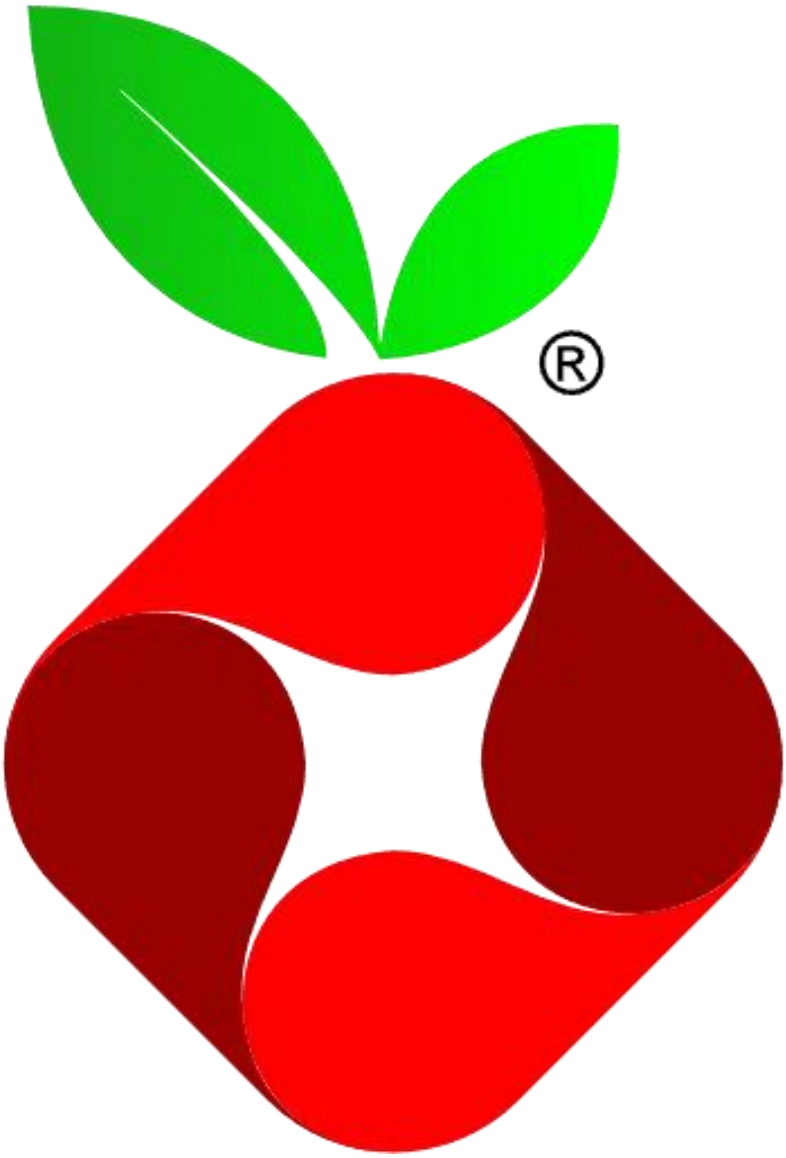
Pi-hole®: An Open-Source Solution for Network Ad Blocking



Originally published April 2024

Revised August 2026

Pi-hole®: An Open-Source Solution for Network Ad Blocking

Open-Source Enthusiasts and Home Network Users

Document: Pi-hole®: An Open-Source Solution for Network Ad Blocking

Author: Dustin Smith

Audience: Open-Source Enthusiasts and Home Network Users

Original Publication: April 30, 2024

Revision: August 2026

Version: 2.0

Revision Note:
This version adds common Pi-hole directories and updates terminology, commands, technical descriptions, and references to align with current Pi-hole documentation. Screenshots captured in 2024 may differ from the current Pi-hole interface.

Executive Summary

Pi-hole is a lightweight, open-source Domain Name System (DNS) sinkhole that blocks requests to configured domains across a network. Originally developed for Raspberry Pi hardware, Pi-hole can run on multiple supported Linux distributions and deployment environments, including physical computers, virtual machines, and containers. By filtering selected advertising, tracking, and unwanted domains at the DNS level, Pi-hole can reduce unnecessary network requests and provide visibility into DNS activity. The Pi-hole software can be installed through the Linux terminal using automated scripts, repository cloning, or Docker containers. These options provide flexibility to fit various system configurations and user preferences.

Pi-hole features a user-friendly web interface for monitoring and managing network traffic in real-time. Users manually configure their devices to use Pi-hole as a local Domain Name System (DNS) server. The server manages domains and blocks unwanted traffic by filtering network queries with domain management tools: Subscribed Lists, Denylists, and Allowlists. Users manage which domains are blocked, providing additional DNS filtering, control, and network visibility. Regular community developer and user updates maintain Pi-hole's security and performance, keeping the network safe and efficient. Pi-hole provides a flexible method for network-wide DNS filtering and monitoring, supported by an active open-source community. Its compatibility with various devices and systems, along with strong community support, makes it an adaptable and reliable choice for both personal and professional use.

Key Features

- ***Network-wide Ad Blocking:*** Operates at the DNS level to block queries for domains included in configured blocking rules and subscribed lists, reducing unwanted requests across configured devices.
- ***Customizable Control:*** Users can deny or allow specific domains and organize filtering rules for different clients or groups.Pi-hole Web Interface Dashboard (2024 Interface)
- ***Community-driven:*** Supported by an active community that contributes to its continuous development and feature enhancements.

