqqqqqqqqqqqqqqqFYI: IP Conflictqqqqqqqqqqqqqqqqqqqqqqqqqqqkqqq
   x                                                                     x
   x It is possible your router could still try to assign this IP to a  x
   x device, which would cause a conflict, but in most cases the router x
   x is smart enough to not do that.                                     x
   x                                                                     x
   x If you are worried, either manually set the address, or modify the  x
   x DHCP reservation pool so it does not include the IP you want.       x
   x                                                                     x
   x It is also possible to use a DHCP reservation, but if you are       x
   x going to do that, you might as well set a static address.           x
   x                                                                     x
   x                                                                     x
   x                                                                     x
   x                                                                     x
   x                                                                     x
   x                                                                     x
   tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
   x                              <  OK  >                               x
   mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Press OK.

Now we are going to pick an upstream DNS provider. This is the DNS server that the Pi-hole will use to do DNS lookups (initial lookups – then they get cached). Since we’re going to be installing Unbound, we’ll be making our own DNS lookups to the primary root domain servers on the Internet, so for now, it’s fine to pick any one of these – I’ll be using Cloudflare (1.1.1.1) for this tutorial.

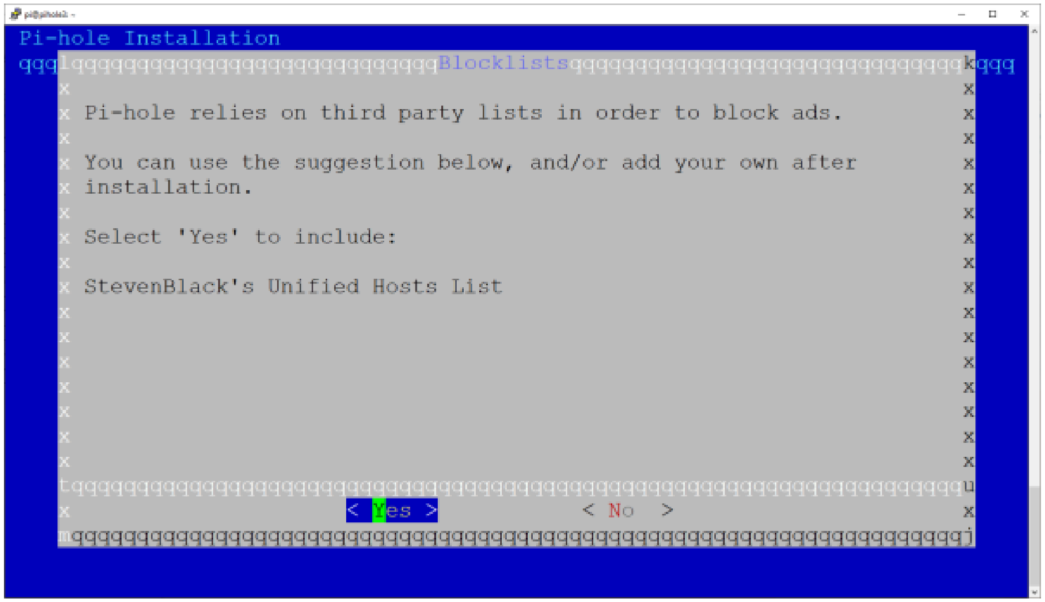


```
Select Upstream DNS Provider. To use your own, select Custom.

            Google (ECS, DNSSEC)
            OpenDNS (ECS, DNSSEC)
            Level3
            Comodo
            DNS.WATCH (DNSSEC)
            Quad9 (filtered, DNSSEC)
            Quad9 (unfiltered, no DNSSEC)
            Quad9 (filtered, ECS, DNSSEC)
            Cloudflare (DNSSEC)
            Custom

        <  OK  >            < Exit >
```

Choose an upstream DNS provider and press OK. I will be choosing Cloudflare, but we will be changing this setting later in the tutorial.

DNS Server Selection – You don't HAVE to use Unbound to perform your own lookups to the Internet's primary root domain servers – you an absolutely skip that part of this tutorial and simply pick one of the DNS servers on this list. BUT – if you do choose one of these DNS servers, you should understand that not all DNS servers are equal – some of them simply do any lookup that you request, and others do some level of content filtering. There's a great breakdown of the various DNS servers in this list (plus some more options), and which ones do phishing/malware/adult content filtering HERE.

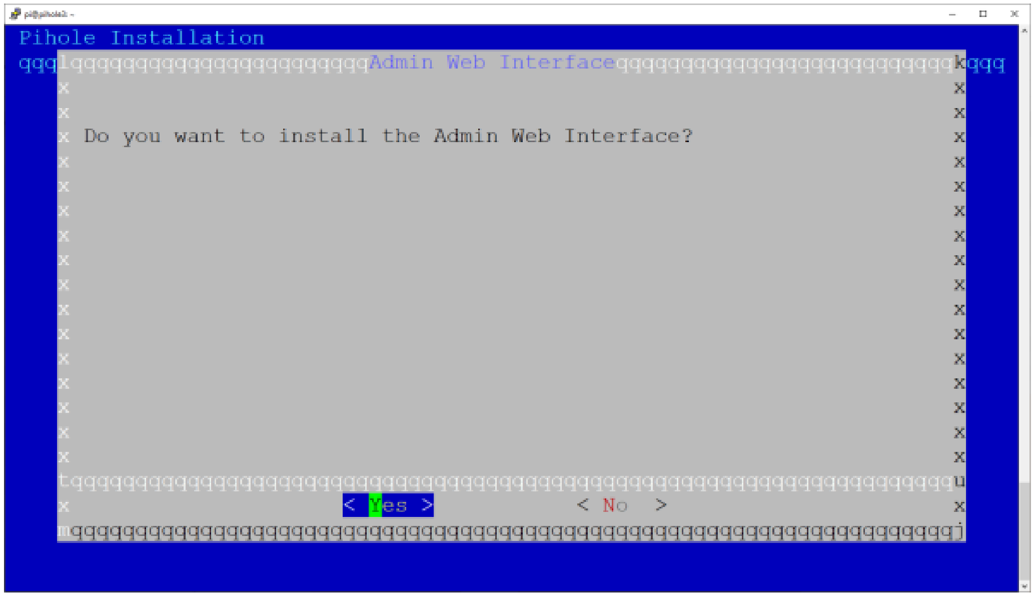
Next we're asked if we want to use the default included block list – this block list is a list of domains that Pi-hole is going to block for us. This list is perfectly fine, and will block a significant chunk of suspect sites – however, there are many block lists available, and you'll likely want to add some more – we'll cover this later in the tutorial. Choose YES and press ENTER.



```
Pi-hole Installation
                          Blocklists

 Pi-hole relies on third party lists in order to block ads.

 You can use the suggestion below, and/or add your own after
 installation.

 Select 'Yes' to include:

 StevenBlack's Unified Hosts List

                     < Yes >            < No  >
```

Choose Yes and press OK.

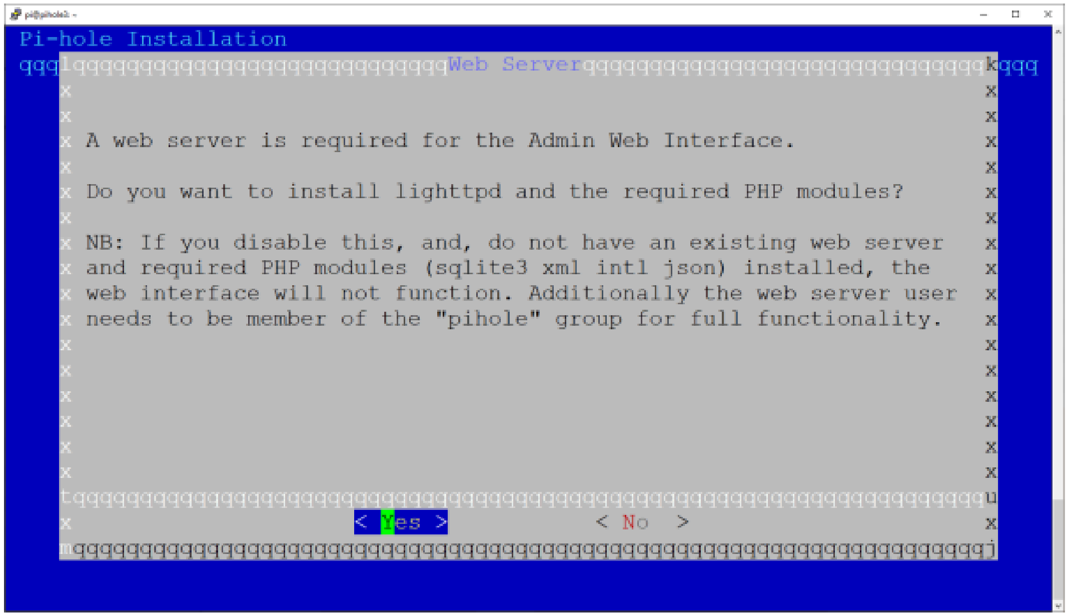
Next we’re asked if we want to install the Admin Web Interface, which of course we do! Select YES and press ENTER.



```
Pihole Installation
                    Admin Web Interface

 Do you want to install the Admin Web Interface?

                     < Yes >            < No  >
```

Choose Yes and press OK.

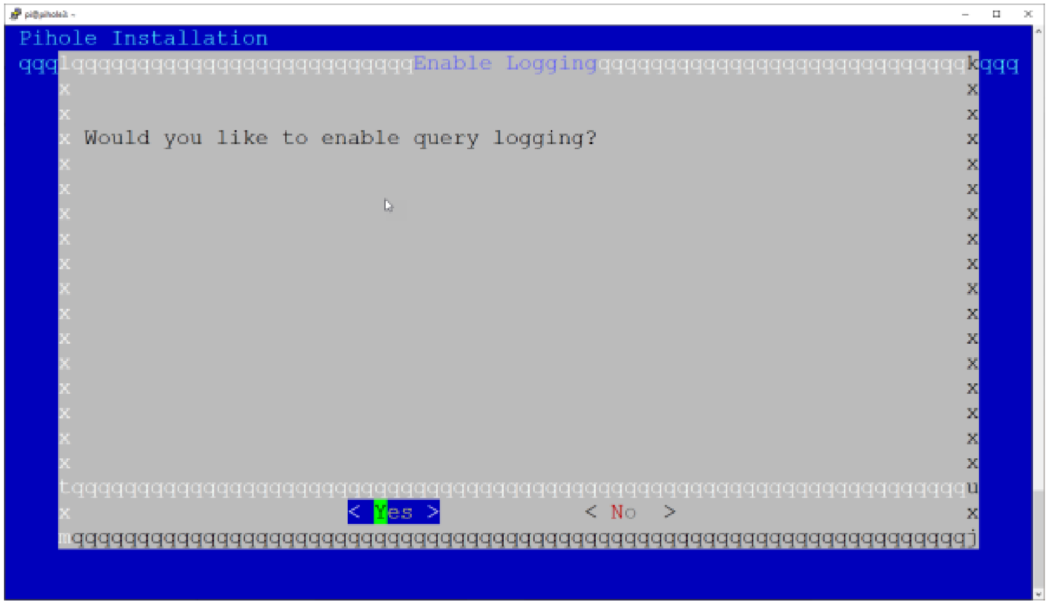
Another notice about the web server – just choose YES and press ENTER.



```
Pi-hole Installation
                                Web Server

 A web server is required for the Admin Web Interface.

 Do you want to install lighttpd and the required PHP modules?

 NB: If you disable this, and, do not have an existing web server
 and required PHP modules (sqlite3 xml intl json) installed, the
 web interface will not function. Additionally the web server user
 needs to be member of the "pihole" group for full functionality.

                        < Yes >            < No  >
```

Choose Yes and press OK.

