Network Guardian: Building a Pi-hole Ad-Blocker on Ubuntu Server

A Project Journey: From Old Laptop to a More Private, Ad-Free Home Network.

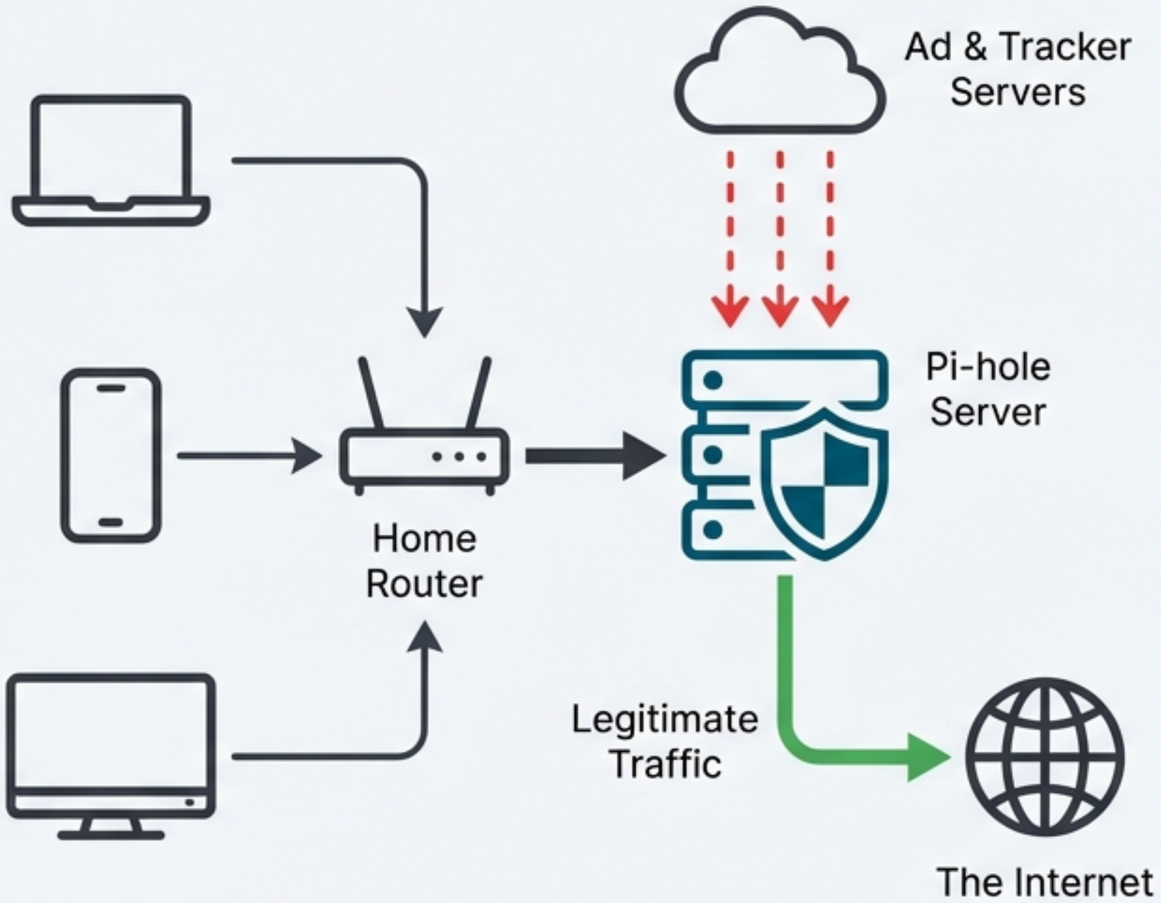
Reclaim Your Network.

The Goal

Convert an old, unused laptop into a dedicated, network-wide ad and tracker-blocking DNS server.

The Payoff

Enhance personal privacy, significantly accelerate web browsing, and gain valuable, hands-on experience with Linux, DNS, and foundational cybersecurity principles.



Assembling the Essentials.

