

# The Ultimate PiHole

---

## What is a PiHole?

A Pi-hole is a network-level ad blocker that acts as a Domain Name System (DNS) sinkhole. It is designed to block ads across your entire network, including on devices where traditional ad-blocking tools (like browser extensions) might not work, such as smart TVs, smartphones, and gaming consoles. It works by intercepting DNS requests and blocking any that are known to be associated with ads, trackers, and other unwanted content.

### Key Features:

1. **DNS Sinkhole:** It blocks DNS requests to known ad-serving domains, preventing devices on your network from fetching ad content.
2. **Network-wide Protection:** Once Pi-hole is set up, all devices on your network benefit from ad blocking.
3. **Customizable Block Lists:** You can add or remove domain block lists to fine-tune what is blocked.
4. **Lightweight and Efficient:** Pi-hole can run on low-powered devices like a Raspberry Pi, making it energy-efficient.
5. **Privacy:** Pi-hole helps to maintain user privacy by blocking online trackers.
6. **Easy Management:** It offers a web-based interface to monitor and control blocked requests.

Pi-hole is often used in home networks, but it can also be employed in corporate environments to reduce bandwidth usage and improve network performance by blocking unwanted traffic.

---

The PiHole keeps us out of reach from those sketchy websites but does come with some privacy concerns...

---

## The Problem: Trusting Upstream DNS Providers

Pi-hole uses [FTLDNS](#) to block ads by forwarding DNS queries to upstream DNS servers. However, trusting third-party DNS providers (e.g., Google or OpenDNS) raises privacy concerns. You can't be sure they honor promises of privacy, and attackers could target large DNS providers, potentially redirecting users to malicious sites.

### Recursive DNS Server:

A recursive DNS server resolves queries by directly contacting the authoritative DNS servers of the domain, rather than relying on an upstream provider. This improves privacy by reducing the need to trust external DNS providers.

---

## The Solution: Unbound

**Unbound** is a DNS resolver, often used for security and privacy-focused setups. It's a recursive DNS resolver, which means it takes a domain query and works through the DNS hierarchy to get an answer directly from authoritative DNS servers rather than relying on a single upstream DNS server like Google DNS or OpenDNS. This approach improves privacy, performance, and resilience.

### Key Features of Unbound:

1. **Recursive DNS Resolution:** Instead of relying on third-party DNS servers, Unbound fetches DNS information directly from authoritative sources, providing better control and privacy.
  2. **DNSSEC Validation:** Unbound supports DNSSEC (Domain Name System Security Extensions), which ensures that DNS responses are not tampered with, providing an added layer of security.
  3. **Caching:** It caches DNS responses to reduce latency and improve performance for future DNS queries.
  4. **Privacy:** Since queries are sent directly to authoritative DNS servers, it reduces the amount of third-party tracking compared to using a public DNS resolver.
-

## The Dynamic Duo: PiHole & Unbound

Pi-hole and Unbound are often paired together to provide a highly private and secure DNS-based ad-blocking setup.

1. **Complete Local DNS Setup:** By combining Pi-hole and Unbound, you create a local DNS server that doesn't rely on external DNS providers, giving full control over DNS queries.
2. **Enhanced Privacy:** Unbound ensures that DNS queries don't go through third-party DNS providers (like Google DNS), reducing potential privacy risks. Pi-hole handles the ad-blocking while Unbound resolves the DNS queries locally, reducing exposure to trackers.
3. **Security with DNSSEC:** Unbound provides DNSSEC validation, which ensures the integrity of DNS queries and responses, guarding against DNS spoofing attacks.
4. **Reduced Dependency on External Services:** Using Unbound means your network isn't dependent on external DNS servers, which can improve reliability if those servers experience downtime.

Together, Pi-hole blocks unwanted content like ads and trackers, while Unbound handles DNS queries in a privacy-preserving way, making it a powerful and secure setup for network-wide DNS and ad-blocking.

---

## Prerequisites

**Device:** A low-power device like a Raspberry Pi or a system running a Debian/Ubuntu-based Linux distribution.

- It can be easily installed on any Linux distribution but I will be using Ubuntu 24.04 LTS

**Storage and Memory:** Minimal, since Pi-hole and Unbound are lightweight applications.

- A Raspberry Pi with at least 512MB of RAM is usually sufficient.

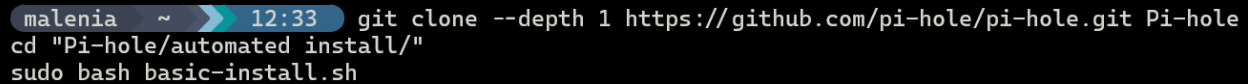
After you have [Set up your Raspberry Pi](#) with Ubuntu we can now begin installing the PiHole software.

