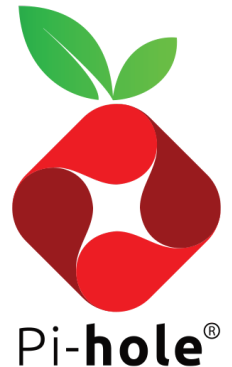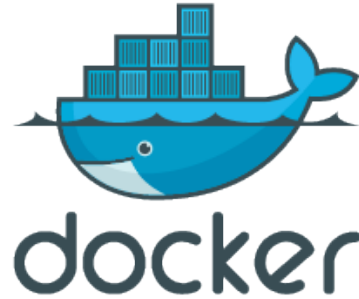# Pihole Setup Guide

Setup your own Pihole Adblocker!

- Getting Started

  - Overview: What is Pihole?

  - Step-by-Step Install Guide for Pihole with Traefik on Docker Swarm

- Other Informations

  - Environment Variables for Pi-hole

  - Configure Adlists

# Getting Started

# Overview: What is Pihole?

## Overview: What is Pihole?

**Pi-hole** is a network-wide ad blocker that acts as a DNS sinkhole, filtering out unwanted content, advertisements, and tracking at the network level. It's typically installed on small devices like a Raspberry Pi but can be deployed on any Linux-based system.
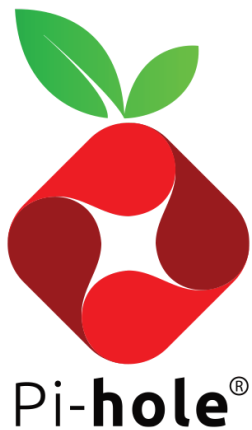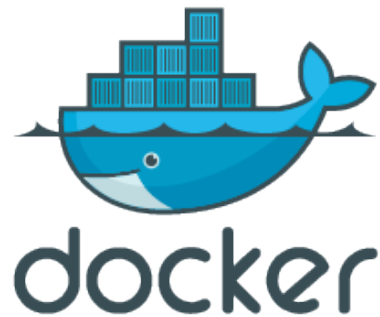
# Key Features:

1. **Ad Blocking**: Blocks ads at the DNS level, preventing them from being loaded on any device connected to the network.
2. **Privacy Protection**: Blocks tracking scripts and domains, enhancing user privacy across devices.
3. **Network-wide Filtering**: Applies ad-blocking to all devices on the network, regardless of platform or browser.

4. **Customizable Blocklists**: Supports user-defined blocklists, allowing for granular control over what gets blocked.
5. **Web Interface**: Provides an easy-to-use web dashboard for managing blocklists, monitoring DNS requests, and viewing statistics.
6. **Low Resource Usage**: Can run efficiently on low-power devices like Raspberry Pi without affecting network performance.
7. **Whitelist/Blacklist Control**: Allows users to add specific domains to whitelist or blacklist for customized filtering.
8. **DNS Performance**: Acts as a local DNS resolver, improving browsing speed by reducing the need for external DNS queries.

Pi-hole is widely used for ad-blocking, network security, and improving overall internet performance.

# Step-by-Step Install Guide for Pihole with Traefik on Docker Swarm

This guide will walk you through setting up **Pi-hole**, a powerful network-wide ad blocker, on a Docker Swarm with **Traefik** as the reverse proxy. We will use Docker to deploy Pi-hole and Traefik for secure and managed access.

# Prerequisites

- A Docker Swarm environment with at least one Linux node. -> Check this Article
- Traefik set up as a reverse proxy (Assuming a Traefik stack is already configured). -> Check this Article
- A mounted shared volume with **GlusterFS** for data persistence. -> Check this Article
- A domain or local DNS entry pointing to your Pi-hole service (e.g., `pihole.local` ).
- Access to Docker CLI and necessary credentials.

# Step 1: Prepare the Directory Structure

To ensure persistent storage for Pi-hole, we will create directories on the shared GlusterFS mount:

```
mkdir /mnt/glustermount/data/pihole_data
mkdir /mnt/glustermount/data/pihole_data/dns
mkdir /mnt/glustermount/data/pihole_data/etc
```

These directories will store Pi-hole's configuration files and DNS settings.