Hardware: An old laptop (any brand will do)

Hardware: 4GB+ USB Drive

Hardware: Home Router

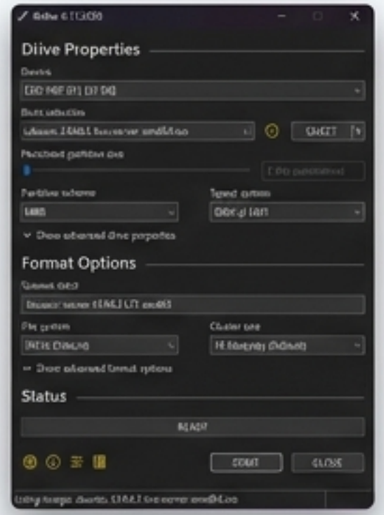
Software: Ubuntu Server 24.04.3 LTS

Software: Rufus (or similar USB imaging tool)

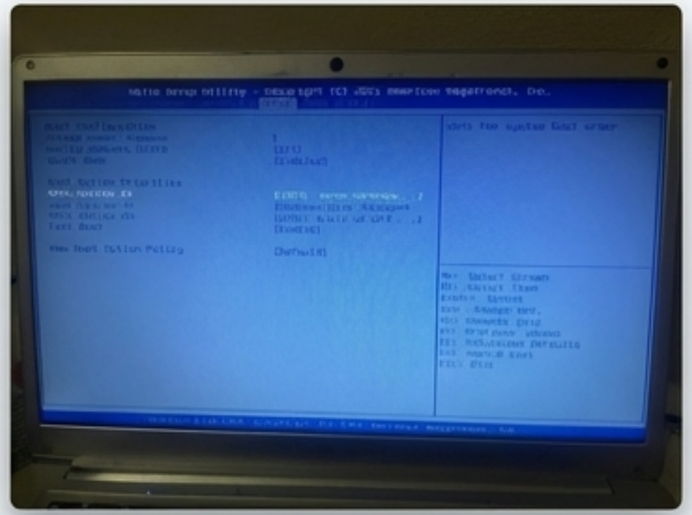
Software: Pi-hole

Phase 1: Forging the Foundation.

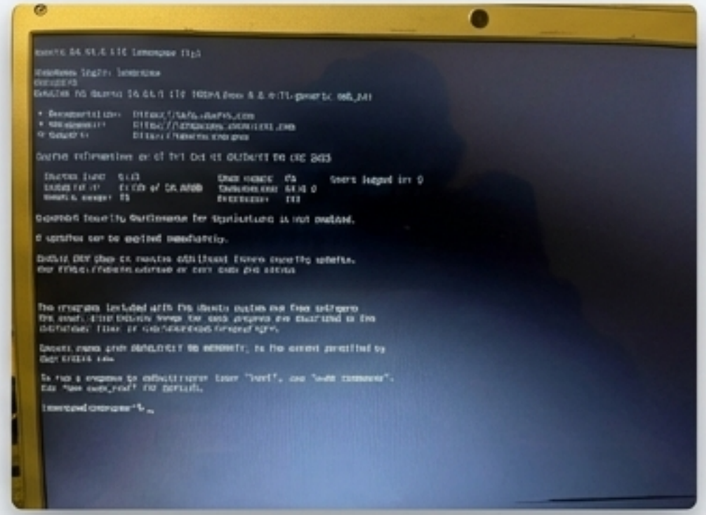
From Bare Metal to Linux Server.



1. Create Bootable USB



2. Configure BIOS



3. Successful Install

> Key Takeaway: Ubuntu Server was chosen for its lightweight, stable, and well-supported nature. It runs well on older hardware and avoids unnecessary desktop software, making it perfect for a dedicated server role.

Phase 2: Establishing a Command Center.

Initial Server Setup and Network Configuration.

1. System Update & Security

The first action was to ensure all packages were current, patching any potential vulnerabilities from the start.