---

## PiHole Installation:

There are a couple ways to go about installing PiHole but this is my personal choice as it is a bit more secure than other options.

```
git clone --depth 1 https://github.com/pi-hole/pi-hole.git Pi-hole
cd "Pi-hole/automated install/"
sudo bash basic-install.sh
```

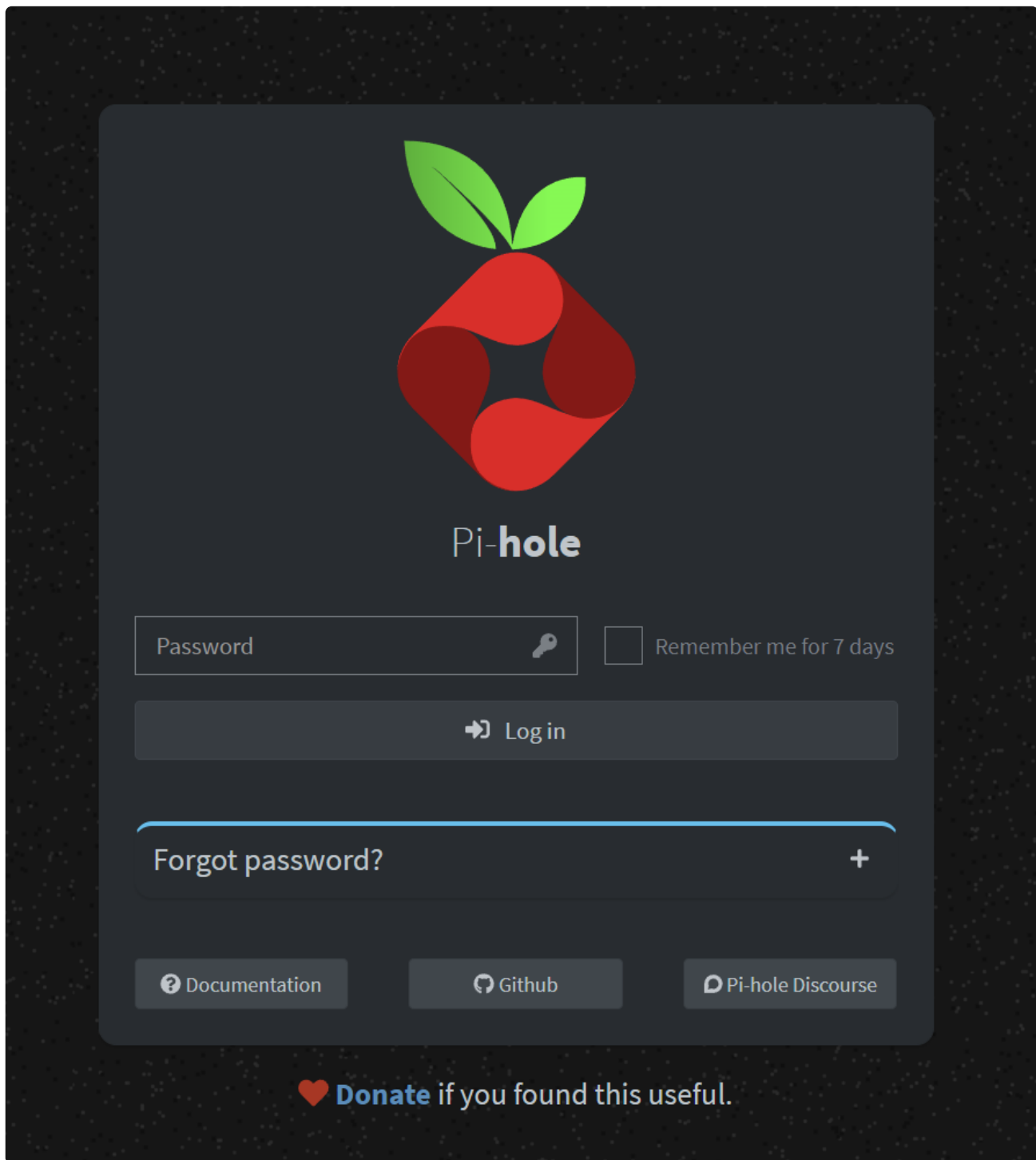
A terminal window screenshot showing the execution of the installation commands. The prompt is 'malenia ~' and the time is '12:33'. The commands entered are 'git clone --depth 1 https://github.com/pi-hole/pi-hole.git Pi-hole', 'cd "Pi-hole/automated install/"', and 'sudo bash basic-install.sh'.

```
malenia ~ 12:33 git clone --depth 1 https://github.com/pi-hole/pi-hole.git Pi-hole
cd "Pi-hole/automated install/"
sudo bash basic-install.sh
```

Once you've cloned the Pi-hole repository and run the installer script, you'll be guided through a series of configuration options. Here's a brief outline of the steps involved:

1. **Choose Interface:** Select the network interface on which you want Pi-hole to listen.
  - For most setups, this will be your primary Ethernet or Wi-Fi interface.
2. **Set Static IP Address:** Pi-hole works best when your Raspberry Pi or device has a static IP address.
  - The installer will help you configure this.
3. **Select Upstream DNS Providers:** You can select from common public DNS providers (e.g., Google, OpenDNS), but if you plan to use Unbound, this will change later.
4. **Block Lists:** Pi-hole comes preconfigured with commonly used ad-blocking lists.
  - You can customize or add additional block lists later if desired.  
([More DNS Blocklists](#))
5. **Web Interface:** You'll be prompted to install the web admin interface, which is highly recommended as it makes managing your Pi-hole much easier.
6. **Query Logging:** Pi-hole can log DNS queries made by devices on your network.
  - You'll be asked whether or not you want to enable logging.

After installation, the web interface can be accessed by entering your device's IP address followed by `/admin` in a web browser (e.g., `http://192.168.1.2/admin`).



Use the password given to you in the installation process to login

- If you missed it or want to change it simply run...

```
sudo pihole -a -p <new password>
```

---

## Setting Up Unbound

Once Pi-hole is installed, you can install and configure **Unbound** to serve as your recursive DNS server.

### 1. Install Unbound:

```
sudo apt-get install unbound
```

### 2. Configure Unbound:

You'll need to configure Unbound to work properly with Pi-hole.

- Create or modify the configuration file at `/etc/unbound/unbound.conf.d/pi-hole.conf` with the following settings:

```
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 sometimes
# 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
```

3. **Update Pi-hole to Use Unbound:** You now need to tell Pi-hole to use your new local Unbound DNS server. In the Pi-hole admin interface:

- Navigate to **Settings** → **DNS**.



- Under **Custom 1 (IPv4)**, enter `127.0.0.1#5335` to point Pi-hole to your local Unbound resolver.

The screenshot shows the Pi-hole web interface with the following details:

- Status:** Active, Load: 0.26 0.13 0.08, Memory usage: 4.5%, Temp: 114.4°F
- Navigation:** Dashboard, Query Log, Long-term Data, Groups, Clients, Domains, Adlists, Disable Blocking, Local DNS, Tools, Settings, Donate
- Upstream DNS Servers Table:**

IPv4	IPv6	Name
<input type="checkbox"/>	<input type="checkbox"/>	Google (ECS, DNSSEC)
<input type="checkbox"/>	<input type="checkbox"/>	OpenDNS (ECS, DNSSEC)
<input type="checkbox"/>	<input type="checkbox"/>	Level3
<input type="checkbox"/>	<input type="checkbox"/>	Comodo
<input type="checkbox"/>	<input type="checkbox"/>	DNS.WATCH (DNSSEC)
<input type="checkbox"/>	<input type="checkbox"/>	Quad9 (filtered, DNSSEC)
<input type="checkbox"/>	<input type="checkbox"/>	Quad9 (unfiltered, no DNSSEC)
<input type="checkbox"/>	<input type="checkbox"/>	Quad9 (filtered, ECS, DNSSEC)
<input type="checkbox"/>	<input type="checkbox"/>	Cloudflare (DNSSEC)
- Custom DNS Settings:**
  - Custom 1 (IPv4):** ☒ 127.0.0.1#5335
  - Custom 2 (IPv4):** ☐
  - Custom 3 (IPv6):** ☐
  - Custom 4 (IPv6):** ☐
- 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:**
    - ☐ Respond only on interface wlan0
    - ☐ Bind only to interface wlan0
    - ☐ Permit all origins

4. **Test Your Setup:** You can verify that Unbound is functioning correctly by running:

```
dig pi-hole.net @127.0.0.1 -p 5335
```

This command should return an IP address for pi-hole.net, indicating that Unbound is resolving DNS queries properly.

```
malenia ~ 13:03 sudo dig pi-hole.net @127.0.0.1 -p 5335

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> pi-hole.net @127.0.0.1 -p 5335
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4394
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 1232
;; QUESTION SECTION:
;pi-hole.net.                IN      A

;; ANSWER SECTION:
pi-hole.net.                 300     IN      A      3.18.136.52

;; Query time: 74 msec
;; SERVER: 127.0.0.1#5335(127.0.0.1) (UDP)
;; WHEN: Sat Sep 21 13:04:00 CDT 2024
;; MSG SIZE rcvd: 56
```

With Pi-hole and Unbound, you now have a powerful, privacy-focused ad-blocking solution. You'll no longer rely on third-party DNS providers, and your network will be shielded from unwanted ads and trackers, all while maintaining high levels of security and control. Additionally, your Pi-hole is now caching DNS responses locally, speeding up subsequent queries and ensuring that your browsing experience remains fast and seamless.

---