# Step 2: Create the `docker-compose.yml` File

In your working directory, create a `docker-compose.yml` file with the following content:

```
version: '3'

services:
  pihole:
    networks:
      - management_net  # For management via Traefik
```

```yaml
    image: pihole/pihole:latest
    ports:
      - "53:53/tcp"
      - "53:53/udp"
      - "888:80"
    environment:
      TZ: 'Europe/Zurich'
      WEBPASSWORD: '${PIHOLE_PASSWORD}'
    volumes:
      - '/mnt/glustermount/data/pihole_data/etc:/etc/pihole'
      - '/mnt/glustermount/data/pihole_data/dns:/etc/dnsmasq.d'
    restart: unless-stopped
    deploy:
      mode: replicated
      replicas: 1
      placement:
        constraints: [node.platform.os == linux]
      labels:
        - 'traefik.enable=true'
        - "traefik.http.services.pihole.loadbalancer.server.port=888"
networks:
  management_net:
    external: true
```

- **Volumes**: The `/mnt/glustermount/data/pihole_data/etc` and `/mnt/glustermount/data/pihole_data/dns` directories are used to persist Pi-hole data.
- **Ports**: Port `53` for DNS queries is exposed on both TCP and UDP.
- **Environment Variables**: Set the timezone (`TZ`) and the Pi-hole admin password ( `WEBPASSWORD` ).
- **Traefik Labels**: These labels enable Pi-hole to be accessible through Traefik via the domain `pihole.local` using HTTPS.

# Step 3: Deploy the Stack

**Deploy the Stack**: Use the following command to deploy Pi-hole with Traefik in Docker Swarm.

```
docker stack deploy -c docker-compose.yml pihole
```

## Step 4: Access Pi-hole Web Interface

After the deployment completes, you can access Pi-hole's admin interface by navigating to `https://pihole.local/admin` in your browser. Log in with the password you specified in the environment variable (`PIHOLE_PASSWORD`).

## Step 5: Update DNS Settings

To start using Pi-hole, configure your router or devices to use your Pi-hole instance as the DNS server. The IP address of your Pi-hole service is the one assigned by Docker, which you can retrieve using:

```
docker service ps pihole_pihole
```

OR if you use **Keepalived** you can use your VIP.

# Conclusion

Setting up Pi-hole in a Docker Swarm environment with Traefik as a reverse proxy provides network-wide ad blocking, improving privacy and performance for all devices connected to your network. By leveraging Docker Swarm and Traefik, you achieve high availability, flexibility, and ease of management for your Pi-hole deployment.

# Other Informations

# Environment Variables for Pi-hole

| Variable | Default Value | Description |
|---|---|---|
| **PIHOLE_DNS_** | 8.8.8.8;8.8.4.4 | Upstream DNS servers, separated by `;`. Supports custom ports (e.g., `127.0.0.1#5053`). DNS servers added via the web interface will be overwritten on restart. |
| **DNSSEC** | false | Enable DNSSEC support (`true` or `false`). |
| **DNS_BOGUS_PRIV** | true | Prevents forwarding reverse lookups for private ranges. |
| **DNS_FQDN_REQUIRED** | true | Prevents forwarding of non-FQDNs (Fully Qualified Domain Names). |