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

2. Network Identification

Connected the server to Wi-Fi and used the command line to find its assigned IP address, which is critical for all subsequent steps.

```
ip a
```

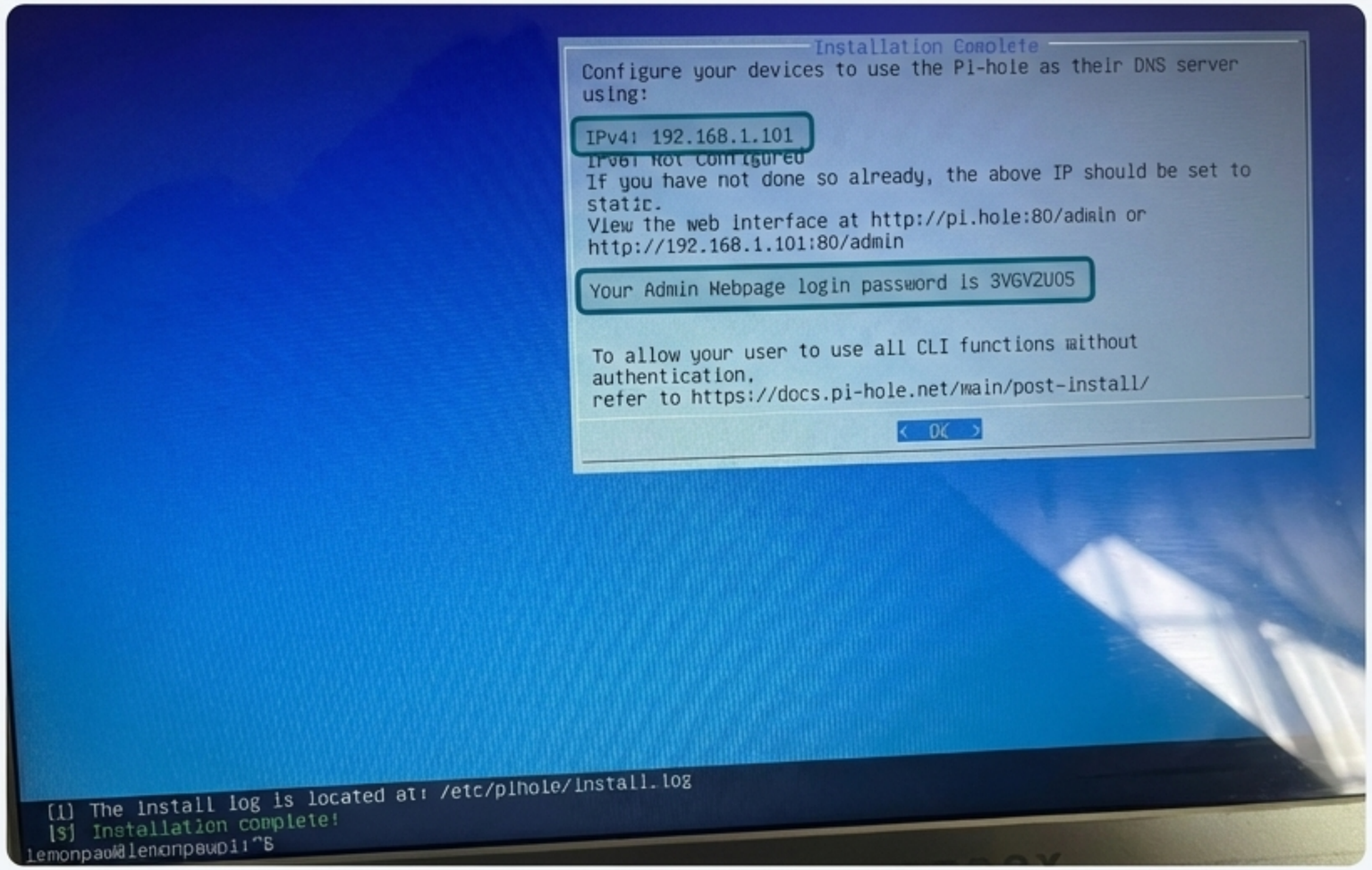
Server IP Address: `192.168.1.101`

```
Reading state information... Done
5 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  libmagickcore-6.q16-7t64 imagemagick libmagickcore-6.q16-7-extra
  imagemagick-6.q16 imagemagick-6-common libmagickwand-6.q16-7t64
Learn more about Ubuntu Pro at https://ubuntu.com/pro
The following upgrades have been deferred due to phasing:
  gir1.2-glib-2.0 landscape-common libglib2.0-0t64 libglib2.0-bin libg
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
lemonpaw@lemonpawpi:~$ _
```

```
lemonpaw@lemonpawpi:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    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 noprefixroute
       valid_lft forever preferred_lft forever
2: wlp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlenec qlen 1000
    link/Ether b8:4d:43:d9:e7:b0:b0 brd ff:ff:ff:ff:ff
    inet 192.168.1.101/24 metric 600 brd 192.168.1.255 scope global
dynamic wlp1s0
       valid_lft 369sec preferred_lft 369sec
    inet6 fe80::ba4d:43ff:fed9:e7b0/64 scope link
       valid_lft forever preferred_lft forever
lemonpaw@lemonpawpi:~$ _
```