Community Support

Pi-hole is supported by an active online community that helps in development and maintenance. Additional resources, discussion forums, and documentation are available on the official Pi-hole website: https://pi-hole.net/

Contents

List of Figures

Overview

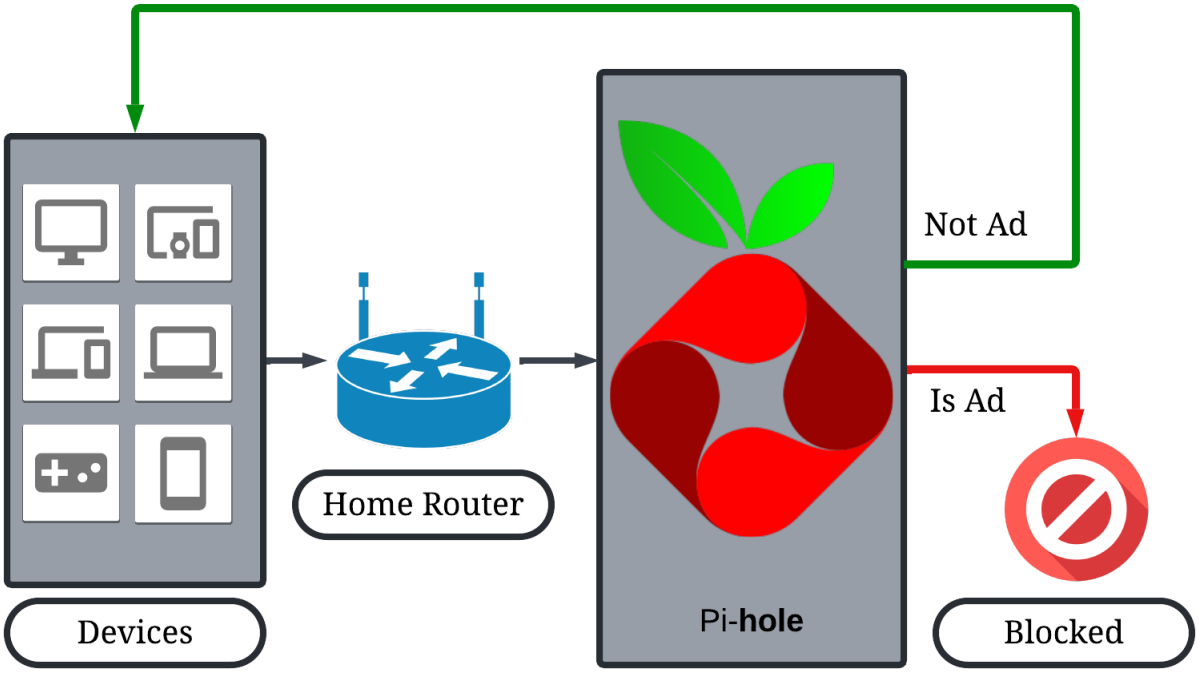
Pi-hole is free, open-source network software that functions as a DNS sinkhole. It evaluates DNS requests from configured network devices and blocks requests that match configured filtering rules. Originally developed for Raspberry Pi hardware, Pi-hole now supports several Linux-based systems and deployment environments. Originally, Pi-hole was made for the Raspberry Pi, but now it works on many operating systems and devices, such as personal computers and virtual machines. Additionally, there is a vibrant community of developers and users who continually improve and update Pi-hole's capabilities.

Benefits of Pi-hole

Blocking unwanted advertisements can speed up your network, reduce bandwidth usage, and protect you from malicious web traffic. Unlike traditional ad blockers that work as browser extensions or standalone apps, Pi-hole functions at the Domain Name System (DNS) level, intercepting requests from known ad-serving domains and preventing them from reaching your configured network devices.

This network-wide blocking capability means that all devices connected to your network, from smartphones and tablets to smart TVs, can benefit from reduced advertising, tracking, and other unwanted DNS requests. Reducing unnecessary domain requests may decrease network traffic and improve the perceived responsiveness of some browsing experiences. Pi-hole also allows users to block or allow specific web addresses, offering additional control and visibility over DNS activity that monitors individual queries and identifies unwanted network traffic.

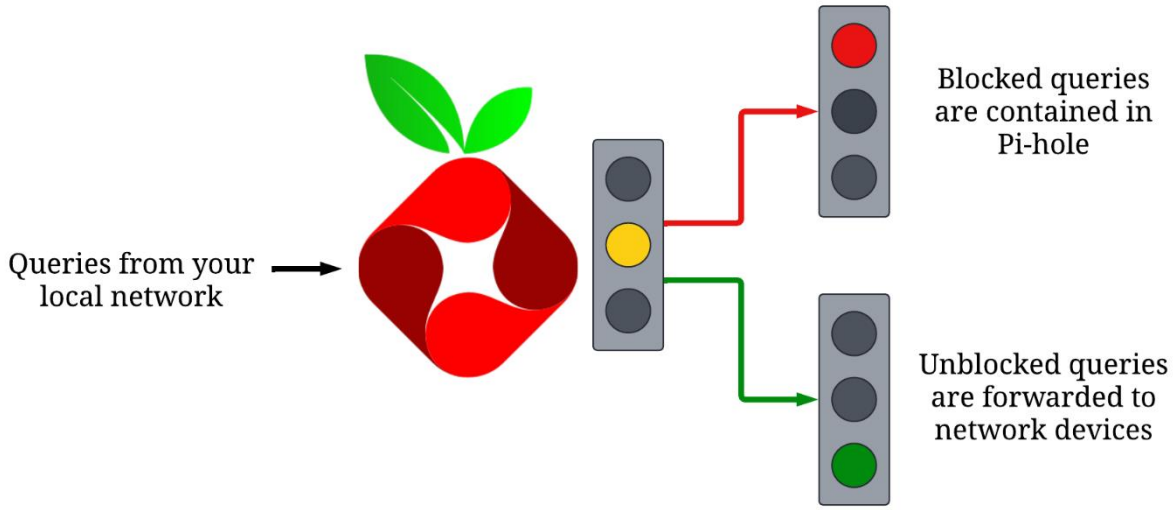
Figure 1: *Pi-hole Diagram*



Pi-hole as a DNS Server

Pi-hole evaluates DNS queries from configured network clients before returning or forwarding a DNS response. Queries that match configured deny rules or subscribed blocking lists are blocked according to Pi-hole's configured blocking behavior. Queries that are allowed are resolved from cache or forwarded to an upstream DNS server. Allowlist entries can override applicable blocking rules so required domains continue to resolve normally.

Figure 2: Pi-hole Filtering



System Requirements

Hardware Requirements

Pi-hole can be installed on any device with a Linux-based operating system:

- ***Raspberry Pi*:** Ideal for users seeking a dedicated, low-power device to run Pi-hole continuously without significant energy consumption.
- ***Personal Computer*:** Use an old desktop or laptop computer to take advantage of the more robust hardware capabilities for monitoring network traffic.
- ***Virtual Machine*:** Pi-hole can run within a virtual machine, allowing you to direct and monitor network traffic without hardware.