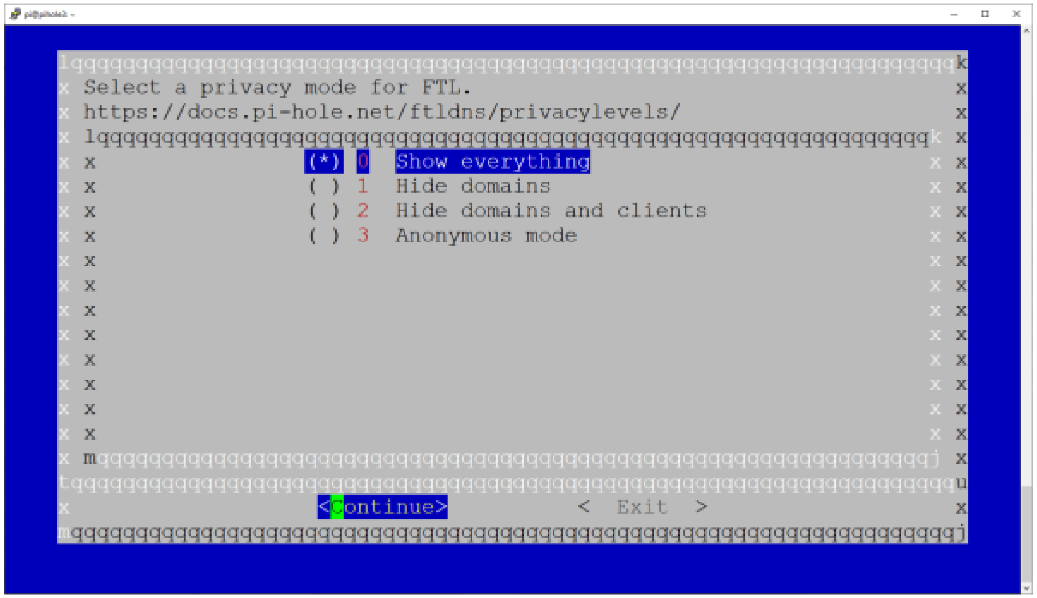
Next, we're asked if we want to enable query logging. Typically, you will want to say Yes here, but if you're super security conscious and you don't want any of your DNS lookups logged, you can also choose No.



```
Pihole Installation
                             Enable Logging

 Would you like to enable query logging?

                        < Yes >            < No  >
```

Choose Yes and press OK.

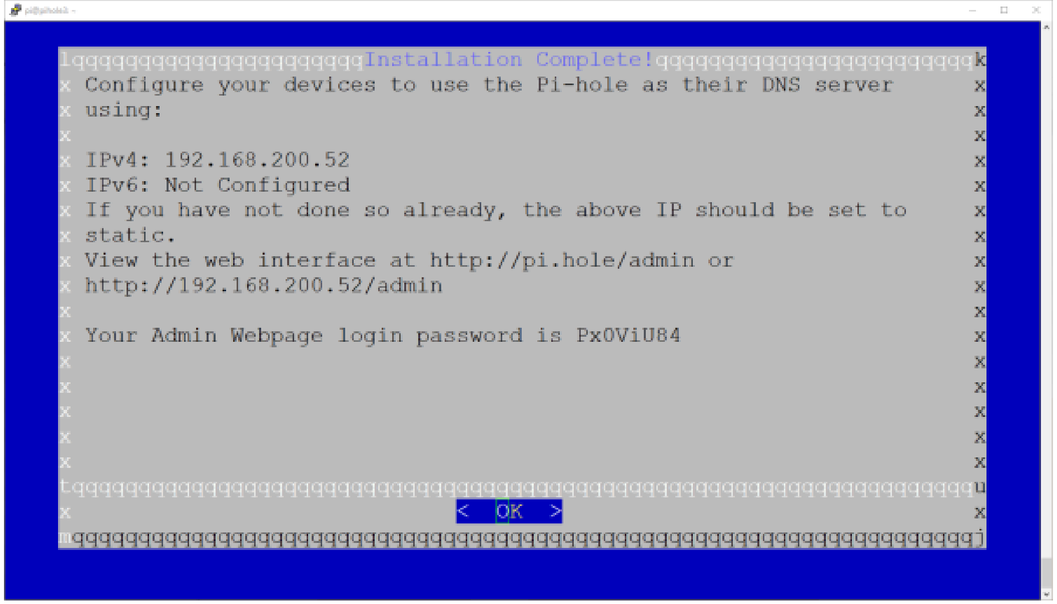
Now we’re asked about the level of privacy – you can visit the website listed for more information. For this tutorial, we’re going to select ‘Show everything’ and press Continue.



```
Select a privacy mode for FTL.
https://docs.pi-hole.net/ftldns/privacylevels/

                    (*) 0  Show everything
                    ( ) 1  Hide domains
                    ( ) 2  Hide domains and clients
                    ( ) 3  Anonymous mode

                    <Continue>          <  Exit  >
```

Leave this selected on Show everything and press Continue.

That’s it! After this step, Pi-hole has all of the information it needs to get started. You’ll see it run through a bunch of scripts in the CLI (takes about 2 minutes), and will eventually be brought to this screen:



```
Installation Complete!
Configure your devices to use the Pi-hole as their DNS server
using:

IPv4: 192.168.200.52
IPv6: Not Configured
If you have not done so already, the above IP should be set to
static.
View the web interface at http://pi.hole/admin or
http://192.168.200.52/admin

Your Admin Webpage login password is Px0ViU84

                              <  OK  >
```

Press OK to finish the installation.

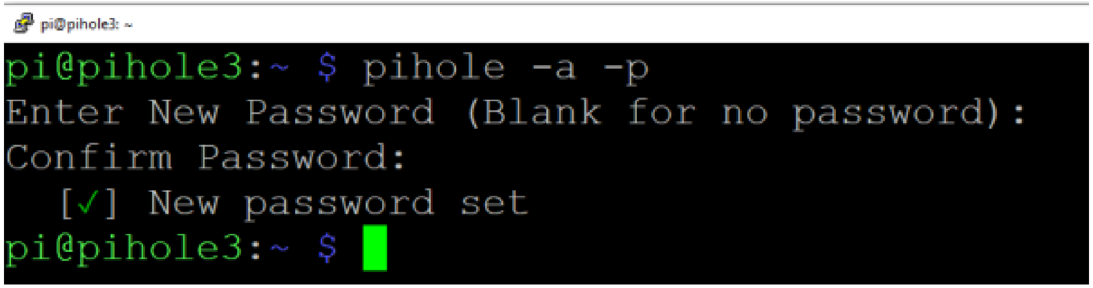
This final screen is giving a summary of our install and also shows us our Admin Webpage login password. You don't need to copy this down – we're going to change that password in the next step.

Change the Pi-hole Web GUI Admin Password

You'll want to set a nice strong password for the Pi-hole admin GUI – to do that, run this command:

```
pihole -a -p
```

This will have you input a password and then confirm it.



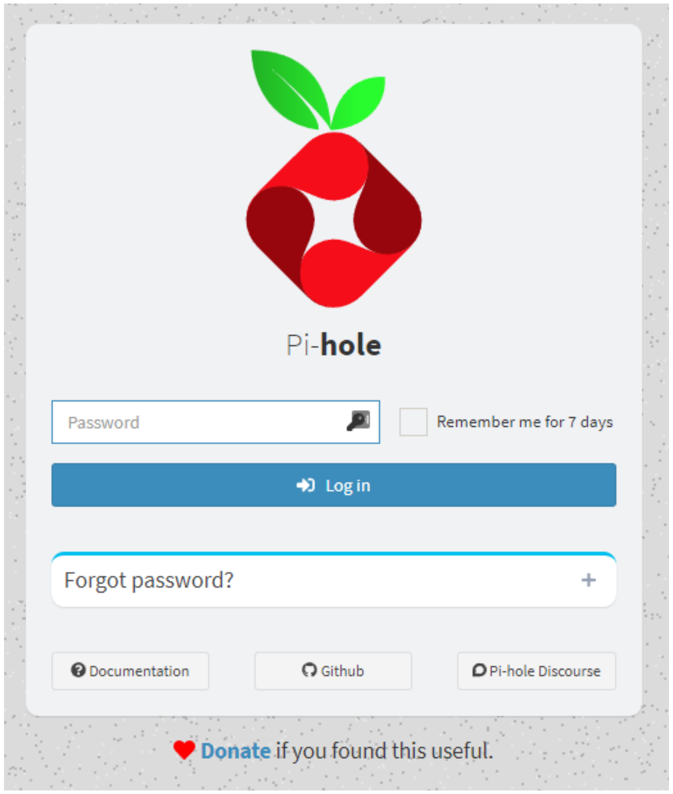
```
pi@pihole3:~ $ pihole -a -p
Enter New Password (Blank for no password):
Confirm Password:
  [✓] New password set
pi@pihole3:~ $
```

Log into the Pi-hole GUI

We’re now ready to log into the GUI for the first time! Open up a browser and input the IP address followed by /admin in this format:

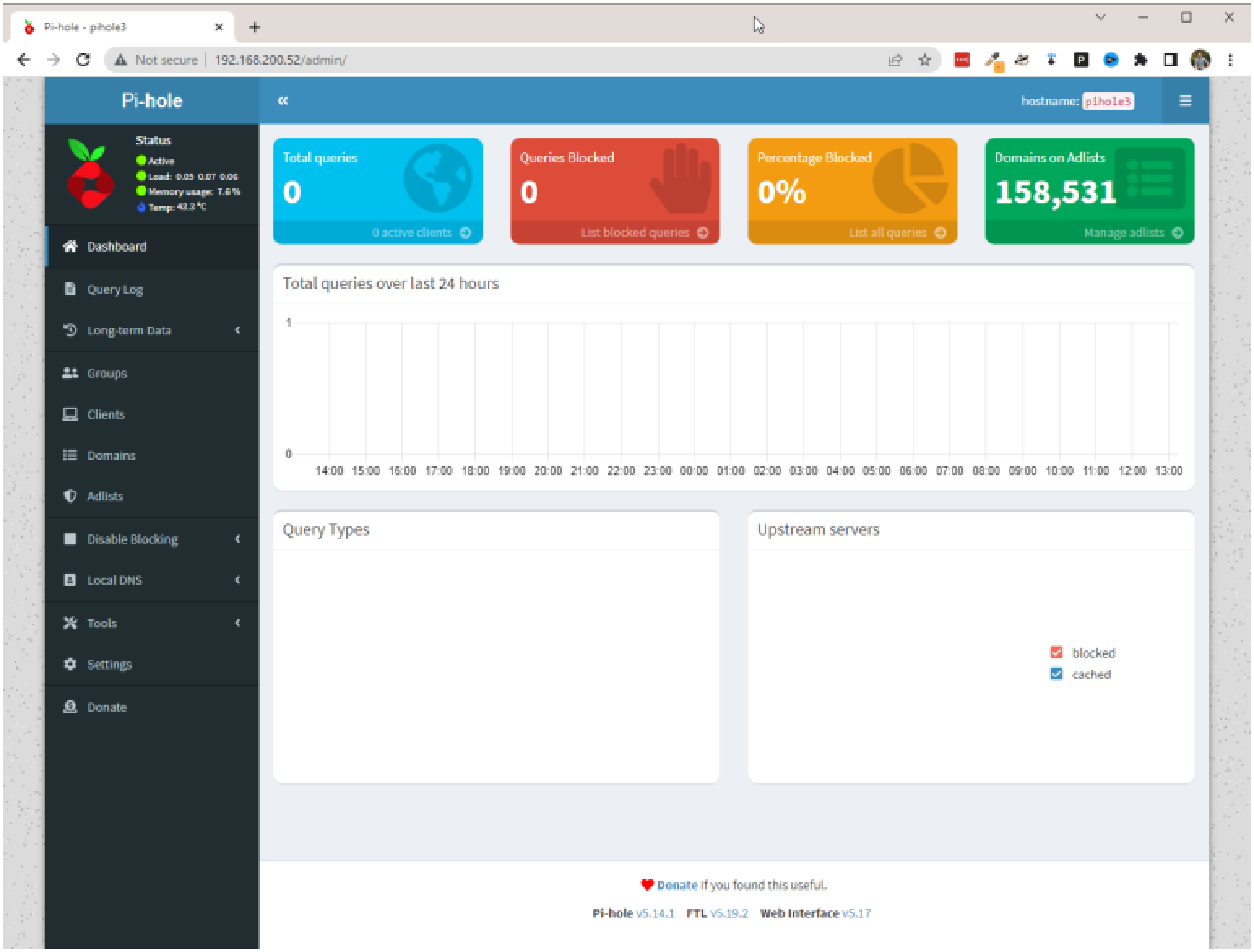
http://192.168.200.52/admin

(Substitute my IP address with your own). Note also that this is HTTP and not HTTPS.



Pi-hole Admin GUI login screen – enter your Admin password and click ‘Log in.’

Enter the Admin password that you set in the previous step and click ‘Log in.’ Upon logging in, we’re presented with the Pi-hole Dashboard – let’s take a look around!