Phase 3: Activating the Shield.

Installing and Configuring Pi-hole.



The Core Action

The entire Pi-hole installation was initiated with a single command.

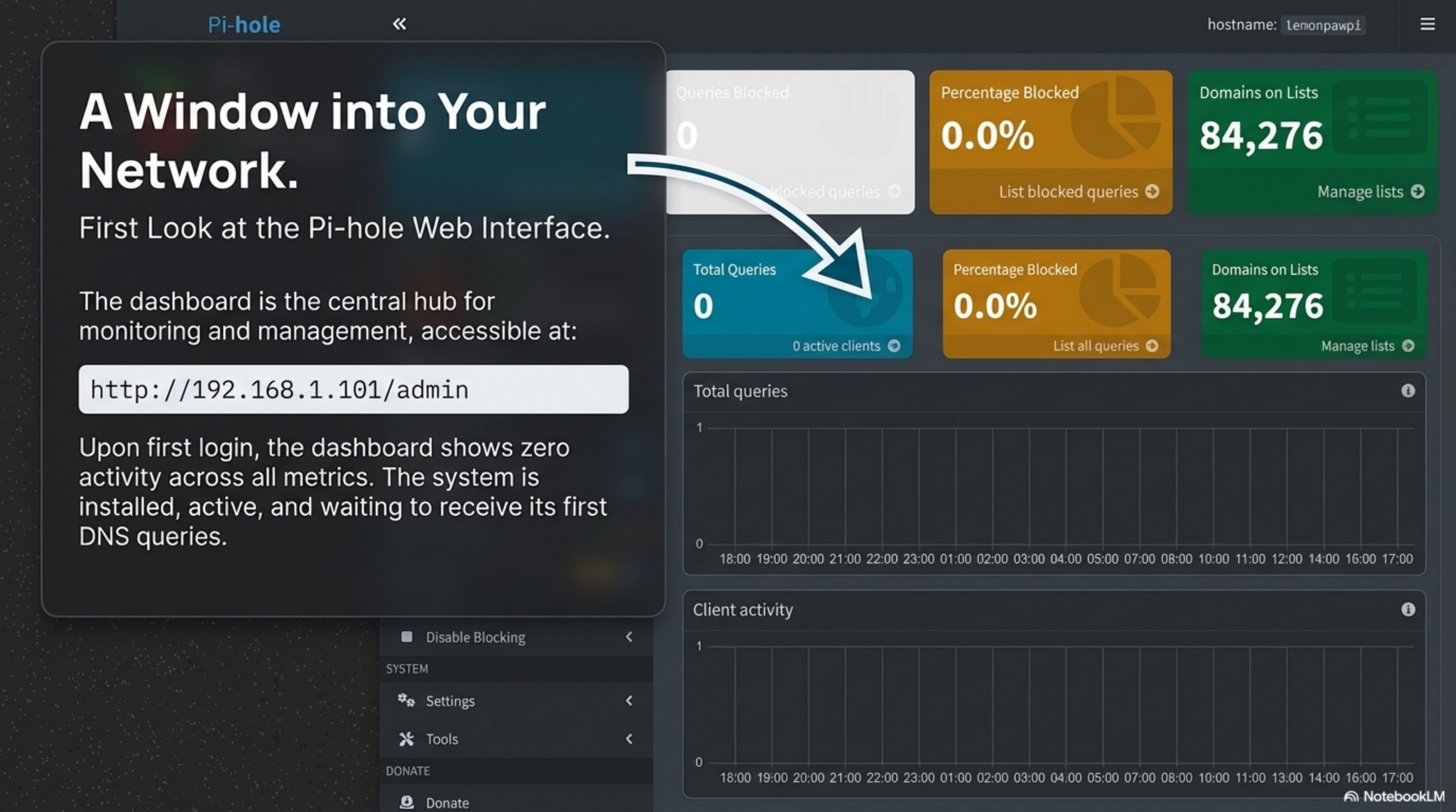
```
curl -sSL https://install.pi-hole.net | bash
```

