Pi-hole: Block Ads on Your Entire Network

Network-Wide Ad Blocking for Raspberry Pi, Home Server, and Proxmox

Nova X Lab · Pi-hole v6

Pi-hole: Block Ads on Your Entire Network
Network-Wide Ad Blocking for Raspberry Pi, Home Server, and Proxmox

Éditeur : Nova Publishing SAS, 01280 Prévessin-Moëns, France
ISBN : 978-2-489108-04-0

Contents

Introduction

How Pi-hole Works and Why You Want It

Every device in your home is leaking DNS requests to someone's server. Your phone, your smart TV, your thermostat — before any of them can load a page or check for updates, they send a DNS query to find out where to go. Those queries leave your network dozens of times per second, and by default they go to your ISP's servers, or Google, or Cloudflare. Every one of those providers can see those lookups, and depending on the provider and settings, they may be logged. Every ad network on the web depends on them.

A basic Pi-hole setup can be running in under an hour. You install it on one device in your house (a Raspberry Pi 5, a Docker container, or a Proxmox LXC), point your router at it, and every device on your network gets network-wide ad blocking and DNS privacy without you touching any of them individually. No app installs. No browser extensions. No per-device configuration. It just works.

If Pi-hole has seemed intimidating to set up, that reputation comes from one specific thing: the guides that exist online skip the steps that actually trip people up, the port 53 conflict, the router that will not accept custom DNS, the device that ignores Pi-hole anyway. This book covers all of those steps.

What DNS actually is

DNS stands for Domain Name System. Think of it as a phone book for the internet: when your browser wants to load bbc.co.uk, it first asks a DNS server "what is the IP address for bbc.co.uk?" and gets back a number like 151.101.0.81. Your browser then connects to that address. Without DNS, you would need to memorize IP addresses for every website you visit.

What most people do not realize is that this lookup happens for every resource a page loads: the page itself, images, fonts, analytics scripts, and ad servers. A single page load can trigger dozens of DNS queries. Each of those queries is sent to whatever DNS server your router is configured to use. By default, that's your ISP's server, or whichever provider your router manufacturer hardcoded.

Pi-hole is what's called a DNS sinkhole: it answers DNS queries for blocked domains with a fake 'not found' response, preventing the connection from ever being made. Pi-hole acts as your DNS server. Every device on your network sends its DNS queries to Pi-hole instead of directly to an outside DNS provider. Pi-hole checks each query

against a blocklist of known ad-serving and tracking domains. If the domain is on the list, Pi-hole returns nothing: the ad server address is never looked up, the connection is never made, and the ad is never downloaded. If the domain is not on the list, Pi-hole forwards the query to a real upstream DNS server such as Cloudflare or Google. Chapter 8 shows how to replace those upstream providers with your own recursive resolver, so no third-party DNS provider sees your queries at all.

What you will have when you are done

By the time you finish this book you will have:

- A working Pi-hole v6 instance on the hardware of your choice, Raspberry Pi 5, Docker, or Proxmox LXC
- Your router configured to use Pi-hole as the DNS server for every device on your network
- A curated blocklist that blocks ads, trackers, and malware domains with minimal false positives
- Local DNS records so your homelab services (Jellyfin, Nextcloud, Vaultwarden, Home Assistant) are accessible by name rather than by IP address
- Full DNS privacy with Unbound: no upstream provider sees your DNS queries, not Cloudflare, not Google, not your ISP. Chapter 8 covers this for readers who want complete DNS independence.
- Automated backups and Uptime Kuma monitoring so Pi-hole stays running
- A troubleshooting reference covering the sixteen most common problems and their exact fixes

A note on what Pi-hole cannot do

Pi-hole is a DNS-level blocker. It is very good at blocking ads that are served from domains separate from the content you are trying to view. It cannot block YouTube ads because YouTube serves video content and advertisements from the same domains, blocking those domains would block YouTube entirely. It cannot intercept HTTPS traffic. It is not a VPN or a firewall. These are limitations worth understanding now rather than discovering them while troubleshooting in Chapter 6.

A healthy Pi-hole setup typically blocks between 10% and 40% of DNS queries on a home network. If you see 15%, that's normal, not a misconfiguration. If you see 70%, something legitimate is probably being blocked. Chapter 6 covers how to read the statistics and interpret them correctly.