| Variable | Default Value | Description |
| --- | --- | --- |
| **REV_SERVER** | false | Enables DNS conditional forwarding for local device name resolution. |
| **REV_SERVER_DOMAIN** | unset | Set the domain of the local network router if conditional forwarding is enabled. |
| **REV_SERVER_TARGET** | unset | IP of the local network router when conditional forwarding is enabled. |
| **REV_SERVER_CIDR** | unset | Reverse DNS zone (e.g., `192.168.0.0/24`) for conditional forwarding. |
| **DHCP_ACTIVE** | false | Enable DHCP server (`true` or `false`). |
| **DHCP_START** | unset | Start IP for DHCP server (if DHCP is enabled). |
| **DHCP_END** | unset | End IP for DHCP server (if DHCP is enabled). |
| **DHCP_ROUTER** | unset | Router (gateway) IP for the DHCP server (if DHCP is enabled). |
| **DHCP_LEASETIME** | 24 | Lease time for DHCP (in hours). |
| **PIHOLE_DOMAIN** | lan | Domain name sent by the DHCP server. |
| **DHCP_IPv6** | false | Enable DHCP IPv6 support (`true` or `false`). |
| **DHCP_rapid_commit** | false | Enable DHCPv4 rapid commit. |
| **VIRTUAL_HOST** | ${HOSTNAME} | Sets the virtual host for web access (e.g., `http://pi.hole/admin`). |
| **IPv6** | true | Disables IPv6 configuration when set to `false` (helpful for Unraid). |
| **TEMPERATUREUNIT** | c | Sets temperature unit (`c`: Celsius, `k`: Kelvin, or `f`: Fahrenheit). |
| **WEBUIBOXEDLAYOUT** | boxed | Use boxed or traditional layout for the web interface. |
| **QUERY_LOGGING** | true | Enables or disables query logging. |
| **WEBTHEME** | default-light | User interface theme (options: `default-dark`, `default-light`, `default-auto`, etc.). |
| **WEBPASSWORD_FILE** | unset | Set admin password via Docker secrets. Ignored if `WEBPASSWORD` is set. |

# Advanced Variables

| Variable | Default Value | Description |
|---|---|---|
| **INTERFACE** | unset | NIC interface for DNS or DHCP services. |
| **DNSMASQ_LISTENING** | unset | Listening behavior (`local`, `all`, `single`). |
| **WEB_PORT** | unset | Custom web interface port (may affect the "blocked" page functionality). |
| **WEB_BIND_ADDR** | unset | Bind address for the web interface. |
| **SKIPGRAVITYONBOOT** | unset | Skip updating Gravity Database on boot (set to `1` to skip). |
| **CORS_HOSTS** | unset | List of FQDNs allowed for CORS (comma-separated). |
| **CUSTOM_CACHE_SIZE** | 10000 | Sets cache size for `dnsmasq`. Ignored if DNSSEC is enabled. |
| **FTL_CMD** | no-daemon | Customize `dnsmasq` options (e.g., `no-daemon -- --dns-forward-max 300`). |
| **FTLCONF_[SETTING]** | unset | Customize `pihole-FTL.conf` settings (e.g., `FTLCONF_LOCAL_IPV4`). |

# Experimental Variables

| Variable | Default Value | Description |
|---|---|---|
| **DNSMASQ_USER** | unset | Change the user that `FTLDNS` runs as (`pihole` or `root`). |
| **PIHOLE_UID** | 999 | Override Pi-hole's default user ID. |
| **PIHOLE_GID** | 999 | Override Pi-hole's default group ID. |
| **WEB_UID** | 33 | Override the `www-data` user ID. |
| **WEB_GID** | 33 | Override the `www-data` group ID. |
| **WEBLOGS_STDOUT** | 0 | Redirects web logs to stdout when set to `1`. |

# Configure Adlists

To enhance Pi-hole's ability to block unwanted ads, trackers, and malicious content, you can add custom adlists. Below is a step-by-step guide to add adlists in Pi-hole, followed by a comprehensive list of popular adlists.

## Step 1: Access Pi-hole's Web Interface

1. Open a web browser and navigate to your Pi-hole admin page. If you set up Pi-hole to run at a specific IP and port (e.g., `888`), you can visit the following URL:

   http://my-server-ip:888/admin/groups-adlists.php

2. Log in using your Pi-hole admin password.

## Step 2: Navigate to the Adlists Section

1. Once logged in, go to the **Group Management** tab.
2. Select the **Adlists** option from the sidebar.

## Step 3: Add New Adlists

1. In the **Adlists** page, you'll see an option to **Add a new adlist**.
2. Enter the URL of the adlist you wish to add in the "Address" field.
3. Add a comment or label for future reference (e.g., `Default`).
4. Click on **Add**.