Key Configuration Choices

- **Network Interface**: Selected the active Wi-Fi interface (wlp1s0).
- **Upstream DNS**: Chose Cloudflare for its speed and privacy.

Critical Outcome

The installation finished by generating the web admin password, which was securely saved.



A Window into Your Network.

First Look at the Pi-hole Web Interface.

The dashboard is the central hub for monitoring and management, accessible at:

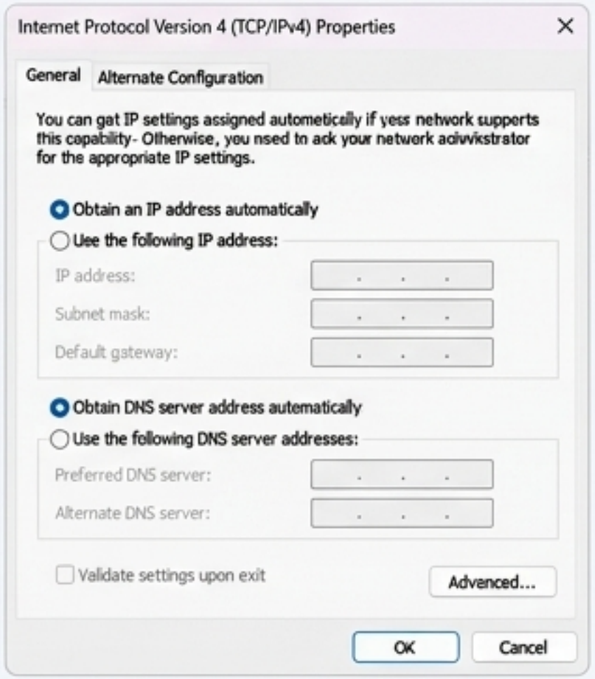
`http://192.168.1.101/admin`

Upon first login, the dashboard shows zero activity across all metrics. The system is installed, active, and waiting to receive its first DNS queries.

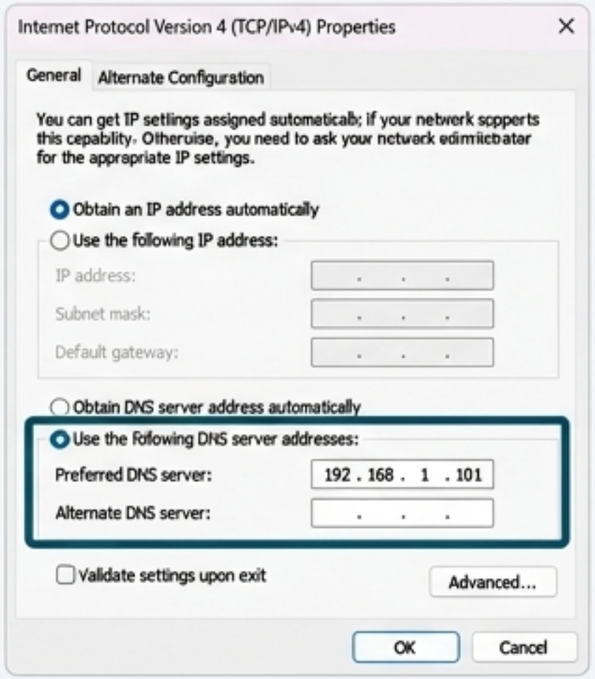
Phase 4: Directing the Traffic.

Pointing Devices to Their New Guardian.

BEFORE



AFTER



On a client device, the network adapter's IPv4 properties were changed from automatic DNS to manually point to the Pi-hole server at 192.168.1.101.