This is a v6 book

Pi-hole v6 was released in February 2025 and is a significant departure from v5. The web server changed from lighttpd to a built-in server inside the FTL binary. The configuration file changed from setupVars.conf to pihole.toml. The Docker deployment changed. The upgrade process changed. Every command, every file path, and every configuration value in this book has been written for Pi-hole v6. If you are running v5 and want to upgrade, Chapter 9 covers the migration. The current version as of April 2026 is FTL v6.6 / Web v6.5 / Core v6.4.1 (Docker tag: 2026.04.0). All procedures in this book remain fully compatible — v6.5 and v6.6 are maintenance updates with security patches and performance improvements, with no breaking changes to any setup steps covered here.

The commands in this book are shown exactly as you would type them. Every configuration value is explained. If you have a Raspberry Pi 5, a running Docker host, or a Proxmox server, you have everything you need to get started.

Chapter 1

Before You Begin: Prerequisites and Planning

Before you install Pi-hole, two decisions will shape everything that follows: which hardware you are going to run it on, and how you will give it a permanent IP address. Get these right now, and the rest of the book will proceed cleanly. Skip them and you will be troubleshooting later.

The three deployment paths

Pi-hole runs on any Linux system. This book covers the three paths used in the overwhelming majority of home setups. Each chapter from Chapter 2 onward is self-contained. Read the section for your chosen path and follow it through. Choose one installation path only: Raspberry Pi, Docker, or Proxmox LXC. Do not install Pi-hole by more than one method on the same network unless you intentionally want multiple Pi-hole servers. Most home networks need one Pi-hole instance with one permanent IP address. You don't need to read the others.

Path	Best for	Requires	Skip if
Raspberry Pi 5	New to homelab. Want a dedicated always-on device. Classic setup with no shared dependencies.	Raspberry Pi 5 (or 4), microSD card, power supply, Ethernet cable.	You already run a Docker host or Proxmox server, those paths are simpler if you do.
Docker	Already running a Linux server with Docker. Want containerized management and easy pull updates.	Linux server with Docker and Docker Compose installed. Root or sudo access.	You do not already have Docker running. The Raspberry Pi path is easier to start fresh.
Proxmox LXC	Have a Proxmox homelab. Want Pi-hole isolated in its own container without VM overhead.	A running Proxmox VE 8 installation. Familiarity with the Proxmox web UI.	You have never used Proxmox. Start with the Raspberry Pi path instead.

One principle applies across all three: Pi-hole is a network service, not an application. It needs to run at all times. If the device running Pi-hole goes down, DNS resolution stops for every device on your network. Dedicated hardware (a Pi, a container with auto-restart, a Proxmox LXC set to start at boot) is not optional. Do not run Pi-hole on a laptop you close at night.

The static IP requirement

Pi-hole must have a permanent, unchanging IP address on your network. Here is why: in the next chapter you will configure your router to send all DNS queries to Pi-hole's IP address. If that address changes (because Pi-hole's DHCP lease expired and it got a new one) your router will be pointing at the wrong address, and DNS will stop working for every device on your network. You will spend twenty minutes wondering why the internet has broken before you realize the Pi-hole IP changed.

There are two ways to give Pi-hole a permanent address. Choose one before you start your deployment chapter.

Method 1: DHCP reservation in your router (recommended)

Your router assigns IP addresses to devices via DHCP. Most routers let you reserve a specific IP address for a specific device, identified by the device's MAC address. Once reserved, the router always hands out the same address to that device whenever it connects.

This is the preferred method because the configuration lives in the router rather than on the Pi-hole device. If you reinstall Pi-hole or replace the hardware, the IP reservation stays in place automatically. You will find this setting in most routers under a name like "DHCP Reservations", "Static DHCP", "Address Reservation", or "Fixed DHCP". You need the device's MAC address, a unique 12-character hardware identifier that looks like `dc:a6:32:a1:b2:c3`. The steps for finding the MAC address are in each deployment chapter.

Method 2: Static IP configured on the Pi-hole device

The alternative is to configure Pi-hole's network interface to use a fixed IP address directly on the device, bypassing DHCP entirely. This is covered in each deployment chapter for readers whose routers do not support DHCP reservations.

A word of caution: if you choose this method, pick an IP address outside your router's DHCP range to avoid an address conflict. For example, if your router hands out addresses in the range 192.168.1.100–192.168.1.200, use something like 192.168.1.5