Confirm that your device meets the following requirements before installing Pi-hole:

- ***Storage Space*:** 2GB minimum, 4GB recommended.
- ***Random Access Memory (RAM)***: 512MB minimum.

Software Requirements

Pi-hole supports actively maintained Linux distributions that use systemd or sysvinit. Unsupported operating systems, versions, or configurations may not install or function correctly. Keep both the operating system and Pi-hole software updated to maintain security and compatibility.

For the latest supported operating systems, visit the official Pi-hole documentation: https://docs.pi-hole.net/main/prerequisites/#supported-operating-systems

Figure 3: *Supported Operating Systems*

Distribution	**Architecture**
Alpine	x86_64 / armv6 / armv7 / armv8 / riscv64
Armbian OS	ARM / x86_64 / riscv64
CentOS Stream	X86_64
Debian	ARM / x86_64 / i386
Fedora	ARM / x86_64
Raspberry Pi OS	ARM
Ubuntu	ARM / x86_64

Installation Methods

Pi-hole offers four different installation methods. Confirm that your device uses a Linux-based operating system and is connected to the Internet. Wired Ethernet connections are preferred for software installation. Choose an installation method and follow the steps to obtain login credentials.

Prerequisites

To install and configure Pi-hole, you need administrative privileges, which can be obtained through "sudo" or "root" permissions.

- ***Sudo Permission:*** This permission allows you to run commands with administrative privileges. Your Linux user account must be listed in the '**/etc/sudoers**' file. Check if your user account can use the 'sudo' by typing the following command in your terminal:
    - **sudo whoami**
- ***Root Permission:*** Alternatively, you can log in as the root user to gain administrative privileges. Type the following command into the terminal:
    - **su -**

Automated Installation

After you have signed in with administrative privileges, you can download the '**curl**' command tool by opening your Linux terminal and typing:

- **sudo apt install curl**

Executing the Pi-hole script manages necessary dependencies by configuring your network settings so Pi-hole can function in the Linux environment. **Note:** Piping a remote script directly to Bash executes the script without reviewing it first. Users who prefer to inspect the installation code should use the repository-cloning method. Enter the following command to initiate automated installation.

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

Cloning Repository

Installing Pi-hole by cloning the repository and running the setup locally allows you to review the code before installation. Cloning this repository provides the software specifics, allowing customizations that better fit specific needs. Enter the following command sequence to clone the Pi-hole repository.

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

Docker Installation

Create an isolated containerized environment for Pi-hole by using Docker. This installation method allows Pi-hole to become platform-independent and function on a wider variety of devices. Follow the current Pi-hole Docker deployment documentation and use the official container image:

- **pihole/pihole**

Manual Installation

Manually downloading and running the installer gives you more control over the installation process. Use this method to install a specific version of Pi-hole rather than the latest version, which is fetched automatically with the automated and cloning methods. To install Pi-hole manually, enter the following command sequence in the Linux terminal.

- **wget -O basic-install.sh https://install.pi-hole.net**
- **sudo bash basic-install.sh**

User Credentials

After installation, Pi-hole provides credentials for managing network traffic after installation is complete. Figure 4 is a screenshot of an example user's credentials; note that your Internet Protocol version 4 (IPv4) address will be different for each device, and the web interface password is unique to each Pi-hole.

Change these credentials by entering the command in the Linux terminal:

- **pihole setpassword**

Figure 4: *User Credentials*

Update Pi-hole

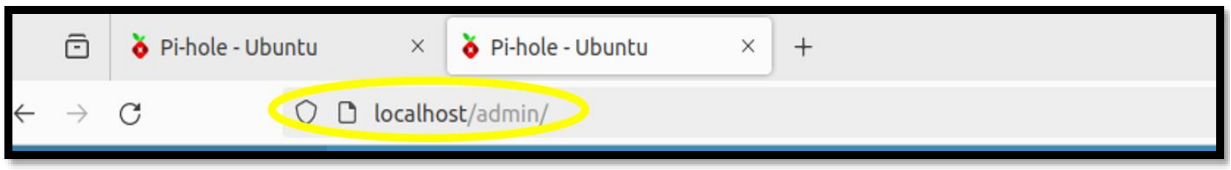
Review available Pi-hole updates regularly and read the release notes before performing major upgrades. Update Pi-hole by entering the following **c**ommand into the terminal:

- **pihole -up**

Configuring Pi-hole

Access the Web Interface

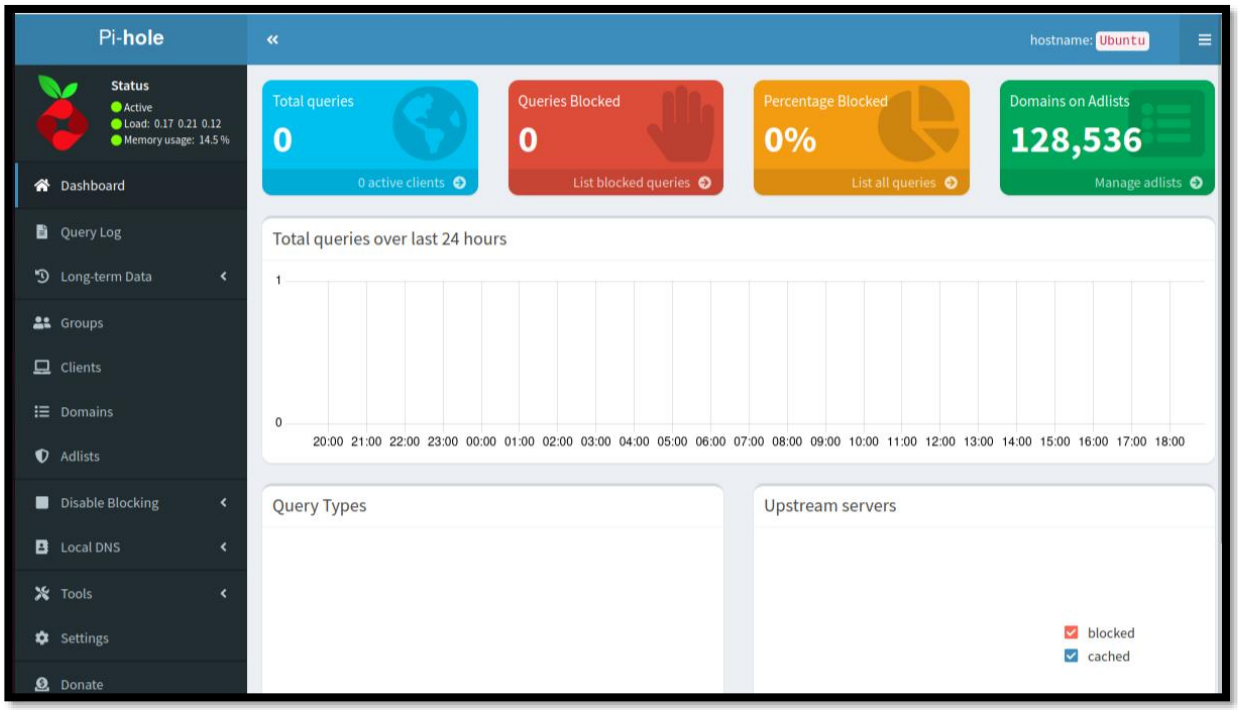
Pi-hole offers a web interface for monitoring and managing network traffic with real-time data. To access the Pi-hole web interface from a device on the configured network, open a browser and navigate to **https://pi.hole/admin/** or the Pi-hole host's IP address followed by **/admin/**. To access the interface, on your Pi-hole device, open a web browser, select the address bar, and type **localhost/admin/**. Log into the Pi-hole GUI web interface by entering the password for your Pi-hole.