Donate if you found this useful.

Pi-hole v5.14.1 FTL v5.19.2 Web Interface v5.17

Pi-hole dashboard.

We're not using this Pi-hole yet, so we don't have a ton of useful statistics…those will start to populate as soon as we start using this server for our DNS queries.

Down the left-hand side of the dashboard are our menu options. The 4 colored blocks across the top are some statistics. The blue box is the total number of queries Pi-hole has processed. The red box is how many of those queries matched FQDN's on the block list and were blocked. The yellow box is a percentage of blocked requests (red box divided into the blue box). Finally the green box is the number of domains that are on the block lists. Since we only have a single block list for now, we can see we have about 158,000 domains that will be blocked by this Pi-hole.

Going down the left-hand menu – I won't cover every single setting, but I will point out some of the more important ones.

Query Log – this shows you the Pi-hole's history of DNS lookups in descending order (most recent lookups at the top). You can see all domain statuses – both passed and blocked domains. For passed domains, there is a button to blacklist, and for blocked domains, there is a button to whitelist.

Long-Term Data – much like the Query Log, this section gives you a deeper dive look into the Pi-hole's DNS history and lets you filter into that data in great detail.

Groups and Clients – some pretty interesting functionality here in Groups and Clients – we're not going to dig into this section for this tutorial, but you can use this section if you wanted to block DNS queries for all devices except for devices in a specific group. Or, if you wanted to ONLY block DNS queries for devices in a specific group – you can get very granular with which clients and groups use which block lists as well.

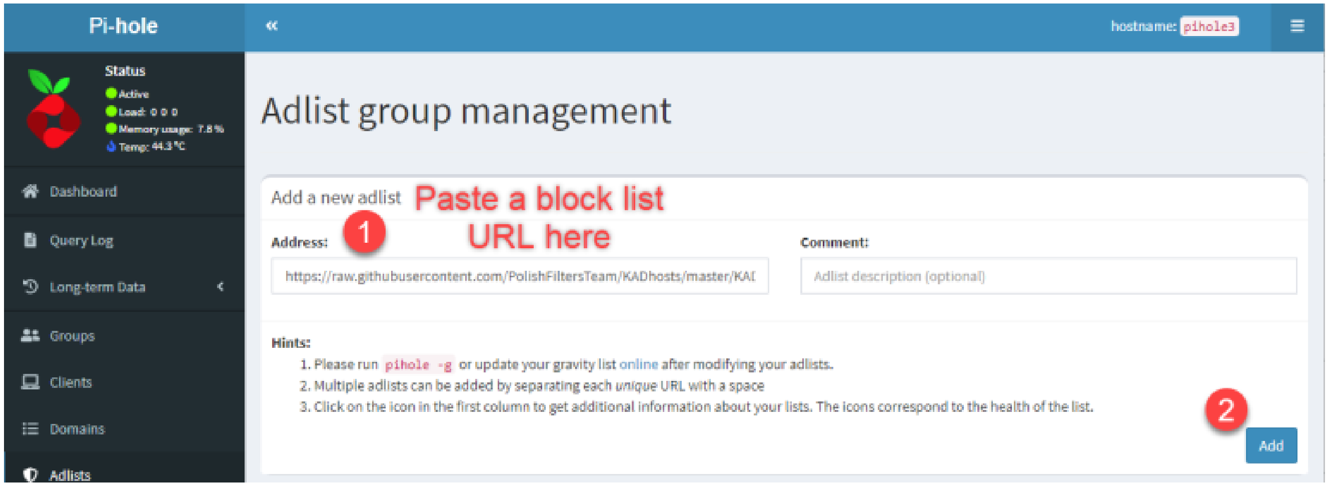
Domains – this section allows you to specifically add domains to a blacklist or whitelist. For instance, if there are specific stores where you like to shop online, and those stores are being blocked, you can whitelist them here.

> Some Blocking Resources – if you're interested in taking a deeper dive into what block lists you can add to Pi-hole, here are some good resources:
>
> The Best Pi-hole Blocklists – Avoidthehack.com – This article does an excellent job of explaining the different types of block lists, and then lists a number of resources for lists in different categories of blocking.

> The Firebog (blocklist collection) – This blocklist resource does an excellent job of providing sources of blocklists in multiple categories such as Suspicious, Advertising, Tracking & Telemetry, & Malicious. It further breaks those lists down into green links and blue links – the green links are the ones least likely to interfere with normal Internet activity. A good rule of thumb is to add one or two of these lists from each category to your Pi-hole.
>
> One more thing to point out – there is a list of commonly whitelisted domains over at Pi-hole.net. If you're having issues with a particular website or service (say Spotify or Xbox functionality for instance), go see if there are resources for whitelisting that particular service on that page – it may save you a lot of headache.

Adlists – by default, we have the default block list which is well maintained and blocks plenty of sites without breaking normal Internet functionality. That being said, you can get VERY detailed in what you're blocking – malicious sites, adult content, ad sites, tracking & telemetry sites – there are specific public block lists for each of these types of content, but keep in mind that the MORE sites you add to your block lists, the greater chance you have of breaking something. Then you'll have your family complaining to you about 'the Internet' not working, and you'll have to deal with it. The trick is to find the happy medium that does plenty of blocking without affecting the user experience.

Since we're taking a look at various block lists, let's go ahead and add a few from The Firebog page. Copy the first link under 'Suspicious Lists' and paste it into the 'Address:' field – then click Add.



Adlist group management

Add a new adlist

Address: https://raw.githubusercontent.com/PolishFiltersTeam/KADhosts/master/KAI

Comment: Adlist description (optional)

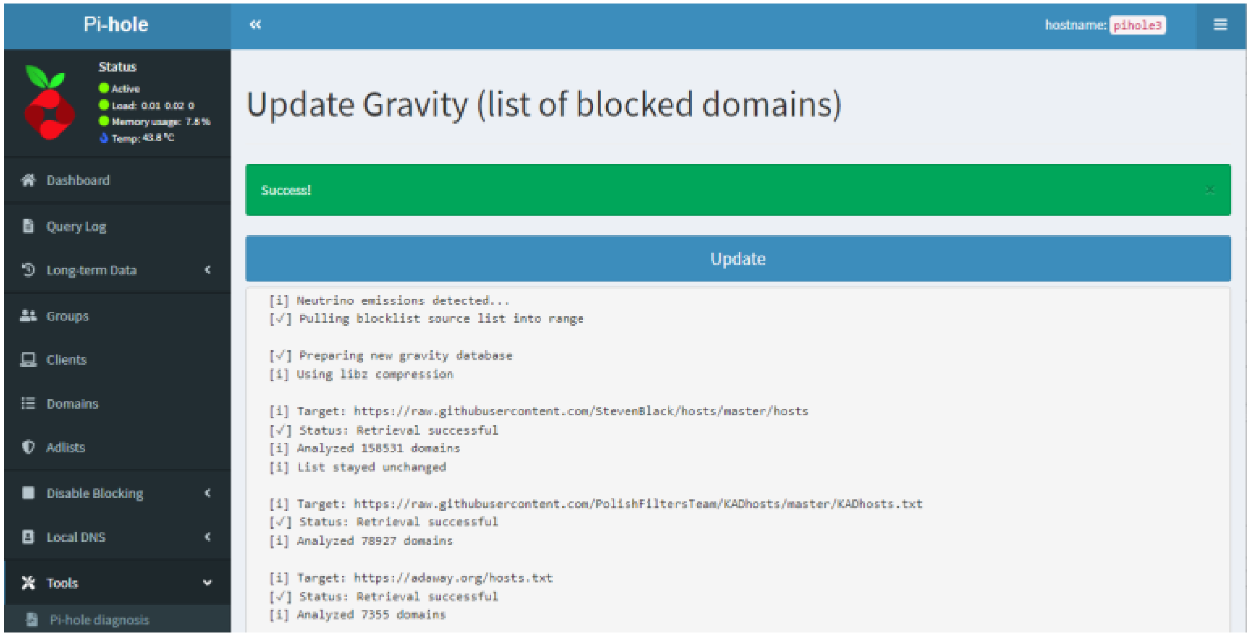
Hints:

1. Please run `pihole -g` or update your gravity list online after modifying your adlists.
2. Multiple adlists can be added by separating each *unique* URL with a space
3. Click on the icon in the first column to get additional information about your lists. The icons correspond to the health of the list.

Copy/paste a block list into Adlist group management, and click Add.

Repeat this for as many block lists as you want – for this tutorial, I'm going to copy/paste the first two green URLs from each of the Firebog sections. Once completed, we need to tell Pi-hole to import these lists.

Navigate to Tools –> Update Gravity and then click the 'Update' button. This will comb through all of the block lists and add the blocked URLs to the Pi-hole database. Once complete, you'll see a green 'Success' banner at the top of the screen.



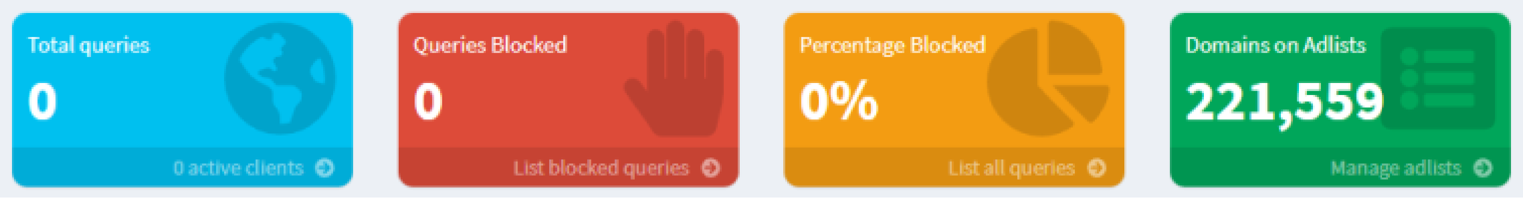
Update Gravity (list of blocked domains)

Success!

```
[i] Neutrino emissions detected...
[✓] Pulling blocklist source list into range

[✓] Preparing new gravity database
[i] Using libz compression

[i] Target: https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts
[✓] Status: Retrieval successful
[i] Analyzed 158531 domains
[i] List stayed unchanged

[i] Target: https://raw.githubusercontent.com/PolishFiltersTeam/KADhosts/master/KADhosts.txt
[✓] Status: Retrieval successful
[i] Analyzed 78927 domains

[i] Target: https://adaway.org/hosts.txt
[✓] Status: Retrieval successful
[i] Analyzed 7355 domains
```

Update Gravity to refresh your list of blocked domains.

Now click on Dashboard from the left-hand menu – notice anything different? Our 'Domains on Adlists' jumped from about 158k to 221k blocked domains.



Total queries
0
0 active clients

Queries Blocked
0
List blocked queries

Percentage Blocked
0%
List all queries

Domains on Adlists
221,559
Manage adlists

More blocked domains now!

Let's go back to our tour of the left-hand menu!

Disable Blocking – this lets you stop Pi-hole from blocking any domains for various amounts of time (or indefinitely). Very useful in troubleshooting – if something isn't working in your network or Internet browsing, try disabling blocking for 5 minutes and see if that fixes it. If it DOES fix it, then you know there are probably some domains you're going to have to whitelist.

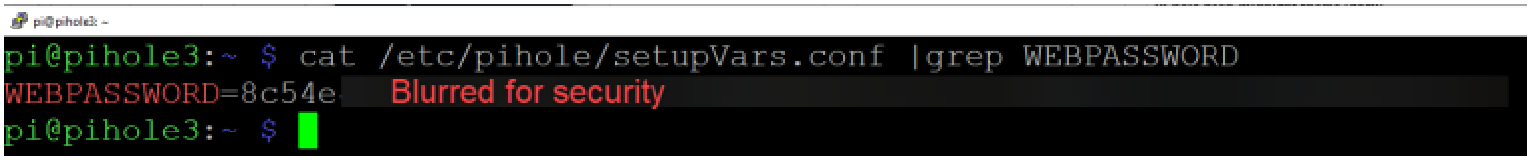
One extra tip since we're talking about disabling blocking – logging into the Pi-hole Admin GUI to disable blocking can be cumbersome! I find myself doing this on a fairly regular basis (once or twice a week), and I have multiple Pi-holes running for redundancy! So – it can take some time. To shorten up the time it takes to disable blocking, you can do some scripting. It all starts with this URL:

```
http://192.168.200.52/admin/api.php?disable=300&auth=PWHASH
```

Let's break this down – the IP address is the IP address of your Pi-hole (don't use mine for this). The 'disable=300' means disable for 300 seconds (5 minutes), and the auth=PWHASH is the hashed value of WEBPASSWORD which can be found in the /etc/pihole/setupVars.conf file. You can see it if you log into your Pi-hole with SSH and run the command:

```
cat /etc/pihole/setupVars.conf |grep WEBPASSWORD
```

This will show you your hashed password – copy and paste everything after the = and append it to the end of the URL above.