The System is Live.

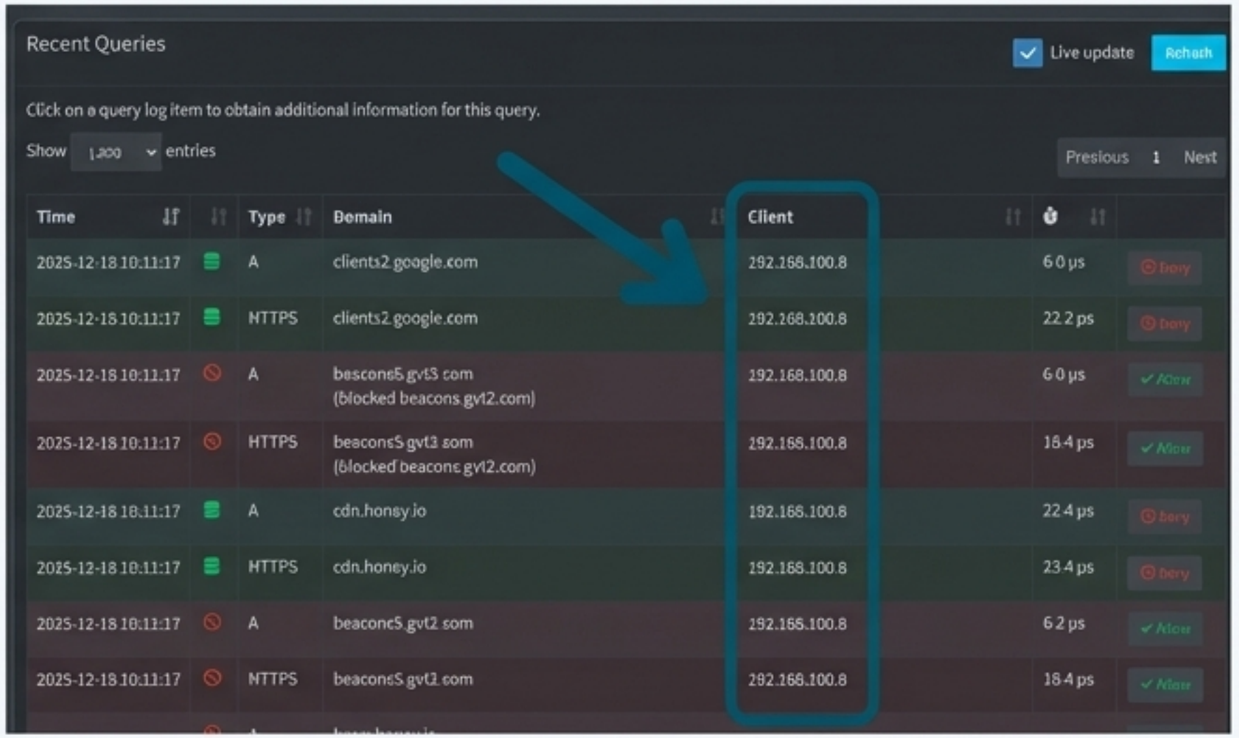
Verifying the First DNS Queries.

Confirmation

Immediately after configuring the client device, DNS requests began appearing in the Pi-hole's Query Log.

What this means

This confirms that the Windows 11 machine is no longer asking the router or ISP for DNS resolution. Instead, all requests are being sent to and processed by the new Pi-hole server.



Every Project Has Its Puzzles.