Web Interface Dashboard

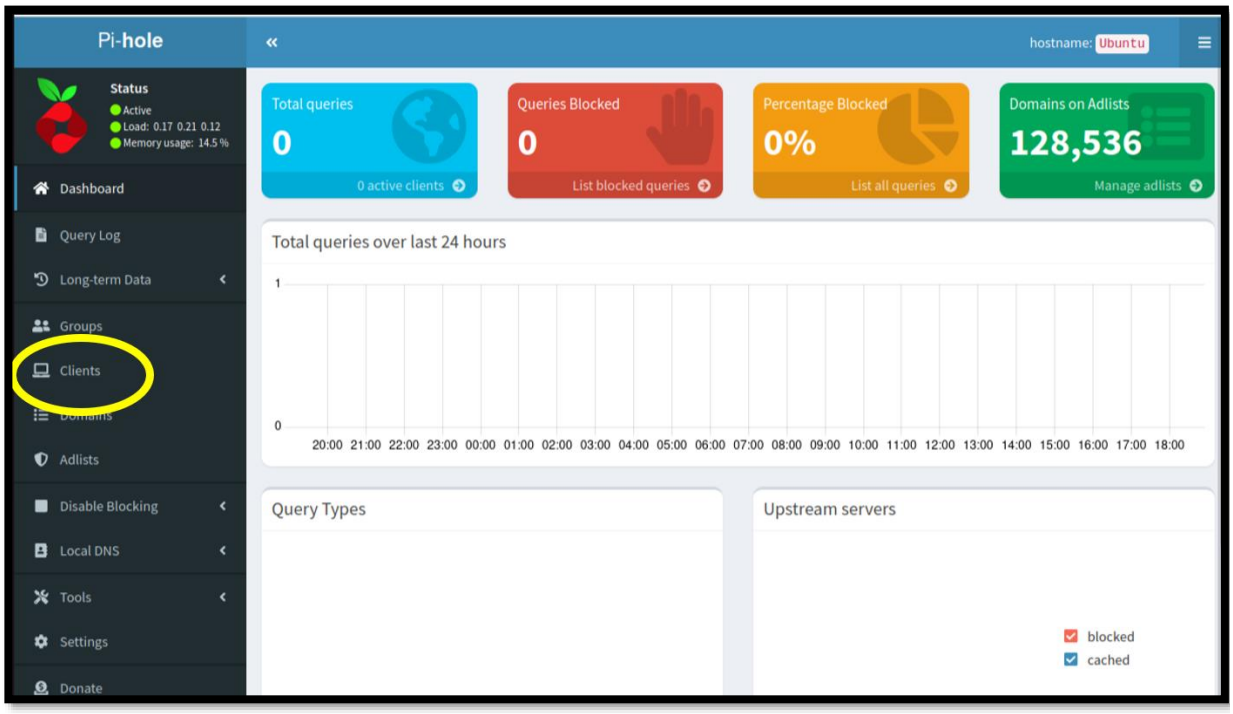
The web interface provides DNS query statistics, filtering status, client information, and configuration controls.

Figure 5: Pi-hole Web Interface Dashboard (2024 Interface)

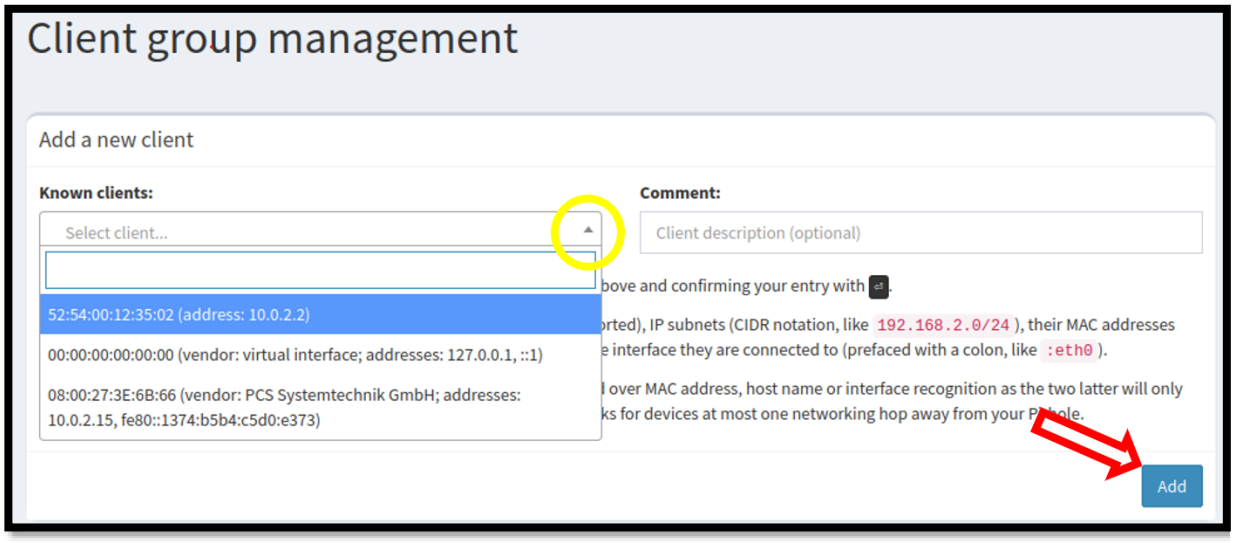


Adding Client Devices

Add client devices to the Pi-hole by selecting the "Clients" tab on the side navigation panel.