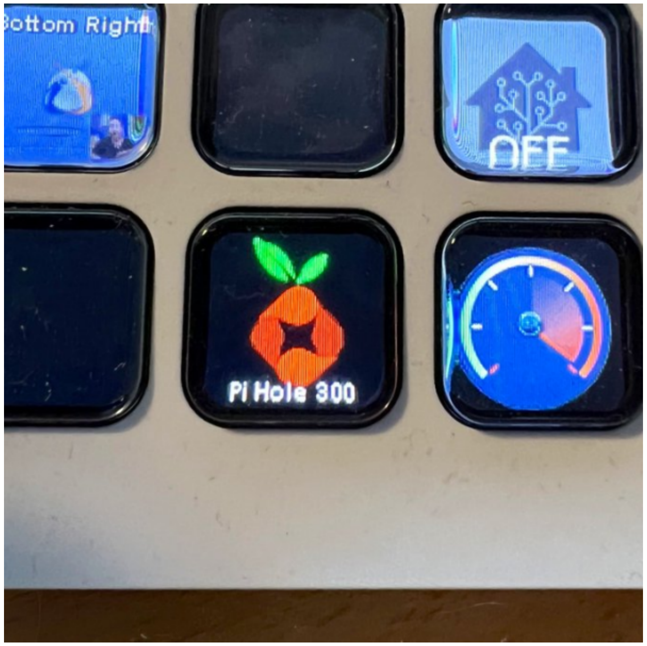


```
pi@pihole3: ~
pi@pihole3:~ $ cat /etc/pihole/setupVars.conf |grep WEBPASSWORD
WEBPASSWORD=8c54e[redacted]   Blurred for security
pi@pihole3:~ $ 
```

How to find your hashed password value

Pro tip – in PuTTY, you can select text and then do CTRL+INS to copy it to your clipboard.

If you now add that hashed password to the end of the URL string above and then pop it into a browser, this will authenticate to your Pi-hole and disable blocking for 5 minutes. You can now use that URL in scripts for Home Assistant, create a shortcut on your desktop that runs that URL, or do what I did and create a 'Disable Pi-hole' button on your Stream Deck!



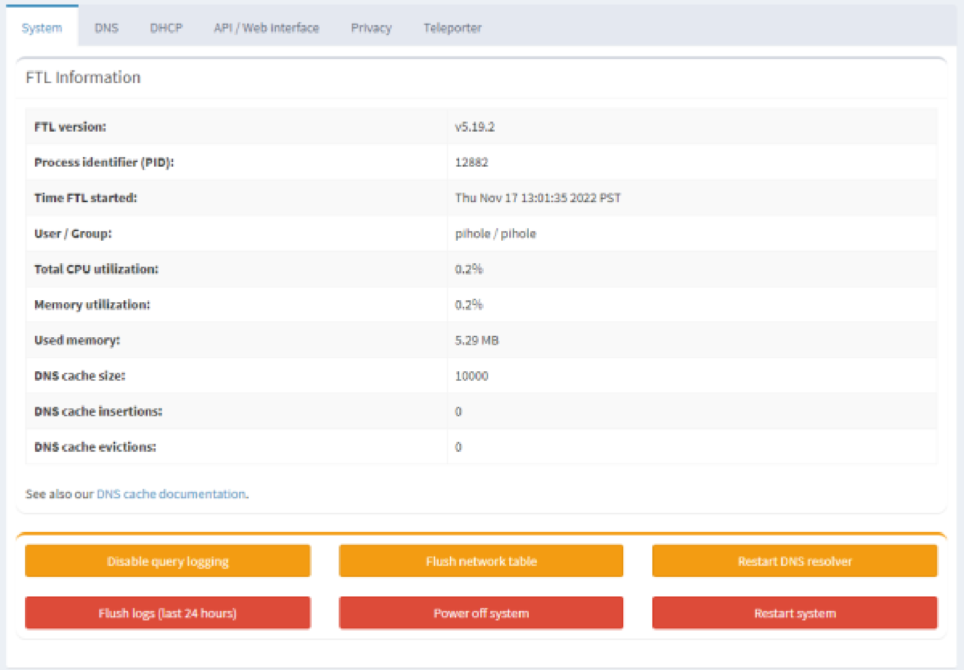
In the case of my Stream Deck button, it runs the disable URL for my first Pi-hole, pauses 1 second, and then runs the disable URL for my 2nd Pi-hole. Works great!

Local DNS – This allows you to create your own local DNS entries – very useful for using FQDNs with internal servers and devices.

Tools – yea…tools. Check 'em out.

Settings

Last but certainly not least, we have our Settings. This demands its own section of this tutorial. We'll start with the 'System' tab – this shows us various info about the Pi-hole – software versions, resource utilization, etc. We also have the option of controlling DNS and the server itself using the buttons at the bottom.



FTL Information

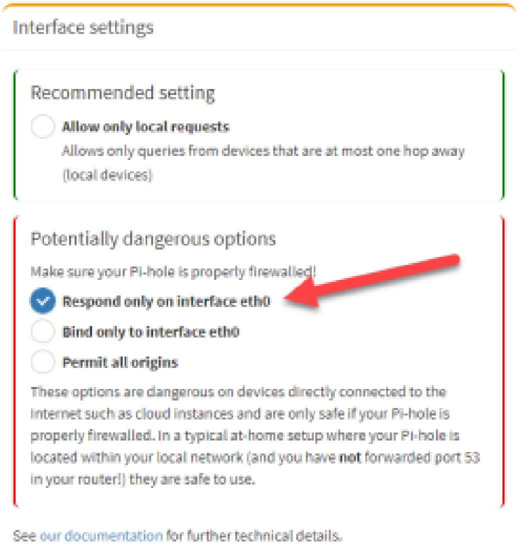
FTL version:	v5.19.2
Process identifier (PID):	12882
Time FTL started:	Thu Nov 17 13:01:35 2022 PST
User / Group:	pihole / pihole
Total CPU utilization:	0.2%
Memory utilization:	0.2%
Used memory:	5.29 MB
DNS cache size:	10000
DNS cache insertions:	0
DNS cache evictions:	0

See also our DNS cache documentation.

Settings –> System

The DNS tab shows us which upstream providers we're using for non-cached DNS queries. Since I picked Cloudflare during the setup wizard, we can see Cloudflare is my primary and secondary lookup source. You can also play around with the primary and secondary checkboxes here to have different DNS query sources – for instance, Cloudflare + Google. If you're doing this full tutorial and installing Unbound, ignore this for now – we'll be changing this shortly.

One other thing to point out on the DNS tab – on the right hand side is the 'Interface settings' section. By default, the Pi-hole is configured to ONLY respond to DNS queries from your local LAN. But in my case, I have 3 separate VLANs in my home network – LAN, Guests, and IoT Devices. I want all of them to be able to use the Pi-hole, so we need to change that setting to allow requests from other subnets. In my case, I'm going to choose the 'Respond only on interface eth0' which basically allows any DNS requests that come into the network port.



Interface settings

Recommended setting

○ **Allow only local requests**

Allows only queries from devices that are at most one hop away (local devices)

Potentially dangerous options

Make sure your Pi-hole is properly firewalled!

☑ **Respond only on interface eth0**

○ **Bind only to interface eth0**

○ **Permit all origins**

These options are dangerous on devices directly connected to the Internet such as cloud instances and are only safe if your Pi-hole is properly firewalled. In a typical at-home setup where your Pi-hole is located within your local network (and you have **not** forwarded port 53 in your router!) they are safe to use.

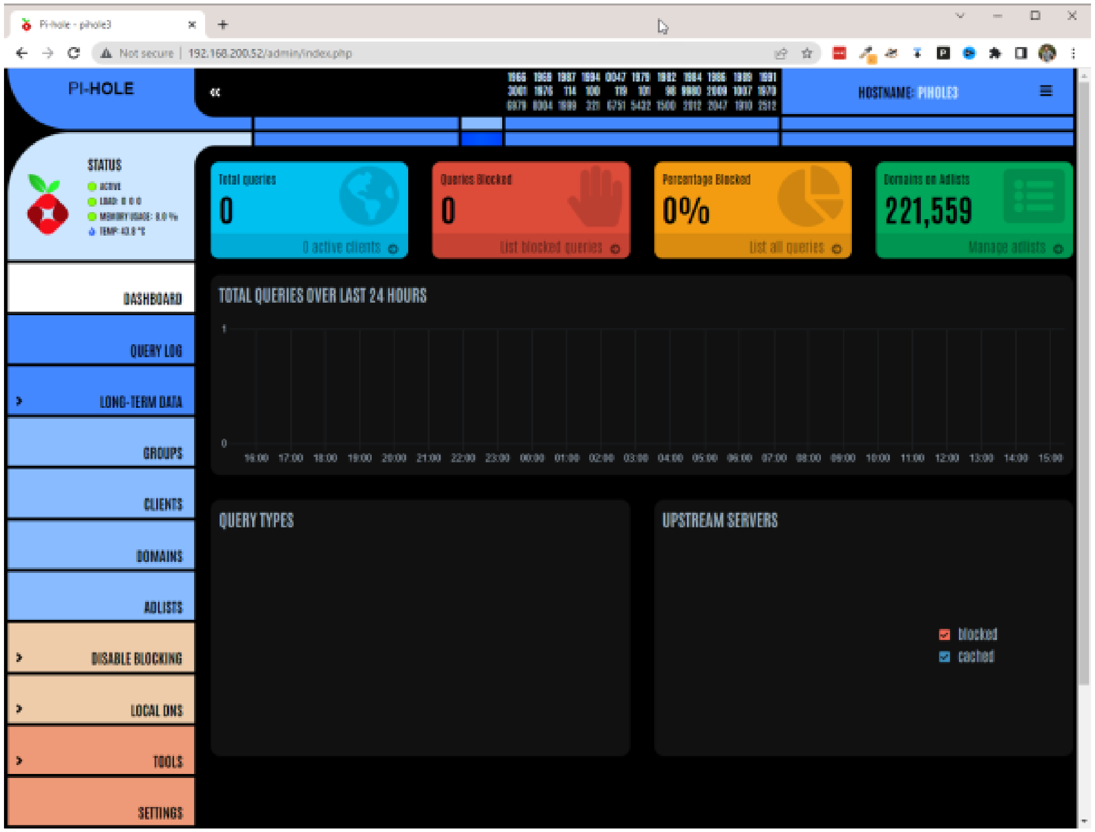
See our documentation for further technical details.

Allow DNS queries from other subnets.

Be sure to hit Save at the bottom of the screen when you make this change!

DHCP tab – the Pi-hole can be used as a DHCP server if you want, but this is beyond the scope of this tutorial.

API / Web Interface tab – this tab allows you to filter out some specific domains you don't want to be shown in the dashboard stats. It also lets you change the look and feel of the Admin GUI! Multiple light and dark themes, plus the Star Trek LCARS theme!



Star Trek LCARS theme for the Pi-hole Admin GUI!

The Privacy tab lets you change which domains are displayed or hidden – this was an option in the initial setup wizard – if you want to change these settings…here ya go.

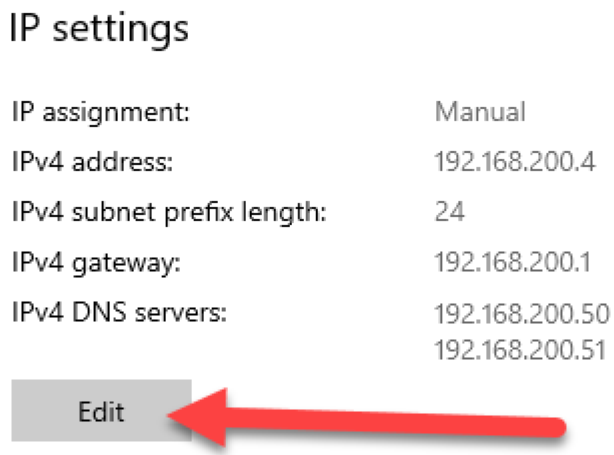
Finally, the Teleporter tab lets you backup and restore your Pi-hole settings. This is incredibly useful for keeping settings in sync across multiple Pi-holes. Once you get your whitelists/blacklists, block lists, and groups in place, you can just back up all of that data and restore it onto your 2nd Pi-hole server.