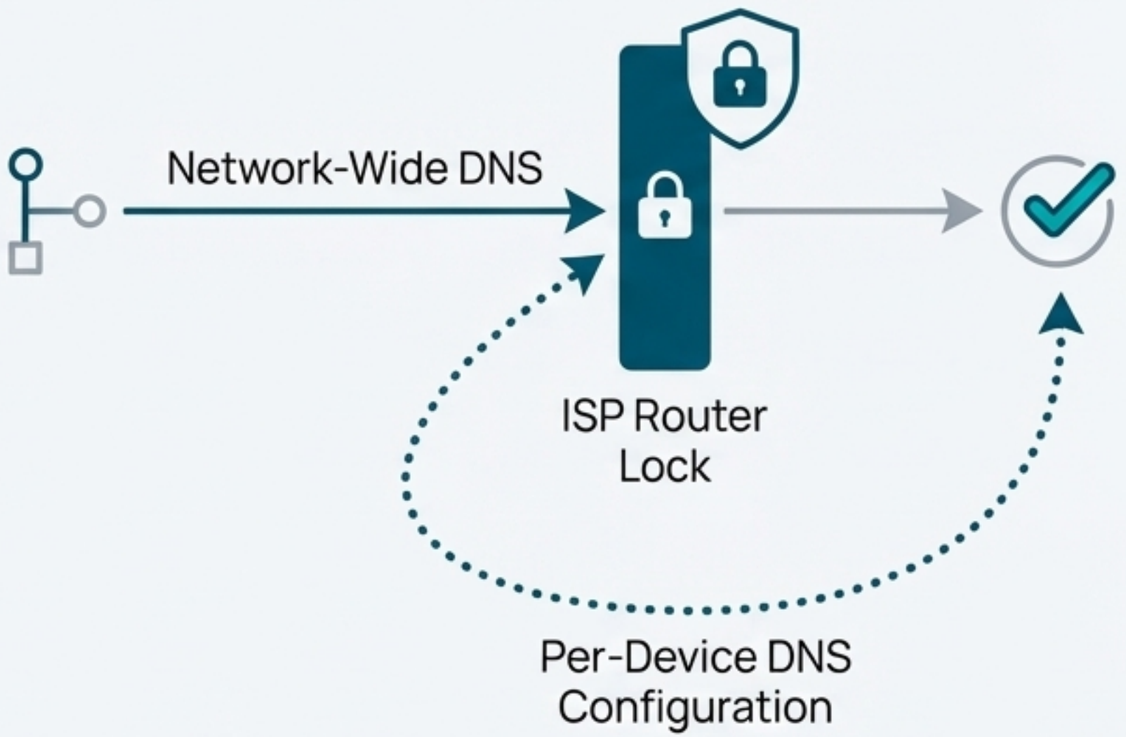
Overcoming Obstacles on the Path to Success

Challenges Encountered

- **Initial Boot Failure:** Required reconfiguring the BIOS boot order.
- **Wi-Fi Driver Issues:** Needed command-line troubleshooting to establish a stable connection.
- **Locked Router Settings:** The ISP-provided router restricted network-wide DNS changes.

The Solution: Adaptability

Instead of being stopped by the router limitation, the strategy shifted. DNS was configured on a per-device basis. This workaround still achieved the primary goal and demonstrated practical problem-solving.



The Victory: A Cleaner, Faster Web.

Seeing the Difference in Real-Time.

Testing on ad-heavy websites confirmed the setup was a success. Pages loaded faster and without intrusive ads or trackers, and the dashboard came alive with proof.

Mission Accomplished.

Project Outcome & Key Learnings

Final Outcome

An old laptop was successfully repurposed into a fully functional, network-enhancing DNS ad-blocker, improving privacy and speed for connected devices.

✓ Improved Privacy ✓ Faster Browsing ✓ Valuable Skills Acquired

Skills Gained

This project provided practical, real-world experience in:

- Linux command-line navigation and package management.
- Core server administration and maintenance.
- DNS principles and network configuration.
- Troubleshooting hardware drivers and locked-down network gear.

The Journey Doesn’t End Here.

Potential Next Steps for Your Home Lab.

Enhance Privacy with `Unbound`

Add a self-hosted, recursive DNS resolver to encrypt your DNS queries, keeping them private even from your ISP.

Modernize with Docker

Run Pi-hole inside a Docker container for easier management, portability, and resource isolation.

Secure Remote Access with a VPN

Configure a VPN server (like WireGuard or OpenVPN) to route your mobile devices' traffic through your Pi-hole from anywhere in the world.

Final Verdict.

“You successfully turned an old laptop into a fully working Pi-hole DNS server... your setup improved network privacy, reduced unwanted ads, and became a solid home lab project you can proudly build on.”

Thank You.

Questions & Answers