Select a device IP address from the Known Clients drop-down menu or enter the device's IP address manually and click the "Add" button. The client can associate with groups and filtering rules within Pi-hole. The device must still be configured to use Pi-hole for DNS resolution.



Configuring Client Devices

Configuration Settings

Your Pi-hole host requires a stable IP address, either statically (static IP address) configured or reserved through Dynamic Host Configuration Protocol (DHCP). If the Pi-hole IP address changes, the DNS configuration settings of your network devices must be updated to continue allowing, blocking, and monitoring network traffic with Pi-hole. Each network device can be configured to use Pi-hole as its DNS network server. The process for configuring individual devices varies depending on product type, manufacturer, and operating system.

Pi-hole IP address

Identify your Pi-hole IP address by typing the following command into your Linux terminal:

- **hostname -I**

Configure Local Network Devices

Find the manual network connection setting option on your client device and enter the Pi-hole IP address (e.g., 10.0.2.15) into the primary DNS server text input field. Repeat this process for each device you want to use with Pi-hole. Note that some devices or applications may need to be restarted for primary DNS changes to take effect. Consult the operating system or device manufacturer's current documentation for instructions on manually configuring DNS servers.

Pi-hole and Smart Home Devices

Smart home devices, such as smart TVs and streaming devices, can also benefit from Pi-hole's ad-blocking capabilities. However, some devices use hard-coded or encrypted DNS services and may not honor locally configured DNS settings. For example, blocking domains related to your smart TV manufacturer might prevent it from accessing features or processing voice commands. Some smart home devices may not have an option to configure DNS settings manually. In these cases, you must edit your router's DHCP settings and configure Pi-hole as your home router's DNS server.

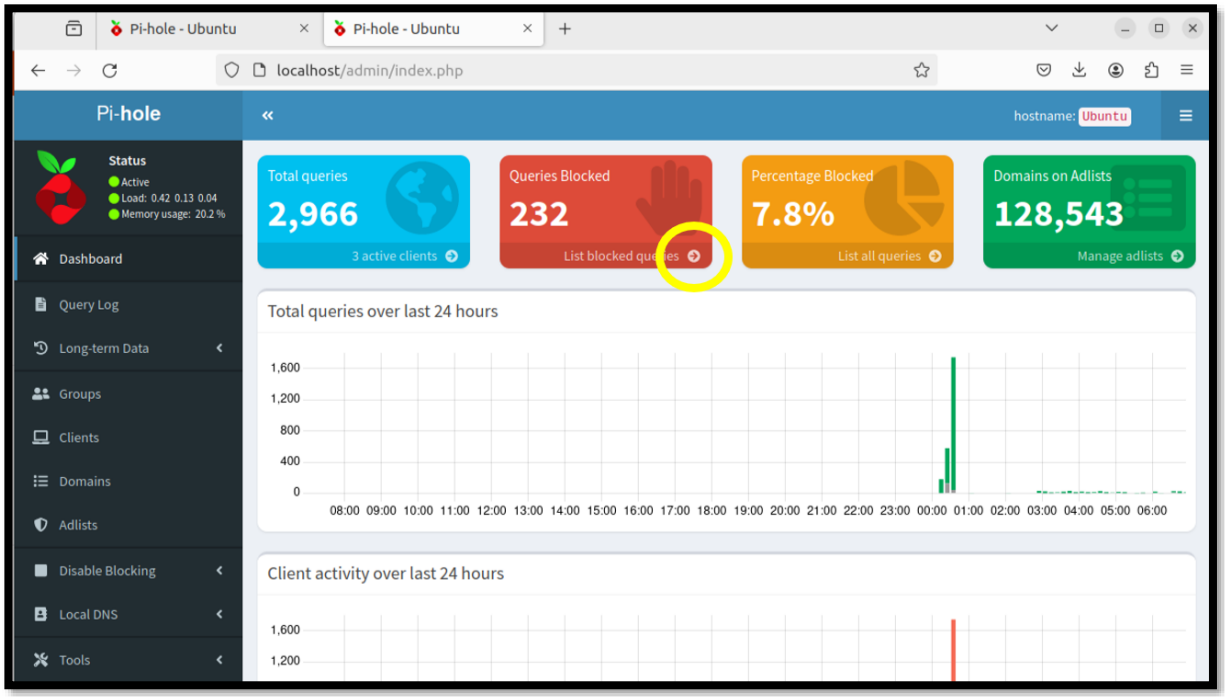
Configuring the Local DNS Server

Configure Network Router

Configuring your router's DHCP service to advertise Pi-hole as the DNS server can automatically direct compatible network clients to Pi-hole for DNS resolution. You can set Pi-hole as the DNS server for your entire network, but this process varies depending on your router's make and model. Enter the router's IP address into the address bar of a web browser to access your router configuration dashboard. Consult the router's manual for access instructions and login credentials.

Test Pi-hole Configuration

Access the web interface dashboard on your Pi-hole device and select the "Queries Blocked" tab to view the domains Pi-hole is blocking.