Configure Devices to use the Pi-hole

Awesome – we've got our Pi-hole all up and running, and configured to block ads and other bad guys. But – right now, nothing is using this Pi-hole! Let's fix that – there are two main ways to configure your devices to use the Pi-hole – manually, and via DHCP.

Manual device configuration – to configure your devices to use the Pi-hole manually, you need to open up your device's network settings and set your DNS server to be the IP address of the Pi-hole (or Pi-holes if you're setting up multiple for redundancy).

In Windows 11, you'll need to go to Network & Internet settings, click on 'Change adapter options' (or Properties of the current adapter). Then 'Edit' IP settings.

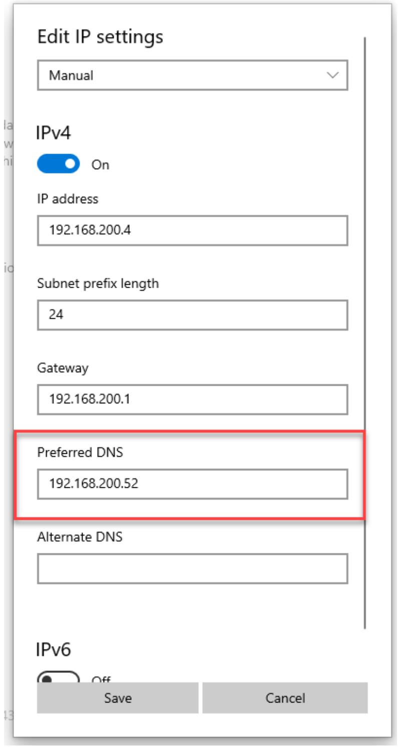


IP settings

IP assignment:	Manual
IPv4 address:	192.168.200.4
IPv4 subnet prefix length:	24
IPv4 gateway:	192.168.200.1
IPv4 DNS servers:	192.168.200.50 192.168.200.51

Windows 11 – Edit IP settings

You'll have to set your IP address for your device to 'Manual' and then give it an IP, gateway, and input the IP address of your Pi-hole for 'Preferred DNS.' If you have multiple Pi-holes, enter the 2nd one as the 'Alternate DNS.'



Edit IP settings

Manual

IPv4

On

IP address: 192.168.200.4

Subnet prefix length: 24

Gateway: 192.168.200.1

Preferred DNS: 192.168.200.52

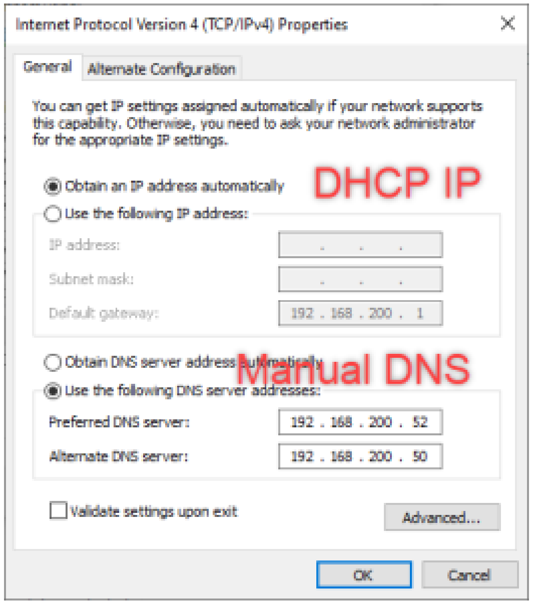
Alternate DNS:

IPv6

Off

Modify DNS settings manually.

This however means you have to set an IP address statically for your Windows 11 device. Alternatively, you can go to Control Panel –> Network and Internet –> Network Connections, find your NIC, right-click on it, choose Properties, double-click on Internet Protocol Version 4 (TCP/IPv4), and then leave 'Obtain an IP address automatically' set at the top, but choose 'Use the following DNS server addresses:' down below and then set the IP address of your primary and secondary Pi-hole servers.



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: 192 . 168 . 200 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 192 . 168 . 200 . 52

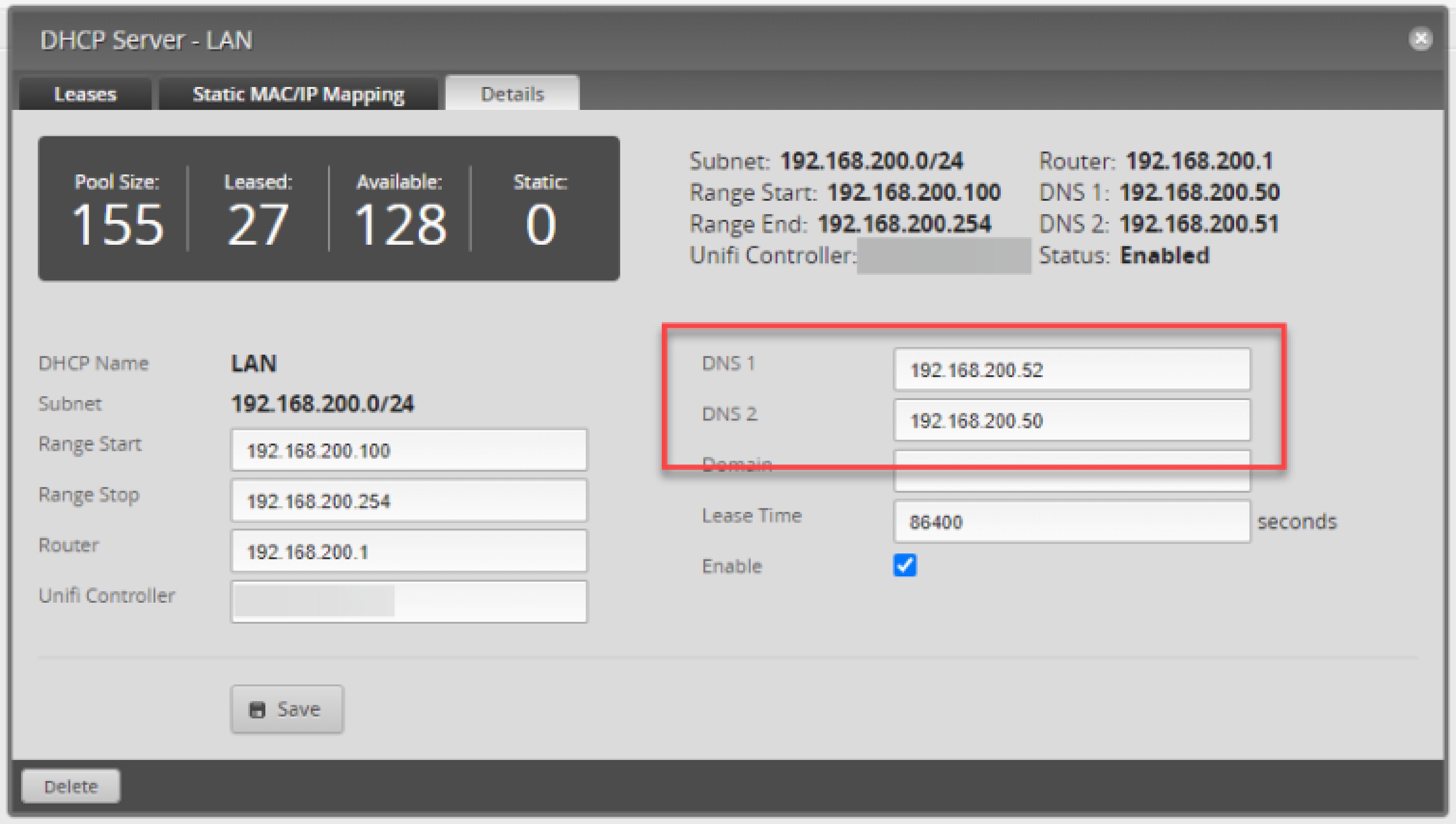
Alternate DNS server: 192 . 168 . 200 . 50

□ Validate settings upon exit

Windows settings with a DHCP'd IP address,
but manual DNS server settings.

Holy crap what a PITA. Now imagine going around and doing this for all of your devices – no thank you.

The far superior method is to use your DHCP server to set these Pi-hole settings. This way, when devices connect to your network and request IP address information, they're automatically provided with your Pi-hole DNS servers. This method is going to vary depending on your DHCP server, but it's usually going to be similar to the way I have mine set in the EdgeRouter 4.



Pool Size: 155
Leased: 27
Available: 128
Static: 0

Subnet: **192.168.200.0/24**
Range Start: **192.168.200.100**
Range End: **192.168.200.254**
Unifi Controller: [redacted]
Router: **192.168.200.1**
DNS 1: **192.168.200.50**
DNS 2: **192.168.200.51**
Status: **Enabled**

DHCP Name: **LAN**
Subnet: **192.168.200.0/24**
Range Start: 192.168.200.100
Range Stop: 192.168.200.254
Router: 192.168.200.1
Unifi Controller: [redacted]

DNS 1: 192.168.200.52
DNS 2: 192.168.200.50
Domain:
Lease Time: 86400 seconds
Enable: ☑

Set the Pi-hole IP address as the primary DNS server for your DHCP leases. Add a 2nd Pi-hole as the secondary DNS if you have one.

In the screenshot above, you can see that I added the IP address of my Pi-hole server (192.168.200.52) as the Primary DNS server that my clients will receive when they connect to my network. I have also added a 2nd Pi-hole server IP as DNS 2 for backup purposes.

If you don't have a second Pi-hole, you can also populate DNS 2 with a public DNS server such as Cloudflare (1.1.1.1). If you go through the list of available upstream DNS providers that Pi-hole offers (Settings –> DNS), there are some that provide similar content filtering to Pi-hole that would work as a decent secondary DNS server for your clients.

Once you've made this change, you should shortly start seeing stats hit the Pi-hole dashboard:

Total queries
170
2 active clients

Queries Blocked
28
List blocked queries

Percentage Blocked
16.5%
List all queries

Domains on Adlists
221,559
Manage adlists

Stuff starting to get blocked!

Unbound Setup

OK – so we have our Pi-hole working and blocking stuff – excellent! But let's take this a step further. Right now, when we look up a website, if the Pi-hole doesn't know where to find it, it forwards that request to an upstream DNS provider (as we have it configured right now, that upstream provider is Cloudflare). Those who are more security-conscious may not want some random 3rd party knowing which domains they're surfing to or looking up (even though Cloudflare doesn't keep logs of DNS queries).

For those folks, you can install Unbound on your Pi-hole. Unbound is a service that directly queries the DNS root domain servers for any uncached FQDN requests. First we'll need to install Unbound, and then we'll configure it for use with our Pi-hole.