Adlists with **7.303.876 Domain**s on the Adlists:

https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts https://big.oisd.nl

https://nsfw.oisd.nlhttps://raw.githubusercontent.com/PolishFiltersTeam/KADhosts/master/KADhosts.txt

https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.Spam/hosts

https://v.firebog.net/hosts/static/w3kbl.txt https://raw.githubusercontent.com/matomo-org/referrer-spam-blacklist/master/spammers.txt https://someonewhocares.org/hosts/zero/hosts

https://raw.githubusercontent.com/VeleSila/yhosts/master/hosts https://winhelp2002.mvps.org/hosts.txt

https://v.firebog.net/hosts/neohostsbasic.txt https://raw.githubusercontent.com/RooneyMcNibNug/pihole-stuff/master/SNAFU.txt https://paulgb.github.io/BarbBlock/blacklists/hosts-file.txt https://adaway.org/hosts.txt

https://v.firebog.net/hosts/AdguardDNS.txt https://v.firebog.net/hosts/Admiral.txt

https://raw.githubusercontent.com/anudeepND/blacklist/master/adservers.txt

https://v.firebog.net/hosts/Easylist.txt

https://pgl.yoyo.org/adservers/serverlist.php?hostformat=hosts&showintro=0&mimetype=plaintext

https://raw.githubusercontent.com/FadeMind/hosts.extras/master/UncheckyAds/hosts

https://raw.githubusercontent.com/bigdargon/hostsVN/master/hosts

https://raw.githubusercontent.com/jdlingyu/ad-wars/master/hosts https://v.firebog.net/hosts/Easyprivacy.txt

https://v.firebog.net/hosts/Prigent-Ads.txt

https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.2o7Net/hosts

https://raw.githubusercontent.com/crazy-max/WindowsSpyBlocker/master/data/hosts/spy.txt

https://hostfiles.frogeye.fr/firstparty-trackers-hosts.txt https://www.github.developerdan.com/hosts/lists/ads-and-tracking-extended.txt https://raw.githubusercontent.com/Perflyst/PiHoleBlocklist/master/android-tracking.txt

https://raw.githubusercontent.com/Perflyst/PiHoleBlocklist/master/SmartTV.txt

https://raw.githubusercontent.com/Perflyst/PiHoleBlocklist/master/AmazonFireTV.txt

https://gitlab.com/quidsup/notrack-blocklists/raw/master/notrack-blocklist.txt

https://raw.githubusercontent.com/DandelionSprout/adfilt/master/Alternate%20versions%20Anti-Malware%20List/AntiMalwareHosts.txt https://osint.digitalside.it/Threat-Intel/lists/latestdomains.txt

https://v.firebog.net/hosts/Prigent-Crypto.txt

https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.Risk/hosts

https://bitbucket.org/ethanr/dns-blacklists/raw/8575c9f96e5b4a1308f2f12394abd86d0927a4a0/bad_lists/Mandiant_APT1_Report_Appendix_D.txt

https://phishing.army/download/phishing_army_blocklist_extended.txt https://gitlab.com/quidsup/notrack-blocklists/raw/master/notrack-malware.txt https://v.firebog.net/hosts/RPiList-Malware.txt

https://v.firebog.net/hosts/RPiList-Phishing.txt https://raw.githubusercontent.com/Spam404/lists/master/main-blacklist.txt https://raw.githubusercontent.com/AssoEchap/stalkerware-indicators/master/generated/hosts

https://urlhaus.abuse.ch/downloads/hostfile/ https://malware-filter.gitlab.io/malware-filter/phishing-filter-hosts.txt

https://v.firebog.net/hosts/Prigent-Malware.txt https://zerodot1.gitlab.io/CoinBlockerLists/hosts_browser

https://raw.githubusercontent.com/chadmayfield/my-pihole-blocklists/master/lists/pi_blocklist_porn_top1m.list

https://v.firebog.net/hosts/Prigent-Adult.txt

https://raw.githubusercontent.com/anudeepND/blacklist/master/facebook.txt

https://raw.githubusercontent.com/xxcriticxx/.pl-host-file/master/hosts.txt

https://raw.githubusercontent.com/Goooler/1024_hosts/master/hosts https://tgc.cloud/downloads/hosts.txt

# Conclusion

By adding these adlists, you can extend Pi-hole's capabilities to block a wider variety of ads, trackers, and malicious content. This can significantly improve browsing speed, privacy, and security across your network. Make sure to regularly update Pi-hole's Gravity to keep the adlists up to date.