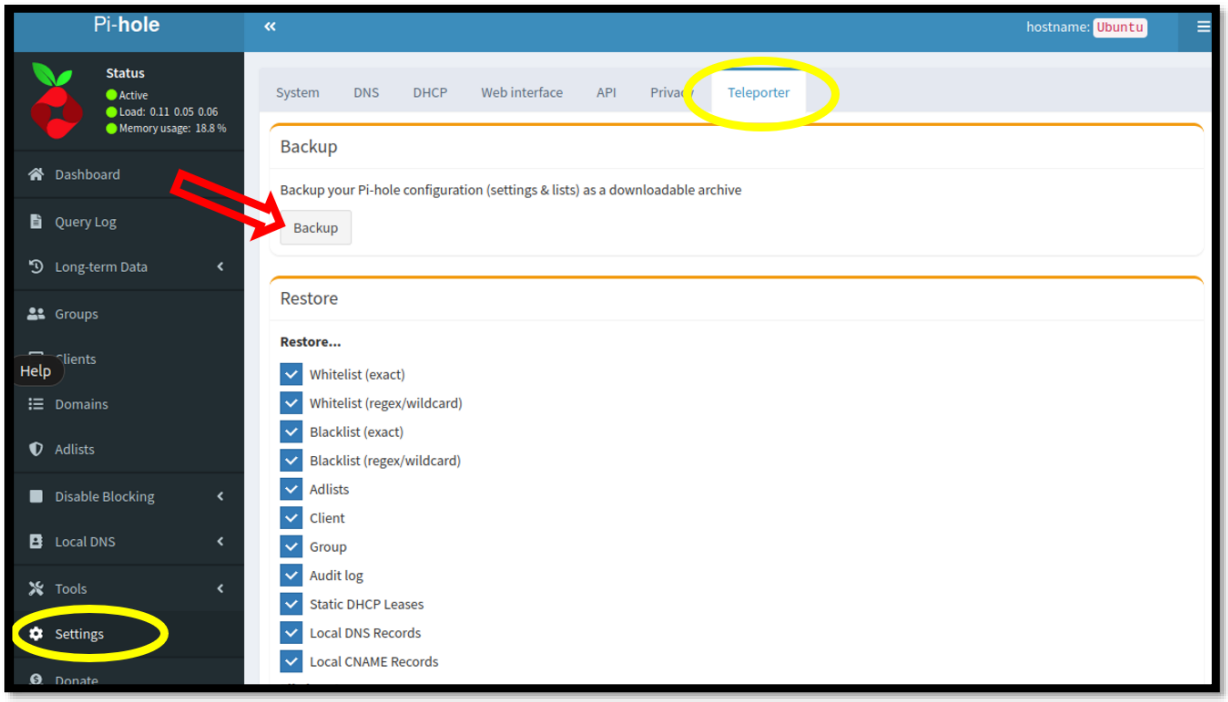


Verify Pi-hole is Working

To verify that Pi-hole is working correctly and blocking domain traffic on your network, open the Pi-hole Query Log and verify that DNS requests from the configured client appear. Then query a domain known to exist in one of your configured blocking lists and confirm that Pi-hole reports the request as blocked.

Backing up Pi-hole Configuration

After establishing that Pi-hole is blocking domains on your network, you should create a backup using the Teleporter backup/export function available in the web interface. Store the exported configuration file in a secure location separate from the Pi-hole host. Visit the Pi-hole web interface, navigate to the “Teleporter” section of the "Settings" tab, and click on the "Backup" button.



Note that you can also create a backup of your Pi-hole configuration through your Linux terminal by entering the following command:

- **pihole -a -t**

Restoring Pi-hole Configuration

After the backup process is complete, the configuration can be restored and transferred to other devices. You can restore a configuration through the Pi-hole web interface by navigating to the “Teleporter” section of the "Settings" tab, scrolling down, clicking the “Restore” button, and selecting the backup file from your Linux directory. Click "Open," and confirm that you want to restore the selected configuration. Restart your Pi-hole after restoring the configuration by typing the following command into your Linux terminal:

- **pihole reloaddn**

Domain Management

Denylist

The denylist blocks DNS resolution for specific configured domains. The Denylist domain management tool gives you complete control over the domains that are blocked instead of relying on an ad list to block unwanted domain traffic.

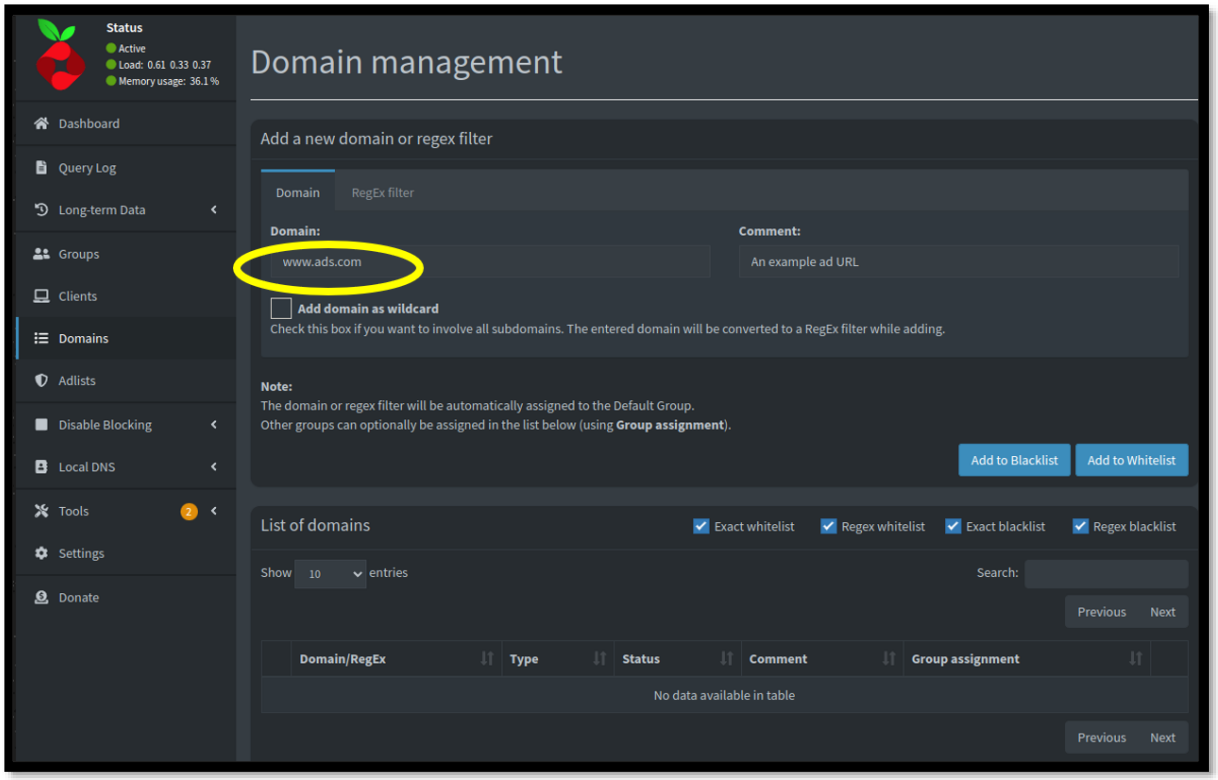
Allowlist

The allowlist permits specified domains to resolve even when they would otherwise match an applicable blocking rule. Adding a specific domain URL to the Allowlist will allow all traffic from that domain to communicate with Pi-hole devices. Smart home and other devices sometimes rely on specific domains to function. In these cases, those domains must be added to the Allowlist.