To install Unbound, SSH into the Pi-hole and run this command:

```
sudo apt install unbound -y
```

This should only take about 10-15 seconds. Next, we need to add a whole wall of text to an Unbound configuration file – create this file by running this command to edit it:

```
sudo nano -w /etc/unbound/unbound.conf.d/pi-hole.conf
```

Copy all of this text below – you can also find this text in the Pi-hole documentation HERE.

```
server:
# If no logfile is specified, syslog is used
# logfile: "/var/log/unbound/unbound.log"
verbosity: 0

interface: 127.0.0.1
port: 5335
do-ip4: yes
do-udp: yes
do-tcp: yes

# May be set to yes if you have IPv6 connectivity
do-ip6: no

# You want to leave this to no unless you have *native* IPv6. With 6to4 and
# Terredo tunnels your web browser should favor IPv4 for the same reasons
prefer-ip6: no

# Use this only when you downloaded the list of primary root servers!
# If you use the default dns-root-data package, unbound will find it automatically
#root-hints: "/var/lib/unbound/root.hints"

# Trust glue only if it is within the server's authority
```

```
harden-glue: yes

# Require DNSSEC data for trust-anchored zones, if such data is absent, the zone becomes BOGUS
harden-dnssec-stripped: yes

# Don't use Capitalization randomization as it known to cause DNSSEC issues some times
# see https://discourse.pi-hole.net/t/unbound-stubby-or-dnscrypt-proxy/9378 for further details
use-caps-for-id: no

# Reduce EDNS reassembly buffer size.
# IP fragmentation is unreliable on the Internet today, and can cause
# transmission failures when large DNS messages are sent via UDP. Even
# when fragmentation does work, it may not be secure; it is theoretically
# possible to spoof parts of a fragmented DNS message, without easy
# detection at the receiving end. Recently, there was an excellent study
# >>> Defragmenting DNS - Determining the optimal maximum UDP response size for DNS <<<
# by Axel Koolhaas, and Tjeerd Slokker (https://indico.dns-oarc.net/event/36/contributions/776/)
# in collaboration with NLnet Labs explored DNS using real world data from the
# the RIPE Atlas probes and the researchers suggested different values for
# IPv4 and IPv6 and in different scenarios. They advise that servers should
# be configured to limit DNS messages sent over UDP to a size that will not
# trigger fragmentation on typical network links. DNS servers can switch
# from UDP to TCP when a DNS response is too big to fit in this limited
# buffer size. This value has also been suggested in DNS Flag Day 2020.
edns-buffer-size: 1232

# Perform prefetching of close to expired message cache entries
# This only applies to domains that have been frequently queried
prefetch: yes

# One thread should be sufficient, can be increased on beefy machines. In reality for most users running on small networks or on a single machine, it should be unnecessary to seek performance enhancement by increasing num-threads above 1.
num-threads: 1

# Ensure kernel buffer is large enough to not lose messages in traffic spikes
so-rcvbuf: 1m

# Ensure privacy of local IP ranges
private-address: 192.168.0.0/16
private-address: 169.254.0.0/16
private-address: 172.16.0.0/12
private-address: 10.0.0.0/8
private-address: fd00::/8
private-address: fe80::/10
```

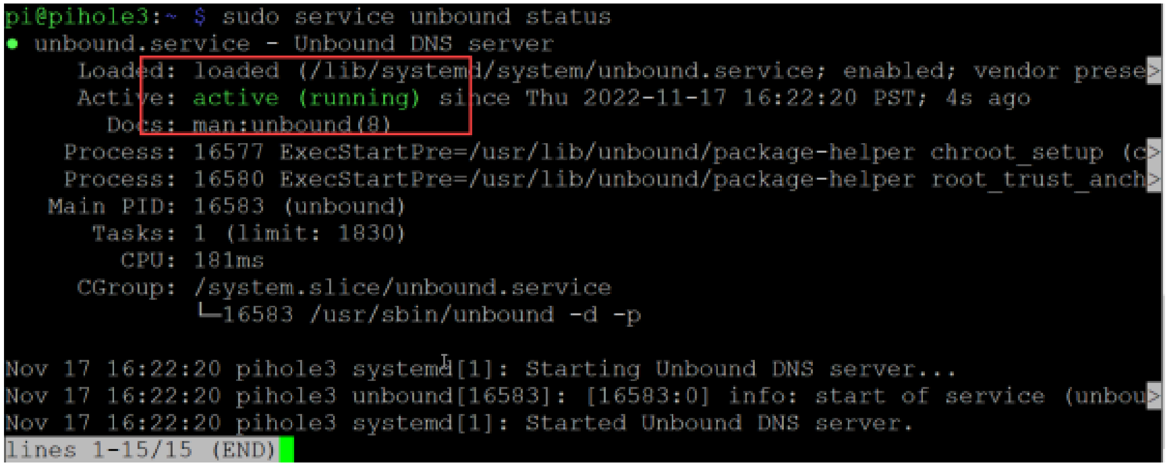
Once you've pasted in that text (use SHIFT+INS to paste into PuTTY), hit CTRL+X followed by Y and ENTER to save and exit. Then we'll restart the Unbound service:

```
sudo service unbound restart
```

If you then run:

```
sudo service unbound status
```

You should see that the Unbound service is active (running).

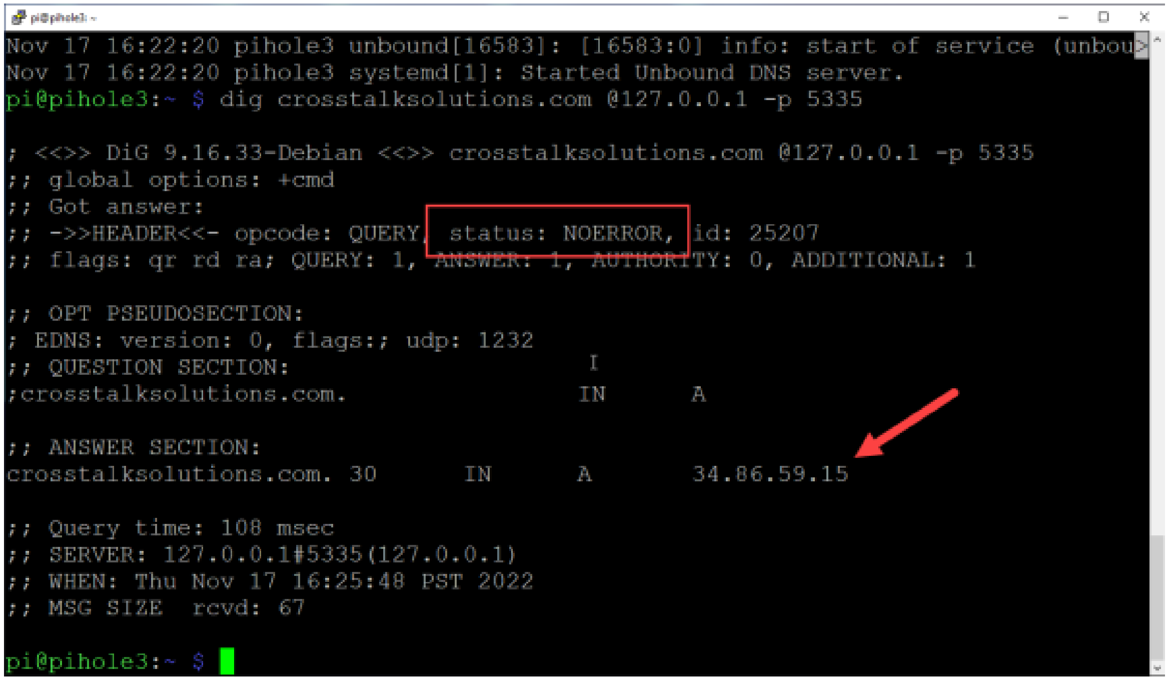


```
pi@pihole3:~ $ sudo service unbound status
● unbound.service - Unbound DNS server
     Loaded: loaded (/lib/systemd/system/unbound.service; enabled; vendor prese>
     Active: active (running) since Thu 2022-11-17 16:22:20 PST; 4s ago
       Docs: man:unbound(8)
    Process: 16577 ExecStartPre=/usr/lib/unbound/package-helper chroot_setup (c>
    Process: 16580 ExecStartPre=/usr/lib/unbound/package-helper root_trust_anch>
   Main PID: 16583 (unbound)
      Tasks: 1 (limit: 1830)
        CPU: 181ms
     CGroup: /system.slice/unbound.service
             └─16583 /usr/sbin/unbound -d -p

Nov 17 16:22:20 pihole3 systemd[1]: Starting Unbound DNS server...
Nov 17 16:22:20 pihole3 unbound[16583]: [16583:0] info: start of service (unbou>
Nov 17 16:22:20 pihole3 systemd[1]: Started Unbound DNS server.
lines 1-15/15 (END)
```

Unbound service is running.

Note in the Unbound config that we pasted in a few steps ago that we're running Unbound on IP 127.0.0.1 (localhost) and port 5335. In order to test that Unbound is working, we can send a DNS query to that IP and port:

dig crosstalksolutions.com @127.0.0.1 -p 5335



```
Nov 17 16:22:20 pihole3 unbound[16583]: [16583:0] info: start of service (unbou>
Nov 17 16:22:20 pihole3 systemd[1]: Started Unbound DNS server.
pi@pihole3:~ $ dig crosstalksolutions.com @127.0.0.1 -p 5335

; <<>> DiG 9.16.33-Debian <<>> crosstalksolutions.com @127.0.0.1 -p 5335
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25207
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;crosstalksolutions.com.                IN      A

;; ANSWER SECTION:
crosstalksolutions.com. 30      IN      A       34.86.59.15

;; Query time: 108 msec
;; SERVER: 127.0.0.1#5335(127.0.0.1)
;; WHEN: Thu Nov 17 16:25:48 PST 2022
;; MSG SIZE  rcvd: 67

pi@pihole3:~ $
```

Performing a DNS lookup to Unbound.

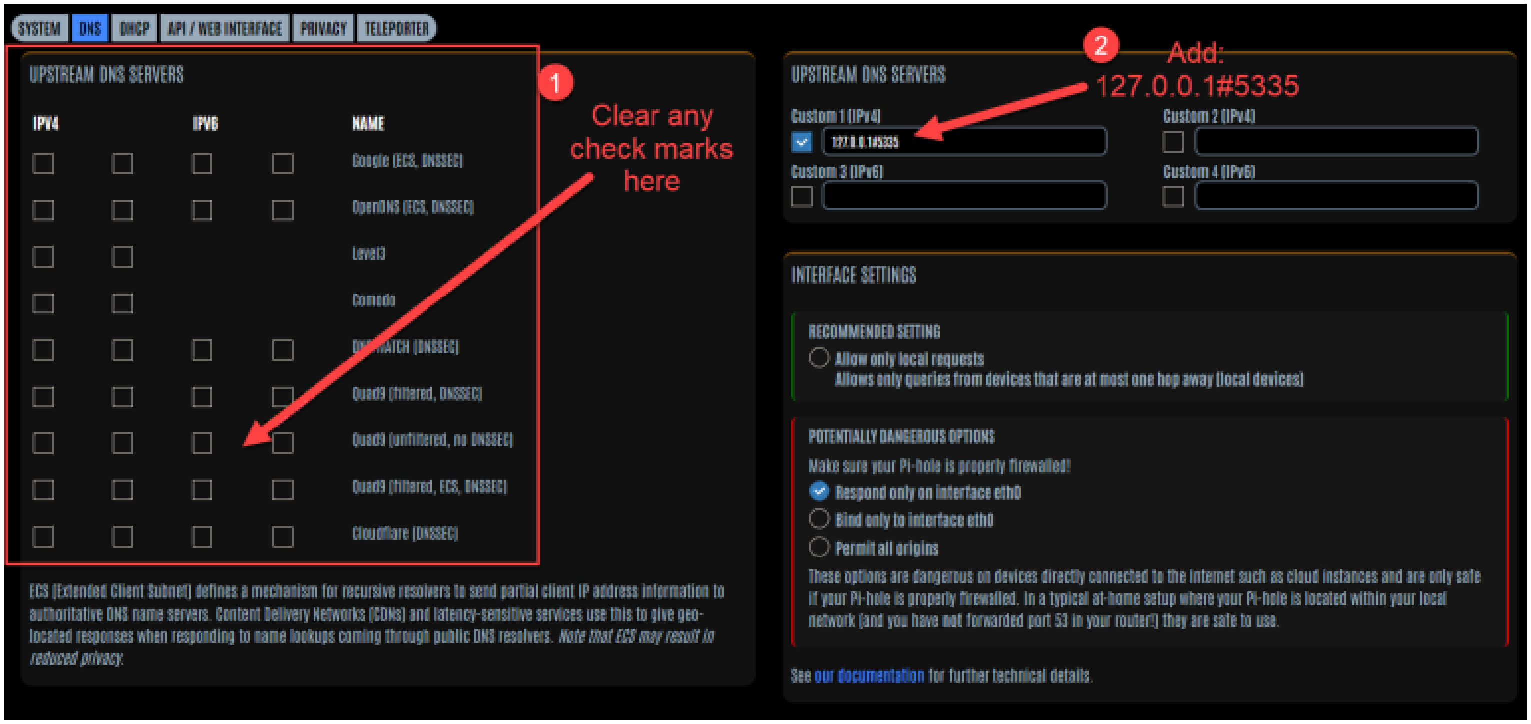
In this case, we can see that the status is NOERROR and we received an IP address for our query. Looks like we're ready to use Unbound with Pi-hole!

Using Unbound with Pi-hole

Log into the Pi-hole Admin GUI and navigate to Settings –> DNS.