Managing Allowlist and Denylist Content

Locate the ‘Domains’ tab in the sidebar navigation panel to access the Domain Management page. Add domain URLs directly to the Denylist or Allowlist by entering the website address under the ‘Domain’ tab, then click ‘Add to Denylist’ to block the address or click ‘Add to Allowlist’ to allow the website to communicate with client devices. You can also enable, disable, or delete Denylist and Allowlist items from the lists.

Figure 6: Domain Management Interface (2024)



Regular expressions (RegEx) allow Pi-hole to match domain-name patterns rather than requiring an exact domain entry. RegEx rules can be used to create flexible deny or allow rules for groups of domains that share predictable patterns. Add a 'RegEx' filter to the Denylist or Allowlist by selecting the 'RegEx' tab and following the previous domain list management process.

For more detailed information on RegEx filters, visit the official Pi-hole website: https://docs.pi-hole.net/regex/overview/

Updating the Gravity Database

The Gravity database contains domains compiled from configured subscribed lists. Update the Gravity database monthly through the web interface by selecting the dropdown menu on the "Tools" tab and clicking "Update Gravity."

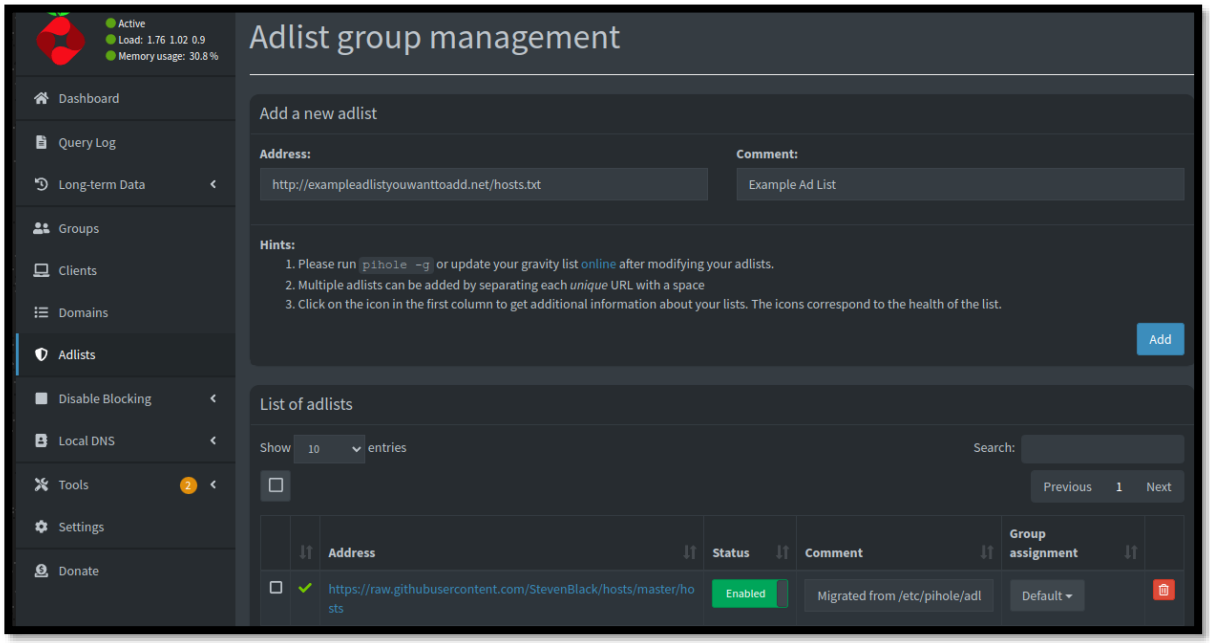
Note that the Gravity database can also be updated through your Linux terminal by entering the command:

- **pihole updateGravity**
- *Legacy command:* **pihole -g**

Subscribed Lists

Subscribed lists provide Pi-hole with external sources of domains to process for filtering. The number of domains available to Pi-hole varies according to the lists configured by the administrator and changes as those lists are updated. The Pi-hole installation contains a community-made Subscribed List. Locate community-made Subscribed Lists in discussion forums and Pi-hole community posts for updated lists that target specific domains containing malware, tracking telemetry, and adult content. Add text file links by selecting the “Subscribed Lists” tab on the side navigation menu. From here, you can remove, disable, or enable a newly added Subscribed List.

Figure 7: Subscribed List Management Interface (2024)



Configure Subscribed Blocking Lists

Locate community-made Subscribed Lists through discussion forums, enthusiast groups, and Pi-hole community posts. Search for a current updated list tailored to your network's threat model. Only subscribe to lists from sources you trust and that are actively maintained. Large or aggressive lists can increase false positives and may interfere with legitimate services.

Create a Personalized Subscribed List

The Subscribed Lists text files follow a very straightforward format. Each file contains the domains you wish to block, and these lists can contain multiple domains. Comments can be inserted into the Subscribed List with a # character. “Commenting” a domain temporarily allows the domain traffic to communicate with your network devices. Alternate format options, like Ad Guard, can also be personalized.

Troubleshooting

Even with regular updates and maintenance, you may encounter issues with your Pi-hole. Always check that your Pi-hole device is powered on, connected to your network, and uses a static IP address.

Ads Not Being Blocked

If you notice that ads are not blocked, ensure that your device is using Pi-hole as its DNS server. Check the network settings on your devices or router to confirm that Pi-hole's IP address is listed as the primary DNS server (p. 10).