Uncheck the boxes next to Cloudflare (or whichever DNS provider you picked during the install wizard), and then add a new custom entry for Unbound:

```
127.0.0.1#5335
```



UPSTREAM DNS SERVERS

IPV4		IPV6		NAME
□	□	□	□	Google (ECS, DNSSEC)
□	□	□	□	OpenDNS (ECS, DNSSEC)
□	□			Level3
□	□			Comodo
□	□	□	□	DNSWATCH (DNSSEC)
□	□	□	□	Quad9 (filtered, DNSSEC)
□	□	□	□	Quad9 (unfiltered, no DNSSEC)
□	□	□	□	Quad9 (filtered, ECS, DNSSEC)
□	□	□	□	Cloudflare (DNSSEC)

ECS (Extended Client Subnet) defines a mechanism for recursive resolvers to send partial client IP address information to authoritative DNS name servers. Content Delivery Networks (CDNs) and latency-sensitive services use this to give geo-located responses when responding to name lookups coming through public DNS resolvers. *Note that ECS may result in reduced privacy.*

Clear any check marks here

UPSTREAM DNS SERVERS

Custom 1 (IPv4)
☑ 127.0.0.1#5335

Custom 2 (IPv4)
□

Custom 3 (IPv6)
□

Custom 4 (IPv6)
□

Add:
127.0.0.1#5335

INTERFACE SETTINGS

RECOMMENDED SETTING

○ Allow only local requests
Allows only queries from devices that are at most one hop away (local devices)

POTENTIALLY DANGEROUS OPTIONS

Make sure your Pi-hole is properly firewalled!

● Respond only on interface eth0
○ Bind only to interface eth0
○ Permit all origins

These options are dangerous on devices directly connected to the Internet such as cloud instances and are only safe if your Pi-hole is properly firewalled. In a typical at-home setup where your Pi-hole is located within your local network (and you have not forwarded port 53 in your router!) they are safe to use.

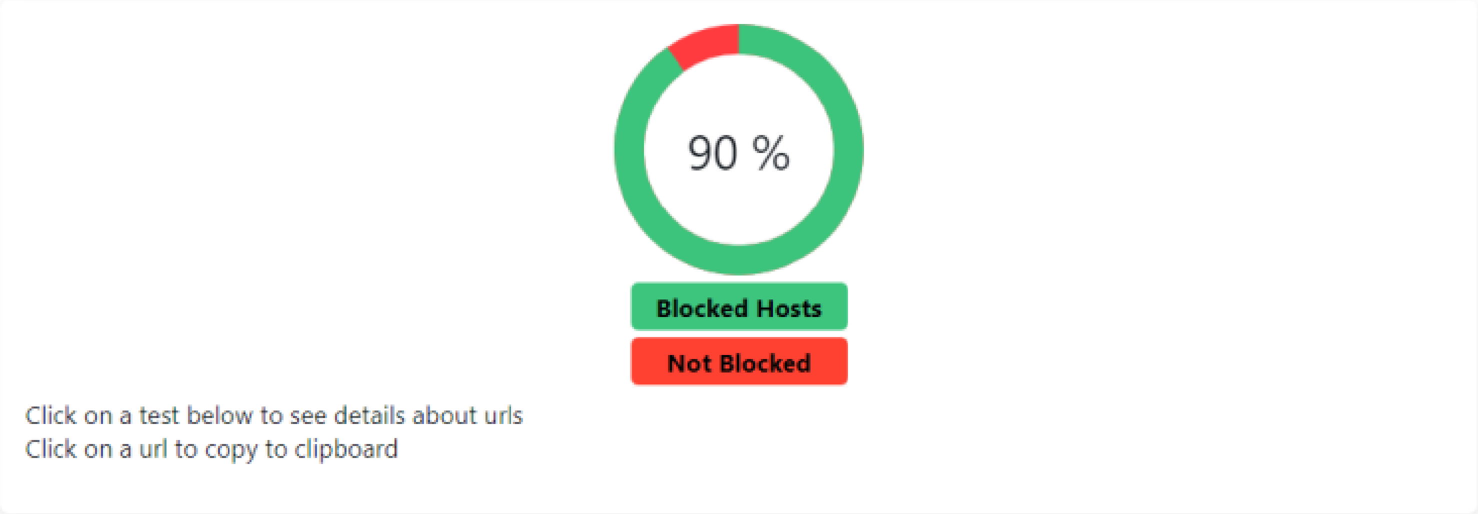
See our documentation for further technical details.

Clear existing upstream DNS servers and add a new custom entry.

Click ‘Save’ at the bottom of this screen. All done! Treat yourself to a beer.

Testing Ad Blocking

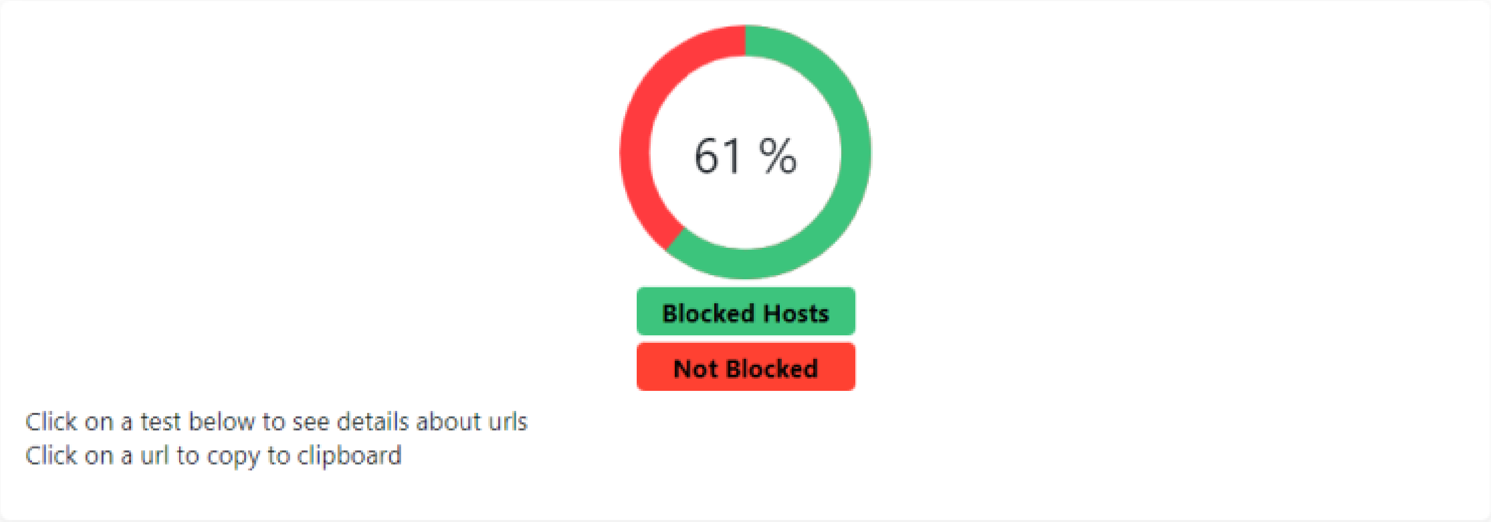
If you want to test out Ad Blocking, there’s a great tool for this HERE. Open up that page, and you should be around 90% blocking with this tutorial:



Click on a test below to see details about urls
Click on a url to copy to clipboard

Ad blocking test results.

By contrast, if I bypass my Pi-hole and just use 8.8.8.8 (Google) as my DNS resolver, I get 61%. Yikes! Let's get that Pi-hole back on ASAP!



Click on a test below to see details about urls
Click on a url to copy to clipboard

Ad blocking results with 8.8.8.8.

Updating the Pi-hole

Every so often, you'll want to update Pi-hole. To do so, simply log into the Pi-hole via SSH and run this command:

```
sudo pihole -up
```

Firewall considerations

This is beyond the scope of this tutorial, but at least worth mentioning. If you have kids that being blocked from stuff they're searching for, or if you're running Pi-hole in your business and your employees want to bypass your blocking attempts, by default, there's nothing stopping them from manually adjusting their DNS settings to use a non-Pi-hole DNS server.

To help with this, you'll need to set up some firewall rules. For instance, you may create a rule that allows DNS services on port 53 for the IP addresses of your Pi-holes, but BLOCKS port 53 everywhere else. This way – even if someone is savvy enough to manually bypass their DNS settings to get around your Pi-hole, they still won't be able to resolve anything. Haha…gotcha sucker.

Thank you!

Thanks so much for using this Pi-hole tutorial – I will update it with any corrections or updates to the installation instructions.

Remember to DONATE to the Pi-hole project if you are getting a lot out of this ad blocking server!

If you would like to support Crosstalk Solutions, you can BUY ME A BEER HERE! Or check out some of the cool stuff we have in our Crosstalk merch store!

2FA FTW T-Shirt

$35.00

Select options

Network the Moon T-Shirt

$35.00

Select options

Crosstalk Solutions #DALL-E

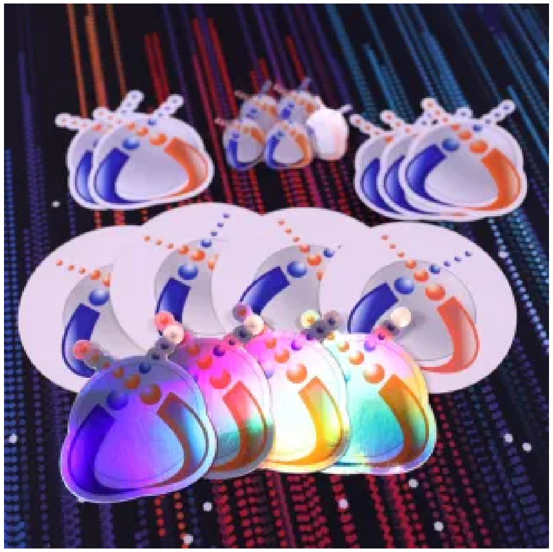
Felix the Network Weasel DALL-E Sticker

$5.00

Add to cart

Crosstalk Magnetic Dish

$12.75



Crosstalk Schwag Bag

$9.99

HOW TO

PI-HOLE

RASPBERRY PI

Comments 33

Jason Stillion

NOVEMBER 22, 2022 AT 12:02 PM

Great Tutorial.
May want to include the instructions to expand the usable space on the memory card.
sudo raspi-config
Advanced Options
Expand Filesystem
Ok / Exit out – Reboot when prompted

Reply

Crosstalk Solutions

NOVEMBER 22, 2022 AT 12:30 PM

I’m pretty sure that happens automatically now – I did a before/after comparison of df -h on my Pi, and it was the exact same output before and after running the ‘Expand file system’ command…pretty sure you haven’t had to do that for a while now.

Reply ↩

Jean-Luc Kapetanovic

NOVEMBER 22, 2022 AT 10:38 PM

Very useful, especially the Blocking Resources. I did block too much at the first place and disable pihole for a while giving more time to tune it. Your video just came at the right moment for me. Now better understanding of pihole blocking list & tune it sucessfully. Thank for the video.

Reply ↩

Chris

DECEMBER 26, 2022 AT 12:15 AM

Hello Jean, is there a good URL address for blocking Youtube ads? Please let me know. thanks

Reply ↩

K

NOVEMBER 23, 2022 AT 6:42 AM

Hello Chris! thanks for an amazing tutorial, I just followed it with pinhole running in a docker container. Please maybe make a short video with firewall rules for pi hole for your complete Unifi setup playlist. I have your exact setup and would love to see a video that would help me with setting it up correctly. I'm sure many home users who have teenage kids would appreciate it.

Reply ↩

Brian Kehoe

NOVEMBER 28, 2022 AT 4:53 PM

Awesome tutorial and video. Super helpful. A follow-up video with firewall rules would be awesome.

Reply ↩

JC

JANUARY 12, 2023 AT 10:29 AM

Agree, Chris it would be great if you could do a follow up to how to setup Unify firewalls to block 'other' DNS and point / allow only the use of the Pi-Hole. Is this something you're planning on as I didn't see it covered in your UDMP setup series either ? Thanks

Reply ↩

Raj

NOVEMBER 24, 2022 AT 2:22 PM

Excellent tutorial. Installation went without hitch, thanks to detailed instructions. Works as intended.
One thing though, this is not blocking youtube ads on smart tv and this is the only reason I wanted to add pihole to my network. Is there any other solution for blocking ads on smart tv?