Verify that the appropriate Subscribed Lists are enabled in Pi-hole's settings. Navigate to the "Subscribed Lists" tab in the web interface and check that the desired Subscribed Lists are enabled. Open your Linux terminal and type the following command to clear the cache and force Pi-hole to re-fetch the blocklists:

- **pihole reloaddnss**

Unable to Access the Web Interface

1. Connect a monitor and keyboard directly to your Pi-hole device. Verify that you are using the correct IP address to access the web interface (p. 10). The Pi-hole device's IP address can be viewed through your Linux terminal by typing the command:

- **hostname -I**

2. From another network device, verify that the Pi-hole host responds to its IP address and that the Pi-hole service is running with the command:

- **pihole status**

Website Stops Working

Open the web interface and click on the ‘Disable Blocking’ button to disable blocking for 30 seconds. During this time window, visit the problematic domain again. If it works, it is an issue with a Subscribed List or Denylist. If not, the website in question may be experiencing server issues. If it is, in fact, an issue with a Subscribed List or Denylist, go to the “Domains” tab and add the problematic domain to the Allowlist. This will allow web traffic from that domain to communicate across your network (p.13).

Slow Network Performance

In some cases, Pi-hole may cause slower network speeds, particularly if it is blocking a large number of domains. To mitigate this issue, consider reviewing your Subscribed Lists and remove any that are unnecessarily broad, duplicated, or no longer maintained. Stick to well-known, reputable blocklists. Also, review your Denylist domains to remove legitimate domains that are being blocked unnecessarily in the "Domains" tab of the web interface (p.13).

Debugging

The debugger gathers information about the Pi-hole installation and can generate diagnostic information for use when requesting support. For persistent problems, Pi-hole also provides a built-in diagnostic utility:

- **pihole debug**

Community Support

Pi-hole is supported by an active open-source community of developers, contributors, and users. If you encounter an issue that is not resolved by the troubleshooting guidance in this document, consult the official Pi-hole documentation and community forum for current installation, configuration, and troubleshooting information.

Before creating a new support post, search existing discussions for similar issues and review the most recent Pi-hole documentation. When requesting assistance, include relevant information such as your Pi-hole version, operating system, symptoms, and troubleshooting steps already completed. Avoid sharing passwords, private network information, or other sensitive data.

Additional resources are available through the official Pi-hole documentation, community forum, and GitHub repositories. For comprehensive source documentation and resources, visit the official Pi-hole website: https://docs.pi-hole.net/main/basic-install/

Common Pi-hole Directories

The following Linux directories are commonly used by Pi-hole and may be useful when troubleshooting or reviewing a configuration.

Directory	Purpose
/etc/pihole/	Primary Pi-hole configuration and database directory.
/var/log/pihole/	Stores Pi-hole DNS, FTL, and web server logs.
/opt/pihole/	Contains Pi-hole management and supporting scripts.
/usr/local/bin/	Contains the primary **pihole** command-line utility.
/etc/.pihole/	Contains installation, repair, and uninstall resources used by Pi-hole.

Important files within **/etc/pihole/** include **pihole.toml**, the primary Pi-hole FTL configuration file; **gravity.db**, which stores filtering and domain-management data; and **pihole-FTL.db**, which stores long-term query information.

Whenever possible, modify Pi-hole settings through the web interface, command-line interface, or API rather than editing configuration files or databases directly.

References

Ashley. (2024, March 11). *The best pi-hole blocklists (2024)*. Avoid The Hack. Retrieved from https://avoidthehack.com/best-pihole-blocklists

Domain name system. *Domain Name System* | National Telecommunications and Information Administration. (2023, May 1). Retrieved from https://www.ntia.gov/category/domain-name-system

Gus. (2022, September 13). *How to backup your raspberry pi SD card*. Pi My Life Up. Retrieved from https://pimylifeup.com/backup-raspberry-pi/

Jacob. Salmela. (2017, May 15). *How do I troubleshoot problems with my pi-hole?*. Pi. Retrieved from https://discourse.pi-hole.net/t/how-do-i-troubleshoot-problems-with-my-pi-hole/3170

Pi-hole. (2026, August 12). *Group management*. Retrieved from https://docs.pi-hole.net/group_management/

Pi-hole. (2026, June 12). *Installation*. Retrieved from https://docs.pi-hole.net/main/basic-install/

Pi-Hole (2026, August 6). *Overview*. Retrieved from https://docs.pi-hole.net/

Pi-Hole. (2022, December 3). *Pi-hole/PI-hole: A black hole for internet advertisements*. GitHub. Retrieved from https://github.com/pi-hole/pi-hole

Pi-Hole. (n.d.). *Pi-hole/docker-pi-hole: Pi-hole in a Docker container*. GitHub. Retrieved from https://github.com/pi-hole/docker-pi-hole

Pi-hole. (2026, August 12). *Pi-hole regular expressions tutorial*. Retrieved from https://docs.pi-hole.net/regex/tutorial/

Pi-hole. (2026, June 9). *Prerequisites*. Retrieved from https://docs.pi-hole.net/main/prerequisites/

Pi-hole. (2025, May 3). *Updating*. Retrieved from https://docs.pi-hole.net/main/update/

Pi-hole. (2025, August 11). *The pihole command*. Retrieved from https://docs.pi-hole.net/core/pihole-command/

Glossary

Term	Definition
ARM	Advanced RISC Machine processing architecture is the instruction set that tells your CPU what to do.
DNS	Domain Name System. A distributed naming system that translates domain names, such as *example.com*, into IP addresses and other DNS records used by networked devices.
DNS sinkhole	A DNS server that blocks or redirects requests for specified domains instead of resolving them normally.
IP	Internet Protocol is a set of rules that govern the format of data sent through the Internet and local networks. Each device and website has an IP address that forms its unique identity.
LAN	A Local Area Network is a network that connects computers and devices within a limited geographic area, such as a home, school, office building, or closely situated group of buildings.
Linux terminal	A text-based user interface that allows users to interact with Linux operating systems.
Network device	A computer, phone, server, router, smart device, or other electronic device that communicates over a network.
RAM	Random Access Memory stores data temporarily while the device is running. RAM allows data to be read and written at the same time.
root	The Linux superuser account with unrestricted administrative privileges over the operating system.
sudo	A command that allows an authorized user to execute commands with elevated privileges, subject to system configuration.
systemd	A Linux system and service manager responsible for starting, stopping, and supervising system services and other operating-system components.
sysvinit	Controls the booting and shutdown of your Linux system.
Threat model	A structured evaluation of assets, potential threats, vulnerabilities, and appropriate security controls.

Index