Reply ↩

Hebert

NOVEMBER 24, 2022 AT 6:56 PM

Thanks for the tutorial. Unbound didn't work for me.
I'm only getting SERVERFAIL. Did everything that is in your tutorial. And a little more.
Created a directory to unbound.log. But the thing is it is not recording any data. I'm honestly at lost here.
https://pastebin.com/nEcmMqa6
the log directory also has the appropriate ownership:
https://pastebin.com/nEcmMqa6
When I added a second DNS address on my OpenWRT (1.1.1.1) the percentage of blocked IPs dropped to 48% from 76%
Is that normal?

Reply ↩

Geoff

DECEMBER 1, 2022 AT 9:43 PM

I too am having an issue with unbound. Service is running and I can dig a dns entry using 127.0.0.1 on port 5335 on the pihole server. However, when I set it in the admin interface then run a dns query from a network machine it fails.

Reply ↩

Geoff

DECEMBER 2, 2022 AT 4:31 PM

I had to comment out the unbound entry in resolvconf.conf:

```
#dnsmasq_resolv=/var/run/dnsmasq/resolv.conf
pdnsd_conf=/etc/pdnsd.conf
#unbound_conf=/etc/unbound/unbound.conf.d/resolvconf_resolvers.conf
```

The file it was creating in the unbound folder was creating a dns forward to my UDM-PRO, which, was pointing to the pihole causing a loop.

Reply ↩

Bill

DECEMBER 2, 2022 AT 10:52 AM

I get similar results. While browsing to http://www.cisco.com works from chrome on a windows machine, I see the following when dns points to unbound port:

```
nslookup wwww.cisco.com
Server: pi.hole
Address: 192.168.423.887 🙂
```

```
*** pi.hole can’t find wwww.cisco.com: Non-existent domain
```

But dig on the pi-hole machine gets:

```
dig http://www.cisco.com
; <> DiG 9.11.5-P4-5.1+deb10u8-Raspbian <> http://www.cisco.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER< server 8.8.8.8
Default Server: dns.google
Address: 8.8.8.8
> http://www.cisco.com
Server: dns.google
Address: 8.8.8.8
Non-authoritative answer:
Name: e2867.dsca.akamaiedge.net
Addresses: 2600:1408:c400:38e::b33
2600:1408:c400:38d::b33
104.106.160.119
Aliases: http://www.cisco.com
http://www.cisco.com.akadns.net
wwwds.cisco.com.edgekey.net
wwwds.cisco.com.edgekey.net.globalredir.akadns.net
```

apparently cnames cause glitches for pihole.

Reply ↩

Crosstalk Solutions

DECEMBER 2, 2022 AT 3:15 PM

Your Dig looks like it’s using Google (8.8.8.8) as the DNS resolver no?

Reply ↩

Bill

DECEMBER 20, 2022 AT 1:13 PM

Sorry for the delayed reply. I was just using your blog post again to set up my 2nd pihole for redundancy and noticed you replied.
I used google’s DNS to show what happens without unbound. It may have been that I recreated the error with wwww.cisco.com (4 w’s) which is a typo. Maybe that was my original problem but I was pretty sure I tried http://www.cisco.com at the time.
Regardless, I don’t get the not found error anymore.
Your tutorial is great as are all of your videos. Thank you.

Reply ↩

Ralph

NOVEMBER 25, 2022 AT 10:45 AM

Yeah, that is done at first bootup automatically these days.

Reply ↩

Ralph

NOVEMBER 25, 2022 AT 10:47 AM

Great tutorial btw.

Reply

Rob Wyrick

NOVEMBER 26, 2022 AT 11:17 AM

Consider adding instructions for common whitelists?

https://github.com/anudeepND/whitelist

Reply

Thomas

NOVEMBER 29, 2022 AT 9:01 AM

This whitelist-repo hasn’t got any update in 2022 – so you should consider it dead.

Reply

Thomas

NOVEMBER 29, 2022 AT 8:58 AM

This is really a great step-by-step tutorial 🙂
I have some additions when you “are going to pick an upstream DNS provider”:
From a privacy point of view you may not want to use a DNS provider that collects data about your DNS queries or is even tracking you! Therefore I have written some infos on my GitHub Page:
https://thomasmerz.github.io/pihole-wireguard-knowhow/#upstream-resolvers
My project is measuring very close to an end user via WiFi and via Vodafone ISP (coax/cable). So these results are what you also can expect on your home internet connection regardless which ISP (Telekom, Vodafone, 1-und-1, …) and which technology (DSL, Coax/Cable, Fiber or even mobile (4G/5G)).
There’s also a link to another GitHub repo if me with daily/nightly updated real data from my home:
https://github.com/thomasmerz/dnspingtest_rrd_ka

Reply ↩

Shawn

NOVEMBER 29, 2022 AT 9:56 AM

I’ve started using DietPi for my Pi-hole installs to squeeze out the most of my aging Pis (Gen1 & Gen2).
“DietPi is extremely lightweight at its core, with features of low process/memory footprint and DietPi-RAMlog installed by default, to get the maximum performance from your device.”

Reply ↩

Al R.

NOVEMBER 29, 2022 AT 2:03 PM

I helped write a script with a buddy of mine that automates most of the install of pihole and unbound. It only works with Ubuntu 20.04 at the moment. Setting static IP address at the begining is a good idea.
https://github.com/Dhovin/pihole-unbound

Reply ↩

Valerian

NOVEMBER 29, 2022 AT 9:14 PM

Too bad raspberry is still twice as expensive as normal

Reply ↩

nirv

NOVEMBER 29, 2022 AT 10:19 PM

Is there a good reason why you are blocking archive.today from archiving this page?
https://archive.ph/ds3eV

Reply ↩

Djeimiss selemane

NOVEMBER 29, 2022 AT 10:47 PM

Como posso adqurir

Reply ↩

Christian

NOVEMBER 30, 2022 AT 4:56 AM

Thank you very much Chris! Great tutorial. I have UDM with 2 VLANs isolated from each other and LAN. I would be really grateful if you (or someone from this chart) could let me know which firewall rule should be created in order to provide access from VLANs to Pi-hole. Sorry for this “stupid” question but I’m not IT specialist, all other firewall rules were created basing only on your videos. Thank you again.

Reply ↩

Iuri

JANUARY 2, 2023 AT 4:26 PM

Thank you, but for some reason my Tp-link router Archer C-5400X has the following error when setting up the DNS there: “DNS server IP address and LAN IP address cannot be in the same subnet. Please enter another one.”

Reply ↩

Joacim

JANUARY 9, 2023 AT 9:04 AM

From my tests this configuration of unbound still uses the resolvers found in the “/etc/dhcpcd.conf” file
“static domain_name_servers=192.168.200.1 1.1.1.1”
it is not recursive as it is meant to be.
and if its not recursive what’s the point with unbund in this tutorial?
Unbound will use port 53/tcp in some requests so that port need to be open alongside port 53/udp
found out that one the hard way ;P

Reply ↩

Mike

JANUARY 11, 2023 AT 5:51 AM

I was wondering about this. He never went back into the Pi to reconfigure the eth0 to point to itself for DNS, it’s still using this default GW and cloudflare? Followed his instructions to a T and my device is still using my default gateway for DNS too.

Reply ↩

Romain

JANUARY 14, 2023 AT 10:44 AM

Thank you very much Chris, very helpful, as a total new Pi user, this is very helpful.
I was able to put a hand on a 4GB pi 4 and have spent the day on it already.
I’m getting an issue with the Unbound DNS query.
Everything went well, my DHCP is correct and I see the dashboard updating with new blocked queries.
If anyone can provide some help, here is the dig report below.
Have I missed anything?
/

```
pirom@pi:~ $ dig crosstalksolutions.com @127.0.0.1 -p 5335
; <> DiG 9.16.33-Debian <> crosstalksolutions.com @127.0.0.1 -p 5335
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 11322
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;crosstalksolutions.com. IN A
;; Query time: 39 msec
;; SERVER: 127.0.0.1#5335(127.0.0.1)
;; WHEN: Sat Jan 14 19:13:44 CET 2023
;; MSG SIZE rcvd: 51
```

/
Many thanks for any further guidance that anyone can provide.

Reply ↩

Shaz

JANUARY 14, 2023 AT 2:03 PM

Could you please update the tutorial to include the following fix:
https://www.reddit.com/r/pihole/comments/tsperl/comment/i2sr22h/
I’ve been banging my head against a wall for hours wondering why DNS wouldn’t resolve when unbound was used. I even rebuilt my pi from scratch several times! Then I found one of several threads on reddit suggesting this solution which, fixed my issue immediately:
This forwarding configuration was installed by Bullseye through resolvconf.
Edit file /etc/resolvconf.conf and comment out the last line which should read:
unbound_conf=/etc/unbound/unbound.conf.d/resolvconf_resolvers.conf
Delete the unwanted unbound configuration file:
sudo rm /etc/unbound/unbound.conf.d/resolvconf_resolvers.conf
restart unbound:
sudo service unbound restart

Reply ↩

Tony

JANUARY 15, 2023 AT 9:16 AM

Many thanks for this tutorial. I managed to get pihole and unbound up and running in a few hours. Get som errors when setting up unbound but that was the stupid copy and paste that screwed up the line breaks, had to put in # in two places and then it works.

Reply

Owen

JANUARY 16, 2023 AT 8:53 PM

This was a great tutorial, but I am struggling getting the unbound DNS to work with my UDM-Pro.
I can ping and dig @127.0.0.1 -p 5335 just fine on my raspberry pi 4.
– I am using the tutorial setup for /etc/unbound/unbound.conf.d/pi-hole.conf
– I followed this https://www.reddit.com/r/pihole/comments/tsperl/comment/i2sr22h/
– I disabled resolvconf for unbound in https://docs.pi-hole.net/guides/dns/unbound/
– I tried adding allow all firewall rules and disabled any blocking firewall rules
No matter what I do when ever I try to dig on my LAN network, I get the following error: connection timed out; no servers could be reached
Unifi logs are not mentioning anything either. I am completely stuck.

Reply

Art

JANUARY 27, 2023 AT 7:40 AM

Great guide. I got it set up in a few hours on my LAN network with not much effort. However, I set up my Ubiquiti Dream Machine Pro following your guide as well. Would love for you to make a quick video or post on how to adjust your firewall settings for this to allow the IOT Network to work. Even in your IOT network setup video you mention how you wouldnt explain how to do it in the video.
We seemed to have locked it down pretty well in that video and while the PiHole is working great for my LAN network, if I change the settings for the IOT network, I dont get any internet. I am pretty new to this which is why I have loved using your guides and I feel I have learned alot. However this has me stumped. I feel the firewall settings are a bit confusing and I have tried a few things on how to make the pihole an exception to the rule for a few days now with no luck.
Thanks!

Reply ↩

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment*

Name*

Your Name *

Email*

Your Email *

Website

Your Website

Submit

Search

Recent Posts

- Orange Pi 5 – Simple Overview and Installation with M.2 SSD
- Starlink Side Hustle – How To Make Money with your Internet
- Mastodon Easy Server Setup
- The World’s Greatest Pi-hole (and Unbound) Tutorial 2023
- 027 – Lone Wolf Watch Party – Also Checking Out Aptera’s new Solar Powered Car!

Recent Comments

sangguen on

Mastodon Easy Server Setup

Art on

The World’s Greatest Pi-hole (and Unbound) Tutorial 2023

Ihor on

Orange Pi 5 – Simple Overview and Installation with M.2 SSD

Marcio Torres on

Orange Pi 5 – Simple Overview and Installation with M.2 SSD

Chris Sherwood on

Orange Pi 5 – Simple Overview and Installation with M.2 SSD

Archives

January 2023

December 2022

November 2022

October 2022

September 2022

July 2022

April 2022

March 2022

February 2022

January 2022

September 2021

June 2021

April 2021

March 2021

January 2021

December 2020

November 2020

October 2020

August 2020

June 2020

April 2020

March 2020

February 2020

December 2019

November 2019

August 2019

July 2019

June 2019

May 2019

April 2019

March 2019

November 2018

October 2018

September 2018

January 2018

August 2017

July 2017

- November 2016
- April 2016
- March 2016
- December 2015
- September 2015
- July 2015
- April 2015
- December 2014
- November 2014

Categories

- FreePBX
- hosted voip
- HowTo
- network

- opinion
- Raspberry Pi
- security
- Ubiquiti
- Uncategorized
- UniFi Video
- WiFi

Meta

- Log in
- Entries feed
- Comments feed
- WordPress.org